

Confluent Cloud vs. Amazon MSK

The rise of Apache Kafka managed services

Apache Kafka was designed with scale in mind. Capable of scaling horizontally to handle extremely high fanout and throughput as needed, Kafka is an ideal data streaming technology for use cases that require high performance and increasing adoption within an organization.

However, due to its distributed architecture, the operational burden of managing Apache Kafka can quickly become a limiting factor on either adoption or developer agility. With an ever-increasing demand for technical talent, especially those with deep Apache Kafka expertise, organizations often look to managed services to accelerate development timelines and boost developer productivity while lowering total cost of operating Apache Kafka.

Every managed Apache Kafka service has the potential to check certain boxes in your list of your technical requirements, since they are all leveraging the same open source technology. What sets them apart is each service's level of automation and cloud-native features across planning, provisioning, operating and scaling stages in order to enable your team to focus on application development versus managing infrastructure.

This guide will walk through the complete journey of an Apache Kafka deployment while comparing the experience between a cloud-native service and a cloud-hosted service as to inform application developers, architects and technical leaders on whether their choice of managed Apache Kafka service will truly enable them to meet their unique business objectives.

Types of Apache Kafka managed services

Cloud-Native

A true cloud-native service for Apache Kafka, such as Confluent Cloud, is one that is built from the ground up specifically for the cloud. It embraces the scalability and elasticity of public cloud infrastructure by decoupling compute and storage and abstracting the underlying complexities of operating Kafka. With this type of service, users can focus on their applications; operational tasks such as deployment, maintenance, scaling, and security management are automatically handled.

Cloud-Hosted

A hosted service for Apache Kafka, such as Amazon Managed Streaming for Apache Kafka (Amazon MSK), is one that takes existing software and installs it in a public cloud environment and adds automation to some operations. With this implementation, the end user is still required to manage and monitor clusters at a broker level. In the long term, this adds complexity, overhead and potential operational risk as teams are handed the responsibility of maintaining highly available clusters during manual cluster scaling and upgrades. Amazon offers a serverless offering as well (MSK Serverless), but it lacks many mission-critical components for end-to-end data streaming use cases.

Confluent Cloud				vs		Amazon MSK			
Automated operations enable efficient growth						Operational burden rapidly increases beyond deploy			
manual operations (less is better)	topic monitoring			non-Java clients		sizing		infra monitoring	cluster expansion
	upgrades							up-to-date software patches	scale to 0
	topic monitoring							Kafka UI	non-Java clients
								ksqlDB apps	
automated product capabilities (more is better)	sizing	provisioning	infra monitoring	cluster expansion					
	connect worker infra	upgrades	connectors						
	stream processing infra	up-to-date software patches	scale to 0	stream processing infra					
	Kafka UI	storage capacity & mgmt	connect worker infra						
	REST proxy	provisioning							
Plan > Deploy > Operate > Scale				Plan > Deploy > Operate > Scale					
Freedom of Choice Committer-Driven Kafka Expertise				AWS Only Limited Kafka Support & Expertise					
Cloud-Native				Self-Managed		Custom			

Although MSK addresses some operational gaps, it still has many limitations

Confluent Cloud				VS	MSK Serverless			
Automated operations enable efficient growth					Operational burden rapidly increases beyond deploy			
manual operations (less is better)								REST proxy
						Kafka UI		non-Java clients
		topic monitoring	non-Java clients			topic monitoring		connectors
automated product capabilities (more is better)			REST proxy					
			Kafka UI					
		stream processing infra	up-to-date software patches			stream processing infra	up-to-date software patches	storage capacity & mgmt
		connect worker infra	upgrades			connect worker infra	upgrades	scale to 0
sizing	provisioning	infra monitoring	cluster expansion		sizing	provisioning	infra monitoring	cluster expansion
Plan > Deploy > Operate > Scale					Plan > Deploy > Operate > Scale			
Freedom of Choice Committer-Driven Kafka Expertise					AWS Only Limited Kafka Support & Expertise			
Cloud-Native					Self-Managed			
					Custom/Unsupported			

Plan

Sizing

The first step in the data streaming journey is sizing your initial production environment. In a self-managed world, this means picking instance or VM types for Kafka and Zookeeper nodes - a process that can take anywhere from days to weeks depending on the number of performance tests needed to choose the right instance type. Once that is completed, it is time to then choose the number of brokers needed to meet streaming requirements such as peak throughput and data retention requirements.

