

# **Cloud Log Service**

## **Product Introduction**

### **Product Documentation**



## Copyright Notice

©2013-2022 Tencent Cloud. All rights reserved.

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

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

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

# Contents

## Product Introduction

- Overview

- Strengths

  - Features

  - Cost Effectiveness

- Available Regions

- Concepts

  - Log and Log Group

  - Log Topic and Logset

  - Machine Group

  - Segment and Index

  - Topic Partition

- Limits

  - LogListener Limits

# Product Introduction

## Overview

Last updated : 2022-01-04 15:38:18

Cloud Log Service (CLS) is a centralized logging solution. You can stop worrying about scaling or other resource issues and access CLS within just five minutes. CLS offers solutions for collecting, storing, searching, and analyzing logs, helping you identify business issues, monitor metrics, ensure security, and simplify operations.

## Features

CLS has the following features.

- Log collection: collect logs to CLS from different log sources by using LogListener, API, etc.
- Log storage: store log data with CLS.
- Log index: enabling log index for log query can help you quickly identify log problems.
- Log shipping: you can ship specified logs to other cloud products to meet storage or other computing needs. For example, you can ship a log to a specified COS bucket to manage its lifecycle and meet the need for log auditing.

### Log collection

CLS supports collecting logs by using LogListener and API, enabling you to easily collect log data from different regions, channels, platforms, and data sources in real time and collect logs from various Tencent Cloud products.

- LogListener real-time collection: use LogListener to collect logs. This method is easy to install, reliable, and secure. It supports most mainstream Linux operating systems, delivering high performance while occupying few resources.
- Collection via API: upload logs by calling an API, without the need to install LogListener. Multiple languages are supported.

### Log index and query

- Real-time index: index collected log data in real time to enable data search.
- Excellent query performance: query results are returned within seconds. 100-million-level log data search and quick data locating are supported.
- Flexible query: features such as full-text search, multi-keyword search, range query, fuzzy query, and Chinese search are supported.

### Log shipping

- Shipping logs to COS: ship log data to COS buckets under your account to store and manage the log data.

# Strengths

## Features

Last updated : 2022-07-13 11:05:08

### Rich Features

- LogListener provides one-stop log service, including log collection, storage, search, transfer, and shipping features.
- LogListener parses log structures in many ways, including full text in a single line, full text in multi lines, separator, JSON, and regular expressions.
- LogListener allows you to select from multiple data access methods based on your business needs. For more information, see [Collection Overview](#).
- LogListener provides multiple search syntaxes and allows you to search logs by keyword, fuzzy match, or range.

### Stable and Reliable

- CLS has a highly scalable distributed storage architecture. It supports horizontal scaling and automatic service scaling, and easily stores and manages massive log data.
- The CLS backend storage manages and stores logs in multiple copies, ensuring data reliability.

### Simple and Efficient

- LogListener supports simple, GUI-based configuration. You can quickly access CLS through LogListener.
- Data can be consumed immediately after being written to CLS. Query results of hundreds of millions of data records can be returned in seconds.
- The service is billed by actual usage. You do not need to build or maintain a log system, preventing resource idling and waste.

### Ecological Expansion

- The logs of certain cloud services have already been integrated into CLS. For more information, see [Accessing log sources](#).
- Logs are shipped to Cloud Object Storage (COS), which supports long-term archive storage of logs.

- Logs are shipped to CKafka, which supports real-time log consumption and facilitates further log processing and analysis.

# Cost Effectiveness

Last updated : 2022-08-24 12:32:10

Tencent Cloud Log Service (CLS) allows you to **save about 84%** on your budget compared with the self-built logging system (for example, ELK) of Elasticsearch Service. A detailed comparison is given herein.

## Cost Comparison

The following table compares the costs with an assumption that 100 million pieces of 100-byte logs are generated per day, all logs need to be stored for 15 days, and the peak QPS is 4,000:

Product	List Price	Description
CLS	38 USD/month	<ul style="list-style-type: none"> <li>Write traffic: 0.074 USD/day</li> <li>Index traffic: 0.694 USD/day</li> <li>Log storage: 0.084 USD/day</li> <li>Index storage: 0.403 USD/day</li> <li>Service request: 0.003 USD/day</li> <li>Topic partition tenant: 0.014 USD/day</li> </ul> For more information, please see the <b>Billing Example</b> section in <a href="#">Billing Overview</a> .
Self-built logging system of the Elasticsearch Service	244 USD/month	<ul style="list-style-type: none"> <li>Node model: 125 USD/month</li> <li>Node storage: 119 USD/month</li> </ul>

## Billing Description

Elasticsearch involves the following fees:

### Node model fees

125 USD/month (assume that you purchased the ES.S1.MEDIUM8 model with a 2-core CPU and 4 GB memory for the three nodes)

### Node storage fees

- According to the [Elasticsearch Service](#) documentation, using 1 replica improves data reliability, and reserving 50% of the storage capacity maintains service stability. In this case, the actual storage will be as follows:

- Actual capacity:

$$\text{Storage capacity} = \text{Source data} \times (1 + \text{Number of replicas}) \times 1.45 \times (1 + \text{reserved capacity}) \approx \text{Source data} \times (1 +$$

Number of replicas)  $\times$  2.2 = 614.46 GB

If you use a 210 GB SSD for each node, the three nodes will cost 119 USD per month.

