日志服务

产品简介

产品文档

腾讯云

文档目录

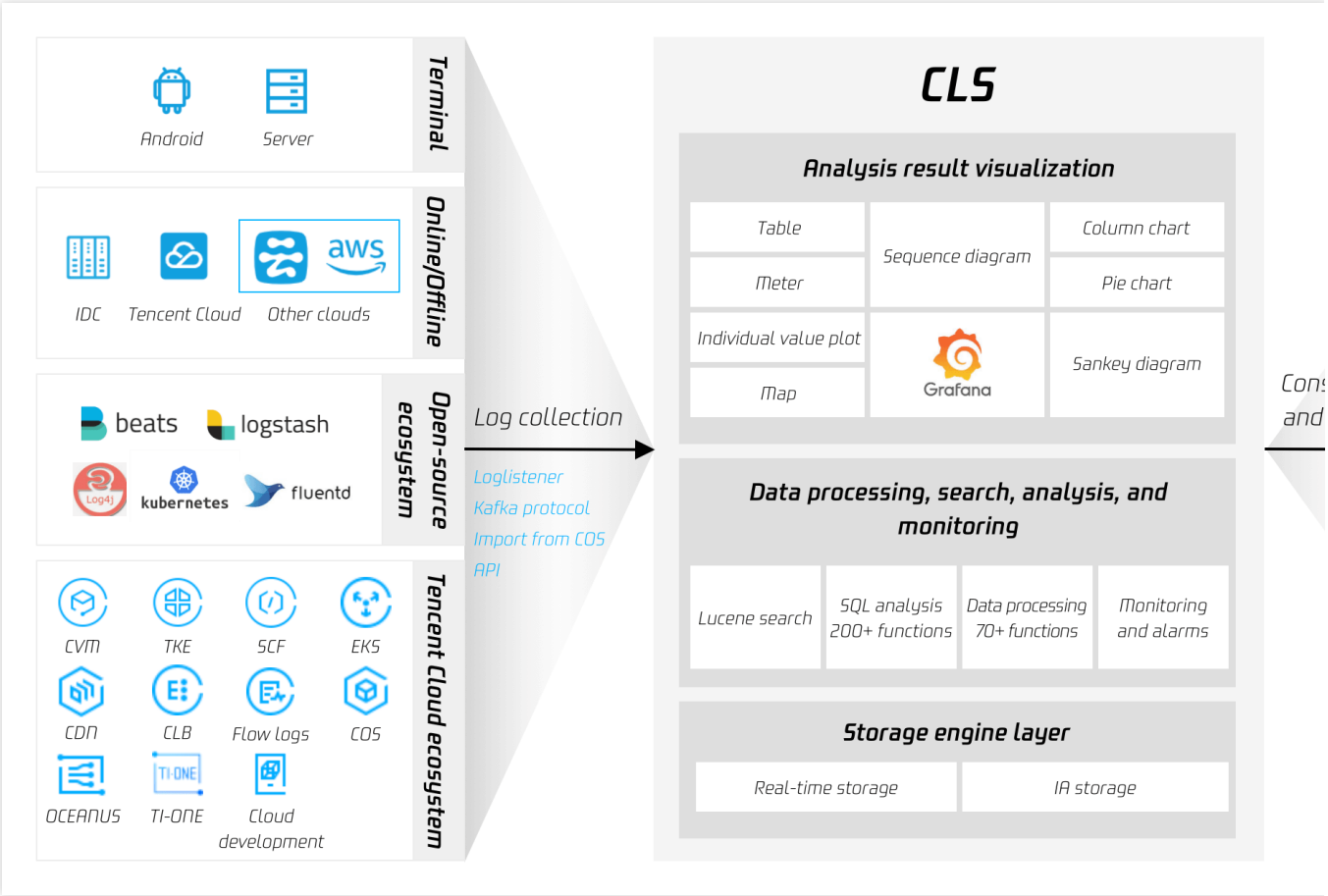
产品简介
产品概述

最近更新时间：2023-05-08 16:45:43

日志服务（Cloud Log Service，CLS）提供一站式的日志数据解决方案。您无需关注扩缩容等资源问题，五分钟快速便捷接入，即可享受日志的采集、存储、加工、检索分析、消费投递、生成仪表盘、告警等全方位稳定可靠服务。全面提升问题定位、指标监控的效率，极大降低日志运维门槛。

功能概览

日志服务主要提供以下功能：
日志采集：便捷实时采集跨地域、多渠道、多平台、不同数据源的日志数据，轻松采集多种其他腾讯云产品日志。
日志存储：提供两种存储类型：实时存储和低频存储。
日志检索分析：使用关键词检索日志，帮助用户快速定位异常日志，同时支持使用 SQL 对日志进行统计分析，获取日志条数随时间变化趋势、错误日志比例等统计指标。
日志数据加工：日志过滤、清洗、脱敏、富化、分发、结构化。
日志投递与消费：投递到腾讯云存储、中间件，消费到流计算。
仪表盘：将检索分析结果快速生成自定义 Dashborad。
告警：异常日志秒级告警，支持通过电话、短信、邮件和自定义接口回调等方式通知用户。



日志采集

日志服务目前支持 LogListener 、API、kafka 协议、对象存储（Cloud Object Storage，COS）导入等多种采集方式：

Loglistener 实时采集：使用腾讯云 LogListener 采集日志，便捷安装，服务稳定可靠安全、支持大部分主流 Linux 操作系统，兼顾高性能与低资源占用。

通过 API：无需安装 LogListener，直接调用 API 上传日志，支持多种语言。

kafka 协议：支持使用 Kafka Producer SDK 上传日志到 CLS。

COS 导入：将腾讯云对象存储中的数据导入到日志服务。

日志存储

根据用户对日志不同的检索时延性及日志处理能力需求，提供两种存储：

实时存储：适用于用户对日志有统计分析的需求，提供日志的秒级检索，实时统计分析，实时监控，流式消费等应用能力。

低频存储：适用于访问频率较低且无统计分析需求的日志，如审计归档日志。低频存储提供全文检索日志的能力，满足用户对历史日志的回溯检索，归档存储等诉求。长期存储下，整体使用成本相比实时存储降低80%。

日志检索分析

针对存储在 CLS 的日志进行实时检索分析，帮助用户快速定位异常日志、统计系统及业务指标。
日志检索：使用关键词按全文或字段检索日志。
统计分析：使用 SQL 灵活统计日志内的系统及业务指标并通过图表进行展示，兼容 SQL 92标准，支持200+ SQL 函数。例如按省份统计业务请求数，按时间查询请求错误率变化趋势。
性能优异：秒级返回查询结果，支持亿级别日志检索分析。

日志数据加工

日志流实时加工和分发：日志数据流式加工，实时生成加工结果，可按使用场景分发到多个主题。
日志过滤清洗：脏数据清洗，按条件过滤日志，然后再开启索引，可以有效节约费用。
日志结构化：将文本日志加工成为结构化数据，方便后续的检索分析、仪表盘、告警。
日志脱敏：日志文本中的敏感信息脱敏。例如手机号、身份证等。

日志投递与消费

用户可以将指定日志投递至其他云产品中，如 COS、Ckafka，用户也可以将 CLS 日志消费到腾讯云 Oceanus。
日志投递至 COS：日志服务支持将日志数据投递至您账户下的 COS 存储桶中，COS 存储费用较低，推荐使用。
日志投递至 Ckafka：适用于将 Ckafka 作为 source 的数据分析和入库场景。
日志实时消费：直接消费日志数据到大数据组件，例如 Flink、Oceanus、Flume。

