Alibaba Cloud
Message Queue for Apache Kafka

Product Overview

Issue: 20200630

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individual s arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary , incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Document conventions

Style	Description	Example
⛔	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	⛔ **Danger:** Resetting will result in the loss of user configuration data.
⚠	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	⚠ **Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
ⓘ	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	ⓘ **Notice:** If the weight is set to 0, the server no longer receives new requests.
📋	A note indicates supplemental instructions, best practices, tips, and other content.	📋 **Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings** > **Network** > **Set network type**.
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands.	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`

Style	Description	Example
{} or {a|b}	This format is used for a required value, where only one item can be selected.	switch {active|stand}

Contents

1 What is Message Queue for Apache Kafka?

Message Queue for Apache Kafka is a distributed, high-throughput, and scalable message queue service provided by Alibaba Cloud. Message Queue for Apache Kafka is widely used in big data fields such as log collection, monitoring data aggregation, streaming data processing, and online and offline analysis. It has become an indispensable part of the big data ecosystem.

Benefits

Message Queue for Apache Kafka provides fully-managed services for open-source Apache Kafka, solving long-term shortcomings of open-source Apache Kafka. By using Message Queue for Apache Kafka, you can focus on business development, without paying attention to deployment and O&M. Message Queue for Apache Kafka improves your service scalability and reliability with lower service costs.

Application ecosystem

Message Queue for Apache Kafka has a rich application ecosystem in the following three fields:

- Big data: Message Queue for Apache Kafka is widely used in fields such as website behavior analysis, log aggregation, application monitoring, streaming data processing, and online and offline data analysis.
- Data integration: Message Queue for Apache Kafka supports importing messages to offline data warehouses such as MaxCompute, Object Storage Service (OSS), ApsaraDB for RDS, Hadoop, and HBase.
- Stream computing integration: Message Queue for Apache Kafka integrates Realtime Compute, E-MapReduce, Spark, Storm, and other stream computing engines.

Note:

Message Queue for Apache Kafka's extended functions, such as MQTT Proxy, REST Proxy, and KSQL Proxy, are not in public review. They will be available for trial in the future.

2 Architecture

This topic describes the architecture of Message Queue for Apache Kafka and the publish/subscribe pattern.

Architecture of Message Queue for Apache Kafka

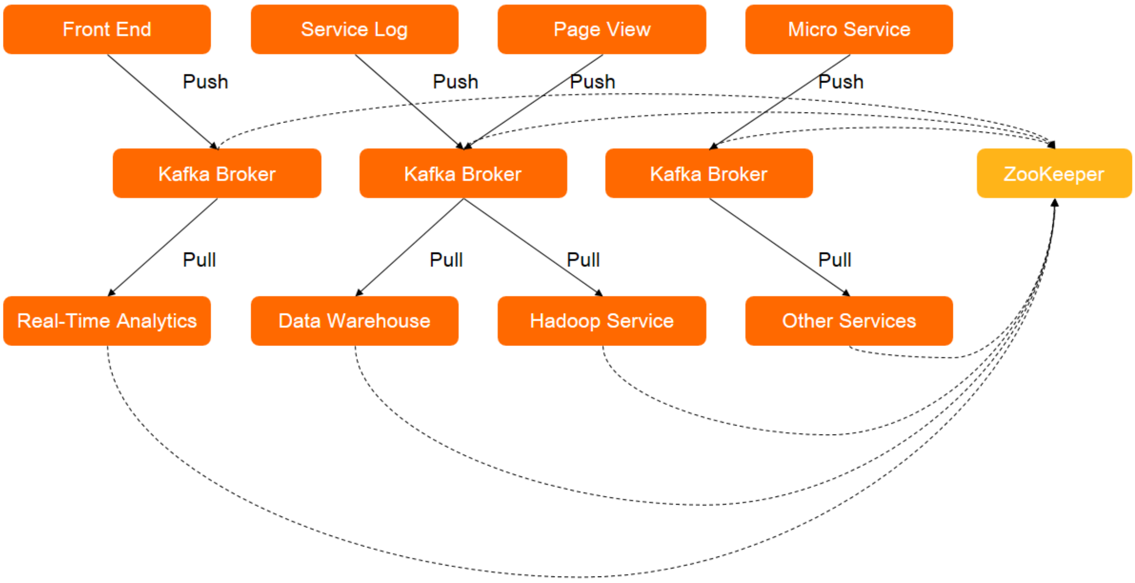
A typical Message Queue for Apache Kafka cluster consists of four parts:

Producer sends messages to the Message Queue for Apache Kafka broker in push mode. The messages sent can be page visits (PVs), server logs, and information related to system resources such as CPU utilization and memory usage.

