ALIBABA CLOUD

阿里云

专有云企业版

日志服务

产品简介

产品版本：V3.13.0

文档版本：20220330

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.什么是日志服务

日志服务SLS（Log Service）是针对日志类数据的一站式服务，在阿里巴巴集团经历大量大数据场景锤炼而成。您无需开发就能快捷完成日志数据采集、消费以及查询分析等功能，提升运维、运营效率，建立DT时代海量日志处理能力。

日志服务提供如下功能：

- 日志采集：支持通过Logtail客户端、JS等方式实时采集Event、Binlog、TextLog等多种格式的日志数据。
- 查询分析：对于采集到的日志数据，提供实时查询、日志分析等功能，并支持根据分析结果生成可视化图表、仪表盘。
- 告警：基于查询分析功能的告警服务，当查询结果满足告警条件时，根据预先配置的告警任务发送实时告警。
- 实时消费：对于采集到服务端的日志数据提供实时消费接口。

2.产品优势

本文介绍日志服务的优势。

全托管服务

- 应用性强，简单快速即可接入服务进行使用。
- LogHub 覆盖 Kafka 全部功能，提供监控、报警等功能数据，并支持弹性伸缩（可支持PB/Day规模）。
- LogSearch/Analytics 提供快速查询、仪表盘和报警功能。
- 提供超过30种接入方式，与开源软件（Storm、Spark）无缝对接。

生态丰富

- LogHub支持30多种日志数据源，无论是嵌入式设备、网页、服务器、程序等都能轻松接入。在消费端，支持与Storm、Spark Streaming等对接。
- LogSearch/Analytics 查询分析语法完整，兼容SQL92，支持JDBC协议与对接Grafana。

实时性强

- LogHub：写入即可消费；Logtail（采集Agent）实时采集传输。
- LogSearch/Analytics：写入后3秒内即可进行查询分析，在千亿数据规模多个查询分析条件下实现数据的秒级查询结果返回。

高吞吐

日志服务具备高吞吐特性，通过日志服务软件优化，单机吞吐性能可参考如下：

- 数据写入：单机原始日志写入可达400 MB/s以上，索引日志写入可达150 MB/s以上。
- 实时消费：实时消费可达400 MB/s以上。
- QPS：单机QPS可达1万以上。

