

Overview

Get Started

Platform Overview

Quick Start

Learn More About Confluent and Kafka

Tutorial: Set Up a Multi-Broker Cluster

Scripted Confluent Platform Demo

Install and Upgrade

Build Client Applications

Confluent REST Proxy for Apache Kafka

ksqldb and Kafka Streams

Process Data With Flink

Connect to External Systems

Manage Schema Registry and Govern Data Streams

Manage Security

Deploy Confluent Platform in a Multi-Datacenter Environment

Configure and Manage

Monitor

Monitor with Health+

Confluent CLI

Release Notes

APIs and Javadocs for Confluent Platform

Glossary

Confluent Platform Overview

Confluent Platform is a full-scale [streaming platform](#) that enables you to easily access, store, and manage data as continuous, real-time streams. Built by the original creator/co-creator of Apache Kafka®, Confluent Platform is an enterprise-ready platform that completes Kafka with advanced capabilities designed to help accelerate application development and connectivity. Confluent Platform enables transformations through stream processing, simplifies enterprise operations at scale, and meets stringent architectural requirements.

Confluent Platform is software you download and manage yourself. Any Kafka use cases are also Confluent Platform use cases. Confluent Platform is a specialized distribution of Kafka that includes additional features and APIs. Many of the commercial Confluent Platform features are built into Kafka brokers as a function of Confluent Server. By integrating historical and real-time data into a single, central source of truth, Confluent makes it easy to build an entirely new category of modern, event-driven applications, gain a universal data pipeline, and unlock powerful new use cases with full scalability, performance, and reliability.

🔔 Looking for a fully managed cloud-native service for Apache Kafka®?

Sign up for [Confluent Cloud](#) and get started for free using the [Cloud quick start](#).

Kafka use cases

Kafka is used for a wide array of use cases across numerous industries, such as:

- Financial services
- Omnichannel retail
- Autonomous cars
- Fraud detection services
- Microservices

By clicking "Accept All Cookies", you agree to the storing of cookies on your device to enhance site navigation, analyze site usage, and assist in our marketing efforts.

Accept All Cookies

Reject All

Cookies Settings

location metrics, stock use case, Confluent your data rather than being transported or transform simplifies s, as well as securing,



monitoring, and managing your Kafka infrastructure.

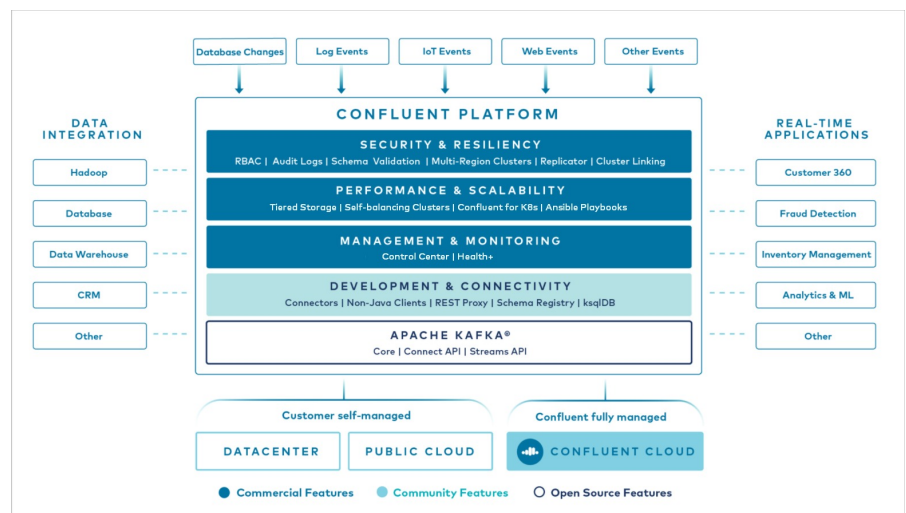
Confluent Platform features

At the core of Confluent Platform is Kafka, the most popular open source distributed streaming platform. Kafka enables you to:

- Publish and subscribe to streams of records
- Store streams of records in a fault tolerant way
- Process streams of records

Each Confluent Platform release includes the latest release of Kafka and additional tools and services that make it easier to build and manage an event streaming platform.

