Week 3: App. Development

DATASTAX + Apache Cassandra

Creating CRUD backend Services

Cassandra Cloud-Native Workshop Series

Building Cloud-Native apps with Cassandra Expertise

The Crew

MATERIALS → bit.ly/CassandraWorkshopMaterials

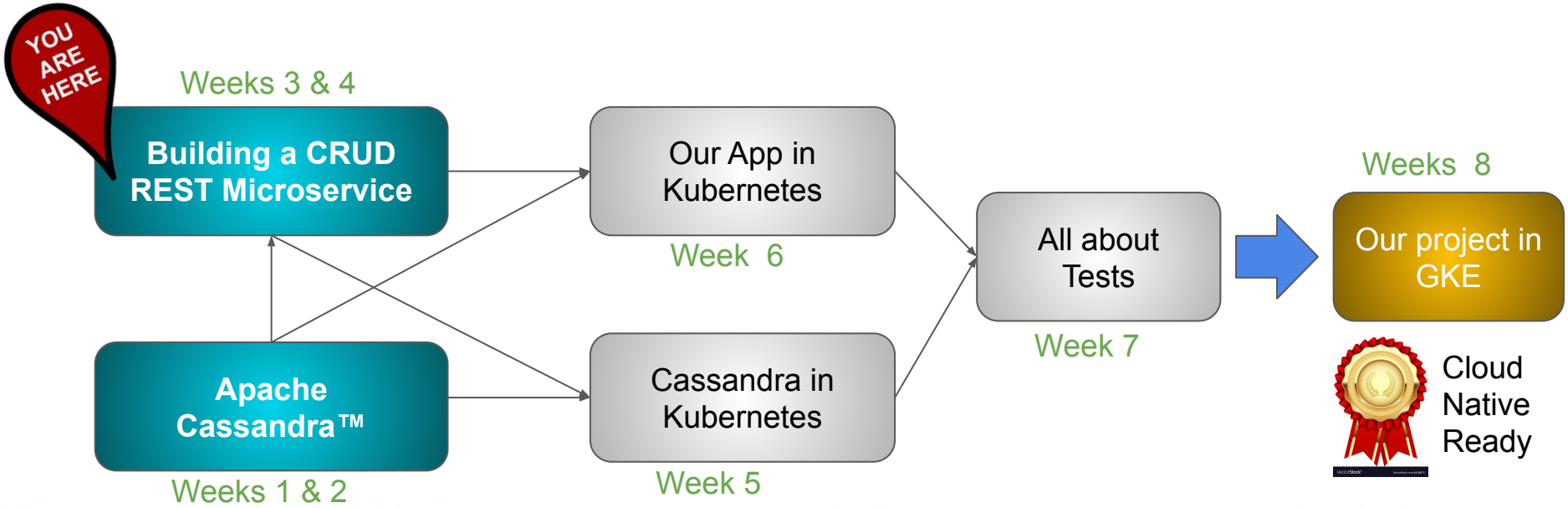
Thank you !

- Registrations as of now : 11k

- Week 1
  - Numbers of views : ~ 18 k
  - Numbers of people done with exercises : ~ 1900

- Week 2
  - Numbers of views : 9.1 k
  - Numbers of people done with exercises : ~ 900

Workshops Series = Not only Cassandra



Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)

Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)

HouseKeeping

- **Pre-requisites, we expect you:**
  - To have already created an Astra instance (week 1)
  - To have knowledge with Cassandra Data Modelling (week 2)
  - To know basics of **one** of the following languages:

- **You don't have to install anything**

Livestreams

YouTube

Twitch

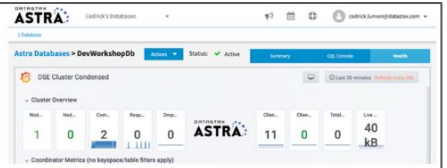
Live Questions

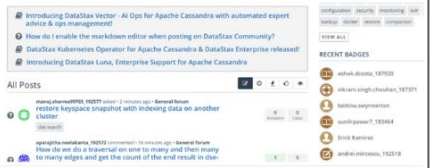
YouTube

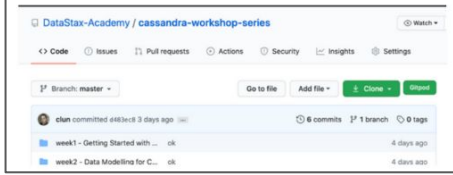
Discord

Materials & Help

DATASTAX COMMUNITY







Astra.datastax.com

cassandra

GitHub

Gitpod

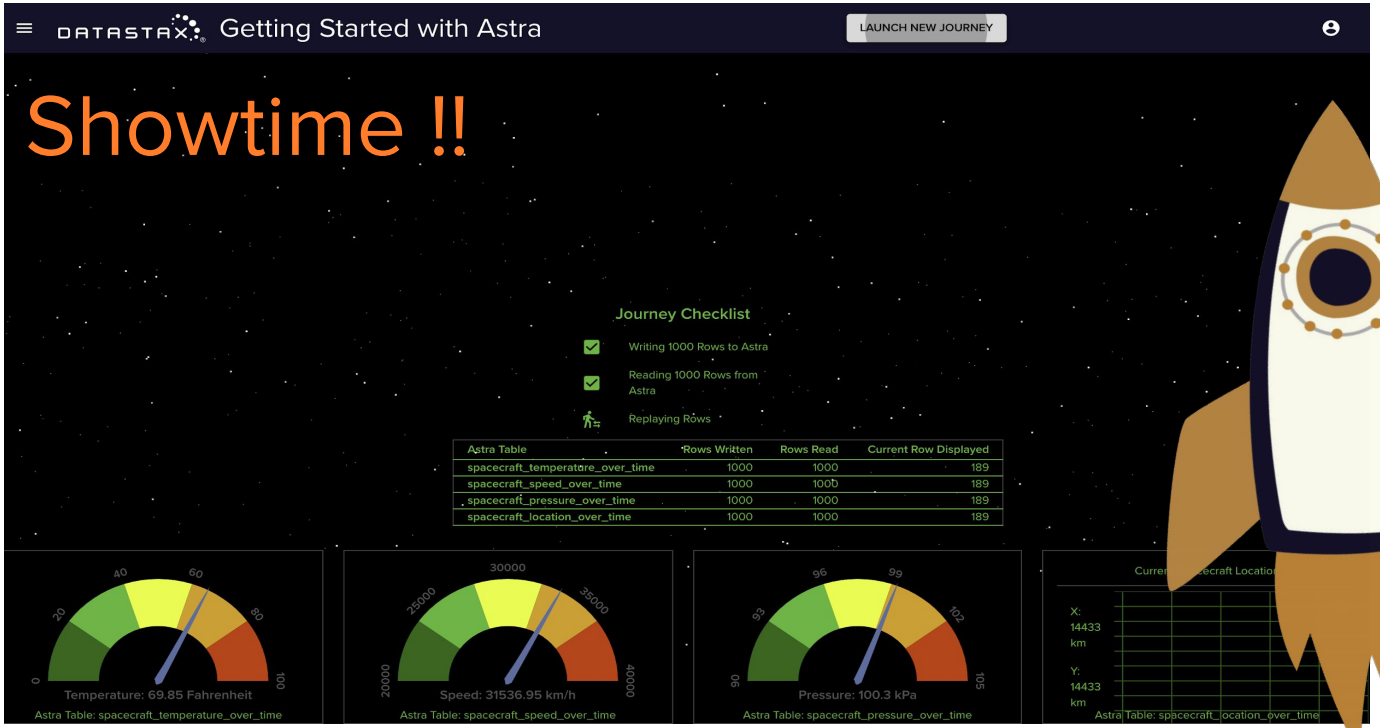
MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra


menti.com
92 05 88
Mentimeter
What is the most desirable feature of your ideal workplace?
Flexible time
Social factor
Big paycheck
Available on the iPhone
App Store
GET IT ON
Google play

Application Development **CRUD**

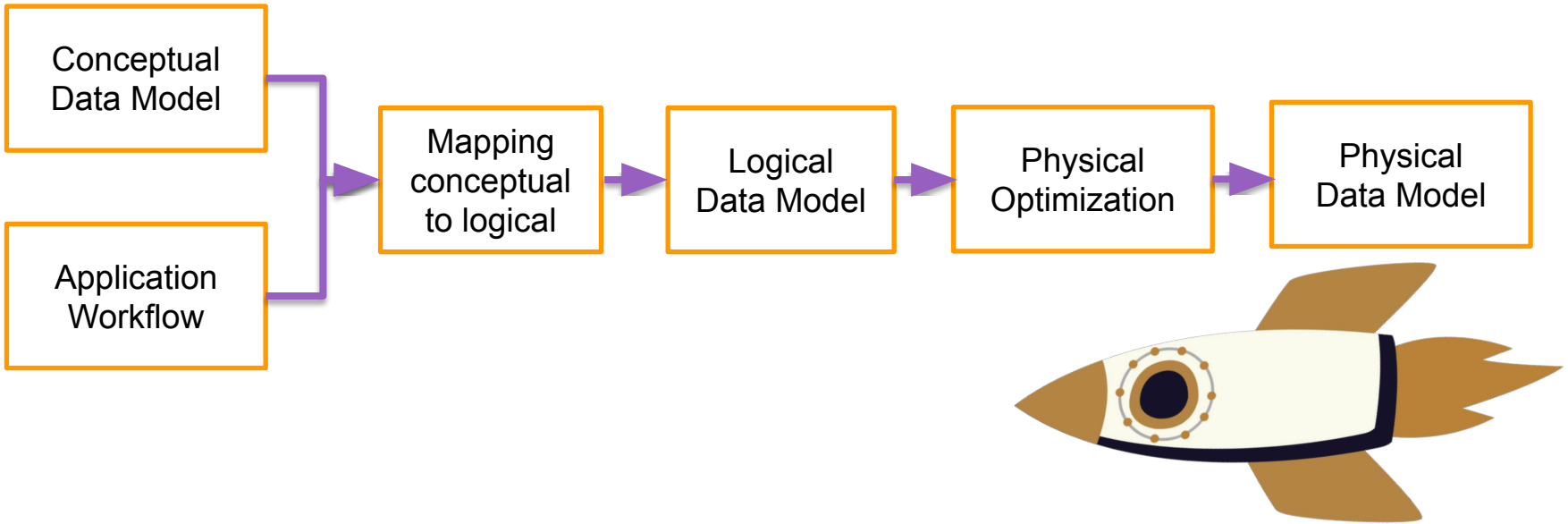
1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)



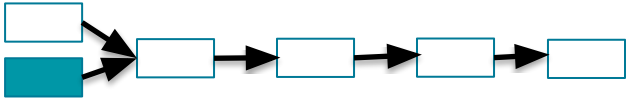
MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Designing your data model