仪表盘

仪表盘提供数据分析全局视图,您可以在仪表盘查看多个基于查询与分析结果的统计图表。
仪表盘：仪表盘支持保存多个基于查询与分析结果的统计图表，构成场景化的分析大盘。
预置仪表盘：面向 CLB、TKE、COS 等腾讯云生态产品，CLS 提供预置仪表盘，实现常见监控能力的开箱即用。
模板变量：模板变量支持数据源变量与快速过滤，实现仪表盘图表统计分析对象与维度的快速切换，千变万化支持更复杂的业务场景。

告警

日志内出现较多错误日志或系统及业务指标超出阈值时秒级告警，主动发现系统及业务异常问题。
通知渠道：支持电话、短信、邮件、微信、企业微信和自定义接口回调（可对接钉钉及飞书）。
多维分析：触发告警时可针对原始日志进行额外的检索分析，将结果附加在告警通知中，辅助定位告警原因。

产品优势

功能优势

最近更新时间：2024-01-20 16:29:56

功能丰富

提供采集、存储、检索、转存投递等功能一站式日志服务。
采集客户端 LogListener 提供单行/多行全文、分隔符、JSON、正则等日志结构化解析方式。
提供多种数据接入方式，用户可根据业务情况选择适合的接入方式，详情请参考 采集概述。
提供丰富的检索语法，方便用户进行关键词查询、模糊查询、范围查询等日志查询操作。

稳定可靠

日志服务采用高可扩展性的分布式存储架构，支持横向水平扩容，服务弹性伸缩，轻松存储管理海量日志数据。
日志服务后端存储采用多副本机制管理存储日志数据，为数据安全提供可靠性保障。

简单高效

采集端 LogListener 提供界面式的配置方式，配置简单直观，使用 LogListener 可快速接入日志服务。
数据写入 CLS 即可被消费，亿级数据查询支持秒级返回结果。
服务按实际用量收费，无需单独搭建和运维日志系统，避免了资源闲置浪费问题。

生态扩展

部分云产品日志已接入 CLS，详情请参考 接入列表。
日志数据投递 COS，满足对日志数据长时间归档存储的需求。
日志数据投递 Ckafka，满足对日志数据实时消费的需求，便于进一步处理分析。

可用地域

最近更新时间：2024-01-20 16:30:32

概述

用户在使用日志服务（Cloud Log Service，CLS）过程中可选择在不同地域创建日志集与日志主题。地域（Region）是指物理的数据中心的地理区域，不同地域之间网络完全隔离。用户可以根据自己的业务场景以及目标用户所在的地理位置选择就近的地域，以降低日志数据的访问时延、提高访问速度。

可用地域

地域	简称
北京	ap-beijing
广州	ap-guangzhou
上海	ap-shanghai
成都	ap-chengdu
南京	ap-nanjing
重庆	ap-chongqing
中国香港	ap-hongkong
硅谷	na-siliconvalley
弗吉尼亚	na-ashburn
新加坡	ap-singapore
曼谷	ap-bangkok
孟买	ap-mumbai
法兰克福	eu-frankfurt
东京	ap-tokyo
首尔	ap-seoul

莫斯科	eu-moscow
雅加达	ap-jakarta
多伦多	na-toronto
圣保罗	sa-saopaulo

说明：
如果日志服务中接入了其他云产品，请您尽量选择与其他云产品相同的地域。相同地域的云产品之间通过内网读写数据，能有效降低延迟、提高访问速度。

域名

日志服务在不同模块使用的域名有所区别，具体如下：
LogListener
日志服务API3.0
API 上传日志
Kafka上传日志
Kafka 消费日志
LogListener是 CLS 所提供的日志采集客户端，可将机器本地的日志上报至 CLS，其所使用的域名如下：

地域	简称	内网域名	外网域名
北京	ap-beijing	ap-beijing.cls.tencentyun.com	ap-beijing.cls.tencentcs.com
广州	ap-guangzhou	ap-guangzhou.cls.tencentyun.com	ap-guangzhou.cls.tencentcs.com
上海	ap-shanghai	ap-shanghai.cls.tencentyun.com	ap-shanghai.cls.tencentcs.com
成都	ap-chengdu	ap-chengdu.cls.tencentyun.com	ap-chengdu.cls.tencentcs.com
南京	ap-nanjing	ap-nanjing.cls.tencentyun.com	ap-nanjing.cls.tencentcs.com
重庆	ap-chongqing	ap-chongqing.cls.tencentyun.com	ap-chongqing.cls.tencentcs.com
中国香港	ap-hongkong	ap-hongkong.cls.tencentyun.com	ap-hongkong.cls.tencentcs.com
硅谷	na-siliconvalley	na-siliconvalley.cls.tencentyun.com	na-siliconvalley.cls.tencentcs.com

弗吉尼亚	na-ashburn	na-ashburn.cls.tencentyun.com	na-ashburn.cls.tencentcs.com
新加坡	ap-singapore	ap-singapore.cls.tencentyun.com	ap-singapore.cls.tencentcs.com
曼谷	ap-bangkok	ap-bangkok.cls.tencentyun.com	ap-bangkok.cls.tencentcs.com
孟买	ap-mumbai	ap-mumbai.cls.tencentyun.com	ap-mumbai.cls.tencentcs.com
法兰克福	eu-frankfurt	eu-frankfurt.cls.tencentyun.com	eu-frankfurt.cls.tencentcs.com
东京	ap-tokyo	ap-tokyo.cls.tencentyun.com	ap-tokyo.cls.tencentcs.com
首尔	ap-seoul	ap-seoul.cls.tencentyun.com	ap-seoul.cls.tencentcs.com
莫斯科	eu-moscow	eu-moscow.cls.tencentyun.com	eu-moscow.cls.tencentcs.com
雅加达	ap-jakarta	ap-jakarta.cls.tencentyun.com	ap-jakarta.cls.tencentcs.com
多伦多	na-toronto	na-toronto.cls.tencentyun.com	na-toronto.cls.tencentcs.com
圣保罗	sa-saopaulo	sa-saopaulo.cls.tencentyun.com	sa-saopaulo.cls.tencentcs.com

日志服务 API 3.0 是 CLS 最新版本的 API，符合腾讯云统一的 API 规范，可通过 API 管理日志主题和告警策略等资源，其所使用的域名如下：
通过外网访问时，也可使用统一域名 cls.tencentcloudapi.com，将根据调用接口时客户端所在位置，自动解析到最近的某个具体地域的服务器，对时延敏感的业务，建议指定带地域的域名。

地域	简称	内网域名	外网域名
北京	ap-beijing	cls.internal.tencentcloudapi.com	cls.ap-beijing.tencentcloudapi.com
广州	ap-guangzhou	cls.internal.tencentcloudapi.com	cls.ap-guangzhou.tencentcloudapi.com
上海	ap-shanghai	cls.internal.tencentcloudapi.com	cls.ap-shanghai.tencentcloudapi.com
成都	ap-chengdu	cls.internal.tencentcloudapi.com	cls.ap-chengdu.tencentcloudapi.com
南京	ap-nanjing	cls.internal.tencentcloudapi.com	cls.ap-nanjing.tencentcloudapi.com

