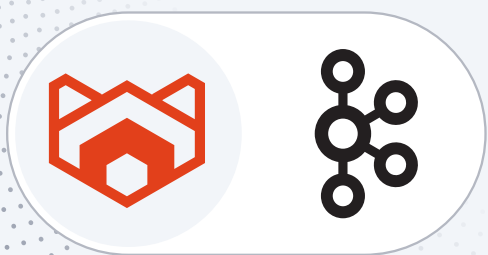




2026

Redpanda vs. Apache Kafka®

A practical guide for leaders working
with real-time data



Introduction

Anyone searching for a real-time data streaming platform will likely land on Apache Kafka®, and for good reason. Kafka is open source and widely used across the industry to build data streaming systems. However, Kafka is also notorious for its complexity. Its JVM-based architecture introduces real operational overhead and unpredictable performance at scale, making it challenging (and expensive) for businesses to keep up with the growing demand for real-time data and analytics.

Enter the **Redpanda Data Platform**, a single platform to connect, query, stream, and govern your data for real-time intelligence and data-intensive applications. Within that platform is **Redpanda Streaming**, which offers all the reliability and convenience of Kafka APIs, minus the complexity and limitations.

Built in C++, Redpanda Streaming is lean, fast, and cost-efficient at scale, simplifying how businesses build real-time data applications and AI pipelines. It's designed to connect data from any source and handle any Kafka workload, just **10X faster** and **6X more cost-effectively**. That's why data teams at the world's largest companies trust Redpanda to stream mission-critical data.

- **Johnson Controls** cut its Kafka bill in half during an 8- to-10-week migration, lowering total cost of ownership by \$1.6 million.
- **StoneX** now runs the same workloads on 5 to 6x fewer resources, with latency dropping from double-digit to single-digit milliseconds.
- **NYSE** achieved a 10x improvement in reliability and performance for its streaming services.

Redpanda Streaming is a reliable, more efficient, and simpler Kafka alternative for real-time data and analytics. This guide covers how it stacks up against Kafka, then gives you a tour of Redpanda Streaming so you know exactly what to do next.



The rise and plateau of Kafka

To understand where Kafka and Redpanda diverge, it's worth looking at how message queuing and data streaming evolved to begin with.

Message queuing: the original decoupling trick

Enterprises faced a basic problem: how do you get one application to reliably hand data to another without both being online and in sync at the exact same time? The answer was message queuing. Wall Street's need to move financial data between trading systems gave rise to what became TIBCO, and IBM was building similar message-switching systems for mainframes as early as the 1960s.

The 1980s then brought "publish-subscribe" (pub/sub), a loosely coupled model, independent in both location and time, that underpins both message queuing and the data streaming that followed.

By the 2000s, message queuing had gone open source and mainstream. As connected devices took off, MQTT and AMQP emerged as the dominant standards of the IoT era: lightweight, broker-mediated, and built for millions of small, intermittent connections.

Queues vs. streams: a fundamental split

"Message queuing" and "data streaming" get used interchangeably, but they describe different models. Traditional message queues are stateless and ephemeral: a message is delivered, acknowledged, and deleted. There's no durable log to replay, so queues excel at task distribution and point-to-point delivery, but were never designed to be a system of record.

Data streaming platforms flipped that model. Events are appended to a durable, ordered log rather than deleted on delivery, so any number of consumers can read, re-read, and replay the same stream independently, at any time.

Kafka: from LinkedIn's growing pains to industry standard

By 2010, LinkedIn had a scaling problem message queues weren't built to solve: existing queues were overkill for high-volume event ingestion, and batch systems were too slow for anything real-time. Jay Kreps, Neha Narkhede, and Jun Rao solved it by treating the message store as an append-only, replayable log rather than a transient queue. Kafka was open-sourced in 2011, donated to the Apache Software Foundation, and quickly became the default backbone for event-driven architectures industry-wide.

The cracks in Kafka's infrastructure

Here's the irony: the same design that made Kafka revolutionary is what holds it back today. Kafka was designed to exploit commodity spinning disks and the JVM ecosystem of its era, so while it can be used with modern hardware and cloud networks, it isn't optimized for them.