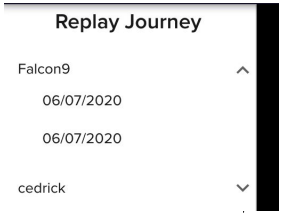


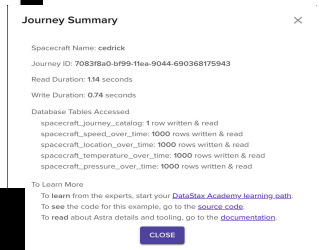
#1 Application Workflow

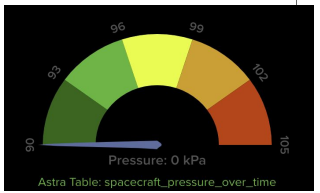


- **Space crafts catalog queries**
  - Look up all of the journeys for a spacecraft
  - Look up the state of a journey
  - Create a new journey

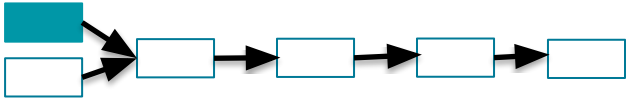
- **Sensor readings queries :** Speed, Pressure, Temperature, Location
  - Save readings over time
  - Analyze each dimension independently
  - Analyze data per journey
  - Less than 100.000 records per journey per dimension

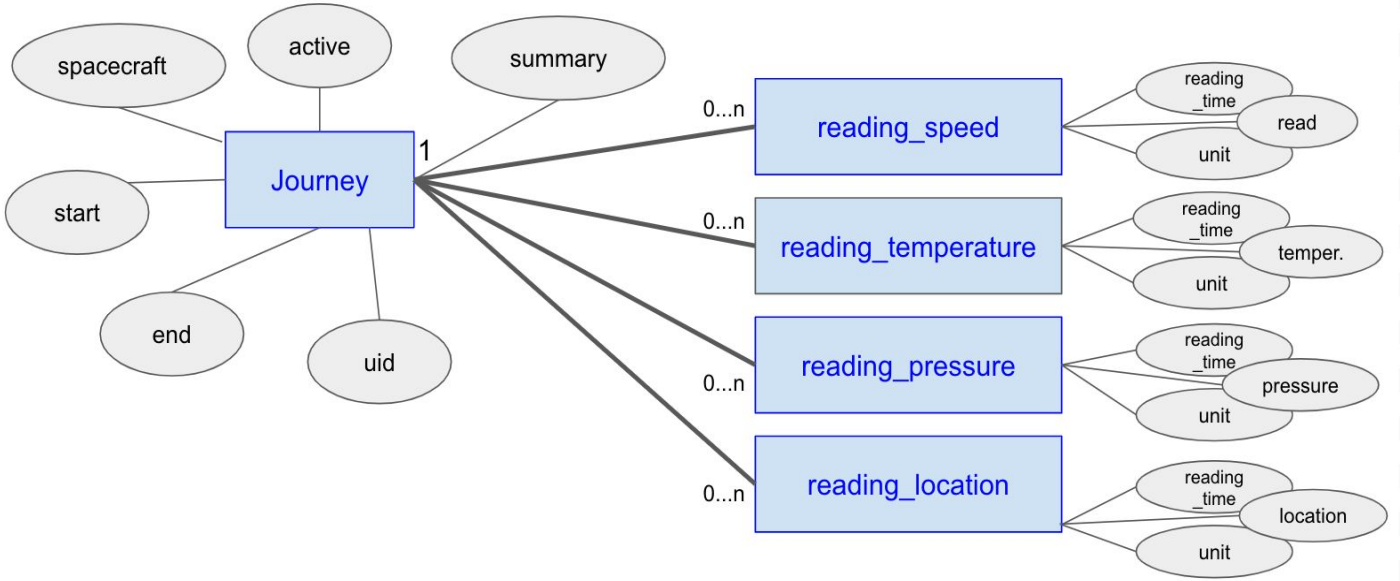




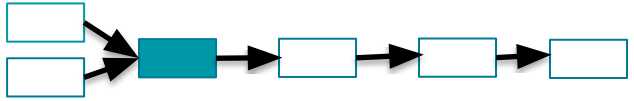


#1 Conceptual Data Model





#2 Map to Logical Data Model



spacecraft_temperature_over_time

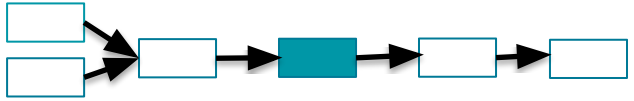
spacecraft_location_over_time

spacecraft_journey_catalog

spacecraft_speed_over_time

spacecraft_pressure_over_time

#3 Logical Data Model



spacecraft_journey_catalog

spacecraft_name	K
journey_id	C ↓
start	
end	
active	
summary	

spacecraft_temperature_over_time

spacecraft_speed_over_time

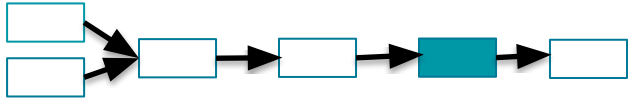
spacecraft_name	K
journey_id	K
reading_time	C ↓
speed	
speed_unit	

spacecraft_location_over_time

spacecraft_pressure_over_time

spacecraft_name	K
journey_id	K
reading_time	C ↓
pressure	
pressure_unit	

#4 Physical Data Model



spacecraft_journey_catalog

spacecraft_name	text
journey_id	timeuuid
start	timestamp
end	timestamp
active	boolean
summary	text

spacecraft_temperature_over_time

spacecraft_speed_over_time

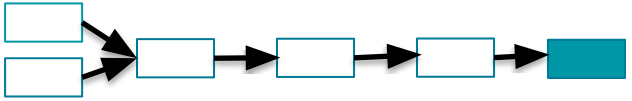
spacecraft_name	text
journey_id	timeuuid
reading_time	timestamp
speed	double
speed_unit	text

spacecraft_location_over_time

spacecraft_pressure_over_time

spacecraft_name	text
journey_id	timeuuid
reading_time	timestamp
pressure	double
pressure_unit	text