重庆	ap-chongqing	cls.internal.tencentcloudapi.com	cls.ap-chongqing.tencentcloudapi.com
中国香港	ap-hongkong	cls.internal.tencentcloudapi.com	cls.ap-hongkong.tencentcloudapi.com
硅谷	na-siliconvalley	cls.internal.tencentcloudapi.com	cls.na-siliconvalley.tencentcloudapi.com
弗吉尼亚	na-ashburn	cls.internal.tencentcloudapi.com	cls.na-ashburn.tencentcloudapi.com
新加坡	ap-singapore	cls.internal.tencentcloudapi.com	cls.ap-singapore.tencentcloudapi.com
曼谷	ap-bangkok	cls.internal.tencentcloudapi.com	cls.ap-bangkok.tencentcloudapi.com
孟买	ap-mumbai	cls.internal.tencentcloudapi.com	cls.ap-mumbai.tencentcloudapi.com
法兰克福	eu-frankfurt	cls.internal.tencentcloudapi.com	cls.eu-frankfurt.tencentcloudapi.com
东京	ap-tokyo	cls.internal.tencentcloudapi.com	cls.ap-tokyo.tencentcloudapi.com
首尔	ap-seoul	cls.internal.tencentcloudapi.com	cls.ap-seoul.tencentcloudapi.com
莫斯科	eu-moscow	cls.internal.tencentcloudapi.com	cls.eu-moscow.tencentcloudapi.com
雅加达	ap-jakarta	cls.internal.tencentcloudapi.com	cls.ap-jakarta.tencentcloudapi.com
多伦多	na-toronto	cls.internal.tencentcloudapi.com	cls.na-toronto.tencentcloudapi.com
圣保罗	sa-saopaulo	cls.internal.tencentcloudapi.com	cls.sa-saopaulo.tencentcloudapi.com

以下域名用于 API 上传日志，具体域名如下：（除上传日志外的其他接口，请更新至日志服务 API 3.0。）

地域	简称	内网域名	外网域名
北京	ap-beijing	ap-beijing.cls.tencentyun.com	ap-beijing.cls.tencentcs.com
广州	ap-guangzhou	ap-	ap-guangzhou.cls.tencentcs.com

		guangzhou.cls.tencentyun.com	
上海	ap-shanghai	ap-shanghai.cls.tencentyun.com	ap-shanghai.cls.tencentcs.com
成都	ap-chengdu	ap-chengdu.cls.tencentyun.com	ap-chengdu.cls.tencentcs.com
南京	ap-nanjing	ap-nanjing.cls.tencentyun.com	ap-nanjing.cls.tencentcs.com
重庆	ap-chongqing	ap-chongqing.cls.tencentyun.com	ap-chongqing.cls.tencentcs.com
中国香港	ap-hongkong	ap-hongkong.cls.tencentyun.com	ap-hongkong.cls.tencentcs.com
硅谷	na-siliconvalley	na-siliconvalley.cls.tencentyun.com	na-siliconvalley.cls.tencentcs.com
弗吉尼亚	na-ashburn	na-ashburn.cls.tencentyun.com	na-ashburn.cls.tencentcs.com
新加坡	ap-singapore	ap-singapore.cls.tencentyun.com	ap-singapore.cls.tencentcs.com
曼谷	ap-bangkok	ap-bangkok.cls.tencentyun.com	ap-bangkok.cls.tencentcs.com
孟买	ap-mumbai	ap-mumbai.cls.tencentyun.com	ap-mumbai.cls.tencentcs.com
法兰克福	eu-frankfurt	eu-frankfurt.cls.tencentyun.com	eu-frankfurt.cls.tencentcs.com
东京	ap-tokyo	ap-tokyo.cls.tencentyun.com	ap-tokyo.cls.tencentcs.com
首尔	ap-seoul	ap-seoul.cls.tencentyun.com	ap-seoul.cls.tencentcs.com
莫斯科	eu-moscow	eu-moscow.cls.tencentyun.com	eu-moscow.cls.tencentcs.com
雅加达	ap-jakarta	ap-jakarta.cls.tencentyun.com	ap-jakarta.cls.tencentcs.com
多伦多	na-toronto	na-toronto.cls.tencentyun.com	na-toronto.cls.tencentcs.com
圣保罗	sa-saopaulo	sa-saopaulo.cls.tencentyun.com	sa-saopaulo.cls.tencentcs.com

使用 Kafka 协议上传日志 支持使用 Kafka Producer SDK 和其他 Kafka 相关 agent 上传日志到日志服务，其所使用的域名如下：

地域	简称	内网域名	外网域名
北京	ap-beijing	bj-producer.cls.tencentyun.com	bj-producer.cls.tencentcs.com

广州	ap-guangzhou	gz-producer.cls.tencentyun.com	gz-producer.cls.tencentcs.com
上海	ap-shanghai	sh-producer.cls.tencentyun.com	sh-producer.cls.tencentcs.com
成都	ap-chengdu	cd-producer.cls.tencentyun.com	cd-producer.cls.tencentcs.com
南京	ap-nanjing	nj-producer.cls.tencentyun.com	nj-producer.cls.tencentcs.com
重庆	ap-chongqing	cq-producer.cls.tencentyun.com	cq-producer.cls.tencentcs.com
中国香港	ap-hongkong	hk-producer.cls.tencentyun.com	hk-producer.cls.tencentcs.com
硅谷	na-siliconvalley	usw-producer.cls.tencentyun.com	usw-producer.cls.tencentcs.com
弗吉尼亚	na-ashburn	use-producer.cls.tencentyun.com	use-producer.cls.tencentcs.com
新加坡	ap-singapore	sg-producer.cls.tencentyun.com	sg-producer.cls.tencentcs.com
曼谷	ap-bangkok	th-producer.cls.tencentyun.com	th-producer.cls.tencentcs.com
孟买	ap-mumbai	in-producer.cls.tencentyun.com	in-producer.cls.tencentcs.com
法兰克福	eu-frankfurt	de-producer.cls.tencentyun.com	de-producer.cls.tencentcs.com
东京	ap-tokyo	jp-producer.cls.tencentyun.com	jp-producer.cls.tencentcs.com
首尔	ap-seoul	kr-producer.cls.tencentyun.com	kr-producer.cls.tencentcs.com
莫斯科	eu-moscow	ru-producer.cls.tencentyun.com	ru-producer.cls.tencentcs.com
雅加达	ap-jakarta	jkt-producer.cls.tencentyun.com	jkt-producer.cls.tencentcs.com

多伦多	na-toronto	ca-producer.cls.tencentyun.com	ca-producer.cls.tencentcs.com

使用 Kafka 协议消费日志 支持使用 Kafka Consumer SDK 和其他大数据组件消费到用户的数据仓库，其所使用的域名如下：

地域	简称	内网域名	外网域名
北京	ap-beijing	kafkaconsumer-ap-beijing.cls.tencentyun.com	kafkaconsumer-ap-beijing.cls.tencentcs.com
广州	ap-guangzhou	kafkaconsumer-ap-guangzhou.cls.tencentyun.com	kafkaconsumer-ap-guangzhou.cls.tencentcs.com
上海	ap-shanghai	kafkaconsumer-ap-shanghai.cls.tencentyun.com	kafkaconsumer-ap-shanghai.cls.tencentcs.com
成都	ap-chengdu	kafkaconsumer-ap-chengdu.cls.tencentyun.com	kafkaconsumer-ap-chengdu.cls.tencentcs.com
南京	ap-nanjing	kafkaconsumer-ap-nanjing.cls.tencentyun.com	kafkaconsumer-ap-nanjing.cls.tencentcs.com
重庆	ap-chongqing	kafkaconsumer-ap-chongqing.cls.tencentyun.com	kafkaconsumer-ap-chongqing.cls.tencentcs.com
中国香港	ap-hongkong	kafkaconsumer-ap-hongkong.cls.tencentyun.com	kafkaconsumer-ap-hongkong.cls.tencentcs.com
硅谷	na-siliconvalley	kafkaconsumer-na-siliconvalley.cls.tencentyun.com	kafkaconsumer-na-siliconvalley.cls.tencentcs.com
弗吉尼亚	na-ashburn	kafkaconsumer-na-ashburn.cls.tencentyun.com	kafkaconsumer-na-ashburn.cls.tencentcs.com
新加坡	ap-singapore	kafkaconsumer-ap-singapore.cls.tencentyun.com	kafkaconsumer-ap-singapore.cls.tencentcs.com
曼谷	ap-bangkok	kafkaconsumer-ap-bangkok.cls.tencentyun.com	kafkaconsumer-ap-bangkok.cls.tencentcs.com
孟买	ap-mumbai	kafkaconsumer-ap-mumbai.cls.tencentyun.com	kafkaconsumer-ap-mumbai.cls.tencentcs.com
法兰克福	eu-frankfurt	kafkaconsumer-eu-frankfurt.cls.tencentyun.com	kafkaconsumer-eu-frankfurt.cls.tencentcs.com

