Brought to you by:

CLOUDERA

Edge to AI

for dummies®

A Wiley Brand

- Identify AI edge use cases
- Apply data governance to the entire data lifecycle
- Integrate data in motion architecture

Cloudera® Special Edition

Eric Butow
Diby Malakar

About Cloudera

Cloudera is the only data and AI platform company that large organizations trust to bring AI to their data anywhere it lives. Unlike other providers, Cloudera delivers a consistent cloud experience that converges public clouds, on-prem data centers, and the edge, leveraging a proven open-source foundation. As the pioneer in big data, Cloudera empowers businesses to apply AI and assert control over 100% of their data, in all forms, improving security, governance, and real-time and predictive insights. The world's largest brands across all industries rely on Cloudera to transform decision-making and ultimately boost bottom lines, safeguard against threats, and save lives. Learn more at cloudera.com.

Edge to AI

Cloudera® Special Edition

by Eric Butow
and Diby Malakar

for dummies®
A Wiley Brand

Edge to AI For Dummies®, Cloudera® Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-394-41965-4 (pbk); ISBN 978-1-394-41966-1 (ebk); ISBN 978-1-394-41967-8 (ebk)

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Development Editor/Project Manager: Rebecca Senninger

Acquisitions Editor: Traci Martin

Editorial Manager: Rev Mengle

Business Development Representative: Jeremith Coward

Production Editor: Tamilmani Varadharaj

Cloudera team: Matthew Burgess, Gary Gaur, Kathryn Gdula, Edvin Kőrösi, Mohammed Nahhas

Table of Contents

Introduction

There's a quote, misattributed to Mark Twain, about data: "Data is like garbage. You'd better know what you are going to do with it before you collect it." In this age of AI, getting data in real time to make decisions and provide your internal and external customers with the data they need may be the difference between a thriving business and a dying one.

You're reading this book not only because you think staying in business is a good thing, but because you want your data, no matter if it's from the edge or from your AI systems, to be timely, accurate, and trustworthy.

Cloudera provides your company with a modern data architecture to help your business make better decisions, safeguard against threats, and boost your bottom line. You can learn more at `cloudera.com`.

About This Book

This isn't just a book — it's your field guide to providing trustworthy data using the best technology and practices in the industry. So, grab your favorite beverage, sit in your comfortable chair, and explore how Cloudera helps you bring data from the edge to your AI applications.

Edge to AI For Dummies, Cloudera Special Edition, breaks down key components of managing data with AI in a simple and digestible manner. Topics you'll find in this book include:

- The data in motion architecture
- Collecting and analyzing edge device data
- Integrating data seamlessly so decision makers can act decisively
- Understanding the role of event streaming
- Using data streaming analytics

Foolish Assumptions

In writing this book, I made a few assumptions about you, the reader:

- You're someone who manages data on a daily basis.
- You want to better leverage and use your data to make better business decisions and enhance your business processes.
- You know that you need to manage your data better, but you have a lot of questions you need answered.

Icons Used in This Book

To make things easier and ensure that you don't miss important details, I use icons throughout this book. Here's what the different icons look like and mean:


TIP

The Tip icon is a small piece of advice that may save you time and make your document management journey easier.


REMEMBER

This book covers a lot of details and information, so every now and then you see a Remember icon to remind you of the most important details. When you're reading every juicy detail of the book, the Remember icon just helps resurface some of those tidbits.


DID YOU KNOW

This icon highlights areas where Cloudera can help address your challenges.


WARNING

Yes, this book has a few warnings. When you see a Warning icon, take a few extra moments to understand the effect of what you've read. This information may just save you a headache or two.


TECHNICAL STUFF

This icon points out paragraphs that take a deep technical turn. Skip them if the finer details don't interest you.

IN THIS CHAPTER

- Using three technologies for data in motion
- Real-time streaming with data in motion
- Protecting data with governance

Chapter 1
Introducing Edge to AI

Data is everywhere, but it's useless if your business can't use it to make sound decisions. And businesses are feeling it: According to a 2023 study by the research firm Gartner, 30 percent of global data and analytics decision makers cite a lack of trust in source data. Twenty-seven percent also cite a lack of trust in data analysis as the most common reason decision makers don't act on that analysis.

But that's not all. We live in an age of customers who want immediate gratification, fast-moving external threats, and (of course) AI. Successful companies today can use real-time data to predict, adapt, and act quickly.

For example, factories can use information from edge devices to create AI applications that process equipment sensor data locally. They are now able to detect anomalies and potential failures in real time, which can reduce downtime by not sending all data to a central cloud.

Edge data is also used to enhance AI with financial services. Batch data processing with real-time data stream processing detects fraud at the point of sale (the edge), not in the data center. This approach allows for instant authorization or blocking of transactions.

Sounds good, you say, but how do you get to real-time data management? We'll take the proverbial 30,000-foot view of what data in motion looks like and the technologies that power it.

Presenting Data in Motion

You may have heard of *data in motion* before, and the definition is almost comically plain: The term refers to digital information that actively moves between systems, devices, or locations. In a data in motion architecture, the word "actively" is defined as data that's used in real time from edge devices, such as the point-of-sale system you use when you purchase groceries.


REMEMBER

Motion is also a synonym for flow, and you may have run across the terms *data flow* and *data streaming* when it comes to moving your data.

Think of data flow as being a glass of water. You have a finite amount of data that completely fills a container much as water fills a glass up to the brim. You transfer the entire container from point A to point B. Once you move all the data, you're done, just as when you empty a glass of water.

In contrast, think of data streaming like spraying water with a garden hose. The data flows continuously, possibly forever. You process the data as it arrives piece by piece, and you never have to worry about waiting for the entire stream to end. The process is ongoing and happens in real time.

Getting to Know the Three Key Components

The Cloudera Data in Motion architecture relies on several open source collection and analysis tools from the Apache Software Foundation. They provide cost effective, scalable, and secure solutions.


TECHNICAL STUFF

The Apache Software Foundation is a non-profit corporation that supports over 300 open-source software projects. You can find out more about it at `apache.org`.

Data flows are managed with NiFi and MiNiFi and processed with Kafka and Flink. Let's take a quick tour of all this cool technology.

Apache NiFi

Apache NiFi is a powerful data integration tool that automates, manages, and secures data flows between systems in real time. Five important features come out of the box:

- **A visual UI:** You get a browser-based interface for creating and managing data pipelines. You can also view detailed reports about processed data easily.
- **Control:** You can manage data throughput from the source to the destination manually or automatically.
- **Extensible:** NiFi is built to integrate with diverse systems and protocols.
- **Security:** NiFi features user authentication and secure HTTPS connections over the web.
- **Scalability:** NiFi supports clustering so your business can scale for larger data volumes.

Apache also offers MiNiFi, a lightweight edge agent that collects data and directly integrates with NiFi. The result is that you can gather and manage data from edge devices, databases, and logs and send that data into the second technology in the Apache triad, Apache Kafka.