#5 CQL DDL



```
CREATE TABLE IF NOT EXISTS spacecraft_journey_catalog
(
  spacecraft_name  text,
  journey_id       timeuuid,
  start            timestamp,
  end              timestamp,
  active           boolean,
  summary          text,
  PRIMARY KEY ((spacecraft_name), journey_id))
  WITH CLUSTERING ORDER BY (journey_id desc);
```

```
CREATE TABLE IF NOT EXISTS
spacecraft_speed_over_time  (
  spacecraft_name  text,
  journey_id       timeuuid,
  speed            double,
  reading_time     timestamp,
  speed_unit       text,
  PRIMARY KEY ((spacecraft_name,
                journey_id), reading_time))
  WITH CLUSTERING ORDER BY (reading_time DESC);
```

Exercise

NOTEBOOK:
“Spacecraft.tar”

DataStax Studio | CCNDW - Week3 | cedrick.lunven@datasta... | Schema

Language: **Markdown**

Welcome to `Cassandra Cloud Native Workshop Series` WEEK 3

This week we start coding.
This notebook will help you create the required tables and data but also make you check that the databases has been updated after coding

Language: **Markdown**

Space Flight Catalog Data Model

In this example, our business requires that we track each spacecraft journey and store the vital signs.
We'll use this data to optimize the onboard conditions and predict when it might be time for a new model.
We know that our data is going to come in at high rates due to the IoT nature of the use case and also grow over time because of the time-series structure of the information.

We'll design five tables to handle the basics:

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)

DataStax Drivers Features

Connectivity

- Token & Datacenter Aware
- Load Balancing Policies
- Retry Policies
- Reconnection Policies
- Connection Pooling
- Health Checks
- Authentication | Authorization
- SSL

Query

- CQL Support
- Schema Management
- Sync/Async/Reactive API
- Query Builder
- Compression
- Paging

Parsing Results

- Lazy Load
- Object Mapper
- Spring Support
- Paging

Install Drivers

Maven

```
<dependency>
   <groupId>com.datastax.oss</groupId>
   <artifactId>java-driver-core</artifactId>
   <version>4.7.2</version>
</dependency>
```

Java

4.5.2 npm

```
npm install cassandra-driver
```

node.js

3.24.0 python Package Index

```
pip install cassandra-driver
```

python

nuget v3.15.0

```
Install-Package CassandraCSharpDriver -Version 3.15.0
```

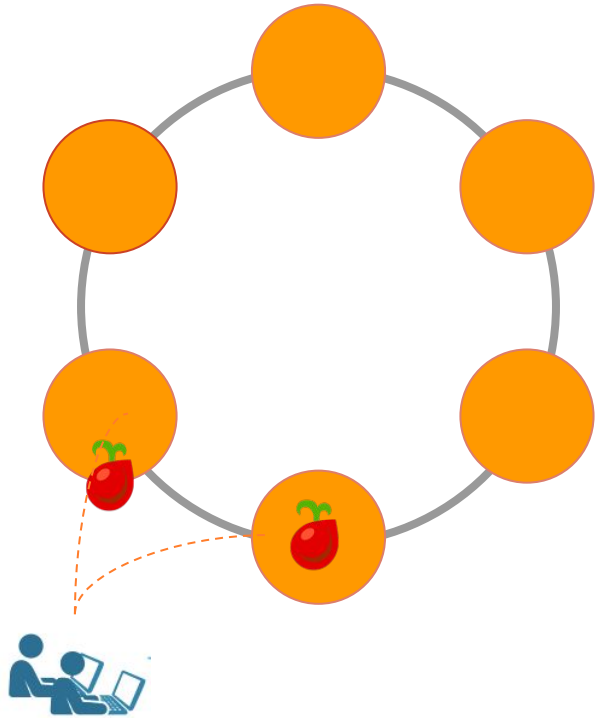
C#

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Contact Points

- Only one necessary
- Unless that node is down
- More are good



Connectivity to Cassandra

```java
CqlSession cqlSession = CqlSession.builder()
 .addContactPoint(new InetSocketAddress("127.0.0.1", 9042))
 .withKeyspace("killrvideo")
 .withLocalDatacenter("dc1")
 .withAuthCredentials("U","P")
 .build();
```


```node
const client = new cassandra.Client({
  contactPoints: ['127.0.0.1'],
  localDataCenter: 'dc1',
  keyspace: 'killrvideo',
  credentials: { username: 'U', password: 'P' }
});
```


```python
uth_provider = PlainTextAuthProvider(
   username='U', password='P')
cluster = Cluster(['127.0.0.1'],
  auth_provider=auth_provider, protocol_version=2)
session = cluster.connect('killrvideo')
```