东京	ap-tokyo	kafkaconsumer-ap-tokyo.cls.tencentyun.com	kafkaconsumer-ap-tokyo.cls.tencentcs.com
首尔	ap-seoul	kafkaconsumer-ap-seoul.cls.tencentyun.com	kafkaconsumer-ap-seoul.cls.tencentcs.com
莫斯科	eu-moscow	kafkaconsumer-eu-moscow.cls.tencentyun.com	kafkaconsumer-eu-moscow.cls.tencentcs.com
雅加达	ap-jakarta	kafkaconsumer-ap-jakarta.cls.tencentyun.com	kafkaconsumer-ap-jakarta.cls.tencentcs.com
多伦多	na-toronto	kafkaconsumer-na-toronto.cls.tencentyun.com	kafkaconsumer-na-toronto.cls.tencentcs.com

基本概念

日志

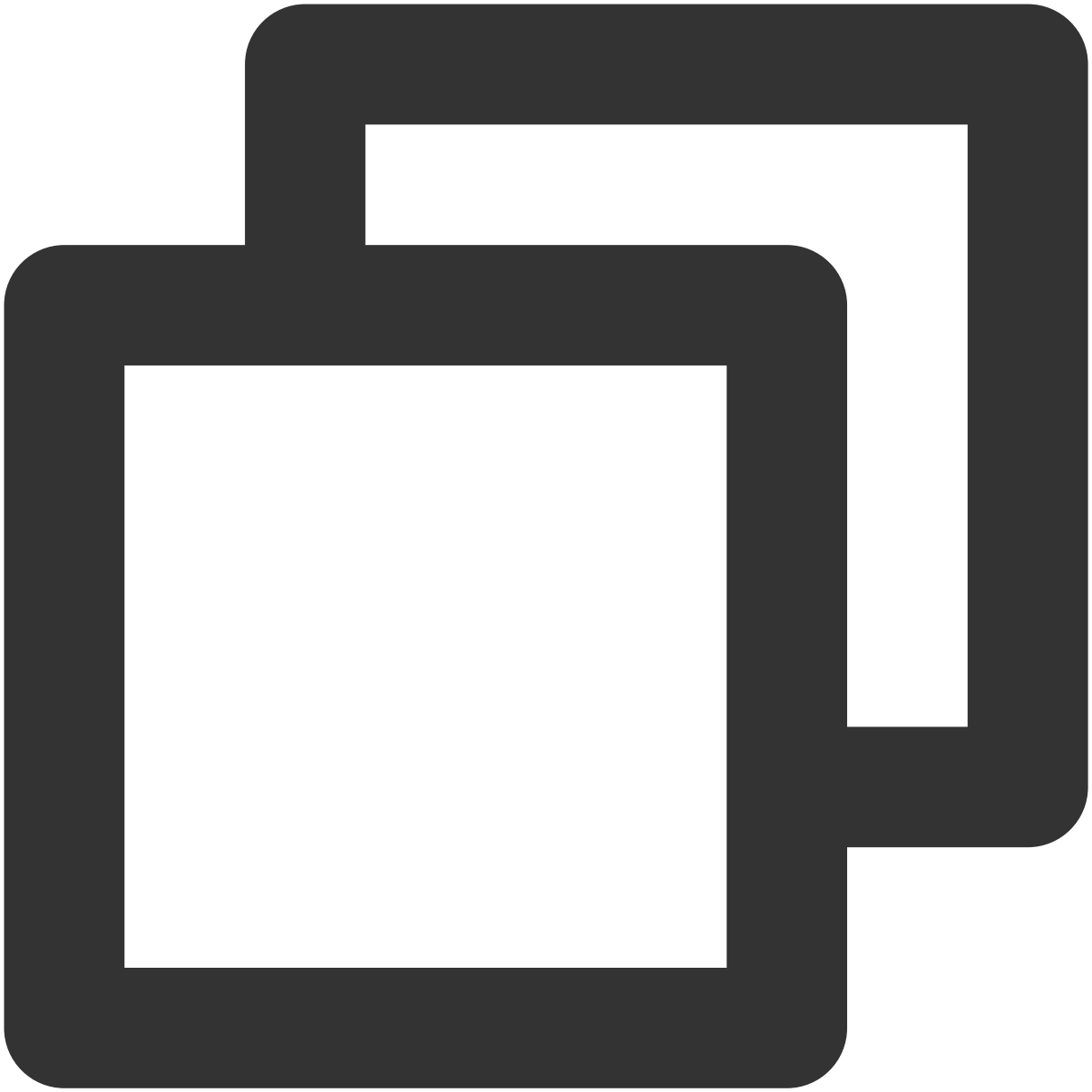
最近更新时间：2024-01-20 16:32:07

日志

日志（Log）是应用系统运行过程中产生的记录数据，如用户操作日志、接口访问日志、系统错误日志等。日志通常以文本的形式存储在应用系统所在的机器上，一条系统运行记录对应的日志可能为一行文本（单行日志），也可能为多行文本（多行日志）。

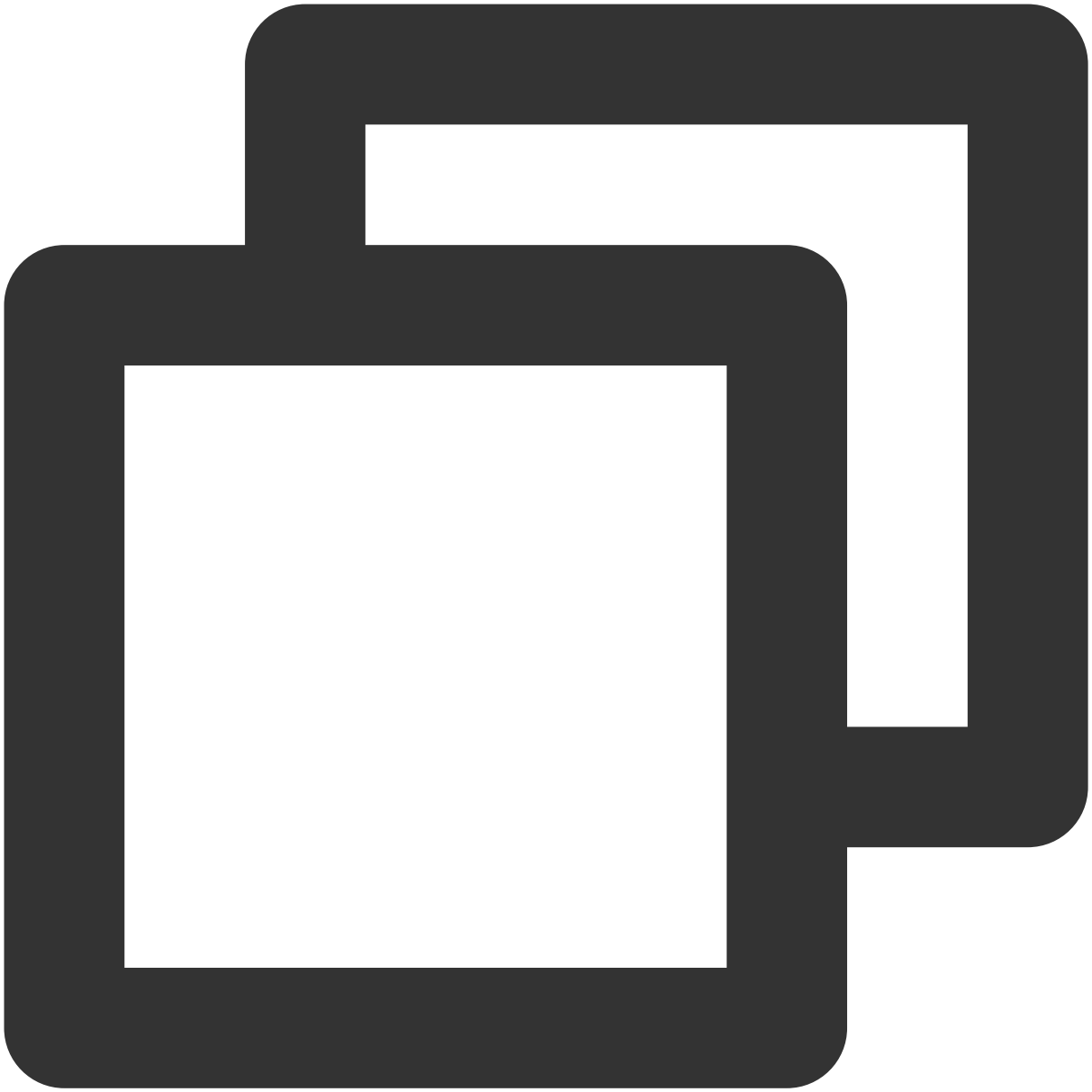
日志可通过 LogListener 上传至日志服务（Cloud Log Service，CLS），也可以通过 API、SDK 等其他方式上传。

