ALIBABA CLOUD

阿里云

日志服务

产品简介

文档版本：20220711

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是云原生观测与分析平台，为Log、Metric、Trace等数据提供大规模、低成本、实时的平台化服务。日志服务一站式提供数据采集、加工、查询与分析、可视化、告警、消费与投递等功能，全面提升您在研发、运维、运营、安全等场景的数字化能力。

学习路径

日志服务学习路径图为您推荐热门功能的操作指引文档，帮助您快速了解日志服务产品。视频与文档结合，全方位提升您的产品使用及文档阅读体验。

基本概念

在使用日志服务前，您需要了解以下基本概念。

术语	说明
项目（Project）	项目是日志服务的资源管理单元，是进行多用户隔离与访问控制的主要边界。更多信息，请参见项目（Project）。
日志库（Logstore）	日志库是日志服务中日志数据的采集、存储和查询单元。更多信息，请参见日志库（Logstore）。
时序库（MetricStore）	时序库是日志服务中时序数据的采集、存储和查询单元。更多信息，请参见时序库（MetricStore）。
日志（Log）	日志是系统运行过程中变化的一种抽象数据，其内容为指定对象的操作和其操作结果按时间的有序集合。更多信息，请参见日志（Log）。
日志组（LogGroup）	日志组是一组日志的集合，是写入与读取日志的基本单位。一个日志组中的日志包含相同Meta信息（IP地址、Source等信息）。更多信息，请参见日志组（LogGroup）。
时序数据（Metric）	时序数据是指时间序列数据。更多信息，请参见时序数据（Metric）。
链路数据（Trace）	链路数据代表一个事务或者流程在（分布式）系统中的执行过程。更多信息，请参见链路数据（Trace）。
分区（Shard）	分区用于控制Logstore的读写能力，数据必定保存在某一个Shard中。每个Shard均有范围，为MD5左闭右开区间。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[00000000000000000000000000000000,ffffffffffffffffffffffffffffffff）。更多信息，请参见分区（Shard）。
日志主题（Topic）	日志主题是日志服务的基础管理单元。您可在采集日志时指定日志主题，日志服务将通过日志主题划分日志。更多信息，请参见日志主题（Topic）。
服务入口（Endpoint）	日志服务的服务入口是访问一个Project及其内部数据的URL。访问不同地域的Project时，所需的服务入口不同。通过内网和外网访问同一地域的Project时，所需的服务入口也是不同的。更多信息，请参见服务入口。

术语	说明
访问密钥（AccessKey）	访问密钥指的是访问身份验证中用到的AccessKey ID和AccessKey Secret。日志服务通过使用AccessKey ID和AccessKey Secret对称加密的方法来验证某个请求的发送者身份。AccessKey ID用于标识用户；AccessKey Secret是用户用于加密签名字符串和日志服务用来验证签名字符串的密钥，必须保密。更多信息，请参见访问密钥。
地域（Region）	地域是日志服务的数据中心所在物理位置。您可以在创建Project时指定地域，一旦指定之后就不允许更改。更多信息，请参见开服地域。

功能概览

日志服务包含以下功能模块，覆盖云原生观测与分析的多种业务场景。

使用方式

您可以通过以下任意一种方式使用日志服务。

方式	说明
控制台	日志服务提供Web服务页面管理您的日志服务资源。更多信息，请参见日志服务控制台。
SDK	日志服务提供各种语言的SDK开发包，方便您快速进行二次开发。更多信息，请参见SDK概述。
API	日志服务提供API管理您的日志服务资源。该方式需要您手动签名验证。更多信息，请参见API概述。 说明 推荐您使用SDK，以免除手动签名验证环节。
CLI	日志服务提供CLI管理您的日志服务资源。更多信息，请参见CLI概述。

