Distributed Message Service for RabbitMQ

Service Overview

Issue 01

Date 2025-07-03

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 What Is DMS for RabbitMQ?..... 1
2 Product Advantages..... 2
3 Application Scenarios..... 4
4 Specifications..... 7
5 Comparing RabbitMQ, Kafka, and RocketMQ..... 12
6 Comparing RabbitMQ AMQP-0-9-1 with Open-Source RabbitMQ..... 15
7 Comparing RabbitMQ Versions..... 19
8 Related Services..... 20
9 Security..... 21
9.1 Shared Responsibilities..... 21
9.2 Identity Authentication and Access Control..... 23
9.3 Data Protection..... 23
9.4 Audit and Logs..... 24
9.5 Resilience..... 24
9.6 Security Risks Monitoring..... 25
9.7 Certificates..... 25
10 Notes and Constraints..... 27
11 Basic Concepts..... 33
12 Exchanges..... 35
13 Permissions Management..... 40

1 What Is DMS for RabbitMQ?

Based on the open-source RabbitMQ, Distributed Message Service (DMS) for RabbitMQ provides messaging services with rich messaging features, flexible routing, alarms, monitoring, and high availability functions. It is applicable to flash sales, flow control, and system decoupling scenarios.

- Immediate use

  DMS for RabbitMQ provides single-node and cluster instances with a range of specifications for you to choose from. Instances can be created with just a few clicks on the console, without requiring you to prepare servers.

- Rich features

  DMS for RabbitMQ supports Advanced Message Queuing Protocol (AMQP) and a variety of messaging features such as message broadcast, delayed delivery, and dead letter queues.

- Flexible routing

  In RabbitMQ, an exchange receives messages from producers and pushes the messages to queues. RabbitMQ provides direct, topic, headers, and fanout exchanges. You can also bind and customize exchanges.

- High availability

  Cluster RabbitMQ instances provide quorum queues, which can be used to replicate queue data between RabbitMQ nodes, ensuring that queues can still run when a node breaks down.

- Monitoring and alarm

  RabbitMQ cluster metrics are monitored and reported, including broker memory, CPU usage, and network flow. If an exception is detected, an alarm will be triggered.

2 Product Advantages

Huawei Cloud DMS for RabbitMQ provides easy-to-use message queuing based on RabbitMQ. Services can be quickly migrated to the cloud without any change, reducing maintenance and usage costs.

- Rapid deployment

  Simply set instance information on the DMS for RabbitMQ console, submit your order, and a complete RabbitMQ instance will be automatically created and deployed.

- Service migration without modifications

  DMS for RabbitMQ is compatible with open-source RabbitMQ APIs and supports all message processing functions of open-source RabbitMQ.

  If your application services are developed based on open-source RabbitMQ, you can easily migrate them to Huawei Cloud DMS for RabbitMQ after specifying a few authentication configurations.

  NOTE

  RabbitMQ instances are compatible with RabbitMQ 3.8.35 and 3.12.13.

- Exclusive experience

  RabbitMQ instances are physically isolated from each other and exclusively owned by the tenant.

- High performance

  Each queue can process up to 100,000 transactions per second (with default configurations). Performance can be increased simply by adding queues.

- Data security

  Operations on RabbitMQ instances are recorded and can be audited. Messages can be encrypted before storage.

  In addition to SSL, VPCs and security groups also provide security controls on network access.

- Simple O&M

  Huawei Cloud provides a whole set of monitoring and alarm services, eliminating the need for 24/7 attendance. RabbitMQ instance metrics are monitored and reported, including the number of partitions, topics, and accumulated messages. You can configure alarm rules and receive SMS or email notifications on how your services are running in real time.

- Multi-language support

  RabbitMQ is an open-source service based on AMQP. It is used to store and forward messages in a distributed system. A RabbitMQ server is compiled in Erlang (supporting high concurrency, distribution, and robust fault tolerance), and a RabbitMQ client can be compiled in various programming languages, including Python, Ruby, .NET, Java, JMS, C, PHP, ActionScript, XMPP, STOMP, and AJAX.

3 Application Scenarios

RabbitMQ is popular message-oriented middleware that features highly reliable, asynchronous message delivery. It can be used for transmitting data between different systems in the enterprise application, payment, telecommunications, e-commerce, social networking, instant messaging, video, Internet of Things, and Internet of Vehicle industries.

For synchronization links that require real-time results, Remote Procedure Call (RPC) is recommended.

Asynchronous Communication

Non-core or less important messages are sent asynchronously to receiving systems, so that the main service process is not kept waiting for the results of other systems, allowing for faster responses.

For example, RabbitMQ can be used to send a notification email and SMS message after a user has registered with a website, providing fast responses throughout the registration process.

Figure 3-1 Serial registration and notification

Figure 3-2 Asynchronous registration and notification using message queues

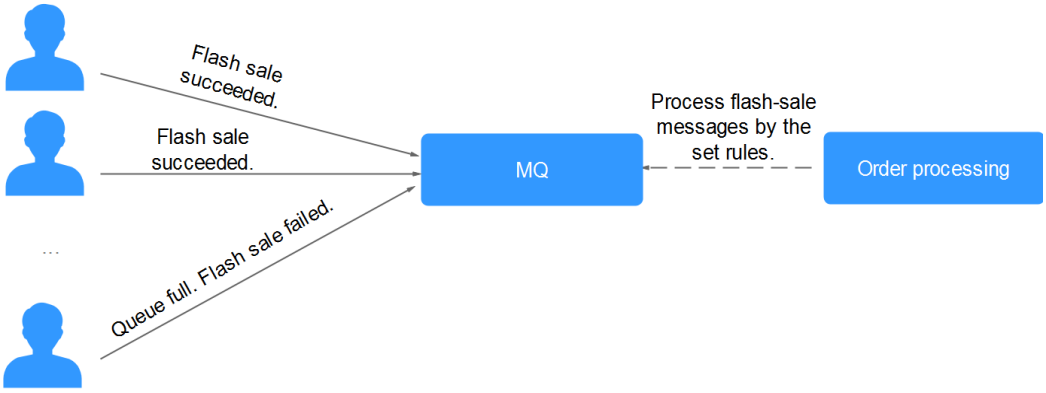
Traffic Control

In e-commerce systems or large-scale websites, there is a processing capability gap between upstream and downstream systems. Traffic bursts from upstream

systems with high processing capabilities may have a large impact on downstream systems with lower processing capabilities. For example, online sales promotions involve a huge amount of traffic flooding into e-commerce systems. RabbitMQ provides a three-day buffer by default for hundreds of millions of messages, such as orders and other information. In this way, message consumption systems can process the messages during off-peak periods.

In addition, flash sale traffic bursts originating from frontend systems can be handled with RabbitMQ, keeping the backend systems from crashing.

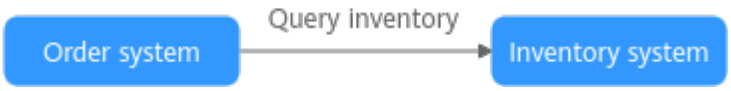
Figure 3-3 Traffic burst handling using RabbitMQ



System Decoupling

Take e-commerce flash sales as an example. Traditionally, an order processing system sends order requests to the inventory system and waits for responses. If the inventory system goes down, the order processing system will not be able to get the data it wants, and the order will fail to be submitted. This means that the order processing system and the inventory system are closely coupled.