单行日志示例：



```
59.x.x.x - - [06/Aug/2019:12:12:19 +0800] "GET /nginx-logo.png HTTP/1.1" 200 368 "h
```

多行日志示例：



```
java.net.SocketTimeoutException:Receive timed out
  at j.n.PlainDatagramSocketImpl.receive0(Native Method)[na:1.8.0_151]
  at j.n.AbstractPlainDatagramSocketImpl.receive(AbstractPlainDatagramSocketImpl.
  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]
```

对于一条日志，其主要组成部分如下：

组成部分	含义	示例
__TIMESTAMP__	日志时间，采用毫秒格式的UNIX时间戳	1640005601188
__FILENAME__	日志来源文件	/data/log/nginx/access.log
__SOURCE__	日志来源 IP	10.0.1.2
日志正文	日志主体内容，以 key:value 的形式存储，key 为字段名称，value 为字段值。 日志提取模式为单行全文或多行全文（即原始日志不经切分，直接将整条日志上报）时，整条日志存储在__CONTENT__字段下。 日志提取模式为其他模式（例如通过分割符切分日志）时，原始日志里的每一部分对应一个 key:value	单行全文或多行全文模式： 10.20.20.10;[2018-07-16 13:12:57];GET /online/sample HTTP/1.1;200 其他模式： IP: 10.20.20.10 request: GET /online/sample HTTP/1.1 status: 200 time: [2018-07-16 13:12:57]
元数据	对日志的简单描述或归类，例如 TKE 日志中，某条日志所属的集群或容器，以 key:value 的形式存储，key 以__TAG__.开头	__TAG__.clusterId:1skzv59c

日志组

日志组（LogGroup）是一个包含多条日志的集合。在上传日志的过程中，为提高数据读写效率，将多条日志打包成一个日志组，并以日志组为单位上传到 CLS。

一个日志组里的日志具有相同的基本信息（__TIMESTAMP__、__FILENAME__、__SOURCE__和元数据等）。

日志主题与日志集

最近更新时间：2024-01-20 16:32:53

日志主题

日志主题（Topic）是日志数据在日志服务（Cloud Log Service，CLS）平台进行采集、存储、检索和分析的基本单元，采集到的海量日志以日志主题为单元进行管理，包括采集规则配置、保存时间配置、日志检索分析以及日志下载/消费/投递等。

一个日志主题通常对应某一个应用/服务，建议将同一个应用/服务在不同机器上的同类日志采集到同一个日志主题。例如，某支付服务（payService）部署在数十台机器上，包含访问日志（access_log）和错误日志（error_log）两类日志。可创建 payService_access_log_topic 和 payService_error_log_topic 两个日志主题，分别对应这数十台机器上的两类日志，通过这两个日志主题即可完成数十台机器上的所有日志的集中检索和分析。

日志主题与应用/服务之间并非严格的一一对应关系，如果两个服务之间的日志结构相似度较高，且经常需要集中分析日志，也可以将这两个服务的日志上报至同一日志主题下。

日志集

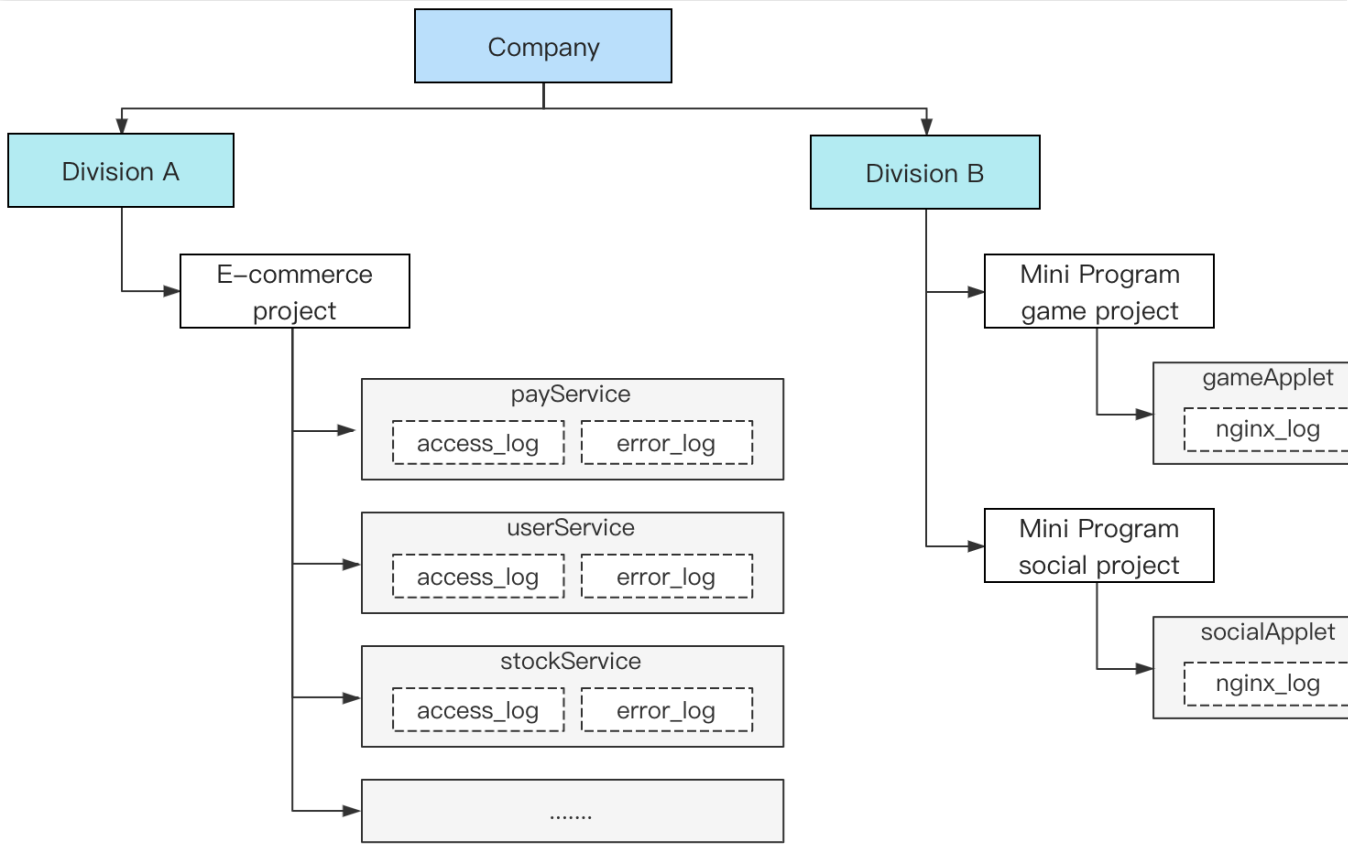
日志集（Logset）是对日志主题的分类，一个日志集可包含多个日志主题。日志集本身不存储任何日志数据，仅方便用户管理日志主题。