Apache Kafka

Apache Kafka is a distributed event streaming platform with robust real-time data processing. This platform provides high throughput, low latency, and durability, so it's no surprise that Kafka is ideal for processing heaps of data. You'll find Kafka tracking website activity, monitoring metrics, and integrating data across distributed systems.

Kafka stores streams reliably so you can use it for real-time streaming and batch processing. (Don't fret; we'll explain what batch processing is in the "Bridging Streaming and Batch Data" section later in this chapter.)

Apache Flink

Apache Flink is a distributed processing engine designed for high-performance, stateful computations across data streams. Stateful computations are processes that remember past events and then compute output for current or future events. Flink accurately processes data based on when events occurred, not when the data was received, so Flink has no issues handling late data.

Flink also doesn't have any trouble with accurate results in case of failure, because it comes with a lightweight, distributed checkpointing mechanism out of the box. It's no wonder that companies use Flink for live data monitoring and detecting patterns that expose anomalies including fraud.

Knowing the Stakeholders for Implementing Data in Motion

When you're deciding how to transform and enrich your data, that decision comes down to your environment and your needs.

You may be working in a large enterprise drawing data from many sources, including from other businesses that want to implement real-time agentic AI systems that help their businesses complete complex tasks with little human supervision.

If you're working with teams in a complex and related industry, a high performing data in motion architecture is essential for managing data across hybrid and multi-cloud environments and integrating data from multiple sources securely.

You may be working in an enterprise that's looking to move from your current batch processing to getting real-time information, and your financial people are struggling with costs and total cost of ownership (TCO).

If you've found yourself nodding as you've read one (or more) of these three organizational environment descriptions, now you have to home in on the three types of leaders you need to consider

as you implement a modern dataflow solution. These leaders may just be one person or three people with distinct roles.

Data leader

The data leader is the person who's responsible for meeting business demand to deliver actionable data embedded in your business processes.

AI is a big part of this because data pipelines are changing, and data platforms need to continuously ingest and process unstructured data at a large scale. The challenge for the data leader is to align technology and AI initiatives to business goals while continuing to push for investment in the right people and technology.

The analytics architect

This person makes decisions related to the design and management of real-time data streaming as well as analyzing that data to turn into action.

This architect has a delicate balancing act: Build a system that delivers near-term results and pleases company leaders, adapt to constantly changing technology, and don't compromise long-term plans for the dataflow system and the business.

The technical leader

Your business may have a chief technical officer (CTO) or related title to describe the technical leader in your company. The technical leader is responsible for accurately and thoroughly scoping AI use cases for the dataflow system.

The technical leader also needs a balance of power and practicality so a solution can be implemented successfully within their organization.

That work may include overseeing construction and execution of real-time machine learning data pipelines to enrich your AI systems so they can produce richer data to help your internal team and external customers.

Understanding the Architecture

A data in motion architecture allows data to be used in real time, no matter if that data comes from edge devices, data streams, any existing data stores, applications, and change data capture services.

This architecture combines all three Apache technologies — NiFi (and MiNiFi), Kafka, and Flink — so data can not only flow in real time from where it's created to where it's needed, the data is also processed and analyzed in real time.

Technology is great, but business is about proving value. To do that, here are three compelling arguments for you:

- Real-time capture processing and distribution of data is required for automated action. For example, the system freezes a fraudulent transaction before it's processed or stops a manufacturing process before it creates a defect.
- Any human-led decision will have up-to-the-minute data, which is quite helpful when you're making inventory and supply chain decisions.
- The simplified architecture gives you the most bang for your buck: sustained development speeds and lower operational costs.

Bridging Streaming and Batch Data

Many organizations still regard streaming data in real-time as a pipe dream. Managers see high volumes of data stored across siloed systems using batch-based processing and think there's no way they can move to real time without putting the company at risk.

Batch-based processing collects data in groups and then holds it to process it all at once before forwarding the results to their intended destination. Each component in the batch-based system adds latency, and then you're making decisions based on yesterday's data.

Artificial intelligence has caused data volume to skyrocket and batch processing can't keep up with the load and the pressure from company leaders to get real-time insights and stay ahead of the competition. Real-time streaming is the solution, because it ingests, processes, and analyzes data as it's created.

When you pair the data in motion architecture with a data lakehouse, which combines flexible data storage with data management and analytics, real-time streaming can take vast amounts of data and generate impactful results. It doesn't matter how the data is structured.

Cloudera combines data in motion with an open data lakehouse architecture to ensure a unified data experience for enterprise AI. Learn more in Chapter 6.

Unified, real-time streaming data in a data lakehouse provides significant advantages, including:

- Access to fast, dynamic decision-making
- Better customer experiences
- Improved operational resilience
- Lower total data processing costs
- Reduced risk of fraud and downtime

Combining Applications and Systems

The ability of your company to beat your competition could hinge on combining your data lakehouse(s) with streaming in a unified system. This powerful combination helps your business effectively process, store, and analyze not only real-time streaming data, but your historical batch data.

To do that, your company needs to get hip: Implement a Hybrid Integration Platform (HIP), that is. A HIP is an advanced software platform that provides seamless integration between different applications, systems, and data sources, regardless of whether they are on premises, in the private cloud, in the public cloud, or all three.

Here are the three HIP components you need to know about:

- Hybrid cloud infrastructure that includes the mix of your on-premises data centers, your private clouds, and/or public clouds such as Amazon Web Services and Google Cloud
- Data fabric/data mesh frameworks that provide unified data management across diverse data storage locations
- API management, which enables communication between cloud-based apps and on-premises data storage systems

What's it mean for your business? You can balance the security of your local legacy data systems with the scalability and agility of modern data solutions and not lose access to any of your data. Whether your customers think you're more hip as a result is up to your marketing department.

Integrating Apache Iceberg

I've talked about the three Apache tools that make data in motion work, but there's another important tool in Apache's toolbox that's also vital to analytics: Iceberg.

Iceberg is an open-source table format that's designed for large-scale analytics. Because all of Apache's technologies are open source, Iceberg is a favorite of organizations because their data isn't locked into one vendor, so the data in Iceberg is fully owned by the business. Any Iceberg-compatible analytics solution the business uses can access the data.

What this means for your business is that everyone who works with your data can work with the same data sets, avoid duplication, leverage Iceberg's strong data governance, and process data faster. To learn more about *Apache Iceberg, see Migrating to Apache Iceberg For Dummies, Cloudera Special Edition*.

Establishing Security and Governance

No matter if you're using batch processing or real-time data streaming, these systems won't do much good if you're not secure. And one of the biggest issues with batch processing is a silo mentality in departments.

Departments and business unit silos guard their data closely and don't like to share with other departments. They use different data movement solutions only for their data flows, use separate user authentication features, and consider even the mention of data governance programs that could break down silos is heresy.

WARNING

Worse, some data silos are stored by individual users in cloud-based storage like Google Drive, often on mobile devices. These create situations where data is completely unsecure.