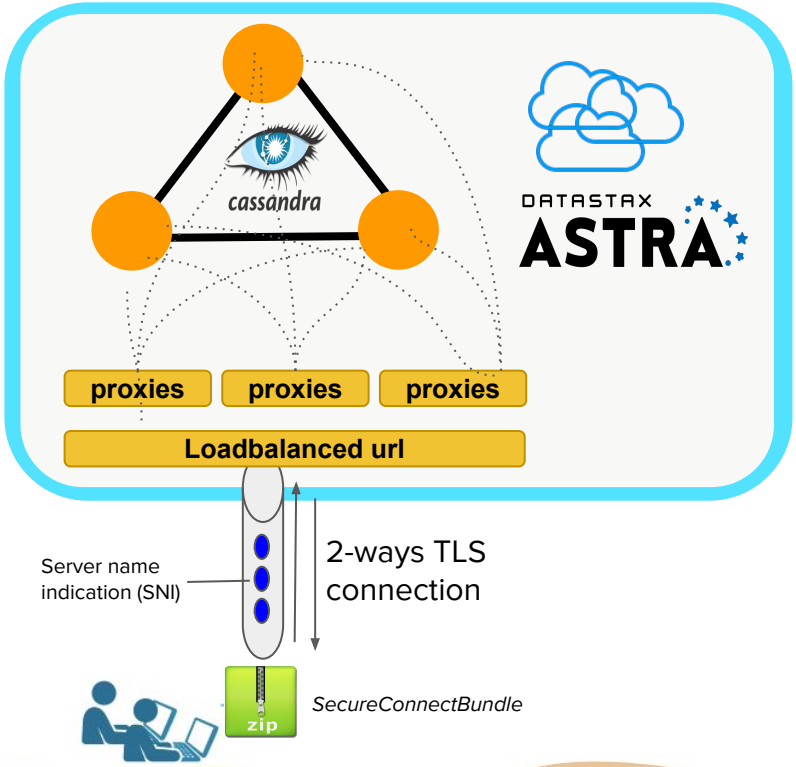

```c#
Cluster cluster = Cluster.Builder()
.AddContactPoint("127.0.0.1")
.WithCredentials("U", "P")
.Build();
session = cluster.Connect("killrvideo");
```

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Contact Points...with ASTRA

- ***SecureConnectBundle*** **contains certificate allowing strong auth 2 ways TLS**
- **Same behaviour (retry, healthcheck, load-balancing) using SNI.**
- **<u>No</u> Single point of failure (*spof*)**



Connection to Cassandra ...with ASTRA

```java
CqlSession cqlSession = CqlSession.builder()
 .withCloudSecureConnectBundle(Paths.get("secure.zip"))
 .withAuthCredentials("U","P")
 .withKeyspace("killrvideo")
 .build();
```

```node
const client = new cassandra.Client({
  cloud: { secureConnectBundle: 'secure.zip' },
  credentials: { username: 'u', password: 'p' }
});
```

```python
auth_provider = PlainTextAuthProvider(
   username='U', password='P')
cluster = Cluster(
   Cloud ={ Secure_connect_bundle: 'secure.zip'},
   auth_provider=auth_provider, protocol_version=2)
session= cluster.connect('killrvideo')
```

```dotnet
var cluster = Cluster.Builder()
 .WithCloudSecureConnectionBundle("secure.zip")
 .WithCredentials("u", "p")
 .Build();
var session = cluster.Connect("killrvideo");
```

MATERIALS → bit.ly/CassandraWorkshopMaterials

Important about a Session/Client

- **S**tateful object handling communications with each node
- **S**hould be **unique** in the Application (*Singleton*)
- **S**hould be **closed** at application shutdown (*shutdown hook*) in order to free opened TCP sockets (*stateful*)

```
Java:       cqlSession.close();
Python:     session.shutdown();
Node:       client.shutdown();
CSharp:     IDisposable
```

Exercise

Java python node

Connect to Cassandra”

✨ Application Development CRUD and Backend Servic

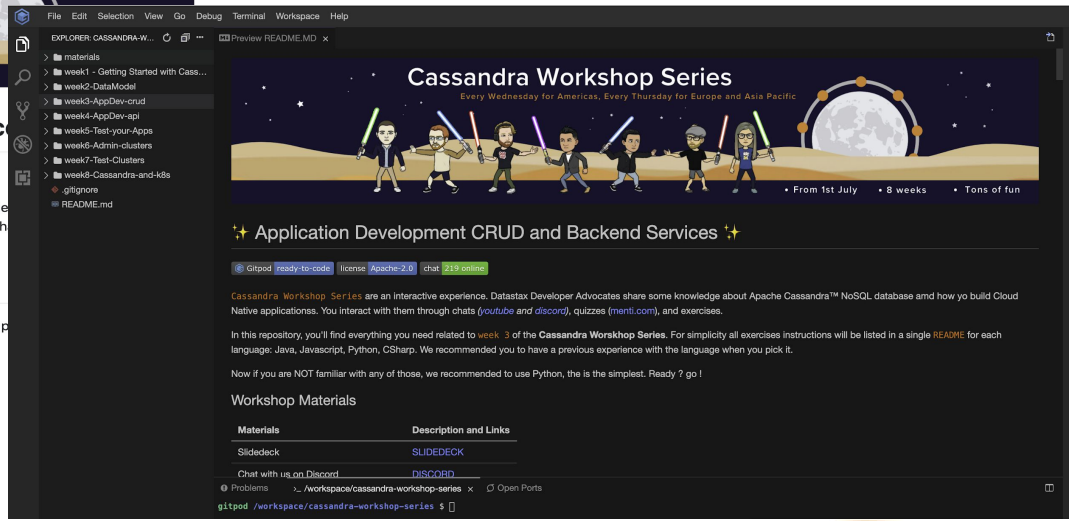
Gitpod ready-to-code | license Apache-2.0 | chat 322 online

`Cassandra Workshop Series` are an interactive experience. Datastax Developer Advocates share some knowle
Cassandra™ NoSQL database amd how yo build Cloud Native applicationss. You interact with them through ch
quizzes (menti.com), and exercises.

