

Introduction to Cassandra 1.2, CQL 3 and the Datastax Java Driver

Course Summary

Description

The Cassandra (C*) NoSQL database is one of the most powerful and widely used non-relational databases available today. It is a fault tolerant, highly scalable database with tunable consistency that meets the demanding requirements of the "can't fail, must scale" systems driving growth for many of the most successful enterprises of today. However, along with that capability comes a new data and programming model that many organizations lack the expertise to use in an optimal fashion.

This course provides a technical introduction to all the conceptual and practical areas needed to use Cassandra successfully. It is written expressly for the new capabilities in the recent versions of C*, including CQL3 and the new DataStax Java driver. The course provides a solid foundation in the architecture and data model of C* and how to work with it. It covers CQL3 in detail, as well as important data modeling techniques to optimize your usage of the database. It includes in-depth coverage of the new Java driver for C*, as well as a full-scale application based on a stock-trading system (StockWatcher) that uses the driver.

After taking this course, you will be ready to work with Cassandra in an informed and productive manner, including using CQL3 and the new Java driver. You will be aware of some common pitfalls as well as best practices for creating your data model and applications. You will gain a clear understanding of how C* works, and be fully prepared to use it in production systems.

Objectives

By the end of this course, students will be able to:

- Understand the motivation for non-relational data stores
- Be familiar with Cassandra at a high-level
- Be familiar with basic installation / setup of Cassandra, and how an installation is structured
- Understand the Cassandra architecture, including the cluster structure and partitioners
- Understand and use data replication and eventual consistency with Cassandra
- Learn the basics of the Cassandra data model, and how to create good data models
- Use CQL 3 to create tables and execute queries
- Learn and use cqlsh
- Know the standard CQL data types
- Understand and use primary keys, compound primary keys, and composite partition keys
- Understand and use collections, secondary indexes, counters, and batches
- Understand the structure of the Java driver
- Use the basic Java API to connect to and work with Cassandra
- Use QueryBuilder to build dynamic queries
- Use asynchronous queries

Topics

- Introduction to Cassandra
- Overview of Architecture and Data Model
- The Cassandra Data Model
- Data Consistency
- How Things Work
- The Java Driver

Prerequisites

Reasonable Java experience for the Java driver labs, some knowledge of databases

Duration

Three days

Introduction to Cassandra 1.2, CQL 3 and the Datastax Java Driver

Course Outline

I. Introduction to Cassandra

A. Overview

1. The motivation for non-relational data stores
2. Why relational databases don't support modern applications well
3. Cassandra at a high-level
4. Use cases
5. Features Strengths (Scalability, robustness, linear performance with scale-out), etc.
6. High Level Structure

B. Acquiring and Installing Cassandra

C. Configuring and Installation Structure

LABS:

- Configure, Start/Stop Cassandra
- StockWatcher Demo

II. Overview of Architecture and Data Model

A. Basic Cassandra Architecture

1. Cluster Structure - Nodes, Virtual Nodes, Ring Topology
2. Consistent Hashing, Tokens, Partitioners, and Data Distribution
3. Data Replication, the Replication Factor, Keyspaces
4. Consistency, the CAP theorem, Eventual Consistency

B. The C* Data Model

1. Data Model and CQL 3 Introduction
2. Using CQL and cqlsh
3. Single primary key tables and how to define them using CQL
4. Inserting Data (INSERT), Data Distribution in the Ring, Upsert
5. Querying for Data (SELECT)
6. CQL Data Types
7. Working with Primary Keys

LABS:

- Spin up the Lab Cluster
- Create Simple Tables
- Insert/Query Tables
- Use copy to Populate a Table

III. The Cassandra Data Model

A. Compound Primary Keys

1. CQL table definition
2. The partition key and clustering columns
3. CQL Mapping vs. Internal Storage View

B. Other Capabilities

1. Expiring Columns / Time To Live (TTL)
2. Batches
3. Clustering order, ORDER BY, and CLUSTERING ORDER BY
4. Filtering results and ALLOW FILTERING