产品定价

日志服务支持按量付费，即按照您的实际使用量收费。相较于自建ELK，使用日志服务，总成本预计可以下降50%。关于日志服务的计量项和计费项，请参见计费项。

立即开通

单击下方按钮可立即前往日志服务开通页面。

扩展阅读

SLS不仅能进行日志查询与分析，还能观测告警、数据库审计、数据加工与处理、实现商业分析、用户画像分析……看看达人们是怎么玩的吧。

直达实战派

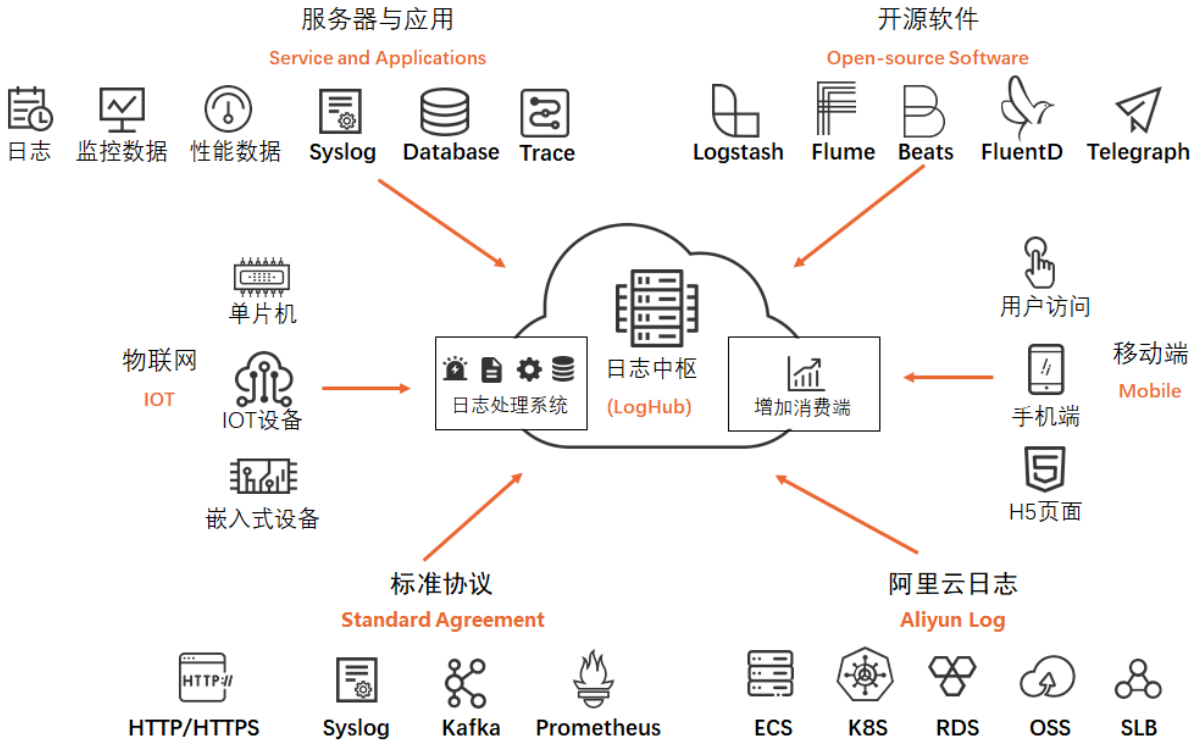
2.功能特性

本文介绍日志服务主要的功能。

数据采集

日志服务提供50多种数据接入方案。具体说明如下：

- 支持采集服务器与应用相关的日志、时序数据和链路数据。
- 支持采集物联网设备日志。
- 支持采集阿里云产品日志。
- 支持采集移动端数据。
- 支持采集Logstash、Flume、Beats、FluentD、Telegraph等开源软件中的数据。
- 支持通过HTTP、HTTPS、Syslog、Kafka、Prometheus等标准协议接入数据。



更多信息，请参见数据采集概述。

查询与分析

日志服务支持实时查询与分析数据。具体说明如下：

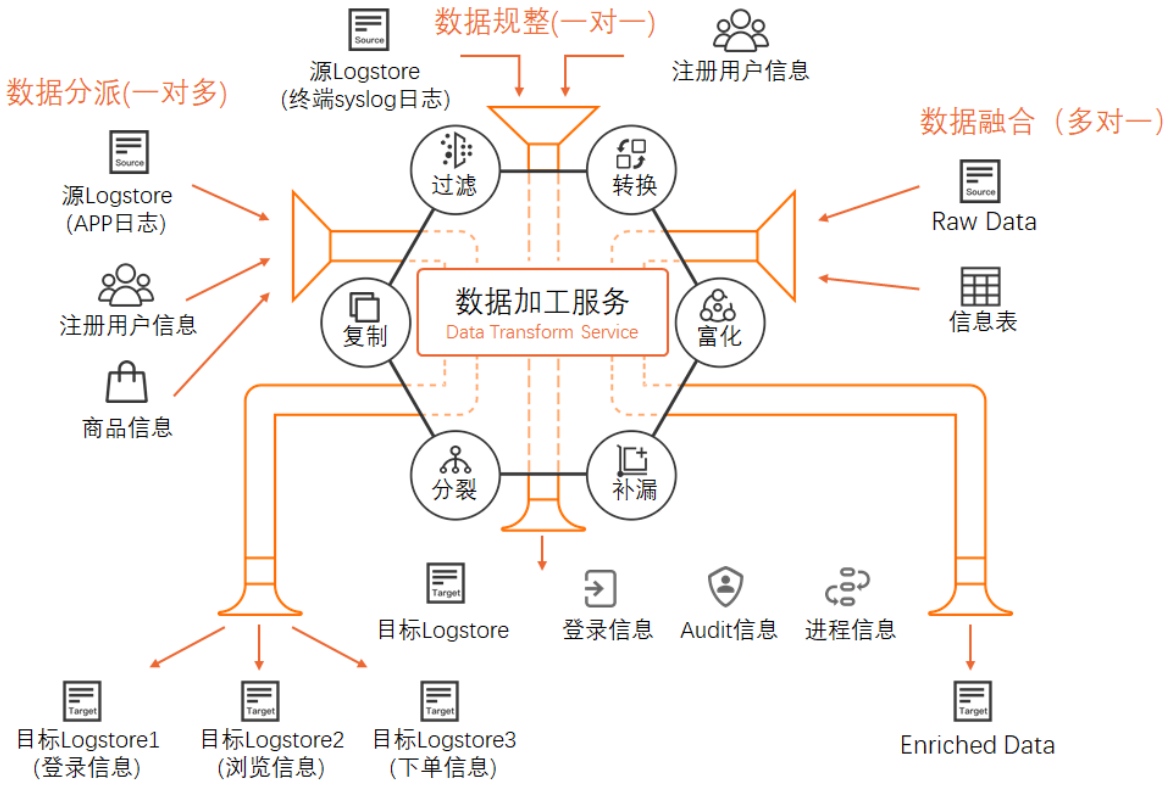
- 支持精确查询、模糊查询、全文查询、字段查询。
- 支持上下文查询、日志聚类、LiveTail、重建索引等功能。
- 支持标准的SQL 92语法。
- 提供SQL独享实例。

