

Cloud Trace Service

User Guide

Issue	01
Date	2025-10-13



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI 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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Service Overview.....	1
1.1 What Is Cloud Trace Service.....	1
1.2 Basic Concepts.....	2
1.3 How CTS Functions.....	4
1.4 Application Scenarios.....	5
1.5 Billing.....	6
1.6 Permissions.....	6
1.7 Notes and Constraints.....	9
2 Getting Started.....	11
2.1 Overview.....	11
2.2 Viewing CTS Traces in the Trace List.....	12
2.3 Querying Transferred Traces.....	14
3 Management Trackers.....	17
3.1 Creating a Tracker.....	17
3.2 Configuring a Tracker.....	17
3.3 Disabling or Enabling a Tracker.....	20
3.4 Deleting a Tracker.....	21
4 Data Trackers.....	23
4.1 Creating a Tracker.....	23
4.2 Configuring a Tracker.....	25
4.3 Disabling or Enabling a Tracker.....	28
4.4 Deleting a Tracker.....	29
5 Creating a Key Event Notification.....	31
6 Application Examples.....	35
6.1 Security Auditing.....	35
6.2 Fault Locating.....	36
6.3 Resource Tracking.....	37
7 Trace References.....	39
7.1 Trace Structure.....	39
7.2 Example Traces.....	44
8 Cross-Tenant Transfer Authorization.....	48

9 Verifying Trace File Integrity..... 51

9.1 Enabling Verification of Trace File Integrity..... 51

9.2 Digest Files..... 51

9.3 Verifying Trace File Integrity..... 55

10 Auditing.....60

11 Supported Services and Operations..... 62

12 Permissions Management..... 64

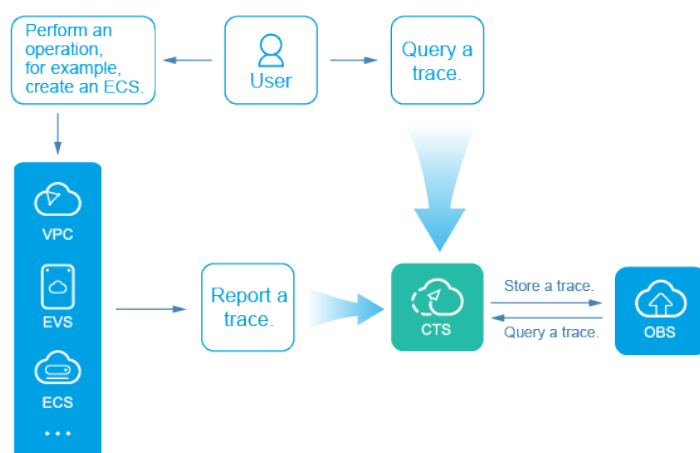
1 Service Overview

1.1 What Is Cloud Trace Service

The log audit module is a core component necessary for information security audit and an important part for the information systems of enterprises and public institutions to provide security risk management and control.

Cloud Trace Service (CTS) is a log audit service for security. It allows you to collect, store, and query resource operation records. You can use these records to perform security analysis, track resource changes, audit compliance, backtrack problems, and locate faults.

Figure 1-1 CTS service diagram



CTS provides the following functions:

- Trace recording: CTS records operations performed on the management console or by calling APIs, as well as operations triggered by each interconnected service.

- Trace query: You can query operation records of the last seven days on the CTS console using filters such as trace type, trace source, resource type, filter, operator, and trace status.
- Trace transfer: Traces can be transferred to Object Storage Service (OBS) buckets or Log Tank Service (LTS) log streams periodically. During transfer, traces are compressed into trace files by service.
- Trace file encryption: Trace files are encrypted using keys provided by Data Encryption Workshop (DEW) during transfer.
- Key event notification: CTS works with Simple Message Notification (SMN) to send notifications to your mobile phones and email addresses to notify you of certain key operations.

CTS records:

- Operations performed on the management console.
- Operations performed by calling supported APIs.
- Operations triggered by connected cloud services.

1.2 Basic Concepts

Trackers

When you enable CTS for the first time, a management tracker named **system** is created automatically.

The management tracker identifies and associates with all cloud services your tenant account is using, and records all operations of your tenant account.

Currently, only one tracker can be created for each tenant.

Traces

Traces are operation logs of cloud service resources and are captured and stored by CTS. You can view traces to get to know details of operations performed on specific resources.

Traces: management traces reported by cloud services.

Trace List

The trace list displays traces generated in the last seven days. These traces record operations (in the last hour by default) on cloud service resources, including creation, modification, and deletion, but do not record query operations.

- Management traces record details about creating, modifying, and deleting cloud service resources in your cloud account.

Trace Files

A trace file is a collection of traces. CTS generates trace files based on services and transfer cycle and send these files to your specified OBS bucket in real time. In most cases, all traces of a service generated in a transfer cycle are compressed

into one trace file. However, if there are a large number of traces, CTS will adjust the number of traces contained in each trace file.

Traces files are in JSON format. The following is an example of a trace file.

Figure 1-2 Trace file example

```
[{"time": 1491482532828,
 "user": {
   "id": "59f40829165447fb9470b56f41dff599",
   "name": " ",
   "domain": {
     "name": " ",
     "id": "0f27bc42d1eb46a69482a72cbfc33ed2"
   }
 },
 "request": {
   "bucket_name": "obs-570f",
   "file_prefix_name": "-RsU",
   "status": "disabled"
 },
 "response": {
   "bucket_name": "obs-570f",
   "file_prefix_name": "-RsU",
   "status": "disabled",
   "tracker_name": "system"
 },
 "service_type": "CTS",
 "resource_type": "tracker",
 "resource_name": "system",
 "source_ip": " ",
 "trace_name": "updateTracker",
 "trace_type": "ConsoleAction",
 "api_version": "1.0",
 "record_time": 1491482532857,
 "trace_id": "7519ef09-lac6-11e7-8cc0-3d812829baf6",
 "trace_status": "normal"
 },
 {
   "time": 1491482535203,
   "user": {
     "id": "59f40829165447fb9470b56f41dff599",
     "name": " ",
     "domain": {
       "name": " ",
       "id": "0f27bc42d1eb46a69482a72cbfc33ed2"
     }
   },
   "request": {
     "bucket_name": "obs-570f",
     "file_prefix_name": "-RsU",
     "status": "enabled"
   },
   "response": {
     "bucket_name": "obs-570f",
     "file_prefix_name": "-RsU",
     "status": "enabled",
     "tracker_name": "system"
   },
   "service_type": "CTS",
   "resource_type": "tracker",
   "resource_name": "system",
   "source_ip": " ",
   "trace_name": "updateTracker",
   "trace_type": "ConsoleAction",
   "api_version": "1.0",
   "record_time": 1491482535224,
   "trace_id": "76831bfb-lac6-11e7-98ff-a1036f244dcd",
   "trace_status": "normal"
 }
 ]]
```

Verifying Trace File Integrity

The authenticity of operation records during a security incident investigation is often affected by trace files being deleted or tampered with. The records therefore cannot be used as an effective basis for investigation. Therefore, CTS provides trace file integrity verification to help you ensure the authenticity of trace files.

The verification function for trace file integrity adopts industry standard algorithms and generates a Hash value for each trace file. This Hash value changes when the trace file is modified or deleted. Therefore, by tracking the Hash value, you can confirm whether the trace file is modified. In addition, the RSA algorithm is used to sign on the digest file to ensure that the file is not modified. In this way, any operations of modifying or deleting trace files are recorded by CTS.

After the verification function for trace file integrity is enabled, CTS generates a digest file for Hash values of all trace files recorded in the past hour and synchronizes the digest file to an OBS bucket configured for the current tracker.

CTS signs on each digest file using public and private keys. You can verify the digest file using the public key after the file is stored to the OBS bucket.

1.3 How CTS Functions

CTS connects to other cloud services on the cloud platform, records operations on cloud resources and the results, and stores these records in the form of trace files to OBS buckets in real time.

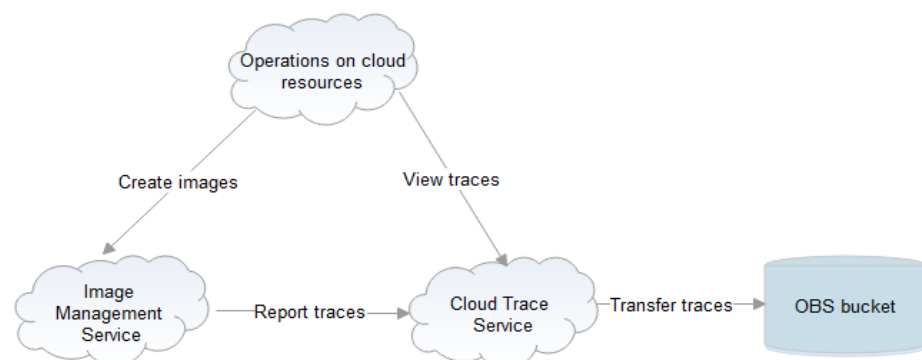
You can use CTS to create trackers to record trace files. If trace transfer has been configured, trace files will be stored in the OBS bucket that you have specified.

You can perform the following operations on a trace file:

- Trace file creation and storage
 - When you add, delete, or modify resources on services interconnected with CTS, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), and Image Management Service (IMS), the target services will record the operations and their results automatically and deliver them in the form of trace files to CTS for archiving.
 - Operation records of the last seven days are displayed on the CTS console. If trace transfer has been enabled, operation records are periodically delivered to the OBS bucket that you have specified for long-term storage.
- Trace file query
 - You can query operation records in the last seven days on the **Trace List** page by time and other filters.
 - To query operation records earlier than seven days, you can download the trace files stored in OBS buckets if trace transfer has been configured.
 - You can enable, disable, configure, or delete a tracker on the **Tracker List** page.

For example, if you create an image using IMS, the service will report the creation operation to CTS. Then, CTS will deliver the trace to an OBS bucket for storage if trace transfer has been configured. You can view trace files in the trace list. [Figure 1-3](#) shows the working principle of CTS.

Figure 1-3 How CTS functions



1.4 Application Scenarios

CTS provides operation records on cloud service resources. A record contains the user who performed the operation, IP address, operation content, and returned response message. With these records, you can better conduct auditing, plan and use resources, and identify operations of high risks or that violate regulations.

CTS can be used in the following scenarios.

- **Compliance auditing**

CTS helps you obtain certifications for auditing in industry standards, such as PCI DSS and ISO 27001, for your service systems. CTS allows you to query all operation records of security control. This is essential for enterprises and organizations, especially financial and payment enterprises, to achieve certification.

If you want to migrate your services to the cloud, you will need to ensure the compliance of your own service systems, and the cloud vendor you choose will need to ensure the compliance of your service systems and resources.

CTS plays an important role in compliance. The service records operations of almost all services and resources, and carries out security measures such as encryption, disaster recovery, and anti-tampering to ensure the integrity of traces during their transmission and storage. In addition, you can use CTS to design and implement solutions that help you obtain compliance certifications for your service systems.

- **Key event notifications**

CTS works with FunctionGraph to send notifications to recipients, including natural persons or service APIs, upon the execution of key operations. The following are real application examples:

You can configure HTTP or HTTPS notifications targeted at your independent systems and synchronize traces received by CTS to your own audit systems for auditing.

You can also select a certain type of log (such as file upload) as a trigger for a preset workflow (for example, file format conversion) in FunctionGraph, simplifying service deployment and O&M as well as preventing risks.

- **Data value mining**

CTS mines data in traces to facilitate service health analysis, risk analysis, resource tracking, and cost analysis. You can also obtain the data from CTS and explore the data value yourself.

A trace contains extensive information, including the time, operator, operation device IP address, operated resource, and operation details. Each trace is worth mining.

By configuring HTTP or HTTPS notifications, you can synchronize traces to your own system for analysis. In addition, CTS is connected to Cloud Eye and Log Tank Service (LTS) to help you monitor high-risk operations, detect unauthorized operations, and analyze resource usage.