Note :

For more information about cluster configurations, please see [Evaluation of Cluster Specification and Capacity Configuration](#).



# Available Regions

Last updated : 2022-10-17 15:31:56

## Overview

You can create logsets and log topics in different regions when using CLS. Regions are independent geographical areas where IDCs are located. Tencent Cloud regions are completely isolated. You can select the nearest region based on different business scenarios and the location of your targeted users to reduce log access latency and improve user experience.

## Available regions

Region	Code
Beijing	ap-beijing
Guangzhou	ap-guangzhou
Shanghai	ap-shanghai
Chengdu	ap-chengdu
Nanjing	ap-nanjing
Chongqing	ap-chongqing
Hong Kong (China)	ap-hongkong
Silicon Valley	na-siliconvalley
Virginia	na-ashburn
Singapore	ap-singapore
Bangkok	ap-bangkok
Mumbai	ap-mumbai
Frankfurt	eu-frankfurt
Tokyo	ap-tokyo

Region	Code
Seoul	ap-seoul
Moscow	eu-moscow
Jakarta	ap-jakarta
Toronto	na-toronto
São Paulo	sa-saopaulo

Note :

- If CLS is integrated into other cloud products, you need to select a logset in the same region as the other cloud products. Cloud products in the same region access each other over a private network, which effectively reduces latency and improves access speed.

## Domain name

CLS has different domain names for each of its modules, as described below:

- LogListener
- CLS API 3.0
- API for log upload
- Uploading Logs via Kafka
- Kafka Consumption Logs

LogListener is a log collection client provided by CLS and can report local logs to CLS. It uses the following domain names:

Region	Code	Private Network Domain Name	Public Network Domain Name
Beijing	ap-beijing	ap-beijing.cls.tencentyun.com	ap-beijing.cls.tencentcs.com
Guangzhou	ap-guangzhou	ap-guangzhou.cls.tencentyun.com	ap-guangzhou.cls.tencentcs.com
Shanghai	ap-shanghai	ap-shanghai.cls.tencentyun.com	ap-shanghai.cls.tencentcs.com
Chengdu	ap-chengdu	ap-chengdu.cls.tencentyun.com	ap-chengdu.cls.tencentcs.com

Region	Code	Private Network Domain Name	Public Network Domain Name
Nanjing	ap-nanjing	ap-nanjing.cls.tencentyun.com	ap-nanjing.cls.tencentcs.com
Chongqing	ap-chongqing	ap-chongqing.cls.tencentyun.com	ap-chongqing.cls.tencentcs.com
Hong Kong (China)	ap-hongkong	ap-hongkong.cls.tencentyun.com	ap-hongkong.cls.tencentcs.com
Silicon Valley	na-siliconvalley	na-siliconvalley.cls.tencentyun.com	na-siliconvalley.cls.tencentcs.com
Virginia	na-ashburn	na-ashburn.cls.tencentyun.com	na-ashburn.cls.tencentcs.com
Singapore	ap-singapore	ap-singapore.cls.tencentyun.com	ap-singapore.cls.tencentcs.com
Bangkok	ap-bangkok	ap-bangkok.cls.tencentyun.com	ap-bangkok.cls.tencentcs.com
Mumbai	ap-mumbai	ap-mumbai.cls.tencentyun.com	ap-mumbai.cls.tencentcs.com
Frankfurt	eu-frankfurt	eu-frankfurt.cls.tencentyun.com	eu-frankfurt.cls.tencentcs.com
Tokyo	ap-tokyo	ap-tokyo.cls.tencentyun.com	ap-tokyo.cls.tencentcs.com
Seoul	ap-seoul	ap-seoul.cls.tencentyun.com	ap-seoul.cls.tencentcs.com
Moscow	eu-moscow	eu-moscow.cls.tencentyun.com	eu-moscow.cls.tencentcs.com
Jakarta	ap-jakarta	ap-jakarta.cls.tencentyun.com	ap-jakarta.cls.tencentcs.com
Toronto	na-toronto	na-toronto.cls.tencentyun.com	na-toronto.cls.tencentcs.com
São Paulo	sa-saopaulo	sa-saopaulo.cls.tencentyun.com	sa-saopaulo.cls.tencentcs.com

# Concepts

## Log and Log Group

Last updated : 2022-03-08 12:48:20

### Log

Logs are record data generated during the running of an application system, such as user operation logs, API access logs, and system error logs. Logs are usually stored in text format on the host where the application system resides. A log corresponding to a system running record may contain one line of text (single-line log) or multiple lines of text (multi-line log).

Logs can be uploaded the CLS via [LogListener](#) or other methods such as APIs or SDKs.