Kafka has also been slow to respond to the evolving needs of developers. Some of the most common complaints include:

- Kafka is built in Java, which makes it tough to reliably scale in production.
- Requires the JVM, which introduces manual heap sizing, garbage collection tuning, and unpredictable performance at scale.
- Choosing between regular and cloud-based topics means deploying different software rather than setting a configuration parameter.

- Data connectors that are easy to misconfigure but hard to customize, monitor, and debug.
- Requires an in-depth understanding of numerous settings and parameters to run efficiently.
- No user interface and the need for third-party tooling for monitoring steepen the learning curve and slow development.

Simply put, **it takes a lot of work to make Kafka work.**

These challenges caught the attention of **Alex Gallego**. In 2019, he founded Redpanda, built from the ground up in C++ with cloud-native primitives, massive scalability, and performance in mind. It was a direct answer to Kafka's architectural ceiling, not just a reimplementaion of it.

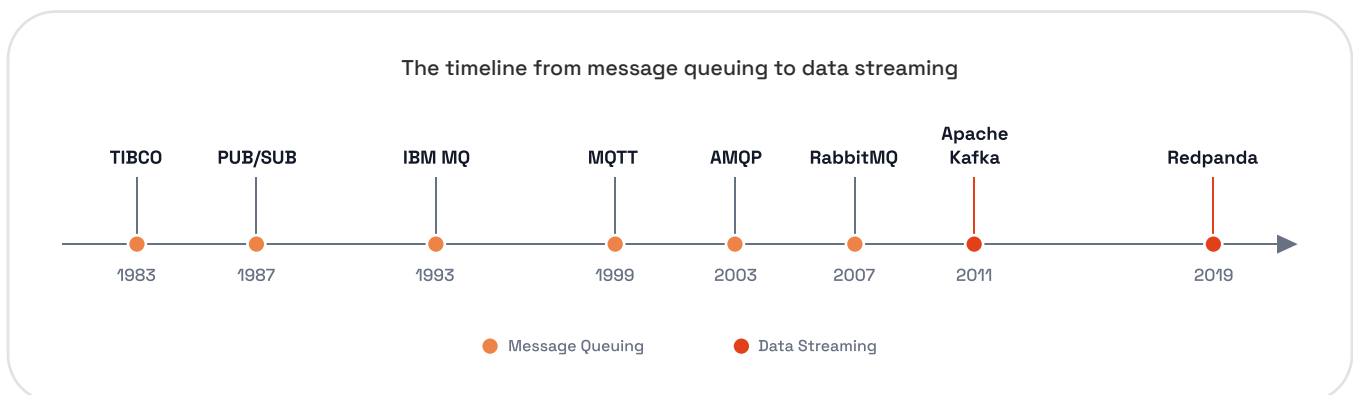
Kafka retires ZooKeeper (sort of)

To give credit where it's due, after years of running metadata management through a bolted-on ZooKeeper ensemble, Kafka 4.0 shipped without ZooKeeper in 2025, making KRaft the only supported metadata management mode. One dependency down.

But it's worth being clear-eyed about what changed. KRaft replaced ZooKeeper with Kafka's own Raft-based consensus layer, but that layer still runs as a distinct controller quorum, separate from the brokers that handle producer and consumer traffic. You've traded an external system for an internal-but-still-separate one; metadata management and data-plane brokering remain two different jobs on two different sets of nodes, coordinated rather than truly unified.

Where that leaves us today

So Kafka got simpler, but not simple. And the everyday cost of running it—Java, the JVM, tuning dozens of settings, bolting on a schema registry and monitoring stack—hasn't gone away. Unless you have a sizeable budget to burn on a fully managed Kafka solution or a dedicated team to run it in-house, you're better off choosing a platform built for modern hardware and designed for how modern real-time and AI workloads actually work.



Meet Redpanda Streaming: built for performance, engineered for simplicity

For years, streaming has forced a choice: speed or efficiency, performance or simplicity. Redpanda Streaming ends that trade-off. It's a Kafka-compatible engine that delivers the throughput and reliability teams expect, without the JVM, the dependencies, or the operational weight that make Kafka costly to run.

Here's how Redpanda Streaming answers Kafka's biggest pain points head-on.

