DataStax

Developers

Eliminate Kubernetes Complexity For Developers

Edward Ionel and Rags Srinivas

01

Intro

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices =
Quarkus+Cassandra

06

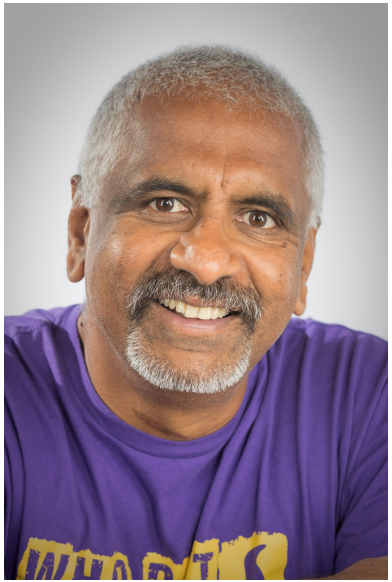
Live Coding

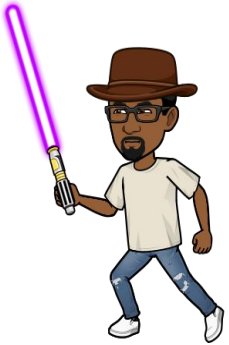
07

Resources

Agenda

Developer Advocate





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

@rags

@ragsns

@ragss

Raghavan "Rags" Srinivas

Developer Advocate

@edward-ionel

@eionel

@loneleward

- Marketer/Developer
- Community Builder
- Team Lens
- Eliminate Kubernetes complexity
  - = My goal

Edward Ionel

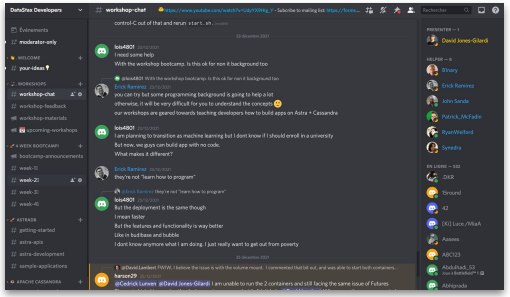
DataStax Developers Crew

Livestream: youtube.com/DataStaxDevs

YouTube

Twitch

Questions: https://dtsx.io/discord

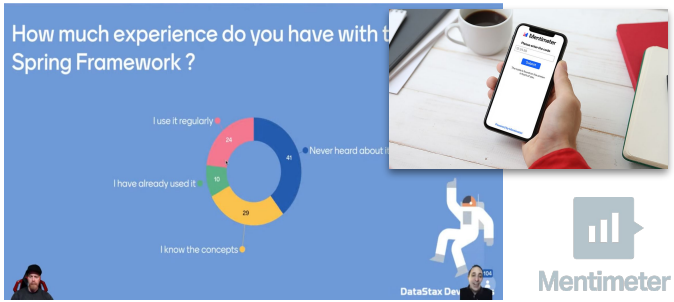


YouTube
(with nighbot)

Discord
(#workshop-chat)

!discord

Games and quizzes: menti.com



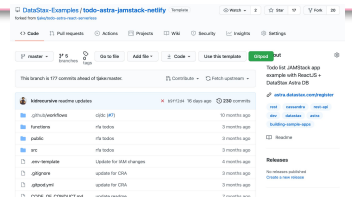
Mentimeter

!menti

Live Sessions

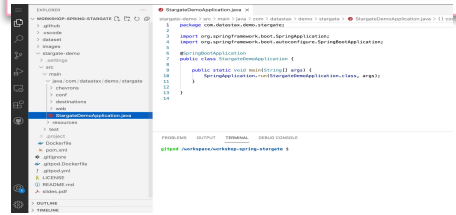
Nothing to install !

Source code + exercises + slides



GitHub

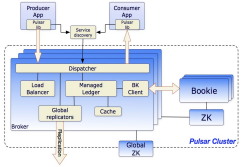
IDE



Gitpod

Event Streaing

PULSAR

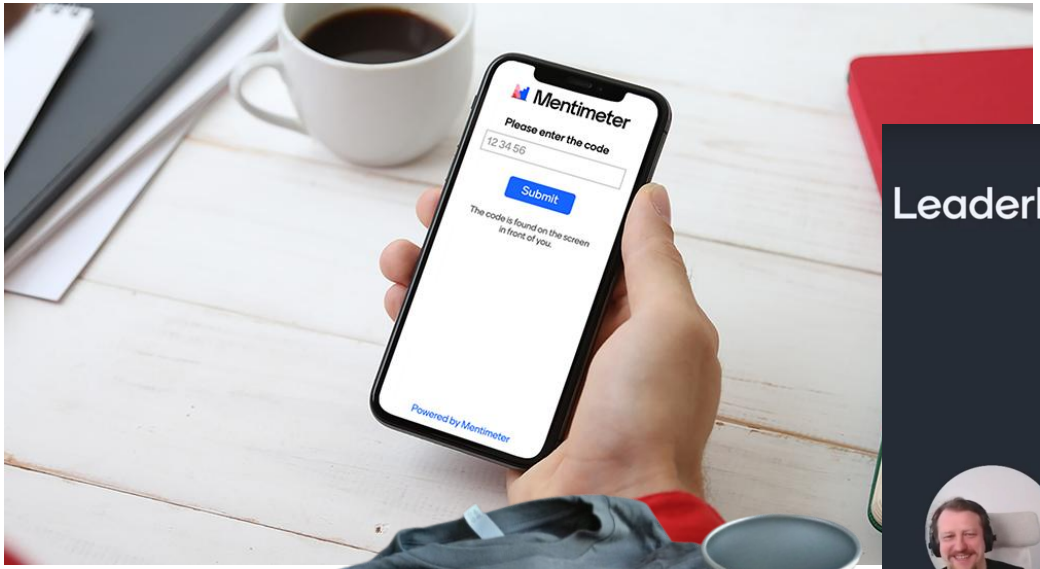


DataStax Astra STREAMING

Database + REST

DataStax Astra DB

Hands-On Housekeeping

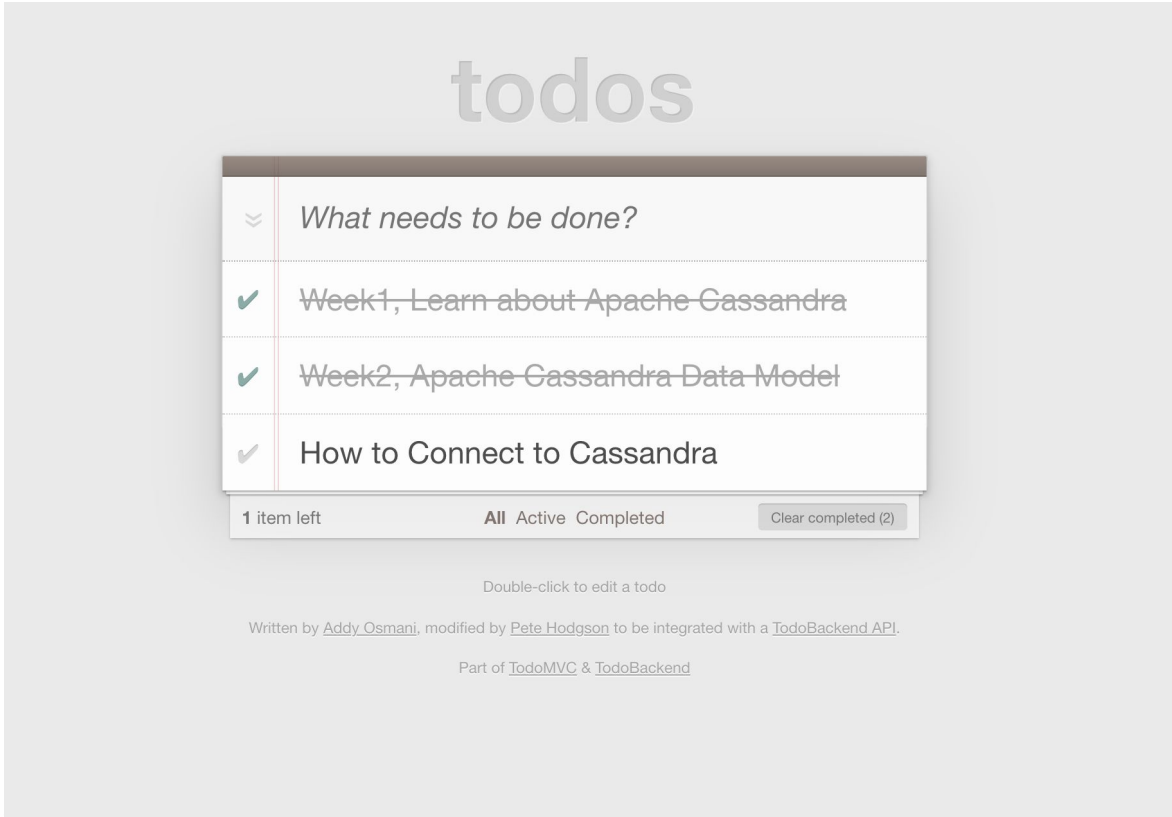


`menti.com` ⇒ enter code
Don't answer in YT chat
Look at phone (not at YT)

Quiz on "Menti" !

Objective

- Creating an inner development workflow as a developer
- Tools that help a Kubernetes developer, particularly the inner loop
- Eliminate the need for developers mastering kubectl



Today's challenge: Todo App

01

Lens

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices =
Quarkus+Cassandra

06

Live Coding

07

Resources

Agenda

The Way The World Runs Kubernetes

Stats Updated: June 9, 2022

#1
KUBERNETES IDE

750k+
UNIQUE USERS

19.5k+
GITHUB STARGAZERS

LENS USERS ARE FROM SOME OF THE MOST ICONIC ORGANIZATIONS IN THE WORLD

1M+
CLUSTERS

3.5M+
MACHINES

Lens Desktop Overview

- All you need to take control of Kubernetes
- Standalone desktop app; no in-cluster components required; connection via K8S APIs
- Provides complete situational awareness of all clusters and workloads on-premises, public clouds and beyond
- Designed for devs & ops who need to work with Kubernetes on a daily basis
    - Lower the barrier of entry for people just getting started
    - Radically improve productivity and quality of life for people with more experience
- Works with any certified Kubernetes distributions
- Built on open source. Free of charge.

Free of charge | Open Source | MIT license | Built binaries available for:

How Value Is Measured With Lens?

Recent Lens User Survey Results; 522 Responses

1 We Accelerate Kubernetes Time to Value

95%

Users of Lens Learn and Adopt Kubernetes Faster

2 We Solve Kubernetes Complexity

75%

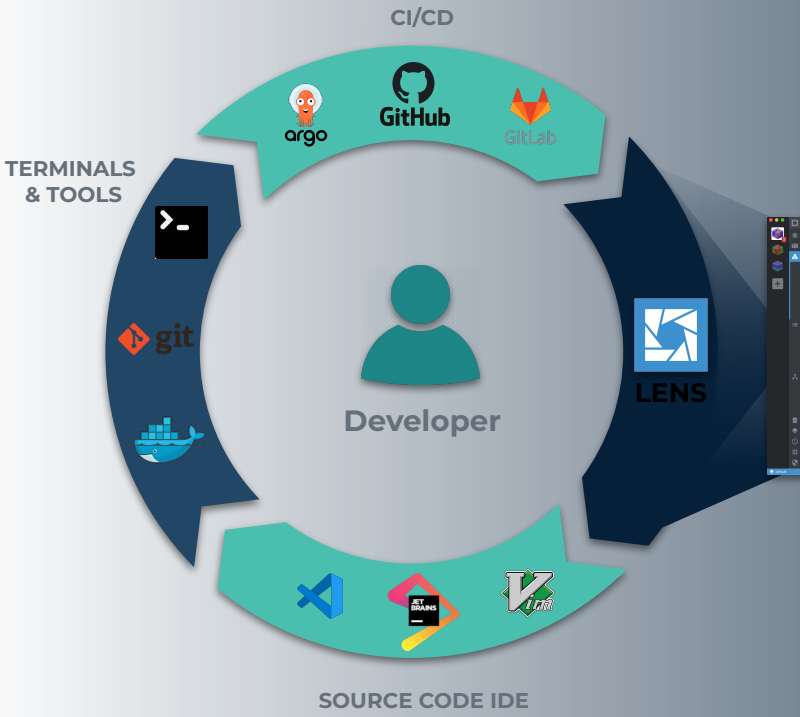
Less Tools Needed When Working With Lens