C. Composite Partition Keys

1. Motivation and uses
2. CQL Definition
3. Effect on Partitioning and Internal Storage View

D. Indexes and Secondary Indexes

1. Partition Key Indexes, token()
2. Non-primary Key (Secondary) Indexes
3. Guidelines and Querying

E. Understand and Use Counters

1. Motivation and Uses
2. Structure, Characteristics, CQL, Usage
3. Limitations

F. Understand and use collections

1. Motivation and uses
2. CQL definition (set, list, and map)
3. Inserting, Updating, Deleting with a Collection
4. Limitations and Internal Storage View

LABS:

- Introduce Compound Primary Keys
- Refine Compound Primary Key
- Work with Composite Partition Keys
- Secondary Indexes
- Counters
- Using Collections

Introduction to Cassandra 1.2, CQL 3 and the Datastax Java Driver

Course Outline (con't)

IV. Data Consistency

- A. Data Consistency in Cassandra
 - 1. Tunable Consistency
 - 2. CAP Theorem
 - 3. Coordinators and Client Requests
 - 4. Consistency Levels in C* - ONE, QUORUM, ALL
 - 5. Configuring Immediate Consistency
 - 6. CL ONE is Your Friend
 - 7. Other CL Levels
- B. Repair Mechanisms
 - 1. Read Repair
 - 2. Hinted Handoff

LABS:

- Data Consistency

V. How Things Work

- A. Write Failures
 - 1. Unavailability, Node Failure
 - 2. Requirements for Writing
- B. Key and Row Caches
 - 1. Cache Overview
 - 2. Guidelines
- C. Multi-Data Center Support
 - 1. Overview
 - 2. Replication Factor Configuration
 - 3. Consistency Levels - LOCAL/EACH QUORUM
- D. Deletes
 - 1. CQL
 - 2. Tombstones
 - 3. Issues and Guidelines

LABS:

- Deletes

VI. The Java Driver

- A. Introduction
 - 1. Overview and Architecture
 - 2. Features
- B. API Introduction
 - 1. Cluster and Cluster.Builder
 - 2. Creating the Cluster
 - 3. Contact Points
 - 4. Getting a Session
 - 5. Working with Sessions
- C. Querying
 - 1. PreparedStatement, Query, BoundStatement

- 2. Using PreparedStatements, Binding Values
- 3. Executing the Query
- 4. Processing Query Results
- 5. CQL to Java Type Mapping
- 6. Working with UUIDs
- 7. Setting Consistency Level

D. QueryBuilder and Dynamic Queries

- 1. Dynamic Queries
- 2. QueryBuilder - Fluent API for Queries
- 3. Building SELECT, Select Type, Select.where(), Chaining WHERE Clauses
- 4. Building DELETE, Delete Type, Delete.where()
- 5. Building INSERT, Insert Type, Insert.insertInto(), Insert.value()
- 6. Building Update, Update Type, Insert.with(), Insert.value()
- 7. Other Queries

E. Asynchronous Querying

- 1. Normal vs Asynchronous Querying
- 2. Interface java.util.concurrent.Future
- 3. ResultSetFuture
- 4. Querying Asynchronously and Processing Future Results
- 5. Listeners for Processing

F. Driver Policies

- 1. Overview
- 2. Load Balancing Policies - RoundRobinPolicy, DCAwareRoundRobinPolicy
- 3. Retry Policies - DefaultRetryPolicy, DowngradingConsistencyRetryPolicy
- 4. The Policies Class

LABS:

- Introducing StockWatcher (the Lab Domain)
- Connect to a Cluster
- Execute Queries
- Use QueryBuilder for Select
- Use QueryBuilder More Extensively
- Use executeAsync() and ResultSetFuture
- Use Driver Policies
- StockWatcher Deep Dive - Examining a Full-fledged Application