

DATASTAX

A large, stylized arrow pointing to the right, composed of several overlapping diagonal bands in shades of red, orange, and yellow.

# DataStax Enterprise

Foundations of Apache Cassandra™

**DS201 - sections 1 & 2**

July 27th, 2023

# ➤ Housekeeping

# ➤ Introductions

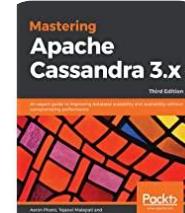
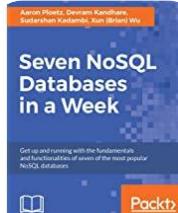
- Name
- Which team you work on
- Prior Cassandra experience (beginner, intermediate, expert)

# Aaron Ploetz

Developer Advocate /  
Author



- 3x Cassandra MVP
- Prior DB Engineering Lead @  
 & 
- Host - Apache Cassandra® Corner  
podcast
- Books:
  - Mastering Apache Cassandra 3.x
  - Seven NoSQL Databases in a Week





# Code with Java 21





## Minneapolis St. Paul Cassandra Meetup

Minneapolis, MN, USA

682 members · Public group

Organized by **DataStax** and 7 others

Share:

Join this group



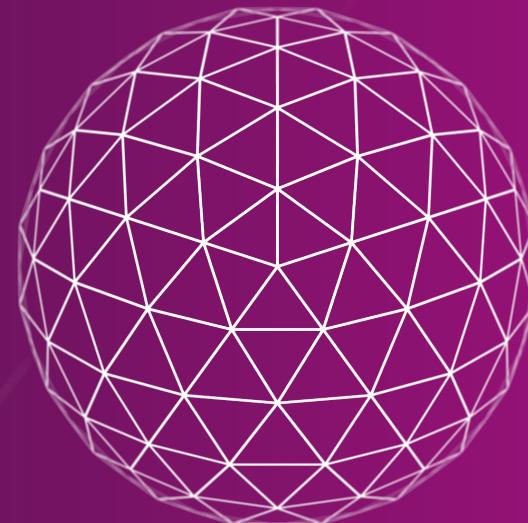
## ➤ Exercises



<https://github.com/aar0np>

/DS-201-DSE

# » #1 - Introduction





# Learning Objectives

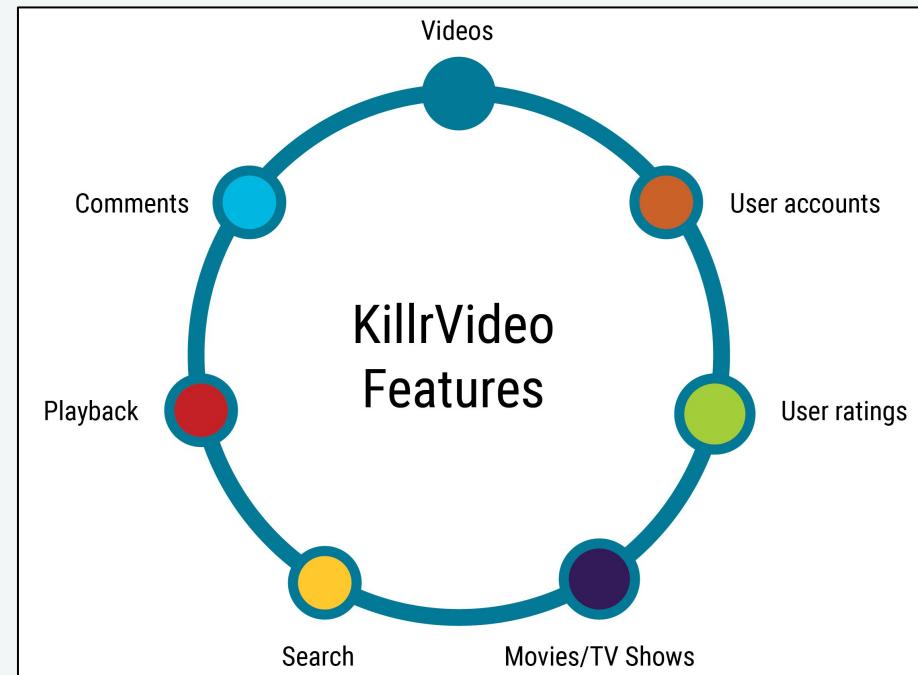
- › Explore KillrVideo domain
- › Install / Start Apache Cassandra™
- › Create tables, store, and retrieve data
- › Understand the Cassandra data model
- › Understand Cassandra architecture
- › Building an application w/ Java

