Data Lakehouse Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Data Lakehouse Training explores modern architectural approaches for storing, processing, and analyzing large volumes of data in a cost-effective, reliable, and scalable manner.

The **Data Lakehouse architecture** combines the analytical capabilities of traditional **Data Warehouses** with the flexibility of **Data Lakes**. This approach enables organizations to manage data from multiple sources, perform analytics through different query engines, and build more efficient data processing workflows.

The course covers technologies including **S3-compatible Object Storage**, **Apache Iceberg**, **Medallion Architecture**, **Apache Spark**, **Trino**, **Dremio**, **Kafka**, **CDC**, **OpenMetadata**, **Apache Ranger**, **dbt**, and **Apache Airflow**.

With approximately **60% hands-on activities**, participants gain practical experience in developing data ingestion and transformation processes, building **batch and streaming pipelines**, optimizing query performance, managing **metadata and access policies**, and designing **CI/CD-enabled Lakehouse workflows**.

Prerequisites

Basic knowledge of **SQL and Python**, along with familiarity with **Docker** and **Kubernetes**, is recommended.

Who Should Attend

- Professionals interested in building **modern data platforms**
- **Platform Team Members**
- **Data Analysts**
- **DevOps Engineers**
- **Senior Data Engineers**
- **Data Infrastructure Managers**

Data Lakehouse Training

Duration: **3 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of this training, you will be able to:

- Compare **Data Warehouse, Data Lake, and Lakehouse** architectures.
- Design data storage structures using **S3-compatible Object Storage**.
- Apply **ACID, schema evolution, and time travel** capabilities with **Apache Iceberg**.
- Build a **Medallion Architecture** with **Bronze, Silver, and Gold** layers.
- Execute analytical and federated SQL queries using **Trino and Dremio**.
- Develop data pipelines with **PySpark** and investigate performance issues.
- Create real-time data streams using **Kafka and Spark Structured Streaming**.
- Process changing source data using **CDC and MERGE** operations.
- Manage governance, access control, and auditing with **OpenMetadata and Apache Ranger**.
- Develop testable transformations using **dbt** and create data models based on **Fact, Dimension, Star Schema, and SCD** structures.
- Design **CI/CD-enabled Lakehouse pipelines** for production environments.

Outline

1. Lakehouse Architecture and SQL Engines

Data Lakehouse Architecture

- Introduction to the **Lakehouse** concept and its evolution
- Understanding the transition from **Data Warehouse to Data Lake to Lakehouse**
- Supporting reliable analytics with **ACID guarantees** on cost-effective Object Storage
- Separation of **compute and storage**
- Engine independence across **Spark, Trino, and Dremio**

S3 Object Storage

- S3-compatible storage solutions such as **MinIO, AWS S3, and Ceph**
- **On-premises** storage scenarios
- File formats: **Parquet, ORC, and Avro**

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Benefits of columnar storage
- Addressing the **small file problem** and applying partitioning strategies

Open Table Formats

- Why Parquet alone may not satisfy all Lakehouse requirements
- Understanding **schema evolution, ACID, and time travel**
- Comparing **Apache Iceberg, Delta Lake, and Apache Hudi**
- Apache Iceberg internals: **snapshots, manifests, and metadata layers**
- **Hidden partitioning** and **partition evolution**

Medallion Architecture

- **Bronze:** Storing raw data received from source systems
- **Silver:** Preparing cleaned, validated, and deduplicated datasets
- **Gold:** Building aggregated, business-ready data for analytics and reporting
- Defining responsibilities across layers and establishing **data quality checkpoints**

Introduction to Apache Spark

- The role of **Apache Spark** in Lakehouse architectures
- Data ingestion and transformation capabilities
- Integration with other Lakehouse components

Lab 1: Building a Mini Lakehouse

- Setting up a basic Lakehouse using **MinIO, Apache Iceberg, and Spark**
- Loading initial data into the **Bronze layer**
- Exploring **time travel** functionality

2. SQL Engines and Analytical Querying

SQL Engine Fundamentals

- Understanding independent SQL query engines
- Querying data directly without moving it
- **MPP (Massively Parallel Processing)** architecture
- **Coordinator and worker** execution models

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Combining multiple data sources through **federated queries**

Trino

- **Coordinator, worker, and connector** architecture
- Configuring connectors for **Iceberg, Hive, PostgreSQL, and Kafka**
- Memory management
- **Fault-tolerant execution**
- Use cases involving **ad-hoc analytics, BI, and federated queries**

Dremio

- **Apache Arrow-based** execution architecture
- **Columnar Cloud Cache (C3)**
- Improving query performance through **Reflections**
- **Virtual datasets** and self-service analytics
- **Apache Polaris-based** integrated catalog and Iceberg write support

Comparing Trino and Dremio

- **Trino as a query engine** versus **Dremio as an integrated platform**
- Manual query optimization versus **Reflection-based acceleration**
- Differences in governance and semantic layers
- Selecting a platform based on licensing, operational requirements, and team expertise

Lab 2: Querying Iceberg Tables

- Querying the same **Apache Iceberg** table using **Trino and Dremio**
- Comparing execution plans with **EXPLAIN**
- Creating a **Reflection** in Dremio
- Observing differences in query performance

3. Software Development with Spark, Metadata, and Governance

This section focuses on **Spark-based data processing**, **metadata management**, and **data governance**.

Key topics reflected in the course learning outcomes include:

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Developing data processing pipelines with **PySpark**
- Analyzing data processing performance
- Building real-time data flows with **Kafka, Spark Structured Streaming, and Iceberg**
- Loading **CDC (Change Data Capture)** records into Lakehouse tables using **MERGE**
- Managing data discovery, lineage, profiling, and data quality with **OpenMetadata**
- Applying access control, column masking, and audit policies using **Apache Ranger**

4. dbt, CI/CD, Airflow, and Data Modeling

This section addresses data transformation, modeling, and automation within a Lakehouse environment.

Core topics based on the stated learning outcomes include:

- Developing testable **Silver-to-Gold transformations** with **dbt**
- **Fact** and **Dimension** tables
- **Star Schema** and **Slowly Changing Dimensions (SCD)**
- Managing data pipelines with **Apache Airflow** and **CI/CD** practices
- Designing **end-to-end Lakehouse pipelines** suitable for production environments