Building End to End Solution with Confluent Kafka

Duration : 5 Days(40 hours)

Table of contents :

Day 1: Kafka Configuration and Administration

- Introduction to Kafka:
 - Overview of Kafka
 - Kafka architecture and components
- Kafka Installation and Configuration:
 - o Installing Kafka on-premises
 - Installing Kafka on AWS
 - Configuring Kafka brokers
 - Configuring Zookeeper
- Kafka Administration:
 - Managing Kafka clusters
 - Creating and managing topics
 - Configuring retention policies
 - Setting up access control and security

Labs :

Lab 1.1: Kafka Installation and Configuration

- **Objective**: Install and configure Kafka on-premises and on AWS.
- Steps:
 - 1. Download and install Kafka on a local machine.
 - 2. Configure Kafka brokers and Zookeeper.
 - 3. Set up Kafka on AWS EC2 instances.
 - 4. Verify the installation by starting Kafka and Zookeeper.

Lab 1.2: Kafka Administration

- **Objective**: Manage Kafka clusters and topics.
- Steps:
 - 1. Create and configure Kafka topics.
 - 2. Set retention policies for topics.
 - 3. Explore Kafka command-line tools for administration.

Day 2: Kafka RBAC and High Availability

- Role-Based Access Control (RBAC):
 - Introduction to RBAC
 - Configuring RBAC in Kafka
 - Managing user roles and permissions
- High Availability and Disaster Recovery:
 - Ensuring high availability in Kafka clusters
 - \circ $\;$ Configuring replication and partitioning
 - Implementing disaster recovery strategies
 - Backup and restore procedures

Labs :

Lab 2.1: Configuring RBAC in Kafka

- **Objective**: Implement RBAC in Kafka to manage user roles and permissions.
- Steps:
 - 1. Set up Confluent Metadata Service (MDS) for RBAC.
 - 2. Define roles and permissions.
 - 3. Assign roles to users and verify access control.
 - 4. Test role-based access to Kafka topics and resources.

Lab 2.2: High Availability and Disaster Recovery

- **Objective**: Ensure high availability and implement disaster recovery.
- Steps:
 - 1. Configure replication and partitioning for high availability.
 - 2. Set up Kafka clusters for fault tolerance.
 - 3. Perform a backup of Kafka data.
 - 4. Simulate a failure and execute disaster recovery procedures.

Day 3: Data Ingestion and Streaming Data

- Data Ingestion:
 - Understanding producers and consumers
 - Configuring Kafka producers
 - Configuring Kafka consumers
 - Best practices for data ingestion
- Streaming Data with Kafka:
 - Introduction to Kafka Streams
 - Building streaming applications
 - Real-time data processing
 - Use cases for streaming data

Labs :

Lab 3.1: Configuring Producers and Consumers

- **Objective**: Set up Kafka producers and consumers for data ingestion.
- Steps:
 - 1. Write a Kafka producer to send messages to a topic.
 - 2. Write a Kafka consumer to read messages from a topic.
 - 3. Configure producer and consumer properties for optimization.
 - 4. Test data ingestion and verify message delivery.

Lab 3.2: Building Streaming Applications with Kafka Streams

- **Objective**: Create real-time streaming applications using Kafka Streams.
- Steps:
 - 1. Set up a Kafka Streams application.
 - 2. Process real-time data streams for aggregation and transformation.
 - 3. Implement stateful and stateless operations.
 - 4. Test and validate the streaming application.

Day 4: Data Transformations, ETL, and Grafana Dashboards

- Data Transformations and ETL:
 - Introduction to ETL with Kafka
 - Data transformation techniques
 - Using Kafka Connect for ETL
 - Integrating with external systems
- Grafana Dashboards:
 - Introduction to Grafana
 - Setting up Grafana for Kafka monitoring
 - Creating and configuring dashboards
 - Visualizing Kafka metrics

Labs :

Lab 4.1: Data Transformations and ETL with Kafka Connect

- **Objective**: Perform ETL operations using Kafka Connect.
- Steps:
 - 1. Set up Kafka Connect and configure connectors.
 - 2. Extract data from external sources (e.g., databases).
 - 3. Transform data and load it into Kafka topics.
 - 4. Validate the ETL pipeline and monitor data flow.

Lab 4.2: Creating Grafana Dashboards for Kafka Monitoring

- **Objective**: Monitor Kafka clusters using Grafana.
- Steps:
 - 1. Install and configure Grafana.

- 2. Set up Prometheus as a data source for Grafana.
- 3. Create and customize Grafana dashboards for Kafka metrics.
- 4. Visualize Kafka performance and set up alerts.

Day 5: Advanced Kafka Monitoring and Management

• Advanced Kafka Monitoring:

- Monitoring Kafka clusters
- Using Prometheus with Grafana
- Setting up alerts and notifications
- Troubleshooting common issues
- Kafka Management and Best Practices:
 - Managing Kafka deployments
 - Performance tuning and optimization
 - Best practices for Kafka in production
 - o Case studies and real-world examples

Labs :

Lab 5.1: Advanced Kafka Monitoring with Prometheus and Grafana

- **Objective**: Implement advanced monitoring for Kafka clusters.
- Steps:
 - 1. Integrate Prometheus with Kafka for monitoring.
 - 2. Configure Grafana dashboards for detailed metrics.
 - 3. Set up alerts and notifications for critical events.
 - 4. Perform troubleshooting and resolve common issues.