> **说明** 单机吞吐性能，根据硬件不同存在波动。

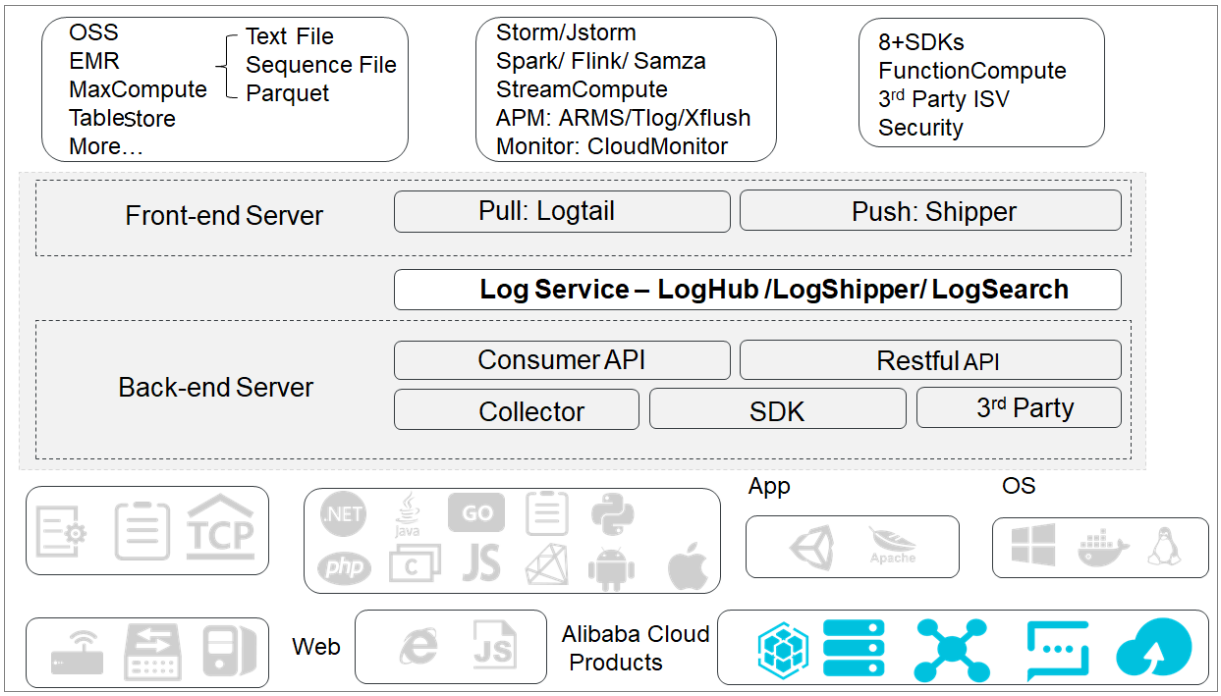
弹性

提供PB级别数据弹性伸缩能力。

3.产品架构

本文介绍日志服务的产品架构。

日志服务的架构如下图所示。



Logtail

帮助您快速收集日志的Agent。其特点如下所示：

- 基于日志文件、无侵入式的收集日志
  - 只读取文件。
  - 日志文件无侵入。
- 安全、可靠
  - 支持文件轮转不丢失数据。
  - 支持本地缓存。
  - 网络异常重试。
- 方便管理
  - Web端操作。
  - 可视化配置。
- 完善的自我保护
  - 实时监控进程CPU占用和内存消耗。
  - 限制使用上限。

前端服务器（Front-end Server）

采用LVS + Nginx构建的前端机器。其特点如下所示：

- HTTP、REST协议

- 水平扩展
  - 流量上涨时可快速提高处理能力。
  - 支持增加前端机。
- 高吞吐、低延时
  - 纯异步处理，单个请求异常不会影响其他请求。
  - 内部采用专门针对日志的Lz4压缩，提高单机处理能力，降低网络带宽。

后端服务器（Back-end Server）

后端是分布式的进程，部署在多个机器上，完成实时对Logstore数据的持久化、索引、查询。整体后端服务的特点如下所示：

- 数据高安全性：
  - 您写入的每条日志，都会被保存3份。
  - 任意磁盘损坏、机器宕机情况下，数据自动复制修复。
- 稳定服务：
  - 进程崩溃和机器宕机时，Logstore会自动迁移。
  - 自动负载均衡，确保无单机热点。
  - 严格的Quota限制，防止单个用户行为异常对其他用户产生影响。
- 水平扩展：
  - 以分区（Shard）为单位进行水平扩展。
  - 用户可以按需动态增加分区来增加吞吐量。