- **Fault locating and analysis**

You can configure filters to pinpoint the faulty operation and its details when a fault occurs, reducing the time and workforce required for detecting, locating, and fixing faults.

If a specific resource or action encounters a fault, you can query operation records on the resource in a specific time period and view the requests and responses to facilitate fault locating.

CTS provides the following search dimensions: trace type, trace source, resource type, filter, operator and trace status. Each trace contains the request and response of an operation. Querying traces is one of the most efficient methods for locating a fault.

If a problem occurs on the cloud, you can configure filters to search for all suspicious operations in a specified time period. You can then synchronize the relevant traces to O&M and customer service personnel who will handle the problem.

1.5 Billing

You can use the basic functions of CTS for free, including enabling a tracker, tracking traces, as well as storing and querying traces of the last seven days. In addition, CTS works with other services to provide you with value-added functions such as trace file transfer. These functions may generate fees in other cloud services, but the fees are usually low. Use the value-added functions as needed.

Value-added functions:

- **Trace transfer:** You can configure a tracker to transfer trace files to OBS buckets. Trace files transferred by the management tracker are permanently stored.
- **Trace analysis:** This function is provided by CTS and is free to use. However, it depends on log storage of Log Tank Service (LTS), which may generate fees.
- **Key event notification:** CTS provides the key event notification function to send notifications to your mobile phones and email addresses when specific operations are performed. You need to subscribe to topics on the Simple Message Notification (SMN) console for this function to take effect.

1.6 Permissions

If you need to grant your enterprise personnel permission to access your CTS resources, use Identity and Access Management (IAM). IAM provides identity authentication, fine-grained permissions management, and access control. IAM helps you secure access to your resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users 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 CTS resources but prevent them from deleting resources or performing any high-risk operations.

If your account does not require IAM users for permissions management, you may skip this section.

IAM is a free service. You only pay for the resources in your account. For details, see *IAM Service Overview*.

CTS 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.

CTS is a project-level service deployed for specific regions. To assign CTS permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing CTS, the users need to switch to a region where they have been authorized to use this service.

You can grant users 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. Roles are not an ideal choice for fine-grained authorization and secure access control.
- **Policies:** A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization for more secure access control. For example, you can grant ECS users only the permissions for managing a certain type of ECSs. Most policies define permissions based on APIs.

For the API actions supported by CTS, see [Table 1-1](#).

Table 1-1 System-defined roles and policies supported by CTS

Role/ Policy Name	Description	Type	Dependency
CTS FullAccess	Full permissions for CTS.	System-defined policy	None
CTS ReadOnlyAccess	Read-only permissions for CTS.	System-defined policy	None

Role/ Policy Name	Description	Type	Dependency
CTS Administrator	Administrator permissions for CTS. Users granted these permissions can perform all operations on CTS. Users with this permission can perform read-only operations on all services except IAM.	System-defined role	This role must be used together with the Tenant Guest , OBS Administrator , and Security Administrator roles in the same project.

Table 1-2 lists the common operations supported by each system-defined policy or role of CTS. Select the policies or roles as required.

Table 1-2 Common operations supported by system-defined policies or roles

Operation	CTS FullAccess	CTS ReadOnlyAccess	CTS Administrator
Querying traces	√	√	√
Querying quotas	√	√	√
Creating a tracker	√	×	√
Modifying a tracker	√	×	√
Disabling a tracker	√	×	√
Enabling a tracker	√	×	√
Querying a tracker	√	√	√
Deleting a tracker	√	×	√
Creating a key event notification	√	×	√
Modifying a key event notification	√	×	√
Disabling a key event notification	√	×	√
Enabling a key event notification	√	×	√

Operation	CTS FullAccess	CTS ReadOnlyAccess	CTS Administrator
Querying a key event notification	√	√	√
Deleting a key event notification	√	×	√
Adding tags in batches	√	×	√
Deleting tags in batches	√	×	√

Custom Permissions Policies

You can create custom permissions policies to supplement the system-defined policies.

- For the actions that can be configured in custom permissions policies, see "Permissions Policies and Supported Actions" in *CTS API Reference*.
- For details, see "Creating a Custom Policy" in the *IAM User Guide*.

1.7 Notes and Constraints

There are fixed quotas on the number of trackers and key event notifications in CTS.

Table 1-3 CTS constraints

Item	Constraint
Maximum number of trackers that can be created by a single cloud account	Management tracker: 1
Maximum number of key event notifications that can be configured by a single cloud account	100
Maximum number of OBS buckets that can be configured for a tracker	1
Time from when an operation is performed to when the operation data can be queried on the console	Management traces: 1 minute

Item	Constraint
Time span of trace query on the console • By default, CTS retains traces for each cloud account for seven days. To store traces older than seven days, you need to transfer them to other cloud services. • These traces are automatically deleted from the CTS console upon expiration after seven days. Manual deletion is not supported.	Seven days

2 Getting Started

2.1 Overview

Scenarios

You need to enable CTS before using it. A management tracker named **system** is automatically created when CTS is enabled. All traces recorded by CTS are associated with the tracker.

You can only query operation records of the last seven days on the CTS console. To query operation records generated in the past seven days, store trace files in an OBS bucket or Log Tank Service (LTS) log stream. Ensure that you have enabled OBS and LTS and have full permissions for the OBS bucket and LTS log stream you are going to use. By default, only the owner of OBS buckets can access the buckets and all objects contained in the buckets, but the owner can grant access permissions to other services and users by configuring access policies.

Prerequisites

- To configure the trace transfer function, you must enable OBS and LTS.
- To enable the key event notification function, you must enable Simple Message Notification (SMN).

Associated Services

- OBS: used to store trace files.
- Data Encryption Workshop (DEW): Provides keys that can be used to encrypt trace files.
- LTS: stores logs.
- SMN: Sends email or SMS message notifications to users when key operations are performed.

Enabling CTS for the First Time

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
- Step 3** Choose **Tracker List** in the navigation pane on the left and click **Enable CTS** in the upper right corner. A management tracker named **system** will be automatically created.

 NOTE

The management tracker logs user operations like creation, login, and deletion on all cloud service resources. For details about the cloud services supported by CTS, see [Supported Services and Operations](#).

- Step 4** Choose **Tracker List** in the navigation pane to view operation records of the last seven days.

----End

2.2 Viewing CTS Traces in the Trace List

Scenarios

Cloud Trace Service (CTS) records operations performed on cloud service resources. A record contains information such as the user who performed the operation, IP address, operation content, and returned response message. These records facilitate security auditing, issue tracking, and resource locating. They also help you plan and use resources, and identify high-risk or non-compliant operations.

What Is a Trace?

A trace is an operation log for a cloud service resource, tracked and stored by CTS. Traces record operations such as adding, modifying, or deleting cloud service resources. You can view them to identify who performed operations and when for detailed tracking.

Viewing Traces in the Trace List

- Step 1** Log in to the management console, click  in the upper left corner, and choose **Management & Deployment > Cloud Trace Service**.
- Step 2** In the navigation pane, choose **Trace List**.
- Step 3** In the upper right corner of the page, set a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also click **Customize** to specify a custom time range within the last seven days.
- Step 4** Set filters to search for your desired traces.

Table 2-1 Trace filtering parameters

Parameter	Description
Trace Type	Select Management or Data . - Management traces record operations performed by users on cloud service resources, including creation, modification, and deletion. - Data traces are reported by OBS and record operations performed on data in OBS buckets, including uploads and downloads.
Trace Source	Select the name of the cloud service that triggers a trace from the drop-down list.
Resource type	Select the type of the resource involved in a trace from the drop-down list. For details about the resource types of each cloud service, see section "Supported Services and Operations" in the <i>Cloud Trace Service User Guide</i> .
Search By	Select one of the following options: Resource ID: ID of the cloud resource involved in a trace. Leave this field empty if the resource has no resource ID or if resource creation failed. Trace name: name of a trace. For details about the operations that can be audited for each cloud service, see section "Supported Services and Operations" in the <i>Cloud Trace Service User Guide</i>. Resource name: name of the cloud resource involved in a trace. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of trace_type in a trace is SystemAction , the operation is triggered by the service and the trace's operator may be empty.
Trace Status	Select one of the following options: Normal: The operation succeeded. Warning: The operation failed. Incident: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.

Step 5 Click **Query**.

Step 6 On the **Trace List** page, you can also export and refresh the trace list.

- Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5,000 records.
- Click  to view the latest information about traces.

Step 7 Click  on the left of a trace to expand its details.

Trace Name	Resource Type	Trace Source	Resource ID	Resource Name	Trace Status	Operator	Operation Time	Operation
login	user	IAM	179b57d16904412691c74ab58...		normal		Jul 3, 2024 11:29:32 GMT+08:00	View Trace
trace_id 0b4e8f1-38ac-11ef-929c-81030b655029 code 302 trace_name login resource_type user trace_rating normal message ["login":{"mode":"password","user_type":"domain owner","login_protect":{"status":"off"}}] source_ip domain_id 38e0cca0987 trace_type ConsoleAction								

Step 8 Click **View Trace** in the **Operation** column. The trace details are displayed.

View Trace

```
{
  "request": "",
  "trace_id": "",
  "code": "200",
  "trace_name": "createDockerConfig",
  "resource_type": "dockerlogincmd",
  "trace_rating": "normal",
  "api_version": "",
  "message": "createDockerConfig, Method: POST Url=/v2/manage/utills/secret, Reason:",
  "source_ip": "",
  "domain_id": "",
  "trace_type": "ApiCall",
  "service_type": "SWR",
  "event_type": "system",
  "project_id": "",
  "response": "",
  "resource_id": "",
  "tracker_name": "system",
  "time": "Nov 16, 2023 10:54:04 GMT+08:00",
  "resource_name": "dockerlogincmd",
  "user": {
    "domain": {
      "name": "",
      "id": ""
    }
  }
}
```

----End

Helpful Links

- For details about the key fields in the trace structure, see section "Trace References" > "Trace Structure" in the *Cloud Trace Service User Guide* and section "Trace References" > "Example Traces" in the *Cloud Trace Service User Guide*.

2.3 Querying Transferred Traces

Scenarios

CTS periodically sends trace files to OBS buckets. A trace file is a collection of traces. CTS generates trace files based on services and transfer cycle, and adjusts the number of traces contained in each trace file as needed. CTS can also save audit logs to LTS log streams.

This section describes how to view historical operation records in trace files downloaded from OBS buckets and in LTS log streams.

Prerequisites

You have configured a tracker in CTS and enabled **Transfer to OBS** or **Transfer to LTS**. For details about how to configure transfer, see [Configuring a Tracker](#).

Querying Traces Transferred to OBS

If you enable **Transfer to OBS** when configuring the tracker, traces will be periodically transferred to a specified OBS bucket as trace files for long-term storage.

1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
3. Choose **Tracker List** in the navigation pane.
4. Click a bucket in the **OBS Bucket** column.
5. In the OBS bucket, locate the file storage path to view the desired trace, and click **Download** on the right to download the file to the default download path of the browser. If you need to save it to a custom path, click **More > Download As** on the right.
 - The trace file storage path is as follows:
OBS bucket name/CloudTraces/Region/Year/Month/Day/Tracker name/Cloud service
Example: **User Define/CloudTraces/Region/2016/5/19/system/ECS**
 - The trace file naming format is as follows:
Trace file prefix_CloudTrace_Region_Year-Month-DayTHour-Minute-SecondZ_Random characters.json.gz (Year-Month-DayTHour-Minute-Second indicates the time when the trace file was uploaded to OBS.)
Example:
FilePrefix_CloudTrace_region_2024-12-13T01-29-19Z_47b9d51830deff47.json.gz

NOTE

- The OBS bucket name and trace file prefix are set by you and other parameters are automatically generated.
- File download will incur request fees and traffic fees.

For details about key fields in a trace, see [Trace Structure](#) and [Example Traces](#).

6. Decompress the downloaded package to obtain a JSON file with the same name as the package, as shown in [Figure 2-1](#). Open the JSON file using a text file editor to view traces.