Confluent Platform provides community and commercially licensed features such as [Schema Registry](#), [Cluster Linking](#), a [REST Proxy](#), [100+ pre-built Kafka connectors](#), and [ksqlDB](#). For more information about Confluent components and the license that applies to them, see [Confluent Licenses](#).



Confluent Platform Components

Kafka capabilities

Confluent Platform provides all of Kafka's open-source features plus additional proprietary components. Following is a summary of Kafka features. For an overview of Kafka use cases, features and terminology, see [Kafka Introduction](#).

- At the core of Kafka is the [Kafka broker](#). A broker stores data in a durable way from clients in one or more topics that can be consumed by one or more clients. Kafka also provides several [command-line tools](#) that enable you to start and stop Kafka, create topics and more.
- Kafka provides security features such as [data encryption](#) between producers and consumers and brokers using SSL / TLS. [Authentication](#) using SSL or SASL and authorization using ACLs. These security features are disabled by default.
- Additionally, Kafka provides the following [Java APIs](#).
 - The Producer API that enables an application to send messages to Kafka. To learn more, see [Producer](#).
 - The Consumer API that enables an application to subscribe to one or more topics

- and process the stream of records produced to them. To learn more, see [Consumer](#).
- [Kafka Connect](#), a component that you can use to stream data between Kafka and other data systems in a scalable and reliable way. It makes it simple to configure connectors to move data into and out of Kafka. Kafka Connect can ingest entire databases or collect metrics from all your application servers into Kafka topics, making the data available for stream processing. Connectors can also deliver data from Kafka topics into secondary indexes like Elasticsearch or into batch systems such as Hadoop for offline analysis.
 - The [Streams API](#) that enables applications to act as a stream processor, consuming an input stream from one or more topics and producing an output stream to one or more output topics, effectively transforming the input streams to output streams. It has a very low barrier to entry, easy operationalization, and a high-level DSL for writing stream processing applications. As such it is the most convenient yet scalable option to process and analyze data that is backed by Kafka.
 - The [Admin API](#) that provides the capability to create, inspect, delete, and manage topics, brokers, ACLs, and other Kafka objects. To learn more, see [Confluent REST Proxy for Apache Kafka](#), which leverages the Admin API.

Development and connectivity features

To supplement Kafka's Java APIs, and to help you connect all of your systems to Kafka, Confluent Platform provides the following features:

- Confluent Connectors, which leverage the Kafka Connect API to connect Kafka to other systems such as databases, key-value stores, search indexes, and file systems. Confluent Hub has downloadable connectors for the most popular data sources and sinks. These include fully tested and supported versions of these connectors with Confluent Platform. See the following documentation for more information:
 - [How to Use Kafka Connect - Get Started](#)
 - [Supported Self-Managed Connectors](#)
 - [Preview Self-Managed Connectors](#)

Confluent provides both commercial and community licensed connectors. For details, and to download connectors, see [Confluent Hub](#).

- [Non-java clients](#) such as a [C/C++](#), [Python](#), [Go](#), and [.NET client](#) libraries in addition to the Java client. These clients are full-featured and performant. For more information, see the [Build Client Applications for Confluent Platform](#). In addition, Confluent provides a JMS-compatible client for Kafka. For more information, see [JMS Client for Confluent Platform](#).
- A [REST Proxy](#), which leverages the Admin API and makes it easy to work with Kafka from any language by providing a RESTful HTTP service for interacting with Kafka clusters. The REST Proxy supports all the admin core functionality: sending messages to Kafka, reading messages, both individually and as part of a consumer group, and inspecting cluster metadata, such as the list of topics and their settings. You get the full benefits of the high quality, officially maintained Java clients from any language. The REST Proxy also integrates with Schema Registry. Because it automatically translates JSON data to and from Avro, you can get all the benefits of centralized schema management from any language using only HTTP and JSON.
- All of the Kafka command-line tools and additional tools, including the [Confluent CLI](#).