4.功能特性

4.1. 核心功能

本文介绍了日志服务实时采集与消费及查询与实时分析两大核心功能。

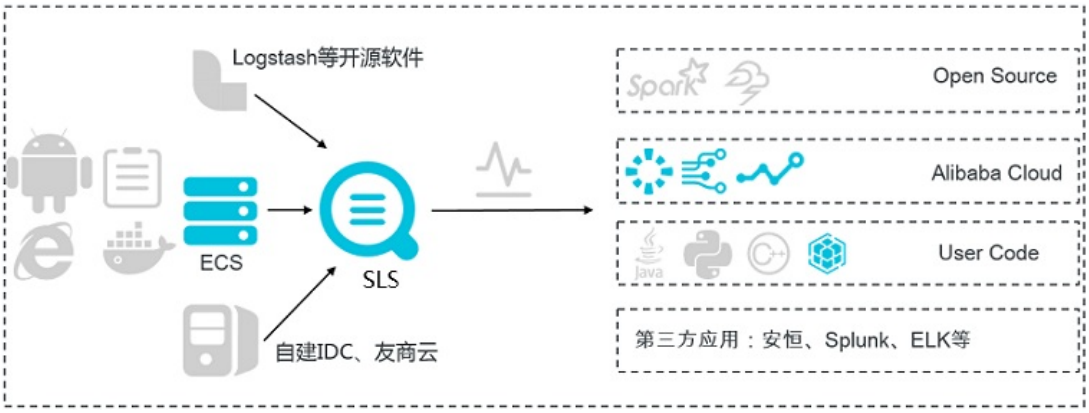
实时采集与消费（LogHub）

LogHub支持客户端Agent、SDK、网页、协议、API等多种日志无损采集方式，支持SDK、Storm Spout、Spark Client等消费方式，支持多种格式日志的实时采集和消费，协助您实现多设备、多来源的日志采集消费流程化处理。

功能：

- 通过ECS、容器、移动端、开源软件、JS等接入实时日志数据（例如Metric、Event、BinLog、TextLog、Click等）。其中通过客户端Agent及C/C++ SDK采集性能可达到单核100MB/s。
- 提供实时消费接口，与实时计算及服务对接。

实时采集与消费

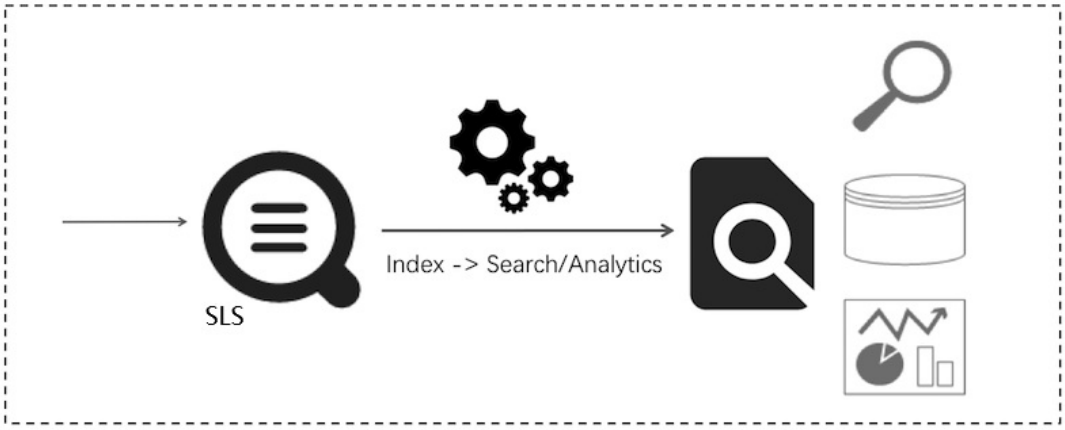


查询与实时分析（Search/Analytics）

针对采集到服务端的日志数据进行实时索引、查询分析，并根据查询分析结果建立动态的数据报表，支持多场景的日志数据可视化分析。

- 查询：关键词、模糊、上下文、范围
- 统计：SQL聚合等丰富查询手段
- 可视化：Dashboard + 报表功能
- 对接：Grafana，JDBC/SQL92

查询与实时分析



4.2. 相关功能

4.2.1. 日志

日志（Log）是系统在运行过程中变化的一种抽象，其内容为指定对象的某些操作和其操作结果按时间的有序集合。

日志服务中的日志（Log）

文件日志（LogFile）、事件（Event）、数据库日志（BinLog）、度量（Metric）数据都是日志的不同载体。在文件日志中，每个日志文件由一条或多条日志组成，每条日志描述了一次单独的系统事件，是日志服务中处理的最小数据单元。

日志服务采用半结构数据模式定义一条日志。该模式中包含主题（Topic）、时间（Time）、内容（Content）和来源（Source）四个数据域。

与此同时，日志服务对日志各字段的格式有不同要求，具体如下表所示。

数据域	含义	格式
主题（Topic）	用户自定义字段，用以标记一批日志。例如访问日志可根据不同站点进行标记。	包括空字符串在内的任意字符串，长度不超过128字节。默认情况下，该字段为空字符串。
时间（Time）	日志中的保留字段，用以表示日志产生的时间，一般由日志中的时间信息直接提取生成。	整型，Unix标准时间格式。单位为秒，表示从1970-01-01 T00:00:00Z UTC计算起的秒数。
内容（Content）	用以记录日志的具体内容。内容部分由一个或多个内容项组成，每一个内容项为一个Key-Value对。	Key为UTF-8编码字符串，包含字母、下划线和数字，且不以数字开头。长度不超过128字节。不可以使用如下关键字：`__time__`、`__source__`、`__topic__`、`__partition_time__`、`_extract_others_`，和`__extract_others__`。Value为任意字符串，长度不超过1024*1024字节。