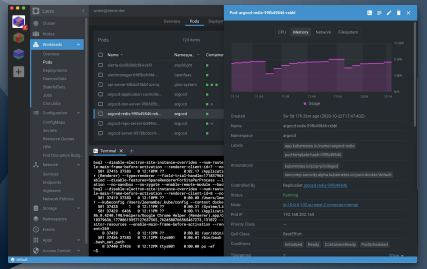
3 We Increase Kubernetes Productivity

20%

Time Saved Per Day / Week When Using Lens

Development with Lens

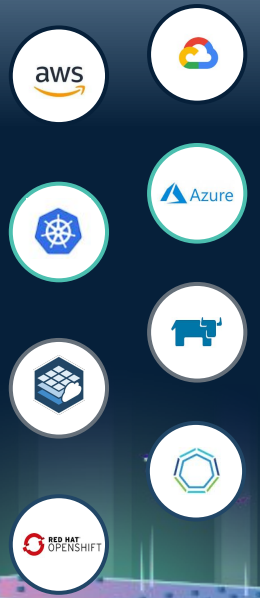




LENS IDE

- Deployment
- Observability
- Testing
- Troubleshooting

Any CNCF Certified K8s Distribution

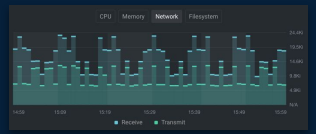


Operations with Lens

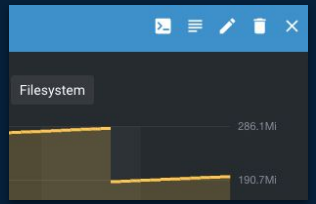
Kubernetes SRE

Daily Operations

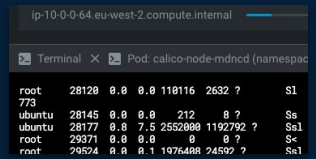
Health and Status



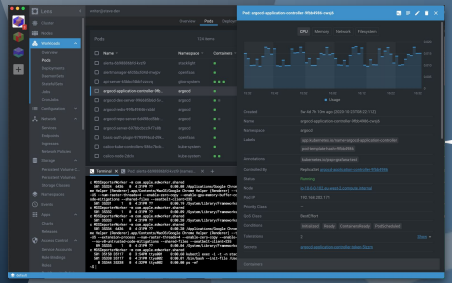
Events



Lifecycle Management



Trouble shooting



LENS IDE

Capacity

Resources, Images

Nodes, Pods, Operators

Network, API, Scheduler, Controller Logs

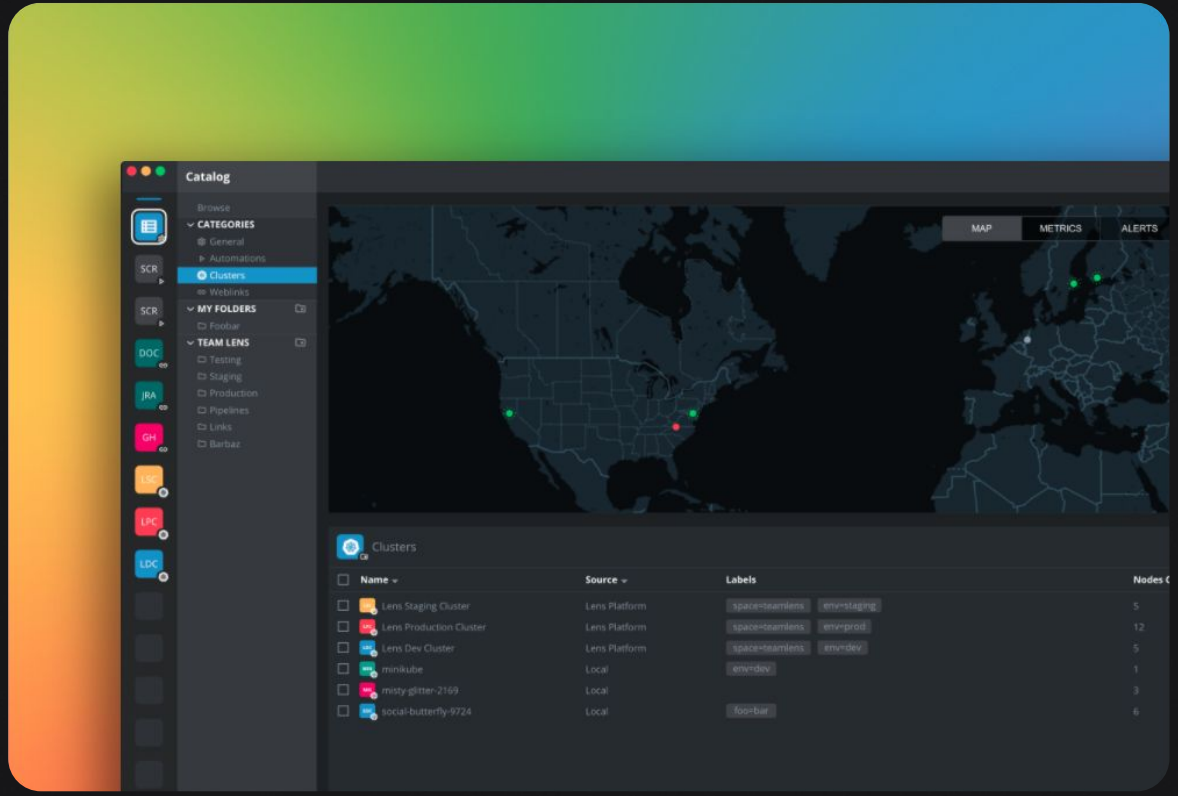
Resource Quota

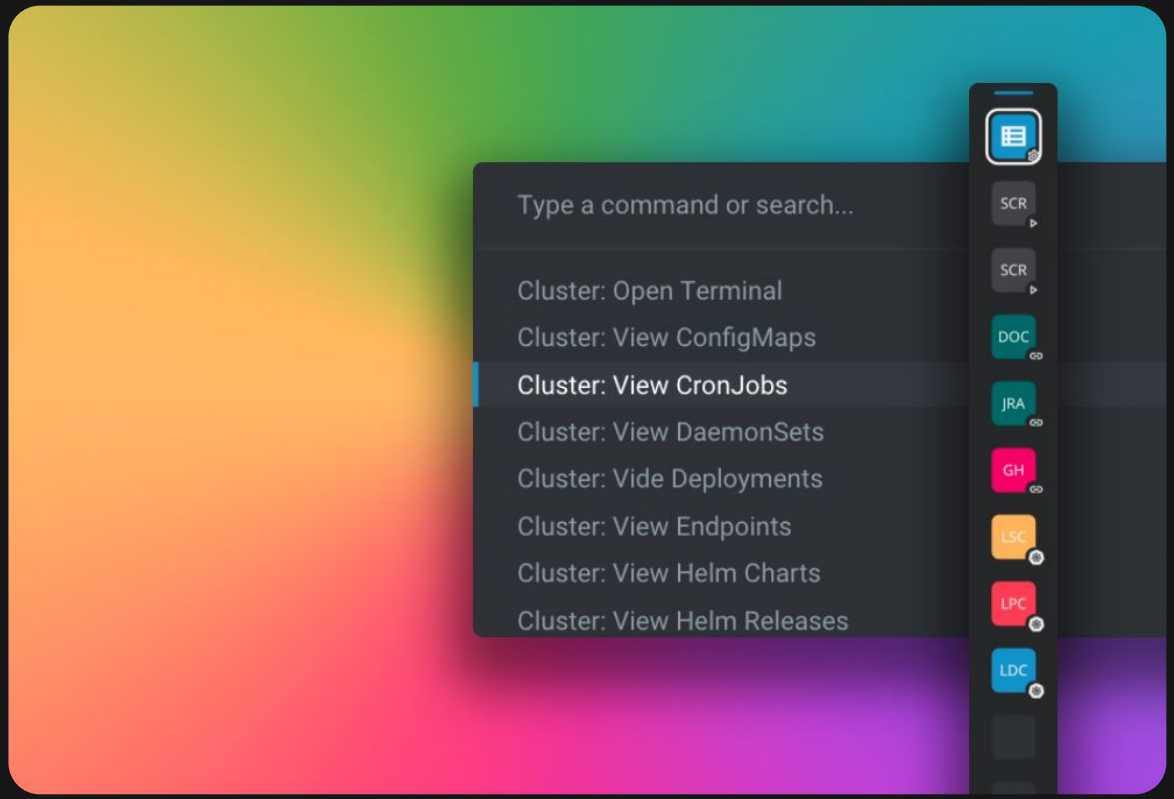
Features at a Glance

Your Cloud Native Technology Stack. Accessible.

Unified Catalog. Bring all your clusters, services, workloads, tools, automations, and related resources together for easy access.

Browse & Organize. Use search, filters, categories and labels to access the resources you need to work with easier than ever before.





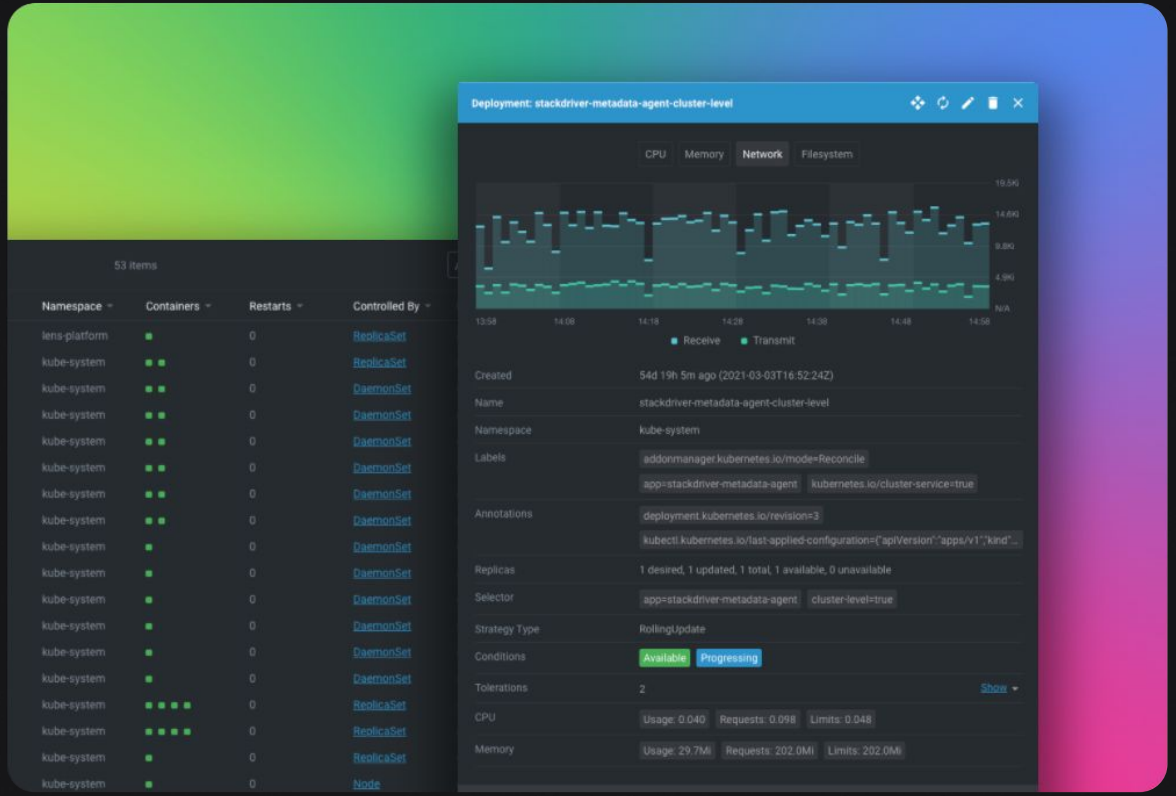
Get What You Want. Fast.

