

# Distributed Message Service for Kafka

## Visão geral de serviço

Edição 01  
Data 2024-05-07



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. Todos os direitos reservados.**

Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio sem consentimento prévio por escrito da Huawei Cloud Computing Technologies Co., Ltd.

## **Marcas registadas e permissões**



HUAWEI e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd.

Todas as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

## **Aviso**

Os produtos, os serviços e as funcionalidades adquiridos são estipulados pelo contrato estabelecido entre a Huawei Cloud e o cliente. Os produtos, os serviços e as funcionalidades descritos neste documento, no todo ou em parte, podem não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÃO" sem garantias ou representações de qualquer tipo, sejam expressas ou implícitas.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

---

# Índice

---

<b>1 Infográficos.....</b>	<b>1</b>
<b>2 O que é DMS for Kafka?.....</b>	<b>3</b>
<b>3 Vantagens do produto.....</b>	<b>4</b>
<b>4 Cenários de aplicação.....</b>	<b>6</b>
<b>5 Especificações.....</b>	<b>9</b>
<b>6 Comparação entre Kafka, RabbitMQ e RocketMQ.....</b>	<b>14</b>
<b>7 Comparação entre DMS for Kafka e Kafka de código aberto.....</b>	<b>16</b>
<b>8 Segurança.....</b>	<b>19</b>
8.1 Responsabilidades compartilhadas.....	19
8.2 Autenticação de identidade e controle de acesso.....	20
8.3 Proteção de dados.....	20
8.4 Auditoria e logs.....	21
8.5 Resiliência.....	22
8.6 Monitoramento de riscos de segurança.....	22
8.7 Certificados.....	23
<b>9 Observações e restrições.....</b>	<b>24</b>
<b>10 Serviços relacionados.....</b>	<b>28</b>
<b>11 Conceitos básicos.....</b>	<b>30</b>
<b>12 Gerenciamento de permissões.....</b>	<b>32</b>

# 1 Infográficos

---

### 1 What is Message-oriented Middleware?

Message-oriented middleware (MOM) routes, stores, and sends messages between distributed systems. Applications communicate with each other by sending messages from one or more message producers to message consumers.

Take online shopping as an example. The order system processes orders and sends the order information as messages to Kafka. Kafka then distributes the order messages to downstream subsystems (such as logistics, warehouse, and distribution systems). The order system is decoupled from downstream subsystems because they communicate through Kafka for reliability.

### 2 Popular MOM

#### DMS for Kafka

Distributed Message Service (DMS) for Kafka is a centrally managed service that delivers high reliability, scalability, and flexibility to meet time-critical transactions, stream data processing, system decoupling, and Kafka forwarding use cases.

#### DMS for RabbitMQ

DMS for RabbitMQ is compatible with open-source RabbitMQ. It provides message prioritization, transactional messages, message acking, dead letter queues, and priority queues for fault cases. For control and system decoupling scenarios.

#### DMS for RocketMQ

DMS for RocketMQ is compatible with open-source RocketMQ. It provides ordered message delivery, improved delivery rates, message retry, and batch messages. Transactional messages, and system messages for a controller and broker scenarios.

### 3 Comparing Kafka, RabbitMQ, and RocketMQ

There is no best strategy for the best choice. Each choice should take its own advantages and disadvantages. We have summarized their differences in performance, reliability, functions, and usability.

#### Performance

Metric	Kafka	RabbitMQ	RocketMQ
Throughput	High	Low	High
Latency	Low	High	Low
Message Size	Large	Small	Large
Queue Size	Large	Small	Large

#### Data Reliability

Feature	Kafka	RabbitMQ	RocketMQ
High reliability for high availability	Yes	No	Yes
Transactional messages	No	Yes	Yes
Message acking	No	Yes	Yes
Dead letter queues	No	Yes	Yes
Priority queues	No	Yes	Yes

#### Functions

Function	Kafka	RabbitMQ	RocketMQ
Message Retention	Yes	No	Yes
Message Retention Time	Yes	No	Yes
Message Retention Size	Yes	No	Yes
Message Retention Location	Yes	No	Yes
Message Retention Policy	Yes	No	Yes

#### Message Retention

Feature	Kafka	RabbitMQ	RocketMQ
Message Retention	Yes	No	Yes
Message Retention Time	Yes	No	Yes
Message Retention Size	Yes	No	Yes
Message Retention Location	Yes	No	Yes
Message Retention Policy	Yes	No	Yes

#### Client Support

Client	Kafka	RabbitMQ	RocketMQ
Java	Yes	Yes	Yes
Python	Yes	Yes	Yes
Go	Yes	Yes	Yes
C++	Yes	Yes	Yes
PHP	Yes	Yes	Yes

#### Other

Feature	Kafka	RabbitMQ	RocketMQ
Message Retention	Yes	No	Yes
Message Retention Time	Yes	No	Yes
Message Retention Size	Yes	No	Yes
Message Retention Location	Yes	No	Yes
Message Retention Policy	Yes	No	Yes

#### Service Availability

Feature	Kafka	RabbitMQ	RocketMQ
High availability	Yes	No	Yes
High reliability	Yes	No	Yes
High scalability	Yes	No	Yes
High flexibility	Yes	No	Yes

#### Summary

Kafka is a distributed system that provides high throughput, high reliability for collecting and delivering large volumes of data and supports real-time analytics.

RabbitMQ is developed in Erlang, which is renowned for its reliability, scalability, and performance. It is suitable for high-throughput scenarios, such as distributed systems, data processing, and real-time analytics.

RocketMQ is developed in Java. It is suitable for scenarios that require high reliability and high throughput, such as distributed systems, data processing, and real-time analytics.

# 2 O que é DMS for Kafka?

---

O Apache Kafka é um middleware de mensagens distribuído que apresenta alta taxa de transferência, persistência de dados, escalabilidade horizontal e processamento de dados de fluxo. Ele adota o padrão de publicação-assinatura e é amplamente utilizado para coleta de logs, fluxo de dados, análise de sistemas on-line/off-line e monitoramento em tempo real.

Distributed Message Service (DMS) for Kafka é um serviço de enfileiramento de mensagens baseado no Apache Kafka. Este serviço fornece instâncias premium do Kafka. Os recursos de computação, armazenamento e largura de banda usados por uma instância são ocupados exclusivamente pelo usuário. Você pode solicitar instâncias conforme necessário. As instâncias podem ser usadas imediatamente, tirando a implementação e a pressão de O&M para que você possa se concentrar no desenvolvimento de seus serviços.

## Guia dos leitores

Esta documentação apresenta o DMS for Kafka e suas diferenças em relação ao Apache Kafka. Você aprenderá as informações detalhadas sobre as especificações, as operações do console, a chamada de API e o acesso do cliente às instâncias do DMS for Kafka da Huawei Cloud.

Para obter mais informações sobre o conhecimento básico do Kafka ou detalhes técnicos sobre como criar e recuperar mensagens, acesse o [site oficial do Apache Kafka](#).

# 3 Vantagens do produto

---

DMS for Kafka da Huawei Cloud fornece enfileiramento de mensagens fácil de usar baseado no Apache Kafka. Os serviços podem ser migrados rapidamente para a nuvem sem qualquer alteração, reduzindo os custos de manutenção e uso.

- **Implementação rápida**

Basta definir as informações da instância no console do DMS for Kafka, enviar seu pedido e uma instância completa do Kafka será criada e implementada automaticamente.

- **Migração de serviço sem modificações**

O DMS for Kafka é compatível com APIs do Kafka de código aberto e suporta todas as funções de processamento de mensagens do Kafka de código aberto.

Se seus serviços de aplicação forem desenvolvidos com base no Kafka de código aberto, você poderá migrá-los facilmente para DMS for Kafka da Huawei Cloud depois de especificar algumas configurações de autenticação.

#### **NOTA**

As instâncias do Kafka são compatíveis com o Apache Kafka v1.1.0 e v2.7. Mantenha as versões de cliente e de servidor iguais.

- **Segurança**

As operações em instâncias do Kafka são registradas e podem ser auditadas. As mensagens podem ser criptografadas antes do armazenamento.

Além da autenticação Simple Authentication and Security Layer (SASL), Virtual Private Clouds (VPCs) e os grupos de segurança também fornecem controles de segurança no acesso à rede.

