ALIBABA CLOUD

Alibaba Cloud

Alibaba Cloud Message Queue for MQTT

Use Case

Document Version: 20220407

Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

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

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

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

Table of Contents

1.Real-time communication solution (MQTT)

Real-Time Communication solution is developed by Alibaba Cloud alibaba Cloud and Alibaba Real-Time Communication RTC jointly launched products to help you quickly build a variety of real-time communication scenarios, such as online audio and video conferencing and one-to-one voice call applications. This topic describes the system architecture, data flow design, and related matters for attention in detail.

Terms

Message Queuing Telemetry Transport (MQTT protocol)

The MQTT protocol is an industry-standard protocol for the Internet of Things (IoT) and mobile Internet, which is suitable for data transmission between mobile devices. The MQTT protocol is supported by default.

MQTT broker

The MQTT broker is used as a server node to interact with MQTT clients and respective message sending and receiving.

MQTT client

The MQTT client is the node that is used to interact with the MQTT broker. In this solution, it specifically indicates the audio/video mobile application that sends or receives audio/video calling requests.

P2P message

A special type of message that is provided based on the standard MQTT protocol. This type of message can be directly sent to a specified target MQTT client without subscription matching. For more information, see P2P Messaging model (MQTT).

Real-Time Communication (RTC)

A real-time network communication method for the voice and video fields. Currently, the mainstream use cases include voice calling, video calling, and video conferencing.

RTC server

This server hosts the audio/video and related media channel services provided by Alibaba Cloud RTC.

RTC service management server

These servers are the control nodes in the audio/video communication system, which are referred to as audio/video management services in this topic. The RTC management service must be manually created, and controls the lifecycles of all RTC sessions. The management node is normally deployed in the cloud, and is constructed using Alibaba Cloud products.

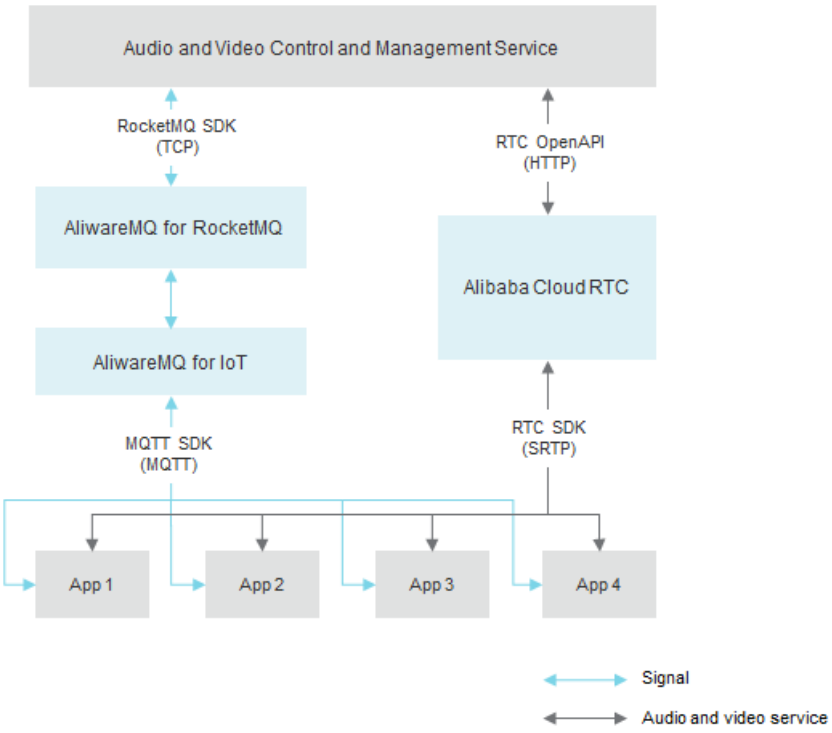
Audio/video mobile application

This is the application on the terminal that is used by the end user in the audio/video communication system, which is referred to as the terminal application in this topic. The user uses this application to initiate or participate in audio/video calls.

Solution architecture

Solution architecture shows the architecture of the audio/video communication solution.

Solution architecture



As shown in Solution architecture, the audio/video management service and the terminal application signal by using the AliwareMQ for RocketMQ, and implement service data interactions through Alibaba Cloud RTC. For more information, see Data interaction.