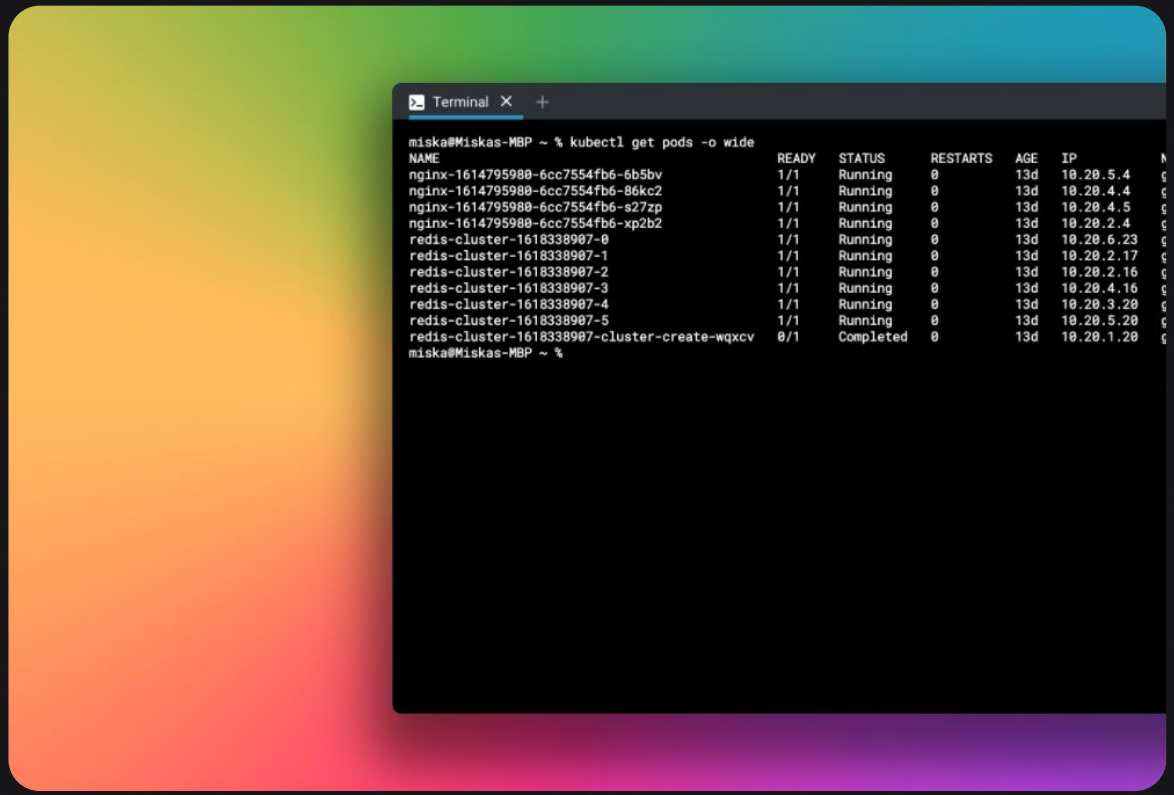
Hotbars. The main navigation, allowing users to build their “workflows” and “automation” within the desktop application. Users can customize items in the Hotbar by assigning different labels, colors, and icons for easy recall.

Command Palette. Command Palette allows users to perform specific keyboard shortcuts to make the most common tasks easier. Improving accessibility and efficiency when working with Lens.

See Everything. In Right Context. Real Time.

Built-In Visualizations. Lens integrates with Prometheus to visualize and see trends in resource usage metrics, including CPU, memory, network and disk, with the total capacity, actual usage, requests and limits. Detailed visualization is automatically generated for each k8s resource.



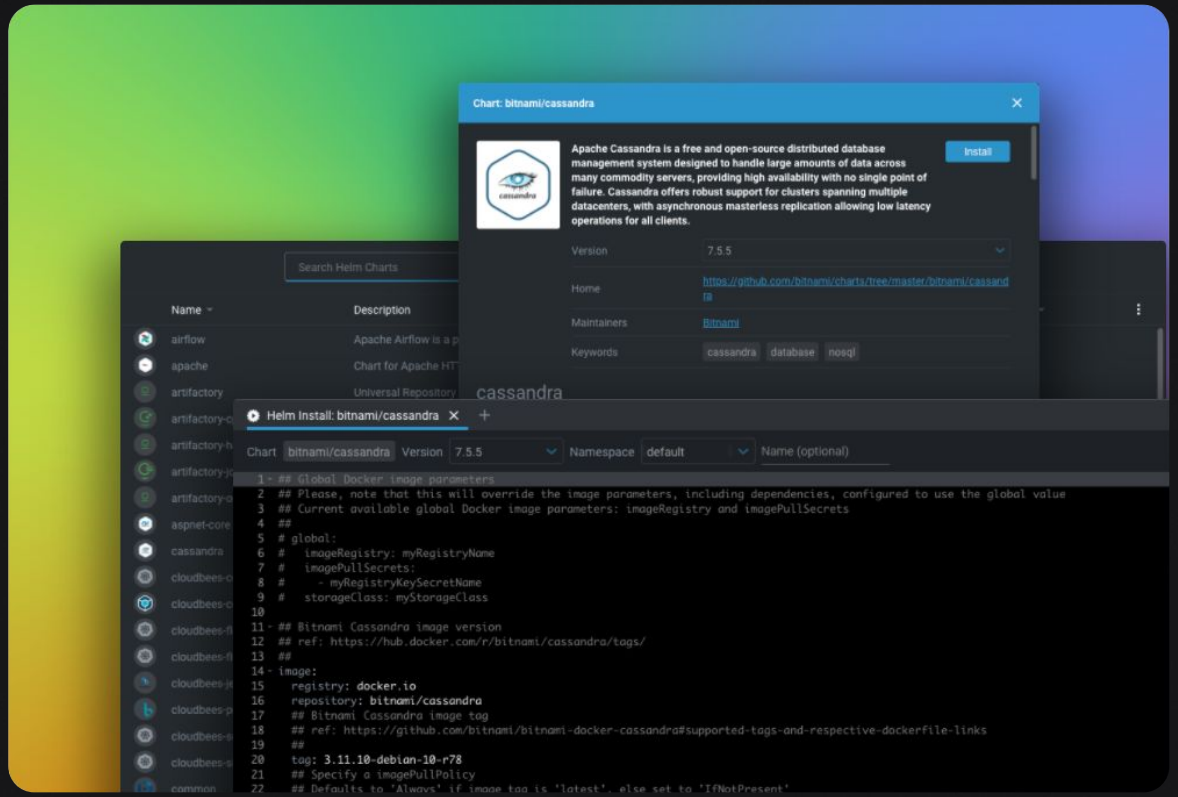


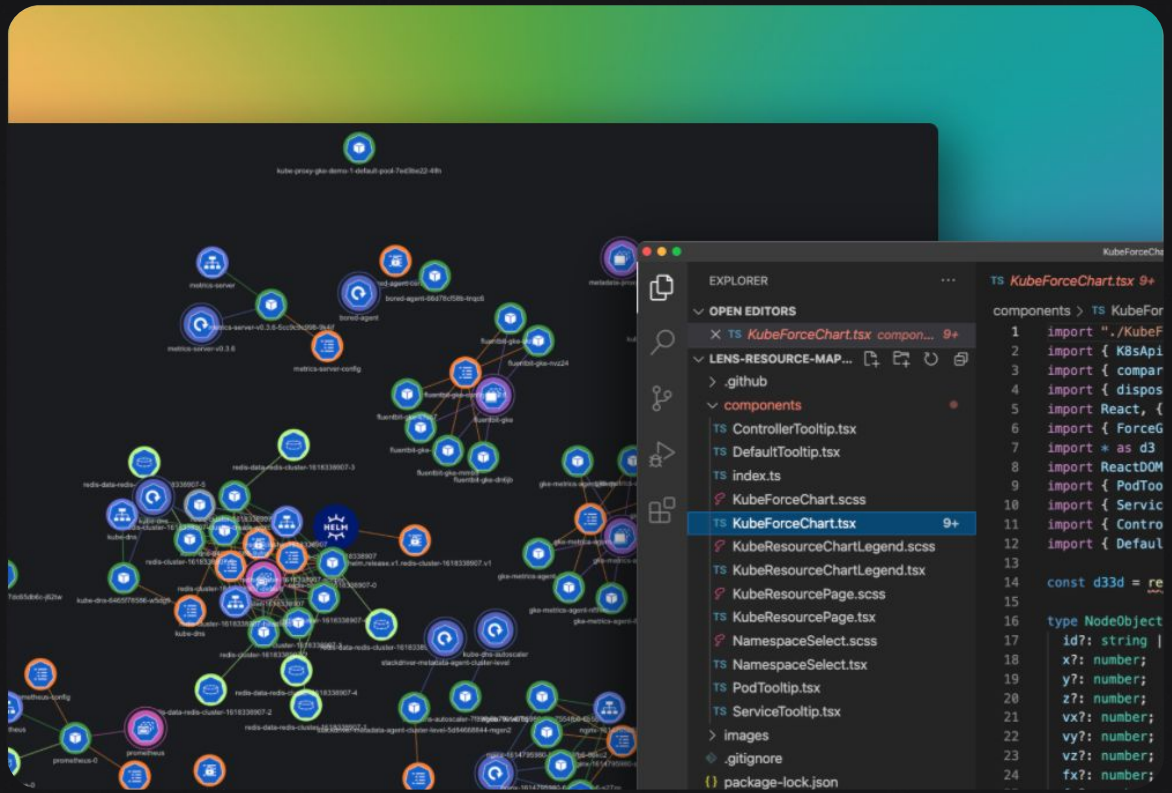
Full Power at Your Fingertips. Always.

Smart Terminal. The smart terminal comes with kubectl and helm, automatically syncing the version of kubectl to match the currently selected K8S cluster API version. Lens will automatically assign the kubeconfig context to match the currently selected K8s cluster.

All Cloud Native Technologies. Deployed in a Click.

Helm Charts. Lens comes with Helm charts management allowing for the discovery and fast deployment of thousands of publicly available Helm charts and management of your own repositories. Explore installed Helm charts and their revisions and upgrade with a single click.





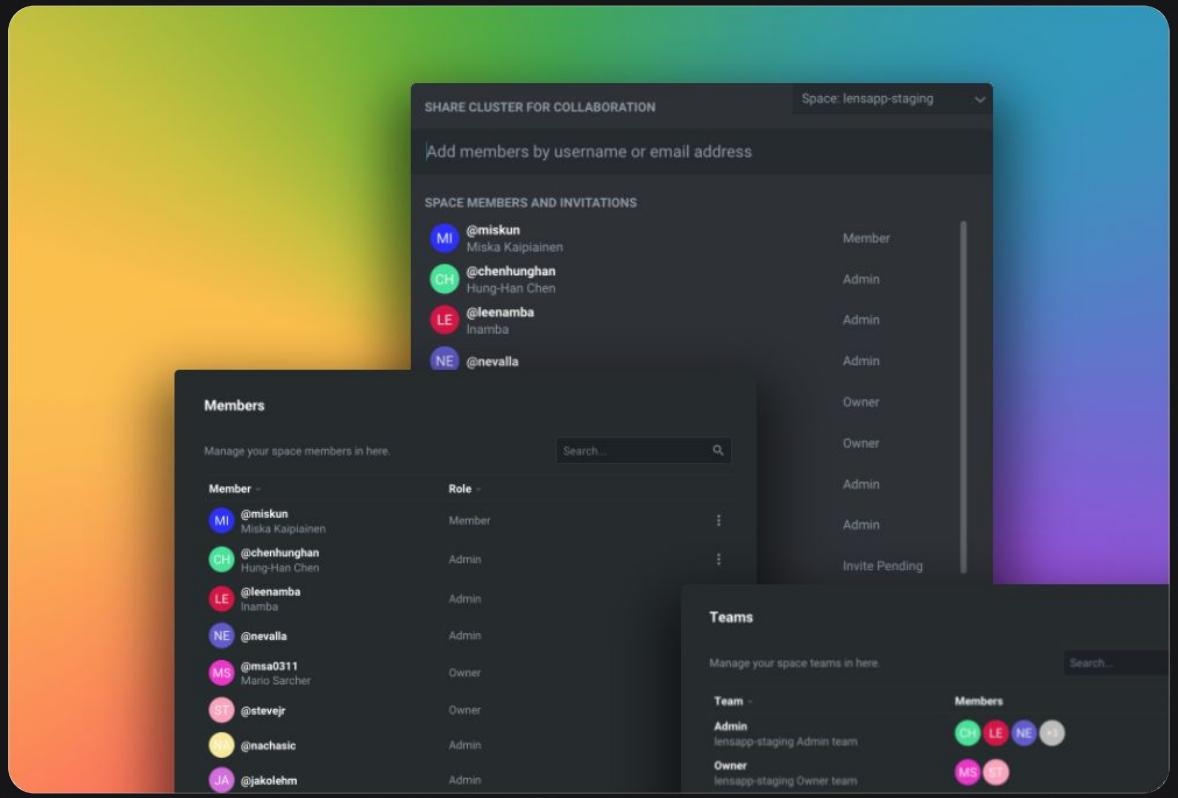
No Limits. Expand. Just the Way You Like It.