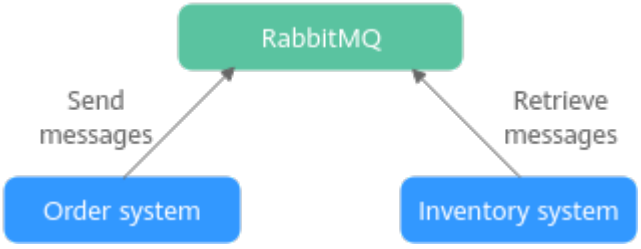
Figure 3-4 Closely coupled systems



With RabbitMQ, order submission data will be stored in queues. Then, a response will be returned indicating that the order has been submitted.

The inventory system consumes the order submission message it has subscribed to. In this way, order submission will not be interrupted even if the inventory system breaks down.

Figure 3-5 System decoupling



High Availability

Normally, there is only one broker. If the broker fails, queues on it will become unavailable.

Queue mirroring is available since RabbitMQ 2.6.0. In a RabbitMQ cluster, queues can be mirrored across brokers. In the event of a broker failure, services are still available because the mirrors will take over.

Quorum queues are available since RabbitMQ 3.8. Quorum queues can be used to replicate queue data, ensuring that queues can still run if a broker breaks down.

4 Specifications

RabbitMQ Instance Specifications

RabbitMQ instances are compatible with open-source RabbitMQ 3.8.35 and 3.12.13, and in-house RabbitMQ AMQP-0-9-1. **Table 4-1** and **Table 4-2** list the specifications of single-node and cluster RabbitMQ instances.

Table 4-1 Specifications of cluster RabbitMQ 3.x.x instances

Flavor	Brokers	Storage Space (GB)	Reference TPS	Maximum Consumers per Broker	Recommended Queues per Broker	Maximum Connections per Broker
rabbitmq.2u4g.cluster	3	300–90,000	3,000	4,000	100	1,000
	5	500–150,000	5,000	4,000	100	1,000
	7	700–210,000	7,000	4,000	100	1,000
rabbitmq.4u8g.cluster	3	300–90,000	6,000	8,000	200	2,000
	5	500–150,000	10,000	8,000	200	2,000
	7	700–210,000	14,000	8,000	200	2,000
rabbitmq.8u16g.cluster	3	300–90,000	12,000	16,000	400	4,000
	5	500–150,000	20,000	16,000	400	4,000
	7	700–210,000	28,000	16,000	400	4,000

Flavor	Brokers	Storage Space (GB)	Reference TPS	Maximum Consumers per Broker	Recommended Queues per Broker	Maximum Connections per Broker
rabbitmq.12u24g.cluster	3	300–90,000	24,000	24,000	600	6,000
	5	500–150,000	40,000	24,000	600	6,000
	7	700–210,000	56,000	24,000	600	6,000
rabbitmq.16u32g.cluster	3	300–90,000	48,000	32,000	800	8,000
	5	500–150,000	80,000	32,000	800	8,000
	7	700–210,000	112,000	32,000	800	8,000
rabbitmq.24u48g.cluster	3	300–90,000	60,000	40,000	1,000	10,000
	5	500–150,000	100,000	40,000	1,000	10,000
	7	700–210,000	140,000	40,000	1,000	10,000
rabbitmq.32u64g.cluster	3	300–90,000	72,000	40,000	1,000	10,000
	5	500–150,000	120,000	40,000	1,000	10,000
	7	700–210,000	168,000	40,000	1,000	10,000

NOTE

- In the preceding tables, TPS (of production and consumption) is represented by the number of messages (2 KB each) processed per second. In the tests, persistence and queue mirroring were not enabled. Messages were retrieved immediately after creation and were not accumulated in the queues. The data is for reference only and may differ from that in your production environment.
- Performance is related to the queue quantity, message accumulation, number of connections, number of channels, number of consumers, queue mirroring, priority queue, message persistence, and the exchange type. Select instance specifications based on the pressure test result of the service model.
- A maximum of 2047 channels can be opened on a connection.
- Single-node instances can be used for testing. Do not use them for message production. Single-node flavors are not yet available.
- The maximum number of connections per broker of RabbitMQ 3.x.x instances is listed in the table above. The maximum number of connections of the entire cluster is 30,000.

Table 4-2 Specifications of cluster RabbitMQ instances (professional edition AMQP-0-9-1)

Flavor	Storage Space (GB)	Reference TPS	Max. Connections	Max. Queues
amqp.p2.large.6	200–60,000	3,000	1000	500
amqp.p2.large.10	200–60,000	5,000	1000	500
amqp.p2.large.14	200–60,000	7,000	2,000	1000
amqp.p2.large.20	200–60,000	10,000	2,000	1000
amqp.p2.large.28	200–60,000	14,000	2,000	1000
amqp.p2.large.40	200–60,000	20,000	3,000	1,500
amqp.p2.large.56	400–120,000	28,000	4,000	2,000
amqp.p2.large.80	400–120,000	40,000	6,000	3,000
amqp.p2.large.112	400–120,000	56,000	8,000	4,000
amqp.p2.large.144	600–180,000	72,000	10,000	5,000
amqp.p2.large.200	600–180,000	100,000	12,000	6,000
amqp.p2.large.240	800–240,000	120,000	16,000	8,000
amqp.p2.large.280	1200–360,000	140,000	16,000	8,000

NOTE

- In the preceding tables, TPS is represented by the number of messages (4 KB each) processed per second. In the tests, messages were retrieved immediately after creation and were not accumulated in the queues. The data is for reference only and may differ from that in your production environment.
- Performance is related to the queue quantity, message accumulation, number of connections, number of channels, number of consumers, and exchange types. Select instance specifications based on the pressure test result of the service model.
- A maximum of 2000 channels can be opened on a connection.
- Single-node instances can be used for testing. Do not use them for message production. Single-node flavors are not yet available.

Mapping Between Old and New Flavors

Old and new RabbitMQ instance flavors are compared as follows.

Table 4-3 Mapping between old and new RabbitMQ instance flavors

Old Flavor		New Flavor	
Flavor	Reference TPS	Flavor	Reference TPS
4 vCPUs | 8 GB × 3	3,000	rabbitmq.4u8g.cluster * 3	6,000
8 vCPUs | 16 GB × 3	6,000	rabbitmq.8u16g.cluster * 3	12,000
16 vCPUs | 32 GB × 3	24,000	rabbitmq.16u32g.cluster * 3	48,000

Instances with new flavors have the following features:

- New flavors provide better performance at the same price.
- Old flavors use non-exclusive resources. If the load is heavy, resources conflicts will occur. By contrast, new flavors use exclusive resources so they provide better performance and stability.
- New flavors support scale-out and scale-up to satisfy service changes.
- Larger flavors up to **rabbitmq.32u64g.cluster** are available now.
- More disk type options: General Purpose SSD and Extreme SSD are now available, in addition to the original disk types.