#### Example of a single-line log:

```
59.x.x.x - - [06/Aug/2019:12:12:19 +0800] "GET /nginx-logo.png HTTP/1.1" 200 368
"http://119.x.x.x/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/75.0.3770.142 Safari/537.36" "-"
```

#### Example of a multi-line log:

```
java.net.SocketTimeoutException:Receive timed out
at j.n.PlainDatagramSocketImpl.receive0(Native Method) [na:1.8.0_151]
at j.n.AbstractPlainDatagramSocketImpl.receive(AbstractPlainDatagramSocketImpl.java:143) [^]
at j.n.DatagramSocket.receive(DatagramSocket.java:812) [^]
at o.s.n.SntpClient.requestTime(SntpClient.java:213) [classes/]
at o.s.n.SntpClient$1.call(^:145) [^]
at ^.call(^:134) [^]
at o.s.f.SyncRetryExecutor.call(SyncRetryExecutor.java:124) [^]
at o.s.f.RetryPolicy.call(RetryPolicy.java:105) [^]
at o.s.f.SyncRetryExecutor.call(SyncRetryExecutor.java:59) [^]
at o.s.n.SntpClient.requestTimeHA(SntpClient.java:134) [^]
at ^.requestTimeHA(^:122) [^]
at o.s.n.SntpClientTest.test2h(SntpClientTest.java:89) [test-classes/]
at s.r.NativeMethodAccessorImpl.invoke0(Native Method) [na:1.8.0_151]
```

The main components of a log are as follows:

Component	Description	Example
__TIMESTAMP__	Log time, in the format of a UNIX timestamp in milliseconds	1640005601188

Component	Description	Example
__FILENAME__	Log source file	/data/log/nginx/access.log
__SOURCE__	Log source IP	10.0.1.2
Log body	<p>Log body content, in <code>key:value</code> format ( <code>key</code> is the field name, and <code>value</code> is the field value.)</p> <p>If the log extraction mode is full text in a single line or full text in multi lines (the entire raw log is reported without splitting), the entire log is stored in the <code>__CONTENT__</code> field.</p> <p>If the log extraction mode is any other mode (such as splitting with a separator), each part of the raw log corresponds to a <code>key:value</code> pair.</p>	<p>Full text in a single line or full text in multi lines:</p> <pre>10.20.20.10:[2018-07-16 13:12:57];GET /online/sample HTTP/1.1;200</pre> <p>Other modes:</p> <pre>IP: 10.20.20.10 request: GET /online/sample HTTP/1.1 status: 200 time: [2018-07-16 13:12:57]</pre>
Metadata	<p>Simple description or categorization of logs. For example, the cluster or container of a TKE log is stored in the <code>key:value</code> format, where <code>key</code> starts with <code>__TAG__</code>.</p>	<code>__TAG__.clusterId:1skzv59c</code>

## Log group

A log group (LogGroup) is a collection of multiple logs. In the process of log upload, to increase data read/write efficiency, multiple logs can be packaged into one log group and then sent to CLS.

Logs in the same log group have the same basic information (`__TIMESTAMP__`, `__FILENAME__`, `__SOURCE__`, metadata, etc.).

# Log Topic and Logset

Last updated : 2022-03-08 22:05:22

## Log topic

A log topic is a basic unit for log data collection, storage, search and analysis on the CLS platform. The massive amounts of logs collected are managed by log topic. For example, you can configure log collection rules and storage time, search for and analyze logs, and download, consume, and ship logs by log topic.

A log topic usually corresponds to an application/service. It is recommended to collect logs of the same application/service from different servers to the same log topic. For example, a payment service (payService) is deployed on dozens of machines and has two types of logs: access logs (access\_log) and error logs (error\_log). You can create two log topics, payService\_access\_log\_topic and payService\_error\_log\_topic, to collect the two types of logs on the machines respectively. You can use these two log topics to centrally search for and analyze all logs on the machines.

Log topics and applications/services are not strictly in one-to-one mapping relationships. If the log structures of two services are similar, and the logs of the two services need to be analyzed in a centralized manner, you can report the logs of the two services to the same log topic.