Extensions. Easily add Lens extensions from the community and cloud native ecosystem vendors or build your own. Lens Extensions are used to add custom functionality and services to accelerate development workflows for all the technologies that integrate with Kubernetes and other cloud-native technologies.

You are not alone. Join the party. Share.

Team Management. Team Space is for collaboration and centralized access. Assign teams and manage Kubernetes RBAC with ease. Take the amazing Lens experience and share it with your team!

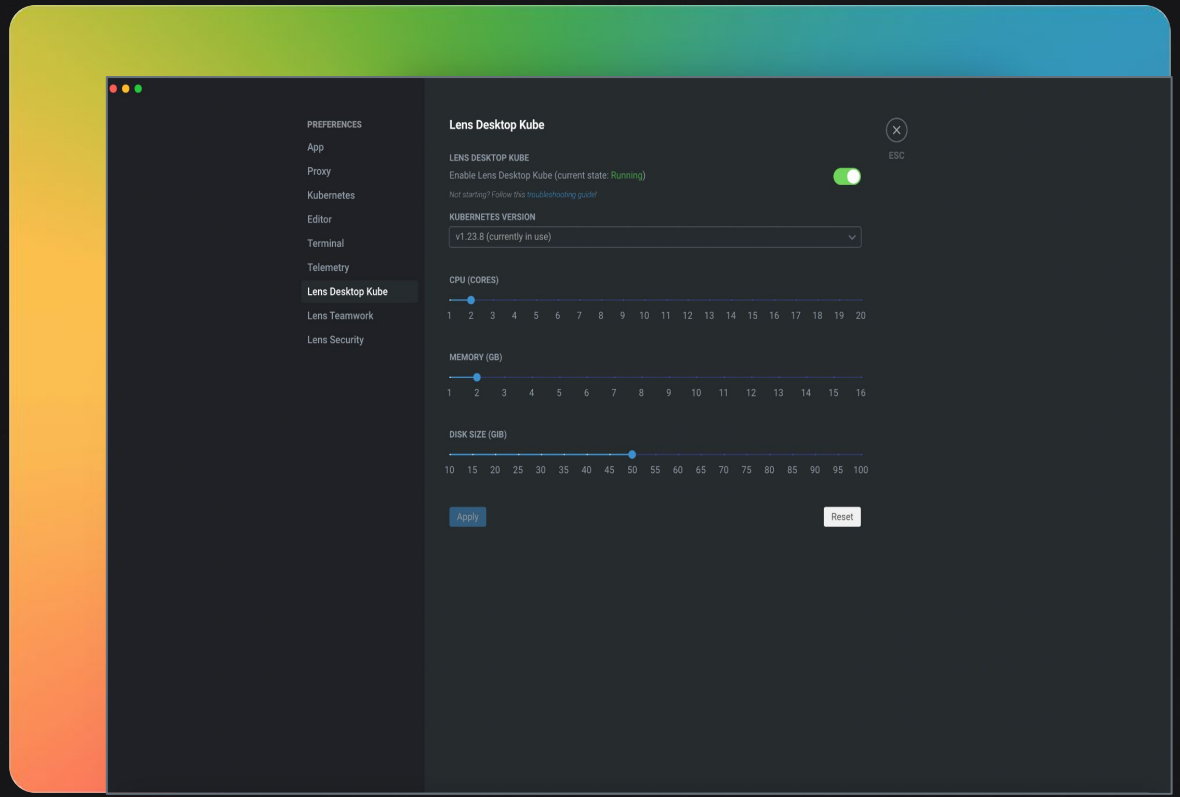
Cloud Service. Spaces is made available to Lens users as centralized cloud based service operated by Team Lens. Signup via Lens application to get started for free!



Lens Desktop Kubernetes

Kubernetes for Everyone: Provides a personal Kubernetes cluster to everyone. From people just getting started to seasoned cloud native developers. Zero hassle!

Intelligent Auto-Shutdown: If you forget to shut down your cluster, we will do it for you. Avoid searching for unused clusters or being surprised by large cloud provider bills. Pay only for the actual use, without any extra charges.



Resources

1. Download Lens
    a. https://k8slens.dev/
2. Getting Started guide
    a. https://medium.com/k8slens/getting-started-with-lens-763290e3b59c

01

Intro/Lens

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices = Quarkus
+ Cassandra

06

Live Coding

07

Resources

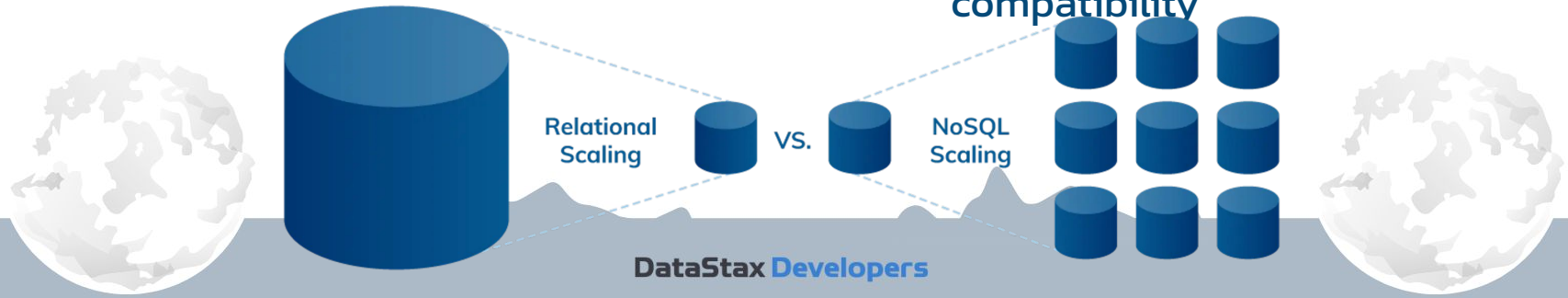
Agenda

Origin of the term “NoSQL”

- Meetup name on June 11, 2009 in San Francisco
  - Catchy hashtag intended to refer to databases like BigTable and DynamoDB
  - Meetup presentations: Cassandra, MongoDB, CouchDB, HBase, Voldemort, Dynomite, and Hypertable
- Sometimes referred to “Not only SQL”

Relational vs. NoSQL

- Relational
  - Standard relational data model and language SQL
  - ACID transactions
  - Integration database
  - Designed for a single machine
  - Hard to scale
  - Impedance mismatch
- NoSQL
  - Variety of data models and languages
  - Lower-guarantee transactions
  - Application database
  - Designed for a cluster
  - Easy to scale
  - Better database-app compatibility



The CAP Theorem

Always responds,
may not always return
the most recent write

Availability

Consistency

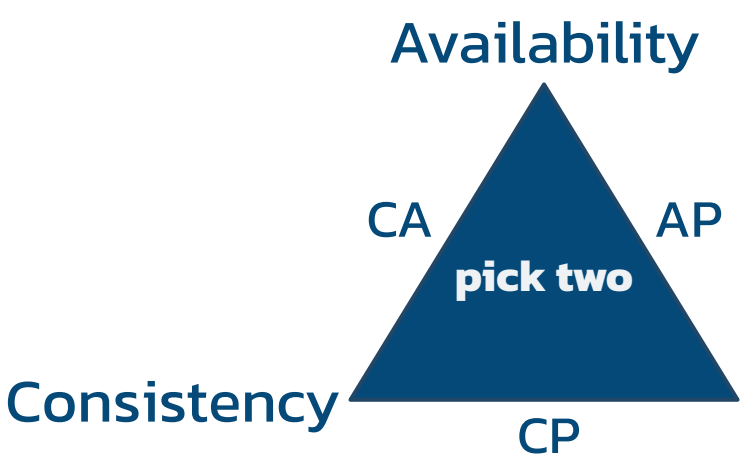
Every read receives
the most recent write
or an error

Partition tolerance

Operates in the
presence of network
partition failures

The CAP Theorem

DataStax Graph



Astra DB

cassandra

DynamoDB

riak

redis

BigTable

APACHE HBASE

mongoDB.

Couchbase

01

Intro/Lens

02

NoSQL

03

Cassandra/Astra/ k8ssandra

04

Quarkus

05

Microservices (Quarkus + Cassandra)

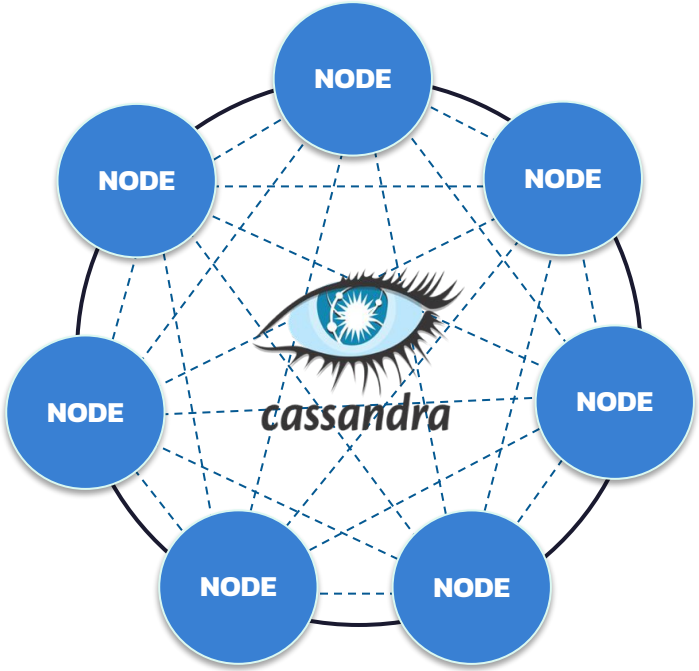
06

Live Coding

07

Resources

Agenda



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

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.

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;

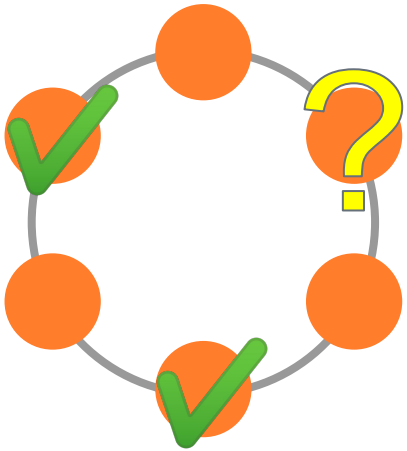
```
PreparedStatement pstmt = session.prepare(
    "INSERT INTO product (sku, description) VALUES (?, ?)"
);
pstmt.setConsistencyLevel(ConsistencyLevel.ONE);
```

```
cqlsh> CONSISTENCY

Current consistency level is QUORUM.

cqlsh> CONSISTENCY ALL

Consistency level set to ALL.
```



Is Cassandra AP or CP?