Sizing the cluster with Amazon MSK is not very different in that users still need to think in terms of brokers. On the contrary, Confluent Cloud uses a throughput-based model to abstract the complexity of sizing the underlying Kafka infrastructure. By not having to worry about the number of brokers, you can get started in minutes versus weeks. With throughput-based sizing, you can remove cumbersome performance testing cycles and eliminate concerns around overprovisioning clusters and running up the cloud bill or underprovisioning and impacting application performance with the risk of causing downtime.

MSK Serverless does offer throughput-based sizing supporting up to 200MBps ingress and 400MBps egress.

Deploy

Provisioning

Automating provisioning is key to accelerating your team's access to production-ready environments. Provisioning automation includes implementation of Kafka security best practices such as automatically enabling at-rest and in-transit data encryption. Both Confluent Cloud and Amazon MSK (both provisioned and serverless) can help gain access to quick, on-demand provisioning as part of a cloud-native experience.

A unique value proposition of a cloud-native solution such as Confluent Cloud is the ability to deploy clusters where you only pay for usage versus infrastructure. For example, Confluent Cloud Basic clusters can be provisioned within seconds, and with their scale-to-zero pricing, they can be leveraged to reduce infrastructure costs during development phases. Once development is complete, these same clusters can be automatically upgraded to Standard clusters which are ready for production with higher SLAs and multi-AZ replication

for higher resiliency. By paying for usage instead of paying for provisioned infrastructure, your team can optimize costs throughout the Kafka journey.

On the other hand, MSK Provisioned does not support scaling down and only supports four auto-scaling operations per day. Therefore, you could end up paying for provisioned storage rather than consumed storage.

Operate

Infrastructure Monitoring

When self-managing Apache Kafka, monitoring infrastructure metrics such as CPU and disk utilization are key to maintaining the uptime and performance of each cluster. Not so differently (other than the additional cost to export metrics), a cloud-hosted service like Amazon MSK requires you to consume low-level metrics with the goal of self monitoring, diagnosing and correcting issues related to cluster performance.

With Confluent Cloud, our Kafka experts are proactively monitoring and optimizing the underlying infrastructure so you can stay focused on your applications.

Like Confluent Cloud, there's no need for monitoring infrastructure or disk usage with MSK Serverless, but storage is limited to 250GB per partition for compacted topics.

Topic Level Monitoring

Topics are a key architecture element in Apache Kafka. Kafka topics are divided into a number of partitions that allow you to parallelize a topic by splitting it across multiple brokers. To monitor how much data is flowing through a topic you must manually aggregate partition level metrics from each broker. As more brokers are added to a cluster and partitions are rebalanced the aggregation logic also needs to evolve.

Amazon MSK provides users with predefined Kafka metrics in Cloudwatch, a separate AWS service built for general monitoring and APM across ~70 AWS offerings. MSK users can access raw metrics to perform manual analysis needed for metrics not delivered out-of-the-box through Cloudwatch. Confluent Cloud provides users with pre-aggregated metrics, some not even available with Kafka such as storage used per topic, to help developers simplify topic level monitoring and gain the most valuable insights about their streaming applications.

With Stream Lineage, visualizing and monitoring end-to-end flow from producers to consumers can easily be done within the Cloud UI with the option to build custom dashboards in your monitoring tool of choice using Metrics API at no additional cost.

MSK Serverless also offers free and pre-aggregated metrics for topic monitoring, but does not offer native integration with APM tools like Dynatrace and Datadog. Customers also have the option to access Cloudwatch metrics, but will incur additional costs.

Upgrades

A fully managed service comes with the expectation of running on the latest software version. Confluent Cloud automatically upgrades customers to the latest version of Kafka, and it delivers upgrades faster than other vendors because of its unmatched insight and investment into open source Kafka.