数据域	含义	格式
来源（Source）	日志的来源地，例如产生该日志机器的IP地址。	任意字符串，长度不超过128字节。默认情况下该字段为空。

实际使用场景中，日志的格式多样。为了帮助理解，以下以一条nginx原始访问日志如何映射到日志服务日志数据模型为例说明。假设用户nginx服务器的IP地址为 `10.249.201.117` ，以下为该服务器的一条原始日志。

```
10.1.168.193 - - [01/Mar/2012:16:12:07 +0800] "GET /Send?AccessKeyId=8225105404 HTTP/1.1" 2
00 5 "-" "Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2"
```

把该条原始日志映射到日志服务日志数据模型，如下所示。

数据域	内容	说明
Topic	“”	沿用默认值，即空字符串。
Time	1330589527	日志产生的精确时间，表示从1970-01-01 T00:00:00Z UTC计算起的秒数。从原始日志中的时间戳转换而来。
Content	Key-Value对	日志具体内容。
Source	“10.249.201.117”	使用服务器IP地址作为日志源。

用户可以自己决定如何提取日志原始内容并组合成Key-Value对，例如下表：

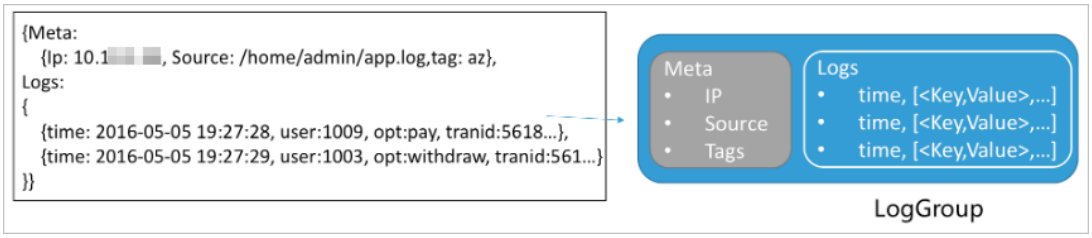
Key	Value
ip	`10.1.168.193`
method	`GET`
status	`200`
length	`5`
ref_url	`-`
browser	`Mozilla/5.0 (X11; Linux i686 on x86_64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2`

日志组

一组日志的集合，写入与读取的基本单位。

日志组的限制为：最大4096条日志，或10MB空间。

日志组



4.2.2. 项目

本文介绍了项目的概念以及项目提供的功能。

项目（Project）是日志服务中的资源管理单元，用于资源隔离和控制。您可以通过项目来管理某一个应用的所有日志及相关的日志源。它管理着用户的所有日志库（Logstore），采集日志的机器配置等信息，同时它也是用户访问日志服务资源的入口。

具体来说，项目可以提供如下功能：

- 帮助您组织、管理不同的Logstore。在实际使用中，您可能需要使用日志服务集中收集、存储不同项目、产品或者环境的日志。您可以把不同项目、产品或者环境的日志分类管理在不同的项目中，方便后续的日志消费、导出或者索引。同时，项目还是日志访问权限管理的载体。
- 提供您日志服务资源的访问入口。每创建一个项目，日志服务会为该项目分配一个独有的访问入口。该访问入口支持通过网络写入、读取及管理日志。