## Logset

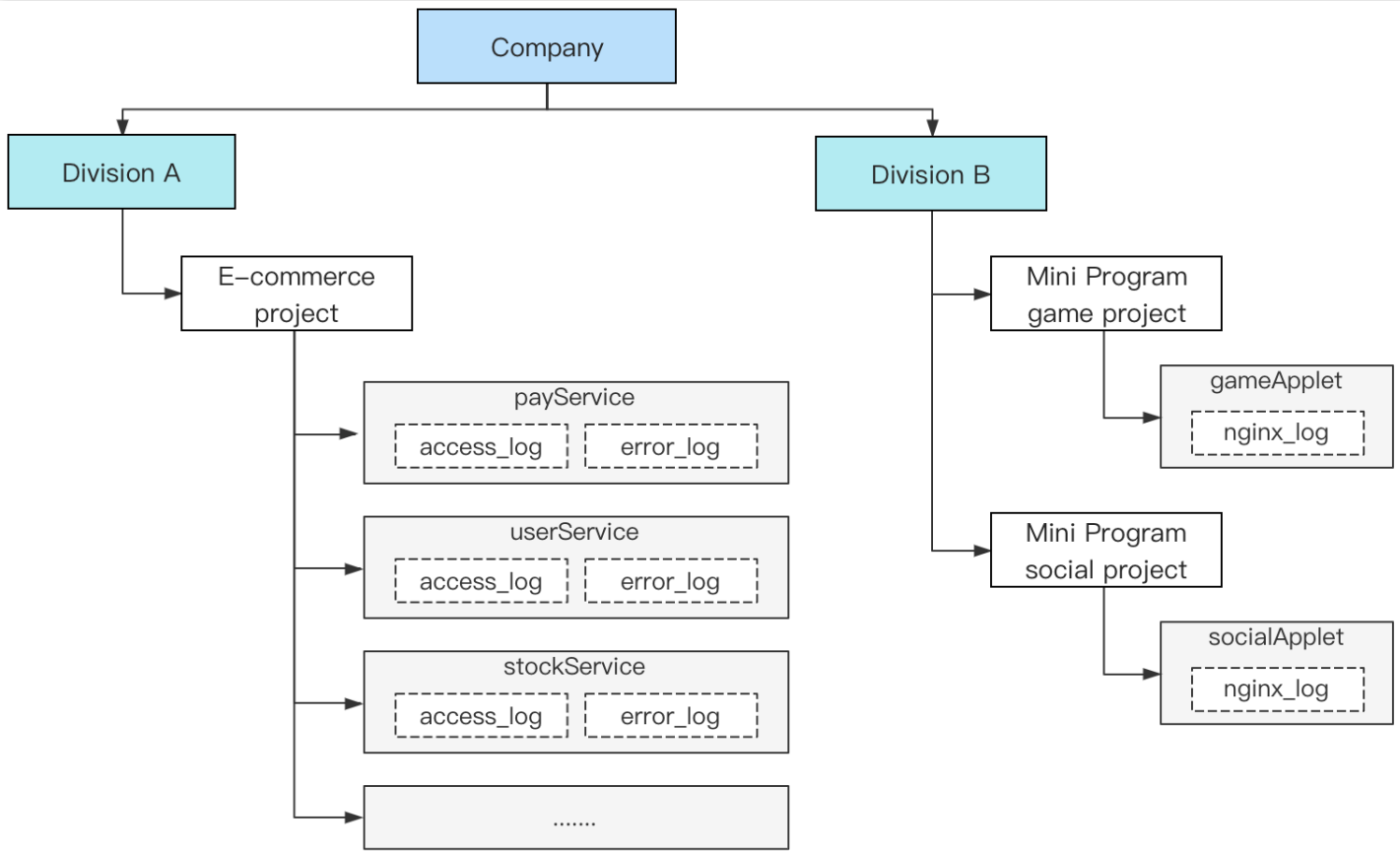
A logset is a class of log topics and can contain multiple log topics. A logset itself does not store any log data, but just makes it easier for users to manage log topics.

A logset usually corresponds to a project/business in a company. You are advised to add the log topics of multiple applications/businesses of a project/business to the same logset. For example, if a company's e-commerce project contains multiple services (payService, userService, stockService, etc.), you can create a logset named e\_commerce\_logset and add the log topics of these services to this logset. In this way, when the company has multiple projects, the project personnel only need to view the log topics in the logset corresponding to the project, without interference from the log topics of other projects.

Note :

The logset to which a log topic belongs can be specified when the log topic is created and cannot be changed once specified.

## Example



As shown in the figure above, the company has two departments:

- Department A has an e-commerce project that adopts the microservice architecture, and each service has two types of logs: access logs (access\_log) and error logs (error\_log).
- Department B has two projects: Mini Program game project and Mini Program social project. The two projects adopt a simple technical architecture and each has one type of logs: nginx access logs (nginx\_log).

To use CLS to monitor the logs of the applications of the company, you can create the following logsets and log topics:

Logset	Log Topic	Label
e_commerce_logset	payService_access_log_topic	Department: Department A
e_commerce_logset	payService_error_log_topic	Department: Department A
e_commerce_logset	userService_access_log_topic	Department: Department A
e_commerce_logset	userService_error_log_topic	Department: Department A
e_commerce_logset	stockService_access_log_topic	Department: Department A
e_commerce_logset	stockService_error_log_topic	Department: Department A

Logset	Log Topic	Label
e_commerce_logset	.....	Department: Department A
gameApplet_logset	gameApplet_nginx_log_topic	Department: Department B
socialApplet_logset	socialApplet_nginx_log_topic	Department: Department B

The labels are used to identify the departments to which the logsets and log topics belong. Combined with permission policies, the personnel in each department can view the data of their own department only.

For department A, the e\_commerce\_logset logset covers the log topics of all e-commerce services. If other projects are added later, you only need to create a logset.