If you're trying to convince company leaders to move to a modern data solution, here are the three benefits to share:

REMEMBER

- **Security:** You reduce the risk of a breach that could damage your business and lose customer trust.

  If you have a cybersecurity person or team in your business, be sure to get them on board.
- **Proper control:** You have clear policies and procedures to manage data so that everyone knows how to share your data responsibly.
- **Better decisions:** Sharing data throughout the organization in real time means you can process data more quickly, make decisions faster, and stay at least one step ahead of your competition.

DID YOU KNOW

The most reliable way to ensure security, control, and reliable data for decision making is to have unified security and governance across your data lifecycle, like Cloudera Shared Data Experience (SDX). Learn more about Cloudera SDX in Chapter 6.

IN THIS CHAPTER

- Using Apache NiFi and MiNiFi
- Using a unified system
- Managing data at the edge

Chapter 2
Collecting and Analyzing Edge Device Data

The term "edge" can mean something that's cool and state of the art, but when it comes to managing data, the term *edge* refers to the source of data creation in your network, such as the Internet of Things. (That's still cool.) There are plenty of use cases for this type of data:

- Data from manufacturing robots, such as vibration and temperature, to predict asset failure and optimize production line efficiency
- Self-driving vehicle data, including speed, braking patterns, and fuel consumption
- Internet of Medical Things (IoMT) data from devices such as bedside monitors, glucose monitors, and digital blood pressure cuffs that transmit usage logs to doctors and adjust medication schedules

When network and data administrators want to collect data from their users at the edge, they rely on Apache NiFi and MiNiFi.

Understanding MiNiFi

Apache NiFi is a web-based data integration tool so you can design, control, and monitor data in real time, and I'm going to go into more details in Chapter 3. For this chapter, I want to focus on the MiNiFi sub-project because that's the tool that provides more pre-processing at the edge. Not only can you analyze local data at the edge for an immediate response, but you also get a global view of the data stream.

MiNiFi (and NiFi, too) works with Kafka and Flink by offering greater optimization and lower latency right up to your users' edge devices. (You'll learn about Kafka in Chapter 4 and Flink in Chapter 5.)

MiNiFi is available as a native binary for macOS, Linux, or Windows (built on the C++ programming language) or as a platform-independent Java distribution.

MiNiFi Java

This is a headless version of NiFi, which means it operates without a graphical user interface. This version is recommended where the 500+ components of NiFi can be leveraged for advanced data processing.

You can use MiNiFi Java with all components of NiFi. The Java flavor of MiNiFi supports site-to-site to exchange data with NiFi and also supports Kerberos authentication.

Here's what you and your team need to know:

- Requires Java (JRE) to be installed on the host.
- Supports Java 8 and Java 11, and Java 21 (the version based on NiFi 2).
- The binary file is 235+ MB.
- The minimum memory heap size required is 256MB, but the total amount will depend on the flow being executed.
- The CPU consumption depends on configuration and the use case.

The ARM version is also available in the container image.

MiNiFi C++

The C++ version of MiNiFi is best suited for collecting logs of applications running Kubernetes sidecar pod or daemonset deployments. This version is recommended when a smaller footprint is required when you're running MiNiFi on laptops or Kubernetes environments. The binary file is around 30 MB in size, and the required memory is usually around 50 MB.

Your actual memory requirements may be higher depending on your use case.

Collecting IoT Device Data

When you want to collect data from IoT devices, here's a primer with the five steps:

1. Devices such as trucks, motion detectors, and temperature detectors feed data into edge processing with MiNiFi.
2. NiFi and edge flow management ingest the data for processing.
3. Apache Kafka is the hub for brokering messages.
4. NiFi and Kafka provide real-time event processing.
5. Apache Flink manages streaming analytics.
6. Store data in a data lakehouse that users and applications can access.

Checking Out AI Action at the Edge

If your business leaders are brooding about AI and how the company can implement that technology like, *yesterday*, you need to remind them that you actually want to have successful AI deployments. So, take a deep breath and let's review what a robust data management infrastructure looks like.

Successful artificial intelligence deployments depend on a robust data management infrastructure that supports machine learning, data science, and the ability to manage the data lifecycle.

You may hear that your AI efforts need the best and most sophisticated algorithms to stave off the competition. Though that indicates that your fellow employees (and hopefully your leaders) understand the need to invest in the technologies, people, and processes to support providing good data to your AI systems.

REMEMBER

The data science piece is just a small part of what you need to build a functional and useful AI capability for your business. Here's what else you need:

- A data management infrastructure that can meet the challenge, which is what we cover in Chapter 1.
- Efficient MLOps (Machine Learning Operations), which is a set of practices that automates and streamlines the lifecycle of machine learning models from development to production. MLOps enables data engineers to collaboratively build, test, deploy, and monitor models. The end result is improved speed, accuracy, and reliability.
- A single, secure management hub to design, deploy, and monitor data flows across thousands of MiNiFi agents.

DID YOU KNOW

Cloudera Edge Management allows you to manage the entire edge flow lifecycle including authorship, deployment, and monitoring from a single hub. Learn more in Chapter 6.

In sum, you need to be able to manage data flow from the edge to the application layer, which in turn is processed into insight and action. If you want your company to implement AI successfully, then you'll need to do that with the least number of steps, the greatest data context for your AI system, consistency, and (let's not forget) security.

Challenges with MiNiFi

While MiNiFi is an incredibly powerful tool for pushing data routing and pre-processing right to the absolute edge of your network, it isn't a magic wand. Deploying and managing software across hundreds or thousands of physical devices, remote servers, or Kubernetes pods comes with a unique set of real-world headaches.

When you are planning your edge architecture, you and your team need to prepare for these challenges:

- **The management problem:** It's easy to manage one or two Apache NiFi data flows using its beautiful web-based graphical user interface. But remember: MiNiFi is *headless*. It does not have a UI. If you have 5,000 remote MiNiFi agents running on delivery trucks or factory floors, you cannot manually log into 5,000 devices to update a data flow or change a configuration file. Without a centralized management hub, controlling agent configurations at scale becomes an operational nightmare.
- **Security vulnerabilities in the wild:** Physical security is a major concern at the edge. A server in a secure cloud data center is safe from tampering; a smart device attached to a traffic light is not. Because MiNiFi agents handle sensitive data ingestion, ensuring secure device authentication, encrypting data at rest on the device, and managing secure updates over-the-air (OTA) is a highly complex necessity.
- **Debugging and monitoring blind spots:** When a data pipeline breaks in the cloud, logs are easily accessible. When a MiNiFi agent on a remote medical device fails, finding out *why* can be incredibly difficult. Gathering remote logs, monitoring agent health, and diagnosing local hardware failures without overwhelming the limited network bandwidth requires a sophisticated, intentional observability strategy.