4.2.3. 日志库

本文为您介绍了日志库的概念以及日志库的功能。

日志库（Logstore）是日志服务中日志数据的采集、存储和查询单元。每个日志库隶属于一个项目，且每个项目可以创建多个日志库。您可以根据实际需求为某一个项目生成多个日志库，其中常见的做法是为一个应用中的每类日志创建一个独立的日志库。例如，用户有一个“big-game”游戏应用，服务器上有三种日志：操作日志（operation_log）、应用程序日志（application_log）以及访问日志（access_log），用户可以首先创建名为“big-game”的项目，然后在该项目下面为这三种日志创建三个日志库，分别用于它们的采集、存储和查询。

无论是写入或者查询日志，您都需要指定操作的 Logstore。如果您希望投递日志数据到 MaxCompute 做离线分析，其数据投递也是以 Logstore 为单元进行数据同步，即一个 Logstore 内的日志数据投递到一张 MaxCompute 的 Table。

具体来说，日志库提供如下功能：

- 采集日志，支持实时日志写入。
- 存储日志，支持实时消费。
- 建立索引，支持日志实时查询。

4.2.4. 分区

本文介绍分区的范围，分区的读写能力以及分区的状态。

Logstore读写日志必定保存在某一个分区（Shard）上。每个日志库（Logstore）分若干个分区，每个分区由MD5左闭右开区间组成，每个区间范围不会相互覆盖，并且所有的区间的范围是MD5整个取值范围。

分区范围

创建Logstore时，指定分区个数，会自动平均划分整个MD5的范围。每个分区均有范围，可用MD5方式来表示，且必定包含于以下范围中：
[00000000000000000000000000000000,ffffffffffffffffffffffffffffffff）。

分区的范围均为左闭右开区间，由以下Key组成：

- BeginKey：分区起始的Key值，分区范围中包含该Key值。
- EndKey：分区结束的Key值，分区范围中不包含该Key值。

分区的范围用于支持指定Hash Key的模式写入，以及分区的分裂和合并操作。在向分区读写数据过程中，读必须指定对应的分区，而写的过程中可以使用负载均衡模式或者指定Hash Key的模式。负载模式下，每个数据包随机写入某一个当前可用的分区中，在指定Hash Key模式下，数据写入分区范围包含指定Key值的分区。

例如，某Logstore共有4个分区，且该Logstore的MD5取值范围是[00,FF)。各个分区范围如下表所示。

分区号	范围
Shard0	[00,40)
Shard1	[40,80)
Shard2	[80,C0)
Shard3	[C0,FF)

当写入日志时，通过指定Hash Key模式指定一个MD5的Key值是5F，日志数据会写入包含5F的Shard1分区上；如果指定一个MD5的Key值是8C，日志数据会写入包含8C的Shard2分区上。

分区的读写能力

每个分区可提供一定的服务能力，建议您根据实际数据流量规划分区个数，流量超出读写能力时，及时分裂分区以增加分区个数，从而达到更大的读写能力；如您的流量远远达不到分区的最大读写能力时，建议您合并分区以减少分区个数，从而节约分区租赁费用。

> **说明**
> - 当写入的API持续报告403或者500错误时，通过监控查看流量和状态码判断是否需要增加分区。
> - 对超过分区服务能力的读写，系统会尽可能服务，但不保证服务质量。

分区的状态

分区的状态包括：

- readwrite：可以读写
- readonly：只读数据

创建分区时，所有分区状态均为readwrite状态，分裂或合并操作会改变分区状态为readonly，并生成新的readwrite分区。分区状态不影响其数据读取的性能，同时，readwrite分区保持正常的数据写入性能，readonly状态分区不提供数据写入服务。

在分裂分区时，需要指定一个处于readwrite状态的ShardId和一个MD5。MD5要求必须大于分区的BeginKey并且小于EndKey。分裂操作可以从一个分区中分裂出另外两个分区，即分裂后分区数量增加2。在分裂完成后，被指定分裂的原分区状态由readwrite变为readonly，数据仍然可以被消费，但不可写入新数据。两个新生成的分区状态为readwrite，排列在原有分区之后，且两个分区的MD5范围覆盖了原来分区的范围。