- **Confiabilidade de dados**

As instâncias do Kafka suportam persistência e replicação de dados. As mensagens podem ser replicadas de forma síncrona ou assíncrona entre réplicas e descarregadas em disco.

- **Alta disponibilidade**

O Kafka é executado em clusters, permitindo failover e tolerância a falhas para que os serviços possam ser executados sem problemas.

Os brokers de instância de Kafka podem ser implementados em AZs para aumentar a disponibilidade do serviço. Os dados são sincronizados entre diferentes AZs com base no mecanismo de réplica em sincronização (ISR) do Kafka. Um tópico deve ter várias

cópias de dados e distribuí-las entre ISRs. Quando a replicação de ISR é normal, o objetivo de ponto de recuperação (RPO) está próximo de 0.

- O&M simples

Huawei Cloud oferece um conjunto completo de serviços de monitoramento e alarme, eliminando a necessidade de atendimento 24/7. As métricas de instância do Kafka são monitoradas e relatadas, incluindo o número de partições, tópicos e mensagens acumuladas. Você pode configurar regras de alarme e receber notificações por SMS ou e-mail sobre como seus serviços estão sendo executados em tempo real.

- Acumulação e dimensionamento maciços

O Kafka possui alta escalabilidade porque é executado em um sistema distribuído ou cluster. Você pode configurar até 200 partições para um tópico. O espaço de armazenamento também pode ser expandido. Isso significa que bilhões de mensagens podem ser acumuladas, adequadas para cenários que exigem alta simultaneidade, alto desempenho e acesso em larga escala.

- Especificações flexíveis

Você pode personalizar a largura de banda e o espaço de armazenamento para a instância e o número de partições e réplicas para tópicos na instância.



# 4 Cenários de aplicação

O Kafka é um middleware popular orientado a mensagens que apresenta entrega de mensagens altamente confiável e assíncrona. É amplamente utilizado para transmitir dados entre diferentes sistemas nas muitas indústrias, incluindo aplicativos corporativos, pagamento, telecomunicações, comércio eletrônico, redes sociais, mensagens instantâneas, vídeo, Internet das Coisas e Internet das Indústrias de Veículos.

## Comunicação assíncrona

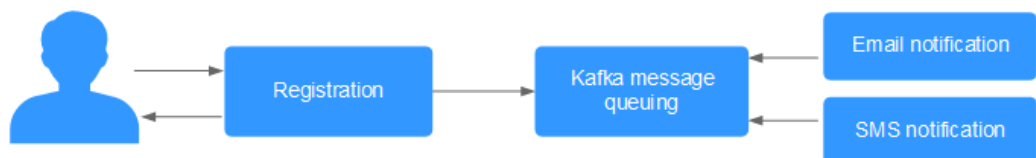
Mensagens não centrais ou menos importantes são enviadas de forma assíncrona para os sistemas receptores, de modo que o processo principal do serviço não seja mantido à espera dos resultados de outros sistemas, permitindo respostas mais rápidas.

Por exemplo, o Kafka pode ser usado para enviar um e-mail de notificação e uma mensagem SMS depois que um usuário se registrou em um site, fornecendo respostas rápidas durante todo o processo de registro.

**Figura 4-1** Registro e notificação em série



**Figura 4-2** Registro assíncrono e notificação usando filas de mensagens



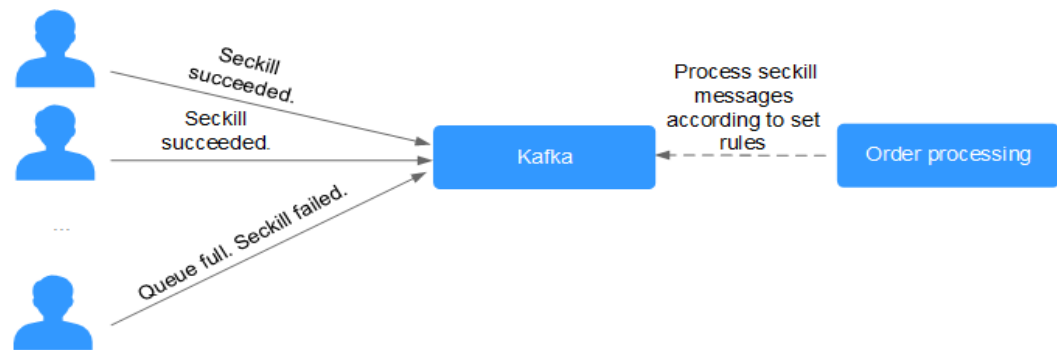
## Controle de tráfego

Em sistemas de comércio eletrônico ou sites de grande escala, há uma lacuna de capacidade de processamento entre os sistemas upstream e downstream. Picos de tráfego de sistemas upstream com altas capacidades de processamento podem ter um grande impacto em sistemas downstream com menores capacidades de processamento. Por exemplo, as promoções de vendas on-line envolvem uma enorme quantidade de tráfego inundando os sistemas de

comércio eletrônico. O Kafka fornece um buffer de três dias por padrão para centenas de milhões de mensagens, como pedidos e outras informações. Desta forma, os sistemas de consumo de mensagens podem processar as mensagens durante os períodos fora de pico.

Além disso, as rajadas de tráfego de venda flash originadas de sistemas front-end podem ser tratadas com o Kafka, impedindo que os sistemas back-end falhem.

**Figura 4-3** Tratamento de intermitência de tráfego usando Kafka



## Sincronização de log

Em sistemas de serviços de larga escala, os logs de diferentes aplicativos são coletados para solução rápida de problemas, rastreamento de link completo e monitoramento em tempo real.

Kafka foi originalmente projetado para esse cenário. Os aplicativos enviam de forma assíncrona mensagens de log para filas de mensagens por canais de transmissão confiáveis. Outros componentes podem ler as mensagens de log das filas de mensagens para análise posterior, em tempo real ou off-line. Além disso, o Kafka pode coletar informações de registro de chaves para monitorar aplicativos.

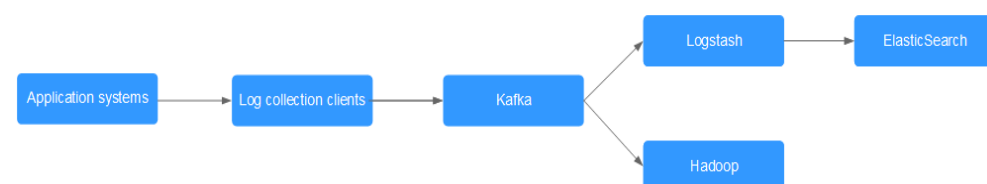
A sincronização de log envolve três componentes principais: clientes de coleta de log, Kafka e aplicativos de processamento de log de back-end.

1. Os clientes de coleta de log coletam dados de log de um serviço de aplicativo de usuário e enviam os dados de log de forma assíncrona em lotes para clientes do Kafka.

Os clientes do Kafka recebem e compactam mensagens em lotes. Isso tem apenas um pequeno impacto no desempenho do serviço.

2. Kafka persiste logs.
3. Aplicativos de processamento de log, como o Logstash, assinam mensagens no Kafka e recuperam mensagens de log do Kafka. Em seguida, as mensagens são pesquisadas por serviços de pesquisa de arquivos ou entregues a aplicativos de big data, como o Hadoop, para armazenamento e análise.

**Figura 4-4** Processo de sincronização de log



 **NOTA**

O Logstash é para análise de log, o Elasticsearch é para pesquisa de log e o Hadoop é para análise de big data. São todas ferramentas de código aberto.

# 5 Especificações

## Especificações da instância do Kafka

As instâncias do Kafka são compatíveis com o Kafka de código aberto Kafka v1.1.0 e v2.7. As especificações da instância são classificadas com base na variante ECS, incluindo kafka.2u4g.cluster.small, kafka.2u4g.cluster, kafka.4u8g.cluster, kafka.8u16g.cluster, kafka.12u24g.cluster e kafka.16u32g.cluster.

### NOTA

Na tabela a seguir, as transações por segundo (TPS) são calculadas assumindo que o tamanho de uma mensagem é de 1 KB. Para obter detalhes sobre o desempenho do TPS, consulte [Cenários de teste](#).