Cloudera is the *only* solution to provide consistent edge to AI support with security and governance throughout the entire data lifecycle. Cloudera Edge Management provides a management hub for developing and monitoring edge flows. It connects seamlessly with Cloudera AI, so you can enhance AI applications with real-time data using advanced functionality and accessible low-code tools. To learn more, see Chapter 6.

IN THIS CHAPTER

- Modernizing ETL workflows
- Learning the benefits of Apache NiFi
- Managing data

Chapter 3
Integrating Data Seamlessly for Action

If you don't have a system to process data effectively, then you and your company leaders can't take any action. If your company has AI systems to help internal and external users, then trustworthy data is vital to keeping the customers buying and the business growing.

In this chapter, I discuss ETL (Extract, Transform, and Load) workflows and why they're important. Modern data in motion systems use ELT (Extract, Load, Transform) workflows and use no code/low code options, and I'll talk about that approach and why Apache NiFi is a compelling data pipeline solution.

Realizing the Importance of Clean and Reliable Data with ETL Workflows

Managing data pipelines isn't for the faint of heart, because the job comes with plenty of challenges.

- **Systems evolve differently:** Dataflow connects a system of widely distributed components that don't work well together if they ever do at all. And the fun doesn't stop there, because protocols and formats in one component can change at any time.

- **Things change fast:** Business priorities are always changing, and you need to enable new flows and change existing ones fast.
- **Compliance and security:** Speaking of business stuff that changes, you need to make sure your dataflow is not only secure, but also keeps up with laws, regulations, and updated business agreements.
- **Too much data:** A data source can overtake some part of the processing or delivery chain.
- **Too much of the wrong data:** It's inevitable that your dataflow system will get data that's too big, too small, too fast, too slow, corrupt, or just plain wrong in some way (like being in the wrong format).
- **Systems break down:** Networks fail. Storage fails. Software crashes. And there's good old user error.

So, any ETL data pipeline system needs to have clean and reliable data processing that includes analyzing the source data for quality issues, standardizing data formats, handling missing data, removing duplicate records, and testing for anomalies.


REMEMBER

Traditional ETL dataflows were created for batch processing. Once the system captures enough data in a batch for processing, it's delivered into a data warehouse and appears in a report. Precious time has passed between ingestion of an event and the report company leaders can act on. By then, your competition may have beaten you to a new market or introduced a new product that leaves you in the dust.

Understanding Apache NiFi Fundamentals

Apache NiFi is built for real-time data streaming so you can get data you need much faster than traditional ETL.


REMEMBER

NiFi has no per-record or connector fees. That lowers the cost of any data management solution you choose. It also delivers over 480 connectors out of the box for flexibility and extensibility, including the ability to build custom processors.

Guaranteed delivery

NiFi guarantees data delivery even at a very high scale and does this through the use of a purpose-built, persistent write-ahead log and content repository. These two components are designed to allow for very high transaction rates, effective load-spreading, and copy-on-write. NiFi also plays to the strengths of traditional disk read/writes.

Visual command and control

NiFi shows you the visual representation of real-time data pipelines that not only make complex flows less so but also identify areas that need to be simplified. It's easy to design your data pipeline with NiFi's visual user interface so you can drag and drop components instead of writing code.

When you make changes, they're isolated to the affected components, and when you make a change to that component, it immediately appears. You don't need to stop an entire flow or set of flows to make a specific modification.

A repository as a rolling buffer

NiFi acts as a rolling buffer of history. That is, data is removed only as it ages off the content repository or as space is required. NiFi data provenance tracks the documented history of a data object.

Combined, NiFi enables click-to-content, download of content, and content replay at a specific point in a data object's lifecycle.

System and user security

A dataflow is only as good as it is secure between systems as well as between systems and users. NiFi enables two-way encryption and authentication protocols for both systems and users, but there's more to security than just protocols.

For systems, NiFi enables system senders and receivers to encrypt and decrypt content and uses shared keys or other mechanisms.

NiFi also provides pluggable authorization modules so that the system controls a user's access at particular levels, such as read-only, dataflow manager, and admin. If a user enters a sensitive authentication property like a password, NiFi immediately encrypts it on the server side, and it's never exposed on the client side even in encrypted form.

Extensible architecture

Data pipelines need to be adaptable to its clients' needs, and NiFi is built for extension. Points of extension include processors, controller services, reporting tasks, prioritizers, and customer user interfaces. If you can't find the right processor or service in NiFi's rich library of 480+, you can build your custom processors using Java or Python.

NiFi's extensible architecture ensures that you can consume any data, from any source and write it to any destination, eliminating the need to acquire specialized, proprietary, and expensive solutions just for one use case.

Scale up, scale down

As part of NiFi's flexibility, you can increase the number of the concurrent tasks on the processor within the user interface's Scheduling tab when you configure your dataflow system.

This feature gives you the ability to set your flow to fit your needs. If you need to execute more processes simultaneously, you can turn up the flow. If you have edge devices where a small footprint is needed due to limited hardware, you can scale it down.

Automating Workflows

The goal of a modern data in motion system is to automate your workflow so that you spend less time managing and more time analyzing your data. One of NiFi's strengths is that it can move any data from any source to any destination. This feature dramatically reduces total cost of ownership (TCO) by eliminating the need for multiple and/or specialized ETL/ELT tools for unique sources and syncs. There are four steps in a data in motion workflow:

1. **Ingest data:** Pull data from sources like databases or IoT sensors.
2. **Transform:** Convert formats (such as JSON to SQL) or enriching data in real-time.
3. **Route:** Dynamically route data to different destinations based on content or metadata attributes.

4. **Deliver:** Send processed data to final endpoints like data lakes or messaging platforms.

NiFi uses six different methods for managing dataflows, starting with the user interface:

- **Visual design:** Users drag and drop components onto a canvas in the NiFi web interface to build automated data pipelines without writing code.
- **Scheduling:** Processors can be configured to run continuously (which are timer-driven) or at specific times using CRON-driven scheduling.
- **Git-based flow registry clients:** Allows you to connect NiFi directly to version control providers like GitHub or GitLab, as well as to Bitbucket and Azure DevOps.
- **REST API and CLI:** The NiFi REST API allows for programmatic control and automation of flow deployments; the NiFi Toolkit CLI supports scripting common tasks.
- **External tools:** Platforms such as Cloudera Data Flow or specialized tools like Data Flow Manager (DFM) allow for centralized management and automated deployment for multi-cluster environments.
- **Error handling and monitoring:** NiFi provides real-time monitoring through a "bulletin board" and visual status indicators in the interface. Detailed data provenance records every event that enables auditing and failure tracking.

Challenges with Apache NiFi

Despite its unparalleled capabilities in visual data routing, Apache NiFi often struggles to achieve enterprise adoption because its operational complexity scales much faster than its business value. While developers love the initial agility of the drag-and-drop canvas, infrastructure teams quickly find themselves overwhelmed by the architectural realities of managing a distributed, stateful, and JVM-dependent system at scale.