在合并操作时，必须指定一个处于readwrite状态的分区，指定的分区不能是最后一个readwrite分区。服务端会自动找到所指定分区的右侧相邻分区，并将两个分区范围合并。在合并完成后，所指定的分区和其右侧相邻分区变成只读（readonly）状态，数据仍然可以被消费，但不能写入新数据。同时新生成一个readwrite 状态的分区，新分区的MD5范围覆盖了原来两个分区的范围。

4.2.5. 日志主题

本文为您介绍了日志主题的概念。

一个日志库内的日志可以通过日志主题（Topic）来划分。您可以在写入时指定日志主题，并在查询时指定查询的日志主题。例如，一个平台用户可以使用用户编号作为日志主题写入日志。这样在查询时可利用日志主题让不同用户仅看到自己的日志。如果不需要划分一个日志库内的日志，让所有日志使用相同的日志主题即可。

> **说明** 空字符串是一个有效的日志主题（Topic），且无论是写入还是查询日志时，默认的日志主题都是空字符串。所以，如果不需要使用日志主题，最简单的方式就是在写入和查询日志时都使用默认日志主题，即空字符串。

5.应用场景

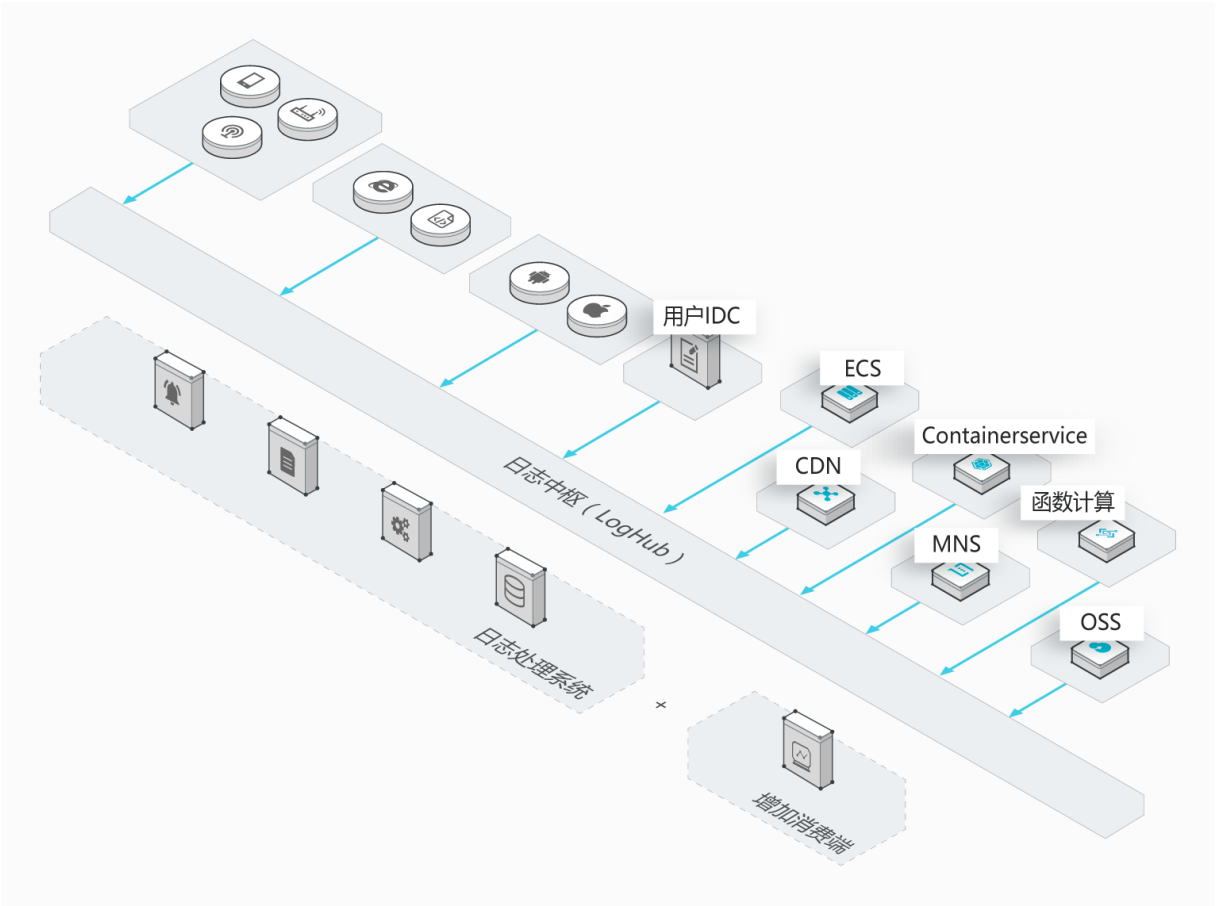
本文介绍日志服务的典型应用场景，包括数据采集、实时计算、数仓与离线分析、产品运营与分析、运维与管理。