You can find a list of all of these tools in [CLI Tools Bundled With Confluent Platform](#).

- [Schema Registry](#), which provides a centralized repository for managing and validating schemas for topic message data, and for serialization and deserialization of data over a network. With a messaging service like Kafka, services that interact with each other must agree on a common format, called a schema, for messages. Schema Registry helps enable safe, zero-downtime evolution of schemas by centralizing schema management. It provides a RESTful interface for storing and retrieving Avro®, JSON Schema, and Protobuf schemas. Schema Registry tracks all versions of schemas and enables the evolution of schemas according to user-defined compatibility settings. Schema Registry also includes plugins for Kafka clients that handle schema storage and retrieval for Kafka messages that are sent in the Avro format. For more information, see the [Schema Registry Documentation](#). For a hands-on introduction to working with schemas, see the [On-Premises Schema Registry Tutorial](#). For a deep dive into supported serialization and deserialization formats, see [Formats, Serializers, and Deserializers](#).
- [ksqlDB](#), a streaming SQL engine for Kafka. It provides an interactive SQL interface for stream processing on Kafka, without the need to write code in a programming language such as Java or Python. ksqlDB is scalable, elastic, fault-tolerant, and real-time. It supports a wide range of streaming operations, including data filtering, transformations, aggregations, joins, windowing, and sessionization. For more information, see the [ksqlDB Documentation](#), or the [ksqlDB getting started guide](#).
- A [MQTT Proxy](#), which provides a way to publish data directly to Kafka from MQTT devices and gateways without the need for a MQTT Broker in the middle. For more information, see [MQTT Proxy](#).

Management and monitoring features

Confluent Platform provides several features to supplement Kafka's Admin API, and built-in JMX monitoring.

- [Confluent Control Center](#), which is a web-based system for managing and monitoring Kafka. It allows you to easily manage Kafka Connect, to create, edit, and manage connections to other systems. Control Center also enables you to monitor data streams from producer to consumer, assuring that every message is delivered, and measuring how long it takes to deliver messages. Using Control Center, you can build a production data pipeline based on Kafka without writing a line of code. Control Center also has the capability to [define alerts](#) on the latency and completeness statistics of data streams, which can be delivered by email or queried from a centralized alerting system.
- [Health+](#), also a web-based tool to help ensure the health of your clusters and minimize business disruption with intelligent alerts, monitoring, and proactive support.
- [Metrics reporter](#) for collecting various metrics from a Kafka cluster. The metrics are produced to a topic in a Kafka cluster.

Performance and scalability features

Confluent offers a number of features to scale effectively and get the maximum performance for your investment.

- To help save money, you can use the [Tiered Storage in Confluent Platform](#) feature, which automatically tiers data to cost-effective object storage, and scale brokers only when you

need more compute resources.

- [Manage Self-Balancing Kafka Clusters in Confluent Platform](#) provides automated load balancing, failure detection and self-healing for your clusters. It provides support for adding or decommissioning brokers as needed, with no manual tuning. Self-Balancing Clusters are the next iteration of Auto Data Balancer in that Self-Balancing auto-monitors clusters for imbalances, and automatically triggers rebalances based on your configurations. Partition reassignment plans and execution are taken care of for you.
- To deploy Confluent Platform at scale, you can deploy with [Ansible Playbooks](#) or [Confluent for Kubernetes](#). - [Confluent for Kubernetes](#) is a Kubernetes operator. Kubernetes operators extend the orchestration capabilities of Kubernetes by providing the unique features and requirements for a specific platform application. For Confluent Platform, this includes greatly simplifying the deployment process of Kafka on Kubernetes and automating typical infrastructure lifecycle tasks. - [Ansible Playbooks](#) enable you to automatically configure, provision, and deploy Confluent Platform clusters using YAML files.