Some of the key challenges include:

- **Stateful architecture and elastic scaling:** Production requires high-availability clustering, but NiFi's stateful

architecture resists modern cloud-native autoscaling. Adding or removing nodes requires complex ZooKeeper coordination and manual rebalancing of in-flight data, making dynamic scaling difficult and error-prone.

- **Distributed observability and debugging:** As flows scale across hundreds of processors and multiple nodes, tracing failures becomes fragmented. Debugging requires navigating disparate logs and node-specific metrics, and maintaining full Data Provenance at scale consumes massive disk and compute resources.
- **Multi-tenancy and canvas governance:** When multiple teams share a single cluster without strict governance, the UI quickly degrades into a cluttered "spaghetti" canvas. This multi-tenant approach creates "noisy neighbor" resource contention, risking one team's inefficient flow impacting another's production pipeline.
- **Disaster recovery and high availability:** Because NiFi holds state and in-flight data on local disks, replicating that data across data centers or cloud regions is not out-of-the-box. Achieving strict RPO/RTO compliance requires engineering complex active-active architectures and external load balancers.

Collectively, these structural and operational challenges transform Apache NiFi from an agile data integration engine into a high-maintenance infrastructure bottleneck at the enterprise level.

DID YOU KNOW

Cloudera Data Flow can help! Cloudera Data Flow directly eliminates these enterprise roadblocks by transforming Apache NiFi into a fully managed, cloud-native service that abstracts away complex infrastructure management and JVM tuning. By providing isolated Kubernetes-based deployments, centralized Flow Catalog versioning, and intelligent auto-scaling, Cloudera Data Flow ensures strict multi-tenant governance while seamlessly automating CI/CD pipelines without the "noisy neighbor" risk.

Ultimately, with highly available and fault-tolerant deployments, resource isolation, automated scaling, granular security controls, and single-pane-of-glass observability, Cloudera Data Flow empowers engineering teams to focus entirely on building high-value data pipelines rather than wrestling with systems administration.

IN THIS CHAPTER

- Data processing with Apache Kafka
- The benefits of Apache Kafka
- Using schemas for event streams

Chapter 4
Event Streaming Data Pipelines

Data is like the new car that you drive off the lot. As soon as you leave the dealership, the car loses value. At the moment data is created, it may be quite accurate, but over time becomes less relevant. And with data coming at high speed, such as *clickstreams* (the recorded, sequential path a user takes when navigating websites or apps), it's irrelevant before you know it.

That's precisely the problem with batch-based data processing. That system collects data in groups and then holds it until there's enough data to process and move the results to their intended destination. Each component in the system adds latency, and by the time you get data you can act upon, the data is already obsolete. When you're acting on obsolete data and your competitor acts on real-time data, who do you think will come up clutch?

You need to up your game with a modern data management solution with real-time streaming. The Apache tool for streaming is Kafka.

Understanding the Role of Event Streaming in an End-to-End Solution

Event streaming is the digital equivalent of the human body's central nervous system. Streaming is the technological foundation for our "always on" world where businesses are more software defined and automated with each passing month. In their quest to get the most up-to-date data possible, the dragon technical and data leaders constantly battle against is *latency*, which is the measurement of delay in a system.

There are multiple sources of latency between a business event and the action taken against that data:

- **Data capture:** There's always latency between reading data from a device at the edge, such as a sensor, and the capture of the raw data.
- **Processing:** Once data is ingested, it's processed. Data processing is the cleaning, integration, and aggregation of raw data to produce contextual information that's useful not only for business processes, but for making decisions. The source of latency varies wildly from one use case to the next.
- **Humans:** There are some use cases where decisions can be made automatically, but many others need one or more humans in the loop. What's more, if data processing is disjointed or incomplete, then the decision maker must try to interpret the data, which can lead to delays or just no decision at all.

Companies that realize they're fighting a latency battle with the competition have incorporated the Apache Kafka distributed event streaming platform.

Knowing the Apache Kafka Fundamentals

Businesses use Kafka to build real-time data pipelines and streaming applications as part of a broader modern data architecture. Kafka is designed to handle large volumes of data in a

scalable and fault-tolerant manner. This approach gives your business three key benefits:

- Continuous data ingestion and processing
- Low latency (milliseconds to a few seconds depending on the use case)
- Event-by-event processing with windowed computation

You use Kafka and data streaming when latency matters, including alerts, monitoring, personalization, and transactions. Kafka is perfect for such tasks as:

- Mission-critical, event-driven architectures
- Real-time context engine for AI
- Financial transactions, such as Robinhood or Stripe
- High-volume log aggregation
- Real-time analytics

REMEMBER

Kafka acts as the digital backbone of a modern business. It breaks down the data silos that keep departments isolated. Instead of every application talking to every other application in a messy web of connections, they all connect to Kafka. This allows your business to scale and innovate instantly. When you add a new service or AI tool, it simply plugs in to the existing stream.

Four building blocks make up Kafka's data processing:

- **Producers:** These applications send events to Kafka topics, such as when a user places an order on an e-commerce website.
- **Topics:** A topic is a named channel where events are published and consumed. These events can be partitioned, such as when a producer decides which partition to send data to via a partitioner or key (such as for orders from different regions) that Kafka places into the proper region partition.
- **Brokers:** A broker is a server that forms the storage layer of a Kafka cluster. A broker receives messages from producers, commits them to disk, and serves them to consumers.

» **Consumers:** These applications read and process data from Kafka topics. For example, a notification service consumes messages from an ordering system that triggers order confirmation emails and/or text messages.


DID YOU KNOW

Cloudera Streams Messaging provides a robust, scalable, and secure messaging backbone for your real-time applications with Apache Kafka. Enterprises choose it as a reliable data transport layer for the most demanding Kafka development workloads. Learn more in Chapter 6.

Designing Event Schemas

Schemas are essential for making event streams consistent and reliable, because they provide structure and definition for the data communicated by an event.

Explicit schemas are essential for event streams in constructing relational database tables, querying API on top of a set of data, and defining the structure of data in an event.


TECHNICAL STUFF

While your compilation options depend on both the schema IDL (Interface Description Language) and the programming language you're using, you can compile the schema into a class or object suitable for your language. Apache Avro, Google's Protobuf, and JSON Schema are three common IDLs that provide structure, format, and documentation for data events. Compiled languages get the benefit of compile-time type checks, which significantly reduces errors in data creation and usage.

Producers rely on the schema registry (which is a separate component and not part of Kafka itself) during data serialization to ensure that the schema they're using matches the expected schema for the Kafka topic. If the schema disagrees with the data being serialized, Kafka will throw an exception and prevent any malformed data from being written into the Kafka topic. Consumers rely on the schema registry to deserialize the schema's binary data back into a usable format.

This structure ensures that the producer and consumer of an event have the same understanding, which saves the consumer the time and stress of understanding the data report.

Incorporating Quality and Reliability

Apache Kafka is widely regarded as a "gold standard" for data streaming because it's reliable, high-throughput, and fault-tolerant.