Figure 2-1 JSON file

```
{
  "code": 200,
  "event_type": "system",
  "project_id": "4008a952b3f44b5a919c9a48d90811f3",
  "record_time": 1723533697290,
  "resource_name": "-",
  "resource_type": "bucket",
  "service_type": "OBS",
  "source_ip": "172.30.33.69",
  "time": 1723533697290,
  "trace_id": "ab0472a4-5944-11ef-acee-294fee19871b",
  "trace_name": "listAllMyBucket",
  "trace_rating": "normal",
  "trace_type": "Others",
  "tracker_name": "system",
  "user": "{\"name\":\"\", \"id\":\"5f2cd06722f24250976264e6e7753a08\", \"domain\":\"\", \"name\":\"\", \"id\":\"25fe78d91e0448f6a37f35427c6a420b\"}"
}
```

Querying Traces Transferred to LTS

If you enable **Transfer to LTS** when configuring a tracker, traces will be transferred to the **CTS/{Tracker Name}** log stream for long-term storage. *{Tracker Name}* indicates the name of the current tracker. For example, the log stream path of the management tracker is **CTS/system-trace**.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 3 Choose **Tracker List** in the navigation pane.

Step 4 Click an LTS log stream in the **Storage** column.

Step 5 On the **Log Stream** tab page in the CTS log group page, select the *{Tracker name}* log stream to view trace logs.

For details about key fields in a trace, see [Trace Structure](#) and [Example Traces](#).

Step 6 Click  to download the log file to your local PC.

----End

3 Management Trackers

The management tracker records management traces, which are operations on all cloud resources, such as creation, login, and deletion.

This section describes how to use the management tracker.

3.1 Creating a Tracker

If you log in to CTS for the first time, click **Enable CTS** on the **Tracker List** page. A management tracker named **system** will be automatically created. The management tracker identifies and associates with all cloud services your tenant account is using, and records all operations of your tenant account.

Constraints

- CTS records operations performed in the last seven days. To store traces for a longer period, configure a tracker to transfer them to OBS or LTS. The tracker will then continuously transfer traces to your specified OBS bucket or LTS log stream for storage.
- CTS can have only one management tracker. The stored historical traces are retained even after the management tracker is deleted. When you enable CTS again, the management tracker is restored.

3.2 Configuring a Tracker

Scenarios

On the CTS console, you can add configurations such as OBS or LTS transfer for the created management tracker.

You can select whether to send recorded traces to an OBS bucket for long-term storage.

After the tracker configuration is complete, CTS will immediately start recording operations under the new settings.

This section describes how to configure the management tracker.

Prerequisites

You have enabled CTS.

Configuring a Management Tracker

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Tracker List** in the navigation pane.

Step 5 Click **Configure** in the **Operation** column in the row of the management tracker.

Step 6 Configure the basic information of the tracker, and click **Next**.

Parameter	Description
Tracker Name	The default value is system and cannot be changed.
Enterprise Project	Select an enterprise project. NOTE Enterprise projects allow you to manage cloud resources and users by project. For details about how to enable the enterprise project function, see "Creating an Enterprise Project" in <i>Enterprise Management User Guide</i> .

Step 7 On the transfer configuration page, configure the transfer parameters of the tracker. You can only query operation records of the last seven days on the CTS console. To store and query operation records beyond seven days, transfer them to OBS or LTS. For details, see [Table 3-1](#) and [Table 3-2](#).

Table 3-1 Parameters for configuring the transfer to OBS

Parameter	Description
Transfer to OBS	Select an existing OBS bucket or select New to create one, and set File Prefix . When Transfer to OBS is disabled, no operation is required.
OBS Bucket	Existing: Select an existing OBS bucket in the current region.

Parameter	Description
Select Bucket	If you select New for OBS Bucket, enter a name for the new OBS bucket. The bucket name cannot be empty. Enter 3 to 63 characters, including only lowercase letters, digits, hyphens (-), and periods (.). It cannot contain two consecutive periods (for example, my..bucket). A period (.) and a hyphen (-) cannot be adjacent to each other (for example, my-.bucket and my.-bucket). Do not use an IP address as a bucket name. If you select Existing for OBS Bucket, select an existing OBS bucket.
Retention Period	For the management tracker, the retention period configured on the OBS console is used by default and cannot be changed.
File Prefix	A file prefix is used to mark transferred trace files. The prefix you set will be automatically added to the beginning of the file names, facilitating file filtering. Enter 0 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
Compression	The usage of object storage space can be reduced. • Do not compress: Transfer files in the *.json format. • gzip: Transfer files in *.json.gz format.
Sort by Cloud Service	• When this function is enabled, the cloud service name is added to the transfer file path, and multiple small files are generated in OBS. • When this function is disabled, the cloud service name will not be added to the transfer file path.
Transfer Path	Log transfer path is automatically set by the system.

Table 3-2 Parameters for configuring the transfer to LTS

Parameter	Description
Transfer to LTS	When Transfer to LTS is enabled, traces are transferred to the log stream.
Log Group	When Transfer to LTS is enabled, the default log group name CTS is set. When Transfer to LTS is disabled, no operation is required.

Step 8 Click **Next > Configure** to complete the configuration of the management tracker.

You can then view the tracker details on the **Tracker List** page.

 NOTE

Traces recorded by CTS are delivered periodically to the OBS bucket for storage. If you configure an OBS bucket for a tracker, traces generated during the current cycle (usually several minutes) will be delivered to the configured OBS bucket. For example, if the current cycle is from 12:00:00 to 12:05:00 and you configure an OBS bucket for a tracker at 12:02:00, traces received from 12:00:00 to 12:02:00 will also be delivered to the configured OBS bucket for storage at 12:05:00.

Step 9 (Optional) On the **Tracker List** page, click  in the **Tag** column to add tags to the tracker.

Tags are key-value pairs, which are used to identify, classify, and search for trackers. Tracker tags are used to filter and manage trackers only. A maximum of 20 tags can be added to a tracker.

If your organization has configured tag policies for CTS, add tags to trackers based on the policies.

Table 3-3 Tag parameters

Parameter	Description	Example
Tag key	A tag key of a tracker must be unique. You can customize a key or select the key of an existing tag created in Tag Management Service (TMS). A tag key: • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>_.:=-@</code>, but cannot start or end with a space or start with <code>_sys_</code>.	Key_0001
Tag value	A tag value can be repetitive or left blank. A tag value: • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>_.:/=-@</code>	Value_0001

----End

3.3 Disabling or Enabling a Tracker

Scenarios

You can enable or disable a tracker on the CTS console. Disabling a tracker does not affect existing operation records.

This section describes how to enable or disable a tracker.

Constraints

After a tracker is disabled, traces can still be reported to CTS. You can view traces of the last seven days in the trace list. However, new traces recorded after you disable the tracker cannot be viewed and transferred to OBS or LTS, and key event notifications will not be sent.

Prerequisites

You have enabled CTS.

Disabling or Enabling the Management Tracker

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Tracker List** in the navigation pane.

Step 5 Click **Disable** in the **Operation** column in the row of the management tracker.

Step 6 Click **OK**. After the tracker is disabled, the **Disable** button changes to **Enable**.

Step 7 To enable the management tracker again, click **Enable** and then click **OK**. The tracker will start recording operations again.

----End

3.4 Deleting a Tracker

Scenarios

You can delete the management tracker on the CTS console. Deleting it does not affect the existing operation records. This section describes how to delete the management tracker on the console.

Constraints

After a tracker is deleted, traces can still be reported to CTS. You can view traces of the last seven days in the trace list. However, new traces recorded after you disable the tracker cannot be viewed and transferred to OBS or LTS, and key event notifications will not be sent.

Enable CTS again to restore the management tracker.

Prerequisites

You have enabled CTS.

Deleting a Management Tracker

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Tracker List** in the navigation pane.

Step 5 Click **Delete** in the **Operation** column of the management tracker.

Step 6 Click **OK**.

----End

4 Data Trackers

CTS provides two types of trackers: a management tracker and multiple data trackers. The management tracker records management traces, which are operations on all cloud resources, such as creation, login, and deletion. Data trackers record data traces, that is, logs of tenant operations (such as upload and download) on data in OBS buckets.

This section describes how to use a data tracker.

4.1 Creating a Tracker

Scenarios

You can create data trackers to log operations on data. Data trackers record data traces, that is, logs of tenant operations (such as upload and download) on data in OBS buckets.

When you enable CTS, a management tracker is created automatically. Only one management tracker can be created. The trackers you created are all data trackers.

Constraints

- CTS records operations performed in the last seven days. To store traces for a longer period, configure your tracker. The tracker will continuously store traces to your specified LTS log stream or OBS bucket.

Prerequisites

You have enabled CTS.

Creating a Data Tracker

1. Log in to the management console.
2. In the service list, choose > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Tracker List** in the navigation pane. In the upper right corner of the displayed page, click **Create Tracker**.

4. Set basic information. Enter a tracker name. Click **Next**.
A data tracker name contains only letters, digits, hyphens (-), and underscores (_), and must start with a letter or digit. It cannot be empty and contains up to 32 characters. Do not use **system** or **system-trace** as a data tracker name.
5. On the **Select Trace** page, set the parameters and click **Next**.

Table 4-1 Parameters for selecting trace objects

Parameter	Description
Data Trace Source	Container for storing data traces. OBS is selected by default.
OBS Bucket	Select an OBS bucket from the drop-down list. After the tracker is created, the OBS bucket name cannot be changed.
Operation	Data operations are classified into read and write operations. Read operations refer to obtaining or downloading object data from OBS buckets. Write operations refer to uploading or writing object data to OBS buckets. • If you select Read, CTS will record only read operations. • If you select Write, CTS will record only write operations.

6. On the transfer configuration page, set related parameters and click **Next**.
You can only query operation records of the last seven days on the CTS console. To store and query operation records beyond seven days, transfer them to OBS or LTS. For details, see [Table 4-2](#) and [Table 4-3](#).

Table 4-2 Parameters for configuring the transfer to OBS

Parameter	Description
Transfer to OBS	If Transfer to OBS is enabled, select an existing OBS bucket and set File Prefix . When Transfer to OBS is disabled, no operation is required.
OBS Bucket	Existing : Select an existing OBS bucket in the current region.
Select Bucket	When you select New , enter an OBS bucket name. The OBS bucket name cannot be empty. It can contain 3 to 63 characters, including only lowercase letters, digits, hyphens (-), and periods (.). It cannot contain two consecutive periods (for example, my..bucket). A period (.) and a hyphen (-) cannot be adjacent to each other (for example, my-.bucket and my.-bucket). Do not use an IP address as a bucket name. If you select Existing for OBS Bucket , select an existing OBS bucket.

Parameter	Description
Retention Period	The duration for storing traces in the OBS bucket. This configuration will apply to the selected bucket and all files in it. Different compliance standards require different trace retention periods. You are advised to set the retention period to at least 180 days. For a data tracker, you can set the duration to 30 days, 60 days, 90 days, 180 days, 3 years, or the same as that of OBS.
File Prefix	A file prefix is used to mark transferred trace files. The prefix you set will be automatically added to the beginning of the file names, facilitating file filtering. Enter 0 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
Verify Trace File	When this function is enabled, integrity verification will be performed to check whether trace files in OBS buckets have been tampered with. For details about file integrity verification, see Verifying Trace File Integrity .

Table 4-3 Parameters for configuring the transfer to LTS

Parameter	Description
Transfer to LTS	When Transfer to LTS is enabled, traces are transferred to the log stream.
Log Group	When Transfer to LTS is enabled, the default log group name CTS is set. When Transfer to LTS is disabled, no operation is required.

7. Preview the tracker information and click **Create**.

4.2 Configuring a Tracker

Scenarios

On the CTS console, you can add configurations such as OBS or LTS transfer for the created data trackers.

You can select whether to send recorded traces to an OBS bucket for long-term storage.

After the tracker configuration is complete, CTS will immediately start recording operations under the new settings.

This section describes how to configure a data tracker.

Prerequisites

You have enabled CTS and created a data tracker.