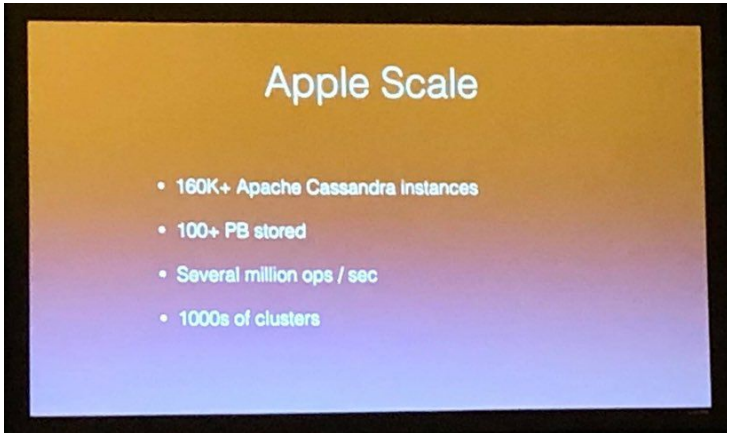
Apache Cassandra @ Netflix

- 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

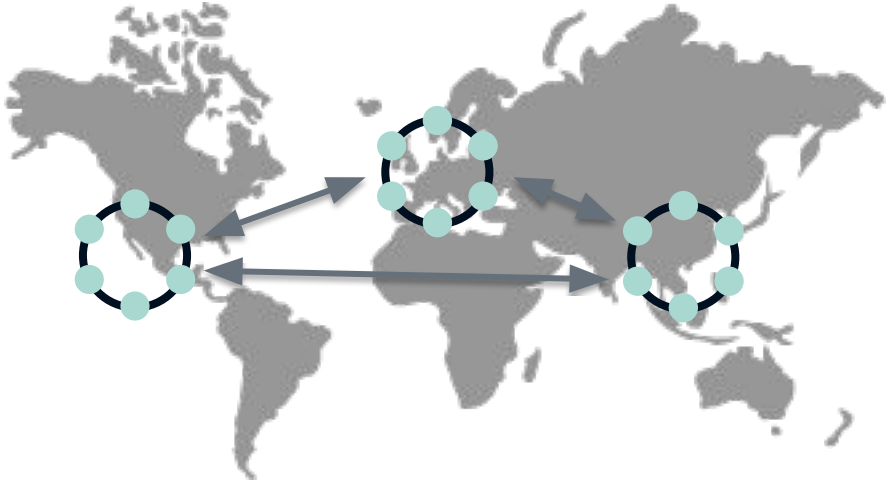
dtsx.io/cassandra-at-netflix



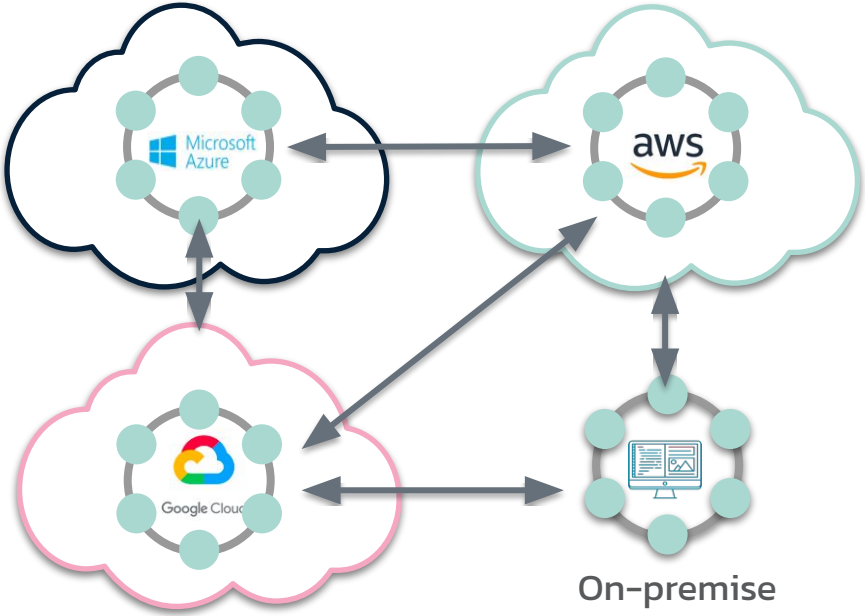
And many others...

Cassandra Biggest Users (and Developers)

Geographical Distribution



Hybrid-Cloud and Multi-Cloud



Data is globally distributed

State-of-the-art NoSQL Database

Astra DB

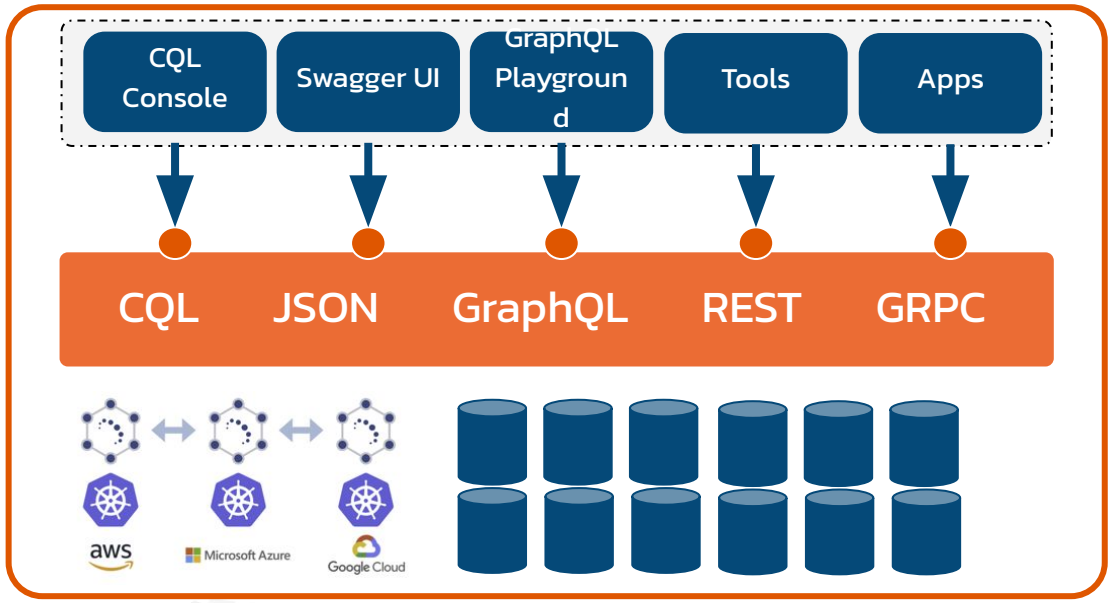
- DBaaS, serverless, auto-scalable
- Multi-cloud, distributed, multi-node cluster
- NoSQL, multi-model
- Tabular, document, key-value
- Based on open-source *Apache cassandra*

DataStax

Astra DB

$25/month credit

Launch a database in the cloud with a few clicks, no credit card required.



User Interface

Web-based developer tools and apps

STARGATE

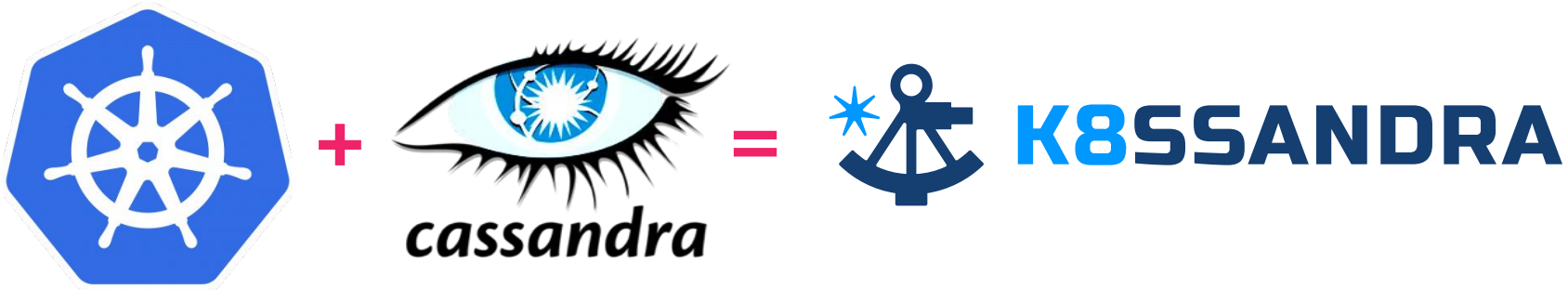
OSS Stargate.io

A data gateway to allow multiple usages

cassandra

OSS Apache Cassandra

A tabular NoSQL database


cassandra
K8SSANDRA

Lab 1

Create AstraDB Instance

https://github.com/datastaxdevs/workshop-[illegible]quarkus-cassandra#1-create-astra-db-instance

01

Intro./Lens

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices = Quarkus
+ Cassandra

06

Live Coding

07

Resources

Agenda

Java Designed for a Different Time

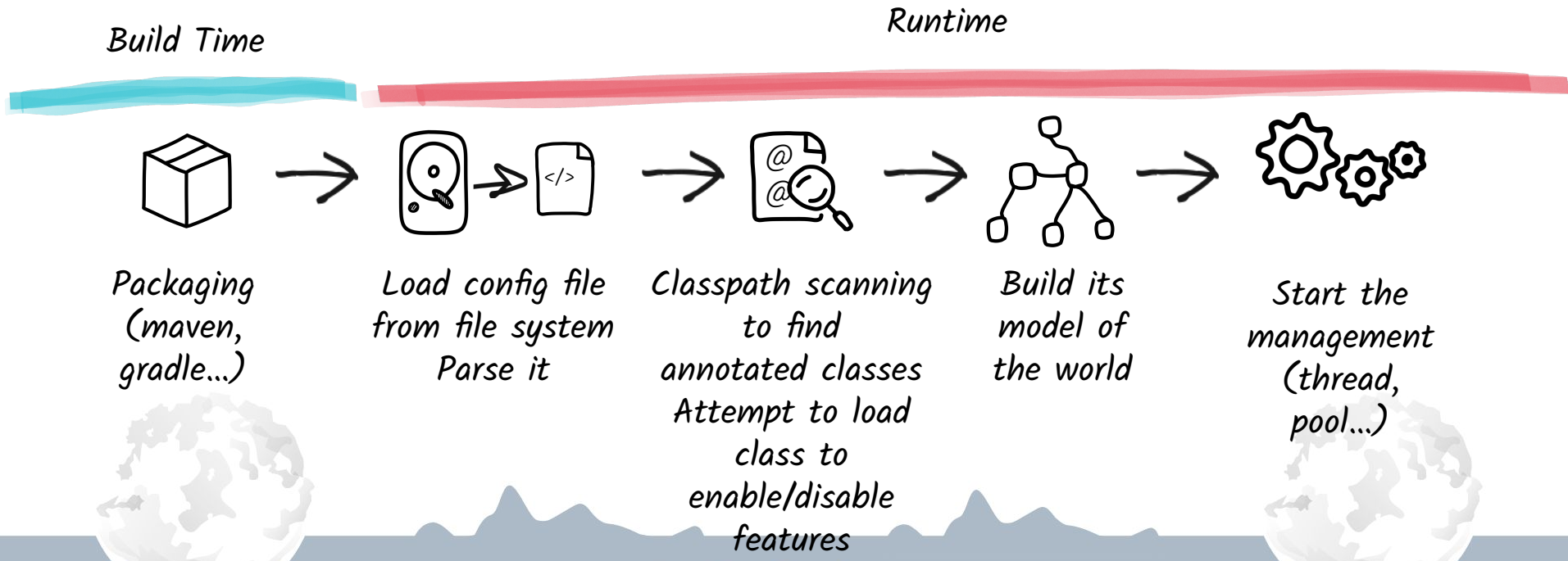
Traditional

- **Throughput** at the expense of **footprint**
- **Long running** at expense of **startup speed**
- Rich, dynamic behavior for mutable systems

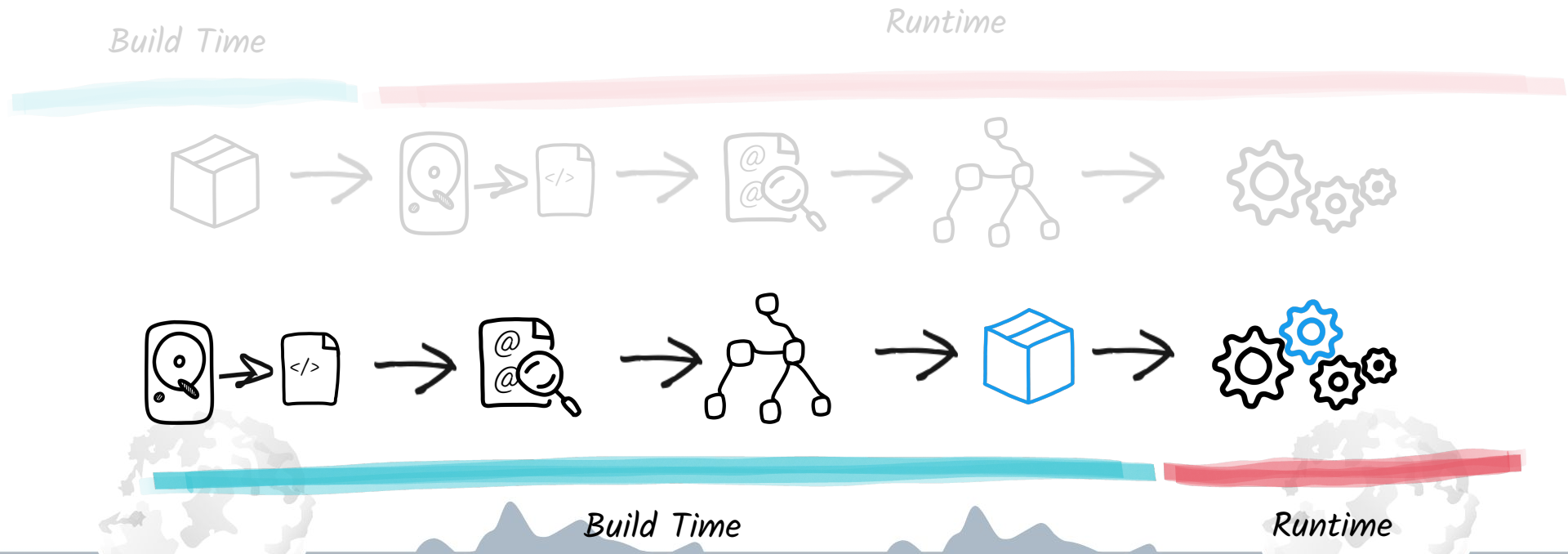
Cloud Native

- **Throughput** solved by scaling
- Ephemeral, immutable systems
- Footprint and performance matter