Highlights

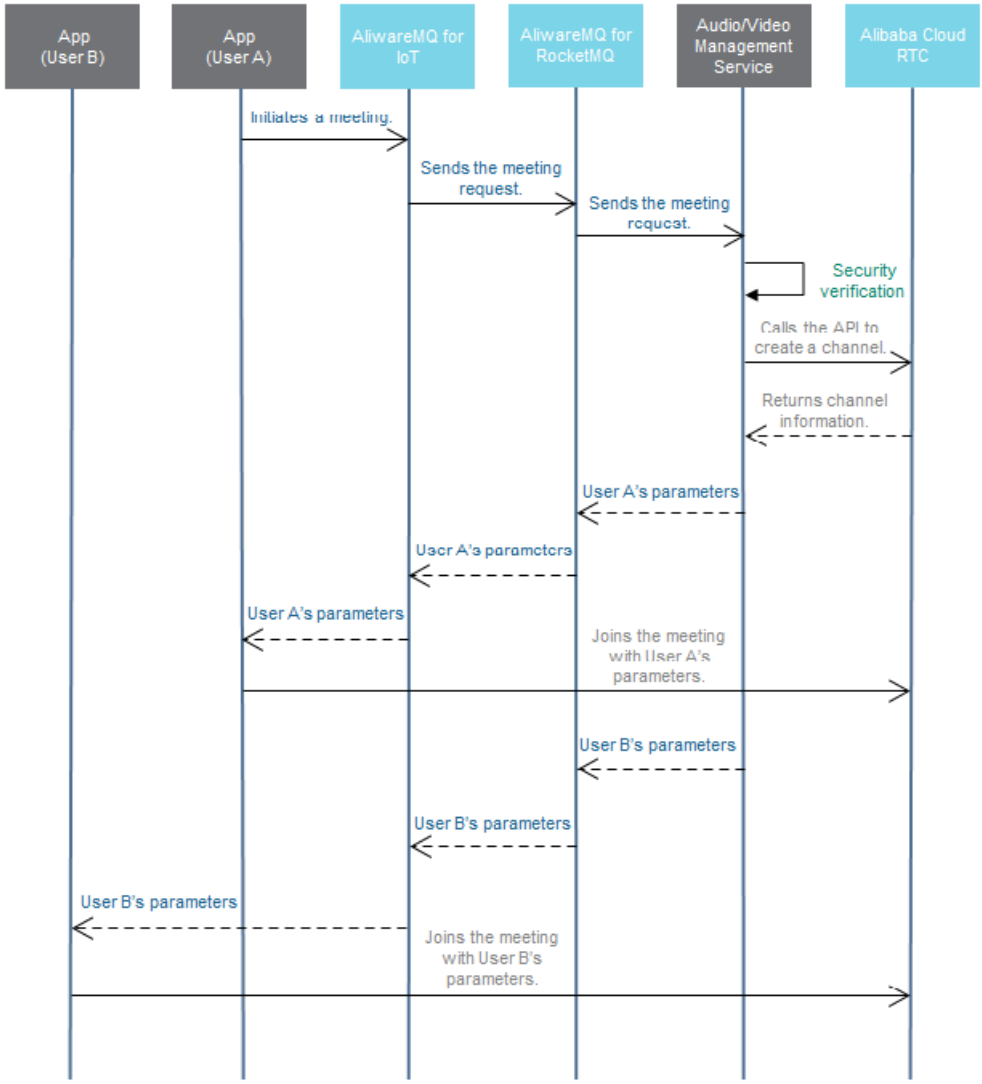
The advantages of the audio/video communication solution are as follows:

- Service capabilities are scalable.
  - and Alibaba Cloud RTC can be used on demand and dynamically scaled up to handle burst traffic peaks.
- Network coverage is widespread.
  - and Alibaba Cloud RTC provide global deployment capabilities to achieve local service access and save cross-zone and cross-nation costs.
- The construction period is short, supporting easy access.
  - The construction process is O&M-free, reducing labor and hardware costs.
  - The API is easy to use, supporting rapid implementation.
- High Reliability
  - All service nodes are highly available and stable.
  - supports SSL/TLS encryption and media streams support SRTP protection.