Configuring a Data Tracker

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose > **Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Tracker List** in the navigation pane.

Step 5 Click **Configure** in the **Operation** column in the row of the target data tracker.

Step 6 On the **Select Trace** page, set the parameters and click **Next**.

Table 4-4 Parameters for selecting traces

Parameter	Description
Data Trace Source	Container for storing data traces. OBS is selected by default.
OBS Bucket	The OBS bucket you set when creating the data tracker is select by default. This option cannot be changed.
Operation	Data operations are classified into read and write operations. Read operations refer to obtaining or downloading object data from OBS buckets. Write operations refer to uploading or writing object data to OBS buckets. - If you select Read, CTS will record only read operations. - If you select Write, CTS will record only write operations.

Step 7 On the transfer configuration page, modify the transfer parameters of the tracker. You can only query operation records of the last seven days on the CTS console. To store and query operation records beyond seven days, transfer them to OBS or LTS. For details, see [Table 4-5](#) and [Table 4-6](#).

Table 4-5 Parameters for configuring the transfer to OBS

Parameter	Description
Transfer to OBS	If Transfer to OBS is enabled, select an existing OBS bucket and set File Prefix . When Transfer to OBS is disabled, no operation is required.
OBS Bucket	Existing: Select an existing OBS bucket in the current region.

Parameter	Description
Select Bucket	When you select New, enter an OBS bucket name. The OBS bucket name cannot be empty. It can contain 3 to 63 characters, including only lowercase letters, digits, hyphens (-), and periods (.). It cannot contain two consecutive periods (for example, my..bucket). A period (.) and a hyphen (-) cannot be adjacent to each other (for example, my-.bucket and my.-bucket). Do not use an IP address as a bucket name. If you select Existing for OBS Bucket, select an existing OBS bucket.
Retention Period	The duration for storing traces in the OBS bucket. This configuration will apply to the selected bucket and all files in it. Different compliance standards require different trace retention periods. You are advised to set the retention period to at least 180 days. For a data tracker, you can set the duration to 30 days, 60 days, 90 days, 180 days, 3 years, or the same as that of OBS.
File Prefix	A file prefix is used to mark transferred trace files. The prefix you set will be automatically added to the beginning of the file names, facilitating file filtering. Enter 0 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and periods (.) are allowed.
Verify Trace File	When this function is enabled, integrity verification will be performed to check whether trace files in OBS buckets have been tampered with. For details about file integrity verification, see Verifying Trace File Integrity.

Table 4-6 Parameters for configuring the transfer to LTS

Parameter	Description
Transfer to LTS	When Transfer to LTS is enabled, traces are transferred to the log stream.
Log Group	When Transfer to LTS is enabled, the default log group name CTS is set. When Transfer to LTS is disabled, no operation is required.

Step 8 Click **Next > Configure** to complete the configuration of the data tracker.

You can then view the tracker details on the **Tracker List** page.

 NOTE

Traces recorded by CTS are delivered periodically to the OBS bucket for storage. If you configure an OBS bucket for a tracker, traces generated during the current cycle (usually several minutes) will be delivered to the configured OBS bucket. For example, if the current cycle is from 12:00:00 to 12:05:00 and you configure an OBS bucket for a tracker at 12:02:00, traces received from 12:00:00 to 12:02:00 will also be delivered to the configured OBS bucket for storage at 12:05:00.

Step 9 (Optional) On the **Tracker List** page, click  in the **Tag** column to add tags to the tracker.

Tags are key-value pairs, which are used to identify, classify, and search for trackers. Tracker tags are used to filter and manage trackers only. A maximum of 20 tags can be added to a tracker.

If your organization has configured tag policies for CTS, add tags to trackers based on the policies.

Table 4-7 Tag parameters

Parameter	Description	Example
Tag key	A tag key of a tracker must be unique. You can customize a key or select the key of an existing tag created in Tag Management Service (TMS). A tag key: • Can contain 1 to 128 characters. • Can contain letters, digits, spaces, and special characters <code>_.:=-@</code>, but cannot start or end with a space or start with <code>_sys_</code>.	Key_0001
Tag value	A tag value can be repetitive or left blank. A tag value: • Can contain 0 to 255 characters. • Can contain letters, digits, spaces, and special characters <code>_.:/=-@</code>	Value_0001

----End

4.3 Disabling or Enabling a Tracker

Scenarios

You can disable a tracker on the CTS console. After a tracker is disabled, it will stop recording operations, but you can still view operation records that have been collected.

Constraints

After a tracker is disabled, traces can still be reported to CTS. You can view traces of the last seven days in the trace list. However, new traces recorded after you disable the tracker cannot be viewed and transferred to OBS or LTS, and key event notifications will not be sent.

Prerequisites

You have created a data tracker on the CTS console.

Disabling or Enabling a Data Tracker

1. Log in to the management console.
2. Click  in the upper left corner to select the desired region and project.
3. Click  in the upper left corner and choose > **Cloud Trace Service**. The CTS console is displayed.
4. Choose **Tracker List** in the navigation pane.
5. Click **Disable** in the **Operation** column in the row of the target data tracker.
6. Click **OK**. After the tracker is disabled, the **Disable** button changes to **Enable**.
7. To enable the tracker, click **Enable** and then click **OK**. The tracker will start recording operations again.

4.4 Deleting a Tracker

Scenarios

Deleting a data tracker on the CTS console is available, and does not affect the existing operation records. This section describes how to delete a data tracker on the console.

Constraints

After a tracker is deleted, traces can still be reported to CTS. You can view traces of the last seven days in the trace list. However, new traces recorded after you disable the tracker cannot be viewed and transferred to OBS or LTS, and key event notifications will not be sent.

Prerequisites

A data tracker has been created.

Deleting a Data Tracker

1. Log in to the management console.
2. Click  in the upper left corner to select the desired region and project.

3. Click  in the upper left corner and choose > **Cloud Trace Service**. The CTS console is displayed.
4. Choose **Tracker List** in the navigation pane.
5. Click **Delete** in the **Operation** column of the target configuration item.
6. Click **OK**.

5 Creating a Key Event Notification

You can create key event notifications on CTS so that SMN sends you SMS, email, or HTTP/HTTPS notifications of key events. This function is triggered by CTS, and notifications are sent by SMN. SMN sends key event notifications to subscribers. Before setting notifications, you need to know how to create topics and add subscriptions on the SMN console.

Scenarios

You can use this function for:

- Real-time detection of high-risk operations (such as VM restart and security configuration changes), cost-sensitive operations (such as creating and deleting expensive resources), and service-sensitive operations (such as network configuration changes).
- Detection of operations such as login of users with admin-level permissions or operations performed by users who do not have the required permissions.
- Connection with your own audit system: You can synchronize all audit logs to your audit system in real time to analyze the API calling success rate, unauthorized operations, security, and costs.

Constraints

- SMN sends key event notifications to subscribers. Before setting notifications, you need to know how to create topics and add subscriptions on the SMN console.
- You can create up to 100 key event notifications on CTS:
 - Specify key operations, users, and topics to customize notifications.
 - Complete key event notifications can be sent to notification topics.
- If CTS and Cloud Eye use the same message topic, they can receive messages from the same terminal but with different content.
- You can configure one key event notification for operations initiated by a maximum of 50 users in 10 user groups. For each key event notification, you can add users from different user groups, but cannot select multiple user groups at once.
- After you disable or delete a key event notification, CTS will not send related notifications to subscribers.

Creating a Key Event Notification

1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. In the navigation pane on the left, choose **Key Event Notifications**.
The **Key Event Notifications** page is displayed.
4. Click **Create Key Event Notification**. On the displayed page, specify required parameters.
5. Enter a key event notification name.

Notification Name: Identifies key event notifications. This parameter is mandatory. The name can contain up to 64 characters. Only letters, digits, and underscores (_) are allowed.

6. Configure key operations.

Select the operations that will trigger notifications. When a selected operation is performed, an SMN notification is sent immediately.

- **Operation Type:** Select **All** or **Custom**.
 - **All:** This option is suitable if you have connected CTS to your own audit system. When **All** is chosen, you cannot deselect operations because all operations on all cloud services that have connected with CTS will trigger notifications. You are advised to use an SMN topic for which HTTPS is selected.
 - **Custom:** This option is suitable for enterprises that require detection of high-risk, cost-sensitive, service-sensitive, and unauthorized operations. You can connect CTS to your own audit system for log analysis.

Customize the operations that will trigger notifications. Up to 1,000 operations of 100 services can be added for each notification. For details, see [Supported Services and Operations](#).
- **Advanced Filter:** You can set an advanced filter to specify the operations that will trigger notifications. Operations can be filtered by fields **api_version**, **code**, **trace_rating**, **trace_type**, **resource_id**, and **resource_name**. Up to six filter conditions can be set. When you configure multiple conditions, specify whether an operation is considered a match when all conditions are met (AND) or any of the conditions are met (OR).

Table 5-1 Advanced filtering parameters

Parameter	Description
api_version	Version of the API called in a trace. Enumerated values: • v1 • v3
code	HTTP status code returned by an API.

Parameter	Description
trace_rating	Trace status. Enumerated values: • normal • warning • incident
trace_type	Trace type, including API calls, actions taken on the console, and system-triggered actions. Enumerated values: • ApiCall • ConsoleAction • SystemAction
resource_id	ID of a cloud service resource on which operations are performed. Example: 5a0215bed7a14de38193a*****facef
resource_name	Resource name recorded in a trace.

7. Configure users.

SMN messages will be sent to subscribers when the specified users perform key operations.

- If you select **All users**, SMN will notify subscribers of key operations initiated by all users.
- If you select **Specified users**, SMN will notify subscribers of key operations initiated by your specified users. You can configure up to 50 users across up to 10 user groups. During each selection, you can choose multiple users within a single group, but not multiple groups at once. To add more groups, click **Add** for each one.

8. Configure an SMN topic.

- When **Yes** is selected for **Send Notification**:
 - **SMN Topic**: You can select an existing topic or click **SMN** to create one on the SMN console.
- If you do not want to send notifications, no further action is required.

9. Click **OK**.

Managing Key Event Notifications

After you create a key event notification, you can view its name, status, template, and SMN topic in the notification list and delete it as required.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 3 Choose **Key Event Notifications** in the navigation pane on the left. On the displayed page, perform the following operations as required. For details, see [Table 5-2](#).

Table 5-2 Related operations

Operation	Description
Viewing a key event notification	Click a notification name to view the operation list and user list details of the notification.
Enable/Disable a key event notification	Click Enable or Disable in the Operation column. CTS can trigger key event notifications only after SMN is configured.
Modifying a key event notification	Click Modify in the Operation column.
Deleting a key event notification	Click Delete in the Operation column.
Searching for a notification	In the search box above the list, you can search for notifications by notification name, status, template type, or SMN topic.
Refreshing the key event notification list	Click  in the upper right corner.
Configuring basic settings	Click  in the upper right corner to set table text wrapping, fixed operation column position, and custom columns.

----End

6 Application Examples

6.1 Security Auditing

Scenarios

You can query operation records matching specified conditions and check whether operations have been performed by authorized users for security analysis.

This section describes how to use CTS to audit EVS creation and deletion operations performed in the last two weeks.

Constraints

To store operation records for longer than seven days, you must configure transfer to OBS or LTS for trackers so that you can view them in OBS buckets or LTS log groups.

Prerequisites

You have enabled CTS and trackers are running properly.

Viewing Real-Time Traces

The following takes the records of EVS disk creation and deletion in the last week as an example.

Step 1 Log in to the management console as a CTS administrator.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Trace List** in the navigation pane.

Step 5 Set the time range to **Last 1 week**.