数据采集与消费

通过日志服务LogHub功能，可以大规模低成本接入各种实时日志数据（包括Metric、Event、BinLog、TextLog、Click等）。

方案优势：

- 使用便捷：提供30多种实时数据采集方式，让您快速搭建平台、减轻运维负担。
- 弹性伸缩：无论是流量高峰还是业务增长都能轻松应对。



数据清洗与实时计算 （ETL/Stream Processing）

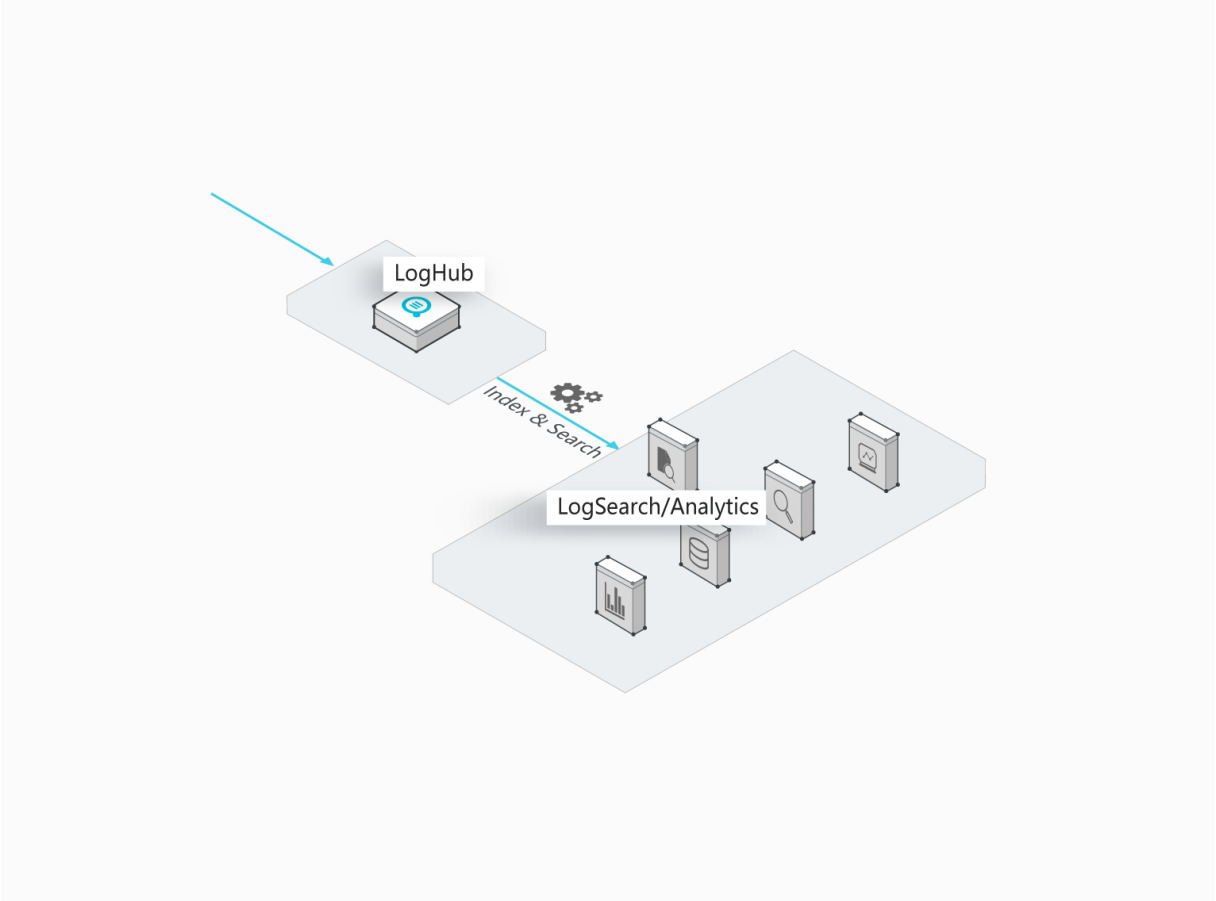
日志中枢（LogHub）支持与各种实时计算及服务对接，并提供完整的进度监控、报警等功能，并可以根据SDK/API实现自定义消费。

