

Training: Cloudera
 Developer Training For Spark and Hadoop


| FORM OF TRAINING | MATERIALS | PRICE | DURATION |
|-------------------|-------------------|----------|----------|
| Traditional | Digital materials | 2180 EUR | 4 days |
| Traditional | CTAB Tablet | 2330 EUR | 4 days |
| Distance learning | Digital materials | 2180 EUR | 4 days |
| Distance learning | CTAB Tablet | 2180 EUR | 4 days |

LOCATIONS

Krakow - 5 Tatarska Street, II floor, hours: 9:00 am - 4:00 pm
 Warsaw - 17 Bielska Street, hours: 9:00 am - 4:00 pm

TRAINING TERMS

2019-10-28 | 4 days | Prague
 2019-11-04 | 4 days | Warszawa
 2019-12-02 | 4 days | Kraków

TRAINING GOALS:

This four-day hands-on training course delivers the key concepts and expertise participants need to ingest and process data on a Hadoop cluster using the most up-to-date tools and techniques. Employing Hadoop ecosystem projects such as Spark, Hive, Flume, Sqoop, and Impala, this training course is the best preparation for the real-world challenges faced by Hadoop developers. Participants learn to identify which tool is the right one to use in a given situation, and will gain hands-on experience in developing using those tools.

Through instructor-led discussion and interactive, hands-on exercises, participants will learn Apache Spark and how it integrates with the entire Hadoop ecosystem, learning:

- How data is distributed, stored, and processed in a Hadoop cluster
- How to use Sqoop and Flume to ingest data
- How to process distributed data with Apache Spark
- How to model structured data as tables in Impala and Hive
- How to choose the best data storage format for different data usage patterns
- Best practices for data storage

CONSPECT:

- Introduction to Hadoop and the Hadoop Ecosystem
 - Problems with Traditional Large-scale Systems
 - Hadoop!
 - The Hadoop EcoSystem
- Hadoop Architecture and HDFS
 - Distributed Processing on a Cluster
 - Storage: HDFS Architecture
 - Storage: Using HDFS
 - Resource Management: YARN Architecture
 - Resource Management: Working with YARN
- Importing Relational Data with Apache Sqoop
 - Sqoop Overview
 - Basic Imports and Exports
 - Limiting Results
 - Improving Sqoop's Performance
 - Sqoop 2
- Introduction to Impala and Hive
 - Introduction to Impala and Hive
 - Why Use Impala and Hive?
 - Comparing Hive to Traditional Databases
 - Hive Use Cases
- Modeling and Managing Data with Impala and Hive
 - Data Storage Overview
 - Creating Databases and Tables
 - Loading Data into Tables
 - HCatalog
 - Impala Metadata Caching
- Data Formats
 - Selecting a File Format
 - Hadoop Tool Support for File Formats
 - Avro Schemas
 - Using Avro with Hive and Sqoop
 - Avro Schema Evolution
 - Compression

- Data Partitioning
 - Partitioning Overview
 - Partitioning in Impala and Hive
- Capturing Data with Apache Flume
 - What is Apache Flume?
 - Basic Flume Architecture
 - Flume Sources
 - Flume Sinks
 - Flume Channels
 - Flume Configuration
- Spark Basics
 - What is Apache Spark?
 - Using the Spark Shell
 - RDDs (Resilient Distributed Datasets)
 - Functional Programming in Spark
- Working with RDDs in Spark
 - A Closer Look at RDDs
 - Key-Value Pair RDDs
 - MapReduce
 - Other Pair RDD Operations
- Writing and Deploying Spark Applications
 - Spark Applications vs. Spark Shell
 - Creating the SparkContext
 - Building a Spark Application (Scala and Java)
 - Running a Spark Application
 - The Spark Application Web UI
 - Configuring Spark Properties
 - Logging
- Parallel Programming with Spark
 - Review: Spark on a Cluster
 - RDD Partitions
 - Partitioning of File-based RDDs
 - HDFS and Data Locality
 - Executing Parallel Operations
 - Stages and Tasks
- Spark Caching and Persistence

- RDD Lineage
- Caching Overview
- Distributed Persistence
- Common Patterns in Spark Data Processing
 - Common Spark Use Cases
 - Iterative Algorithms in Spark
 - Graph Processing and Analysis
 - Machine Learning
 - Example: k-means
- Preview: Spark SQL
 - Spark SQL and the SQL Context
 - Creating DataFrames
 - Transforming and Querying DataFrames
 - Saving DataFrames
 - Comparing Spark SQL with Impala
- Conclusion

REQUIREMENTS:

This course is designed for developers and engineers who have programming experience. Apache Spark examples and hands-on exercises are presented in Scala and Python, so the ability to program in one of those languages is required. Basic familiarity with the Linux command line is assumed. Basic knowledge of SQL is helpful. Prior knowledge of Hadoop is not required.

Difficulty level



CERTIFICATE:

The participants will obtain certificates signed by Cloudera (training completion). Also this course is an excellent place to start for people working towards the CCP: Data Engineer certification. Although further study is required before passing the exam, this course covers many of the subjects tested in the CCP: Data Engineer exam.

TRAINER:

Certified Cloudera Instructor.

ADDITIONAL INFORMATION:

After successfully completing this course, we recommend that participants attend Cloudera's Developer Training for Spark and Hadoop II: Advanced Techniques course, which builds on the foundations taught here.