- Step 6** Set filters above the trace list to query the EVS disk creation and deletion operations.
- To query disk creation operations, set **Trace Type** to **Management**, **Trace Source** to **EVS**, **Resource Type** to **evs**, and **Search by** to **Trace name**. Enter **createVolume** in the text box on the right of **Trace name**.
 - To query disk deletion operations, set **Trace Type** to **Management**, **Trace Source** to **EVS**, **Resource Type** to **evs**, and **Search by** to **Trace name**. Enter **deleteVolume** in the text box on the right of **Trace name**.
- Step 7** Click **Query** to check the filtering results.
- Step 8** Check the user information in the results to identify unauthorized operations or operations that do not conform to security rules.
- Step 9** (Optional) To query operation records older than seven days, go to the OBS bucket or LTS log group. For details, see [Querying Transferred Traces](#).
- End

6.2 Fault Locating

Scenarios

If a resource or action encounters a fault, you can quickly locate and analyze it. Just query the traces for that resource at the time of the fault and check the related actions and responses.

This section provides examples demonstrating how to use CTS to locate an ECS fault or an ECS creation failure.

Prerequisites

You have enabled CTS and trackers are running properly.

Viewing Real-Time Traces

The following shows how to locate an ECS fault which occurred in a morning.

- Step 1** Log in to the management console as a CTS administrator.
- Step 2** Click  in the upper left corner to select the desired region and project.
- Step 3** Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
- Step 4** Choose **Trace List** in the navigation pane.
- Step 5** In the time range setting area in the upper right corner, set the time to 06:00 to 12:00 on the certain day.
- Step 6** Set filters above the trace list. Set **Trace Type** to **Management**, **Trace Source** to **ECS**, **Resource Type** to **ecs**, and **Search By** to **Resource ID**. Enter *{ID of the faulty VM}* in the text box on the right of **Resource ID**.

Step 7 Click **Query**.

Step 8 Check the returned traces, especially the request type and response of each trace. Pay attention to traces whose status is **warning** or **incident**, and traces whose response indicates a failure.

----End

The following shows how to locate a fault after an ECS failed to be created.

Step 1 Log in to the management console as a CTS administrator.

Step 2 Click  in the upper left corner to select the desired region and project.

Step 3 Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.

Step 4 Choose **Trace List** in the navigation pane.

Step 5 Set filters as follows: Set **Management** for **Trace Type**, **ECS** for **Trace Source**, **ecs** for **Resource Type**, and **Warning** for **Trace Status**.

Step 6 Click **Query**.

Step 7 In the filtering result, locate the trace named **createServer**. Based on its error code or message, identify and analyze the fault.

----End

6.3 Resource Tracking

Scenarios

You can view operation records of a cloud resource throughout its lifecycle.

This section describes how to use CTS to view all operation records of an ECS.

Constraints

To store operation records for longer than seven days, you must configure transfer to OBS or LTS for trackers so that you can view them in OBS buckets or LTS log groups.

Prerequisites

You have enabled CTS and trackers are running properly.

Viewing Real-Time Traces

Step 1 Log in to the management console as a CTS administrator.

Step 2 Click  in the upper left corner to select the desired region and project.

- Step 3** Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
- Step 4** Choose **Trace List** in the navigation pane.
- Step 5** Set the time range to **Last 1 week**.
- Step 6** Set filters above the trace list. Set **Trace Type** to **Management**, **Trace Source** to **ECS**, **Resource Type** to **ecs**, and **Search By** to **Resource ID**. Enter *{ID of the VM}* in the text box on the right of **Resource ID**.
- Step 7** Click **Query** to query all operation records of the ECS in the last week.
- Step 8** Choose **Tracker List** in the navigation pane. On the displayed page, obtain the OBS bucket or LTS log group information.
- Step 9** To query ECS operation records older than seven days, see [Querying Transferred Traces](#).
- Step 10** In the preceding query results, check all operation and change records of the ECS.
- End

7 Trace References

7.1 Trace Structure

A trace consists of multiple key fields shown in [Table 7-1](#).

NOTE

- This section describes the key trace fields displayed on the CTS console.
- When some fields are displayed on the CTS console, their formats are optimized for easy understanding.

Table 7-1 Key trace fields

Field	Mandatory	Type	Description
time	Yes	Long	Timestamp when a trace was generated. The value is the local standard time, for example, 1660927593570 . This field is transmitted and stored in the form of a timestamp. It is the total number of milliseconds from 00:00:00, January 1, 1970 to the current time.
user	Yes	UserInfo object	Information of the user who performed the operation that triggered the trace.
request	No	String	Request of an operation on resources.
response	No	String	Response to a user request, that is, the returned information for an operation on resources.

Field	Mandatory	Type	Description
service_type	Yes	String	Type of a cloud service whose traces are to be queried.
event_type	Yes	String	Trace type.
project_id	Yes	String	ID of the project to which the trace belongs.
resource_type	Yes	String	Type of the resource on which the operation was performed.
resource_account_id	No	String	ID of the account to which the resource belongs. This parameter has a value only when resources are operated across tenants. For example, if tenant A operates resources of tenant B, the value is the account ID of account B. Note: In the cross-tenant scenario, CTS copies an audit log so that both tenants can view the trace on the CTS console.
read_only	No	boolean	Whether a user request is read-only.
tracker_name	No	String	Name of the tracker that records the trace. - When trace_type is set to system, the default value system is used. - When trace_type is set to data, the value is the name of the corresponding data tracker.
operation_id	Yes	String	Operation ID of the trace.
resource_name	No	String	Name of a resource on which the recorded operation was performed.
resource_id	No	String	ID of a cloud resource on which the recorded operation was performed.

Field	Mandatory	Type	Description
source_ip	Yes	String	IP address of the tenant who performed the operation that triggered the trace. The value of this parameter is empty if the operation is triggered by the system.
domain_id	Yes	String	ID of the account that triggers the trace.
trace_name	Yes	String	Trace name.
trace_rating	Yes	String	Trace status. The value can be normal , warning , or incident . • normal: The operation succeeded. • warning: The operation failed. • incident: The operation caused a serious consequence, for example, a node failure or service interruption.
trace_type	Yes	String	Trace source. For management traces, the value can be ApiCall , ConsoleAction , or SystemAction .
api_version	No	String	Version of the API called in a trace.
message	No	String	Remarks added by other cloud services to a trace.
record_time	Yes	Number	Timestamp when a trace was recorded by CTS.
trace_id	Yes	String	Trace ID. The value is the UUID generated by the system.
code	No	String	HTTP status code returned by the associated API.
request_id	No	String	Request ID.
location_info	No	String	Additional information required for fault locating after a request error.

Field	Mandatory	Type	Description
endpoint	No	String	Endpoint in the detail page URL of the cloud resource on which a recorded operation was performed.
resource_url	No	String	Detail page URL (excluding the endpoint) of the cloud resource on which a recorded operation was performed.
enterprise_project_id	Yes	String	ID of the enterprise project to which the resource belongs.
user_agent	No	String	ID of the request client agent.
content_length	No	Number	Length of the request message body.
total_time	No	Number	Request response time.

Table 7-2 UserInfo

Field	Mandatory	Type	Description
type	No	String	Identity type of the operator.
principal_id	No	String	Identity ID of the operator. - For an IAM user, the format is <i><user-id></i>. - For an IAM assumed-agency session identity, the format is <i><agency-id>.<agency-session-name></i>. - For an IAM federated identity, the format is <i><idp_id>.<user-session-name></i>.
principal_urn	No	String	URN of the operator. - For an IAM user, the format is <i>iam::<account-id>:user:<user-name></i>. - For an IAM agency session identity, the format is <i>sts::<account-id>:assumed-agency:<agency-name>/<agency-session-name></i>. - For an IAM federated identity, the format is <i>sts::<account-id>:external-user:<idp_id>/<user-session-name></i>.

Field	Mandatory	Type	Description
account_id	No	String	Account ID. To obtain it, hover over the username in the upper right corner of the console, select My Credentials from the drop-down menu, and locate the ID on the right of Account ID .
access_key_id	No	String	Access key ID.
id	Yes	String	User ID. To obtain it, hover over the username in the upper right corner of the console, select My Credentials from the drop-down menu, and locate the ID on the right of IAM User ID .
name	Yes	String	Username. To obtain it, hover over the username in the upper right corner of the console, select My Credentials from the drop-down menu, and locate the name on the right of IAM Username .
domain	Yes	BaseUser object	Domain information of the user who performed the operation generating the trace.
user_name	No	String	Username. The meaning of user_name is the same as that of name .
principal_is_root_user	No	String	Whether the operator is user root . - If the value is true, the operator is user root. - If the value is false, the operator is an IAM user of an assumed-agency session identity, a federated identity, or a non-root user.
invoked_by	No	Array of strings	Name of the service that sends the request. The value is ["service.console"] for console operations.
session_context	No	SessionContext object	Temporary security credential attribute.
OriginUser	No	String	Information about the original user who initiates the assumed session.

Table 7-3 BaseUser

Field	Mandatory	Type	Description
id	Yes	String	Account ID. To obtain it, hover over the username in the upper right corner of the console, select My Credentials from the drop-down menu, and locate the ID on the right of Account ID .
name	Yes	String	Account name. To obtain it, hover over the username in the upper right corner of the console, select My Credentials from the drop-down menu, and locate the name on the right of Account Name .

Table 7-4 SessionContext

Field	Mandatory	Type	Description
attributes	No	Attributes object	Temporary security credential attribute.

Table 7-5 Attributes

Field	Mandatory	Type	Description
mfa_authenticated	No	String	Whether MFA identity authentication has been passed.
created_at	No	String	Time when the temporary security credential was issued.

7.2 Example Traces

This section provides two example traces and describes their key fields to help you better understand traces. You can read other traces in a similar way as shown below.

For details on the fields in a trace file, see [Trace Structure](#).

- [ECS Server Creation](#)
- [EVS Disk Creation](#)

