BYTEQUEST

Immersive Data Engineering Boot Camp – Detailed Course Curriculum

Section 1: Hadoop Version 2.x

Introduction to Big Data
What is Hadoop?
The Hadoop Distributed File System (HDFS)
MapReduce (Data Processing Framework):
- How MapReduce works?
- Developing a MapReduce Application
- MapReduce Features

YARN (Cluster Resource Manager):
- Anatomy of a YARN Application Run
- Scheduling in YARN

Hive (Data Warehouse Framework):
- HiveQL
- Comparison with Traditional Databases:
  - Schema on Read Versus Schema on Write
  - Updates, Transactions, and Indexes

Section 2: Spark Version 2.x

What's Spark?
Spark Architecture:
- Basic Architecture
- Cluster Management: Standalone, YARN, Mesos

Spark EcoSystem:
- Spark SQL, Spark MLLib, GraphFrames, Spark Streaming

Just a Little Scala (Version 2.x) for Spark
Spark Architecture Internals:
- Scheduling and Executing Jobs and Tasks
- Shuffling and Performance
- Partitioning of Data Sources
- Data Reads and Writes

DataFrames, DataSets and Spark SQL:
- DataFrames Programming API
- Spark SQL
- The Catalyst Query Optimizer
- The Tungsten In-Memory Data Format
- The Dataset API, Encoders, and Decoders
- A Review of RDDs

Spark's MLlib API for Machine Learning:
- Logistic Regression (Classification),
- K-Means (Clustering) & Recommendation Engines

Graph Processing with GraphFrames
Spark Structured Streaming
Advanced Spark Programming:
- Accumulators
- Broadcast Variables
- Piping to External Programs

Spark in Production:
- Cluster Sizing (Capacity Planning)
- Tuning - Configuration Management
- Caching and Storage Levels
- Debugging & Logging Mechanisms
- Analyzing Spark Jobs using Spark UI

Section 3: Kafka Version 0.10.x

Introduction to Messaging Systems
- Real-Time Big Data (Fast Data)
- Message Queues
- Streaming Architecture
- Lambda Architecture

Introduction to Kafka:
- What's Kafka?
- Why Kafka?
- Use cases for Kafka

Kafka Architecture:
- Brokers, ZooKeeper, Producers, Consumers

Kafka Producers: How to Write Messages to Kafka?
- Creating a Kafka Producer
- Sending a Message to Kafka Synchronously
- Sending Messages Asynchronously
- Configuring Producers
- Using Serializers
- Partitions

Kafka Consumers: How to Read Messages from Kafka?
- Consumers and Consumer Groups
- Creating a Kafka Consumer
- Subscribing to Topics
- Polling
- Configuring Consumers
- Commits and Offsets
- Rebalancing
- Deserializers
- Stand Alone Consumer

Kafka Architecture Internals:
- How Kafka Replication works?
- How Kafka handles Requests from Producers and Consumers?
- How Kafka Stores Data?

Building Data Pipelines:
- Considerations
- Kafka Connect vs Producer & Consumer

Kafka in Production:
- Security
- Capacity Planning
- Monitoring

Section 4: Cassandra Version 3.x

Beyond Relational Databases:
- What's the problem with Relational Databases?
- The Era of NoSQL

Introduction to Cassandra & NoSQL:
- What's Cassandra?
- Why Cassandra?
- Use cases for Cassandra

Cassandra Architecture:
- Distributed Architecture:
  - Node, Ring (Cluster), Drivers, Peer-to-Peer, Vnodes, Gossip & Snitch
- Replication and Consistency:
  - Replication, Consistency, Hinted HandOff & Read Repair

Cassandra Architecture Internals:
- Write Path
- Read Path
- Compaction

Introduction to CQL:
- Clusters, Keyspaces & Tables
- Rows and Columns
- Choosing Primary Keys (Identifying Partition Keys and Clustering Columns)
- Querying with CQL
- Working with TimeStamps
- Time to live (TTL)
- CQL UPDATE command
- CQL Datatypes
- CQL Collections
- User Defined Types
- Secondary Indexes

Cassandra Data Modeling:
- Conceptual Data Modeling:
  - Relationship Keys, Hierarchy
- Logical Data Modeling:
  - Chebotko Diagrams, Modeling Methodologies & Mapping Rules
- Physical Data Modeling:
  - Partition Size, Data Duplication, Data Consistency, Key Optimizations & Table Optimizations

Cassandra in Production:
- Choosing Proper Hardware
- Cluster Sizing (Capacity Planning)
- Monitoring:
  - Linux Observation Tools
  - System and Debug Logs
  - Generating Heap Dumps
  - Diagnosing Issues with JVM (JVM Profiling):
    - NodeTool Utility
    - JMX Clients: jconsole, visualvm

Section 5: Individual Capstone Project

Building End to End Real-Time Distributed Data Pipeline in Cloud Environment