**Tabela 5-1** Especificações de instância do Kafka

Flavor	Brokers	Máximo de TPS por broker	Máximo de partições por broker	Máximo de grupos de consumidores por broker	Máximo de conexões de clientes por broker	Espaço de armazenamento	Tráfego por broker (MB/s)
kafka.2u4g.cluster.small	3–30	20.000	100	15	2000	300 GB–300.000 GB	40
kafka.2u4g.cluster	3–30	30.000	250	20	2000	300 GB–300.000 GB	100
kafka.4u8g.cluster	3–30	100.000	500	100	4000	300 GB–600.000 GB	200
kafka.8u16g.cluster	3–50	150.000	1000	150	4000	300 GB–1.500.000 GB	250

Flavor	Brokers	Máximo de TPS por broker	Máximo de partições por broker	Máximo de grupos de consumidores por broker	Máximo de conexões de clientes por broker	Espaço de armazenamento	Tráfego por broker (MB/s)
kafka.12u 24g.cluster	3-50	200.000	1500	200	4000	300 GB- 1.500.000 GB	375
kafka.16u 32g.cluster	3-50	250.000	2000	200	4000	300 GB- 1.500.000 GB	500

## Especificações de instância e largura de banda de rede

A largura de banda de rede de uma instância do Kafka consiste no seguinte:

1. Largura de banda da rede usada pelos brokers da instância
2. Largura de banda do disco usado pelos brokers da instância. Para obter detalhes, consulte [Tipos de disco e desempenho](#).

Observação:

- Por padrão, os testes do Kafka são executados no cenário de leitura traseira (ou seja, apenas os dados de produção mais recentes são consumidos) em vez do cenário de leitura a frio (ou seja, os dados históricos são consumidos desde o início).
- A largura de banda de uma instância com um flavor antigo (como 100 MB/s) é a largura de banda total da rede de todos os brokers da instância.

**O cálculo de tráfego de instâncias com novos flavors (como kafka.2u4g.cluster) é descrito da seguinte forma:**

- A taxa de leitura/gravação é de 1:1.
- O número padrão de réplicas de tópicos é 3.
- Total de tráfego de rede = tráfego por broker x quantidade de broker
- Tráfego total de instâncias = tráfego de serviço + tráfego de replicação de dados entre brokers

Suponha que o flavor atual seja kafka.2u4g.cluster, o tráfego por broker seja 100 MB/s e o número de brokers seja 3. Qual é o tráfego total da rede, o tráfego máximo de leitura e o tráfego máximo de gravação da instância?

1. Tráfego total da rede = tráfego por broker x quantidade de broker = 100 MB/s x 3 = 300 MB/s
2. Tráfego máximo de leitura = tráfego de rede total da instância/número padrão de réplicas/2 = 300 MB/s/3/2 = 50 MB/s

- Tráfego máximo de gravação = tráfego de rede total da instância/número padrão de réplicas/2 = 300 MB/s/3/2 = 50 MB/s

## Mapeamento entre sabores anteriores e novos

**Tabela 5-2** compara os sabores de instância anteriores e novos do Kafka a preços semelhantes.

**Tabela 5-2** Mapeamento entre sabores de instância anteriores e novos do Kafka

Flavor anterior			Flavor novo		
Especificações	Máximo de partições	Tráfego de rede de instância total	Especificações	Máximo de partições	Tráfego de rede de instância total
100 MB/s	300	100 MB/s	kafka.2u4g.cluster x 3	750	300 MB/s
300 MB/s	900	300 MB/s	kafka.4u8g.cluster x 3	1500	600 MB/s
600 MB/s	1800	600 MB/s	kafka.8u16g.cluster x 4	4000	1000 MB/s
1200 MB/s	1800	1200 MB/s	kafka.8u16g.cluster x 8	8000	2000 MB/s

As seguintes conclusões podem ser tiradas da tabela anterior:

- Novos sabores proporcionam melhor desempenho ao mesmo preço.
- Sabores anteriores usam recursos não exclusivos. Se a carga for pesada, ocorrerão conflitos de recursos. Em contraste, os novos sabores usam recursos exclusivos para que eles forneçam melhor desempenho e custo-benefício.
- Instâncias com novos sabores suportam as funções mais recentes. Por exemplo, o flavor do broker pode ser ajustado dinamicamente para lidar melhor com mudanças complexas de serviço.

## Seleção de flavor

- kafka.2u4g.cluster.small com 3 brokers  
 Recomendado para até 6000 conexões de clientes, 45 grupos de consumidores e 60.000 TPS
- kafka.2u4g.cluster com 3 brokers  
 Recomendado para até 6000 conexões de clientes, 60 grupos de consumidores e 90.000 TPS
- kafka.4u8g.cluster com 3 brokers  
 Recomendado para até 12.000 conexões de clientes, 300 grupos de consumidores e 300.000 TPS

- kafka.8u16g.cluster com 3 brokers  
Recomendado para até 12.000 conexões de clientes, 450 grupos de consumidores e 450.000 TPS
- kafka.12u24g.cluster com 3 brokers  
Recomendado para até 12.000 conexões de clientes, 600 grupos de consumidores e 600.000 TPS
- kafka.16u32g.cluster com 3 brokers  
Recomendado para até 12.000 conexões de clientes, 600 grupos de consumidores e 750.000 TPS

## Seleção de espaço de armazenamento

As instâncias do Kafka suportam o armazenamento de múltiplas réplicas. O espaço de armazenamento é consumido por todas as réplicas. Ao criar uma instância, especifique seu espaço de armazenamento com base no tamanho esperado da mensagem de serviço e no número de réplicas.

Por exemplo, se o tamanho estimado da mensagem for de 100 GB, a capacidade do disco deverá ser pelo menos: 100 GB x número de réplicas + 100 GB (espaço reservado).

O espaço de armazenamento pode ser expandido à medida que seu serviço cresce.

## Quantidade de tópico

Há limites na quantidade de tópicos e no número agregado de partições nos tópicos. Quando o limite de quantidade de partição é atingido, você não pode mais criar tópicos.

O número de tópicos está relacionado ao número máximo de partições permitidas (consulte [Figura 5-1](#)) e ao número especificado de partições em cada tópico (consulte [Tabela 5-1](#)).

**Figura 5-1** Definir o número de partições

### Create Topic

Topic Name

Partitions  Value range: 1 to 200  
**Cannot be decreased once the topic is created.**

Replicas  Value range: 1 to 3  
**Cannot be greater than the broker quantity.**

Aging Time (h)  Value range: 1 to 720  
How long messages will be preserved in the topic. Messages older than this period will be deleted and cannot be consumed.

Synchronous Replication

Synchronous Flushing

Message Timestamp

Max.Message Size (bytes)

Description   
0/200

**O número máximo de partições permitidas para uma instância com kafka.2u4g.cluster e 3 brokers é 750.**

- Se o número de partições de cada tópico na instância for 3, o número máximo de tópicos será  $750/3 = 250$ .
- Se o número de partições de cada tópico na instância for 1, o número máximo de tópicos será  $750/1 = 750$ .



# 6 Comparação entre Kafka, RabbitMQ e RocketMQ

Recurso	RocketMQ	Kafka	RabbitMQ
Fila de prioridade	Não compatível	Não compatível	Compatível. Recomenda-se que a prioridade seja definida como 0–10.
Fila atrasada	Compatível	Não compatível	Compatível
Fila de mensagens mortas	Compatível	Não compatível	Compatível
Repetição de mensagem	Compatível	Não compatível	Não compatível
Modo de recuperação	Baseado em pull e baseado em push	Baseado em pull	Baseado em pull e baseado em push
Transmissão de mensagens	Compatível	Compatível	Compatível
Rastreamento de mensagens	Compatível	Suporta rastreamento de deslocamento e carimbo de data/hora.	Não compatível. Uma vez que uma recuperação de mensagem tenha sido confirmada, o RabbitMQ será notificado de que a mensagem pode ser excluída.
Acumulação de mensagens	Compatível	Suporta maior desempenho de acumulação do que o RabbitMQ graças ao alto rendimento.	Compatível
Persistência	Compatível	Compatível	Compatível