ECS Server Creation

```
{
  "trace_id": "cbdd4480-2e03-11ef-82de-cf140e2a70fb",
  "trace_name": "createServer",
  "resource_type": "ecs",
  "trace_rating": "normal",
  "api_version": "1.0",
  "source_ip": "124.71.93.243",
  "domain_id": "7e0d78c85***d0b9b7cba",
  "trace_type": "ConsoleAction",
  "service_type": "ECS",
  "event_type": "system",
  "project_id": "07066c6fc90025a02f6dc01e105b286e",
  "read_only": false,
  "tracker_name": "system",
  "operation_id": "ListSubscriptions",
  "resource_account_id": "7e0d78c85***d0b9b7cba",
  "time": 1718777931170,
  "resource_name": "ecs-test",
  "user": {
    "access_key_id": "HSTAZVL6WYS0J5MYE2GA",
    "account_id": "7e0d78c85***d0b9b7cba",
    "user_name": "IAMUserA",
    "domain": {
      "name": "IAMDomainB",
      "id": "7e0d78c85***d0b9b7cba"
    },
    "name": "IAMUserA",
    "principal_is_root_user": "true",
    "id": "f36972ced***d619f1214",
    "principal_urn": "iam::7e0d78c85***d0b9b7cba:user:IAMUserA",
    "type": "User",
    "principal_id": "f36972ced***d619f1214"
  },
  "record_time": 1718777931170,
  "request": "{\n  \"server\": {\n    \"adminPass\": \"*****\",\n    \"extendparam\": {\n      \"chargingMode\": \"0\",\n      \"regionID\": \"cn-north-4\",\n      \"count\": 1,\n      \"metadata\": {\n        \"op_svc_userid\": \"f36972ced***d619f1214\",\n        \"_support_agent_list\": \"hss,ces\",\n        \"availability_zone\": \"cn-north-4c\",\n        \"description\": \"\",\n        \"name\": \"ecs-test\",\n        \"imageRef\": \"7d940784-ac0a-425f-b3fa-8478f1a1df70\",\n        \"root_volume\": {\n          \"volumetype\": \"GPSSD\",\n          \"extendparam\": {\n            \"resourceSpecCode\": \"GPSSD\",\n            \"resourceType\": \"3\",\n            \"size\": 40,\n            \"metadata\": null,\n            \"hw:passthrough\": false,\n            \"cluster_type\": null,\n            \"cluster_id\": null,\n            \"iops\": null,\n            \"throughput\": null,\n            \"data_volumes\": [],\n            \"flavorRef\": \"sn3.small.1\",\n            \"personality\": [],\n            \"vpcid\": \"250ad46d-9c89-44ec-a97d-293da771b06b\",\n            \"security_groups\": [\n              {\n                \"id\": \"3bb87748-e387-42e5-ad7a-4331638f1321\",\n                \"nics\": [\n                  {\n                    \"subnet_id\": \"1a02d148-e7f9-4a3c-ba58-18099dfbf752\",\n                    \"nictype\": \"\",\n                    \"ip_address\": \"\",\n                    \"port_id\": null,\n                    \"binding:profile\": {\n                      \"disable_security_groups\": false,\n                      \"extra_dhcp_opts\": [],\n                      \"ipv6_bandwidth\": null,\n                      \"ipv6_enable\": false,\n                      \"driver_mode\": null,\n                      \"allowed_address_pairs\": null,\n                      \"efi_enable\": false,\n                      \"efi_protocol\": null,\n                      \"publicip\": {\n                        \"id\": null,\n                        \"eip\": {\n                          \"bandwidth\": {\n                            \"name\": \"ecs-test-bandwidth\",\n                            \"size\": 1,\n                            \"id\": null,\n                            \"sharetype\": \"PER\",\n                            \"productid\": \"\",\n                            \"chargemode\": \"traffic\",\n                            \"extendparam\": {\n                              \"chargingMode\": \"postPaid\",\n                              \"iptype\": \"5_bgp\",\n                              \"ipproductid\": \"\",\n                              \"key_name\": \"KeyPair-ebbe\",\n                              \"isAutoRename\": false,\n                              \"server_tags\": [],\n                              \"batch_create_in_multi_az\": false,\n                              \"spod_enable\": false,\n                              \"user_data\": \"\"\n                            }\n                          }\n                        }\n                      }\n                    }\n                  }\n                }\n              }\n            }\n          }\n        }\n      }\n    }\n  },\n  \"message\": \"success\",\n  \"response\": {\n    \"job_id\": \"ff8080828fe9028a01902f2542df1b10\",\n    \"job_type\": \"createSingleServer\",\n    \"begin_time\": \"2024-06-19T06:18:09.502Z\",\n    \"end_time\": \"2024-06-19T06:18:51.169Z\",\n    \"status\": \"SUCCESS\",\n    \"error_code\": null,\n    \"fail_reason\": null,\n    \"entities\": {\n      \"server_id\": \"7285ea5d-f15c-4d9c-9e4e-37d37023f2f4\"\n    },\n    \"resource_id\": \"7285ea5d-f15c-4d9c-9e4e-37d37023f2f4\",\n    \"request_id\": \"null\"\n  }\n}"
```

Note the following fields:

- **time** indicates the timestamp when a trace was generated. In this example, the value is **1718777931170**.
- **user** indicates the user who performed the operation. In this example, the user is **IAMUserA** (**name** field) under the account **IAMDomainB** (**domain** field).

- **request** indicates the request to create an ECS. It contains basic information about the ECS, such as its name (**ecs-test-bandwidth**) and VPC ID (**250ad46d-9c89-44ec-a97d-293da771b06b**).
- **response** indicates the response to the ECS creation request. It contains **status** (**SUCCESS** in this example), **error_code** (**null** in this example), and **fail_reason** (**null** in this example).

EVS Disk Creation

```
{
  "trace_id": "c4ddaa0b-2e05-11ef-bdc6-e1851d8cb7fb",
  "trace_name": "deleteVolume",
  "resource_type": "evs",
  "trace_rating": "normal",
  "api_version": "1.0",
  "source_ip": "124.71.93.243",
  "domain_id": "7e0d78c85***d0b9b7cba",
  "trace_type": "ConsoleAction",
  "service_type": "EVS",
  "event_type": "system",
  "project_id": "07066c6fc90025a02f6dc01e105b286e",
  "read_only": false,
  "resource_id": "bc661a99-3088-4e86-899f-fb4f46c2bb71",
  "tracker_name": "system",
  "resource_account_id": "7e0d78c85***d0b9b7cba",
  "time": 1718778778419,
  "user": {
    "access_key_id": "HSTAA8960GPIROJGW19L",
    "account_id": "7e0d78c85***d0b9b7cba",
    "user_name": "IAMUserA",
    "domain": {
      "name": "IAMDomainB",
      "id": "7e0d78c85***d0b9b7cba"
    },
    "name": "IAMUserA",
    "principal_is_root_user": "true",
    "id": "f36972ced***d619f1214",
    "principal_urn": "iam::7e0d78c85***d0b9b7cba:user:IAMUserA",
    "type": "User",
    "principal_id": "f36972ced***d619f1214"
  },
  "record_time": 1718778778419,
  "request": "",
  "response": "{\"job_id\":\"defe9cf7b5ca4566860edbebb181e17a\",\"job_type\":\"deleteVolume\", \"begin_time\":\"2024-06-19T06:32:53.018Z\", \"end_time\":\"2024-06-19T06:32:58.411Z\", \"status\": \"SUCCESS\", \"error_code\": null, \"fail_reason\": null, \"entities\": {\"volume_type\": \"GPSSD\", \"volume_id\": \"bc661a99-3088-4e86-899f-fb4f46c2bb71\", \"size\": 10, \"name\": \"volume-d64d\"}}, \"resource_name\": \"volume-d64d\", \"request_id\": \"defe9cf7b5ca4566860edbebb181e17a\""}
}
```

Note the following fields:

- **time** indicates the timestamp when a trace was generated. In this example, the value is **1718778778419**.
- **user** indicates the user who performed the operation. In this example, the user is **IAMUserA** (**name** field) under the account **IAMDomainB** (**domain** field).
- **request**: optional. It is null in this example.
- **response** records the returned result of disk deletion.

- **trace_rating** indicates the trace status. It can replace the **response** field to indicate the operation result. In this example, the value is **normal**, indicating that the operation was successful according to [Trace Structure](#).

8 Cross-Tenant Transfer Authorization

Scenarios

To centrally manage management traces, you can configure the management tracker to transfer the traces of multiple accounts to one OBS bucket. This topic describes how to configure cross-tenant transfer.

Authorizing Cross-Tenant Transfer

1. Tenant B logs in to the management console.

NOTE

- Tenant A is the account for which you want to configure cross-tenant transfer, and tenant B is the account where the OBS bucket resides.
- OBS does not support cross-region transfer. Currently, OBS buckets must be located in the same region of different tenants.

2. Click  in the upper left corner to select the desired region and project.
3. Click  in the upper left corner and choose **Storage > Object Storage Service**.
4. In the navigation pane, choose **Buckets**. In the bucket list, click the name of the desired bucket. The **Objects** page is displayed.
5. In the navigation pane, choose **Bucket Policies**.
6. In the upper right corner of the page, select **JSON** and click **Edit**, and grant permissions to tenant A as follows. Set the italic parameters based on site requirements.
 - When tenant A logs in to the console as a common user to configure a CTS tracker:

```
{
  "Statement": [{
    "Sid": "xxxx",
    "Effect": "Allow",
    "Principal": {
      "ID": [
        "domain/Domain ID of tenant A:agency/cts_admin_trust"
      ]
    },
    "Action": [
      "PutObject"
    ]
  }]
```

```
    ],
    "Resource": [
      "Example bucket name/*"
    ]
  }, {
    "Sid": "xxxx1",
    "Effect": "Allow",
    "Principal": {
      "ID": [
        "domain/Domain ID of tenant A:user/*"
      ]
    },
    "Action": [
      "HeadBucket",
      "ListBucket"
    ],
    "Resource": [
      "Example bucket name"
    ]
  }
]
```

Table 8-1 Bucket policy parameters

Parameter	Description
Sid	ID of a statement. The value is a string that describes the statement.
Action	Actions which a statement applies to. This parameter specifies a set of all the operations supported by OBS. Its values are case insensitive. CTS requires only the following actions: HeadBucket and ListBucket .
Effect	Whether the permission in a statement is allowed or denied. The value is Allow or Deny .
Principal	Tenant A is authorized to use the bucket policy. You can obtain the domain ID on the My Credential page. Principal formats: • domain/Tenant A's account ID:agency/cts_admin_trust: indicates that permissions are granted to the cts_admin_trust agency of tenant A, allowing CTS to transfer logs to OBS buckets using the agency. • domain/Account ID:user/*: indicates that permissions are granted to all users of tenant A. • domain/Account ID:identity-provider/provider-name: indicates that permissions are granted to the specified identity provider of tenant A.
Resource	A group of resources on which the statement takes effect. The wildcard (*) is supported, indicating all resources. <i>Example bucket name/*</i> and <i>Example bucket name</i> are required for cross-account transfer.

7. Click **Save**.

8. If the OBS bucket of tenant B is encrypted using a custom key, you need to authorize tenant A in Data Encryption Workshop (DEW).

 NOTE

You are advised to use a custom key when configuring encryption for buckets of different tenants. Otherwise, the default OBS key of tenant A may be used. In this case, tenant B may fail to download transferred files.

9. Tenant A logs in to the management console.
10. Click  in the upper left corner to select the desired region and project.
11. Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
12. Choose **Tracker List** in the navigation pane.
13. Locate a data tracker and click **Configure** in the **Operation** column. On the displayed page, click **Next**.
14. Enable **Transfer to OBS**, set **OBS Bucket Account** to **Other users**, and manually enter the name of the destination OBS bucket authorized by tenant B.
15. Click **OK** to complete the tracker configuration.

9 Verifying Trace File Integrity

9.1 Enabling Verification of Trace File Integrity

Scenarios

During a security investigation, operational records will not be able to serve as effective and authentic evidence if they are deleted or tampered with. You can enable the integrity verification on CTS to ensure the authenticity of trace files. CTS supports integrity verification of trace files for trackers configured with OBS transfer.

Enabling Verification of Trace File Integrity

1. Log in to the management console.
2. Click  in the upper left corner to select the desired region and project.
3. Click  in the upper left corner and choose **Management & Deployment > Cloud Trace Service**. The CTS console is displayed.
4. Choose **Tracker List** in the navigation pane.
 **NOTE**
Click **Enable CTS** if you have not enabled CTS.
5. Click **Configure** in the row of the management tracker **system**. On the displayed **Configure Tracker** page, click **Next**, and enable **Verify Trace File** in the transfer configuration step.

9.2 Digest Files

Overview

A digest file contains the names and hash values of the trace files transferred to an OBS bucket an hour ago as well as the digital signature of the previous digest file. The digital signature of this digest file is stored in metadata attributes of the digest file object. A digest file is stored in the following path:

*OBS bucket name/CloudTraces/Region/Year/Month/Day/Tracker name/Digest/
Cloud service*

Example: **User Define/CloudTraces/Region/2016/5/19/system/Digest/ECS**

Digest File Name Format

The digest files are named as follows:

Trace file prefix_CloudTrace-Digest_Region_Year-Month-DayTHour-Minute-SecondZ.json.gz (Year-Month-DayTHour-Minute-Second indicates the time when the digest file was sent to OBS.)

Example: *File prefix_CloudTrace-Digest_Region_2016-05-30T16-20-56Z.json.gz*

Digest File Structure

Table 9-1 Key fields of a digest file

Field	Mandatory	Type	Description
project_id	Yes	String	Identifies the account to which a trace file covered in the digest file belongs.
digest_start_time	Yes	String	Specifies the start of the UTC time range covered by the digest file.
digest_end_time	Yes	String	Specifies the end of the UTC time range covered by the digest file.
digest_bucket	Yes	String	Specifies the name of the OBS bucket that the digest file has been sent to.
digest_object	Yes	String	Specifies where the digest file is stored in the OBS bucket.
digest_signature_algorithm	Yes	String	Specifies the algorithm used to sign the digest file.
digest_end	Yes	Boolean	Specifies whether the digest file is an ending digest file.
previous_digest_bucket	No	String	Specifies the name of the OBS bucket that the previous digest file was sent to.
previous_digest_object	No	String	Specifies where the previous digest file is stored in the OBS bucket.
previous_digest_hash_value	No	String	Specifies the hexadecimal encoded hash value of the previous digest file.