# ➤ KillrVideo



# › KillrVideo Inc.

KillrVideo is a(nother) video sharing website.



# › KillrVideo - Problems

## Scalability

KillrVideo constantly adds new users and videos.

## Reliability

KillrVideo must always be available.

## Ease of use

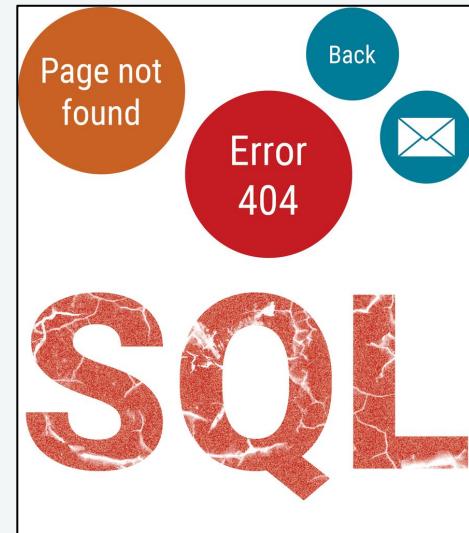
KillrVideo must be easy to manage and maintain.



# › KillrVideo - Solutions attempted

## Relational database problems.

- Single points of failure
- Scaling complexity
- Reliability issues
- Difficult to serve users worldwide



# › KillrVideo - Apache Cassandra™

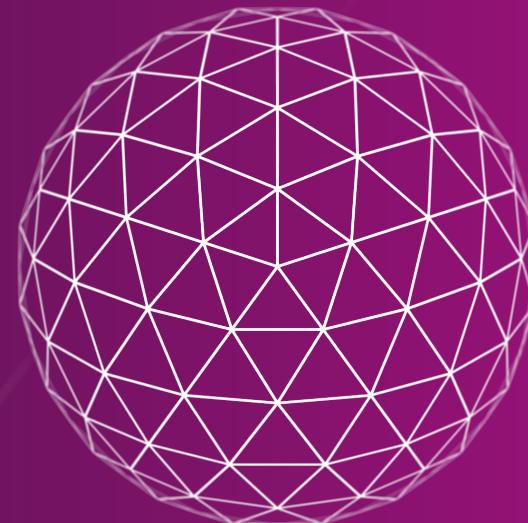


## Why Cassandra?

- Peers instead of leader / follower
- Linear scale performance
- Always on reliability
- Data can be stored geographically close to clients



## » #2 - Quick Wins



# ➤ Install options

## DataStax Enterprise with OpsCenter, DevCenter, and Drivers

- <http://www.datastax.com/download>
- Integrated search and analytics tools
- Free for development
- License required for production

## Apache Cassandra™ (open source)

- <http://cassandra.apache.org/download>
- <http://github.com/apache/cassandra>

# › Or... Don't install anything!

**DataStax Astra DB** with SAIs, Unified Compaction Strategy, Vector Search

- <http://astra.datastax.com/>
- \$25 in credits/month free
- No-touch scalability
- Multi-region, multi-cloud



# › Today

- For this course, we will be using DataStax Enterprise 6.8.37.
- The exercises will guide you through the details.
- But as far as the concepts are concerned, the Cassandra and DSE are nigh-identical.



## › However...

I will be sure to call out any differences between **Cassandra**, **DSE**, and **Astra DB** as they arise.