👩‍🏫 MATERIALS OF THE WEEK

In this repository, you'll find everything you need related to `week 3` of the **Cassandra Worskhop Series**. As a p
have you Astra instance ready and running. You can catchup by creating your Astra instance

Materials	Description and Links
👩‍🏫Slidedeck	Slidedeck for the workshop
📹 Recording LIVESTREAM NA, LATAM	📅 July 15th 12EDT
📹 Recording LIVESTREAM APAC, EMEA	📅 July 16th 12:30 IST
📗 Homeworks	here



MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)

Execute statements

```java
Statement statement = SimpleStatement
  .builder("select * from t1 where c1 = ?")
  .addPositionalValue(5)
session.execute(stmt);
```


```nodejs
client.execute('select * from t1 where c1 = ?', [5]);
```


```python
session.execute("select * from t1 where c1 = %s", 5);
```


```dotnet
var statement = new SimpleStatement("select * from t1
  where c1 = ?"", 5);
session.Execute(statement);
```

Prepared and Bound Statements

```java
PreparedStatement prepared = session.prepare(
    "select * from t1 where c1 = ?");
BoundStatement bound = prepared.bind("5");
session.execute(bound);
```


```node
const query = 'select * from t1 where c1 =?;';
await client.execute(query, [5], { prepare: true })
```


```python
prepared = session.prepare(
    "select * from t1 where c1=?")
result = session.execute(prepared, [5])
```


```c#
var ps = session.Prepare(
      "select * from t1 where c1 = ?");
var statement = ps.Bind("5");
session.Execute(statement);
```

CRUD Repository Pattern

```
public interface IRepository<ID, T> {
   T findById(ID id);
    void save(T entity);
    void delete(ID id);
    void update(T entity);
   Iterable<T> list()/findAll();
}
```

Exercise

CREATE, READ,UPDATE

a **Create a Journey (create)**

b **Take Off (update)**

c **Telemetry (save, batch)**

d **Landing (update)**

✨ Application Development CRUD and Backend Services ✨

Gitpod ready-to-code | license Apache-2.0 | chat 322 online

`Cassandra Workshop Series` are an interactive experience. Datastax Developer Advocates share some knowledge about Apache Cassandra™ NoSQL database amd how yo build Cloud Native applicationss. You interact with them through chats (*youtube* and *discord*), quizzes (menti.com), and exercises.

MATERIALS OF THE WEEK

In this repository, you'll find everything you need related to `week 3` of the **Cassandra Worskhop Series**. As a pre-requisite you need to have you Astra instance ready and running. You can catchup by creating your Astra instance

Materials	Description and Links
Slidedeck	Slidedeck for the workshop
Recording LIVESTREAM NA, LATAM	July 15th 12EDT
Recording LIVESTREAM APAC, EMEA	July 16th 12:30 IST
Homeworks	here

MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra


menti.com
92 05 88
LIVE
QUIZ
Mentimeter
What is the most desirable feature of your ideal workplace?
Flexible time
Social factor
Big paycheck
menti.com
Available on the iPhone
App Store
GET IT ON
Google play

Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)

MATERIALS → bit.ly/CassandraWorkshopMaterials

ResultSet and Rows

- **ResultSet** is the object returned for executing query. It contains **ROWS** (data) and **EXECUTION INFO**.
- **ResultSet** is **iterable** and as such you can navigate from row to row.
- Results are **always paged** for you (avoiding memory and response time issues)

ResultSet

```java
ResultSet resultSet = session.execute("select * from
  t1 where c1 = 5");

Row row = resultSet.one();
```


```javascript
const result = await client.execute("select * from
  t1 where c1 = 5");

const row = result.first();
```


```python
#ResultSet
result = session.execute("select * from t1 where c1 = 5")
#Row
result.one()
```


```c#
var resultSet = session.Execute("select * from t1 where
  c1 = 5");

var row = resultSet.First();
```

MATERIALS → bit.ly/CassandraWorkshopMaterials

Parsing Rows

```java
Row row = resultSet.one();
Boolean isUserNameNull = row.isNull("userName");
var value = row.getString("userName");
```


```nodejs
row = result.first();
//false when null
Boolean(row.userName);
value = row.userName.toString();
```


```python
row = resultSet.one();
#false if None
bool(row.userName)
value = str(row.userName)
```


```dotnet
var row = resultSet.First();
Boolean isUserNameNull = row.IsNull("userName");
var value = row.GetValue<string>("userName");
```

Object Mapper

```java
@Dao
public interface SpacecraftJourneyDao {
   PagingIterable<SpacecraftJourneyCatalog> findAll();
}
 @Mapper
 public interface SpacecraftMapper {
 @DaoFactory SpacecraftJDao dao(@DaoKeyspace CqlIdentifier keyspace);
}
```


```nodejs
const mapper = new cassandra.mapping.Mapper(client, {
models: {
 'Journey': { tables: [‘spacecraft_journey_catalog’],
    keyspace: 'killrvideo', columns: { 'journey_id': 'journeyId'},
    mappings: new UnderscoreCqlToCamelCaseMappings()
} } });
const journeyMapper = mapper.forModel('Journey');
journeyMapper.insert({ videoId });
```


```python
from cassandra.cqlengine import columns
from cassandra.cqlengine.models import Model
from cassandra.cqlengine.management import sync_table
class Journey(Model):
    journey_id = columns.UUID(primary_key=True)
    spacecraft_name = columns.Text()
sync_table(Journey)
j = Journey.create(journey_id=...)
```


```csharp
MappingConfiguration.Global.Define(new Map<Journey>()
  .TableName("spacecraft_journey_catalog")
  .PartitionKey(k => j.journeyId, )
  .Column(u => u.journeyId, cm => cm.WithName("journey_id")
  .Column(u => u.spacecraftName, cm => cm.WithName("spacecraft_name")));

IEnumerable<Journey> journey = mapper.Fetch<Journey>("WHERE spacecraft_name
   = ? AND journey_id =?", a, b);
```