- 操作便捷：提供多种语言的SDK以及编程框架，与各流计算引擎无缝对接。
- 功能完善：支持报警机制，并提供丰富的监控数据。
- 弹性伸缩：PB级弹性能力，0延迟。

日志实时查询与分析

实时查询分析（LogAnalytics）可以实时索引LogHub中数据，提供关键词、模糊、上下文、范围、SQL聚合等丰富查询手段。

- 实时性强：写入后即可查询。
- 海量低成本：支持PB/Day索引能力，成本为自建方案的15%。
- 分析能力强：支持多种查询手段，及SQL进行聚合分析，并提供可视化及报警功能。


LogHub
Index & Search
LogSearch/Analytics

6.使用限制

本文介绍日志服务的使用限制。

资源限制

分类	限制说明	备注
Project	每个部门最多可创建10个Project。	如您有更大的使用需求，请提工单申请。
Logstore	一个Project中最多可创建100个Logstore。	如您有更大的使用需求，请提工单申请。
Shard	• 一个Logstore最多可创建10个Shard。但可以通过**分裂**操作来增加Shard。 • 一个Project中最多可创建100个Shard。	如您有更大的使用需求，请提工单申请。
仪表盘	• 每个Project最多可创建5个仪表盘。 • 每个仪表盘最多可包含十张分析图表。	如您有更大的使用需求，请提工单申请。
快速查询	每个Project最多可创建10个快速查询。	如您有更大的使用需求，请提工单申请。
Logtail配置	每个Project最多可创建100个Logtail配置。	如您有更大的使用需求，请提工单申请。
协同消费组	每个Project最多可创建10个协同消费组。	如您有更大的使用需求，请提工单申请。
机器组	每个Project最多可创建100个机器组。	如您有更大的使用需求，请提工单申请。
日志保存时间	采集到服务端的日志最多可保存365天。	如您有更大的使用需求，请提工单申请。

7.基本概念

本文介绍了日志服务涉及的一些基本概念。

日志

日志（Log）是系统在运行过程中变化的一种抽象，其内容为指定对象的某些操作和其操作结果按时间的有序集合。文件日志（LogFile）、事件（Event）、数据库日志（BinLog）、度量（Metric）数据都是日志的不同载体。在文件日志中，每个日志文件由一条或多条日志组成，每条日志描述了一次单独的系统事件，是日志服务中处理的最小数据单元。

日志组

一组日志的集合，写入与读取的基本单位。

日志主题

一个日志库内的日志可以通过日志主题（Topic）来划分。用户可以在写入时指定日志主题，并在查询时指定查询的日志主题。

项目

项目（Project）是日志服务中的资源管理单元，用于资源隔离和控制。您可以通过项目来管理某一个应用的所有日志及相关的日志源。它管理着您的所有日志库（Logstore），采集日志的机器配置等信息，同时它也是您访问日志服务资源的入口。

日志库

日志库是日志服务中日志数据的收集、存储和查询单元。每个日志库隶属于一个项目，且每个项目可以创建多个日志库。

分区

每个日志库分若干个分区（Shard），每个分区由MD5左闭右开区间组成，每个区间范围不会相互覆盖，并且所有的区间的范围是MD5整个取值范围。