Data interaction

Data flows The presentation is based on and Real-Time Communication RTC to call an audio /video Conference. The gray part indicates your self-built development programs or services. The Blue part indicates , and Real-Time Communication RTC services.

Data flows



As shown in Data flows, User A invites User B to join an audio/video conference. The specific process is as follows:

1. User A of the terminal application initiates A meeting request and sends the request to the MQTT broker by sending an MQTT message. The Message passes through route to the audio /video management service developed by the business side processes the meeting request by receiving the message. After the verification, it calls the Alibaba Cloud RTC API Real-Time Communication register the resources and parameters for this communication.
2. After receiving the parameters, the audio /video management service encapsulates the parameters in an invitation message and sends the parameters to the , after the message is then delivered to the terminal application of user A. The terminal application of user A is added to the conference channel based on the parameters.

3. The audio/video management service also needs to find User B's information based on the information in User A's invitation. Similarly, the service also encapsulates the parameters into an invitation message, and the transfer process for User B is the same as that for User A described in Step 2.
4. The conference member User B joins the conference after receiving the parameters, and the communication initialization is completed.

Based on the preceding design ideas, you can use the implement other custom processes for sending messages. For example, you can destroy a meeting, invite others to join a meeting, and block a speech. plays the signaling role in audio/video conferencing scenarios.

Precautions

The preceding process briefly describes how to use and Real-Time Communication RTC to quickly build your own audio and video calling App. For more information about the SDK description, see Download the SDK, 版本说明 and Real-Time Communication (RTC) documents.

Where use when constructing a signaling channel for real-time communication, follow these principles for message type and parameter design:

- **MQTT client ID mapping**

  The MQTT protocol requires that each client have a globally unique client ID. A client ID consists of two parts concatenated with the @@@ separator. The final client ID must be unique and its total length cannot exceed 64 characters. The following describes the two parts of a client ID:

  - Prefix Group ID: The Group ID must be apply for the console. You are advised to classify the group IDs by application platform or channel to facilitate troubleshooting. For example, Android and iOS clients can be divided into different group IDs, or clients of different versions use different group IDs.
  - Device ID that is the suffix in a client ID: A device ID is generated by an application. Device IDs can be mapped to application account IDs to ensure their global uniqueness.

  For more information about client IDs, see Terms.

- **Topic name mapping**

  Use to send and receive messages, you must understand the publish /subscribe pattern of the MQTT Protocol. For more information, see agreement document and official documentation.

  The MQTT protocol is a message protocol based on the publish/subscribe pattern. The subscriptions and topics follow the directory tree format. Topics can be divided into parent topics and subtopics. The total length of a topic including the parent topic and subtopics cannot exceed 64 characters. The following describes the types of topics:

  - Parent topic: The topic at the first level of the directory tree is a parent topic. The parent Topic must be can be used only after applying for permissions in the console. A Namespace is equivalent to a Namespace.
  - Subtopic: A topic under a level-1 topic in the directory tree in MQTT is a subtopic. You can specify subtopics as needed without having to apply for them.

  For more information about topics, see Terms.

  When designing a topic for sending and receiving messages, you must comply with the following principles:

- Different parent topics are used for upstream messages (messages sent from the terminal application to the management service) and downstream messages (messages sent from the management service to the terminal application).
- Messages with different priorities or large size differences use different parent topics.

For the interaction process described above, we recommend that you use P2P messages do not need to be subscribed, and the sender can directly specify the peer end to receive them. For more information, see P2P Messaging model (MQTT).

- **Parameter design for message sending and receiving**

  Mobile applications may be killed in the background, making the mobile application offline. To handle this situation, you are advised to configure the terminal application as follows to ensure that the terminal application receives the previous message after it goes back online:

  - Set the CleanSession parameter to "false".
  - Set QoS to "1".

  The terminal application should perform deduplication and timeliness verification on the received messages (for example, if the terminal application remains offline for more than one day and then receives the messages from the previous day when it goes online again).

  For more information about CleanSession and QoS, see Terms.

2.New retail digital price tag solution (MQTT)

The new retail digital price tag solution is developed by Alibaba Cloud it is launched to realize the data update management of electronic tags and multimedia screens in shopping malls, supermarkets and public places through MQTT. This topic takes digital price tags as an example to describe the system architecture, data flow design, and other key components of the solution in detail. Other similar industries can refer to this solution, making modifications as necessary for implementation.