MATERIALS → bit.ly/CassandraWorkshopMaterials

Object Mapper

```
@Dao
public interface SpacecraftJourneyDao {
   PagingIterable<SpacecraftJourneyCatalog> findAll();
}
 @Mapper
 public interface SpacecraftMapper {
 @DaoFactory
 SpacecraftJDao dao(@DaoKeyspace CqlIdentifier keyspace);
}
SpacecraftMapper mapper = new SpacecraftMapperBuilder(cqlSession).build();
SpacecraftJourneyDao dao = mapper.dao(cqlSession.getKeyspace().get());
```

```
@Entity
@CqlName(SpacecraftJourneyCatalog.TABLE_NAME)
public class SpacecraftJourneyCatalog implements Serializable {
   @PartitionKey
   @CqlName(COLUMN_SPACECRAFT_NAME)
   private String name;

   @ClusteringColumn
   @CqlName(COLUMN_ID)
   private UUID journeyId;

   @CqlName(COLUMN_START)
   private Instant start;

   @CqlName(COLUMN_SUMMARY)
   private String summary;
```

Exercise

Read & Parse Results

✨ Application Development CRUD and Backend Servic

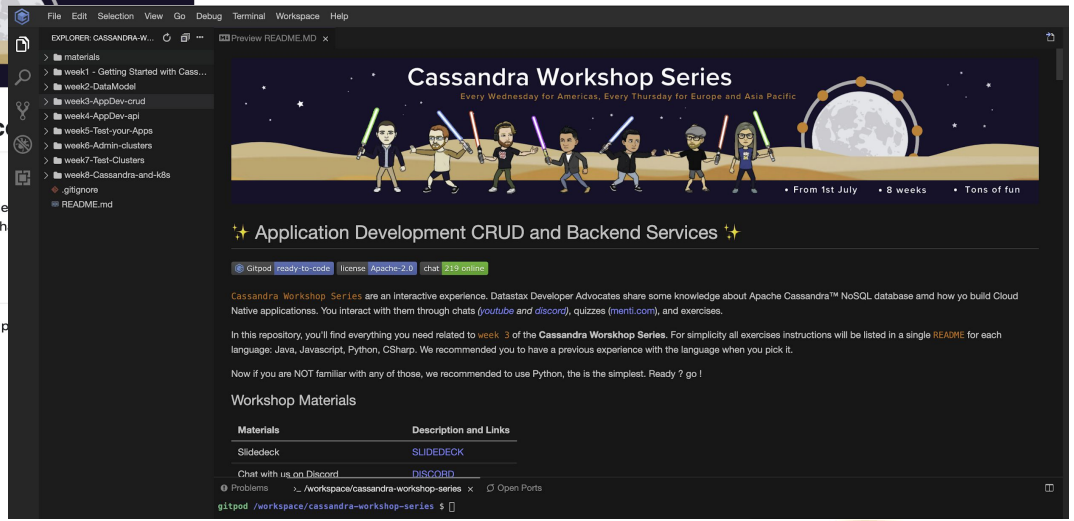
Gitpod ready-to-code | license Apache-2.0 | chat 322 online

`Cassandra Workshop Series` are an interactive experience. Datastax Developer Advocates share some knowle
Cassandra™ NoSQL database amd how yo build Cloud Native applicationss. You interact with them through ch
quizzes (menti.com), and exercises.

🧑‍🏫 MATERIALS OF THE WEEK

In this repository, you'll find everything you need related to `week 3` of the **Cassandra Worskhop Series**. As a p
have you Astra instance ready and running. You can catchup by creating your Astra instance

Materials	Description and Links
🧑‍🏫Slidedeck	Slidedeck for the workshop
📹 Recording LIVESTREAM NA, LATAM	📅 July 15th 12EDT
📹 Recording LIVESTREAM APAC, EMEA	📅 July 16th 12:30 IST
📗 Homeworks	here