一个日志集通常对应公司内的某一个项目/业务，建议将某个项目/业务下的多个应用/服务的日志主题归属到同一个日志集下。例如，公司某电商项目下包含多个服务（支付服务 payService、用户服务 userService、库存管理服务 stockService 等），可创建一个日志集 e_commerce_logset，将这些服务的日志主题均归属至该日志集下。这样当公司有多个项目时，具体的项目人员只需要查看所属项目对应的日志集下的日志主题即可，其它项目的日志主题不会对其产生干扰。

注意：

新建日志主题时可指定其归属的日志集，保存后不能再变更。

场景示例



如上图，该公司有两个部门：

部门 A 有一个电商项目，采用微服务架构，每个服务均包含访问日志（access_log）和错误日志（error_log）两类日志。

部门 B 有两个项目，分别为小程序游戏项目和小程序社交项目，技术架构比较简单，均各有一个 Nginx 的访问日志（nginx_log）。

使用 CLS 监控上述这些应用日志时，可创建如下的日志集及日志主题：

日志集	日志主题	标签
e_commerce_logset	payService_access_log_topic	部门:部门A
e_commerce_logset	payService_error_log_topic	部门:部门A
e_commerce_logset	userService_access_log_topic	部门:部门A
e_commerce_logset	userService_error_log_topic	部门:部门A
e_commerce_logset	stockService_access_log_topic	部门:部门A
e_commerce_logset	stockService_error_log_topic	部门:部门A
e_commerce_logset	.....	部门:部门A
gameApplet_logset	gameApplet_nginx_log_topic	部门:部门B

socialApplet_logset	socialApplet_nginx_log_topic	部门:部门B

其中的标签用来区分日志集及日志主题归属的部门，结合权限策略还可以控制每个部门的人员仅可查看所属部门的数据。

对于部门 A，e_commerce_logset 日志集涵盖了其电商业务下的所有服务的日志主题，后续如果新增其他的项目，新建一个日志集即可。

对于部门 B，虽然目前的技术架构比较简单，总共只有两类日志，但却创建了两个日志集，每个日志集只有一个日志主题，是出于如下目的：

支持后续架构扩展：如果业务规模上升，技术架构也演变为微服务架构后，可继续沿用当前的日志集，在当前日志集下新增日志主题即可，不同的项目之间互不影响。

灵活应对项目调整：如果把整个部门作为一个日志集，当其中的某个项目需要独立为一个部门，或需调整至另一个部门时，由于日志主题不能直接变更其归属的日志集，调整将变得非常麻烦，可能需要重新采集日志。而每个项目分别对应一个日志集时，则不存在该情况，只需调整日志集和日志主题对应的标签即可。

机器组

最近更新时间：2024-01-20 16:33:21

机器组

机器组（MachineGroup）是一组需要采集日志的机器列表，日志服务通过机器组来管理所有需要通过 LogListener 采集日志的机器。
一个机器组可以包含多台机器。当应用/服务部署在多台机器上，且这些机器的日志文件路径相同时，便可将其归为一个机器组。这样在控制台只需要配置一次日志数据采集规则，便可批量生效至机器组内的所有机器。
机器组可关联至一个日志主题，即机器组内所有的日志均上报至同一个日志主题；也可以关联至多个日志主题，即机器组内不同路径的日志分别上报至不同的日志主题。
机器组具备两种定义方式：
IP 地址：添加 IP 地址列表到机器组，机器组将自动添加这些IP对应的机器。
标识（Label）：安装 LogListener 时为所在机器添加标识，机器组将自动添加包含这些标识的机器。

场景示例

某电商项目，共有6台机器及三个服务。其部署方式如下表：
payService 部署在2台机器上，有2个日志文件路径。userService和stockService 混部在4台机器上，共有4个日志文件路径。每个服务的每类日志（访问日志 access_log 和错误日志 error_log）均需上报至单独的日志主题。

机器	部署服务	日志文件路径
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
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

在机器上部署 LogListener 时，可为每台机器按其上运行的服务添加标识，具体如下：

机器	部署服务	标识
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

然后在控制台创建3个机器组，采用标识方式进行定义，分别为 payService、userService 和 stockService，再在日志主题中关联该机器组并添加对应的采集配置即可完成日志上报。