For department B, although the current technical architecture is relatively simple, with only two types of logs in total, two logsets are created, each with only one log topic, for the following purposes:

- Support for subsequent architecture expansion: If the service scale increases and the technical architecture changes to the microservice architecture, you can continue to use the current logset and add log topics to the current logset. In this way, projects do not affect each other.
- Flexible response to project adjustment: If the entire department is used as a logset, and a project needs to be changed to an independent department or moved to another department, log adjustment will be difficult because the logset of a log topic cannot be changed directly, and you may need to collect logs again. If each project corresponds to a logset, this situation does not exist, and you only need to adjust the labels corresponding to the logset and log topic for department adjustment.



# Machine Group

Last updated : 2022-11-21 16:51:27

## Machine group

A machine group is a list of machines for which logs need to be collected. In machine groups, CLS manages all servers for which logs are collected using [LogListener](#).

A machine group can contain multiple machines. When an application/service is deployed on multiple machines with the same log file path, the machines can be grouped into a machine group. In this way, you only need to configure the log data collection rule once in the console, and the rule takes effect on all the machines in the machine group in batches.

A machine group can be associated with a log topic. That is, all the logs in the machine group are reported to the same log topic. You can also associate a machine group with multiple log topics. That is, logs of different paths in a machine group are reported to different log topics.

Machine groups can be defined in two ways:

- IP address: add an IP address list to a machine group. Then the machines corresponding to the IPs will be automatically added to the machine group.
- Label: add labels to machines when installing [LogListener](#). Then the machines with the labels will be automatically added to the machine group.

## Examples

An e-commerce project has 6 machines and 3 services (payService, userService, and stockService). The deployment method is as follows:

payService is deployed on 2 machines, and there are 2 log file paths. userService and stockService are deployed on 4 machines, and there are 4 log file paths. Each of the log types (access\_log and error\_log) of each of the services needs to be reported to an independent log topic.

Machine	Service Deployed	Log File Path
192.168.1.1	payService	/data/log/payService/access_log.log /data/log/payService/error_log.log
192.168.1.2	payService	/data/log/payService/access_log.log /data/log/payService/error_log.log

Machine	Service Deployed	Log File Path
192.168.1.3	userService,stockService	/data/log/userService/access_log.log /data/log/userService/error_log.log /data/log/stockService/access_log.log /data/log/stockService/error_log.log
192.168.1.4	userService,stockService	/data/log/userService/access_log.log /data/log/userService/error_log.log /data/log/stockService/access_log.log /data/log/stockService/error_log.log
192.168.1.5	userService,stockService	/data/log/userService/access_log.log /data/log/userService/error_log.log /data/log/stockService/access_log.log /data/log/stockService/error_log.log
192.168.1.6	userService,stockService	/data/log/userService/access_log.log /data/log/userService/error_log.log /data/log/stockService/access_log.log /data/log/stockService/error_log.log

When deploying LogListener on servers, you can add a label for each machine according to the services running on the machine. See the table below.

Machine	Service Deployed	Label
192.168.1.1	payService	payService
192.168.1.2	payService	payService
192.168.1.3	userService,stockService	userService,stockService
192.168.1.4	userService,stockService	userService,stockService
192.168.1.5	userService,stockService	userService,stockService
192.168.1.6	userService,stockService	userService,stockService

Then create 3 machine groups in the console and define them with labels payService, userService, and stockService. Then, associate the machine groups with log topics and add the corresponding collection configuration. By now, log reporting configuration is completed.

Log Topic	Associated Machine Group	Collection Path
-----------	--------------------------	-----------------

Log Topic	Associated Machine Group	Collection Path
payService_access_log_topic	payService	/data/log/payService/access_log.log
payService_error_log_topic	payService	/data/log/payService/error_log.log
userService_access_log_topic	userService	/data/log/userService/access_log.log
userService_error_log_topic	userService	/data/log/userService/error_log.log
stockService_access_log_topic	stockService	/data/log/stockService/access_log.log
stockService_error_log_topic	stockService	/data/log/stockService/error_log.log

If a service needs to be expanded in the future, you only need to add the service label to the machines to be added so that the new machines can be automatically added to the corresponding machine group for log collection, significantly improving Ops and deployment efficiency.

# Segment and Index

Last updated : 2022-03-08 12:48:20

## Segmentation

### Segmentation explanation

To search for a long log, you usually only need to search for part of the log content. Assume that you need to search for the following complete log that contains `sample` :

```
10.20.20.10; [2018-07-16 13:12:57]; GET /online/sample HTTP/1.1; 200
```