# › Cassandra Tarball Install

- Requires Java 8 - 11.
- Download and extract a simple tar.gz file.

```
$ curl -OL https://archive.apache.org/dist/cassandra/4.1.0/apache-cassandra-4.1.0-bin.tar.gz  
  
$ tar xf apache-cassandra-4.1.0-bin.tar.gz  
  
$ cd apache-cassandra-4.1.0/bin  
  
$ ./cassandra
```

# › DataStax Enterprise Tarball Install

- Requires Java 8
- Download and extract a simple tar.gz file

```
$ curl -OL https://downloads.datastax.com/enterprise/dse-6.8.tar.gz  
  
$ tar xf dse-6.8.tar.gz  
  
$ cd dse-6.8.37/bin  
  
$ ./dse cassandra
```

# › DSE Tarball contents

```
-rw-r--r--  1 ubuntu  ubuntu  26530 Mar 30 08:27 LICENSE.txt
-rw-r--r--  1 ubuntu  ubuntu   3064 Mar 30 08:27 README.md
drwxr-xr-x  2 ubuntu  ubuntu   4096 Mar 30 08:27 bin
drwxr-xr-x  2 ubuntu  ubuntu   4096 Mar 30 08:27 clients
drwxr-xr-x 11 ubuntu  ubuntu   4096 Mar 30 08:27 demos
-rw-r--r--  1 ubuntu  ubuntu     69 Mar 30 08:27 ds_branch.txt
-rw-r--r--  1 ubuntu  ubuntu    24 Mar 30 08:27 ds_timestamp.txt
-rw-r--r--  1 ubuntu  ubuntu    99 Mar 30 08:27 ds_version.txt
drwxr-xr-x  3 ubuntu  ubuntu   4096 Mar 30 08:27 javadoc
drwxr-xr-x  2 ubuntu  ubuntu   4096 Mar 30 08:27 lib
drwxr-xr-x  2 ubuntu  ubuntu   4096 Mar 30 08:28 licenses
drwxr-xr-x 16 ubuntu  ubuntu  4096 Mar 30 08:27 resources
drwxr-xr-x  4 ubuntu  ubuntu   4096 Mar 30 08:27 tools
```

# ➤ Nodetool

## nodetool status

- Connects via JMX on port 7199
- Access configured in each node via **cassandra-env.sh**

```
Datacenter: Cassandra
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving/Stopped
--  Address      Load      Owns (effective)  Host ID                  Token      Rack
UN  127.0.0.1  107.07 MiB  100.0%          4f241a95-5af0-469f-a103-4c7007828562  0          rack1
```

# Pop › Quiz!

## Is this node reachable?

--	Address	Load	Owns (effective)
UN	127.0.0.1	107.07 MiB	100.0%



# No

# Tips

## From the Trenches

### How can I figure out which IP address a node is bound to?

```
$ grep _address conf/cassandra.yaml | grep -v \#  
  
broadcast_address: 10.6.5.5  
broadcast_rpc_address: 10.6.5.5  
listen_address: 192.168.0.18  
rpc_address: 192.168.0.18
```

## ➤ Exercise #1



# Hands-on Exercise #1

- › Download the DSE tarball
- › Extract DSE
- › Start DSE Cassandra
- › Verify that DSE is running



# CQL Fundamentals

- › CQL
- › Keyspaces
- › Tables
- › Data types

# > CQL

## Cassandra Query Language

- Similar to SQL
- Some similar keywords behave differently.

```
> SELECT *\n      FROM users;
```

# Tips

## › From the Trenches

Do not actually run a query without a WHERE clause.

```
> SELECT * FROM users;  
  
com.datastax.driver.core.exceptions.  
ReadTimeoutException
```

# › Keyspaces

## What is a keyspace?

- Top-level namespace / container
- Similar to a relational database schema

```
> CREATE KEYSPACE killrvideo
    WITH REPLICATION = {
        'class': 'NetworkTopologyStrategy',
        'west_dc1' : 3, 'east_dc1' : 3
    };
```