Terms

Message Queuing Telemetry Transport (MQTT protocol)
The MQTT protocol is an industry-standard protocol for the Internet of Things (IoT) and mobile Internet, which is suitable for data transmission between mobile devices. The MQTT protocol is supported by default.

MQTT broker
An MQTT broker is a broker node that interacts with the MQTT protocol and is used to receive and forward messages.

MQTT client
The MQTT client is the node used to interact with the MQTT broker. In this solution, it specifically indicates a smart access point for sending or receiving price change messages.

P2P message
A special type of message that is provided based on the standard MQTT protocol. This type of message can be directly sent to a specified target MQTT client without subscription matching. For more information, see P2P Messaging model (MQTT).

Smart access point (AP)
A common network device (such as a smart router) that supports application programming and can simultaneously handle Internet access and LAN device control functions.

Digital price tags
Electronic screens that are used in malls, supermarkets, and other places. They support networking based on wireless sensor network protocols (such as Bluetooth and ZigBee) as well as smart AP nodes for networking.

Digital price tag management service
The backend service of a digital price tag system. It is used to manage the content that is displayed on the electronic screens and to manage and query manual tasks, such as price changes.

ApsaraDB for RDS
A highly available and scalable online database service provided by Alibaba Cloud. It is used to make persistent the status changes of tasks (such as price changes) in a digital price tag system.

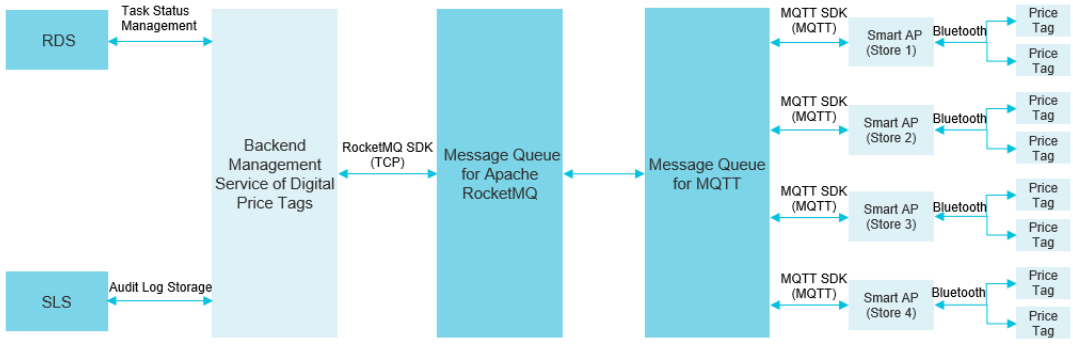
Log Service
The log storage service launched by Alibaba Cloud, which is used in a digital price tag system to

persistently store all operation logs for auditing and tracing.

Solution architecture

In digital price tag solutions, it can be used in conjunction with multiple Alibaba Cloud products to implement update management for price tag data.Solution architecture shows details on how the solution architecture of the digital price tag system works.

Solution architecture



Ru Solution architectureas shown in the figure, the digital price tag system mainly includes price tag nodes, smart AP nodes, , , the backend control service for digital price tags, RDS, and SLS. Each of these components is described as follows:

- The smart AP forwards the status data of the price tag and receives price change commands. The smart AP uses the MQTT SDK to access Alibaba Cloud over the public network based on the distribution of the stores or locations. This link uses SSL/TLS to encrypt transmissions, preventing data leaks.
- One smart AP downlink and several price tag nodes can communicate with each other through a wireless sensor network such as Bluetooth or ZigBee.
- The backend management service for digital price tags is deployed on the cloud based on ECS. The SDK and interaction.
- The backend management service for digital price tags can persistently change the status in the RDS databases when price changes and other tasks are performed. It can store price tag report data and operations logs to Log Service to facilitate tracing and auditing.

Highlights

The advantages of the new retail digital price tag solution are as follows:

- Powerful service capabilities that can be scaled automatically.
  - The message transmission capability is infinitely scalable, allowing you to increase the number of smart terminals without suffering from compromised system capabilities.
  - Information pushing in milliseconds is supported for millions of devices, with an even smaller latency for display updates of digital price tags.
- Extensive application range, versatile generability, and rapid duplication.
  - Based on the MQTT standard protocol, this solution is universally applicable. It can be replicated in similar scenarios by adapting the solution for different data content.
- High Reliability

- and smart AP nodes supports SSL/TLS encryption in data transmission, preventing media business data leakage.
- All service nodes are highly available and stable.