Streaming that just works

Kafka's operational tax is well known: JVM tuning, dozens of settings to get right, and no built-in UI to see what your clusters are doing. Redpanda Streaming ships as a single binary, freeing teams from the JVM and its high memory overhead. This eliminates a whole category of dependencies:

- No external schema registry. Each node functions as both a message broker and schema registry simultaneously, simplifying deployment and reducing operational overhead.
- No Kafka Cruise Control. Redpanda's **Continuous Data Balancing** constantly monitors node and rack availability and disk usage, with self-healing clusters that dynamically balance partitions.
- No third-party monitoring bolt-ons. Built-in observability via **Prometheus** lets teams monitor cluster health with their existing DevOps and open-source tooling, and Redpanda-provided Grafana dashboards get you running fast.

Intelligent memory handling and autotuning keep clusters healthy and available at petabyte scale, so you spend less time tuning parameters and more time building. And with **Redpanda**

Console, you get an interactive UI out of the box for exploring topics, masking data, managing consumer groups, and time-travel debugging. No extra tooling required.

High performance at lower cost

Kafka gives you little room to tune performance for individual workloads, so teams end up over-provisioning or running separate clusters, which becomes expensive. Built with a thread-per-core architecture, Redpanda Streaming achieves 10x lower latencies than Kafka and supports GB/s workloads on a smaller hardware footprint, so you can do more with less without sacrificing reliability or durability. Plus, with cloud-first storage and the Raft consensus protocol as its foundation, it delivers up to 6x faster performance while ensuring zero data loss (**Jepsen-verified**).

In addition, while Kafka forces one configuration across the whole cluster, Redpanda Streaming allows a single cluster to toggle between single-digit-millisecond performance and large-scale efficiency at the topic level. Each topic runs in the mode that fits its workload:

- **Speed.** Latency-critical topics run on local NVMe storage for predictable, Raft-consistent performance.
- **Efficiency.** High-volume topics use **Cloud Topics** to write directly to object storage, cutting networking costs by over 90%. Unlike fully diskless systems that trade away reliability and features like transactions to get there, Redpanda keeps metadata and Raft consensus on local NVMe, so you get the cost savings without sacrificing data integrity.
- **Balance.** **Tiered Storage** spans local disk and object storage, letting operators deploy on fewer, smaller brokers with less storage.

One streaming engine intelligently balances performance, safety, and cost across every workload. As requirements grow, you just add more nodes.

Drop it in, run it anywhere

Kafka's connectors are easy to misconfigure but hard to customize, monitor, and debug. Because Redpanda is Kafka-API compatible, developers can drop it in, and their existing Kafka apps and tools work with no code changes. From there, the tooling is built to keep teams moving:

- **Redpanda Connect** offers 300+ pre-built connectors and YAML-configurable pipelines to move data between systems faster and easier than Kafka Connect.

- **rp** is an all-in-one CLI for setting up, configuring, and managing clusters.
- **HTTP Proxy** provides REST API access, so developers can skip Kafka's binary protocol.
- **WebAssembly (WASM) engine** performs real-time, in-broker transformations without an external stream-processing system, ending the "data ping pong."

The result is a platform that meets your existing stack where it is rather than forcing a rebuild around it.

A closer look: Redpanda Connect vs. Kafka Connect

Redpanda Connect is worth a closer look, since data integration is where a lot of Kafka's day-to-day friction lives. Here's how it compares to Kafka Connect:

Feature	Redpanda Connect	Kafka Connect
Ease of deployment	✓ Easy and fast	✗ Complex setup & tuning
Resource efficiency	✓ Lightweight	✗ Higher overhead
Data transformation	✓ Pre-built or invoke custom processors via YAML	⚠ SMTs require Java for customization
Connector options	✓ 300+ mostly open source	✓ 200+ mostly open source
Error handling	✓ Built-in recovery	⚠ Requires customization
Scalability on Kubernetes	✓ Stateless, resilient	⚠ Mix of stateful & stateless, complex
Programming flexibility	✓ Low-code, Go, or Bloolang (DSL)	✗ Java-only
Cluster replication & migration	✓ One-command Redpanda Shadowing tool	⚠ MirrorMaker 2, multiple components
AI processing	✓ Connectors for AI models & vector DBs	✗ No AI connectors
CDC support	✓ Full suite of CDC including: Oracle, DynamoDB, MySQL, Postgres, Salesforce	✓ Uses the Debezium ecosystem
Community support	✓ Active & growing	✓ Mature
Kafka dependency	✓ Use standalone or with other pub/sub systems	✗ Requires Kafka for large-scale use

Bottom line: Redpanda Connect is faster to deploy, easier to scale, more customizable, has data sovereignty built in, and simplifies migrating from Kafka using just one command. Combine its broad connector library with Redpanda Streaming, and you have a formidable toolkit for real-time data engineering teams. To learn more, read our blog post on [Kafka Connect vs. Redpanda Connect](#).