Recurso	RocketMQ	Kafka	RabbitMQ
Rastreamento de mensagens	Compatível	Não compatível	Suportado pelo recurso firehose ou pelo plug-in rabbitmq_tracing. No entanto, o rabbitmq_tracing reduz o desempenho e deve ser usado apenas para a solução de problemas.
Filtragem de mensagens	Compatível	Compatível	Não compatível, mas pode ser encapsulado.
Multilocatário	Compatível	Não compatível	Compatível
Multiprotocolo	Compatível com RocketMQ.	Suporta apenas o Apache de Kafka.	O RabbitMQ é baseado no AMQP e suporta MQTT e STOMP.
Multilíngue	Suporta clientes em várias linguagens de programação.	Kafka é escrito em Scala e Java e suporta clientes em várias linguagens de programação.	RabbitMQ é escrito em Erlang e suporta clientes em várias linguagens de programação.
Limitação	Planejado	Suporta limitação em clientes produtores ou consumidores.	Apoia a limitação baseada no crédito sobre os produtores, um mecanismo que aciona a proteção a partir de dentro.
Entrega de mensagem ordenada	A ordem das mensagens é mantida dentro de uma fila.	Suporta nível de partição FIFO.	Não compatível. Suporta FIFO somente para enfileiramento de mensagens de thread único sem recursos avançados, como filas atrasadas ou filas prioritárias.
Segurança	Suporta autenticação SSL.	Suporta autenticação SSL e SASL e controle de permissões de leitura/gravação.	Semelhante a Kafka.
Mensagens transacionais	Compatível	Compatível	Compatível

# 7 Comparação entre DMS for Kafka e Kafka de código aberto

DMS for Kafka é compatível com o Kafka de código aberto e tem recursos personalizados e aprimorados do Kafka. Além das vantagens do Kafka de código aberto, o DMS for Kafka oferece recursos mais confiáveis e úteis.

**Tabela 7-1** Diferenças entre o DMS for Kafka e o Kafka de código aberto

<b>Categoria</b>	<b>Item</b>	<b>DMS for Kafka</b>	<b>Kafka de código aberto</b>
Fácil de usar	Prontamente disponível	As instâncias podem ser criadas intuitivamente em minutos e usadas imediatamente com operações visualizadas e monitoramento em tempo real.	Preparar os recursos do servidor e instalar e configurar o software é demorado e propenso a erros.
	APIs	As instâncias podem ser gerenciadas facilmente chamando APIs RESTful.	N/D
Custos	Uso sob demanda	Múltiplas especificações estão disponíveis para atender às diferentes necessidades. A quantidade do broker de instância, o flavor do broker e o espaço em disco podem ser aumentados com alguns cliques.	As despesas são incorridas para configurar um serviço de mensagens e ocupar os recursos subjacentes.
	Totalmente gerenciado	Os serviços estão prontamente disponíveis sem a necessidade de recursos ou despesas adicionais de hardware.	Os usuários devem preparar recursos de hardware e configurar o serviço por si mesmos, e arcar com altos custos de uso e manutenção.

<b>Categoria</b>	<b>Item</b>	<b>DMS for Kafka</b>	<b>Kafka de código aberto</b>
Sucesso comprovado	Maduro	O DMS foi implementado em muitos produtos da Huawei Cloud e provou ser bem-sucedido em grandes eventos de comércio eletrônico. Ele também é usado nas nuvens de clientes de nível de operadora em todo o mundo e atende a rigorosos padrões de confiabilidade de nível de operadora. O DMS acompanha de perto as atualizações da comunidade para corrigir continuamente vulnerabilidades conhecidas de código aberto e adicionar suporte a novos recursos.	O uso de software de código aberto requer longo autodesenvolvimento e verificação e teve poucos casos de sucesso.
	Variada de recursos	Mantendo 100% de compatibilidade de código aberto, o DMS otimiza ainda mais o código de código aberto para melhorar o desempenho e a confiabilidade e fornece consulta, despejo, de mensagens e muitos outros recursos.	A funcionalidade é limitada e requer autodesenvolvimento.
Confiabilidade	Altamente disponível	O DMS suporta a implementação entre AZs para melhorar a confiabilidade. Além disso, a detecção automática de falhas e os alarmes garantem operações confiáveis dos principais serviços.	Alta disponibilidade requer autodesenvolvimento ou implementação de código de código aberto, que são caros e não podem garantir confiabilidade.
	O&M simples	O&M é totalmente transparente para os inquilinos com um conjunto completo de funções de monitoramento e alarme. O pessoal de O&M será informado de quaisquer exceções, eliminando a necessidade de atendimento 24 horas por dia, 7 dias por semana.	Os usuários precisam desenvolver e otimizar funções de O&M, especialmente funções de notificação de alarme. Caso contrário, o comparecimento manual é necessário.

<b>Categoria</b>	<b>Item</b>	<b>DMS for Kafka</b>	<b>Kafka de código aberto</b>
	Seguro	O DMS usa isolamento de VPC e criptografia de canal SSL.	A segurança deve ser reforçada pelos próprios usuários.

# 8 Segurança

---

## 8.1 Responsabilidades compartilhadas

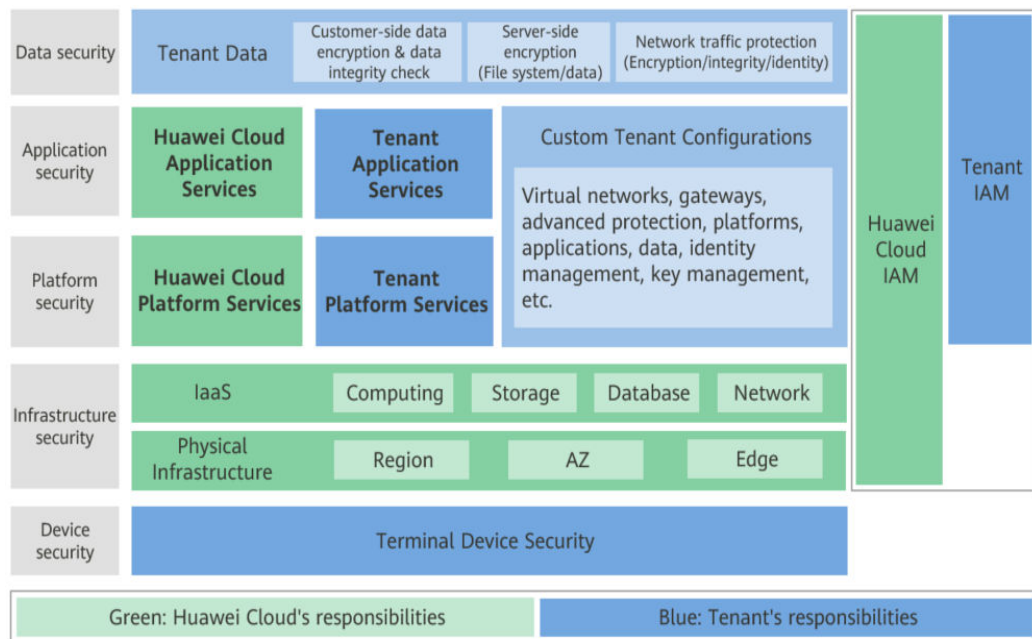
Huawei garante que seu compromisso com a segurança cibernética nunca será superado pela consideração de interesses comerciais. Para lidar com os desafios emergentes de segurança na nuvem e ameaças e ataques à segurança na nuvem, a Huawei Cloud constrói um sistema abrangente de garantia de segurança de serviços em nuvem para diferentes regiões e indústrias com base nas vantagens exclusivas de software e hardware da Huawei, leis, regulamentos, padrões da indústria e ecossistema de segurança.

**Figura 8-1** ilustra as responsabilidades partilhadas pela Huawei Cloud e pelos usuários.

- **Huawei Cloud:** garante a segurança dos serviços de nuvem e fornece nuvens seguras. As responsabilidades de segurança da Huawei Cloud incluem garantir a segurança de nossos serviços de IaaS, PaaS e SaaS, bem como os ambientes físicos dos data centers da Huawei Cloud onde nossos serviços de IaaS, PaaS e SaaS operam. A Huawei Cloud é responsável não apenas pelas funções de segurança e pelo desempenho de nossa infraestrutura, serviços de nuvem e tecnologias, mas também pela segurança geral de O&M na nuvem e, no sentido mais amplo, pela certificação de segurança de nossa infraestrutura e serviços.
- **Locatário:** usa a nuvem com segurança. Os locatários da Huawei Cloud são responsáveis pelo gerenciamento seguro e eficaz das configurações personalizadas dos serviços em nuvem, incluindo IaaS, PaaS e SaaS. Isso inclui, mas não se limita a, redes virtuais, o SO de hosts e convidados de máquinas virtuais, firewalls virtuais, API Gateway, serviços avançados de segurança, todos os tipos de serviços em nuvem, dados de locatários, contas de identidade e gerenciamento de chaves.