Field	Mandatory	Type	Description
previous_digest_hash_algorithm	No	String	Specifies the Hash algorithm used to hash the previous digest file.
previous_digest_signature	No	String	Specifies the digital signature of the previous digest file.
previous_digest_end	Yes	Boolean	Specifies whether the previous digest file is an ending digest file.
log_files	No	Array	Specifies the list of trace files covered in the digest file.
bucket	Yes	String	Specifies the name of the OBS bucket that the trace files have been sent to.
object	Yes	String	Specifies where the trace files are stored in the OBS bucket.
log_hash_value	Yes	String	Specifies the hexadecimal encoded hash value of the trace files.
log_hash_algorithm	Yes	String	Specifies the Hash algorithm used to hash the trace files.

Example Digest File

A digest file contains the names and hash values of the trace files transferred to an OBS bucket an hour ago as well as the digital signature of the previous digest file. The digital signature of this digest file is stored in metadata attributes of the digest file object. A digest file is stored in the following path:

The following is an example digest file:

For details about the fields in the example, see [Table 9-1](#).

```
{
  "project_id": "3cfb09080bd944d0b4cdd72ef2685712",
  "digest_start_time": "2017-03-28T01-09-17Z",
  "digest_end_time": "2017-03-28T02-09-17Z",
  "digest_bucket": "bucket",
  "digest_object": "CloudTraces/eu-west-0/2017/3/28/Digest/EVS/mylog_CloudTrace-Digest_eu-west-0/_2017-03-28T02-09-17Z.json.gz",
  "digest_signature_algorithm": "SHA256withRSA",
  "digest_end": false,
  "previous_digest_bucket": "bucket",
  "previous_digest_object": "CloudTraces/eu-west-0/2017/3/28/Digest/EVS/mylog_CloudTrace-Digest_eu-west-0/_2017-03-28T01-09-17Z.json.gz",
  "previous_digest_hash_value": "5e08875de01b894eda5d1399d7b049fe",
  "previous_digest_hash_algorithm": "MD5",
  "previous_digest_signature":
    "7af7cbef4f3c78eab5048030d402810841400cf70eb22f93d4fedd13b13a8208a5dc04ce2f8bd0a4918f933ca3fc
    b17595ae59386a2dc3e3046fa97688a9815a2d036fa10193534c0ebbecff67221e22ac9cf8b781cbae3a81eaccfc
    0a2bd1a99081b1e4fe99b19caa771876ba7cce16d002feb4578cd89bc6f1faaca639ab48f3cb56007bcc5e248968
    f4a17a95b8cd8bc7d8bbd7c63630da878cd4d471fc75c60bac5f730d94fefe8fdd2f2fa8accd62dbe100eab7773e79
    15e91be4474291b9dacea63a8267390bcb4855b5825554ebbb07d4a29ce077c364213c575c461d1e9fafa0c29fde
  }
```

```
1c6de1d5630e015200821b2f3ae91e53cd8591433dd7c0b4c8bc",
"previous_digest_end": false,
"log_files": [{
  "bucket": "bucket",
  "object": "CloudTraces/eu-west-0/2017/3/28/ECS/mylog_CloudTrace_eu-west-0/
2017-03-28T02-09-17Z_0faa86bc40071242.json.gz",
"log_hash_value": "633a8256ae7996e21430c3a0e9897828",
"log_hash_algorithm": "MD5"
}]
}
```

Digest File Signature

The digital signature information of a digest file is in two metadata attributes of the digest file object. Each digest file has the following two metadata items:

- meta-signature

Hexadecimal encoded value of the digest file signature. Example:

```
7af7cbef4f3c78eab5048030d402810841400cf70eb22f93d4fedd13b13a8208a5dc04ce2f8bd0a4918f933c
a3fcb17595ae59386a2dc3e3046fa97688a9815a2d036fa10193534c0ebbecff67221e22ac9cf8b781cbae3
a81eaccfc0a2bd1a99081b1e4fe99b19caa771876ba7cce16d002feb4578cd89bc6f1faaca639ab48f3cb560
07bcc5e248968f4a17a95b8cd8bc7d8bbd7c63630da878cd4d471fc75c60bac5f730d94fefe8fdd2f2fa8accd
62dbe100eab7773e7915e91be4474291b9dacea63a8267390bcb4855b5825554ebb07d4a29ce077c3642
13c575c461d1e9fafa0c29fde1c6de1d5630e015200821b2f3ae91e53cd8591433dd7c0b4c8bc
```

- meta-signature-algorithm

Algorithm used to sign the digest file. Example:

SHA256withRSA

Supplementary Information

- Starting Digest File

A starting digest file is generated after you start verifying trace file integrity. In a starting digest file, the following fields related to the previous digest file will be left empty:

- previous_digest_bucket
- previous_digest_object
- previous_digest_hash_value
- previous_digest_hash_algorithm
- previous_digest_signature

- "Empty" Digest File

CTS will still send a digest file even if no operations have occurred in your account within the one-hour time period recorded by the digest file. The last field **log_files:[]** of the digest file will be left empty. It helps you to confirm that no trace files have been sent within the one-hour time period recorded by the digest file.

- Digest File Chain

A digest file contains the digital signature and Hash value of the previous digest file (if any) so that a chain is formed. You can verify digest files successively within a specified time, starting with the latest one.

- Digest File Bucket

A digest file is sent to the OBS bucket that stores trace files recorded in the file.

- **Digest File Storage Folder**
A digest file is stored in a folder different from that for trace files, making it easy for you to execute fine-grained security policies.

9.3 Verifying Trace File Integrity

Scenarios

CTS uses public signature algorithms and hash functions in accordance with industry standards, so you can create tools on your own to verify integrity of CTS trace files. Trace files should contain fields **time**, **service_type**, **resource_type**, **trace_name**, **trace_rating**, and **trace_type** for integrity verification. Other fields can be added by services from which traces are collected.

After you enable integrity verification of trace files in CTS, digest files will be sent to your OBS buckets, and you can implement your own verification solution. For details about digest files, see [Digest Files](#).

Prerequisites

You should understand how digest files are signed.

RSA digital signatures are used in CTS. For each digest file, CTS will:

1. Create a message for digital signing, a character string composed of specified digest file fields, and obtain an RSA private key.
2. Produce a hash value of the digest message. Use the RSA algorithm to generate a digital signature with the hash value and private key, and encode the digital signature to hexadecimal format.
3. Put the digital signature into the meta-signature attribute of the digest file object.

The message for digital signing contains the following digest file fields:

- The ending timestamp of the UTC time range covered by the digest file, for example, 2017-03-28T02-09-17Z.
- The path where the current digest file is stored in the OBS bucket.
- The hash value (hexadecimal encoded) of the current digest file (compressed).
- The hexadecimal digital signature of the previous digest file.

Verifying Trace File Integrity

Verify a digest file first and then its referenced trace files.

1. Obtain a digest file.
 - a. Obtain the latest digest file within the time range to be verified from the OBS bucket.
 - b. Check whether the location where the digest file is stored in the OBS bucket matches with the location recorded in the file.
 - c. Obtain the digital signature from the meta-signature attribute of the digest file object.

2. Obtain the RSA public key for verifying the digital signature.

The RSA public key of CTS is as follows:

```
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsjQDkl8COPRhOCvm7Zl8sYZ20ojl+ay/  
gwRSk9q0gkY3pP0RrAhSsEzgYdYjaMCqixkmbpt4AH9AROJU4drnoCAZSMqRngxv0bGC9kVd4q95l4zibsw  
AsksjuNQo/XoJl+rRAqCa+1uetgVU4k4Yx8RryYxYx/tlmvMe/O4mGAlaTf+rsqt3VXR1Qlj5lYR/nx41BEgC/  
Kb1elYAfDaaab8WS5INRprj7qdu6oAo4Ug47WqbecvEtG3JRpj5+oqLyW41Fvse3osC0h5DQdxTt4x00/rVZ  
+gH7Kua00y7gC8YOxFVpYbfn/oW61PUDeHG/N9hUjOrlGDDJpD2YbCIQIDAQAB
```

3. Recreate the message for digital signing.

Compute the message for digital signing.

The message is in the following format:

```
signature_string = digest_end_time  
+ digest_object  
+ Hex(hash(digest-file-content))  
+ previous_digest_signature
```

The following is an example message for digital signing.

```
2017-03-28T02-09-17ZCloudTraces/eu-west-0/2017/3/28/Digest/EVS/mylog_CloudTrace-Digest_eu-  
west-0/  
_2017-03-28T02-09-17Z.json.gze280d203da44015e0eda3faa7a2ec9612221cc0dc8b0fe320db4febe6014  
2350641ad19da18cb6d3f5e7faad792c3efe98836c6d6547f5e5c7a48f7088000a057af26cc3bb913cae163  
7bfa9e4231b7d1fd6d98eaba735e509e7c5ea3c6757f732b4468f7418ef18e3312ac696dd786ec5792eacf  
94aee27cd7be76bf23b641c5e9a686cca6414745787254100c2bee31e584a15c2229270f9dee81f9043574
```

4. Verify a digest file.

Pass the computed message obtained in [3](#), digital signature of the digest file, and public key to the RSA signature verification algorithm. If **true** is returned, the digital signature of the digest file matches with the computed message and the digest file is valid.

5. Verify trace files.

You can verify trace files referenced by the digest file after confirming that the digest file is valid.

The digest file records the hash value of each trace file. After a trace file is uploaded to OBS, its hash value will be stored in ETag metadata. If the trace file is modified after CTS sent it to an OBS bucket, the file's hash value will change, and the digital signatures of the digest file will not match.

Do as follows to verify a trace file:

- a. Obtain **bucket** and **object** information about a trace file from the digest file.
- b. Call the OBS client interface to obtain the ETag metadata value in the trace file object header.
- c. Obtain the hash value of the trace file from the **log_hash_value** field in the digest file.
- d. Compare the ETag metadata value with the hash value obtained in the previous step. If they match, the trace file is valid.

6. Verify the previous digest files and trace files.

In each digest file, the following fields provide the location and signature of the previous digest file:

- previous_digest_bucket
- previous_digest_object
- previous_digest_signature

Repeat steps [4](#) and [5](#) to verify the signature of each previous digest file and all trace files that the file references.

For these previous digest files, you do not need to obtain the digital signature from the meta-signature attribute of the digest file object. The **previous_digest_signature** field in each digest file provides the digital signature of the previous digest file. You can keep verifying the previous digest files and their referenced trace files until you reach the starting digest file or the digest file chain is disconnected.