Data interaction

Status reporting

1. The digital price tag node uses a periodic polling mechanism to exchange data with the smart AP node, reporting its current display status, power capacity, and other information.
2. The smart AP node organizes data and sends MQTT messages to the MQTT broker.
3. The MQTT broker writes the reported message to the topic.
4. The digital price tag management service receives to process and analyze the status of online price tag nodes in the current system, and record the data to log service.

Update displays

1. The digital price tag management service sends a price change message, triggering the price change operation.
2. The MQTT Server routes the to push messages to the target smart AP node by using the MQTT protocol.
3. The smart AP node receives the price change notification and temporarily saves the task.
4. The digital price tag node uses the polling mechanism to exchange data with the smart AP node and receives the new content to be displayed.
5. After the target digital price tag node changes the price, the smart AP node returns an MQTT message to notify the digital price tag management service that the current task has been completed.
6. The digital price tag management service writes the execution log of the current task to Service Log, facilitating subsequent tracing queries.

Precautions

The preceding process briefly describes how to use and to build the digital price tag system, please refer to Download the SDK and 版本说明 document.

Where use and when designing messages and parameters for transmitting commands, follow these principles:

- **SDK and protocol selection**

  In the digital price tag scenario, one application may be used by hundreds of offline stores. Generally, each store is equipped with several smart AP nodes. The number of smart AP nodes can be increased as the business scales out. This makes smart AP nodes suitable for access by using the MQTT Protocol. The digital price tag management service is deployed on the cloud, access.

- **MQTT client ID mapping**

  The MQTT protocol requires that each client have a globally unique client ID. A client ID consists of two parts concatenated with the @@@ separator. The final client ID must be unique and its total length cannot exceed 64 characters. The following describes the two parts of a client ID:

- Prefix Group ID: The Group ID must be apply for the console. Group IDs can be roughly classified by platform vendor or channel to facilitate troubleshooting. For example, different industries or batches can be divided into different group IDs, or clients of different versions can use different group IDs.
- Device ID that is the suffix in a client ID: A device ID is generated by an application. Device IDs can be encoded by using the unique information, such as the MAC address of the smart AP node.

For more information about client IDs, see Terms.

- **Topic name mapping**

  Use to send and receive messages, you must understand the publish /subscribe pattern of the MQTT Protocol. For more information, see agreement document and official documentation.

  The MQTT protocol is a message protocol based on the publish/subscribe pattern. The subscriptions and topics follow the directory tree format. Topics can be divided into parent topics and subtopics. The total length of a topic including the parent topic and subtopics cannot exceed 64 characters. The following describes the types of topics:

  - Parent topic: The topic at the first level of the directory tree is a parent topic. The parent Topic must be can be used only after applying for permissions in the console. A Namespace is equivalent to a Namespace.
  - Subtopic: A topic under a level-1 topic in the directory tree in MQTT is a subtopic. You can specify subtopics as needed without having to apply for them.

  For more information about topics, see Terms.

  When designing a topic for sending and receiving messages, you must comply with the following principles:

  - Different types of tasks use different parent topics. For example, in this scenario, the price change tasks and the terminal status reporting tasks use different parent topics.
  - In the digital price tag system, we recommend that you use P2P messages do not need to be subscribed, and the sender can directly specify the peer end to receive them. For more information, see P2P Messaging model (MQTT).

- **Parameter design for message sending and receiving**

  Generally, price change tasks in the digital price tag scenario require real-time pushing. Therefore, we recommend that you configure the smart AP as follows during the interactions between the smart AP and the MQTT broker to ensure that the smart AP does not need to process the tasks that were pushed when it was disconnected:

  - cleanSession set the parameter to "tru e" .
  - Set QoS to 1 ".

  The smart AP should perform deduplication and timeliness verification on received messages.

  For more information about CleanSession and QoS, see Terms.

3.Unified management of Cloud Storage Gateway (CSG) instances based on MQTT

By A Unified Framework can be used to manage Cloud Storage Gateway Cloud Storage Gateway (CSG) across different deployment modes. This ensures secure and compliant penetration between public and private networks.