MSK requires users to trigger the upgrade cycle, which becomes available to customers after AWS determines whether they will support that given release. For MSK Serverless, AWS does not provide info on the Kafka version supported. Unlike provisioned MSK, the user is not tasked to upgrade. But given the history of MSK lagging behind in versions, Serverless might not be on the latest stable version.

Vulnerability Patching

Streaming with the latest stable version of Kafka means streaming with the latest vulnerability patches. Confluent Cloud's rapid and automatic upgrades ensure that its users never miss the critical patches that any new release of OSS Kafka may contain. When combined with Confluent's Kafka-specific technical support team, these rapid updates allow Confluent to offer an all-encompassing SLA to users. In contrast, MSK provides an SLA specifically for its service uptime, but excludes failures resulting from the given version of Kafka that any number of customers may be running at any given time.

Scale

Cluster Scaling

Cluster scaling, whether out or in, will be required at multiple points in time in your Kafka journey. Some will be part of planned expansions such as onboarding a new application, but

many times there will be unplanned expansions needed as part of unexpected surges in traffic. With elastic scaling and self-balancing clusters, Confluent Cloud is architected to drastically reduce the operational burden of managing cluster capacity changes.

MSK enables customers to scale using their console or an API, but that process requires active manual monitoring of your environment. MSK also requires a manual process to rebalance and reassign partitions, which is a non-trivial task while scaling out. Also, MSK does not allow you to scale in. Like Confluent Cloud, MSK Serverless offers elastic scaling with no need to rebalance the cluster. However, there is no solution to upgrade when exceeding the given quota.

In a hypercompetitive digital world where customer loyalty is aligned to the best performing applications, every millisecond delay could be another lost customer. This could be the difference between hitting or missing revenue targets and making headlines for the wrong reasons. Adjusting to the real-time needs of your mission-critical apps, without paying for over-provisioned infrastructure, is only possible with a cloud-native service like Confluent Cloud. With self-balancing clusters, Confluent Cloud is continuously optimizing resource allocation to manage consumer lag without any operational burden to you.

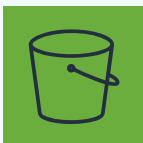
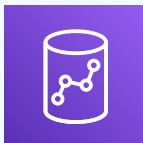






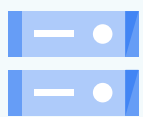



Connectors

In order to maintain a real-time view across your entire business, Apache Kafka as the backbone of your application data architecture is only as valuable as the number of data services connected to it. On average, it takes Confluent engineers anywhere from 3-6 engineering months to design, build, and test each connector before providing it as a fully managed component. By leveraging our 120+ connectors available either as fully managed components or as pre-built components covered by Confluent support, or custom connectors, you can de-risk time-to-value and accelerate your team's productivity. Connecting existing data systems to Kafka in a repeatable way will enable your team to focus on developing streaming applications instead of building and operating foundational tools such as integrations to data sources and sinks.

Without AWS supported Kafka connectors for Amazon MSK, teams must own the responsibility of building, managing and providing technical support for integrations that must meet production-level requirements. Confluent also connects to more native AWS services than Amazon MSK or MSK Serverless.

Non-Java Clients

Enabling developers to easily connect to clusters in their most productive language is a key step in making Apache Kafka widely available to all developers. Confluent Cloud enables the adoption of Kafka through your organization with a wide variety of Confluent clients that are easy to use, high performing and battle-tested to meet the demands of modern applications. AWS does not provide technical support for Confluent or other community-built clients leaving teams with the operational burden of integrating and supporting non-Java clients on their MSK clusters.

CONNECTORS FOR AWS			
			
AMAZON S3 SINK	AMAZON REDSHIFT SINK	AWS LAMBDA SINK	AMAZON KINESIS SOURCE
CONNECTORS FOR AZURE			
			
BLOB STORAGE SINK	AZURE SYNAPSE SINK	AZURE FUNCTIONS SINK	EVENT HUBS SOURCE
CONNECTORS FOR GOOGLE CLOUD			
			
CLOUD STORAGE SINK	BIGQUERY SINK	CLOUD FUNCTIONS SINK	PUB/SUB SOURCE

See all available connectors at www.confluent.io/hub

Expertise

Support

Businesses that embrace Kafka often build their most critical applications and processes on it. This makes expert support critical, because if your Kafka service goes down, your business can go down.