The following code segment is an example for verifying CTS digest and trace files. The code segment uses the following JAR packages, and you are recommended to use these packages:

- esdk-obs-java-2.1.16.jar
- commons-logging-1.2.jar
- httpasyncclient-4.1.2.jar
- httpclient-4.5.3.jar
- httpcore-4.4.4.jar
- httpcore-nio-4.4.4.jar
- java-xmlbuilder-1.1.jar
- jna-4.1.0.jar
- log4j-api-2.8.2.jar
- log4j-core-2.8.2.jar
- commons-codec-1.9.jar
- json-20160810.jar
- commons-io-2.5.jar

Example code segment:

```
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.security.KeyFactory;
import java.security.MessageDigest;
import java.security.PublicKey;
import java.security.Signature;
import java.security.spec.X509EncodedKeySpec;
import java.util.Arrays;
import java.util.zip.GZIPInputStream;

import org.apache.commons.codec.binary.Base64;
import org.apache.commons.codec.binary.Hex;
import org.apache.commons.io.IOUtils;
import org.json.JSONObject;

import com.obs.services.ObsClient;
import com.obs.services.ObsConfiguration;
import com.obs.services.model.ObjectMetadata;
import com.obs.services.model.S3Object;

public class DigestFileValidator {
    public static void main(String[] args) {
        // Name of the bucket where a digest file is located.
        String digestBucket = "bucketname";
        // Path where a digest file is stored. Example: CloudTraces/eu-de/2017/11/15/Digest/ECS/
        tGPYa_CloudTrace-Digest_eu-de_2017-11-15T10-12-10Z.json.gz.
        String digestObject = "digestObject";

        ObsConfiguration obsConfig = new ObsConfiguration();
```

```
obsConfig.setEndPoint("****Provide OBS EndPoint ****");
ObsClient client = new ObsClient(ak, sk, obsConfig);

try {
    // Obtain a digest file object.
    S3Object object = client.getObject(digestBucket, digestObject);

    InputStream is = new BufferedInputStream(object.getObjectContent());
    byte[] digestFileBytes = IOUtils.toByteArray(is);

    // Obtain the hash value of a digest file.
    MessageDigest messageDigest = MessageDigest.getInstance("MD5");
    messageDigest.update(digestFileBytes);
    byte[] digestFileHashBytes = messageDigest.digest();

    StringBuilder outStr = new StringBuilder();
    GZIPInputStream gis = new GZIPInputStream(new ByteArrayInputStream(digestFileBytes));
    BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(gis, "UTF-8"));
    String line;
    while ((line = bufferedReader.readLine()) != null) {
        outStr.append(line);
    }
    bufferedReader.close();
    String digestInfo = outStr.toString();

    // Obtain the meta-signature value from the digest file header in an OBS bucket, which is the
    digital signature of the digest file.
    ObjectMetadata objectMetadata = client.getObjectMetadata(digestBucket, digestObject);
    String digestSignature = objectMetadata.getMetadata().get("meta-signature").toString();
    JSONObject digestFile = new JSONObject(digestInfo);
    // Check whether the digest file has been moved in the OBS bucket.
    if (!digestFile.getString("digest_bucket").equals(digestBucket) || !
    digestFile.getString("digest_object")
    .equals(digestObject)) {
        System.err.println("Digest file has been moved from its original location.");
    } else {
        // Obtain the message for digital signing.
        String signatureString = digestFile.getString("digest_end_time") +
    digestFile.getString("digest_object")
        + Hex.encodeHexString(digestFileHashBytes) +
    digestFile.getString("previous_digest_signature");

        String publicKeyString
        =
    "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsjQDkl8COPRhOCvm7ZI8sYZ20ojl+ay/
    gwRSk9q0gkY3pP0RrAhSsEzgYdYjaMCqixkmbpt4AH9AROU4drnoCAZSMqRxcgv0bGC9kVd4q95l4zibsw
    AsksjuNQo/XoJBl+rRAqCa+1uetgVU4k4Yx8RryYxYx/tlmvMe/O4mGAlaTf+rsqt3VXR1QIj5LYR/nx41BEgC/
    Kb1eLYAfDaaab8W55INRprj7qdu6oAo4Ug47WqbecvEtG3JRpj5+oqLyW41Fvse3osC0h5DQdxTt4x00/rVZ
    +gH7Kua00y7gC8YOxFVpYbfn/oW61PUDeHG/N9hUjOrlgDDJpD2YbCIQIDAQAB";

        // Public key used for decryption.
        byte[] publicKeyBytes = Base64.decodeBase64(publicKeyString);
        // Form the X509EncodedKeySpec object.
        X509EncodedKeySpec x509EncodedKeySpec = new X509EncodedKeySpec(publicKeyBytes);

        // Specify a cryptographic algorithm.
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        // Obtain the public key object.
        PublicKey publicKey = keyFactory.generatePublic(x509EncodedKeySpec);

        Signature signatureInstance = Signature.getInstance("SHA256withRSA");
        signatureInstance.initVerify(publicKey);
        signatureInstance.update(signatureString.getBytes("UTF-8"));

        byte[] signatureHashExpect = Hex.decodeHex(digestSignature.toCharArray());

        // Verify whether the signature is valid.
        if (signatureInstance.verify(signatureHashExpect)) {
```

```
System.out.println("Digest file signature is valid, validating log files...");

for (int i = 0; i < digestFile.getJSONArray("log_files").length(); i++) {
    JSONObject logFileJson = digestFile.getJSONArray("log_files").getJSONObject(i);
    String logBucket = logFileJson.getString("bucket");
    String logObject = logFileJson.getString("object");

    // Obtain the ETag value from the trace file header in the OBS bucket, which is the
    // recorded hash value of the trace file.
    ObjectMetadata objectLogMetadata = client.getObjectMetadata(logBucket,
logObject);
    String logHashValue = objectLogMetadata.getMetadata().get("ETag").toString();
    logHashValue = logHashValue.replace("\"", "");
    byte[] logFileHash = Hex.decodeHex(logHashValue.toCharArray());

    // Obtain the hash value of each trace file from the digest file.
    byte[] expectedHash = logFileJson.getString("log_hash_value").getBytes();
    boolean hashMatch = Arrays.equals(expectedHash, logFileHash);
    if (!hashMatch) {
        System.err.println("Validate log file hash failed.");
    } else {
        System.out.println("Log file hash is valid.");
    }
} else {
    System.err.println("Validate digest signature failed.");
}

System.out.println("Digest file validation completed.");

// Obtain values of fields previous_digest_bucket, previous_digest_object, and
previous_digest_signature of the previous digest file. After obtaining the digest file, verify its hash
value and digital signature.
String previousDigestBucket = digestFile.getString("previous_digest_bucket");
String previousDigestObject = digestFile.getString("previous_digest_object");

// Obtain the digital signature from the meta-signature attribute of the digest file object
header.
ObjectMetadata objectPreviousMetadata = client.getObjectMetadata(previousDigestBucket,
previousDigestObject);
String signatruePrevious = objectPreviousMetadata.getMetadata().get("meta-
signature").toString();
String signatruePreviousExpect = digestFile.getString("previous_digest_signature");
if (signatruePrevious.equals(signatruePreviousExpect)) {
    System.out.println(
        "Previous digest file signature is valid, " + "validating previous digest file hash
value...");

    String digestPreviousHashValue =
objectPreviousMetadata.getMetadata().get("ETag").toString();
    // The ETag metadata value is the trace file hash value enclosed with quotation marks.
    You need to remove the quotation marks.
    String digestPreviousHashValueExpect = "\"" +
digestFile.getString("previous_digest_hash_value")
        + "\"";
    if (digestPreviousHashValue.equals(digestPreviousHashValueExpect)) {
        System.out.println("Previous digest file hash value is valid.");
    } else {
        System.err.println("Validate previous digest file hash value failed.");
    }
}
} catch (Exception e) {
    System.out.println("Validate digest file failed.");
}
}
```

10 Auditing

Cloud Trace Service (CTS) provides records of operations performed on cloud service resources.

With CTS, you can record operations associated with CTS itself for later query, audit, and backtracking.

Table 10-1 CTS operations (creation, modification, and deletion) that can be recorded by itself

Operation	Resource Type	Trace Name
Creating a tracker	tracker	createTracker
Modifying a tracker	tracker	updateTracker
Disabling a tracker	tracker	updateTracker
Enabling a tracker	tracker	updateTracker
Deleting a tracker	tracker	deleteTracker
Creating a key event notification	notification	createNotification
Deleting a key event notification	notification	deleteNotification
Modifying a key event notification	notification	updateNotification
Changing the status of a key event notification	notification	updateNotificationStatus
Disabling a key event notification	notification	updateNotification
Enabling a key event notification	notification	updateNotification
Modifying settings of the configuration center	configuration	updateConfiguration

Table 10-2 CTS operations (query) that can be recorded by itself

Operation	Resource Type	Trace Name
Querying the trace list	trace	listTraces
Exporting the trace list	trace	exportTrace
Querying one or more key event notifications	notification	listNotification
Querying one or more trackers	tracker	listTrackers
Querying the configuration center	configuration	showConfiguration

11

Supported Services and Operations

Table 11-1 Supported services and operations

Category	Cloud Service	Operations
Compute	Elastic Cloud Server (ECS)	<i>ECS User Guide</i> > Operations that can be recorded by CTS
	Image Management Service (IMS)	<i>IMS User Guide</i> > Operations that can be recorded by CTS
	Auto Scaling (AS)	<i>AS User Guide</i> > Operations that can be recorded by CTS
Storage	Object Storage Service (OBS)	<i>OBS User Guide</i> > Operations that can be recorded by CTS
	Elastic Volume Service (EVS)	<i>EVS User Guide</i> > Operations that can be recorded by CTS
	Cloud Backup and Recovery (CBR)	<i>CBR User Guide</i> > Operations that can be recorded by CTS
Network	Virtual Private Cloud (VPC)	<i>VPC User Guide</i> > Operations that can be recorded by CTS
	Elastic Load Balance (ELB)	<i>ELB User Guide</i> > Operations that can be recorded by CTS
	NAT Gateway (NAT)	<i>NAT User Guide</i> > Operations that can be recorded by CTS
	VPC Endpoint (VPCEP)	<i>VPCEP User Guide</i> > Operations that can be recorded by CTS
	Elastic IP (EIP)	<i>EIP User Guide</i> > Operations that can be recorded by CTS
Management & Governance	Cloud Eye Service (CES)	<i>CES User Guide</i> > Operations that can be recorded by CTS

Category	Cloud Service	Operations
	Cloud Trace Service (CTS)	<i>CTS User Guide</i> > Operations that can be recorded by CTS
	Identity and Access Management (IAM)	<i>IAM User Guide</i> > Operations that can be recorded by CTS
	Log Tank Service (LTS)	<i>LTS User Guide</i> > Operations that can be recorded by CTS
Application Services	Application Operations Management (AOM)	<i>AOM User Guide</i> > Operations that can be recorded by CTS
	Simple Message Notification (SMN)	<i>SMN User Guide</i> > Operations that can be recorded by CTS
Database	Relational Database Service for PostgreSQL (RDS for PostgreSQL)	<i>RDS for PostgreSQL User Guide</i> > Operations that can be recorded by CTS
	Relational Database Service for MySQL (RDS for MySQL)	<i>RDS for MySQL User Guide</i> > Operations that can be recorded by CTS
	Data Replication Service (DRS)	<i>DRS User Guide</i> > Operations that can be recorded by CTS
Security	Web Application Firewall (WAF)	<i>WAF User Guide</i> > Operations that can be recorded by CTS
	Host Security Service (HSS)	<i>HSS User Guide</i> > Operations that can be recorded by CTS
	Anti-DDoS (Anti-DDoS)	<i>Anti-DDoS User Guide</i> > Operations that can be recorded by CTS

12 Permissions Management

You can use Identity and Access Management (IAM) for fine-grained permissions control for your CTS. With IAM, you can:

- Create IAM users for personnel based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing CTS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust other accounts or cloud services to perform efficient O&M on your CTS resources.

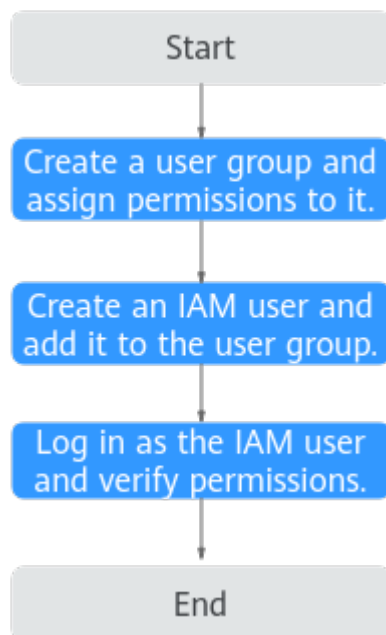
If your account does not require individual IAM users, you can skip this section.

Prerequisites

Before granting permissions to user groups, learn about system-defined permissions in [Permissions](#)).

Process Flow

Figure 12-1 Process of granting CTS permissions



1. On the IAM console, create a user group and grant it permissions.
Create a user group on the IAM console, and attach the **CTS Administrator** policy to the group.
2. Create an IAM user and add it to the created user group.
Create a user on the IAM console and add it to the user group created in [1](#).
3. Log in as the IAM user and verify permissions.
Log in to the console as the user you created, and verify that the user has the assigned permissions.