- **Data durability and persistence:** Kafka stores data on disk and replicates it across multiple brokers (usually a factor of three), which allows Kafka to tolerate failures.
- **Scalability:** The partitioned log model enables horizontal scaling, so Kafka can handle millions of messages per second and petabytes of data.
- **High availability:** Kafka can extend clusters across availability zones (AZs) or geographic regions, which ensures continuous operation.
- **Exactly-Once Semantics (EOS):** With the proper configuration, Kafka guarantees that messages are written and processed exactly once, which prevents duplicates during failures. EOS is very useful for financial transactions where multiple reads can lead to repetitive transactions.
- **Order guarantees:** Apache guarantees that Kafka processes messages within a single partition in the order they were written. A key use case is with ecommerce order processing, where state changes must follow a logical sequence, such as creating an order, processing a payment, shipping an order, and delivering an order.

In sum, Kafka's core reliability stems from its design as a persistent, distributed commit log, which ensures that data isn't lost even if other components in the data management platform fail.

Challenges with Apache Kafka

Distributed data streaming with Apache Kafka is undeniably powerful, but managing the underlying infrastructure presents a steep learning curve. Without third-party tools, organizations frequently collide with three critical operational roadblocks:

- **Missing visual controls:** Out of the box, Kafka operates entirely via a command-line interface and is completely

invisible. When a data pipeline slows down, engineers are left in the dark because there is no native, visual dashboard to track message flow, find bottlenecks, or monitor data health.

- **Manual cluster balancing:** As data volumes fluctuate, some Kafka servers (brokers) get overloaded while others sit idle. Administrators must manually rebalance data across the cluster — a risky, complex process. Additionally, connecting legacy systems without a dedicated way to manage data contracts often requires writing fragile, custom code.
- **Complex disaster recovery:** Open-source Kafka does not naturally mirror data across different regions or hybrid clouds out of the box. If a cloud data center goes offline, ensuring your streams instantly fail over to a backup cluster without losing messages or scrambling their order is an incredibly complex engineering feat.

DID YOU KNOW

Cloudera Streams Messaging can help! Features like Streams Messaging Manager and the Cloudera Surveyor provide a complete visual control room to track every message, map dependencies, and pinpoint latency instantly. The Cruise Control capability automates cluster balancing so servers never overheat, while Schema Registry and Kafka Connect handle data contracts and system integrations seamlessly without code. Additionally, Streams Replication Manager simplifies cross-cluster replication, ensuring bulletproof disaster recovery and keeping data synchronized across your entire hybrid cloud infrastructure.

In short, while raw Kafka provides a powerful engine, conquering its visibility, balancing, and disaster recovery hurdles requires either immense engineering overhead or an enterprise solution like Cloudera Streams Messaging to automate the chaos. Learn more about Cloudera's solution in Chapter 6.

IN THIS CHAPTER

- The benefits of Apache Flink
- Understanding Apache Flink fundamentals
- Using Flink in edge-to-AI workflow

Chapter 5
Streaming Analytics

You're probably reading this book because you want your business to use real-time analytics as a competitive advantage. When you pair data streaming with real-time analytics, you can better serve your customers, detect and react to operational problems, and respond decisively to your competitors. (There's AI in the mix, too.)

You can only get real-time data streaming analytics from a modern data management platform, but many companies (maybe even yours) are using analytical systems that were built for different purposes.

Switching to a data streaming platform isn't enough, though. When you move streaming data through a traditional analytics platform adds a tremendous amount of processing time. By the time you get your data so you can act upon it, that data is ice cold and you may have lost business.

Earlier in this book, I've talked about Apache technologies and how they're industry standards. This chapter is no different, and I introduce you to the Apache Flink stream processing platform.

Introducing Apache Flink

No matter how fast the data moves, the performance of the entire data management system determines the speed at which a business can act. To make timely business decisions, you can't bring the data to the processing, you need to bring the processing to the data.

Enter Apache Flink, which is a distributed stream processing engine designed for stateful computations over unbounded and bounded data streams.

In stateful operations, the system must remember past events to compute the current result. For example, a bank blocks a credit card because transactions in a one-hour period exceed $2,000 and the system remembers previous transactions during that hour.

REMEMBER

Bounded data streams have a defined start and end, which is suitable for batch processing. Once an unbounded stream starts, the data flow continues without end.

Flink works together with Apache Kafka (which Chapter 4 covers) as a dynamic duo in a modern data management system. Kafka provides the durable data transport and storage, and Flink provides the robust real-time analytics. What kind of analytics, you ask? Here are a few common use cases:

- **Fraud detection:** Flink analyzes transactions in real time, and when it comes to fraud detection, the difference between finding an anomaly within 50ms and 2000ms is the difference between blocking a transaction and losing money.
- **Personalization:** Flink processes user behavior streams such as clicks and views in milliseconds, which allows for instant personalization and real-time recommendations.
- **Predictive maintenance:** Flink can analyze sensor data for immediate alerts.
- **Generative AI chatbots:** Flink provides up-to-date context for users to get accurate information to act upon.

DID YOU KNOW

Cloudera Streaming Analytics powered by Flink offers a framework and built-in connectors for real-time stream processing and streaming analytics. Combined with Cloudera Streams Messaging, Cloudera gives you the dynamic duo of Kafka and Flink in one place, accelerating AI development and time-to-insight. Learn more in Chapter 6.

Understanding Flink Fundamentals

Stream processing use cases have four criteria:

- The urgent need to react and respond to data as it decays rapidly and the window of opportunity to act diminishes.
- Relevant data is dispersed into different streams, such as transaction streams and customer tables.
- Apps involved in stream processing must continually evolve to embed new insights into business processes, such as new environments, formats, or consumers.
- Analytical insights are required because raw data doesn't contain all necessary information, such as an anomaly in banking activity that could signal fraud.

It's the fourth criteria that Flink addresses with its analytical superpowers.

Stateful processing

Flink remembers past events. It doesn't just see "Input A," it sees "Input A, which follows Input B from 5 minutes ago, which follows Input C from 15 minutes ago."

Flink's memory enables complex logic you can give to it that triggers alerts, such as alerting an app (and the user) stating that the user has failed to log in three times in one minute and performing an action, such as informing the user to try again in five minutes.

Event time

Flink processes data based on when the event happened, not when it arrived at the server. This solves the "late data" problem.

For example, if an employee needs to send data remotely from a client site but the mobile phone loses signal, that employee has to wait for an hour to drive to a location with a strong signal. Once Flink acquires that data, Flink processes the event that happened one hour earlier.

Exactly-once semantics

I talked about how Flink offers exactly-once processing for AI earlier in this chapter. Apache guarantees that Flink processes every event exactly one time. This means there aren't any duplicates and no data loss. This is crucial for financial and audit use cases.

Sound architecture

Flink maintains memory, or "state," locally in memory or on disk for extremely high performance, as opposed to slow remote database lookups with obsolete analysis solutions. Flink also guarantees reliability by periodically writing consistent checkpoints to durable remote storage such as HDFS (Hadoop Distributed File System).