Confluent's Technical Support Engineers have over 1 million hours of Kafka expertise solving tens of thousands of Kafka-related issues. Our support engineers resolve issues hand in hand with the same engineers that contribute to the open source project. As such, you are working with a committer-led team specialized in maintenance and development of Kafka, Connect, Flink, ksqlDB, and Kafka Streams as opposed to general support. With a global team of experts ready to resolve even the most complicated Kafka issues 24x7, you can minimize the time required to take Kafka to production and truly free your team from spending time monitoring clusters.

To accelerate your project's time to market even further, Confluent also offers customers advisory services to provide expert guidance and share best practices for your streaming architecture and streaming applications.

AWS offers support across the over 200 services in its portfolio, including MSK. With this breadth, it's important to understand the variety of support levels offered for each service. Amazon MSK offers an SLA, but has exclusions around issues caused by underlying Apache Kafka or Apache Zookeeper software.

Future Proof

Environments

Confluent provides the ultimate freedom in terms of flexibility and choice to truly enable your strategic initiatives for hybrid and multi-cloud deployments whereas Amazon MSK and MSK Serverless were built to run on AWS.

As a cloud-native service, Confluent Cloud provides a consistent Kafka experience in more than 85+ regions across all three major cloud providers - AWS, Azure and Google Cloud. And with Confluent Platform, our self-managed software, you can also deploy a consistent data architecture in your on-premises or private cloud environments.

Ecosystem

Confluent Cloud is more than just a managed Apache Kafka service while Amazon MSK and MSK Serverless are services that are limited to helping you deploy Apache Kafka clusters only.

In addition to pre-built and fully managed connectors to popular data sources and sinks, Confluent Cloud also offers Schema Registry, Flink, and ksqlDB as fully managed components. With managed Confluent Schema Registry, developers can adhere to standard schemas across Kafka applications in a simple, centralized and scalable way to maintain application compatibility with no additional infrastructure to manage. AWS offers a schema registry for Kafka through AWS Glue, a separate data integration service.

With Confluent's fully managed Flink and ksqlDB services, developers can enrich and process real-time data streams instantaneously with a few SQL statements without having to manage any infrastructure. AWS offers stream processing capabilities as well through Lambda and Managed Service for Apache Flink (MSF); MSF is powerful but adds complexities as users need to configure networking, author the job using SQL-like syntax, and other additional steps.

For data governance, Confluent provides a suite of fully managed services that work together to manage the availability, integrity, and security of data used across the organization. This suite is called Stream Governance and is built upon three key strategic pillars: Stream Lineage, Stream Catalog, Stream Quality. On the other hand, neither MSK nor MSK Serverless have lineage or catalog capabilities and need to rely on free community tools without support or paid third-party tools to provide stream governance. For data quality, both integrate with Glue Schema registry. However, they lack broker-side schema validation that enforces that data producers to use schema registry to control schema evolution, Data Quality Rules to validate and constrain individual field values within a data stream, and schema linking that syncs schemas across different environments.