**O livro branco de segurança da Huawei Cloud** elabora as ideias e medidas para a construção da segurança da Huawei Cloud, incluindo estratégias de segurança na nuvem, o modelo de responsabilidade compartilhada, conformidade e privacidade, organizações e pessoal de segurança, segurança de infraestrutura, serviço e segurança de locatários, segurança de engenharia, segurança de O&M e segurança do ecossistema.

**Figura 8-1** Modelo de responsabilidade de segurança compartilhada da Huawei Cloud



## 8.2 Autenticação de identidade e controle de acesso

### Autenticação de identificação

Os solicitantes de acesso devem apresentar a credencial de identidade para verificação de validade de identidade ao acessar o DMS for Kafka no console ou chamando APIs. DMS for Kafka usa Identity and Access Management (IAM) para fornecer três modos de autenticação de identidade: **senhas**, **chaves de acesso** e **chaves de acesso temporárias**. Além disso, o DMS for Kafka fornece **proteção de logon** e **políticas de autenticação de logon** para fortalecer a segurança da autenticação de identidade.

### Controle de acesso

Você pode atribuir permissões diferentes para o DMS for Kafka aos funcionários em sua organização para o gerenciamento de permissões refinado. O IAM fornece autenticação de identidade, gerenciamento de permissões e controle de acesso, ajudando você a proteger o acesso aos seus recursos da Huawei Cloud. Para obter detalhes sobre as permissões do DMS for Kafka, consulte [Gerenciamento de permissões](#).

## 8.3 Proteção de dados

O DMS for Kafka toma medidas diferentes para manter os dados seguros e confiáveis.

**Tabela 8-1** Métodos e recursos de proteção de dados do DMS for Kafka

Medida	Descrição	Referência
DR e multiativos	Para atender aos requisitos de confiabilidade de seus dados e serviços, você pode implementar uma instância do Kafka em uma única AZ ou entre AZs (DR intra-cidade) e usar a ferramenta de sincronização de dados MirrorMaker para sincronizar dados entre instâncias.	<ul style="list-style-type: none"> <li>● <a href="#">Compra de uma instância dentro de uma AZ ou entre AZs</a></li> <li>● <a href="#">Uso de MirrorMaker para sincronizar dados entre instâncias</a></li> </ul>
Replicação de dados	Os dados podem ser sincronizados entre réplicas para consistência de dados. Quando ocorre uma exceção de rede ou falha de nó, um failover é executado automaticamente usando as réplicas. Depois que a falha é corrigida, os dados são sincronizados a partir da réplica líder para garantir a consistência dos dados.	<a href="#">Criação de um tópico com várias réplicas</a>
Persistência dos dados	Exceções podem ocorrer durante a execução diária do sistema de serviço. Alguns sistemas de serviços exigem alta confiabilidade, incluindo alta disponibilidade de instâncias, segurança de dados e capacidade de recuperação, para que os dados de backup possam ser usados para restaurar instâncias se ocorrer uma exceção, garantindo a execução do serviço.	-

## 8.4 Auditoria e logs

Cloud Trace Service (CTS) registra as operações nos recursos em nuvem em sua conta. Você pode usar os logs gerados pelo CTS para realizar análises de segurança, rastrear alterações de recursos, auditar a conformidade e localizar falhas.

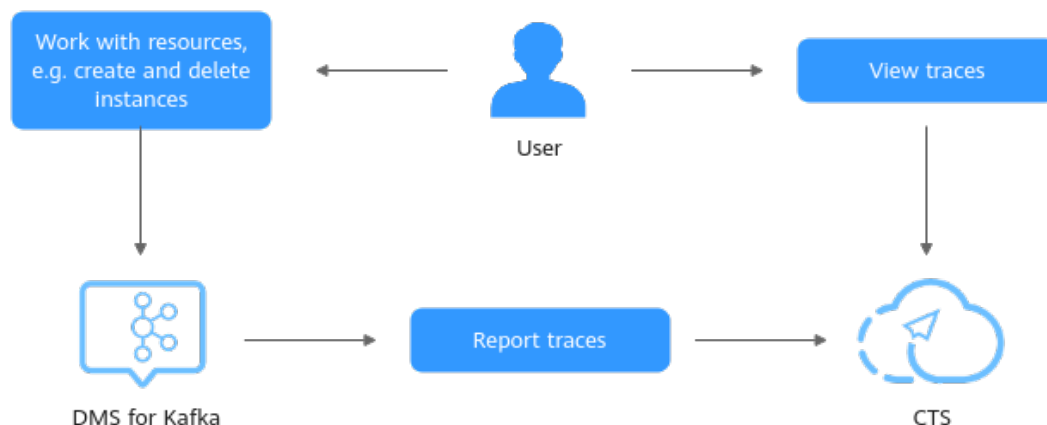
Depois de habilitar o CTS e configurar um rastreador, o CTS registra rastreamentos de gerenciamento do DMS for Kafka para auditoria.

Para obter detalhes sobre como habilitar e configurar o CTS, consulte [Habilitação do CTS](#).

Para obter detalhes sobre os rastreamentos de gerenciamento do DMS for Kafka que podem ser gravados, consulte [Operações registradas por CTS](#).



Figura 8-2 CTS



## 8.5 Resiliência

O DMS for Kafka fornece uma arquitetura de confiabilidade de três níveis e usa DR entre AZs, DR de instância dentro AZ e replicação de dados de instância para garantir a durabilidade e a confiabilidade do serviço.

Tabela 8-2 Arquitetura de confiabilidade do DMS for Kafka

Solução de confiabilidade	Descrição
DR entre AZs	O DMS for Kafka fornece instâncias entre AZs que oferecem suporte a DR entre AZs. Quando uma AZ é anormal, as instâncias ainda podem fornecer serviços.
DR de instância dentro AZ	Uma instância do Kafka tem várias réplicas para DR em nível de instância em uma AZ. Se a réplica do líder estiver com defeito, outro líder é rapidamente eleito para garantir serviços ininterruptos de Kafka.
DR de dados	A DR de dados é implementado por meio da replicação de dados.

## 8.6 Monitoramento de riscos de segurança

O DMS for Kafka usa o Cloud Eye para ajudá-lo a monitorar suas instâncias do Kafka e receber alarmes e notificações em tempo real. Você pode obter informações importantes sobre instâncias em tempo real, como solicitações de serviço, uso de recursos, tráfego, número de conexões e número de mensagens acumuladas.

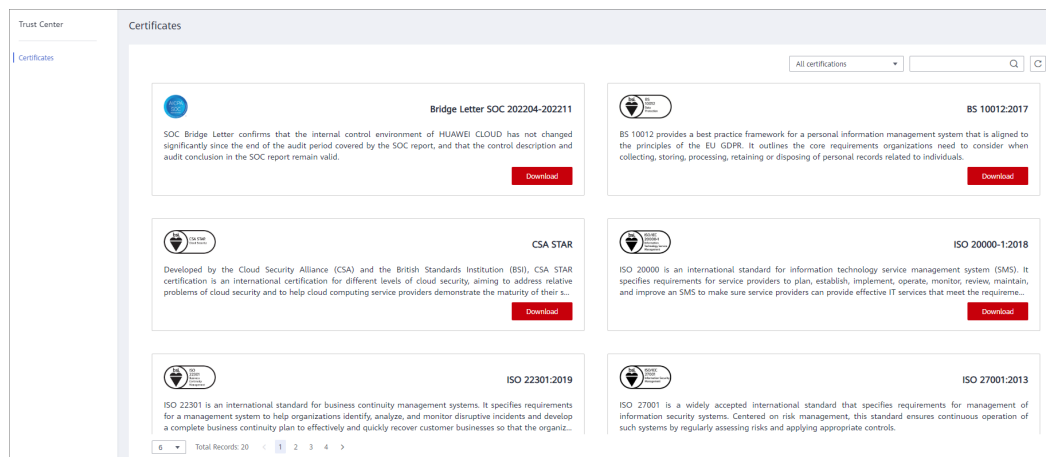
Para obter detalhes sobre as métricas do DMS for Kafka e como criar regras de alarme, consulte [Kafka Metrics](#).