Storage Space Selection

In cluster mode, RabbitMQ persists messages to disk. When creating a RabbitMQ instance, select a proper storage space size based on the estimated message size and the number of replicas in a mirrored queue, which can be maximally equal to the number of brokers in the cluster.

For example, if the estimated message size is 100 GB, the disk capacity must be at least: 100 GB x Number of mirrored replicas + 100 GB (reserved).

For single-node instances, select a storage space size based on the estimated message size and the reserved disk space.

You can change the number of brokers in a cluster, but cannot change the specifications of a single-node instance.

5 Comparing RabbitMQ, Kafka, and RocketMQ

Table 5-1 Functions

Feature	RocketMQ	Kafka	RabbitMQ
Priority queue	Not supported	Not supported	• V3.x.x: Supported. It is recommended that the priority be set to 0–10. • AMQP-0-9-1: Supported. Set the priority to 1–9.
Delayed queue	Supported	Not supported	• V3.x.x: Not supported. • AMQP-0-9-1: Supported.
Dead letter queue	Supported	Not supported	Supported
Message retry	Supported	Not supported	• V3.x.x: Not supported. • AMQP-0-9-1: Supported.
Retrieval mode	Pull-based and push-based	Pull-based	Pull-based and push-based
Message broadcasting	Supported	Supported	Supported

Feature	RocketMQ	Kafka	RabbitMQ
Message tracking	Supported	Supports offset and timestamp tracking.	• V3.x.x: Not supported. Once a message retrieval has been acknowledged, RabbitMQ will be notified that the message can be deleted. • AMQP-0-9-1: Supported.
Message accumulation	Supported	Supports higher accumulation performance than RabbitMQ thanks to high throughput.	Supported
Persistence	Supported	Supported	Supported
Message tracing	Supported	Not supported	• V3.x.x: Not supported. • AMQP-0-9-1: Supported.
Message filtering	Supported	Supported	• V3.x.x: Not supported, but can be encapsulated. • AMQP-0-9-1: Supported.
Multi-tenancy	Supported	Supported	Supported
Multi-protocol	Compatible with RocketMQ.	Only supports Apache Kafka.	RabbitMQ is based on AMQP.
Multi-language	Supports clients in multiple programming languages.	Kafka is written in Scala and Java and supports clients in multiple programming languages.	Supports clients in multiple programming languages.
Throttling	RocketMQ 5.x supports traffic control based on instance specifications.	Supports throttling on producer or consumer clients, users, and topics.	Supports credit-based throttling on producers, a mechanism that triggers protection from within.

Feature	RocketMQ	Kafka	RabbitMQ
Ordered message delivery	Message order is maintained within a queue.	Supports partition-level FIFO.	Supports FIFO only for single-threaded message queuing without advanced features such as delayed queues or priority queues.
Security	Supports SSL authentication.	Supports SSL and SASL authentication and read/write permissions control.	• 3.x.x: SSL authentication is supported. • AMQP-0-9-1: ACL is supported.
Transactional messages	Supported	Supported	Supported

6 Comparing RabbitMQ AMQP-0-9-1 with Open-Source RabbitMQ

DMS for RabbitMQ of the AMQP-0-9-1 version is message-oriented middleware using the highly available distributed architecture. This version is compatible with open-source RabbitMQ clients. DMS for RabbitMQ addresses message stacking, split-brain, and other stability issues in open-source RabbitMQ. In addition, DMS for RabbitMQ features high concurrency, distributed deployment, and flexible scaling.

The following section describes the comparison by function, stability, exchange, and queue.

Functions

Table 6-1 Comparing functions

Function	AMQP-0-9-1	Open Source
Client SDKs	Open-source SDKs of all languages and versions are supported.	Open-source SDKs are supported.
Scheduled messages	Can be scheduled at any time, precise in seconds, and stacked massively.	By plug-ins or message time-to-live (TTL).
Transactional messages	Not supported.	Supported.
Ordered messages	Not supported.	Supported.
Message priority	Supported.	Supported.

Function	AMQP-0-9-1	Open Source
Message retry mechanism	Supported. Messages will be re-delivered if no response is received within a specified period of consumption. Retry intervals: 1 minute. Maximum of retries: 16. Messages will be discarded or sent to dead letter exchanges later.	Not supported.
Monitoring metrics	Rich metrics of virtual host, exchange, and queue.	The following two schemes are supported: • 1: Management UI is available with rich metrics. The metric storage and display need to be manually developed. • 2: Prometheus+Grafana can be used to obtain basic metrics which may not ensure precision.
Message tracing	A whole message lifecycle can be traced and displayed on a white screen. Messages can be indexed by queue, message ID, and message processing duration.	Message traces are stored by text in the log file on servers. Querying or locating may be inefficient.

Stability

Table 6-2 Comparing stability

Function	AMQP-0-9-1	Open Source
Message stacking	Massive messages are allowed to be stacked. High performance is not affected.	Stacked messages easily cause memory-induced breakdown.
Scalability	An architecture of distributed clusters without masters supports rapid scale-out of clusters.	Expansions and reductions by changing machine specifications.
Service availability	99.95% achieved by distributed clusters across availability zones (AZs).	Developed by Erlang. Experiential O&M. Stability issues remain unresolved.

Function	AMQP-0-9-1	Open Source
Data reliability	Data in triple replicas, which does not affect the TPS performance.	Increasing replicas deteriorates TPS performance.
Inspection	Dead locks or breakdowns can be automatically detected and rectified.	None.

Exchange

Table 6-3 Comparing exchanges

Function	AMQP-0-9-1	Open Source
Exchange type	Direct, fanout, headers, topic, x-delayed-message and x-consistent-hash.	Direct, fanout, headers, topic, x-delayed-message and x-consistent-hash.
Persistence	Enabled by default.	Can be configured.
Automatic deletion	Supported.	Supported.
Internal	Not supported.	Supported.
Alternate exchange	Not supported.	Supported.
Consistent hash exchange	Supported.	Supported.

Queue

Table 6-4 Comparing queues

Function	AMQP-0-9-1	Open Source
Queue type	Distributed high availability clusters by default.	The following types can be configured: • Classic • Quorum
Node	O&M-free by default.	Needs to be configured.
Persistence	Enabled by default.	Persistence or non-persistence is supported.

Function	AMQP-0-9-1	Open Source
Max length	By default, massive messages can be stacked.	Needs to be configured to avoid system breakdowns due to message stack-induced memory issues.
Max length bytes	By default, massive messages can be stacked.	Needs to be configured to avoid system breakdowns due to message stack-induced memory issues.
Max in memory length	By default, massive messages can be stacked.	Needs to be configured to avoid system breakdowns due to message stack-induced memory issues.
Max in memory bytes	By default, massive messages can be stacked.	Needs to be configured to avoid system breakdowns due to message stack-induced memory issues.
Delivery limit	Fixed, 16 by default.	Needs to be configured.
Dead letter exchange	Supported.	Supported.
Dead letter routing key	Supported.	Supported.
Single active consumer	Not supported.	Supported.

7 Comparing RabbitMQ Versions

Table 7-1 compares RabbitMQ 3.x.x and AMQP-0-9-1.