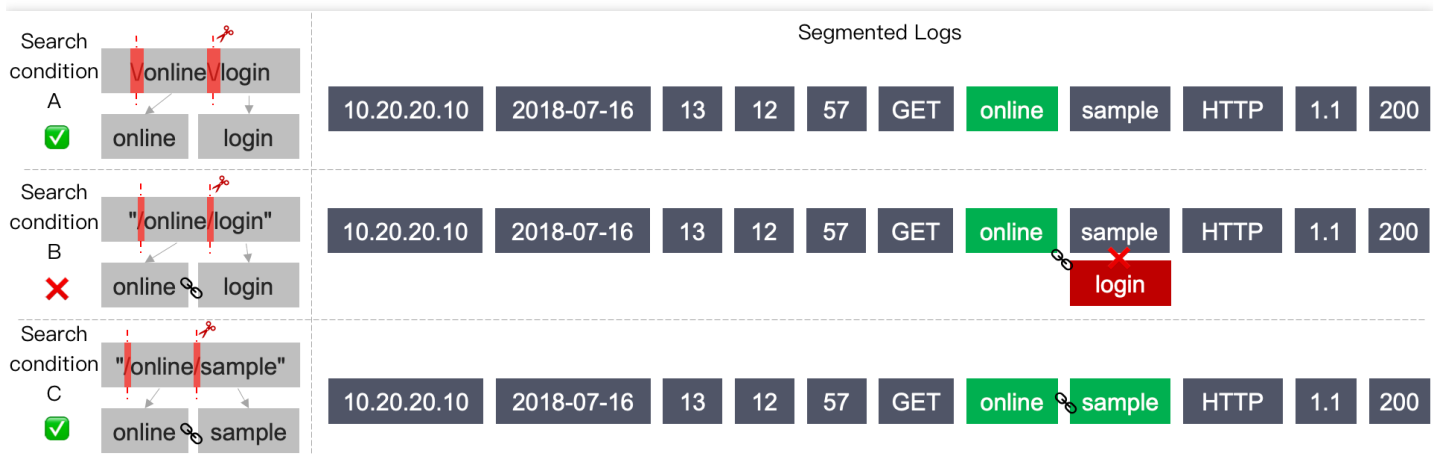
The full text of the log is long, and the log contains many other content in addition to `sample` . The complete log and the search condition are not identical. Therefore, `sample` cannot be used as the search condition to search for the log. To meet this search requirement, the full text of the log needs to be divided into multiple fragments, and each fragment is called a word, and this process is called "segmentation".

For example, the above sample log can be divided based on symbols `@&()=',';:&lt;>[]{} / \n\t\r` , that is, wherever the symbols occur, segmentation is executed. The segmentation result is as follows:



If the search condition is `sample` , the preceding log after segmentation contains `sample` and therefore is considered to meet the search condition.

The search condition itself is also segmented, such as the 3 types of search conditions in the figure below:



- Search condition A: `\\/online\\/login`
  - `\\` is used to escape the `/` symbol (this symbol is a reserved symbol of the search syntax and therefore needs to be escaped).
  - The escaped `/` symbol is a delimiter, so the actual search condition is `online OR login`. A log containing `online` **or** `login` is considered to meet the search condition.
  - The example log provided above **meets** the search condition.
- Search condition B: `"/online/login"`
  - Being enclosed by double quotation marks, the `/` symbol does not need to be escaped.
  - The content in the double quotation marks is also divided into two words. However, the double quotation marks indicate that only a log that **contains both the two words in the exact order** as that in the search condition is considered to meet the search condition.
  - The example log provided above does not contain `login` and therefore **does not meet** the search condition.
- Search condition C: `"/online/sample"`
  - The example log provided above contains both `online` and `sample` in the exact order as that in the search condition and therefore is considered to **meet** the search condition.

In most scenarios, if the search condition contains delimiters, you are advised to use double quotation marks to avoid adding escape characters while accurately searching for desired logs. For more information about the search syntax, see [Syntax and Rules](#).

## Segmentation configuration

The basis of log segmentation falls into two categories. For configuration details, please see [Configuring Indexes](#).

- Delimiter: you can customize the symbols for log segmentation, including English symbols and `\n\t\r`. For example, in the example above, `@&()='",;:&lt;t;>[]{} / \n\t\r` are delimiters.

- Allow Chinese Characters: Chinese is special, and Chinese symbols cannot be used as delimiters, and segmentation according only to symbols often fails to achieve the desired effect. For example, if a log in Chinese is

用户登录失败, 密码错误

and you want the log to be searched for by

"登录失败"

, segmentation according to symbols cannot be used to meet the search requirement. In this case, you can set the log to **Allow Chinese Characters** by referring to [Configuring Indexes](#) so that CLS will automatically segment every Chinese character into a separate word.

## Index

To enable quick log searches, CLS performs a lot of preprocessing, including segmentation, on the logs uploaded to the platform, and this process is called "indexing". Indexes determine the conditions under which logs can be searched and analyzed. Therefore, before uploading log data, you need to set a reasonable index rule for the corresponding log topic to facilitate subsequent search and analysis. For more information, please see [Configuring Indexes](#).

Index rules take effect only for newly uploaded log data, not historical data. Index creation involves index traffic and index storage, resulting in certain usage costs.

# Topic Partition

Last updated : 2022-08-24 12:10:06

## Overview

Partition is the smallest read/write unit of CLS. A log topic can be divided into several partitions and must have at least one partition. CLS uses the value range of MD5 as the valid range and controls the overall throughput performance by merging or splitting partitions. A log topic supports up to 50 partitions. We recommend that you use and work with topic partitions rationally to prevent waste of resources.