日志主题	关联机器组	采集路径
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

后续服务需要扩容时，只需要在新增的机器上添加部署的服务作为标识，即可自动将新增的机器添加至对应的机器组并采集日志，极大的提升运维部署效率。

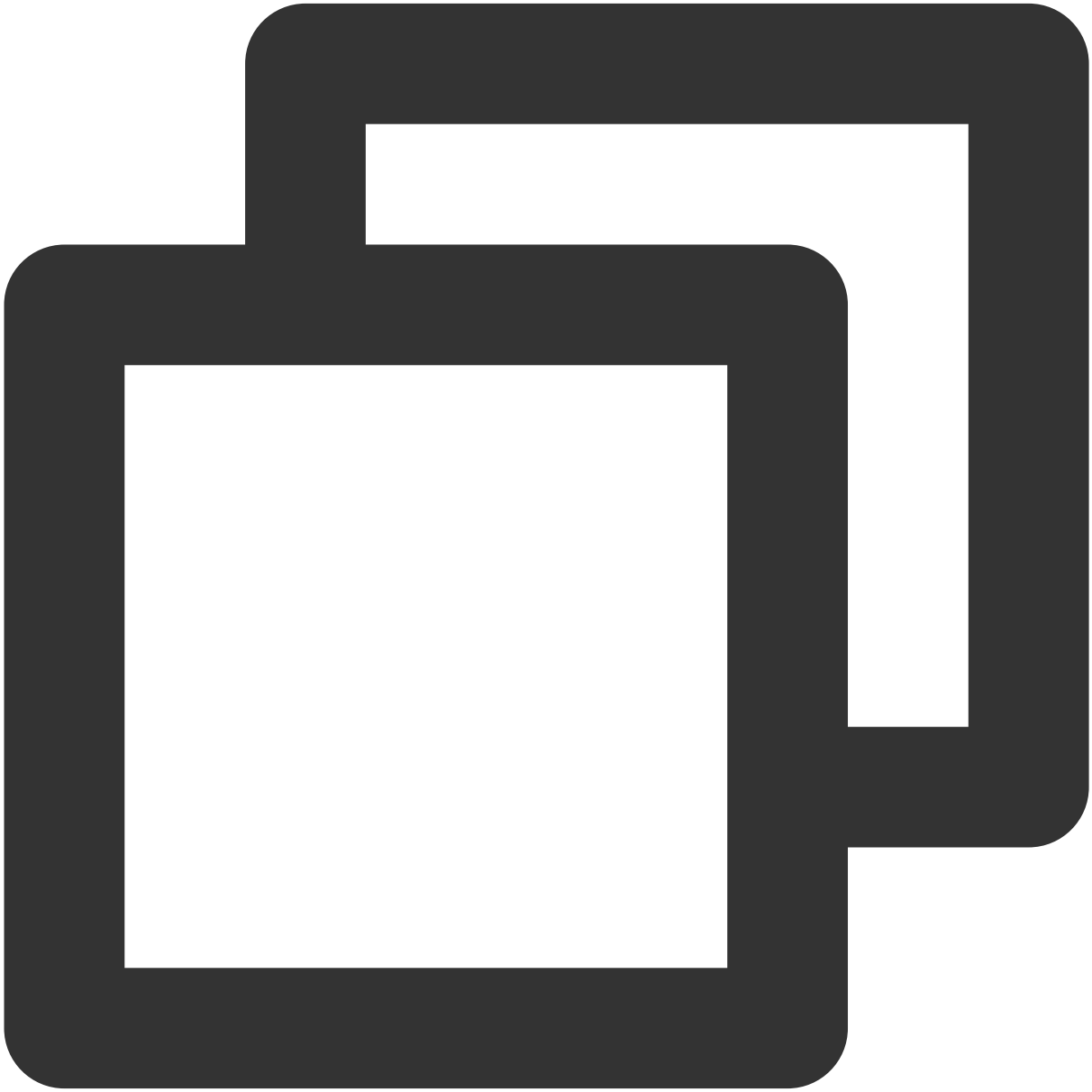
分词与索引

最近更新时间：2024-01-20 16:33:55

分词

分词示意

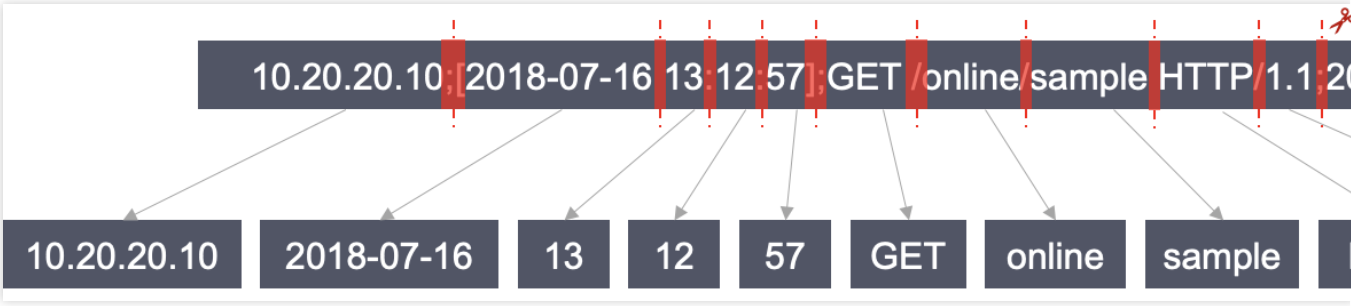
检索一个较长的日志时，通常只使用其中一部分内容进行检索。例如：需要检索出包含 `sample` 的如下完整日志：



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

由于日志全文很长，除 `sample` 外还有很多其他内容，其与检索条件并不是直接相等，因此不能直接使用 `sample` 作为检索条件来检索该日志。为满足该检索需求，需要将日志全文切分为多个片段，每个片段称之为一个“词”，而这个过程称之为“分词”。

例如：按符号对上述示例日志进行切分，只要出现了 `@&()='",;:<>[]{}/ \\n\\t\\r` 范围内的符号，就切分日志，将得到如下的词：



检索条件为 `sample` 时，上述分词后的日志包含 `sample`，即认为符合检索条件。

检索条件本身也会进行分词，例如下图3种检索条件：

检索条件A：`\\/online\\/login`

其中 `\\` 用于转义 `/` 符号（该符号为检索语法保留符号，因此需要转义）。

转义后的 `/` 符号是分词符，因此实际的检索条件为 `online OR login`，日志中只要包含 online **或** login，即符合检索条件。

上述示例日志**符合**该检索条件。

检索条件B：`"/online/login"`

由于双引号的存在，`/` 符号无需再进行转义。

双引号内的内容同样会分为两个词，但双引号表示日志需**同时存在**这两个词，且两个词**顺序严格一致**才符合检索条件。

上述示例日志不包含 `login`，**不符合**该检索条件。

检索条件C： `"/online/sample"`

上述示例日志同时包含 `online` 和 `sample` ，且顺序与检索条件一致，**符合**该检索条件。

在大多数使用场景下，检索条件如果包含分词符，建议使用双引号语法进行检索，既能避免添加转义符，又能更加精准的检索到所需的日志。更多检索语法说明参见 语法与规则。

分词设置

日志分词依据包含两类，可在 配置索引 中设置。

分词符：可自定义需要按照哪些符号对日志进行切分，支持英文符号及\\n\\t\\r。在上面的例子中， `@&()='",;:<>[]{}/ \\n\\t\\r` 即为分词符。

是否包含中文：中文较为特殊，不能使用中文符号作为分词符，而且仅按照符号对中文进行分词也往往达不到预期效果。例如，日志为 `用户登录失败，密码错误` ，需要使用 `"登录失败"` 进行检索，是不能通过符号对日志进行切分来满足检索需求的。此时可在 配置索引 中设置该日志“包含中文”，日志服务（Cloud Log Service，CLS）将自动将日志中的每一个汉字切分为独立的词。

索引

为了快速检索出需要的日志，CLS 对上传至平台的日志进行包括分词在内的很多预处理，这个过程称之为创建“索引”。索引决定了日志能够以什么样的条件来进行检索和分析，因此在上传日志数据前，需要为日志主题设置一个合理的索引规则，以方便后续检索分析，详细介绍请参考 配置索引。

索引规则仅针对新上传的日志数据生效，历史数据不生效，创建索引的过程中会产生索引流量及索引存储量，存在一定的使用成本。

主题分区

最近更新时间：2024-01-20 16:34:21

概述

主题分区（Partition）是日志服务（Cloud Log Service，CLS）的最小读写单元。一个日志主题可以划分多个主题分区，但至少有一个分区。日志服务将 MD5 取值范围作为有效区间范围，通过合并或分裂操作可以自由划分区间，从而控制服务的整体吞吐性能。每个日志主题最多支持50个分区数，建议合理操作使用主题分区，避免资源浪费。

主题分区的基本属性信息：

分区编号：每个分区在同一个日志主题下有唯一编号，该编号在创建或操作后由系统确定。

分区范围：每个分区均有区间范围，每个区间范围均为左闭右开区间。

分区状态：

读写态：表示当前分区可以进行读写操作。

只读态：表示当前分区仅允许进行读操作，不可再写入数据。

说明：

主题分区概念较为复杂，实际使用过程中建议使用 自动分裂 功能，CLS 将根据日志数据量自动调整主题分区，无需过多关注。

分区范围

分区范围主要用于支持日志按指定 HashKey 的模式写入，一个日志主题的有效范围为 MD5 的取值范围：`[00000000000000000000000000000000,ffffffffffffffffffffffffffffffff)`，所有读写态的主题分区会按左闭右开的原则切分整个取值范围，保证采集的每条日志都能写入到对应的分区里。

日志服务提供两种写入模式：负载均衡模式和 HashKey 模式。

负载均衡模式：每个数据包会随机写入日志主题的某个分区。

HashKey 模式：每个数据包会写入包含当前 Key 值的主题分区。

例如，一个日志主题有3个可读写分区，各个分区范围如下所示：

分区编号	状态	分区范围
1	读写	`[00000000000000000000000000000000,7fffffffffffffffffffffffffffffff)`
2	读写	`[7fffffffffffffffffffffffffffffff,a0000000000000000000000000000000)`
3	读	`[a0000000000000000000000000000000,ffffffffffffffffffffffffffffffff)`

