

**Introduction to Kafka for C# Developers**

**Course Number:** DVOP-168WA
**Duration:** 4 days

**Overview**

This Introduction to Kafka for C# Developers training course teaches attendees how to use the Apache Kafka event streaming platform for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. .NET Core is used as the underlying framework.

**Prerequisites**

* Having a basic understanding of messaging, cloud, development, architecture, and virtualization is beneficial.
* Experience developing .NET applications with C# is required. Prior .NET Core experience is recommended.

**Materials**

All Kafka training students receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will work well.

**Objectives**

* Understand the use of Kafka for high-performance messaging
* Identify the usages for Kafka in Microservices
* Explain the benefits of Kafka patterns
* Differentiate between messaging and message brokers
* Describe Kafka messaging environments
* Develop producers and consumers for Kafka
* Recognize how Kafka enables Cloud-native applications
* Summarize the characteristics and architecture of Kafka
* Demonstrate how to process and consume messages from Kafka using .NET Core Web API, MVC, and Worker (BackgroundService)
* Demonstrate Kafka’s role in the end-to-end project involving .NET MVC frontend, .NET Web API backend, .NET Worker, Azure SQL database, and Redis cache
* Design distributed high throughput systems based on Kafka
* Describe the built-in partitioning, replication, and fault-tolerance of Kafka

**Outline**

* Introduction to Kafka
	+ Messaging Architectures
	+ What is Kafka?
	+ When to Use Kafka?
	+ Kafka Architecture
	+ Core concepts in Kafka
	+ Why Kafka Cluster?
	+ Sample Multi-Broker Cluster
	+ Overview of ZooKeeper
	+ Kafka Cluster & ZooKeeper
	+ Schema Registry
	+ Who Uses Kafka?
* The Inner Workings of Apache Kafka
	+ A Kafka Cluster High-Level Interaction Diagram
	+ Topics & Partitions
	+ The Terms Event/Message/Record
	+ Message Offset
	+ Message Retention Settings
	+ Deleting Messages
	+ The Flush Policies
	+ Writing to Partitions
	+ Batches
	+ Batch Compression
	+ Partitions as a Unit of Parallelism
	+ Message Ordering
	+ Kafka Default Partitioner
	+ The Load Balancing Aspect
	+ Kafka Message Production Schematics
	+ ZooKeeper
	+ Reading from a Topic
	+ Consumer Lag
	+ Consumer Group
	+ Consumer Group Diagram
	+ The Broker
	+ Broker Hardware Consideration
	+ OS and File System
	+ The Leader and Followers Pattern
	+ Partition Replication Diagram
	+ Controlled Shutdown
	+ Controlling Message Durability with Minimum In-Sync Replicas
	+ Log Compaction
	+ Frequent Operational Problems
	+ Some Kafka Design FAQs
* Using Apache Kafka
	+ What is Confluent?
	+ Confluent Cloud
	+ Confluent Cloud Resource Hierarchy
	+ Setting up Confluent Cloud on Azure
	+ Setting up Confluent Cloud using Confluent.io
	+ Select the Confluent Cloud Cluster Type
	+ Choose the Cloud Provider
	+ Setting up Confluent Cloud using Azure Marketplace
	+ Select Confluent Cloud in Azure Marketplace
	+ Purchase Confluent Cloud
	+ The Cluster View
	+ Exploring the Confluent Cloud Console
	+ Topics
	+ Topics Advanced Settings
	+ Searching for Messages in a Topic
	+ The Confluent CLI
	+ The confluent CLI Command Examples
	+ Kafka Cluster Planning – Producer/Consumer Throughput
	+ Managing Topics in Confluent Cloud Console
	+ Editing an Existing Topic
	+ Delete a Topic
	+ Kafka and .NET
	+ .NET Kafka Architectures
	+ Packages
	+ Installing the Packages
	+ Navigating .NET Client Documentation
	+ Important Classes and Interfaces
	+ appsettings.json Kafka Configuration
	+ Loading the Configuration from appsettings.json
	+ Produce and ProduceAsync Methods
	+ Produce vs. ProduceAsync
	+ Error Handling
	+ Consuming Messages
	+ Creating and Deleting Topics
	+ Copying Data from Between Environments
	+ Mocking Datasets using Datagen Connector
	+ Monitoring Confluent Cloud
	+ Monitoring Confluent Cloud using cURL
	+ Motoring Confluent Cloud using third-party Tools
* Building Data Pipelines
	+ Building Data Pipelines
	+ What to Consider When Building Data Pipelines
	+ Timeliness
	+ Reliability
	+ High and Varying Throughput
	+ Evolving Schema
	+ Data Formats
	+ Protobuf (Protocol Buffers) Overview
	+ Avro Overview
	+ Avro Schema Example
	+ JSON Schema Example
	+ Managing Data Evolution Using Schemas
	+ Confluent Schema Registry
	+ Confluent Schema Registry in a Nutshell
	+ Schema Management on Confluent Cloud
	+ Create a Schema using Confluent CLI
	+ Create a Schema from the Web UI
	+ Schema Change and Backward Compatibility
	+ Collaborating over Schema Change
	+ Handling Unreadable Messages
	+ Deleting Data
	+ Segregating Public and Private Topics
	+ Transformations
	+ Security
	+ Failure Handling
	+ Agility and Coupling
	+ Ad-hoc Pipelines
	+ Metadata Loss
	+ Extreme Processing
	+ Kafka Connect vs. Producer and Consumer
* Integrating Kafka with Other Systems
	+ Introduction to Kafka Integration
	+ Kafka Connect
	+ Running Kafka Connect Operating Modes
	+ Key Configurations for Connect workers:
	+ Kafka Connect API
	+ Kafka Connect Example – File Source
	+ Kafka Connect Example – File Sink
* Kafka Security
	+ Kafka Security
	+ Encryption and Authentication using SSL
	+ Configuring Kafka Brokers
	+ Authenticating Using SASL
	+ Authorization and ACLs
	+ Securing a Running Cluster
	+ ZooKeeper Authentication
* Monitoring Kafka
	+ Metrics Basics
	+ JVM Monitoring
	+ Garbage collection
	+ Java OS monitoring
	+ OS Monitoring
	+ Kafka Broker Metrics
	+ Under-Replicated Partitions
	+ Active controller count
	+ Request handler idle ratio
	+ Intelligent Thread Usage
	+ All topics bytes in
	+ All topics bytes out
	+ All topics messages in
	+ Partition count
	+ Leader count
	+ Offline partitions
	+ Request metrics
	+ Logging
	+ Client Monitoring
	+ Producer Metrics
	+ Overall producer metrics
	+ Per-broker and per-topic metrics
	+ Consumer Metrics
	+ Fetch Manager Metrics
	+ Per-broker and per-topic metrics
	+ Consumer coordinator metrics
	+ Quotas
	+ Lag Monitoring
	+ End-to-End Monitoring
* Apache Kafka Best Practices
	+ Partitions
	+ Consumers
	+ Producers
	+ Brokers
* Conclusion