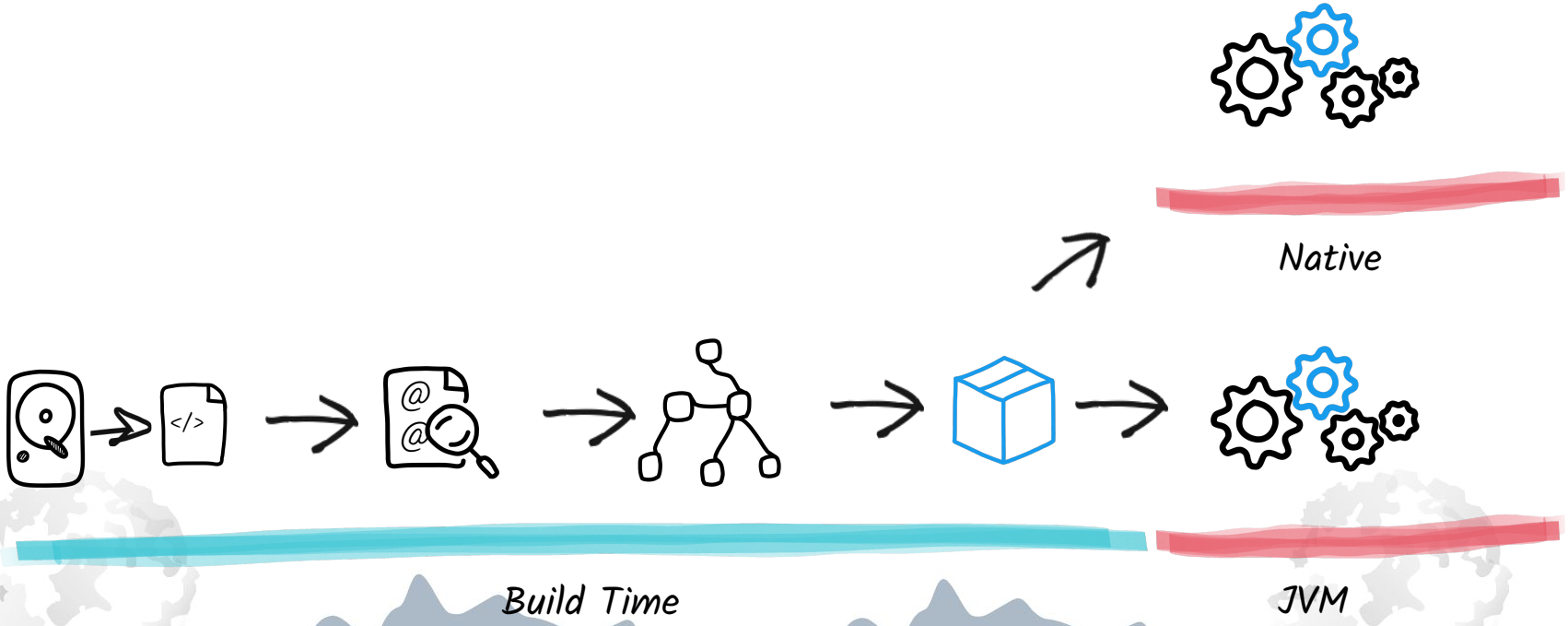
How does a Typical Java Framework Work?



The Quarkus Way



The Quarkus Way Enables Native Compilation



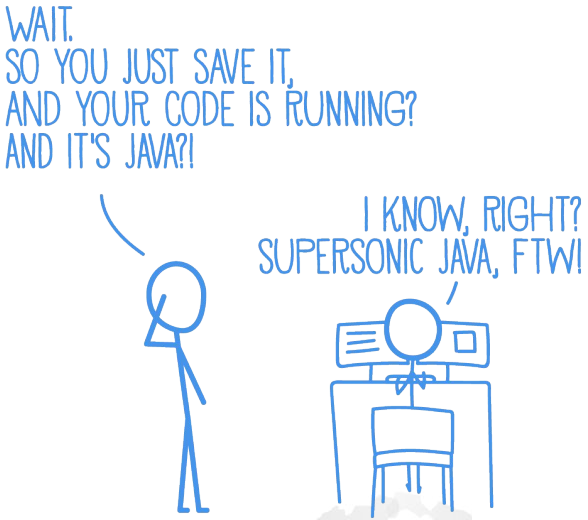
IT'S STILL JAVA!

Inner loop == Developer Productivity

"Our developers used to wait **2 to 3 mins** to see their changes. **Live coding** does away with this!"

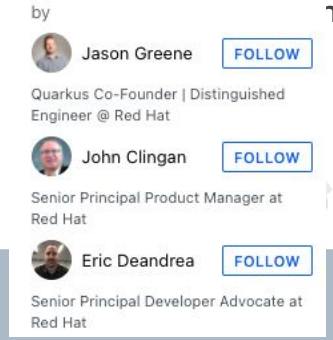
A cohesive platform for Microservices

- Developer productivity
  - Zero-config Live coding
  - Developer services
  - Continuous testing
  - Dev UI and CLI
- Streamlined code for the 80% common usages
- Native executable generation



Kubernetes Native Java with Quarkus

- Quarkus is an industry leader in startup time and memory utilization for native and JVM-based Java applications
- Building a native executable is easy
- Quarkus runs Java applications more efficiently instead of making a costly switch to Node.js or Golang
- Developers can utilize their existing Java ecosystem expertise with build tools and APIs like Jakarta EE, MicroProfile, Spring, and more, with imperative or reactive programming styles – or both!
- Kubernetes is a first-class deployment platform in Quarkus
- Quarkus unites the benefits of rapid development seen in scripting languages, the maturity of the Java ne efficiency of native compilation in a single runtime



https://www.infoq.com/articles/native-java-quarkus

01

Intro./Lens

02

NoSQL

03

Cassandra/Astra
k8ssandra

04

Quarkus

05

Microservices = Quarkus
+ Cassandra

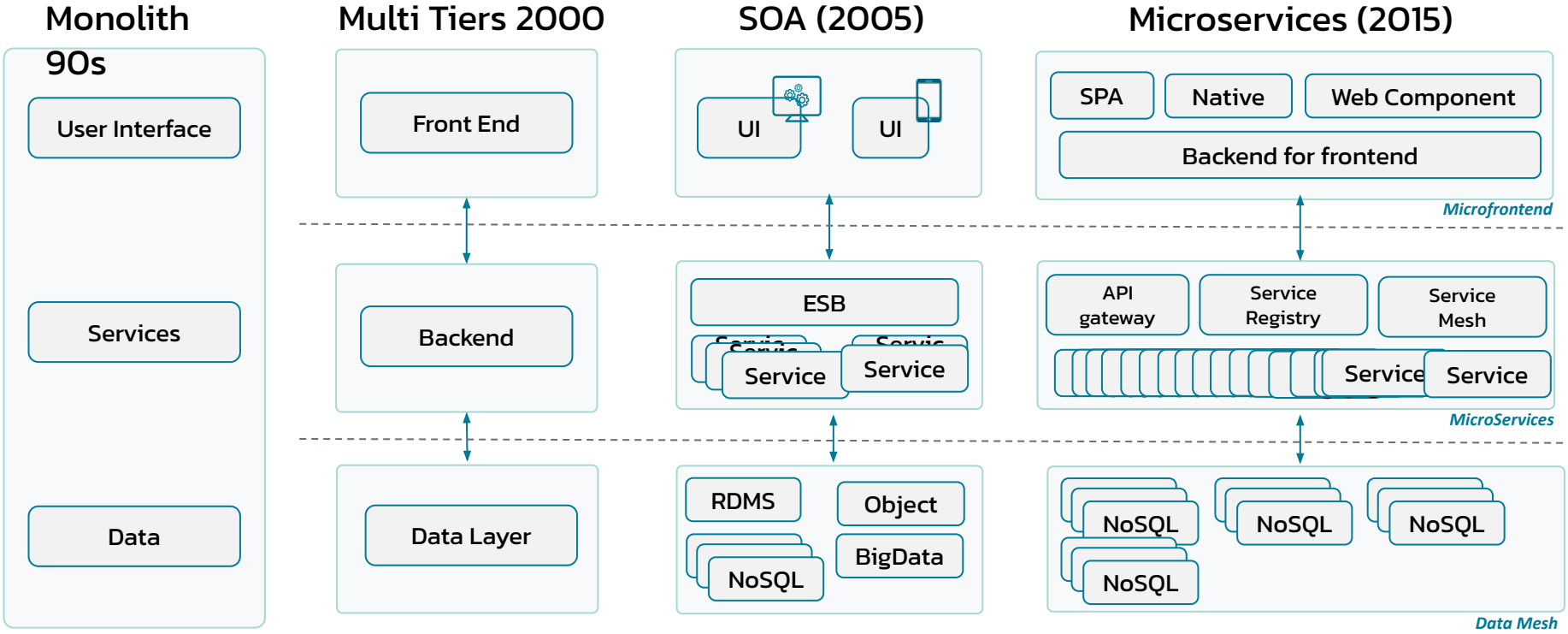
06

Live Coding

07

Resources

Agenda

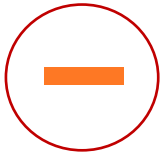


Microservices Architecture evolution

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

Quarkus Cassandra Extension

- Native Quarkus Config
- Cassandra Driver Session Support
- Cassandra Driver Object Mapper Support
- Support for Mutiny Types (Reactive Types)
- Native Image Support
- Support for DataStax Astra (Cassandra DBaaS)

Native Quarkus Config

```
quarkus.cassandra.cloud.secure-connect-bundle=/path/to/astra/bundle.zip
quarkus.cassandra.keyspace=ks1

  quarkus.cassandra.auth.username=alice
  quarkus.cassandra.auth.password=s3cr3t

  quarkus.cassandra.request.timeout=5 seconds
  quarkus.cassandra.request.consistency-level=LOCAL_ONE
  quarkus.cassandra.request.page-size=1000

  quarkus.cassandra.metrics.enabled=true
  quarkus.cassandra.health.enabled=true
```

Cassandra Driver Session Support

```
@Inject QuarkusCqlSession session;


@GET
@Produces(MediaType.APPLICATION_JSON)
@Path(“/product/{id}”)
public Uni<Response> findProduct(@PathParam(“id”) String id) {
  return session.executeReactive("SELECT * FROM table_name WHERE key = " + id)
    .toUni()
    .map(row -> new Product(row.getString("id"), row.getString("name")))
    .map(product -> Response.ok(product).build())
    .ifNoItem().after(Duration.ofSeconds(5))
    .recoverWithItem(Response.status(Status.NOT_FOUND).build());
```

Cassandra Driver Object Mapper Support

```
@Dao interface ProductDao {
  @Insert Uni<Void> create(Product product);
  @Select Uni<Product> findById(String id);
  @Select Multi<Product> findAll();
}

interface ProductMapper {
  @DaoFactory
  ProductDao productDao();
}

@ApplicationScoped class ProductService {
  @Inject ProductDao dao;
```