Safe, secure streaming that keeps you compliant

On open-source Kafka, enterprise security features like RBAC, audit logging, and single sign-on typically require a commercial distribution or third-party add-ons. Redpanda includes them out of the box:

- **End-to-end encryption**, Kerberos authentication, Kafka-compatible ACLs, and cloud IAM roles.
- **Single sign-on and RBAC** for standardized, centralized access control.
- **Group-based access control (GBAC)**, which maps permissions to groups defined in your identity provider, so access control absorbs organizational changes automatically instead of managing individual ACLs cluster by cluster.
- **Platform-wide audit logging** of all cluster administration and produce/consume events, providing data provenance and proof of controls to auditors.
- **A FIPS-compliant binary**, letting Redpanda Enterprise customers deploy in regulated industries like finance and the public sector.

For organizations handling sensitive data, **Redpanda BYOC**'s privacy-first architecture runs everything inside your own VPC, so your data never leaves your environment, giving you full ownership and keeping you compliant, even in the cloud. (More on that in our Redpanda deployments section below.)

Comparison table: Redpanda vs. Kafka

Now that you know each platform better, it's time for the showdown. Below is a detailed comparison of the key features and capabilities that set Redpanda Streaming apart from Kafka.

	Redpanda Enterprise	Apache Kafka®
Architecture		
Streamlined components	✓ Offers single-binary deployment with brokers, schema registry, HTTP proxy, and Raft consensus built in	✗ Imposes external dependencies on JVM, schema registry, and HTTP proxy
Engine configurations	✓ Ability to dynamically optimize individual workloads within clusters for better security and availability, greater consistency, and lower latency	✗ Less flexible with configurations that affect clusters globally and unsafe with an ISR replication strategy that creates potential for data loss
Cloud storage options	✓ Production-proven tiered storage: Redpanda Cloud Topics provides direct-to-object-storage persistence	✓ Unproven tiered storage: production-ready as of version 3.9.0. No option to use object storage directly
Provision		
Cluster sizing	✓ Consultative guidance on sizing and deployment is included	✗ DIY effort
Automation	✓ Continuous data and partition balancing built in	✗ Requires separate system (Kafka Cruise Control)
Monitoring	✓ Available natively via Prometheus	✗ Requires third-party tools
Ecosystem		
Data connectors	✓ Redpanda Connect: 300+ data and AI connectors, config-based and stateless for ease of use	✓ Kafka Connect: 200+ integrations, none for AI, includes a mix of stateful/stateless connectors
Kafka API	✓ Redpanda natively supports the Kafka API	✓ Kafka API is the industry standard for data streaming
UI	✓ Redpanda Console is open source and designed to work with Apache Kafka	✗ Kafka requires third-party tools (like Redpanda Console)
Schema registry	✓ Built into the Redpanda binary	✓ Requires separate infrastructure
Transactions	✓ Supports transactional messaging and exactly-once semantics	✓ Supports transactional messaging and exactly-once semantics
Support		
Expert support 24/7	✓ 24/7/365 coverage for production outages	✗ Community-based, best-effort support
Security	✓ Enterprise-grade features such as SSO, RBAC, GBAC, audit logging, OIDC-compliant identity provider (IdP), & more	✗ DIY security, often requires third-party tools

Redpanda deployments: connect, build, and run anywhere

Redpanda's flexible deployment options are a major reason many customers switch from Kafka. Redpanda Streaming runs everywhere your systems and applications do: at the edge, on bare metal, in the cloud, in Kubernetes, and beyond—all without breaking the bank on escalating infrastructure and storage costs.

Here's a quick look at each deployment option.