Terms

Message Queuing Telemetry Transport (MQTT protocol)
The MQTT protocol is an industry-standard protocol for the Internet of Things (IoT) and mobile Internet, which is suitable for data transmission between mobile devices. The MQTT protocol is supported by default.

CSG
CSG is a gateway service that can be deployed in your on-premises data center and in Alibaba Cloud. CSG uses Alibaba Cloud Object Storage Service (OSS) buckets as back-end storage devices. Moreover, by using low-cost virtual machines, CSG provides on-premises and off-premises applications with standard file storage services over the Network Attached Storage (NFS) and Common Internet File Systems (CIFS) protocols, and block storage services over the Internet Small Computer Systems Interface (iSCSI) protocol. For more information, see Cloud Storage Gateway.

MQTT broker
An MQTT broker is a broker node that interacts with the MQTT protocol and is used to receive and forward messages.

MQTT client
An MQTT client is a mobile node that interacts with an MQTT broker. In this topic, it typically refers to a client that uses CSG.

ApsaraDB for RDS
A highly available and scalable online database service provided by Alibaba Cloud.

Log Service
Alibaba Cloud Log Service is used for auditing and tracing.

Background information

CSG is a storage service that helps you seamlessly integrate existing on-premises applications, IT infrastructure, and data storage with Alibaba Cloud. You can deploy virtual devices compatible with standard storage protocols on the premises and in the cloud. In this way, you can seamlessly connect on-premises storage applications and workload to Alibaba Cloud storage and computing services.

CSG can be deployed in the following two modes:

- Deploy and run CSG in your on-premises data center through virtual machines.
- Deploy and run CSG in your Elastic Compute Service (ECS) instance in the virtual network environment

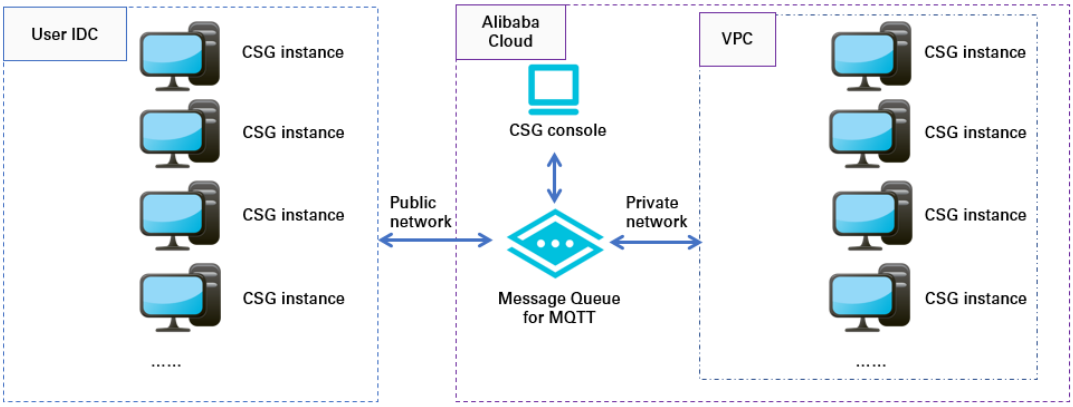
of Alibaba Cloud.

Under the two deployment modes, CSG instances use the intranet, especially virtual machines deployed in your on-premises data center. To ensure security, the CSG console cannot directly access non-public IP addresses. The communication of CSG instances deployed in Alibaba Cloud with the Alibaba Cloud console through public network IP addresses is also inappropriate because this may cause security hazards and vulnerability to attacks.

enables the connection between the CSG console and CSG instances under these two deployment modes.

For more information about the architecture of CSG, see. Hybrid cloud storage deployment Shown in.

