

# TDMQ for CKafka

## Security and Compliance

### Product Documentation



## Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## Security and Compliance

- Permission Management

- Network Security

- Deletion Protection

- Event Record

- CloudAudit

# Security and Compliance

## Permission Management

Last updated: 2026-01-20 17:19:22

TDMQ for CKafka (CKafka) provides a comprehensive enterprise-level security protection system. Through root account/sub-account management and strict authorization and authentication mechanisms, it builds multi-layered and all-round security protection, ensuring reliable protection for each stage in message transmission and comprehensively safeguarding data security.

### Control Plane Permissions (Account-Level)

Cross-account authorization services between root accounts/sub-accounts and across enterprises are achieved through root accounts/sub-accounts, collaborators, and other features of Cloud Access Management (CAM). In addition, account access key management can be used to control cloud resources called using APIs.

### Identity Authentication

To access CKafka resources through the console or by calling cloud APIs, identity authentication is required, and resources can be accessed after authentication is successful.

- Logging in to the console: The login password needs to be verified, and login protection and login verification policies are provided to enhance identity authentication security. For detailed information, see [Changing the Login Password](#) and [Setting Login Protection](#).
- Calling cloud APIs: The AccessKey needs to be verified. AccessKeys are security credentials used for identity authentication when users access TencentCloud APIs, which consist of SecretId and SecretKey. For detailed information, see [Account AccessKey Management](#).

### Access Control

Through CAM, fine-grained permission management for TDMQ for CKafka resources can be implemented at the account level.

- User and permission assignment: Based on the enterprise organizational structure, independent users or roles are created for members of different functional departments, and dedicated security credentials (such as the console login password and cloud API key) or temporary credentials are assigned to ensure secure and controlled access to CKafka resources.
- Fine-grained permission control: Set differentiated access policies based on employee responsibilities to precisely control the types of operations each user or role can perform and the scope of resources they can access, achieving strict permission isolation.

For detailed introduction and operation methods, see [Account Permission Management Overview](#).

## Data Plane Permissions (Resource-Level)

CKafka provides dual-layer security protection through Simple Authentication and Security Layer (SASL) authentication and access control lists (ACLs). SASL verifies user identities, while ACLs enable fine-grained management of topic read/write permissions, ensuring access isolation at the resource level.

### Identity Authentication

SASL is a security protocol used for identity authentication, supporting two verification mechanisms:

- **PLAIN mechanism:** uses simple authentication where usernames and passwords are transmitted in plain text.
- **SCRAM mechanism:** uses hash algorithms to securely authenticate usernames and passwords between the server and client. CKafka supports two SCRAM encryption algorithms with different security strengths: SCRAM-SHA-256 and SCRAM\_SHA\_512.

CKafka performs authentication through the SASL protocol. After SASL authentication is enabled, only authenticated users can access CKafka resources.

### Access Control

ACL policies enable resource-level access control by customizing user settings in the console and configuring rules, such as allowing or denying specific users to read or write topic resources based on IP addresses. By combining user identities with ACL policies, CKafka enforces isolation of production and consumption permissions at the topic level, enhancing access control for both public and private network transmissions. For detailed introductions and operation methods, see [Configuring Topic Read/Write Permissions](#).

# Network Security

Last updated: 2026-01-20 17:19:22

CKafka supports both private and public network access. For different network types, CKafka provides multiple security protection mechanisms to ensure data transmission security.

Security Mechanism	Mechanism Description	Supported by the VPC Network or Not	Supported by the Public Network or Not	Reference Documentation
Bind to Security Group	A virtual firewall with stateful data packet filtering feature, used to set network access control for instances to control inbound and outbound traffic at the instance level.	✓	×	<a href="#">Configuring VPC</a>
ACL policies	By customizing user settings and configuring similar policies: allow/deny user user to read/write Topic resources via IP address, using the dual restrictions of "user + policy", to achieve isolation of production/consumption permissions at the Topic level, enhancing user access control during public/private network transmission.	✓	✓	<a href="#">Configuring Topic Read/Write Permissions</a>
SSL encryption	A data transmission security protocol that employs encryption technology to ensure data is not stolen or tampered with during transmission, effectively enhancing communication security.	✓	✓	<a href="#">Configuring Custom SSL Certificates</a>

For more detailed information, see [Network Connectivity Specifications](#).

# Deletion Protection

Last updated: 2026-01-20 17:19:22

TDMQ for CKafka (CKafka) provides a multi-dimensional instance deletion protection mechanism that prevents accidental deletion of instances through the console or APIs, ensuring business data security.

Operation	Protection Mechanism	Description	Reference Documentation
Pre-deletion protection	Deletion protection	<ul style="list-style-type: none"> <li>After instance deletion protection is enabled, the instance cannot be deleted through the console or an API. To delete the instance, you need to manually disable the deletion protection feature first. It is recommended to keep this feature enabled for key businesses.</li> <li>Instance deletion protection does not apply to system-level deletion. For example, postpaid by hour instances are isolated and subsequently released due to overdue payment, and yearly/monthly subscription instances are isolated and released after expiration.</li> </ul>	<a href="#">Configuring Instance Deletion Protection</a>
Deletion verification	Multi-factor authentication (MFA)	<ul style="list-style-type: none"> <li>MFA is a simple and effective security authentication method. It adds an extra layer of security beyond usernames and passwords.</li> <li>MFA is enabled by default for CKafka instance deletion. To delete an instance, you should complete identity verification through WeChat QR code and mobile Captcha to ensure the action is authorized by the account owner.</li> </ul>	<a href="#">MFA Operation Protection</a>
Post-deletion retention period	Isolation period retention	<ul style="list-style-type: none"> <li>After a yearly/monthly subscription instance is deleted, the instance will be retained in the console for 7 days in an isolated status. Isolated instances cannot produce or consume data, but the data and configurations stored in the CKafka instance will not be deleted. Expired messages will continue to be automatically deleted according to the Apache Kafka mechanism. During this period, you can manually renew the instance. After renewal, the instance will recover to its normal status and become fully functional.</li> </ul>	<a href="#">Deleting/Returning an Instance</a>

- Pay-as-you-go instances cannot be recovered after deletion. Proceed with caution.

# Event Record

Last updated: 2026-01-20 17:19:22

Event Center in CKafka supports centralized management, storage, analysis, and visualization of various Ops events, diagnosis events, and broker change events that occur during instance operation, facilitating future querying, auditing, and tracing. It also supports event alarm capabilities. You can configure alarm rules for key events (such as node offline or disk expansion failure) on TCOP, so that Ops personnel can handle them promptly.

For details on event types and query methods, see [View Event Records](#).

# CloudAudit

Last updated: 2026-01-20 17:19:22

CloudAudit is a service that supports monitoring, compliance checks, operation audits, and risk audits for your Tencent Cloud account. With CloudAudit, you can record logs and continuously monitor and retain operation-related account activities in Tencent Cloud Data Center Operating System (DCOS).

CloudAudit provides event history for Tencent Cloud account activities. These activities include operations performed through the Tencent Cloud console, API services, Tencent Cloud Command Line Interface (TCCLI), and other Tencent Cloud services. The event history can simplify security analysis, resource change tracking, and troubleshooting.

- For detailed information about CloudAudit and its activation and configuration methods, see [Getting Started with CloudAudit](#).
- For the list of CKafka operation events supported by CloudAudit, see the [list of CKafka operations supported by CloudAudit](#).