Redpanda Self-Managed

Redpanda Self-Managed offers two options for businesses that need full control over their data environment:

Redpanda Community Edition

A free, source-available version under the BSL license, which allows you to install the single binary on your machine for use in your development and test operations.

Redpanda Enterprise

Redpanda Enterprise is a more robust, self-managed solution with full commercial support, enterprise-grade security, and Enterprise features, including data balancing, one-click disaster recovery (shadowing), read replicas, and intelligent tiered storage for near-infinite data retention.

Both can be hosted on-premises or in your private cloud and, while very developer-friendly, are best suited for organizations with strong DevOps/SRE teams.

Redpanda Cloud

Redpanda Cloud offers two options: Serverless and Bring Your Own Cloud (BYOC).

Redpanda Bring Your Own Cloud (BYOC)

With Redpanda BYOC, clusters are hosted on the customer's private cloud while Redpanda remotely manages provisioning, monitoring, and maintenance. This option is designed for organizations that need to keep sensitive data within their own cloud for privacy, IP, or compliance reasons.

With BYOC, sensitive data and credentials never leave your environment, while platform engineering teams have complete visibility and control and can offload operations and maintenance to Redpanda. In short, you get full **data sovereignty** along with the convenience of a fully managed solution. Win-win.

Redpanda Serverless

Redpanda Serverless is the fastest and easiest way to start data streaming. It offers a fully self-service experience that makes creating clusters as easy as clicking a button. You don't have to talk to Redpanda to get started, so you can just start streaming data in seconds.

There's no need to choose between different capabilities, pricing structures, or availability, so you can pick the "right" cluster from the start. Redpanda Serverless eliminates the hassle of choosing the right size. With one price, you only pay for what you use.

Comparison table: Redpanda Cloud

Not sure which Redpanda Cloud edition fits? To help you weigh our most popular Cloud editions against each other, here's how Serverless and BYOC stack up.

	Redpanda Serverless	Redpanda BYOC
Provision		
Cloud providers	AWS, GCP	AWS, GCP, Azure
Data residency	Redpanda-provided infrastructure	Customer-provided infrastructure
Tenancy	Multi-tenant	Single-tenant
Elastic scaling	✓	✓
Redpanda Console UI	✓	✓
Prometheus metrics	✗	✓
Networking		
Public internet	✓	✓
Private Link	⚠️ ¹	✓
Private Service Connect	✗	✓
Security		
Login with Google	✓	✓
Login with GitHub	✓	✓
OpenID Connect	⚠️ ²	✓
Access Control Lists (ACLs)	✓	✓
Role-Based Access Controls	✗	✓
Group-Based Access Controls	✗	✓
Audit logs	⚠️ ²	✓
Redpanda Streaming		
Kafka API	✓	✓
Transactions	✓	✓
Schema registry	✓	✓
Partition replicas	Up to 5,000 ³	Up to 112,500 ³
Ingress throughput (MB/s)	Up to 100 ³	Up to 5,000 ³
Egress throughput (MB/s)	Up to 300 ³	Up to 15,000 ³
Redpanda Connect		
Available connectors	300+	300+
Disaster recovery		
Whole cluster recovery	✓	✓
Shadow replication	✗	✓
Support		
Availability SLA	99.9%	99.9%
24x7 enterprise support	⚠️ ⁴	✓

¹ Private Link is currently only available on Serverless for AWS.

² Some features are only available on the control plane for Serverless deployments.

³ Limits are dependent on workload and configurations.

⁴ Enterprise support available only with an annual commitment.

No matter which deployment you choose, you'll get a data streaming platform that's purpose-built for today's workloads, without bolt-ons or external dependencies, and the freedom to deploy anywhere.

Get started with simpler, faster data streaming

Event-driven applications demand fresher context and lower latency than legacy streaming systems can deliver. Redpanda Streaming is a Kafka-compatible platform purpose-built for today's workloads. No bolt-ons, no external dependencies, ready to deploy anywhere.

If your workloads are already on Kafka, don't worry; we're even making migration easy. With [Shadowing](#), you can simply replicate your

data into Redpanda over the Kafka API. Since Redpanda speaks the Kafka API natively, your existing applications and tools will keep working, so you can migrate without disruption.