Support for Mutiny Types

```
@GET
@Produces(MediaType.APPLICATION_JSON)
@Path(“/product/{id}”)
public Uni<Response> findProduct(@PathParam(“id”) String id) {
  return dao.findById(id)
    .map(todo -> Response.ok(todo).build())
    .ifNoItem().after(Duration.ofSeconds(5))
    .recoverWithItem(Response.status(Status.NOT_FOUND).build());
}
```

01

Intro./Lens

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices = Quarkus
+ Cassandra

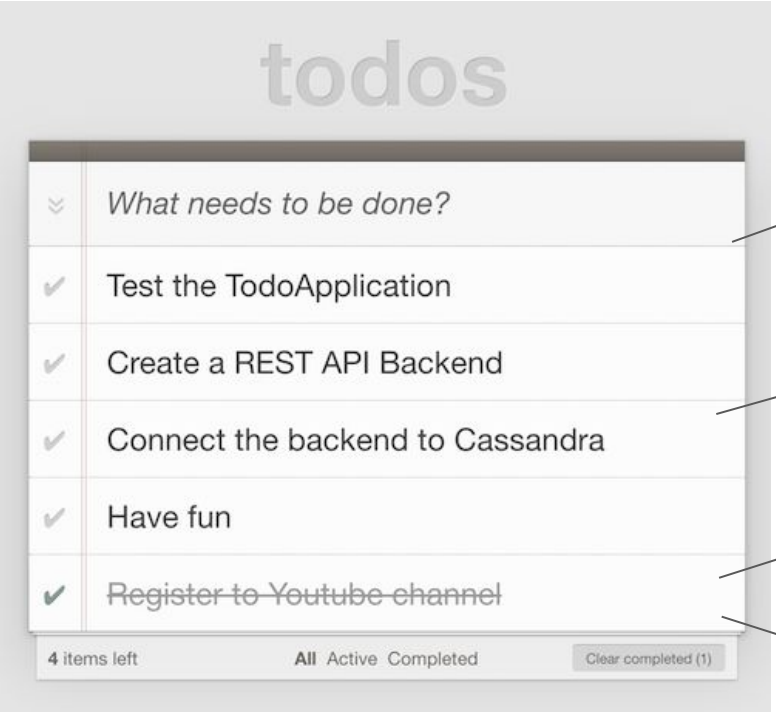
06

Live Coding

07

Resources

Agenda



List all tasks

Create a new Task

Mark a task as completed/uncomplete

Delete a task

Specification of Service layer

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

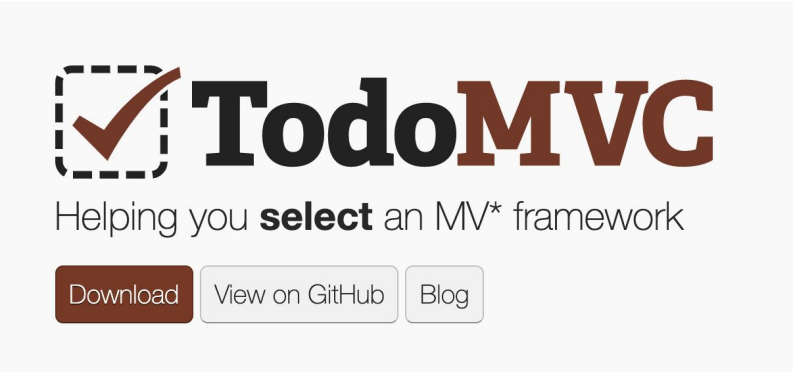
```
CREATE TABLE todos.todoitems (
    user_id    text,
    item_id    timeuuid,
    completed  boolean,
    title      text,
    PRIMARY KEY ((user_id),item_id)
);
```

Data Model

- Our Partition key is user_id
  - We chose to have one todo list per user (avoiding any select * from table)
- Service
  - findTodos() for user
  - createTodo() for user
  - deleteTodo() from its id (userid + itemid)
  - updateTodo() from its id (both to mark it as complete and update title)
- REST API
  - The userid will appear in the URL
  - Provide major version (best practice)

/q/dev

REST API



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

TodoMVC.com

Todo-Backend

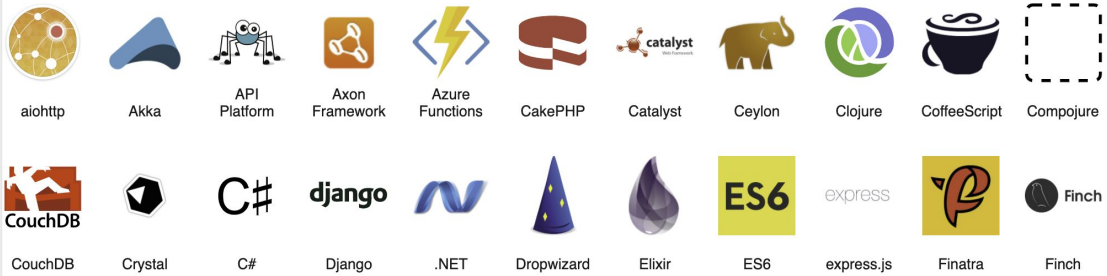
a shared example to showcase backend tech stacks

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.

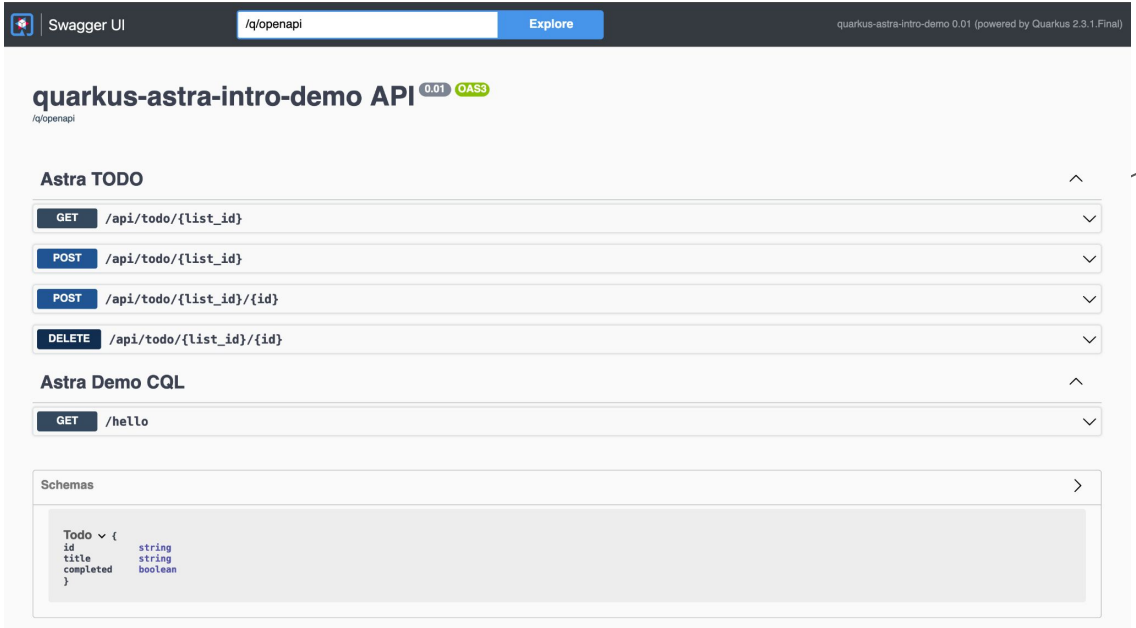
The Todo-Backend project was inspired by the TodoMVC project, and some code (specifically the todo client app) was borrowed directly from TodoMVC.

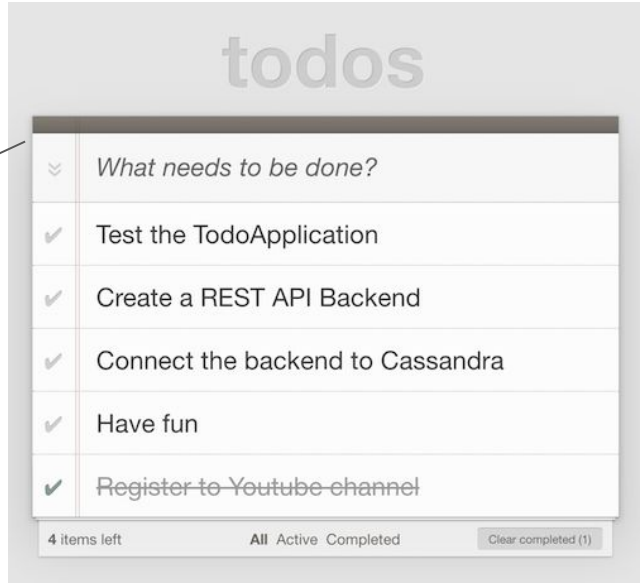
Created and curated by Pete Hodgson.

featuring HTTP APIs built with:

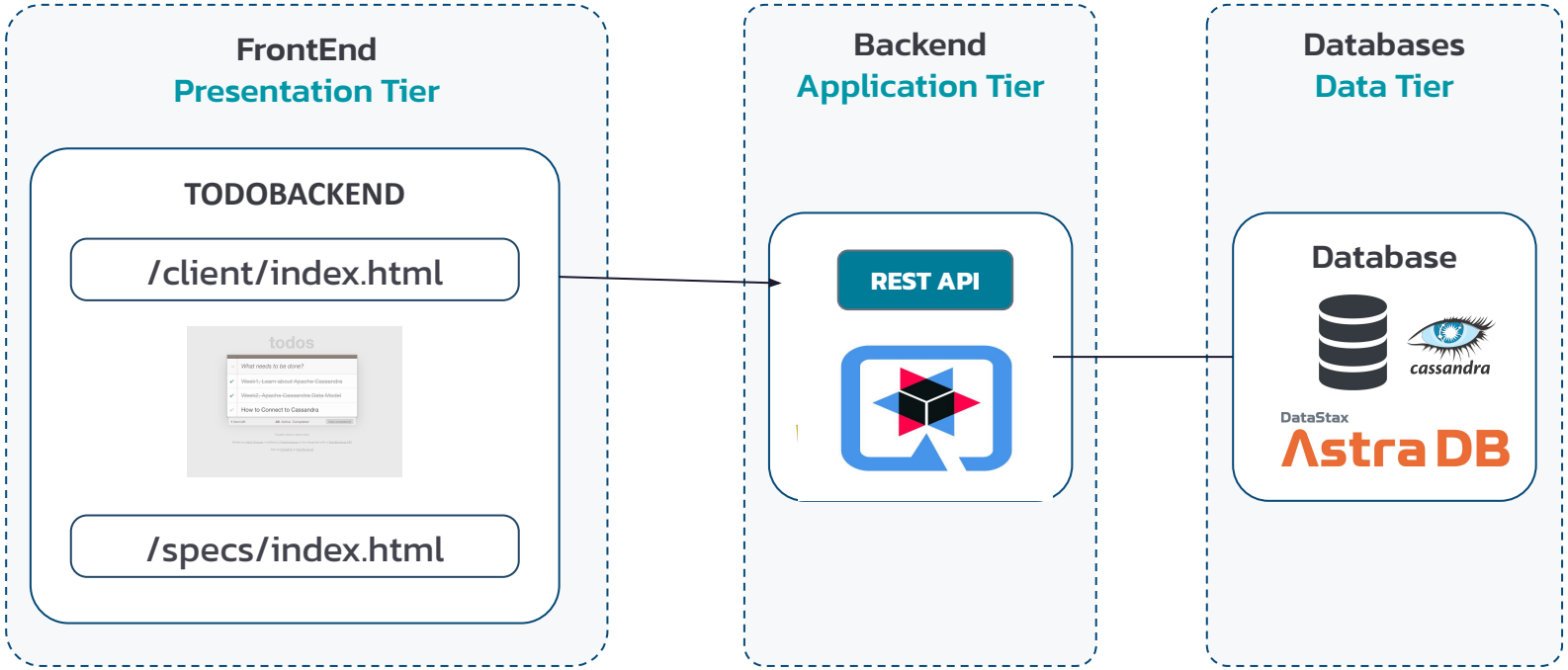


Todo Backend.com





Rest API Specifications (specs)



Architecture

Lab 2,3,4: Create Token ...

https://github.com/datastaxdevs/workshop-intro-quarkus-cassandra#2-create-astra-token

Lab 5,6: Setup App and Test connection

https://github.com/datastaxdevs/workshop-intro-quarkus-cassandra#5-setup-your-application

Lab 10: Containerize, Deploy to Lens

https://github.com/datastaxdevs/workshop-intro-quarkus-cassandra#10-containerizing