Broker is a server that stores messages. You can add Message Queue for Apache Kafka brokers as needed. The more Message Queue for Apache Kafka brokers there are, the higher the throughput of the Message Queue for Apache Kafka cluster is.

Consumer group subscribes to and consumes messages from the Message Queue for Apache Kafka broker in pull mode.

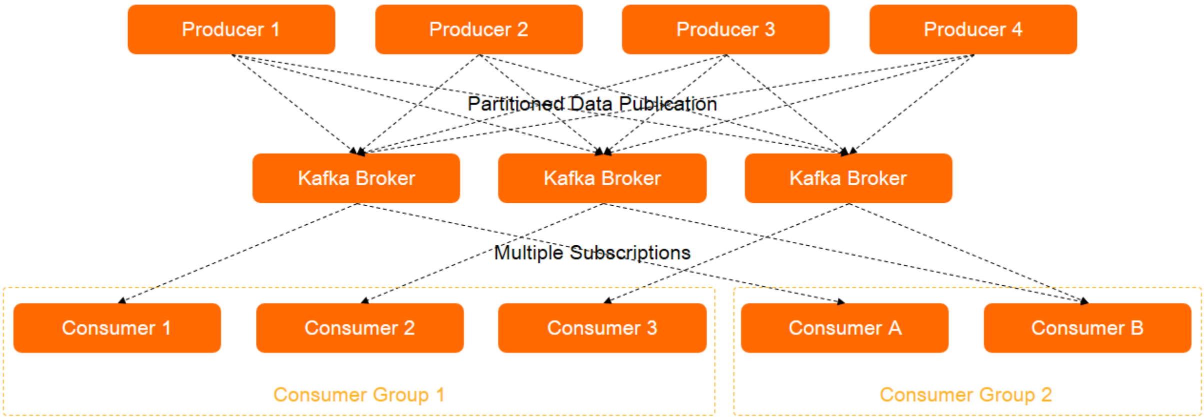
ZooKeeper manages the cluster configuration, elects the leader partition, and balances the load when the consumer group changes.



The publish/subscribe pattern of Message Queue for Apache Kafka

Message Queue for Apache Kafka uses a publish/subscribe pattern:

- The relationship between consumer groups and topics is N:N. One consumer group can subscribe to multiple topics, and one topic can be subscribed to by multiple consumer groups.
- One message under a topic can be subscribed to by multiple consumer groups but can be consumed only by one consumer in the same consumer group.



3 Benefits

This topic describes benefits of Message Queue for Apache Kafka compared with the open-source Apache Kafka.

Out-of-the-box

- Message Queue for Apache Kafka is completely compatible with Apache Kafka, that is, open-source Apache Kafka clients can communicate with Message Queue for Apache Kafka. Currently, Message Queue for Apache Kafka supports open-source Apache Kafka of versions 0.9.0 to 2.x. Later versions will be supported in the future.
- You can migrate your business systems that are developed based on existing open-source Apache Kafka ecosystems to Alibaba Cloud Message Queue for Apache Kafka without any modification.

Fully managed service

Alibaba has an experienced and professional O&M team and mature O&M system for Message Queue for Apache Kafka.

- HouseKeeping: checks the running status of the core links of Message Queue for Apache Kafka. It scans and diagnoses the whole cluster every minute and generates alerts for unhealthy instances. It also provides an overall inspection report every day for the O&M personnel to monitor the health status of Message Queue for Apache Kafka.
- Service monitoring and alerting: You can enable monitoring and alerting for message accumulation in each consumer group, helping you discover problems promptly.
- API: Message Queue for Apache Kafka provides a complete set of control-type API operations for you to implement a series of resource management and O&M functions.

High availability

The R&D and performance optimization team of Alibaba Cloud message products solves the long-term shortcomings of open-source Apache Kafka and provides you with better services.

- High reliability and high availability: With support from the professional team, Message Queue for Apache Kafka is highly available and allows messages to be persisted to message queues, achieving data reliability of up to 99.999999% and service availability of up to 99.9%.

- Massive message accumulation: The Message Queue for Apache Kafka cluster can always maintain a high throughput when a large number of messages are accumulated.
- Tens of thousands of topics: Message Queue for Apache Kafka supports concurrent read/write operations on tens of thousands of topics while maintaining a high throughput of the Message Queue for Apache Kafka cluster.

