

HDP Developer: Storm and Trident

Learn via: **Classroom**

Duration: **4 Day**

Overview

This course provides a technical introduction to the fundamentals of Apache Storm and Trident that includes the concepts, terminology, architecture, installation, operation, and management of Storm and Trident. Simple Storm and Trident code excerpts are provided throughout the course. The course also includes an introduction to, and code samples for, Apache Kafka. Apache Kafka is a messaging system that is commonly used in concert with Storm and Trident.

Prerequisites

Students must have experience developing Java applications and using a Java IDE. Labs are completed using the Eclipse IDE and Gradle. Students should have a basic understanding of Hadoop.

Who Should Attend

Hadoop developers who need to be able to design and build Storm and Kafka applications using Java and the Trident API.

What You Will Learn

At the completion of the course students will be able to:

- Recognize differences between batch and real-time data processing
- Define Storm elements including tuples, streams, spouts, topologies, worker processes, executors, and stream groupings
- Explain and install Storm architectural components, including Nimbus, Supervisors, and ZooKeeper cluster
- Recognize/interpret Java code for a spout, bolt, or topologies
- Identify how to develop and submit a topology to a local or remote distributed cluster
- Recognize and explain the differences between reliable and unreliable Storm operation
- Manage and monitor Storm using the command-line client or browser-based Storm User Interface (UI)
- Define Kafka topics, producers, consumers, and brokers
- Publish Kafka messages to Storm or Trident topologies
- Define Trident elements including tuples, streams, batches, partitions, topologies, Trident spouts, and operations
- Recognize and interpret the code for Trident operations, including filters, functions, aggregations, merges, and joins
- Recognize the differences between the different types of Trident state
- Identify how Trident state supports exactly-once processing semantics and idempotent operation
- Recognize the differences in fault tolerance between different types of Trident spouts
- Recognize and interpret the code for Trident state-based operations