Redpanda Streaming is already trusted by Fortune 500 companies and fast-growing startups, from stock exchanges to global manufacturers. Here's what our customers are saying:



"Redpanda is the only Kafka-compatible data streaming platform that can cost-effectively and reliably manage the massive scale of the NYSE. When milliseconds matter, Redpanda delivers."



"We have operational agility with Redpanda, meeting stringent SL0s while keeping developers focused on high-value work. Redpanda's tiered storage saves up to 30% on storage costs, easily handling 14.5GB/second."



"Redpanda rose to the top of our list because we knew it obsessed over performance and reliability at scale. We tested the data platform for up to 100M users with zero performance blips at 100GB/min throughput and peaks of 100K transactions/second."



"Redpanda just works. We had a large Kafka environment (800M events per day), dropped Redpanda in and didn't need to change any code. I was completely sold."



"I recommend Redpanda to anybody running a significant event-based workload. It's Kafka-compliant, comes with batteries included, and brings beautiful operational simplicity."

Ready when you are

If you're ready to switch to the simpler side of data streaming and learn how Redpanda can help you win in your industry, [try Redpanda Streaming for free](#). If you want to explore it more first, chat with one of our friendly product experts.

Book a demo

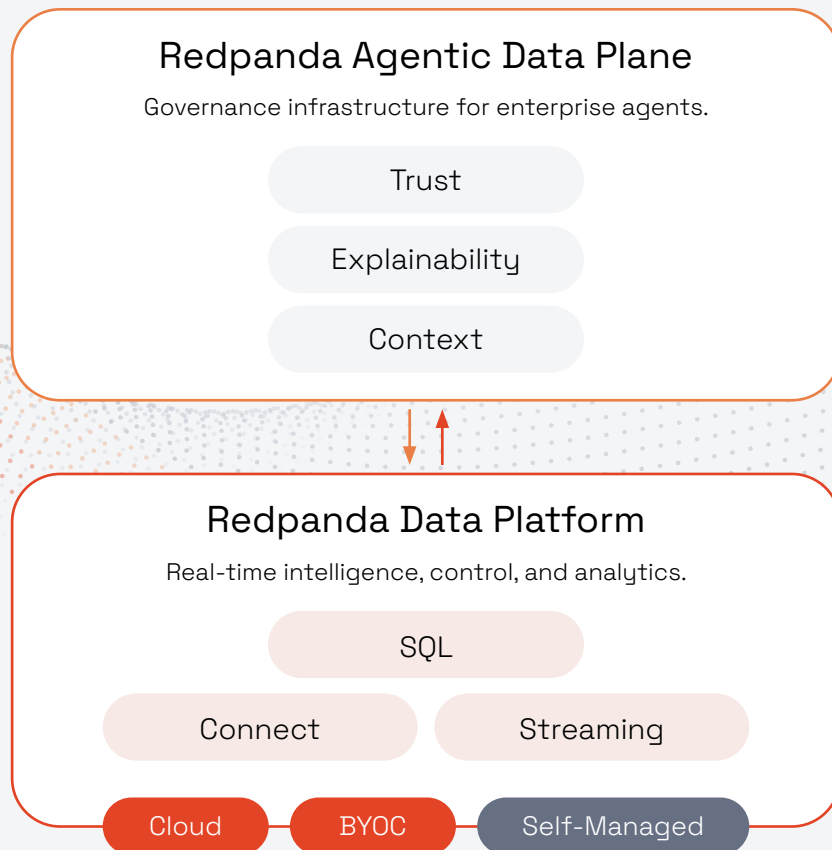
About Redpanda

Redpanda unifies data streaming, intelligent connectivity, and SQL analytics into one platform built for the world's most mission-critical environments. Redpanda Streaming is one part of that platform.

It sits within the Redpanda Data Platform alongside two complementary products:

- **Redpanda Connect** is an ecosystem of 300+ prebuilt connectors to integrate disparate data systems, fast.
- **Redpanda SQL** is a Postgres-compatible engine that queries live and historical data in one SQL statement. No waiting or pipelines required.

The Redpanda Data Platform is also the foundation for the Redpanda Agentic Data Plane, a **governance layer for enterprise AI**. Together, teams adopting Redpanda can stream, connect, query, and govern—all from a single platform.



Learn more at redpanda.com