Table 7-1 Comparing versions

Function	3.x.x	AMQP-0-9-1
SSL	√	×
Public access	Can be enabled on the RabbitMQ console	×
Quorum queues	√	×
Mirrored queues	√	×
Priority queues	√	√
Plug-ins	√	×
Management UI	√	×
Instance password reset	√	×
Instance specification change	√	√
User management	Set on the management UI	Set on the RabbitMQ console
Message query	×	√

8 Related Services

- Elastic Cloud Server (ECS)

  An ECS is a basic computing unit that consists of vCPUs, memory, OS, and EVS disks. RabbitMQ instances run on ECSs. A broker corresponds to an ECS.

- Elastic Volume Service (EVS)

  EVS provides block storage services for ECSs. All RabbitMQ data, such as messages, metadata, and logs, is stored in EVS disks.

- Cloud Trace Service (CTS)

  Cloud Trace Service (CTS) generates traces to provide you with a history of operations performed on cloud service resources. The traces include operation requests sent using the Huawei Cloud management console or open APIs as well as the operation results. You can view all generated traces to query, audit, and backtrack performed operations.

- VPC

  RabbitMQ instances run in VPCs and use the IP addresses and bandwidth of VPC. Security groups of VPCs enhance the security of network access to the RabbitMQ instances.

- Cloud Eye

  Cloud Eye is an open platform that provides monitoring, alarm reporting, and alarm notification for your resources in real time.

  NOTE

  The values of all RabbitMQ instance metrics are reported to Cloud Eye every minute.

- Elastic IP (EIP)

  The EIP service provides independent public IP addresses and bandwidth for Internet access. RabbitMQ instances bound with EIPs can be accessed over public networks.

- Tag Management Service (TMS)

  TMS is a visualized service for fast and unified cross-region tagging and categorization of cloud services.

  Tags facilitate RabbitMQ instance identification and management.

9 Security

9.1 Shared Responsibilities

Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Unlike traditional on-premises data centers, cloud computing separates operators from users. This approach not only enhances flexibility and control for users but also greatly reduces their operational workload. For this reason, cloud security cannot be fully ensured by one party. Cloud security requires joint efforts of Huawei Cloud and you, as shown in **Figure 9-1**.

- **Huawei Cloud**: Huawei Cloud is responsible for infrastructure security, including security and compliance, regardless of cloud service categories. The infrastructure consists of physical data centers, which house compute, storage, and network resources, virtualization platforms, and cloud services Huawei Cloud provides for you. In PaaS and SaaS scenarios, Huawei Cloud is responsible for security settings, vulnerability remediation, security controls, and detecting any intrusions into the network where your services or Huawei Cloud components are deployed.
- **Customer**: As our customer, your ownership of and control over your data assets will not be transferred under any cloud service category. Without your explicit authorization, Huawei Cloud will not use or monetize your data, but you are responsible for protecting your data and managing identities and access. This includes ensuring the legal compliance of your data on the cloud, using secure credentials (such as strong passwords and multi-factor authentication), and properly managing those credentials, as well as monitoring and managing content security, looking out for abnormal account behavior, and responding to it, when discovered, in a timely manner.

Figure 9-1 Huawei Cloud shared security responsibility model

Cloud security responsibilities are determined by control, visibility, and availability. When you migrate services to the cloud, assets, such as devices, hardware, software, media, VMs, OSs, and data, are controlled by both you and Huawei Cloud. This means that your responsibilities depend on the cloud services you select. As shown in **Figure 9-1**, customers can select different cloud service types (such as IaaS, PaaS, and SaaS services) based on their service requirements. As control over components varies across different cloud service categories, the responsibilities are shared differently.

- In on-premises scenarios, customers have full control over assets such as hardware, software, and data, so tenants are responsible for the security of all components.
- In IaaS scenarios, customers have control over all components except the underlying infrastructure. So, customers are responsible for securing these components. This includes ensuring the legal compliance of the applications, maintaining development and design security, and managing vulnerability remediation, configuration security, and security controls for related components such as middleware, databases, and operating systems.
- In PaaS scenarios, customers are responsible for the applications they deploy, as well as the security settings and policies of the middleware, database, and network access under their control.
- In SaaS scenarios, customers have control over their content, accounts, and permissions. They need to protect their content, and properly configure and protect their accounts and permissions in compliance with laws and regulations.

9.2 Identity Authentication and Access Control

Identity Authentication

No matter whether you access DMS for RabbitMQ through the console or APIs, you are required to provide the identity credential and verify the identity validity. In addition, login and login authentication policies are provided to harden identity authentication security.

DMS for RabbitMQ uses IAM to provide three identity authentication modes: **passwords**, **access keys**, and **temporary access keys**. **Login protection** and **login authentication policies** are also provided.

Access Control

You can assign different permissions for DMS for RabbitMQ to employees in your organization for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your Huawei Cloud resources. For details about DMS for RabbitMQ permissions, see **Permissions Management**.

9.3 Data Protection

DMS for RabbitMQ takes different measures to keep data secure and reliable.

Table 9-1 DMS for RabbitMQ data protection methods and features

Measure	Description	Reference
DR and multi-active	To meet reliability requirements of your data and services, you can deploy a RabbitMQ instance in a single AZ (single equipment room) or across AZs (intra-city DR).	**Deploying an Instance Within an AZ or Across AZs**
Data replication	Data can be synchronized between replicas for data consistency. When a network exception or node fault occurs, a failover is automatically performed using the replicas. After the fault is rectified, data is synchronized from the leader replica to ensure data consistency.	• **Configuring Queue Mirroring** • **Quorum Queues**

Measure	Description	Reference
Data persistence	Exceptions may occur during daily running of the service system. Some service systems require high reliability, including high availability of instances, data security, and recoverability, so that backup data can be used to restore instances if an exception occurs, ensuring service running.	**Message Persistence**

9.4 Audit and Logs

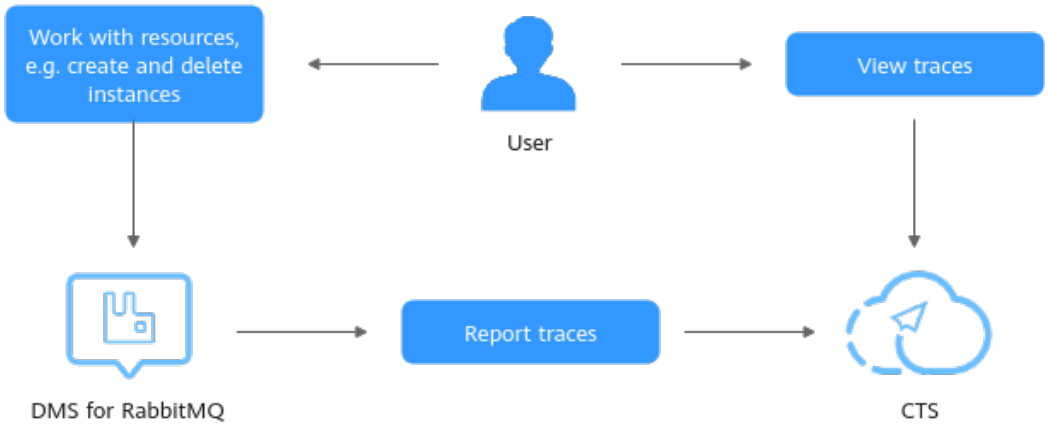
Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, trace resource changes, audit compliance, and locate faults.