Security and resilience features

Confluent Platform also offers a number of features that build on Kafka's security features to help ensure your deployment stays secure and resilient.

- You can set authorization by role with Confluent's [Role-based Access Control \(RBAC\)](#) feature.
- If you use Control Center, you can set up [Single Sign On \(SSO\)](#) that integrates with a supported OIDC identity provider, and enable additional security measures such as multi-factor authentication.
- The [REST Proxy Security Plugins in Confluent Platform](#) and [Schema Registry Security Plugin for Confluent Platform](#) add security capabilities to the Confluent Platform REST Proxy and Schema Registry. The Confluent REST Proxy Security Plugin helps in authenticating the incoming requests and propagating the authenticated principal to requests to Kafka. This enables Confluent REST Proxy clients to utilize the multi-tenant security features of the Kafka broker. The Schema Registry Security Plugin supports authorization for both role-based access control (RBAC) and ACLs.
- [Audit logs](#) provide the ability to capture, protect, and preserve authorization activity into topics in Kafka clusters on Confluent Platform using [Confluent Server Authorizer](#).
- The [Cluster Linking](#) feature enables you to directly connect clusters and mirror topics from one cluster to another. This makes it easier to build multi-datacenter, multi-region and hybrid cloud deployments.
- [Confluent Replicator](#) makes it easier to maintain multiple Kafka clusters in multiple data centers. Managing replication of data and topic configuration between data centers enables use-cases such as active geo-localized deployments, centralized analytics and cloud migration. You can use Replicator to configure and manage replication for all these scenarios from either Control Center or command-line tools. To get started, see the [Replicator documentation](#), including the [Replicator Quick Start](#).

Install Confluent Platform

You can install Confluent Platform and its components manually using [ZIP or TAR installation](#) packages, using package managers on [CentOS](#), [RHEL](#), [Rocky Linux](#), [Ubuntu](#), [Debian](#), or [Docker images](#). The [Quick Start for Confluent Platform](#) uses Docker Compose to get you up and running quickly.

Docker images of Confluent Platform are available on [Docker Hub](#). To learn more about the packages, see [Docker Image Reference for Confluent Platform](#). Some of these images contain proprietary components that require a Confluent enterprise license.


You can also install using an orchestrator like [Ansible Playbooks](#) or [Confluent for Kubernetes](#). For more information about all of the installation options, see [Install Confluent Platform On-Premises](#).

If you would rather take advantage of all of Confluent Platform's features in a managed cloud environment, you can use [Confluent Cloud](#) and get started for free using the [Cloud quick start](#).

Was this doc page
helpful?

 [Give us feedback](#)

Do you still need help?

 [Confluent support portal](#)

 [Ask the community](#)

Be the first to get updates and new content

SIGN UP

By clicking "SIGN UP" you agree that your personal data will be processed in accordance with our [Privacy Policy](#).

Confluent

[About](#)

[Careers](#)

[Contact](#)

[Professional Services](#)

Developer

[Free Courses](#)

[Tutorials](#)

[Event Streaming Patterns](#)

Product

[Confluent Cloud](#)

[ksqlDB](#)

Community

[Forum](#)

[Meetups](#)

[Kafka Summit](#)

[Documentation](#)

[Catalysts](#)

[Blog](#)

[Podcast](#)



[Terms & Conditions](#) | [Privacy Policy](#) | [Do Not Sell My Information](#) | [Modern Slavery Policy](#) |

[Cookie Settings](#) | [Feedback](#)

Copyright © Confluent, Inc. 2014- 2024 Apache, Apache Kafka, Kafka, the Kafka logo, Apache Flink, Flink, the Flink logo, Apache Iceberg, Iceberg, and the Iceberg logo are trademarks of the **Apache Software Foundation**