## 8.7 Certificados

### Certificados de conformidade

Os serviços e plataformas da Huawei Cloud obtiveram várias certificações de segurança e de conformidade das organizações autorizadas, como a Organização Internacional de Normalização (ISO). Você pode [baixá-los](#) do console.

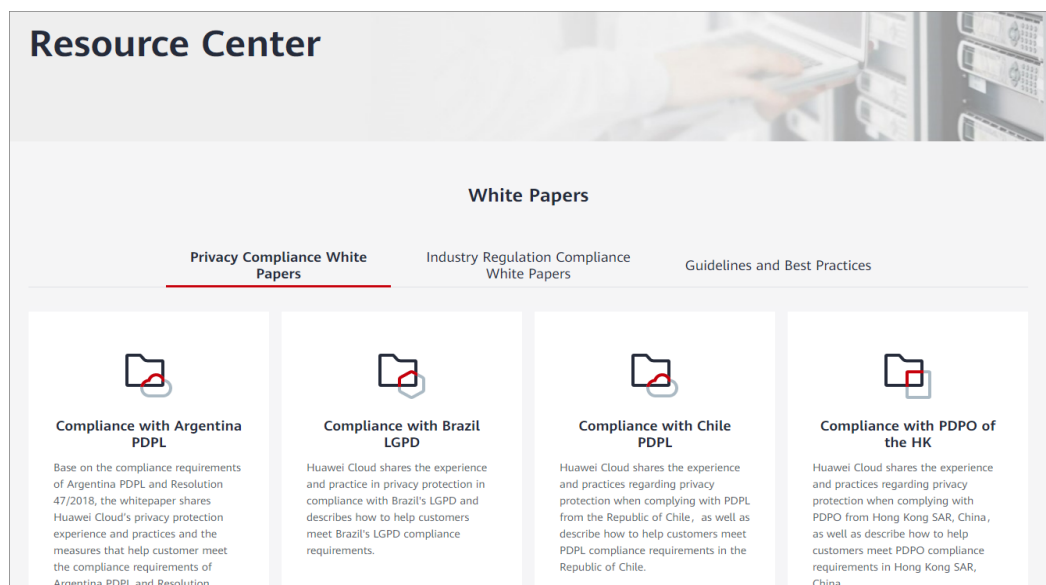
Figura 8-3 Download de certificados de conformidade



### Central de recursos

A Huawei Cloud também fornece os seguintes recursos para ajudar os usuários a atender aos requisitos de conformidade. Para obter detalhes, consulte [Central de recursos](#).

Figura 8-4 Central de recursos



# 9 Observações e restrições

Esta seção descreve as anotações e restrições no DMS for Kafka.

## Instância

**Tabela 9-1** Observações e restrições de instância

Item	Observações e restrições
Kafka em ZooKeeper	Os clusters do Kafka são gerenciados usando o ZooKeeper. A abertura de ZooKeeper pode causar falhas de operação e perdas de serviço. Atualmente, o ZooKeeper é usado somente em clusters do Kafka e não fornece serviços externamente.
Versão	<ul style="list-style-type: none"><li>● A versão de serviço pode ser 1.1.0 ou 2.7. As instâncias do Kafka não podem ser atualizadas depois de criadas.</li><li>● Clientes posteriores à versão 0.10 são suportados. Use uma versão que seja consistente com a versão de serviço.</li></ul>
Fazer logon na VM onde os brokers do Kafka residem	Não compatível
Armazenamento	<ul style="list-style-type: none"><li>● O espaço de armazenamento pode ser expandido, mas não pode ser reduzido.</li><li>● Você pode expandir o espaço de armazenamento em até 20 vezes.</li></ul>
Quantidade de largura de banda ou de broker	A largura de banda e a quantidade do broker podem ser aumentadas, mas não podem ser diminuídas.

Item	Observações e restrições
Flavor do broker	<ul style="list-style-type: none"> <li>● O flavor do broker pode ser aumentado ou diminuído.</li> <li>● Os tópicos de réplica única não oferecem suporte à criação e recuperação de mensagens durante esse período. Os serviços serão interrompidos.</li> <li>● Se um tópico tiver várias réplicas, aumentar ou diminuir o sabor do broker não interrompe os serviços, mas pode causar desordem das mensagens de partição. Avalie esse impacto e evite os horários de pico.</li> <li>● As reinicializações contínuas do broker causarão alterações no líder da partição, interrompendo as conexões por menos de um minuto quando a rede estiver estável. Para tópicos de várias réplicas, configure o mecanismo de repetição no cliente de produtor.</li> <li>● Se o número total de partições criadas para uma instância for maior que o limite superior permitido por uma nova variação, a redução não poderá ser executada.</li> </ul>
VPC, sub-rede e AZ	Depois que uma instância é criada, a VPC, a sub-rede e a AZ não podem ser modificadas.
Autenticação de Kerberos	Não compatível
Conexões de cliente de cada endereço IP	Para instâncias compradas em julho de 2020 e posteriormente, cada broker de Kafka permite no máximo 1.000 conexões de cada endereço IP por padrão. Para instâncias compradas antes de julho de 2020, cada broker de Kafka permite no máximo 200 conexões de cada endereço IP por padrão. O excesso de conexões será rejeitado.

## Tópico

**Tabela 9-2** Observações e restrições de tópico

Item	Observações e restrições
Número total de partições de tópicos	<p>O número total de partições de tópico está relacionado às especificações da instância. Para mais detalhes, consulte <a href="#">Especificações</a>.</p> <p>O Kafka gerencia mensagens por partição. Se houver muitas partições, a criação, o armazenamento e a recuperação de mensagens serão fragmentados, afetando o desempenho e a estabilidade. Se o número total de partições de tópicos atingir o limite superior, não será possível criar mais tópicos.</p>
Número de partições em um tópico	Com base nas restrições do Kafka de código aberto, o número de partições em um tópico pode ser aumentado, mas não pode ser diminuído.

Item	Observações e restrições
Quantidade de tópico	A quantidade de tópicos está relacionada ao número total de partições de tópicos e ao número de partições em cada tópico. Para mais detalhes, consulte <a href="#">Especificações</a> .
Criação automática de tópicos	<p>Compatível. Se a criação automática de tópicos estiver ativada, o sistema cria automaticamente um tópico quando uma mensagem é criada ou recuperada de um tópico que não existe. Este tópico tem as seguintes configurações padrão: 3 partições, 3 réplicas, tempo de envelhecimento de 72 horas e replicação síncrona e liberação desabilitadas.</p> <p>Depois de alterar o valor do parâmetro <b>log.retention.hours</b>, <b>default.replication.factor</b> ou <b>num.partitions</b>, os tópicos criados automaticamente mais tarde usam o novo valor. Por exemplo, se <b>num.partitions</b> for definido como <b>5</b>, um tópico criado automaticamente terá as seguintes configurações: 5 partições, 3 réplicas, tempo de envelhecimento de 72 horas e replicação síncrona e liberação desabilitadas.</p>
Replicação síncrona	Se um tópico houver apenas uma réplica, a replicação síncrona não pode ser habilitada.
Quantidade de réplicas	Tópicos de réplica única não são recomendados. Se um nó de instância estiver com defeito, um erro de serviço interno poderá ser relatado quando você consultar mensagens em um tópico com apenas uma réplica. Portanto, não é aconselhável usar um tópico com apenas uma réplica.
Tempo de envelhecimento	<p>O valor do parâmetro <b>log.retention.hours</b> terá efeito somente se o tempo de envelhecimento não tiver sido definido para o tópico.</p> <p>Por exemplo, se o tempo de envelhecimento de Topic01 estiver definido como 60 horas e <b>log.retention.hours</b> estiver definido como 72 horas, o tempo de envelhecimento real de Topic01 será 60 horas.</p>
Tópicos de importação e exportação em lote	A exportação em lote é suportada, mas a importação em lote não é suportada.
Nome do tópico	Se um nome de tópico começar com um caractere especial, por exemplo, um sinal de número (#), os dados de monitoramento não poderão ser exibidos.
Filas de atraso	Não compatível

## Grupo de consumidor

**Tabela 9-3** Observações e restrições do grupo de consumidores

Item	Observações e restrições
Criar grupos de consumidores, consumidores e produtores	Grupos de consumidores, consumidores e produtores são gerados automaticamente quando você usa a instância.
Redefinir deslocamento do consumidor	As mensagens podem ser recuperadas mais de uma vez depois que o deslocamento é redefinido.
Nome do grupo de consumidores	Se um nome de grupo de consumidores começar com um caractere especial, por exemplo, um sinal de número (#), os dados de monitoramento não poderão ser exibidos.