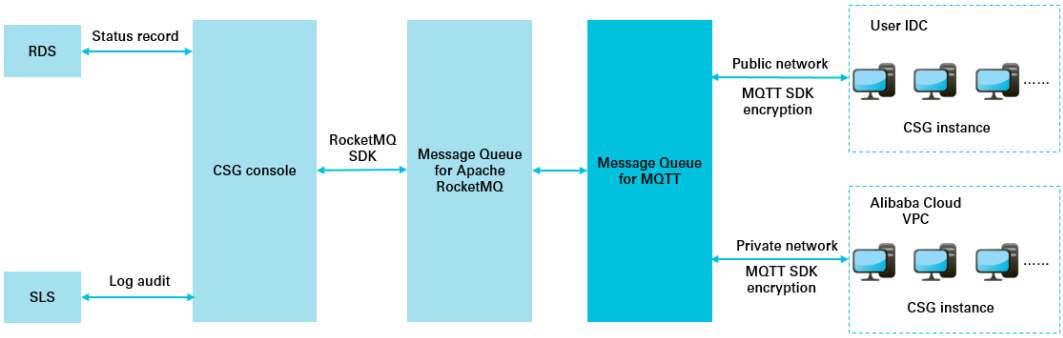
Hybrid cloud storage deployment



Solution architecture

Message Queue for MQTT (MQTT) is used to push and collect messages. It can display the statuses of CSG instances in the CSG console by giving commands and collecting these statuses through message pushing and active pulling. Based on in addition, the output API is used to manage each CSG instance. Solution architecture shows the overall architecture:

Solution architecture



Highlights

The following describes the benefits of this scheme.

- Powerful service capabilities that can be automatically scaled are available.
- The message transmission capability is infinitely scalable, allowing you to increase the number of CSG instances without compromising system capabilities.
- Messages are pushed to more than a million devices in milliseconds, and no congestion occurs to control commands issued by CSG.

- Encryption based on the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols is supported to relieve you from the worry about data leakage.
- All service nodes are highly available and stable.
- Multi-language SDKs make it convenient to develop MQTT brokers and applications.
- Both the internal network and public network support different network requirements.

Precautions

The preceding process briefly describes how to use and A unified cross-network management architecture is implemented. For more information about the SDK, see Download the SDK and 版本说明 document.

Where use when constructing cross-network signaling, follow these principles for message type and parameter design:

- **MQTT client ID mapping**

  The MQTT protocol requires that each client have a globally unique client ID. A client ID consists of two parts concatenated with the @@@ separator. The final client ID must be unique and its total length cannot exceed 64 characters. The following describes the two parts of a client ID:

  - Prefix Group ID: The Group ID must be apply for the console. We recommend that you classify your group IDs based on the Alibaba Cloud regions where the CSG console resides. For example, you can use different group IDs for **China (Hangzhou)** and **China (Shanghai)** so that problem regions and message queues can be easily located.
  - Device ID that is the suffix in a client ID: A device ID is generated by an application. Device IDs can be mapped to the IDs of CSG instances one by one to ensure global uniqueness.

  For more information about client IDs, see Terms.

- **Topic name mapping**

  Use to send and receive messages, you must understand the publish /subscribe pattern of the MQTT Protocol. For more information, see agreement document and official documentation.

  The MQTT protocol is a message protocol based on the publish/subscribe pattern. The subscriptions and topics follow the directory tree format. Topics can be divided into parent topics and subtopics. The total length of a topic including the parent topic and subtopics cannot exceed 64 characters. The following describes the types of topics:

  - Parent topic: The topic at the first level of the directory tree is a parent topic. The parent Topic must be can be used only after applying for permissions in the console. A Namespace is equivalent to a Namespace.
  - Subtopic: A topic under a level-1 topic in the directory tree in MQTT is a subtopic. You can specify subtopics as needed without having to apply for them.

  For more information about topics, see Terms.

  When designing a topic for sending and receiving messages, you must comply with the following principles:

  - Different parent topics are used for uplink messages and downlink messages. Uplink messages are messages sent from CSG instances to management and control services. Downlink messages are messages sent from management and control services to CSG instances.
  - Messages with different priorities or large size differences use different parent topics.

For the interaction process described above, we recommend that you use P2P messages provided by. P2P messages do not need to be subscribed, and therefore the sender can directly specify the peer end to receive them. For more information, see P2P Messaging model (MQTT).

- **Parameter design for message sending and receiving**

  The CSG console uses MQTT to simulate RPC calls. Therefore, the messages sent by the console directly use noreply without return values. Previous messages do not need to be received again. After receiving the error code, the console prompts you to issue the command again. You need to configure your MQTT client as follows:

  - Set the cleanSession parameter to true.

  For more information about cleanSession , see Terms.