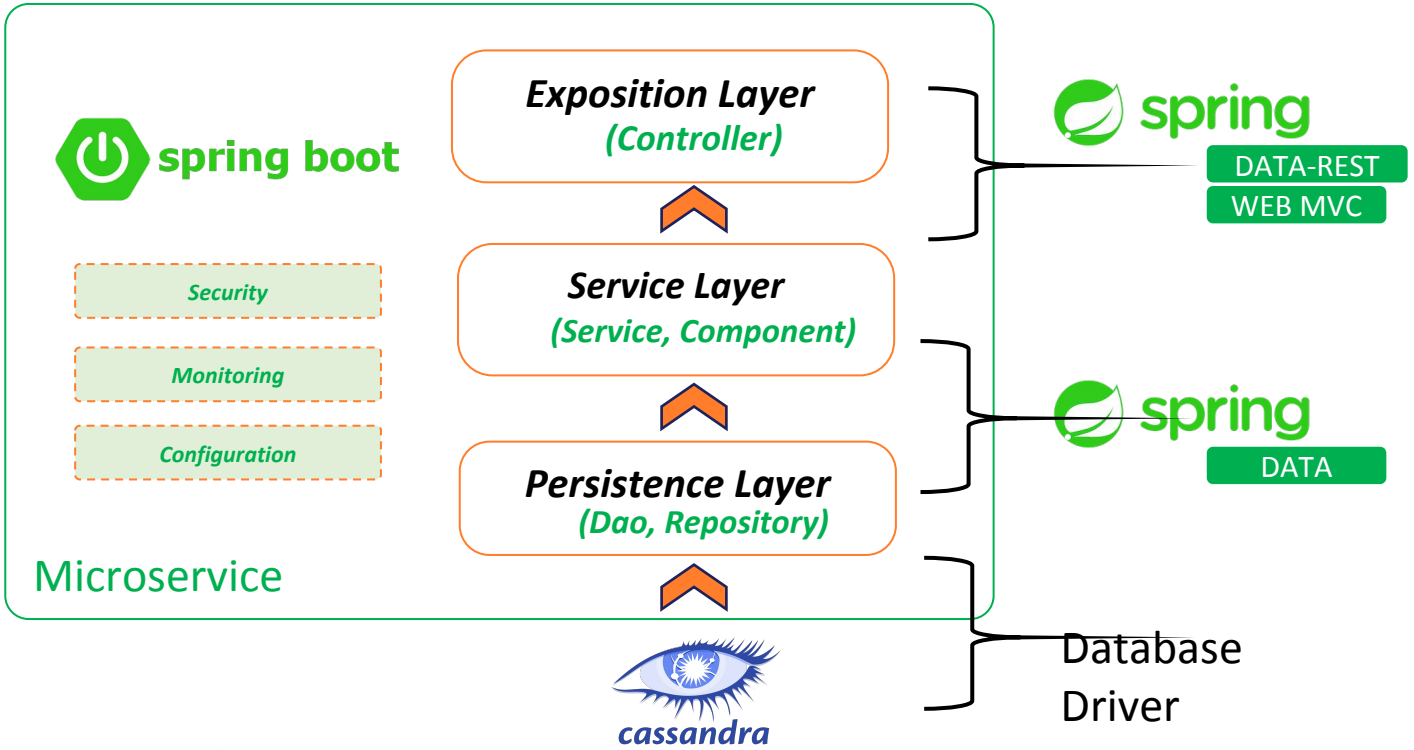


MATERIALS → bit.ly/CassandraWorkshopMaterials

DATASTAX + Apache Cassandra

Application Development **CRUD**

1. Housekeeping
2. Demo & Use Case Definition
3. Connectivity to Cassandra
4. Execute Queries and Statements
5. Parsing Results and Mappings
6. Spring Framework (Java)


spring boot
Exposition Layer
(Controller)
Service Layer
(Service, Component)
Persistence Layer
(Dao, Repository)
Security
Monitoring
Configuration
Microservice
spring
DATA-REST
WEB MVC
spring
DATA
cassandra
Database
Driver

Convention over configuration

```
# ------------------------------------------------
# Full Convention
# ------------------------------------------------
spring:
  data:
    cassandra:
      contact-points: localhost
      port: 9042
      local-datacenter: dc1
      keyspace-name: betterbotz
      schema-action: create-if-not-exists
```

```
@Configuration
public class SpringDataCassandraJavaConfig
                extends AbstractCassandraConfiguration
                implements CqlSessionBuilderCustomizer {

    @Override
    protected String getKeyspaceName() {
        return keyspaceName;
    }

    @Override
    protected String getLocalDataCenter() {
        return localDataCenter;
    }
```

Entity and Repository

```
@Entity
public class Task {

@Id
@PrimaryKeyColumn(
    name = "uid", ordinal = 0,
    type = PrimaryKeyType.PARTITIONED)
private UUID uid;

private String title;

private boolean complete;

private int offset;

private Task() {}

//...
```

```
public interface TaskRepository extends
CassandraRepository<Task, UUID> {

@Query("SELECT * FROM todos_tasks WHERE uid=?0" )
Optional<TaskSpringData> findByTaskByIdO(UUID
taskid);


}
```

Homework Week 3

1. **Learn**
    a. Keep working on DS220 (this is long)
    b. Visit https://github.com/datastax-examples = tons of samples.

2. **Practice**
    a. Finish workshop exercises if needed following github.
    b. Try to run it on your laptop
    c. Bonus with docker-compose make it connect to local instance

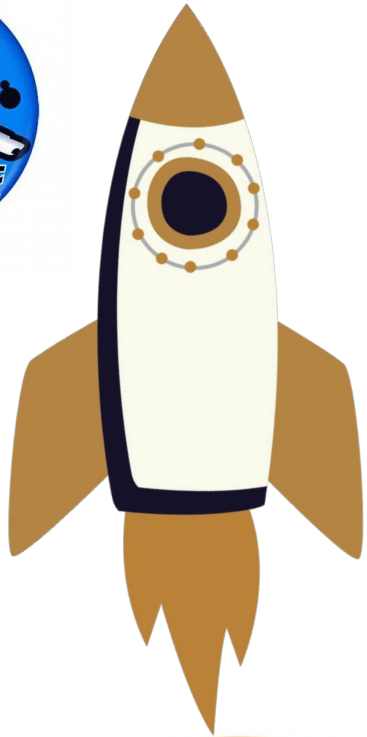
3. **Validation form of the week :** https://forms.gle/mtdzFoVGSoZ2vYa36

Engage !

Share with us you Cassandra use cases !

Share with your vision and future of Cloud Native

Share what you need to succeed with Cassandra.



Developer Resources

LEARN

- Join academy.datastax.com
- Browse www.datastax.com/dev

ASK/SHARE

Join community.datastax.com

Ask/answer community user questions - share your expertise

CONNECT

Follow us

We are on Youtube - Twitter - Twitch!

MATERIALS

Slides and code for this course are available at https://github.com/DataStax-Academy/cassandra-workshop-series

DATASTAX®

Thank You