Basic attributes of a partition:

- **Partition number:** every partition has a unique number under the same log topic, which is assigned by the system after partition creation or operation.
- **Partition range:** every partition has a left-closed and right-open interval.
- **Partition status:**
  - Read-write: the current partition allows read and write.
  - Read-only: the current partition is read-only and no data can be written to it.

Note :

Topic partitioning is a complex concept. You are advised to use the [Auto Split](#) function in actual practice. CLS automatically adjusts topic partitioning based on the volume of log data.

## Partition Range

With a partition range, logs can be written in the HashKey mode. The valid range of a log topic is the value range of MD5, which is `[00000000000000000000000000000000,ffffffffffffffffffffffffffffffffffff)`. All read-write partitions segment the entire value range of a log topic and each occupies a left-closed and right-open interval to ensure that every log record collected is written to the corresponding partition.

CLS provides two write modes: load-balancing mode and HashKey mode.

- **Load-balancing mode:** every data packet is written to a random log topic partition.
- **HashKey mode:** every data packet is written to the topic partition that contains the current Key value.

For example, a log topic has three read-write partitions and their ranges are as follows:

Partition No.	Status	Partition Range
1	Read-write	[00000000000000000000000000000000, 7fff]
2	Read-write	[7fff, a0000000000000000000000000000000]
3	Read-write	[a0000000000000000000000000000000, ffffffffffffffffffffffffffffffffff]

If the write mode is HashKey, log data with the Key value of `2fffffffffffffffffffffffffffffffff` is written to partition 1 and log data with the Key value of `9f000000000000000000000000000000` is written to partition 2.

## Read and Write Capability of Partitions

Each partition has a certain level of read and write capability. We recommend that you plan partition count based on the actual log traffic of your business. Split partitions when the traffic is higher than the read and write capability of the log topic, and merge partitions when the traffic is lower than the read and write capability of the log topic to save resources.

Feature	Item	Description
Frequency control	Write requests	Every partition supports a maximum of 500 QPS of write operations. It will reject requests and return the status code 429 with an error message of "SpeedQuotaExceed" when the limit is exceeded.
	Read requests	Every partition supports a maximum of 200 QPS of read operations. It will reject requests and return the status code 429 with an error message of "SpeedQuotaExceed" when the limit is exceeded.
Traffic throttling	Write traffic	Every partition supports the write traffic of up to 5 MB/sec. It will truncate data and return the status code 429 with an error message of "SpeedQuotaExceed" when the limit is exceeded.

## Partition Status

A partition can be in **read-write** or **read-only** mode. Only read-write partitions support data writing. Read-only partitions do not allow data writing but can be consumed within their lifetime. All partitions are readable and writable



when they are created, but merging and splitting operations will change the status to read-only.

### **Merging partitions**

Two adjacent read-write partitions can be merged into one partition. After two partitions are merged, the two original partitions become read-only, which only allow data consumption, but not data writing. The new partition is readable and writable and covers the range of the two original partitions.

### **Splitting a partition**

A read-write partition can be split into two partitions with smaller ranges. When splitting a partition, you need to specify the MD5 value of a split point, which must be larger than the value of the start point and smaller than the value of the end point. After a partition is split, the original partition becomes read-only, which only allows data consumption, but not data writing. The new partition is readable and writable and covers the range of the original partition.

# Limits

## LogListener Limits

Last updated : 2022-05-07 11:20:50

This document describes LogListener's capabilities and limits of file collection, configuration, resources, performance, and error handling related to data collection.

### File Collection Capabilities and Limits

Item	Capability and Limit
File encoding	Supports UTF8 and GBK codec. Note: only LogListener v2.6.2 and above support GBK codec.
Soft link	Supported.
Size of a single log	The size of a single-line log is limited to 512 KB. If the size of a single-line log exceeds 512 KB, the log will be truncated and only the first 512 KB is retained. For a multi-line log, after it is divided according to the first-line regular expression, the maximum limit for a single log is 1 MB.
Regular expression	Perl-compatible regular expressions are supported.
First log collection behavior	LogListener supports full/incremental collection policies: <ul style="list-style-type: none"><li>• Full collection: after LogListener is started for the first time after installation, all logs that meet requirements are collected, including files that have not been written.</li><li>• Incremental collection: after LogListener is started for the first time after installation, existing files will be collected from the latest position.</li></ul> <b>Note:</b> Full/Incremental collection policies are available starting from LogListener v2.6.2.
Log file rotation	Supported. (It is recommended that the file name after rotation not be matched by the wildcard collection path.)
Collection behavior when log parsing fails	We recommend you enable the feature of uploading parsing-failure logs. If the feature is enabled, parsing-failure logs will be uploaded to the preset index in the format of full text in a single line. Otherwise, the logs will be discarded.
File opening	LogListener opens files when reading files for log collection and closes the files when the reading is completed.