Data security

Message Queue for Apache Kafka provides functions such as authentication and authorization, Alibaba Cloud accounts and RAM users, and enterprise-level security protection.

- Alibaba Cloud accounts and RAM users: Message Queue for Apache Kafka provides functions such as Alibaba Cloud accounts, RAM users, blacklist, whitelist, and Security Token Service (STS), and supports authorization of Alibaba Cloud accounts, RAM users, and accounts between enterprises.
- Access security: Based on the Alibaba Cloud account system, Message Queue for Apache Kafka authenticates the user identity by using the Simple Authentication and Security Layer (SASL) mechanism and encrypts the channel based on SSL to protect your data from being stolen or tampered with during transmission.
- Alibaba Cloud VPC: You can access Message Queue for Apache Kafka from Alibaba Cloud Virtual Private Cloud (VPC).

Elastic computing

You can scale up the service as needed, without being perceived by the upper-layer services.

- Cluster scale-out: Brokers can be scaled out across zones (data centers).
- Partitioning: Message Queue for Apache Kafka supports quick scaling of tens of thousands of topics and unlimited queues.

4 Scenarios

This topic describes classic scenarios of Message Queue for Apache Kafka.

Website activity tracking

Successful website operations require close tracking and analysis of the user behavior of the website. By using Message Queue for Apache Kafka, you can collect website activity data in real time, including user behaviors such as webpage browsing and searches, and implement the following functions based on the publish/subscribe pattern:

- Publish messages to different topics based on the type of the business data.
- Through real-time delivery of subscribed messages, use message streams for real-time monitoring and service analysis, or load the message streams to offline data warehouse systems such as Hadoop and MaxCompute for offline processing and service reporting.

Benefits:

- High throughput: High throughput is required to support the large amount of behavior information generated by all users on the website.
- Elastic scaling: The website activity causes a sharp increase in behavior data, and the cloud platform can be quickly scaled out on demand.
- Big data analysis: Message Queue for Apache Kafka can connect to real-time stream computing engines such as Storm and Spark, as well as offline data warehouse systems such as Hadoop and MaxCompute.

Log aggregation

Many platforms, such as Taobao and Tmall, generate a large number of logs every day, which are usually streaming data, such as page visits (PVs) and query records of search engines. Compared with log-oriented systems such as Scribe and Flume, Message Queue for Apache Kafka features higher efficiency, longer data persistence, and shorter end-to-end response times (RT). With these benefits, Message Queue for Apache Kafka is suitable for log collection:

- In Message Queue for Apache Kafka, file details are ignored and logs of multiple hosts or applications are abstracted as log or event message streams and then asynchronously sent to the Message Queue for Apache Kafka cluster, greatly reducing the RT.

- The Message Queue for Apache Kafka client submits and compresses messages in batches, without increasing the performance overhead of the producer.
- Consumers can use offline warehouse systems such as Hadoop and MaxCompute and real-time online analysis systems such as Storm and Spark to perform statistical analysis of logs.

Benefits:

- Application and analysis decoupling: Message Queue for Apache Kafka serves as a bridge between an application system and an analysis system and decouples the two systems.
- High scalability: Message Queue for Apache Kafka is scalable. When the data size increases, you can add nodes to quickly scale out your application.
- Online and offline analysis systems: Message Queue for Apache Kafka supports real-time online analysis systems and offline analysis systems such as Hadoop.

Stream computing

In many fields, such as stock market trend analysis, meteorological data monitoring and control, and website user behavior analysis, due to the huge amount of data generated in real time, it is difficult to collect and store all the data in the database before processing it. Therefore, traditional data processing architectures cannot meet user needs.

Different from traditional architectures, Message Queue for Apache Kafka and stream computing engines such as Storm, Samza, and Spark can efficiently solve the preceding problems. The stream computing model captures and processes data in real time during data flow, computes and analyzes the data based on service requirements, and then saves or distributes the results to relevant components.

Benefits:

- Data flow: Message Queue for Apache Kafka serves as a bridge between an application system and an analysis system and decouples the two systems.
- High scalability: Message Queue for Apache Kafka is highly scalable to cope with the huge amount of data generated in real time.
- Stream computing engine: Message Queue for Apache Kafka can connect to open-source Storm, Samza, and Spark, and Alibaba Cloud products such as E-MapReduce, Blink, and Realtime Compute.