Using Real-Time Feature Engineering

Flink often serves as the "upstream" processing layer in a larger data streaming pipeline, and you can create real-time features (aggregations, windows) for recommendation systems or predictive maintenance. Even better, the Flink ML library with 33 built-in algorithms specifically for feature engineering, including:

- **Bucketizer,** which maps continuous features into feature buckets
- **FeatureHasher,** which hashes categorical or numerical features into sparse vectors of a fixed dimension
- **One-Hot Encoding,** which converts categorical data into binary vectors
- **StandardScaler,** which standardizes features by removing the mean and scaling to unit variance
- **Vector Assembler,** which combines multiple columns into a single vector column

Connecting the End-to-End Pipeline for Edge-to-AI Use Cases

Flink is the tool to bridge your distributed edge devices such as IoT devices, mobile devices, and sensors to centralized AI models.

A standard Flink edge-to-AI pipeline typically follows a "shift left" architecture, where data is processed and enriched at the streaming layer before it ever reaches a database.

- **Edge ingestion:** Flink works with its sister Apache tools, NiFi and Kafka, to collect raw sensor readings, logs, or user events from edge devices and then transmit them to a centralized broker.
- **Real-time processing:** Flink performs "shift left" operations at the streaming layer, such as cleaning, deduplication, and feature extraction. In this layer, Flink can join streaming data with historical context to create high-quality data products.
- **AI inference and action:** Flink integrates with AI models in two ways:
  - **In-stream inference:** Calling remote model endpoints such as OpenAI, Amazon Web Services (AWS) SageMaker, or Azure ML using user-defined functions (UDFs).
  - **Agentic AI:** Companies use Flink agents framework to build autonomous agents that react to live events rather than just user prompts.

DID YOU KNOW

Cloudera is the only vendor in the market that offers a cohesive edge-to-AI solution that incorporates Flink as well as MiNiFi, NiFi, and Kafka under one unified governance and security system. That's why enterprises in regulated industries need Cloudera Data in Motion's Edge-to-AI solution now more than ever. Learn more in Chapter 6.

Challenges with Apache Flink

While Apache Flink provides unparalleled capabilities for real-time analytics and AI integration, deploying and maintaining a distributed, stateful stream processing engine is not without its

hurdles. To successfully leverage Flink, organizations must navigate several technical and operational complexities.

Some of the key challenges include:

- **Operation and management complexity:** Managing a Flink cluster requires specialized knowledge of its architecture, including JobManagers, TaskManagers, and slots. This may include tools for resource tuning, and infrastructure overhead.
- **State management and large-scale tuning:** Managing stateful computations at scale is highly challenging. When state exceeds available memory, tuning for optimal performance requires deep expertise in storage I/O and serialization.
- **The cost of checkpointing and alignment:** To guarantee exactly-once processing, Flink relies on periodic checkpoints, which can cause temporary spikes in latency. Frequent checkpoints of massive states can saturate network bandwidth and storage infrastructure, creating backpressure that slows down the entire real-time pipeline.

Cloudera Streaming Analytics can help! Cloudera Streaming Analytics offers a framework for real-time stream processing and streaming analytics that reduces management complexity and costs. Access built-in connectors of runtime components, cluster and service management, all under a single unified security and governance component. Learn more about Cloudera's unique solution in Chapter 6.

IN THIS CHAPTER

- The Cloudera Data in Motion architecture
- The Cloudera Shared Data Experience (SDX)

Chapter 6
Almost Ten Key Takeaways

This chapter is all about the ten things you should learn about Cloudera's data in motion architecture built upon the technologies and concepts in this book.

Understand Data in Motion Principles

Data in motion adheres to the following list of principles to ensure real-time data is consistently available to systems that need it:

- **Low latency:** Data must be processed and delivered with minimal delay.
- **High throughput:** Systems must be able to handle large volumes of data efficiently.
- **Reliability:** Data must be delivered accurately, completely, and consistently without errors or loss.
- **Fault tolerance:** Systems must be able to handle failures and recover quickly.
- **Scalability:** The infrastructure must be able to handle increasing data volumes and processing demands as business needs grow.

- **Security:** Data must be protected from unauthorized access and tampering.
- **Data governance:** Clear policies and procedures must be in place for data management.

Get to Know Cloudera Data in Motion and Reference Architectures

Cloudera offers several core data in motion services to accelerate real-time AI development: Cloudera Edge Management, Cloudera Data Flow, Cloudera Streams Messaging, and Cloudera Streaming Analytics.

Cloudera Edge Management is powered by Apache MiNiFi, which you learn about in Chapter 2. Cloudera Edge Management enhances AI and other data applications with real-time data using advanced functionality and accessible low-code tools.

Cloudera Data Flow is powered by Apache NiFi (Chapter 3 is all about NiFi). Cloudera Data Flow enables developers to connect to any data source anywhere with any structure, process it, and deliver to any destination using a low-code authoring experience.

Cloudera Streams Messaging uses Apache Kafka, which Chapter 4 covers. Cloudera Streams Messaging acts as a durable layer for collecting data from different sources and manages that data in secure and scalable ways.

Cloudera Streaming Analytics is powered by Apache Flink (see Chapters 4 and 5 to find out more about Flink). Cloudera Streaming Analytics provides real-time stream processing and analytics with low latency.

Cloudera is the only vendor to combine all of these technologies in one place to deliver enterprise grade Edge to AI solutions. You can take data from anywhere and deliver it anywhere, whether that means on-premises or in cloud deployments or both. This, coupled with end-to-end security and governance, makes Cloudera stand out to enterprises across industries.

Leverage Data at the Edge with Cloudera Edge Management

Cloudera Edge Management allows you to ingest, capture, and deliver data in real time from any streaming source — including clickstreams, social media, mobile, and IoT devices — to build AI applications and scale operations.

You command and control your devices with the Edge Management hub, the single management layer for all edge agents. Hundreds of prebuilt processors enable easy connection with a range of data sources, devices, and protocols. An intuitive, low-code, drag-and-drop UI lets you build sophisticated data flow pipelines with ease.

Edge Flow Designer handles any throughput at scale, moving petabytes of data in just a few hours from data center to cloud or vice versa. Enable a multi-cloud model with a cloud-vendor-agnostic approach to managing data. Cloudera Edge Management brings it all together in a central location that allows you to design dataflows to be executed by the agents, and to monitor and control the agents.

Achieve Universal Data Distribution at Scale with Cloudera Data Flow

Cloudera Data Flow facilitates universal data distribution by streamlining the end-to-end process of data movement. This enables enterprises to seamlessly move any data from any source to any destination across data centers and cloud deployments, allowing them to take on IoT-scale use cases.