	写	

如果写入模式是 HashKey 模式，那么 Key 值为 `2fffffffffffffffffffffffffffffff` 的日志数据会写入到分区1中，Key 值为 `9f000000000000000000000000000000` 的日志数据会写入到分区2中。

分区读写能力

每个主题分区提供一定能力的读写能力，建议业务根据实际的日志流量规划好分区数，流量超出日志主题的读写能力时应及时分裂分区，若业务流量远低于日志主题的读写能力，建议合并分区节约资源。

功能	具体项	说明
频控	写请求	单个分区写上限500qps。超限会拒绝请求，返回状态码429，提示错误SpeedQuotaExceed
	读请求	单个分区读上限200qps。超限会拒绝请求，返回状态码429，提示错误SpeedQuotaExceed
流控	写流量	单个分区写流量上限5MB/s。超限会截断数据，返回状态码429，提示错误SpeedQuotaExceed

分区状态

主题分区有两种状态：**读写态**和**只读态**。只有读写态的分区提供数据写入服务，只读态分区不允许写入数据，但在有效期内仍可被消费。创建主题分区时，所有分区状态均为读写态，但合并和分裂操作会改变状态为只读态。

合并分区

合并分区是指将两个范围相邻的两个读写态分区合并成一个分区。合并完成之后，原来的两个分区状态将会变成只读态，数据仍可被消费但是不能写入新数据。新合成的分区为读写态，新分区的范围会覆盖原来两个分区范围。

分裂分区

分裂分区是指将一个读写态的分区分裂成两个小范围分区，分裂分区时需指定一个分裂点的 MD5 值（该值必须大于起止位置，小于终止位置）。分裂成功后，原来的分区状态将会变成只读态，数据仍可被消费但是不能写入新数据。新分裂的分区为读写态，且新分区的范围会覆盖原来分区的范围。

规格与限制
日志采集
LogListener 限制说明

最近更新时间：2024-01-20 16:34:51

本文简介 Loglistener 采集数据时在文件采集、配置、资源、性能、错误处理等方面的能力与限制，以及相关使用说明。

文件采集限制

项目	能力与限制
文件编码	支持 UTF8，GBK 格式编码。 **注意：**GBK 编码格式需要 Loglistener 2.6.2及以上版本。
软链接	支持。
单条日志大小	单行日志大小限制为512KB，若日志超过512KB后，会截断只保留前512KB。多行日志按行首正则表达式划分后，单条日志最大限制为1M。
正则表达式	正则表达式类型支持 Perl 兼容正则表达式。
首次日志采集行为	Loglistener 支持全量采集/增量采集策略： 全量采集：首次安装启动 Loglistener 后，会采集所有所以符合条件的日志，包括已经没有写入的文件。 增量采集：首次安装启动 Loglistener 后，会存量文件会从最新位置开始采集。 **注意：**增量/全量采集需要 Loglistener 2.6.2及以上版本。
日志文件轮转	支持。（推荐轮转后的文件名，不要被采集通配路径覆盖到）
日志解析失败时采集行为	推荐开启解析失败上传功能，开启后将会把解析失败的日志按照单行全文格式上传至预设索引中。否则，解析失败日志将被丢弃。
文件打开行为	Loglistener 会在采集读取文件时打开，读完关闭。

Checkpoint 管理

项目	能力与限制

Checkpoint 保存位置	保存路径默认为 Loglistener 安装目录下 data 目录下。可以在 loglistener.conf 文件中自定义配置保存位置。
Checkpoint 保存策略	Loglistener 存有两份 checkpoint 元数据： 一份只记录已上传完成的位点信息，实时持久化到磁盘上。 一份同时记录了已完成位点信息与已采集但尚未完成的位点信息，周期性持久化到磁盘上。程序退出时也会优先进行持久化。

资源、性能限制

项目	能力与限制
默认 CPU 资源限制	Loglistener 默认未做 CPU 资源限制，支持通过配置文件对其使用的 CPU 资源进行限制。 目前 Loglistener 的实现架构，如未作 CPU 资源限制，其最高能够达到的 CPU 使用率为110%左右（单个业务线程最大100%， 管控线程10%左右）。
默认内存资源限制	Loglistener 默认内存阈值设置为2G，用户可以根据自己的实际情况修改此限制，建议不低于300M。
默认带宽资源限制	Loglistener 默认未做带宽资源限制，支持通过配置 loglistener.conf 文件修改对程序使用的网络带宽进行限制。
资源超限处理策略	若 Loglistener 占用相关 CPU 和内存资源超过最大限制的时间超过5分钟，则采集程序会强制自动重启。
日志压缩	采集日志默认会按照压缩发送，支持通过配置 loglistener.conf 文件修改。
监控目录数	推荐最大监控目录数5000，如超出可能会引起采集失败。
监控文件数	推荐最大监控文件数10000，如超出可能会引起采集失败。

错误处理

项目	能力与限制
网络错误处理	非需要特殊处理的异常（如日志主题删除），其它错误都会进行重试（网络异常、超时、频控、欠费等）。
超时最大重试时间	若数据持续发送失败超过1小时，则丢弃该数据。 默认行为是间隔重试，且重试间隔时间会越来越长，直至超过超时最大尝试时间。

重试次数	可通过 loglistener.conf 配置文件设置，默认不配置。此时默认会一直重试，直至超过最大重试时间，之后丢弃。 如果配置了重试次数，则按照重试次数进行重试，超过最大次数则丢弃。

文件采集规则

项目	能力与限制
日志上传策略	Loglistener 会将同一文件的日志自动聚合上传，聚合条件为：10000条日志、日志集合总大小达到1M或者日志采集时间超过3秒，任一条件满足则触发上传行为。
文件采集的处理策略	单个目标文件（采集路径所能匹配到的一个文件）只支持上传日志到一个日志主题中，不支持多个 topic 的采集路径覆盖到同一个文件。 如果想要将一个文件上传到多个日志主题中，可以通过软链接进行；对同一个目标文件创建多个软链接，不同的日志主题采集不同的软链接。
日志采集延迟	实时采集情况下，会在1分钟之内完成数据采集、传输、存储落盘，达到控制台可检索的效果。 如果日志生产量巨大，或者将采集程序使用的资源限制的较小，会有一定的采集延迟。

机器组相关规则

项目	能力与限制
机器组逻辑	目前机器组分为两类，其使用方法是相互独立的，且两种用法是不兼容的，如果混合使用，采集机器将拉取不到正确的采集配置，造成不采集的现象。 IP 机器组，机器 IP 需要在控制台上手动加入机器组，对应机器上 loglistener.conf 的 group_label 需为空。 标签机器组，控制台设置机器组标签，对应机器上 loglistener.conf 的 group_label 需要设置为相同的标签。
机器组与日志主题的关系	单个日志主题，可以绑定多个机器组。 单个机器组，可以绑定到多个日志主题上。
机器组与采集机器的关系	单个采集机器，可以加入到多个机器组中。 对于 IP 机器组，采集机器加入的机器组数量不受限制。 对于标签机器组，采集机器加入的机器组数量上限为20个。
标签机器组 label 限制	标签机器组的 label，目前限制为32字符。 标签机器组，单个机器组最多可设置20个 label。

采集路径/采集黑名单用法

项目	能力与限制
采集黑名单	此功能是用来指定采集路径下，需要忽略采集的内容，目前采集黑名单分为两类： FILE 模式：采集路径下，需要忽略采集的文件，支持通配模式。 PATH 模式：采集路径下，需要忽略采集的子目录，支持通配模式。 **注意：** FILE/PATH 模式可以同时使用。 采集黑名单是在采集路径下进行排除，因此无论是 FILE 模式，还是 PATH 模式，其指定路径要求为采集路径的子集。