

Hadoop and Big Data Ecosystem Foundations

Storage, processing, resource management and operational context

FA-0730

Data & Analytics · DevOps, Cloud & Infrastructure

Introduction

Explore Hadoop's storage, processing and resource-management components and the roles of related data tools. Combine architectural explanations with selected supplied examples to understand how an ecosystem workflow is assembled and operated.

Tool availability and setup depend on the chosen supported distribution. Historical components are identified as legacy context; the course does not promise immediate competence in every production scenario or guaranteed development-speed gains.

Learning outcomes

- Explain the responsibilities of HDFS, MapReduce and YARN.
- Use selected HDFS commands and inspect replication, permissions and job evidence.
- Compare the roles of Hive, HBase, Spark and related ingestion/workflow tools.
- Recognise legacy tools and assess compatible alternatives for the selected environment.
- Review cluster planning, monitoring, backup, security, high availability and federation concepts.

Prerequisites

- Basic command-line, file-system, SQL and programming familiarity for practical exercises.
- Access to a compatible supplied Hadoop environment and selected example data; installation privileges only where the lab requires them.

- No fixed vendor cloud account or unverified historical distribution image is assumed.

Training outline

1. Module 1 — Big-data context and architecture

- Big Data Case Studies
- Brief History of Hadoop
- Need for Hadoop
- Hadoop Architecture
- RDBMS vs Hadoop
- Vendor Comparison
- Hardware Recommendations
- Hadoop Installation

Compare trade-offs and a supported lab configuration; do not assume Hadoop is preferable for every dataset.

2. Module 2 — HDFS architecture and practical access

- HDFS Basics
- HDFS Architecture
- Data Read and Write Process
- HDFS Permissions
- Data Replication
- HDFS Accessibility
- HDFS Filesystem Operations
- HDFS Interfaces
- Heartbeats
- Rack Awareness
- distcp

Inspect supplied data, permissions and replication evidence; distinguish replication from a complete backup/recovery strategy.

3. Module 3 — MapReduce

- MapReduce Basics

- MapReduce Workflow
- MapReduce Framework
- Hadoop Data Types
- MapReduce Internals
- Job Formats
- Debugging and Profiling
- Distributed Cache
- Combiner Functions
- MapReduce Streaming
- MapReduce Counters, Sorting and Joins

Trace a small supplied job, inspect counters and failures, and relate its stages to the data flow.

4. Module 4 — YARN resource management

- YARN Infrastructure
- YARN ResourceManager
- YARN ApplicationMaster
- YARN NodeManager
- YARN Container

Separate resource allocation, per-node execution and per-application coordination.

5. Module 5 — Ecosystem services and legacy context

HBase

- HBase Architecture
- HBase Installation
- HBase Configuration
- HBase Schema Design
- HBase Commands
- MapReduce Integration
- HBase Security
- Hive architecture, types and HiveQL through a supported SQL client, such as Beeline where appropriate.
- Introduce Spark's role alongside Hadoop storage/resource management without presenting it as another MapReduce-only interface.

- Pig architecture, Pig Latin, modes and UDFs are legacy/environment-dependent comparison topics.
- Sqoop bulk import/export is historical: Apache Sqoop is retired; review compatible data-movement alternatives.
- Flume and workflow tools are discussed according to the selected distribution and actual supported ingestion requirements.

6. Module 6 — Cluster planning and operations

- Cluster Planning
- Cluster Installation and Configuration
- Cluster Testing
- Cluster Benchmarking
- Cluster Monitoring
- dfsadmin, fsck and balancer
- Hadoop Logging
- Hadoop Data Backup
- Addition and removal of nodes

Review controlled node changes, output validation and a recovery plan in the supplied environment.

7. Module 7 — Security and resilient architecture

- Authentication
- Data Confidentiality
- Configuration
- HDFS HA
- HDFS Federation

Discuss authentication, access controls, confidentiality and configuration alongside HA/federation limits; redundancy does not remove recovery testing needs.