# ➤ Use

## The “USE” command:

- USE switches between keyspaces.

```
> USE killrvideo;
```

# ➤ Tables

## Working with tables:

- Keyspaces contain tables.
- Tables contain data (rows).

```
> CREATE TABLE table1 (
    column1 TEXT,
    column2 TEXT,
    column3 INT,
    PRIMARY KEY (column1)
);
```

```
> CREATE TABLE users (
    user_id UUID,
    first_name TEXT,
    last_name TEXT,
    PRIMARY KEY (user_id)
);
```

# › CQL Data Types

Data Type	Details	Java Type
text	<ul style="list-style-type: none"><li>• UTF8 encoded</li><li>• same as varchar</li></ul>	String
smallint	<ul style="list-style-type: none"><li>• Signed, 16 bits</li></ul>	short
int	<ul style="list-style-type: none"><li>• Signed, 32 bits</li></ul>	int
bigint	<ul style="list-style-type: none"><li>• Signed, 64 bits</li></ul>	long
inet	<ul style="list-style-type: none"><li>• IP address</li><li>• IPv4 or IPv6</li></ul>	InetAddress

# › CQL Data Types (*part 2*)

Data Type	Details	Java Type
float	32-bit IEEE-754 fixed precision	float
double	64-bit IEEE-754 fixed precision	double
decimal	Variable precision floating point type	BigDecimal
blob	Raw, arbitrary bytes	ByteBuffer
vector	Array of float	Object

# Pop Quiz!

# What is...

$$0.1 + 0.2 = ?$$

Answer at: <https://0.3000000000000004.com/>

# › CQL Data Types (*part 3*)

Data Type	Details	Java Type
date	<ul style="list-style-type: none"><li>Date only</li></ul>	LocalDate
time	<ul style="list-style-type: none"><li>Time only, nanosecond precision</li></ul>	LocalTime
timestamp	<ul style="list-style-type: none"><li>Date and time, millisecond precision.</li></ul>	Instant
timeuuid	<ul style="list-style-type: none"><li>Type 1 UUID</li></ul>	UUID
uuid	<ul style="list-style-type: none"><li>Type 4 UUID</li></ul>	UUID

# › UUID and TimeUUID

Typically use in place of “autonumber” requirements

- Universally Unique Identifier
  - Ex: 52b11d6d-16e2-4ee2-b2a9-5ef1e9589328
  - Generate via `uuid()`
- TIMEUUID embeds a TIMESTAMP value
  - Ex: 1be43390-9fe4-11e3-8d05-425861b86ab6
  - Generate via `now()`

# Tips

## › From the Trenches

Want to know the time on your cluster?

```
> SELECT toTimestamp(now())
  FROM system.local;
```

2023-07-19 18:30:37.998000+0000

> **INSERT**

Similar to SQL syntax

```
> INSERT INTO users (user_id, first_name, last_name)  
VALUES (uuid(), 'Joseph', 'Chu');
```

# ➤ UPDATE

Similar to SQL syntax

```
> UPDATE users
    SET first_name='Joseph', last_name='Chu'
    WHERE user_id=uuid();
```

# Tips

## › From the Trenches

**INSERT and UPDATE are interchangeable in Cassandra!**

They are both writes.  
Technically, so is a DELETE.

> 

# SELECT

## Similar to SQL syntax

```
> SELECT first_name, last_name
   FROM users
 WHERE user_id = 4b516be3-ddf0-4c43-bab6-b91d674b64a5;

> SELECT *
   FROM users
 WHERE user_id = 4b516be3-ddf0-4c43-bab6-b91d674b64a5;
```

# › COPY

## Similar to SQL syntax

- Imports / exports CSV (comma-separated values)

```
COPY table1 (column1, column2, column3) FROM 'table1data.csv';
```

- Header parameter skips the first line in the file

```
COPY table1 (column1, column2, column3) FROM 'table1data.csv'  
WITH HEADER=true;
```

› **COPY** (*continued*)

**There are several ways to get data into Apache Cassandra™**

- COPY
- DataStax Bulk Loader
- Drivers
- Apache Spark™

## ➤ Exercise #2



# Hands-on Exercise #2

- › Create a keyspace for KillrVideo
- › Create a table to store video metadata
- › Load the data for the video table from a CSV file