## Checkpoint Management Capabilities and Limits

Item	Capability and Limit
Checkpoint storage location	The default save path is the <code>data</code> directory in the LogListener installation directory. You can customize a save path in the <code>loglistener.conf</code> file.
Checkpoint storage policy	LogListener stores two copies of checkpoint metadata: <ul style="list-style-type: none"> <li>• One copy records the information only of checkpoints where upload is completed, and the information is persisted to a disk in real time.</li> <li>• The other copy records the information of checkpoints where upload is completed and checkpoints where upload is not completed, and the information is periodically persisted to a disk. Persistence takes precedence when the program exits.</li> </ul>

## Resource and Performance Capabilities and Limits

Item	Capability and Limit
Default CPU resource limits	LogListener does not limit CPU resources by default. You can configure CPU resource limits in the configuration file. Without CPU resource limits, the current implementation architecture of LogListener can achieve a maximum CPU usage of 110% (up to 100% for a single business thread and about 10% for a control thread).
Default memory resource limits	LogListener's default memory threshold is 2 GB. You can change the threshold to 300 MB or higher as needed.
Default bandwidth resource limits	LogListener does not limit bandwidth resources by default. You can configure bandwidth resource limits in the <code>loglistener.conf</code> file.
Resource overrun handling policy	If LogListener overruns CPU and memory resources for more than 5 minutes, LogListener will be forced to restart automatically.
Log compression	Collected logs are shipped as compact logs by default. You can modify log compression configuration in the <code>loglistener.conf</code> file.

Item	Capability and Limit
Number of monitored directories	The recommended maximum number of monitored directories is 5,000. If this limit is exceeded, log collection may fail.
Number of monitored files	The recommended maximum number of monitored files is 10,000. If this limit is exceeded, log collection may fail.

## Error Handling

Item	Capability and Limit
Network error handling	Except exceptions requiring special handling (such as log topic deletion), all other errors (such as network exceptions, timeout, frequency control, and overdue payment) will be retried.
Maximum retry timeout duration	If data fails to be sent after retrying for more than 1 hour, the system discards the data. The default behavior is to retry at an interval, and the retry interval becomes longer and longer until the maximum retry timeout duration is exceeded.
Maximum number of retries	The maximum number of retries can be set in the <code>loglistener.conf</code> configuration file and is not set by default. By default, the system retries until the maximum retry duration is exceeded, and then discards the data. If the maximum number of retries is set, the system retries until the maximum number of retries is exceeded, and then discards the data.

## File Collection Rules

Item	Capability and Limit
Log upload policy	LogListener automatically aggregates and uploads logs of the same file when any of the following aggregation conditions is met: 10,000 logs, the total logset size reaching 1 MB, or the log collection time exceeding 3 seconds.
File collection handling policy	A single target file (a file that can be matched by the collection path) can be uploaded to only one log topic. The same file cannot be matched by the collection paths of multiple topics. If you need to upload a file to multiple topics, use soft links. You can create different soft links for the same target file so that it can be collected to different topics via different soft links.

Item	Capability and Limit
Log collection delay	In the case of real-time collection, data collection, transmission, and storage to disks will be completed within 1 minute, and the logs are retrievable in the console. If the log volume is large, or LogListener can use only a limited amount of resources, there will be some collection delay.

## Machine Group Related Rules

Item	Capability and Limit
Machine group logic	<p>At present, machine groups are divided into the following two categories, and their usage methods are independent of and incompatible with each other. If the two usage methods are used together, LogListener will not be able to pull the correct collection configuration, resulting in no collection.</p> <ol style="list-style-type: none"> <li>1. IP machine group: a machine IP must be manually added to the machine group in the console, and the <code>group_label</code> field in <code>loglistener.conf</code> on the corresponding machine must be empty.</li> <li>2. Label machine group: the machine group label is set in the console, and the <code>group_label</code> field in <code>loglistener.conf</code> on the corresponding machine must be set to the same label.</li> </ol>
Relationships between machine groups and log topics	<ul style="list-style-type: none"> <li>• A single log topic can be bound to multiple machine groups.</li> <li>• A single machine group can be bound to multiple log topics.</li> </ul>
Relationships between machine groups and machines	<ul style="list-style-type: none"> <li>• A single machine can be added to multiple machine groups.</li> <li>• For an IP machine group, there is no limit to the number of machine groups that a machine can join.</li> <li>• For a label machine group, the maximum number of machine groups that a machine can join is 20.</li> </ul>
Limits on the labels of label machine groups	<ul style="list-style-type: none"> <li>• Currently, the label of a label machine group can contain up to 32 characters.</li> <li>• Up to 20 labels can be configured for a single label machine group.</li> </ul>

## Collection Path/Collection Blocklist Usage

Item	Capability and Limit
Collection blocklist	<p>This feature is used to specify the content to ignore in the collection path. Currently, collection blocklists support two modes:</p> <ul style="list-style-type: none"><li>• FILE mode: specify the files to ignore in the collection path.</li><li>• PATH mode: specify the subdirectories to ignore in the collection path. Wildcard filtering is supported.</li></ul> <p><b>Notes:</b></p> <ul style="list-style-type: none"><li>• The FILE and PATH modes can be used together.</li><li>• A collection blocklist is to exclude content from the collection path, and for both the FILE and PATH modes, the specified path to exclude must be a subset of the collection path.</li></ul>