Data transfer hub

Over the past 10 years, dedicated systems such as key-value storage (HBase), search (Elasticsearch), stream processing (Storm, Spark Streaming, and Samza), and time series database (OpenTSDB) have emerged. These systems are designed for single problems, making it easy and cost-effective to build distributed systems on commercial hardware.

Generally, the same dataset needs to be injected into multiple dedicated systems. For example, when application logs are used for offline analysis, searching for a single log is also required. However, it is impractical to construct independent workflows to collect data of each type and then import the data to their own dedicated systems. In this case, you can use Message Queue for Apache Kafka as a data transfer hub to import the same data record to different dedicated systems.

Benefits:

- High-capacity storage: Message Queue for Apache Kafka can store a large amount of data on commercial hardware and implement a horizontally scalable distributed system.
- One-to-many consumption model: Based on the publish/subscribe pattern, the same dataset can be consumed multiple times.
- Real-time and batch processing: Message Queue for Apache Kafka supports local data persistence and page cache, and transmits messages to consumers for real-time and batch processing at the same time without performance loss.

5 Terms

This topic describes related terms of Message Queue for Apache Kafka to help you better understand and use Message Queue for Apache Kafka.

B

Broker An independent server in the Message Queue for Apache Kafka cluster.

C

Cloud storage A way to store messages. The underlying layer is connected to multiple replicas of Alibaba Cloud disks. In Message Queue for Apache Kafka, each partition only needs one replica.

Consumer The message subscriber, also known as the message consumer. It reads messages from the Message Queue for Apache Kafka broker and consumes the messages.

Consumer group A group of consumers that subscribe to and consume messages of the same type based on the same consumption logic. The relationship between consumer groups and topics is N:N. One consumer group can subscribe to multiple topics, and one topic can be subscribed to by multiple consumer groups.

L

Local storage A method to store messages. Based on the In-Sync Replicas (ISR) algorithm of Apache Kafka, three replicas are stored in distributed mode and `min.insync.replicas` is set to 2.

N

Normal messages By default, messages of the same key are stored in the same partition in the order they are sent. When an instance in the cluster fails, the messages may be out of order.

P

Partition A physical partition. One topic contains one or more partitions.

Partitionally ordered messages By default, messages of the same key are stored in the same partition in the order they are sent. When an instance in the cluster fails, the messages are still in order. However, some messages in the partition cannot be sent until the failed instance is restored.

Producer The message publisher, also known as the message producer. It generates and sends messages to the Message Queue for Apache Kafka broker.

T

Topic The type of the message. Message Queue for Apache Kafka classifies messages by topic. A topic consists of one or more partitions and is stored on one or more brokers.

6 Limits

Message Queue for Apache Kafka sets constraints for some metrics. To avoid program exceptions when using Message Queue for Apache Kafka, do not exceed the maximum limits. For more information about the maximum limits of relevant items, see the following table.

Item	Limit	Description
Automatic creation of topics and consumer groups	Not supported	Automatic creation of topics and consumer groups facilitates usage but increases the O&M difficulty and easily causes system instability. In Message Queue for Apache Kafka, topics and consumer groups must be authenticated. Therefore, Message Queue for Apache Kafka does not automatically create topics or consumer groups. However, you can create topics and consumer groups by using the console, API operations, or automated orchestration tools. • In the console: - Create a topic - Create a consumer group • By using API operations: - #unique_10 - #unique_11 • By using Terraform: - CreateTopic - CreateConsumerGroup

Item	Limit	Description
Total topic (partition) quantity limit	Supported	In Message Queue for Apache Kafka, messages are stored and scheduled by partition. If messages are stored in too many topics (partitions), serious storage fragmentation occurs, which greatly reduces the cluster performance and stability.
Reduction in the number of partitions of a topic	Not supported	This is caused by design constraints of Message Queue for Apache Kafka.
ZooKeeper exposure	Not supported	ZooKeeper has been masked since Apache Kafka 0.9. Therefore, you do not need to access ZooKeeper to use the client. ZooKeeper in Message Queue for Apache Kafka is partially shared and is not exposed for security purposes. You do not need to learn about ZooKeeper.
Topic-based authentication	Not supported	Topic-based authentication depends on ZooKeeper. Message Queue for Apache Kafka is deployed in Virtual Private Clouds (VPCs) and provides sufficient security protection through the security group and whitelist features.
Logon to the node in which your Message Queue for Apache Kafka instance is deployed	Not supported	None