After you enable CTS and configure a tracker, CTS records management traces of DMS for RabbitMQ for auditing.

For details about how to enable and configure CTS, see **Enabling CTS**.

For details about the DMS for RabbitMQ management traces that can be recorded, see **Operations Logged by CTS**.

Figure 9-2 CTS



9.5 Resilience

DMS for RabbitMQ provides a three-level reliability architecture and uses cross-AZ DR, intra-AZ instance DR, and instance data replication to ensure service durability and reliability.

Table 9-2 Reliability architecture of DMS for RabbitMQ

Reliability Solution	Description
Cross-AZ DR	DMS for RabbitMQ provides cross-AZ instances that support cross-AZ DR. When an AZ is abnormal, the instances can still provide services.
Intra-AZ instance DR	In a RabbitMQ cluster, data is replicated to all nodes through mirrored queues, preventing service interruption in case of a node breakdown.
Data DR	Data DR is implemented through data replication.

9.6 Security Risks Monitoring

DMS for RabbitMQ uses Cloud Eye to help you monitor your RabbitMQ instances and receive alarms and notifications in real time. You can obtain key information about instances in real time, such as service requests, resource usage, traffic, number of connections, and number of accumulated messages.

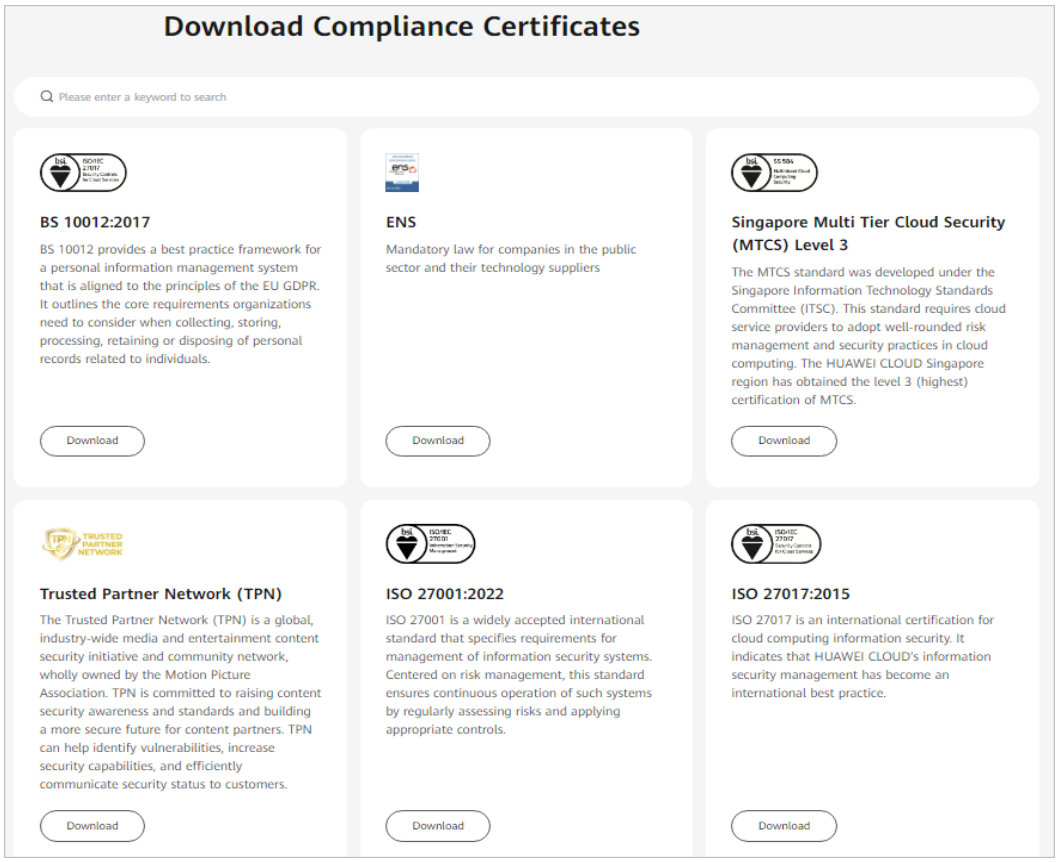
For details about DMS for RabbitMQ metrics and how to create alarm rules, see **RabbitMQ Metrics**.

9.7 Certificates

Compliance Certificates

Huawei Cloud services and platforms have obtained various security and compliance certifications from authoritative organizations, such as International Organization for Standardization (ISO). You can **download** them from the console.

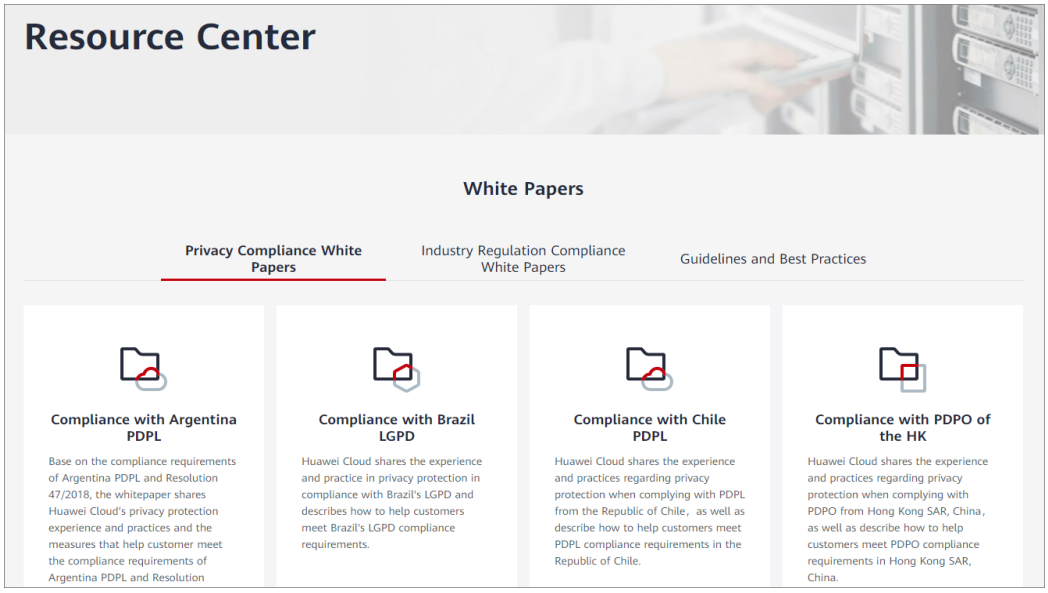
Figure 9-3 Downloading compliance certificates



Resource Center

Huawei Cloud also provides the following resources to help users meet compliance requirements. For details, see **Resource Center**.

Figure 9-4 Resource center



10 Notes and Constraints

This section describes the notes and constraints on Huawei Cloud Distributed Message Service (DMS) for RabbitMQ. Use your RabbitMQ instances as prescribed to avoid program exceptions.

NOTICE

Any instability caused by ignorance of the notes and constraints is not covered by the SLA.