## Mensagem

**Tabela 9-4** Observações e restrições de mensagens

Item	Observações e restrições
Tamanho da mensagem	O comprimento máximo de uma mensagem é de 10 MB. Se o comprimento exceder 10 MB, a produção falhará.

## Usuário

**Tabela 9-5** Observações e restrições do usuário

Item	Observações e restrições
Número de usuários	Para instâncias do Kafka criadas antes de 15 de julho de 2023, é possível criar no máximo 20 usuários para cada instância. Para instâncias do Kafka criadas desde 15 de julho de 2023, é possível criar no máximo 500 usuários para cada instância.

# 10 Serviços relacionados

---

- Cloud Trace Service (CTS)

O CTS gera rastreamentos para fornecer um histórico de operações realizadas em recursos de serviço de nuvem. Os rastreamentos incluem solicitações de operação enviadas usando o console de gerenciamento ou APIs abertas, bem como os resultados da operação. Você pode exibir todos os rastreamentos gerados para consultar, auditar e rastrear as operações executadas.

Para obter detalhes sobre as operações registradas pelo CTS, consulte [Operações registradas por CTS](#).

- Virtual Private Cloud (VPC)

As instâncias do Kafka são executadas em VPCs e usam os endereços IP e a largura de banda da VPC. Os grupos de segurança de VPCs aprimoram a segurança do acesso à rede às instâncias do Kafka.

- Elastic Cloud Server (ECS)

Um ECS é uma unidade básica de computação que consiste nas vCPU, memória, SO e discos EVS. Instâncias do Kafka são executadas em ECSs. Um broker corresponde a um ECS.

- Elastic Volume Service (EVS)

O EVS fornece serviços de armazenamento em bloco para ECSs. Todos os dados do Kafka, como mensagens, metadados e logs, são armazenados em discos EVS.

- Cloud Eye

O Cloud Eye é uma plataforma aberta que fornece monitoramento, relatórios de alarme e notificação de alarme para seus recursos em tempo real.

## NOTA

Os valores de todas as métricas de instância do Kafka são relatados ao Cloud Eye a cada minuto.

- Elastic IP (EIP)

O serviço EIP fornece endereços IP públicos independentes e largura de banda para o acesso à Internet. As instâncias do Kafka vinculadas a EIPs podem ser acessadas em redes públicas.

- Tag Management Service (TMS)

O TMS é um serviço visualizado para marcação e categorização rápida e unificada de serviços em nuvem entre regiões.

As tags facilitam a identificação e o gerenciamento de instâncias do Kafka.



# 11 Conceitos básicos

---

DMS para Kafka da Huawei Cloud usa o Kafka como mecanismo de mensagens. Este capítulo apresenta explicações de conceitos básicos de Kafka.

## Tópico

Um tópico é uma categoria para mensagens. As mensagens são criadas, recuperadas e gerenciadas na forma de tópicos.

Os tópicos adotam o padrão de publicação-assinatura. Produtores publicam mensagens em tópicos. Um ou mais consumidores assinam as mensagens nos tópicos. Produtores e consumidores não estão diretamente ligados entre si.

## Produtor

Um produtor publica mensagens em tópicos. As mensagens são então entregues a outros sistemas ou módulos para processamento conforme acordado.

## Consumidor

Um consumidor assina mensagens em tópicos e processa as mensagens. Por exemplo, uma plataforma de monitoramento e alarme (um consumidor) que se inscreva para registrar mensagens em determinados tópicos pode identificar logs de alarme e, em seguida, enviar notificações de alarme por SMS ou e-mail.

## Broker

Um broker é um processo de Kafka em um cluster de Kafka. Cada processo é executado em um servidor, portanto, um broker inclui o armazenamento, a largura de banda e outros recursos do servidor.

## Partição

Um tópico é dividido em partições. As mensagens são distribuídas para várias partições para obter escalabilidade e tolerância a falhas.

## Réplicas

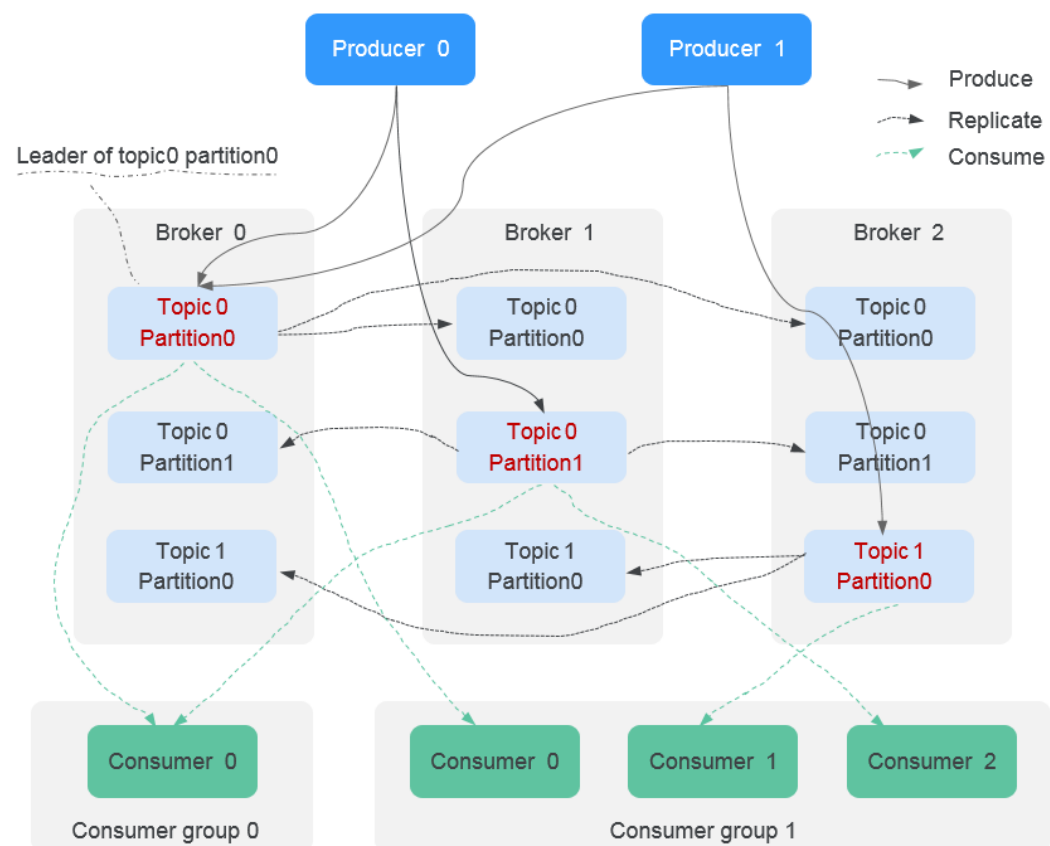
Uma réplica é uma cópia redundante de uma partição em um tópico. Cada partição pode ter uma ou mais réplicas, permitindo a confiabilidade da mensagem.

As mensagens em cada partição são totalmente replicadas e sincronizadas, evitando a perda de dados se uma réplica falhar.

Cada partição tem uma réplica como o líder que lida com a criação e recuperações de todas as mensagens. As restantes réplicas são seguidores que replicam o líder.

Tópicos e partições são conceitos lógicos, enquanto réplicas e brokers são conceitos físicos. O diagrama a seguir mostra as relações entre partições, brokers e tópicos no fluxo de mensagens.

**Figura 11-1** Fluxo de mensagens de Kafka



## Tempo de envelhecimento

O período durante o qual as mensagens são retidas. Os consumidores devem recuperar as mensagens antes que esse período termine. Caso contrário, as mensagens serão apagadas e não poderão mais ser recuperadas.

# 12 Gerenciamento de permissões

---

Você pode usar o Identity and Access Management (IAM) para gerenciar as permissões do DMS for Kafka e controlar o acesso aos seus recursos. IAM fornece autenticação de identidade, gerenciamento de permissões e controle de acesso.