By removing the burden of systems administration, Cloudera Data Flow delivers next-level agility. It replaces complex infrastructure management and tedious JVM (Java Virtual Machine) tuning with no-code, developer self-service tools across all phases of the data pipeline lifecycle. Key features include:

- **Data Flow Designer:** A modernized, cloud-native development canvas built specifically for developers to build, test,

and manage Apache NiFi data flows directly within a web browser. It transforms the traditional "black box" of data integration into a highly visual, drag-and-drop interface where developers can design complex routing logic, parameterize connections, and interactively explore data without writing a single line of code.

- **Built-in ReadyFlows:** A library of pre-built, fully parameterized Apache NiFi data pipelines designed to execute common enterprise data movement and ingestion use cases right out of the box. These templates allow users to implement complex data integrations — such as moving data from Kafka to Snowflake, or Salesforce to Amazon S3 — simply by filling in connection credentials, completely bypassing the need to design the pipeline on a visual canvas.
- **Data Flow Catalog:** A centralized, cloud-native repository where developers and admins can store, version, manage, and deploy Apache NiFi data flows. This central repository is where granular security controls are managed (solving the security challenge identified in Chapter 3). It acts as the single source of truth for your organization's data pipelines, allowing you to easily manage the entire lifecycle of a flow definition and deploy it seamlessly across multiple environments.
- **Advanced Deployment Monitoring and Management:** A centralized, single-pane-of-glass interface for running, monitoring, and scaling Apache NiFi data pipelines across environments. By utilizing isolated, Kubernetes-based deployments, it guarantees strict resource isolation — effectively eliminating the "noisy neighbor" risk inherent in shared environments. An operational dashboard allows infrastructure and data teams to track real-time health of pipeline deployments, auto-scale compute resources, and ensure strict SLAs are met without the need to build custom dashboards.

Use Data in Real-Time with Cloudera Streaming Analytics

Cloudera Streaming Analytics takes any data, processes any business event, as defined by any data analyst and delivers results to any data consumer. The data relevant to a given business problem

will typically be dispersed amongst streaming sources, data at rest, and high value change data capture data.

Cloudera Streaming Analytics takes advantage of unified processing, making this data instantly available as virtual tables with enforceable schemas.


TIP

Cloudera can process any business event, even complex ones, by integrating and applying analytics. Cloudera Streaming Analytics can analyze data while in motion, set up conditions to continuously monitor for in the data, and stay on top of trends and anomalies.

Any data analyst can do this using a no-code UI and author-once-publish-anywhere capabilities to push results continuously to any data consumer.

Transport Data Reliably with Cloudera Streams Messaging

If Apache Kafka is the engine under the hood, Cloudera Streams Messaging is the high-tech dashboard and automated co-pilot that makes driving it effortless. Cloudera Streams Messaging strips away the manual complexity of raw, open-source Kafka, turning it into a secure, enterprise-ready platform. By acting as the operational control center, it allows businesses to seamlessly scale their real-time applications and feed live, continuous context directly into modern AI models without the usual infrastructure headaches.

Specifically, Cloudera Streams Messaging cures key Kafka pain points by rolling Cloudera's specialized management tools into a unified ecosystem. It eliminates data blindness by using Streams Messaging Manager and Kafka Surveyor for instant, end-to-end visual tracking. It replaces manual server balancing and custom coding with automated wizards like Cruise Control, Schema Registry, and Kafka Connect, which keep cluster workloads perfectly optimized and data streams clean. Finally, it tames hybrid-cloud complexity via Streams Replication Manager, making cross-cluster replication and disaster recovery a seamless, point-and-click experience.

Ultimately, Cloudera Streams Messaging shifts your team's focus from babysitting infrastructure to delivering immediate business value. Instead of wasting time troubleshooting data lag or manually balancing servers, organizations can confidently rely on a rock-solid, automated data transport layer. It bridges the gap between your engineering realities and your strategic goals, ensuring your real-time data pipeline remains robust, secure, and always-on.

Manage the Entire Data Lifecycle with Cloudera SDX

Cloudera Shared Data Experience (SDX) offers end-to-end lineage/auditing including NiFi flows, Kafka consumers, producers and topics, and other Cloudera data services. Cloudera SDX manages and secures the entire data lifecycle from the Edge to AI in five ways:

- **Control:** Moves data and workloads between deployments for optimum performance, cost, and resilience to meet ever changing business needs
- **Metadata:** Establishes information assets for increased usability, trust, and value leveraging all metadata — structural, operational, business, and social.
- **Encryption:** Your data gets ultimate protection through automatic configuration of Kerberos backed authentication, and strong cryptography for data in motion and at rest.
- **Security:** Includes granular, dynamic, role- and attribute-based security policies.
- **Governance:** Enterprise-grade auditing, lineage, and governance capabilities applied across the platform with rich extensibility for partner integrations.

Realize Long-Term Value

Cloudera Data in Motion is built on the principles of scalable, open data structures for hybrid and multi cloud environments to deliver the best data for your AI applications. Cloudera is built to

maximize the effectiveness of domain experts, with author once, deploy anywhere capabilities, and of course enterprise grade security, governance, and support from a leading contributor to the open-source communities that revolutionized the modern data architecture.


REMEMBER

Cloudera is the only vendor with all Data in Motion components as part of an end-to-end platform — NiFi, Kafka, Flink, and Iceberg — all under unified governance. It runs anywhere: on-premises, Amazon Web Services (AWS), Azure, and Google Cloud Platform.

This means Cloudera is built on consistent improvement of its technologies well into the future.

Clear Next Steps for Data Leaders

If you need to convince your data leaders about the benefits of Cloudera — no matter if those customers are internal like your leadership or external customers who want to know their data is safe — here are the four things to know:

- **Unlock data anywhere:** Cloudera provides the same code runs anywhere with no rewrites for hybrid or multi-cloud deployments.
- **End-to-end platform:** Cloudera has NiFi, Kafka, Flink, and Iceberg, all under unified governance and security with Cloudera SDX.
- **Enterprise-grade streaming:** Cloudera offers sub-100ms latency versus 1 to 5 seconds for micro-batch.
- **Cost predictability:** Cloudera offers a flat-rate versus usage-based surprises.

Get AI-ready data from the edge

Many organizations recognize the growing importance of data in motion as a foundation for real-time AI and the next generation of intelligent applications. However, the path to effectively implementing edge to AI strategies is often unclear. This guide offers a practical roadmap to transforming data from edge devices into real-time AI use cases, along with best practices, implementation tips, tactical insights on common architectures, tooling options, and tradeoffs.

Inside...

- Understand how data in motion works
- Learn real-time AI use cases
- Practice strict governance and security
- Adopt no or low code tools
- Collect and analyze edge device data
- Accelerate enterprise AI and agentic AI
- Create streaming data pipelines
- Learn the best practices for an edge to AI strategy

CLOUDERA

Eric Butow owns Butow Communications Group and has authored or co-authored 58 books. **Diby Malakar,** Senior Director of Product Management at Cloudera, brings over 20 years of experience in data management and product leadership.

ISBN: 978-1-394-41965-4
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.