Instance

Table 10-1 Notes and constraints

Item	Constraint
Version	Server version: 3.8.35, 3.12.13, and AMQP-0-9-1
Number of connections	The allowed number of connections differs by instance specifications and mode (single-node or cluster). For details, see **Specifications**.
Channels	Number of channels that can be created for a single connection is as follows: • RabbitMQ 3.x.x: ≤ 2,047 • RabbitMQ AMQP-0-9-1: ≤ 2,000
Memory high watermark	≤ 40% If the memory usage exceeds 40%, the high memory watermark may be triggered, blocking publishers. Available only for RabbitMQ 3.x.x instances.

Item	Constraint
Disk high watermark	≥ 5 GB If the remaining disk space is less than 5 GB, the high disk watermark is triggered, blocking publishers. Available only for RabbitMQ 3.x.x instances.
cluster_partition_handling	pause_minority When a network partition occurs in a cluster, cluster brokers will determine whether they are in a minority, that is, fewer than or equal to the total number of brokers. Minority brokers pause when a partition starts, detect the network status periodically, and start again when the partition ends. If queue mirroring is not enabled, queue replicas in the minority will no longer be available for message creation and retrieval. This strategy sacrifices availability for data consistency.
RabbitMQ plug-ins	RabbitMQ plug-ins can be used for testing and service migration. Do not use them for production. Reliability issues caused from using plug-ins are not within commitments on SLAs. Available only for RabbitMQ 3.x.x.
VPC, subnet, and AZ	After an instance is created, its VPC, subnet, and AZ cannot be modified.
Storage space per broker	• The storage space can be expanded but cannot be reduced. • You can expand the storage space 20 times.
Broker quantity	• The broker quantity can be increased but cannot be decreased for a cluster instance. • Services may temporarily stutter during the broker increase. Ensure that your client can auto-reconnect. Modify specifications during off-peak hours. • This function is not available for single-node instances. Available only for RabbitMQ 3.x.x.

Item	Constraint
Broker Flavor	• The broker flavor can be increased or decreased for RabbitMQ 3.x.x. The broker flavor can only be increased for RabbitMQ AMQP-0-9-1. • RabbitMQ AMQP-0-9-1: For single-node and cluster instances, intermittent disconnections may occur for several seconds during the modification. Ensure that your client can auto-reconnect. Modify specifications during off-peak hours. • For single-node RabbitMQ 3.x.x instances, nodes are restarted and services may temporarily stutter in minutes during the modification. Ensure that your client can auto-reconnect. Modify specifications during off-peak hours. • For cluster RabbitMQ 3.x.x instances not configured with mirrored/quorum queues, nodes are restarted in sequence and services may stutter in minutes during the modification. Ensure that your client can auto-reconnect. Modify specifications during off-peak hours. • For cluster RabbitMQ 3.x.x instances not configured with mirrored/quorum queues, nodes are restarted in sequence and services may stutter for several seconds during the modification. Ensure that your client can auto-reconnect. Modify specifications during off-peak hours. • During the modification of cluster RabbitMQ 3.x.x instances, connections to the changed nodes will be switched to other nodes, posing overload risks (for example, connections excess or memory high watermark) on them. You are advised to use them within the instance specifications. For more information, see **Specifications**. • Persistent exchanges, queues, and messages are required. Otherwise, messages may be lost after node restarts.
ping command	• Single-node instances support the **ping** command with both private and public connection addresses. • Cluster instances only support the **ping** command with private connection addresses.

Virtual Host

Table 10-2 Constraint

Item	Constraint
Deleting a virtual host	The default virtual host created in instance creation cannot be deleted.
Vhost	• When a virtual host name of a RabbitMQ 3.x.x instance starts with a special character, such as a period (.), the monitoring data may not be displayed. When a virtual host name contains special characters such as percent (%), vertical bar (|), or slash (/), this name is not consistently displayed on the monitoring page. The special characters are displayed as underscores (_) instead. For example, virtual host **Vhost.1%1|2_3/** is displayed as **Vhost.1_1_2_3_** in monitoring. • When a virtual host name of a RabbitMQ AMQP-0-9-1 instance contains special characters such as percent (%), vertical bar (|), slash (/), and period (.), this name is not consistently displayed on the monitoring page. The special characters are displayed as underscores (_). For example, virtual host **Vhost.1%1|2_3/** is displayed as **Vhost_1_1_2_3_** in monitoring.

Exchange

Table 10-3 Notes and exchanges

Item	Constraint
Default exchange	For RabbitMQ 3.x.x instances, seven exchanges are created by default after virtual host creation. These exchanges include (AMQP default), amq.direct, amq.fanout, amq.headers, amq.match, amq.rabbitmq.trace, and amq.topic.
Binding an exchange	• For RabbitMQ 3.x.x, the exchange **(AMQP default)** cannot be bound with any exchange. • **Internal** exchanges can only be bound with exchanges and not queues. • In RabbitMQ AMQP-0-9-1, exchanges can only be bound with queues and not exchanges.
Deleting an exchange	For RabbitMQ 3.x.x, the default exchange cannot be deleted.

Item	Constraint
Exchange	When an exchange name of a RabbitMQ AMQP-0-9-1 instance contains special characters such as percent (%), vertical bar (|), slash (/), andperiod (.), this name is not consistently displayed on the monitoring page. The special characters are displayed as underscores (_). For example, exchange **Exchange.1%1|2_3/** is displayed as **Exchange_1_1_2_3_** in monitoring.

Queue

Table 10-4 Notes and constraints

Item	Constraint
Binding a queue	• For RabbitMQ 3.x.x, the exchange **(AMQP default)** cannot be bound with any queue. • **Internal** exchanges can only be bound with exchanges and not queues.
Lazy queues	Available for RabbitMQ 3.8.35 and later.
Quorum queues	Available for RabbitMQ 3.8.35 and later.
Single active consumer	Available for RabbitMQ 3.8.35 and later.
Queue	• When a queue name of a RabbitMQ 3.x.x instance starts with a special character, such as a period (.), the monitoring data may not be displayed. When a queue name contains special characters such as percent (%), vertical bar (|), and slash (/), this name is not consistently displayed on the monitoring page. The special characters are displayed as underscores (_) instead. For example, queue **Queue.1%1|2_3/** is displayed as **Queue.1_1_2_3_** in monitoring. • When a queue name of a RabbitMQ AMQP-0-9-1 instance contains special characters such as percent (%), vertical bar (|), slash (/), and period (.), this name is not consistently displayed on the monitoring page. The special characters are displayed as underscores (_). For example, queue **Queue.1%1|2_3/** is displayed as **Queue_1_1_2_3_** in monitoring.

Message

Table 10-5 Notes and constraints

Item	Constraint
Message size	≤ 50 MB per message for RabbitMQ 3.x.x; ≤ 4 MB per message for RabbitMQ AMQP-0-9-1. Do not send a message larger than 50 MB. Otherwise, the message will fail to be created.

11 Basic Concepts

Huawei Cloud DMS for RabbitMQ uses RabbitMQ as the messaging engine. In RabbitMQ, messages are sent by producers, stored in queues, and received by consumers. The following explains basic concepts of RabbitMQ.