Start Your Cloud Native Kafka Journey with Confluent Cloud Today

confluent.io/cloud-native-kafka

	CONFLUENT CLOUD	AMAZON MSK	MSK SERVERLESS
PLAN			
Sizing	Throughput-based <ul style="list-style-type: none"> Eliminate cumbersome performance testing and reduce infrastructure costs with elastically scalable, scale-to-zero pricing clusters where you only pay for usage versus provisioned infrastructure 	Broker-based <ul style="list-style-type: none"> Allocate time and resources to run performance tests to pick broker types and count Overprovision infrastructure to reduce the need for complex expansions later due to auto-scaling limitations (four operations per day and no scaling down) 	Throughput-based <ul style="list-style-type: none"> Elastically scale with support up to 200MBps in and 400MBps out
DEPLOY			
Provision	Self-serve, on-demand <ul style="list-style-type: none"> Provision Kafka clusters along with any other Confluent Cloud component including Schema Registry, Connect & Flink 	Self-serve, on-demand for Kafka only <ul style="list-style-type: none"> Provision Kafka clusters along with Glue Schema Registry and Flink Custom efforts needed for connectors and Kafka proxy 	Self-serve, on-demand for Kafka only <ul style="list-style-type: none"> Provision Kafka clusters along with Glue Schema Registry and Flink Custom efforts needed for connectors and Kafka proxy
Infra as Code	For both control plane and data plane <ul style="list-style-type: none"> Unified experience for deploying and managing Confluent resources across diverse environments. Leverage Terraform provider to manage both control plane resources—like Clusters and Schema Registry—and data plane resources, such as Topics and ACLs 	For control plane only <ul style="list-style-type: none"> Terraform is only able to deploy and manage control plane resources. Need to build custom operators and processes to manage data plane resources 	For control plane only <ul style="list-style-type: none"> Terraform is only able to deploy and manage control plane resources. Need to build custom operators and processes to manage data plane resources
OPERATE			
Infra Monitoring	Proactive monitoring <ul style="list-style-type: none"> Stay focused on app development with proactive cluster monitoring and maintenance from the Kafka experts Infinite Storage enables unlimited cluster-level storage use cases while reducing risk of disk space-related failures 	Manual monitoring <ul style="list-style-type: none"> Assign resources to monitor broker metrics such as CPU utilization to proactively manage cluster performance Monitor and create alerts for disk space to prevent failures due to storage capacity 	Proactive monitoring <ul style="list-style-type: none"> Stay focused on app development with proactive cluster monitoring and maintenance Infinite Storage enables unlimited cluster-level storage use cases while reducing risk of disk space-related failures
Topic monitoring	Free, pre-aggregated metrics <ul style="list-style-type: none"> Access most important metrics pre-aggregated at the topic and cluster level at no additional cost; consume metrics with your third party monitoring service of choice using Metrics API 	Topic-level metrics cost extra <ul style="list-style-type: none"> Pay to consume and manually aggregate per-broker and per-topic level metrics to monitor overall usage 	Default monitoring is free <ul style="list-style-type: none"> Access topic-level metrics for free using CloudWatch console, a separate tool for monitoring multiple AWS products; does not include partition level metrics, or native-integration with popular monitoring tools like Datadog and Dynatrace

	CONFLUENT CLOUD	AMAZON MSK	MSK SERVERLESS
OPERATE			
Upgrades	Always latest stable version <ul style="list-style-type: none"> Zero intervention as part of non-disruptive rolling upgrades to latest stable Kafka version that includes strategic patches ahead of scheduled Apache releases 	Limited version support <ul style="list-style-type: none"> Manually trigger upgrades once AWS adds support for it after the scheduled Apache release; MSK only supports a subset of Kafka releases 	Limited version support <ul style="list-style-type: none"> Zero intervention as part of non-disruptive rolling upgrades; MSK only supports a subset of Kafka releases and latest version is unknown and fully abstracted
Software patches	Proactive fixes <ul style="list-style-type: none"> Stream confidently and reliably with Kafka experts that proactively address known bugs and vulnerabilities and resolve even the most complicated Kafka issues 	Reactive fixes <ul style="list-style-type: none"> MSK chooses to offer only select versions of Kafka and failures due to Kafka software not covered by MSK uptime SLAs. Embracing a subset of releases forces a reactive approach to fixing vulnerabilities. 	Reactive fixes <ul style="list-style-type: none"> MSK chooses to offer only select versions of Kafka and failures due to Kafka software not covered by MSK uptime SLAs. Embracing a subset of releases forces a reactive approach to fixing vulnerabilities.
SCALE			
Cluster expansions	Elastic scalability <ul style="list-style-type: none"> Automatic resource allocation to your cluster to manage consumer lag as throughput scales up or down with self-balancing clusters Eliminate cluster compute over provisioning when increasing topic retention with Infinite Storage feature 	Manual data rebalancing <ul style="list-style-type: none"> Manual data rebalancing process required using Cruise Control after brokers are added to any cluster Storage per broker limitations force users to either overpay for compute or force exporting data out of Kafka for long data retention use cases 	Elastic scalability <ul style="list-style-type: none"> Effortless scaling up and down from 0 to 200 MBps with automatic cluster rebalancing Eliminate cluster compute over-provisioning when increasing topic retention with Infinite Storage feature
Connectors	Pre-built and fully managed <ul style="list-style-type: none"> Accelerate integration to modern and legacy services across on-premises and public clouds with a continuously growing portfolio of 120+ connectors available either as fully managed components or as pre-built components covered by Confluent support 	Self develop and manage <ul style="list-style-type: none"> Leverage self-built or community-built connectors without expressed technical support from AWS; only underlying MSK Connect infrastructure is included 	Self develop and manage <ul style="list-style-type: none"> Leverage self-built or community-built connectors without expressed technical support from AWS; only underlying MSK Connect infrastructure is included
Non-Java clients	Battle tested and Confluent supported <ul style="list-style-type: none"> Enable developer velocity and make Kafka widely accessible to applications and services with a wide variety of battle-tested clients for C, Java, .Net, Go, Python and more 	Self supported <ul style="list-style-type: none"> Leverage a variety of clients available through open source licenses without any technical support from AWS 	Self supported <ul style="list-style-type: none"> Leverage a variety of clients available through open source licenses without any technical support from AWS