Você pode criar usuários de IAM para seus funcionários e atribuir permissões a esses usuários com base em princípio de privilégio mínimo (PoLP) para controlar o seu acesso a tipos de recursos específicos. Por exemplo, você pode criar usuários do IAM para desenvolvedores de software e atribuir permissões específicas para permitir que eles usem os recursos da instância do Kafka, mas impedir que eles possam excluir recursos ou realizar operações de alto risco.

Se sua HUAWEI ID não exigir usuários individuais de IAM para gerenciamento de permissões, pule esta seção.

O IAM pode ser usado gratuitamente. Você paga apenas pelos recursos na sua conta.

Para obter mais informações, consulte [Visão geral do serviço IAM](#).

## NOTA

As políticas de permissões do DMS for Kafka são baseadas no DMS. Portanto, ao atribuir permissões, selecione as políticas de permissões do DMS.

## Permissões do DMS for Kafka

Por padrão, os novos usuários do IAM não têm nenhuma permissão atribuída. Para atribuir permissões a esses novos usuários, adicione-os a um ou mais grupos e anexe políticas de permissões ou funções a esses grupos.

DMS for Kafka é um serviço em nível de projeto implantado e acessado em regiões físicas específicas. Ao atribuir permissões do DMS for Kafka a um grupo de usuários, especifique os projetos específicos da região onde as permissões entrarão em vigor. Se você selecionar **All projects**, as permissões serão concedidas para todos os projetos específicos da região. Ao acessar o DMS for Kafka, os usuários precisam mudar para uma região onde foram autorizados a usar esse serviço.

Você pode conceder permissões usando funções e políticas.

- **Funções:** um tipo de mecanismo de autorização de alta granularidade que fornece apenas um número limitado de funções de nível de serviço. Ao usar funções para conceder permissões, você também precisa atribuir funções de dependência. No entanto, as

funções não são uma escolha adequada para autorização refinada e controle de acesso seguro.

- Políticas: um tipo de mecanismo de autorização refinado que define as permissões necessárias para realizar operações em recursos em nuvem específicos sob determinadas condições. Esse mecanismo permite uma autorização baseada em políticas mais flexível para um controle de acesso mais seguro. Por exemplo, você pode conceder aos usuários de DMS for Kafka somente as permissões para gerenciar instâncias. A maioria das políticas define permissões com base em APIs. Para as ações de API suportadas por DMS for Kafka, consulte [Políticas de permissões e ações suportadas](#).

**Tabela 12-1** lista todas as funções e políticas definidas pelo sistema suportadas pelo DMS for Kafka.

**Tabela 12-1** Funções e políticas definidas pelo sistema suportadas pelo DMS for Kafka

Nome da função/política	Descrição	Tipo	Dependência
DMS FullAccess	Permissões de administrador para DMS. Os usuários com essas permissões podem executar todas as operações no DMS.	Política definida pelo sistema	Nenhuma
DMS UserAccess	Permissões comuns de usuário para o DMS, excluindo permissões para criar, modificar, excluir, despejar, e escalar instâncias.	Política definida pelo sistema	Nenhuma
DMS ReadOnlyAccess	Permissões somente leitura para DMS. Os usuários com essas permissões só podem exibir dados do DMS.	Política definida pelo sistema	Nenhuma
DMS VPCAccess	Permissões de operação de VPC para atribuir a agências do DMS.	Política definida pelo sistema	Nenhuma
DMS KMSAccess	Permissões de operação de KMS para atribuir a agências do DMS.	Política definida pelo sistema	Nenhuma
DMS Administrator	Permissões de administrador para DMS.	Função definida pelo sistema	Essa função depende das funções <b>Tenant Guest</b> e <b>VPC Administrator</b> .

 **NOTA**

As políticas definidas pelo sistema contêm ações do OBS. Devido ao armazenamento em cache de dados, as políticas entram em vigor cinco minutos depois de serem anexadas a um usuário, grupo de usuários ou projeto corporativo.

**Tabela 2** lista as operações comuns suportadas por cada política do sistema do DMS for Kafka. Selecione as políticas conforme necessário.

**Tabela 12-2** Operações comuns suportadas por cada política definida pelo sistema do DMS for Kafka

Operação	DMS FullAccess	DMS UserAccess	DMS ReadOnly Access	DMS VPCAccess	DMS KMSAccess
Criar instâncias	√	×	×	×	×
Modificar instâncias	√	×	×	×	×
Deletar instâncias	√	×	×	×	×
Modificar especificações de instância	√	×	×	×	×
Ativar o despejo	√	×	×	×	×
Criar tarefas de despejo	√	√	×	×	×
Reiniciar instâncias	√	√	×	×	×
Consultar informações da instância	√	√	√	×	×

## Autorização refinada

Para usar uma política refinada personalizada, faça logon no console do IAM como administrador e selecione as permissões refinadas desejadas para o DMS. **Tabela 12-3** descreve dependências de permissão refinadas do DMS for Kafka.

**Tabela 12-3** Dependências de permissão refinadas do DMS for Kafka

Permissão	Descrição	Dependência
dms:instance:get	Exibir detalhes da instância	Nenhuma
dms:instance:getConnectorSinkTask	Exibir detalhes da tarefa de despejo	Nenhuma
dms:instance:getBackgroundTask	Exibir detalhes da tarefa em segundo plano	Nenhuma

Permissão	Descrição	Dependência
dms:instance:modifyAuthInfo	Alterar senha de uma instância	Nenhuma
dms:instance:resetAuthInfo	Redefinir senha de uma instância	Nenhuma
dms:instance:scale	Expandir uma instância	<ul style="list-style-type: none"> <li>● vpc:vpcs:get</li> <li>● vpc:ports:create</li> <li>● vpc:securityGroups:get</li> <li>● vpc:ports:get</li> <li>● vpc:subnets:get</li> <li>● vpc:vpcs:list</li> <li>● vpc:publicIps:get</li> <li>● vpc:publicIps:list</li> <li>● vpc:ports:update</li> <li>● vpc:publicIps:update</li> </ul>
dms:instance:connector	Ativar o despejo	<ul style="list-style-type: none"> <li>● vpc:vpcs:get</li> <li>● vpc:ports:create</li> <li>● vpc:securityGroups:get</li> <li>● vpc:ports:get</li> <li>● vpc:subnets:get</li> <li>● vpc:vpcs:list</li> <li>● vpc:publicIps:get</li> <li>● vpc:publicIps:list</li> <li>● vpc:ports:update</li> <li>● vpc:publicIps:update</li> </ul>
dms:instance:deleteConnectorSinkTask	Excluir uma tarefa de despejo	Nenhuma
dms:instance:modify	Modificar uma instância	<ul style="list-style-type: none"> <li>● vpc:vpcs:get</li> <li>● vpc:ports:create</li> <li>● vpc:securityGroups:get</li> <li>● vpc:ports:get</li> <li>● vpc:subnets:get</li> <li>● vpc:vpcs:list</li> <li>● vpc:publicIps:get</li> <li>● vpc:publicIps:list</li> <li>● vpc:ports:update</li> <li>● vpc:publicIps:update</li> </ul>
dms:instance:deleteBackgroundTask	Excluir uma tarefa em segundo plano	Nenhuma

Permissão	Descrição	Dependência
dms:instance:modifyStatus	Reinicializar uma instância	Nenhuma
dms:instance:createConnectorSinkTask	Criar tarefas de despejo	Nenhuma
dms:instance:delete	Excluir uma instância	Nenhuma
dms:instance:create	Criar uma instância	<ul style="list-style-type: none"> <li>● vpc:vpcs:get</li> <li>● vpc:ports:create</li> <li>● vpc:securityGroups:get</li> <li>● vpc:ports:get</li> <li>● vpc:subnets:get</li> <li>● vpc:vpcs:list</li> <li>● vpc:publicIps:get</li> <li>● vpc:publicIps:list</li> <li>● vpc:ports:update</li> <li>● vpc:publicIps:update</li> </ul>
dms:instance:listConnectorSinkTask	Exibir a lista de tarefas de despejo	Nenhuma
dms:instance:list	Exibir a lista de instâncias	Nenhuma

## Links úteis

- [O que é o IAM?](#)
- [Criação de grupos de usuários, usuários e concessão de permissões de DMS for Kafka](#)
- [Políticas de permissões e ações suportadas](#)