Message

A message has a message body and a label. The message body, in JSON or other formats, contains the content of the message. The label only describes the message.

Messages are sent by producers and retrieved by consumers, but a producer and a consumer are not directly linked to each other.

Producer

A producer is an application that sends messages to queues. The messages are then delivered to other systems or modules for processing as agreed.

Consumer

A consumer is an application that receives messages. Consumers subscribe to queues. During routing, only the message body will be stored in the queue, so only the message body will be consumed by consumers.

Queue

A queue stores messages that are sent from producers and await retrievals by consumers. If different consumers subscribe to the same queue, the messages in that queue will be distributed across the consumers.

Broker

Nodes that provide message middleware services.

Virtual Host

Virtual hosts provide logical separation of exchanges, queues, and bindings. Different applications run on different virtual hosts without interfering with each other.

Exchange

Exchanges receive and assign messages. Producers send messages to exchanges first, rather than directly to queues. Exchanges route messages to one or more queues based on routing keys. If there are no matching queues, the messages are discarded. To learn about exchange types, see **Exchanges**.

12 Exchanges

Direct, fanout, topic, headers, x-delayed-message, and x-consistent-hash exchanges are available.

Direct Exchange

How It Works

1. A queue is bound to a direct exchange with a routing key.
2. The direct exchange routes a received message to the bound queue whose routing key is matched.

Routing

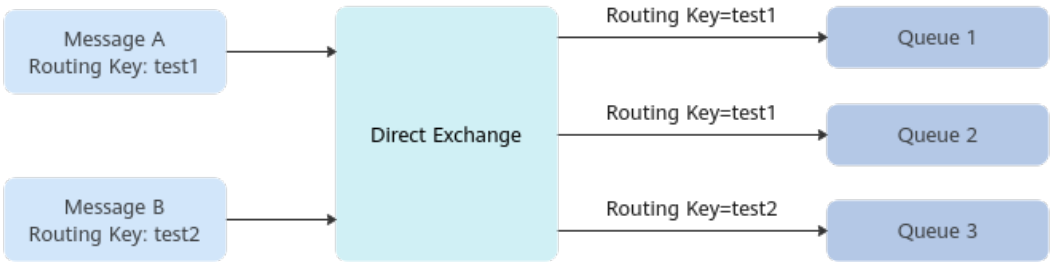
Based on a routing key matching

Scenario

Unicast routing

Example

Figure 12-1 Example direct exchange



As shown in this chart, Message A will be sent to Queue 1 and Queue 2. Message B will be sent to Queue 3.

Fanout Exchange

How It Works

A fanout exchange bound with multiple queues routes received messages to each queue. Fanout exchanges forward messages faster than other exchanges.

Routing

The fanout exchange delivers messages to all bound queues.

Scenario

Broadcast routing

Example

Figure 12-2 Example fanout exchange

As shown in this chart, Message A will be sent to Queue 1 and Queue 2.

Topic Exchange

How It Works

1. A queue is bound to a topic exchange with a routing key that includes a wildcard.
2. The topic exchange routes a received message to the queue if the message's routing key wildcard is matched.

Supported wildcards are stars (*****) and hashes (**#**). Separate wildcards and words by periods (.), for example, **test.#**.

- ***** matches one word.
- **#** matches zero or more words.

Routing

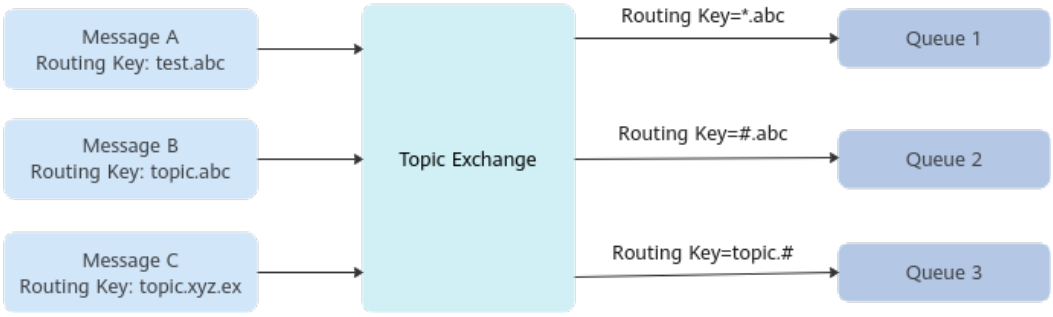
Based on a routing key wildcard matching

Scenario

Multicast routing

Example

Figure 12-3 Example topic exchange



As shown in this chart, Message A will be sent to Queue 1 and Queue 2, Message B to Queue 1, Queue 2, and Queue 3, and Message C to Queue 3.

Headers Exchange

How It Works

1. A queue is bound to a headers exchange with a binding expressed in a key-value pair.
2. The headers exchange routes a message to the queue if the binding matches the message's key-value header.

The matching algorithm uses a specific binding key-value pair, which is **x-match**. Values:

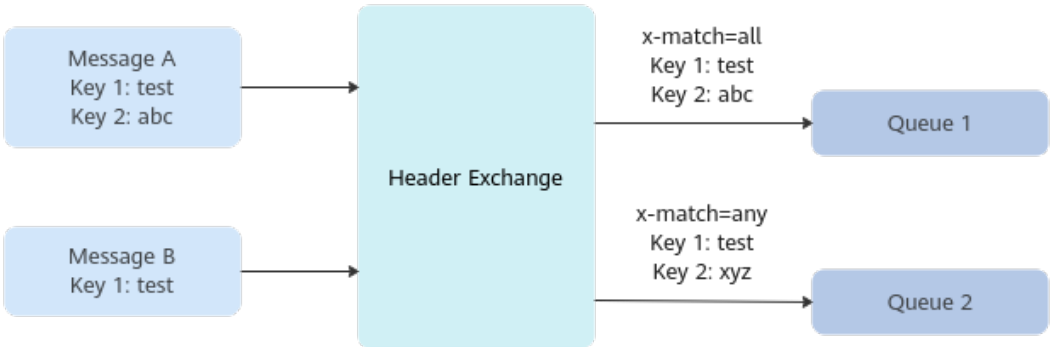
- **all**: Messages are routed only when all header pairs match.
- **any**: Messages are routed when any header pair matches.

Routing

Based on matching between key-value pairs in the message headers and the binding (a key-value pair)

Example

Figure 12-4 Example headers exchange



As shown in this chart, Message A will be sent to Queue 1 and Queue 2, and Message B to Queue 2.

x-delayed-message Exchange

How It Works

1. An x-delayed-message exchange is created with one of these exchange types: **Direct Exchange**, **Fanout Exchange**, **Topic Exchange**, and **Headers Exchange**. The exchange type specifies the routing rules.
2. A queue is bound to the x-delayed-message exchange.
3. A message with the header attribute **x-delay** is sent to the x-delayed-message exchange. **x-delay** specifies the message delivery schedule, in milliseconds.
4. The x-delayed-message exchange does not deliver the message immediately after receiving it. The message will be routed as scheduled based on the exchange type-specified routing rules.

Routing

Messages are routed based on the rules of the exchange type set in the x-delayed-message exchange creation.