CLOUD-NATIVE

SELF-MANAGED

CUSTOM

CONFLUENT CLOUD

AMAZON MSK

MSK SERVERLESS

EXPERTISE

Support

Committer-driven expertise

- Expert 24x7 support engineers have solved tens of thousands of Kafka-related issues that become commits to the open source.
- Accelerate time to market even further with expert guidance with advisory services team.

General AWS Support

- AWS boasts over 200 services, of which MSK is just one. Their support is not as focused as Confluent's on Kafka.
- Amazon MSK's SLA has exclusions for issues caused by underlying Apache Kafka engine and Zookeeper failures.

General AWS Support

- AWS boasts over 200 services, of which MSK is just one. Their support is not as focused as Confluent's on Kafka.
- Amazon MSK's SLA has exclusions for issues caused by underlying Apache Kafka engine and Zookeeper failures.

FUTURE PROOF

Environments

Freedom of Choice

- Consistent cloud native experience across AWS, Azure and Google Cloud.
- Extend consistent data architecture to on-premises or private cloud environments with Confluent Platform, our self-managed software.

AWS only

- Only available on AWS limiting hybrid cloud and multi-cloud use cases.

AWS only

- Only available on AWS limiting hybrid cloud and multi-cloud use cases.

Ecosystem

Complete

- Maintain application compatibility with fully managed Confluent Schema Registry.
- Develop real-time streaming data pipelines with a lightweight SQL syntax with fully managed Flink & ksqldb.
- Build streaming data pipelines visually using a point-and-click interface with Stream Designer.
- Provides Stream Governance, a suite of fully managed services that work together to manage the availability, integrity, and security of data used across the organization - including Stream Lineage, Stream Catalog, Stream Quality. Data Quality Rules guarantee high-quality streams.

Limited

- Supports self managed Confluent Schema Registry or fully managed Glue Schema Registry that only works with Java.
- Supports Lambda and self-managed ksqldb for stream processing.
- Supports Managed Service for Apache Flink (MSF), but adds complexity and requires additional configuration.
- Custom efforts are required to build your own no-code stream processing solution.
- Does not have lineage or catalog capabilities. Need to rely on free community tools without support or paid third-party tools to provide stream governance. For data quality, lacks broker-side schema validation or Data Quality Rules.

Limited

- Supports self managed Confluent Schema Registry or fully managed Glue Schema Registry that only works with Java.
- Supports Lambda for stream processing.
- Only supports Managed Service for Apache Flink (MSF) for stream processing, but adds complexity and requires additional configuration.
- Custom efforts are required to build your own no-code stream processing solution.
- Does not have lineage or catalog capabilities. Need to rely on free community tools without support or paid third-party tools to provide stream governance. For data quality, lacks broker-side schema validation or Data Quality Rules.

Start Your Cloud Native Kafka Journey with Confluent Cloud Today
confluent.io/cloud-native-kafka