

Big Data with Hadoop, Spark and Databricks

From Linux learning clusters to selected data-engineering workflows

FA-0591 · 5 days

Data & Analytics · DevOps, Cloud & Infrastructure · Software Development

Explore Hadoop architecture and practise selected HDFS, Spark and Databricks workflows, distinguishing legacy tools and demonstrations from production deployment.

Introduction

This five-day technical course follows the source's detailed agenda: three days of Hadoop foundations, then Spark and Databricks. It combines architecture and administration concepts with selected exercises in a prepared Linux learning environment.

Participants work on small sample datasets and compatible platform releases. Legacy ecosystem tools are discussed for understanding or migration where appropriate. Spark/Databricks labs focus on selected transformations and pipeline demonstrations, not a complete production lakehouse, secure enterprise cluster or certification-preparation programme.

Learning outcomes

The course teaches participants to:

- Explain big-data workload characteristics and Hadoop's HDFS, YARN and MapReduce roles.
- Configure and inspect a prepared Linux Hadoop learning environment.
- Run selected HDFS commands and compare Hive, HBase and legacy ecosystem patterns.
- Use Spark DataFrame/SQL transformations and explore a bounded Structured Streaming example.
- Explain Delta Lake data-management capabilities and compare relevant layout/optimisation options.
- Review monitoring, orchestration, access-control and maintenance requirements before operational deployment.

Prerequisites

- Basic familiarity with the Linux command-line and system administration.
- Programming knowledge in Python, Java, or Scala.
- Working understanding of SQL and data structures.
- Prior experience with databases or scripting is helpful but not mandatory.
- A prepared Linux learning environment with compatible Hadoop, Java and Spark versions.
- Access to an approved Databricks learning workspace with suitable permissions and agreed resource use.
- Sufficient lab resources for selected cluster demonstrations; production services are not configured during the course.

Training outline

1. Day 1 – Hadoop and Big-Data Architecture

Module 1 – Workloads and Core Architecture

- Illustrative big-data workloads and criteria for choosing a distributed platform; avoid undated adoption rankings.
- Defining the 5 Vs: volume, velocity, variety, veracity, value.
- Evaluate when relational databases are sufficient and when distributed processing is justified.
- Core Hadoop architecture: HDFS, YARN, and MapReduce explained in detail.
- HDFS internals: block storage, replication, failover, NameNode and DataNode roles.
- YARN resource scheduling, containers and availability concepts.
- The MapReduce paradigm: execution pipeline, mapping, shuffling, reducing, combiners.

Module 2 – Ecosystem Overview

- Compare Hive and HBase with Pig, Flume and other ecosystem tools; distinguish current deployment choices from legacy integration patterns.

2. Day 2 – Linux Hadoop Learning Environment

Module 3 – Installation and Configuration

- Prepare an isolated Linux lab with appropriate accounts, SSH access and permissions.

- Install compatible Hadoop and Java releases using official requirements and verified distributions; do not assume OpenJDK 8/11 supports every current release.
- Configuration essentials: editing core-site.xml, hdfs-site.xml, yarn-site.xml, and environment variables.
- Initialise only the new isolated learning filesystem and explain why formatting an existing HDFS would be destructive.
- Start, stop and inspect daemons in the prepared lab.

Module 4 — Operations and Security Review

- Accessing and interpreting Hadoop and YARN web interfaces for monitoring.
- Running basic HDFS commands: file upload/download, directory creation, permissions, quotas.
- Troubleshooting common installation and configuration issues.
- Health checks: using hdfs fsck, logs, and cluster status commands.
- Review authentication, authorisation and network isolation; discuss Kerberos-secured mode rather than equate a firewall with full cluster security.

3. Day 3 — Ingestion and Processing Tools

Module 5 — Hive, HBase and Legacy Patterns

- Hive architecture, warehouse/metastore concepts and selected HiveQL queries.
- Pig/Pig Latin as a comparative or legacy ETL pattern, not a mandatory new production deployment.
- HBase: NoSQL concepts, column family design, read/write patterns, integrating with HDFS.
- Explain Oozie's historical scheduling model and its retired status; compare supported orchestration approaches conceptually.
- RDBMS ingestion patterns and streaming connectors; Sqoop is retired and treated as legacy migration context, with selected maintained connector demonstrations.
- Compare MapReduce, Hive, HBase and legacy Pig use cases against operational constraints.

Module 6 — Pipeline Exercise and Monitoring

- Selected sample-data cleansing, transformation and aggregation exercise.
- Performance monitoring and resource tuning at the platform level.

4. Day 4 — Spark Data Processing

Module 7 — Core APIs and Selected Exercises

- Spark architecture: driver, executors, cluster manager, task scheduling, DAG execution.
- Compare RDDs with DataFrames and language-specific Dataset support; use DataFrame/SQL APIs for the core exercise.
- Spark SQL: schema inference, SQL queries, window functions, and UDF integration.
- Selected map/filter/join/grouping and aggregation examples using appropriate APIs.
- Structured Streaming demonstration: event-time windows, watermarks and late-data considerations.
- Performance tuning: partitioning, caching/persistence, shuffle optimization, broadcast variables.
- Introductory selected Spark ML pipeline demonstration and evaluation.

Module 8 — Version-Specific Capabilities

Survey relevant capabilities in the chosen supported Spark/runtime release rather than promise all 'latest' features.

- Spark Connect client/server concepts; introduced before Spark 4.0, not unique to that release.
- ANSI SQL behaviours, session variables and SQL feature compatibility for the selected release.
- Semi-structured data and VARIANT where supported.
- Stateful Structured Streaming and state-management options where supported.
- Structured logging and available observability features.
- Selected Python APIs and data-source extensions, with runtime-specific support checks.
- Compare YARN, standalone and Kubernetes deployment options; do not build all three.

5. Day 5 — Databricks and Delta Lake

Module 9 — Workspace and Pipeline Demonstrations

- Databricks workspace, notebooks and compatible serverless/classic compute options.
- Delta Lake fundamentals: ACID transactions, schema enforcement, time travel, upserts, deletes.
- Build or inspect a small batch pipeline and a prepared streaming example using Delta Lake.
- Compare compaction and applicable data-layout options: liquid clustering for compatible new tables and legacy Z-ordering/partitioning where appropriate; do not combine incompatible layouts.

Module 10 — Operational Design Review

- Monitoring, debugging, and profiling Spark jobs in Databricks.

- Review job orchestration, parameters and CI/CD integration requirements using a selected example.
- Access controls, secrets and audit-log concepts with environment/entitlement checks.
- Collaboration features: notebooks, dashboards, versioning, and team workflows.
- Discussion: Comparing Spark, Databricks, and other modern big data platforms for real-world needs.
- Review the selected workflow, its measured behaviour and remaining validation before production use.
- Identify relevant further study; no exam coverage, certification or independent production readiness is guaranteed.