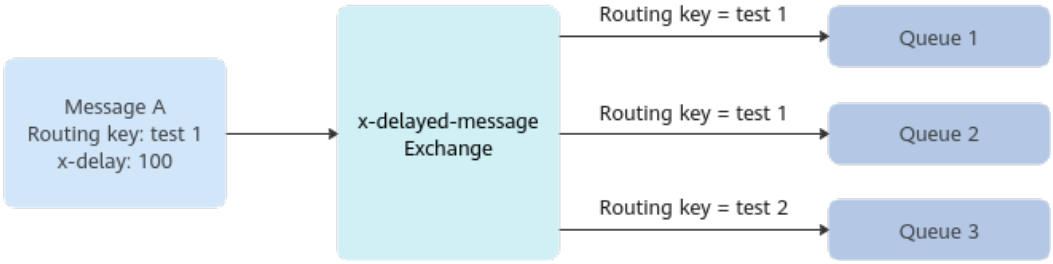
Scenario

x-delayed-message exchanges are applicable to scheduled message delivery.

Example

This example uses a **direct** exchange.

Figure 12-5 Example of an x-delayed-message exchange



As shown in the preceding figure, the x-delayed-message exchange receives message A, waits 100 ms, and sends it to queue 1 and queue 2.

x-consistent-hash Exchange

How It Works

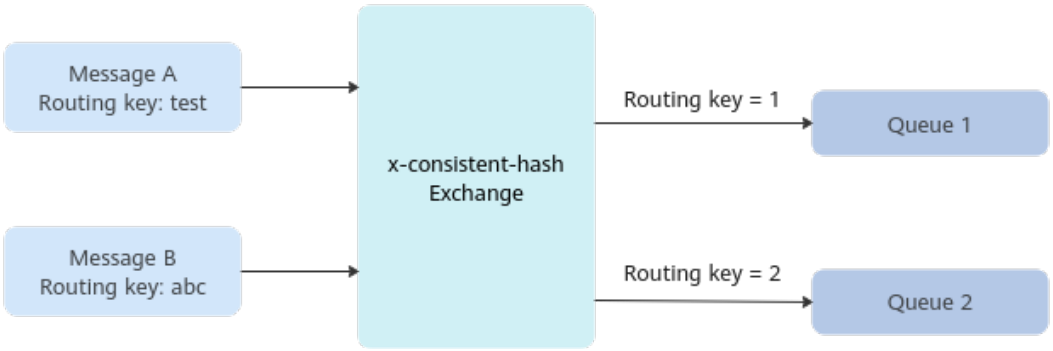
1. A queue is bound to an x-consistent-hash exchange with a routing key. The key must be a positive integer. A routing key indicates the weight of the queue. The larger the key, the heavier the weight, which means that the queue receives more messages.
2. A message with a routing key is sent to the x-consistent-hash exchange. The exchange calculates a hash value based on the key, and distributes the message to the hashed queue.

Routing

x-consistent-hash exchange calculates a hash value based on a routing key, and routes a message to the hashed queue.

Example

Figure 12-6 Example of an x-consistent-hash exchange



As shown in the preceding figure, x-consistent-hash exchange receives messages A and B with routing keys, calculates hash values based on the keys, and routes the messages to the hashed queues. Queue 1 weighs 1 and queue 2 weighs 2. In this case, queue 2 receives twice as many messages as queue 1.

13 Permissions Management

You can use Identity and Access Management (IAM) to manage DMS for RabbitMQ permissions and control access to your resources. IAM provides identity authentication, permissions management, and access control.

You can create IAM users for your employees, and assign permissions to these users on a principle of least privilege (PoLP) basis to control their access to specific resource types. For example, you can create IAM users for software developers and assign specific permissions to allow them to use DMS for RabbitMQ resources but prevent them from being able to delete resources or perform any high-risk operations.

If your HUAWEI ID does not require individual IAM users for permissions management, skip this section.

IAM is free of charge. You pay only for the resources you use. For more information, see **IAM Service Overview**.

DMS for RabbitMQ Permissions

By default, new IAM users do not have any permissions assigned. To assign permissions to these new users, add them to one or more groups, and attach permissions policies or roles to these groups.

DMS for RabbitMQ is a project-level service deployed and accessed in specific physical regions. When assigning DMS for RabbitMQ permissions to a user group, specify region-specific projects where the permissions will take effect. If you select **All projects**, the permissions will be granted for all region-specific projects. When accessing DMS for RabbitMQ, the users need to switch to a region where they have been authorized to use this service.

You can grant permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that provides only a limited number of service-level roles. When using roles to grant permissions, you also need to assign dependency roles. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for securer access control. For example, you can grant DMS for

RabbitMQ users only the permissions for managing a certain type of DMS for RabbitMQ instances. Most policies define permissions based on APIs. For the API actions supported by DMS for RabbitMQ, see **Permissions Policies and Supported Actions**.

NOTE

Permissions policies of DMS for RabbitMQ are based on DMS. Therefore, when assigning permissions, select DMS permissions policies.

Table 13-1 lists all the system-defined roles and policies supported by DMS for RabbitMQ.

Table 13-1 System-defined roles and policies supported by DMS for RabbitMQ

Role/Policy Name	Description	Type	Dependency
DMS FullAccess	Administrator permissions for DMS. Users granted these permissions can perform all operations on DMS.	System-defined policy	None
DMS UserAccess	Common user permissions for DMS, excluding permissions for creating, modifying, deleting, and scaling up instances.	System-defined policy	None
DMS ReadOnlyAccess	Read-only permissions for DMS. Users granted these permissions can only view DMS data.	System-defined policy	None
DMS VPCAccess	VPC operation permissions to assign to DMS agencies.	System-defined policies	None
DMS KMSAccess	KMS operation permissions to assign to DMS agencies.	System-defined policies	None
DMS ELBAccess	ELB operation permissions to assign to DMS agencies.	System-defined policies	None
DMS VPCEndpointAccess	VPC endpoint operation permissions to assign to DMS agencies.	System-defined policies	None
DMSAgencyCheckAccessPolicy	IAM operation permissions to assign to DMS agencies.	System-defined policies	None

Role/Policy Name	Description	Type	Dependency
DMS Administrator	Administrator permissions for DMS.	System-defined role	This role depends on the **Tenant Guest** and **VPC Administrator** roles.

Table 13-2 lists the common operations supported by each DMS for RabbitMQ system policy or role. Select the policies or roles as required.

Table 13-2 Common operations supported by system-defined policies

Operation	DMS FullAccess	DMS UserAccess	DMS ReadOnlyAccess	DMS VPCAccess	DMS KMSAccess	DMS ELBAccess	DMS VPCEndpoint Access	DMSAgency Check Access Policy
Creating an instance	√	×	×	×	×	×	×	×
Modifying instances	√	×	×	×	×	×	×	×
Deleting instances	√	×	×	×	×	×	×	×
Modifying instance specifications	√	×	×	×	×	×	×	×
Querying instance information	√	√	√	×	×	×	×	×

Helpful Links

- What Is IAM?
- Creating a User and Granting DMS for RabbitMQ Permissions
- Permissions Policies and Supported Actions