Paused for Hands-on
(Don’t miss Menti Quiz at end)

01

Intro

02

NoSQL

03

Cassandra/Astra/
k8ssandra

04

Quarkus

05

Microservices = Quarkus
+ Cassandra

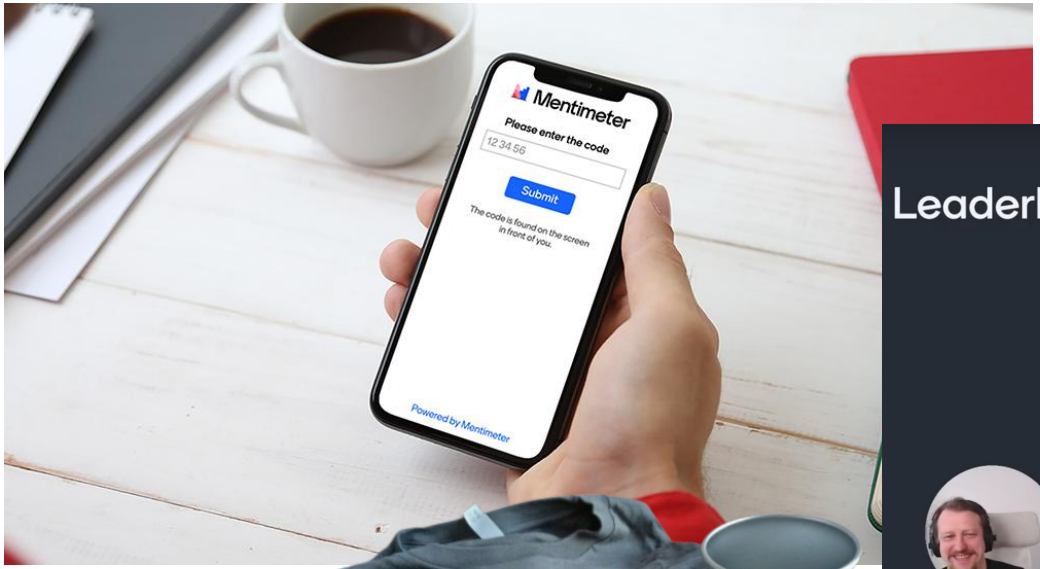
06

Live Coding

07

Resources

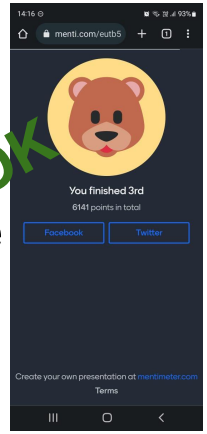
Agenda



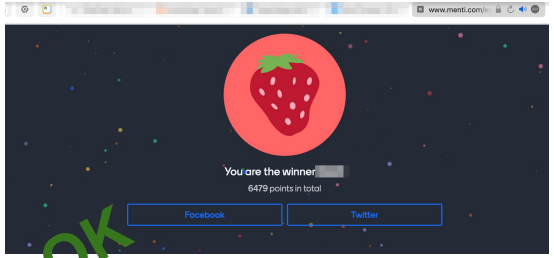
`menti.com` ⇒ enter code
Don't answer in YT chat
Look at phone (not at YT)

Quiz on "Menti" !

SWAG WINNERS 🎉



OK



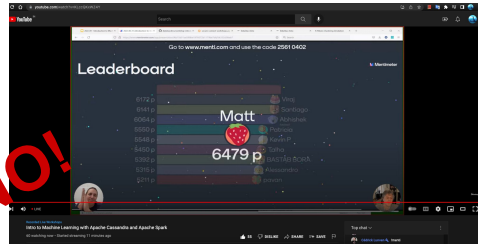
OK

Congratulations to 1st, 2nd and 3rd place

on the Menti quiz!

To claim your prize:

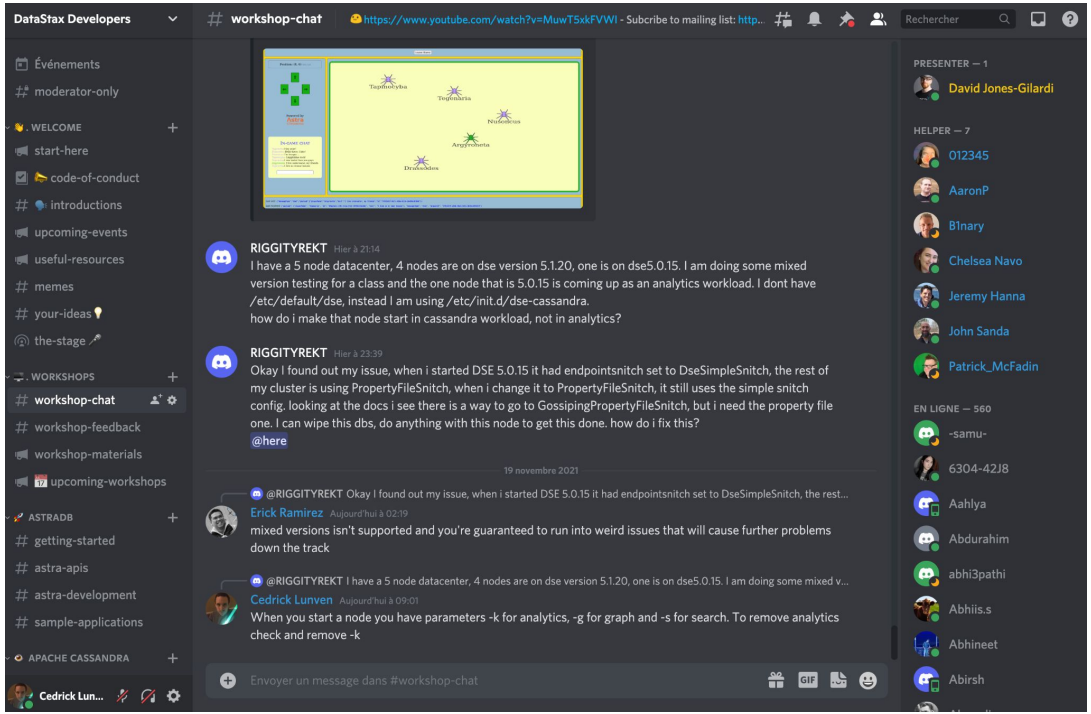
Take a screenshot of your Menti screen



NO!

Fill the form at dtsx.io/workshop-swag

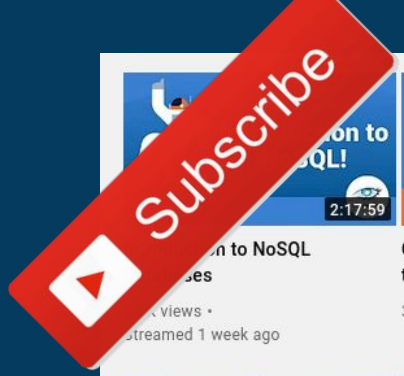
Swag Winners!



!discord

dtsx.io/discord

Datastax Developers Discord (18k+)



...n to NoSQL ...ses

...K views •
...Streamed 1 week ago

Crash Course | Introduction to Cassandra for Developers

331 views • 1 week ago

Introduction to NoSQL Databases

3.5K views •
Streamed 1 week ago

Introduction to NoSQL Databases

10K views •
Streamed 1 week ago

Build your own NETFLIX clone!

4K views •
Streamed 2 weeks ago

Build your own NETFL... clone!

7.4K views •
Streamed 2 weeks ago

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)

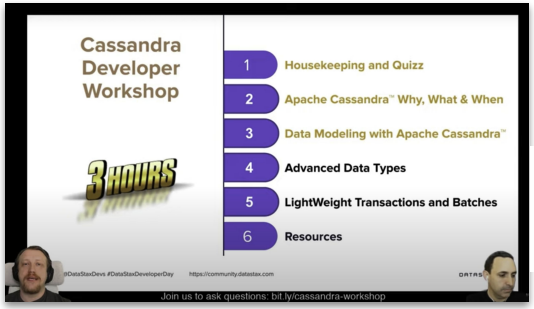
2.4K views •
Streamed 4 weeks ago

Cassandra Day India: RF Room (Talks)

1.3K views •
Streamed 1 month ago

Live and interactive

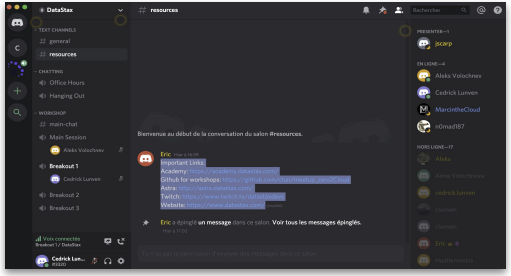
Livestream: youtube.com/DataStaxDevs



YouTube

Twitch

Questions: https://dtsx.io/discord

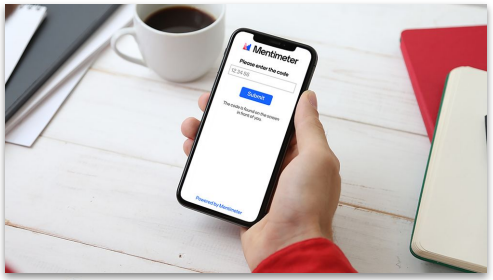


Discord

YouTube

Available on the iPhone App Store

Games menti.com



Mentimeter

GET IT ON Google play

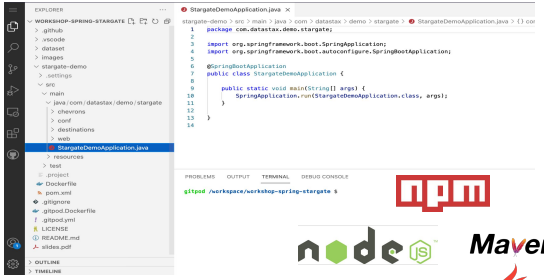
1 Attend the sessions

AstraDB astra.datastax.com

Database + GraphQL + PlayGround

DataStax Astra DB

Gitpod IDE: https://gitpod.io



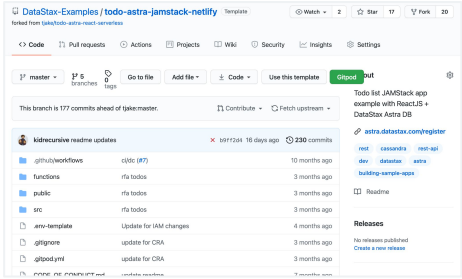
Gitpod

npm

node

Maven

GitHub



2 Complete Workshops Labs

datastax.com/workshops

3 Become a Jedi Master of Astra

- Showcase & explain Quarkus, how it enables modern Java development & the Kubernetes-native experience
- Introduce familiar Spring concepts, constructs, & conventions and how they map to Quarkus
- Equivalent code examples between Quarkus and Spring as well as emphasis on testing patterns & practices
- Chapters devoted to
  - Why the need for Quarkus in the first place?
  - Getting started
  - RESTful applications
  - Persistence
  - Event-driven services
  - Cloud environments, containers, and Kubernetes

https://red.ht/quarkus-spring-devs

- Create a Quarkus Serverless Project
  - Run Serverless Functions Locally
  - Test Serverless Functions Continuously
  - Deploy Functions to a Knative Service on OpenShift
- Make Serverless Functions Run Faster With GraalVM
- Make Portable Functions Across Serverless Platforms
  - Deploy a Quarkus Funqy Application to AWS Lambda
- Bind CloudEvents on Knative With Quarkus Serverless Functions

https://dzone.com/refcardz/getting-started-with-quarkus-serverless-functions

Try Cassandra+Quarkus

- Create your quarkus + cassandra app (code.quarkus.io or running the following):

```
$ mvn io.quarkus:quarkus-maven-plugin:2.9.1.Final:create \
  -DprojectGroupId=io.quarkus.astra \
  -DprojectArtifactId=quarkus-astra-demo \
  -DprojectVersion=1.0.0 \
  -DclassName="io.quarkus.astra" \
  -Dextensions="resteasy-reactive-jackson, micrometer-registry-prometheus, smallrye-openapi, smallrye-health,
cassandra-quarkus-client"
```

- Stand up your Astra free database (astra.datastax.com)
- Check out https://k8ssandra.io
- Get coding + see docs for more info and try the Quarkus + Cassandra workshop

https://quarkus.io/guides/cassandra

https://github.com/datastaxdevs/workshop-intro-quarkus-cassandra/

Thank You!