Item	Limit	Description
Version	• Message Queue for Apache Kafka Standard Edition Only version 0.10.x is supported and deployed by default. • Message Queue for Apache Kafka Professional Edition Versions 0.10.x to 2.x are supported. Version 0.10.x is deployed by default.	• Version 2.x is compatible with versions 0.10.x and 0 .9.0. • Version 0.10.x is compatible with version 0 .9.0. • To upgrade a Standard Edition instance from 0.10.x to 2.x, you must first upgrade the instance to the Professional Edition and then upgrade the open-source version of the instance to 2.x. For more information, see. #unique_12 and #unique_13. • To upgrade a Professional Edition instance from 0.10.x to 2.x, see #unique_13.
Consumer group count	The consumer group count is twice the topic count.	For example, if an instance has 50 topics, you can create up to 100 consumer groups in this instance. To increase the number of consumer groups, you can increase the number of topics. The number of consumer groups is increased by two each time a topic is added. For more information, see #unique_12.

Item	Limit	Description
Relationship between the number of topics and that of partitions	1:16	In addition to the default number of partitions, 16 partitions are added for each additional topic. For example, if you purchase an instance with 50 topics , 20 MB/s peak traffic, and 400 partitions by default, after you add 10 topics, the number of partitions of this instance is increased by 160 , and the total number of partitions becomes 560.
Number of topics of a Professional Edition instance	The number of topics is twice that of your purchased topics.	For example, if you purchase an instance with 50 topics , the actual number of topics you can create in the instance is 100.
Change of the region or network properties of an instance	Not supported	After an instance is purchased and deployed , its region and network properties are closely integrated with its physical resources and cannot be changed. To change the region or network properties of an instance, release the instance and purchase a new instance.
Message size	10 MB	The maximum size of a message is 10 MB. A message larger than 10 MB cannot be sent.

7 Performance comparison between Message Queue for Apache Kafka and Apache Kafka

This topic compares Message Queue for Apache Kafka and open-source Apache Kafka in terms of stability, kernel capability, and governance capability.

Stability

Item	**Message Queue for Apache Kafka**	**Apache Kafka**
Disk usage	Deletes old data when no free disk space is available.	Experiences downtime when no free disk space is available.
Thread pool isolation	Ensures normal data write while reading cold data.	Experiences thread blocking while reading cold data, which causes frequent data write failures.
Partition size	Writes data to tens of thousands of partitions in a stable manner.	Experiences frequent jitter when writing data to thousands of partitions.
Inspection system	Automatically detects and fixes deadlocks and breakdowns.	None
Bug fixing	Detects and fixes bugs in a timely manner.	Waits for the community to release new versions to fix bugs, which takes a long time.

Kernel capabilities

Item	**Message Queue for Apache Kafka**	**Apache Kafka**
Elastic scaling	Supports elastic scaling in seconds, without being perceived by the upper-layer services.	Supports elastic scaling in hours, which affects clusters due to increased traffic from replication.

Item	Message Queue for Apache Kafka	Apache Kafka
Storage cost	Provides highly reliable cloud storage in the Professional Edition to save a lot of storage space.	Provides triplicate storage to ensure availability and reliability, which imposes a heavy load on storage.

Governance capabilities

Item	Message Queue for Apache Kafka	Apache Kafka
Version upgrade	Supports one-click upgrade.	Supports manual upgrade, which is prone to errors.
Metrics curve	Displays a complete metrics curve to facilitate traffic tracing and troubleshooting.	Displays only real-time metrics. Historical data is difficult to access.
Message accumulation alerts	Triggers timely alerts on message accumulation.	None
Subscription relationship	Provides a complete subscription relationship.	Provides a relatively brief subscription relationship.
Partition status	Displays a complete partition status diagram.	Displays a relatively brief partition status diagram.
Message sending	Allows you to send messages directly from the console.	Allows you to send messages only through the command-line interface, which results in high costs.
Message query	Allows you to directly view messages by time or offset in the console.	Allows you to consume messages through the command-line interface, but you cannot view messages by time or offset.

8 Endpoint comparison

This topic compares three types of Message Queue for Apache Kafka endpoints to help you choose an appropriate access method.

Item	Default endpoint	SASL endpoint	
Instance type	• Standard Edition • Professional Edition	Professional Edition	
Open-source instance version	0.10.x to 2.x	2.x	
Network type	VPC	VPC	
Activation method	Automatic	Submit a ticket	
Security protocol	PLAINTEXT	SASL_PLAINTEXT	
Port number	9092	9094	