更多信息，请参见查询概述和分析简介。

数据加工

日志服务提供数据加工功能，用于数据的规整、富化、流转、脱敏和过滤。具体说明如下：

- 数据规整：针对混乱格式的日志进行字段提取、格式转换，获取结构化数据以支持后续的流处理、数据仓库计算。
- 数据富化：对日志（例如订单日志）和维表（例如用户信息表）进行字段连接（JOIN），为日志添加更多维度的信息，用于数据分析。
- 数据流转：通过全球加速功能将海外地域的日志传输到中心地域，实现全球日志集中化管理。
- 数据脱敏：对数据中包含的密码、手机号、地址等敏感信息进行脱敏。
- 数据过滤：过滤出关键服务的日志，用于重点分析。

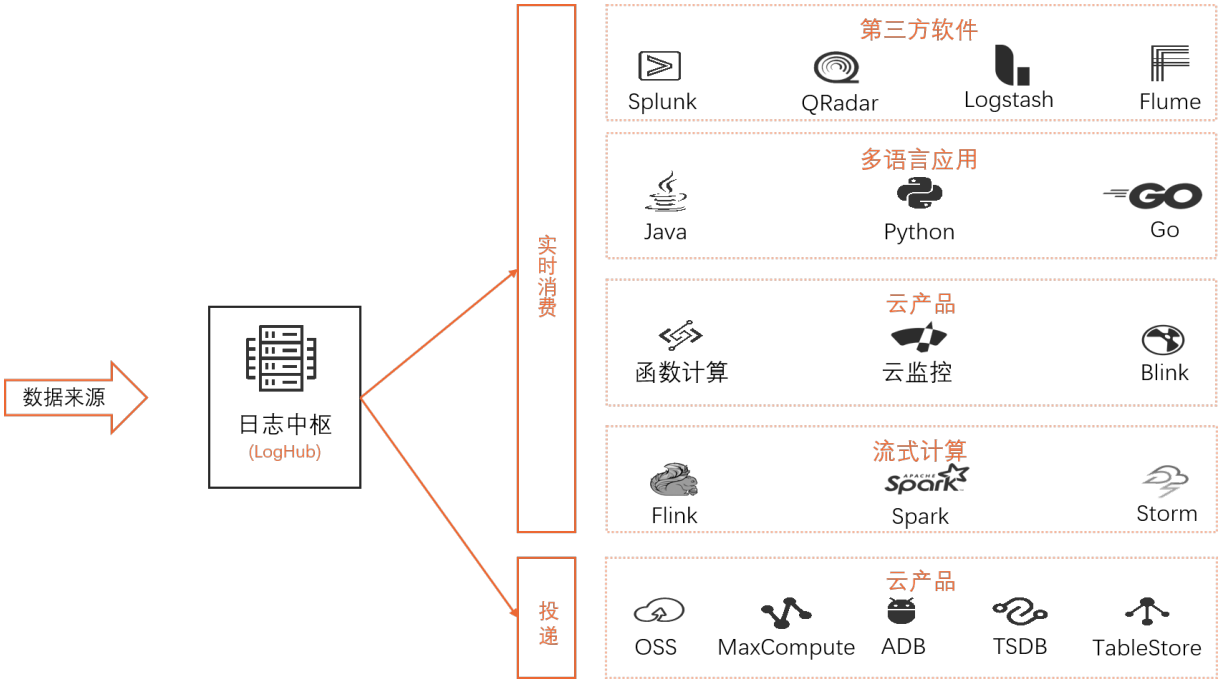


更多信息，请参见数据加工概述。

消费与投递

日志服务提供消费与投递功能，支持通过SDK、API实时消费数据；支持通过控制台将数据实时投递至OSS、MaxCompute等阿里云产品中。具体说明如下：

- 支持通过Splunk、QRadar、Logstash、Flume等第三方软件消费数据。
- 支持通过Java、Python、GO等语言消费数据。
- 支持通过函数计算、实时计算、云监控等阿里云产品消费数据。
- 支持通过Flink、Spark、Storm等流式计算平台消费数据。
- 支持将数据投递到OSS、MaxCompute、AnalyticDB、TSDB、TableStore等阿里云产品。



更多信息，请参见实时消费概述和数据投递概述。

可视化

日志服务支持可视化展示查询和分析结果。具体说明如下：

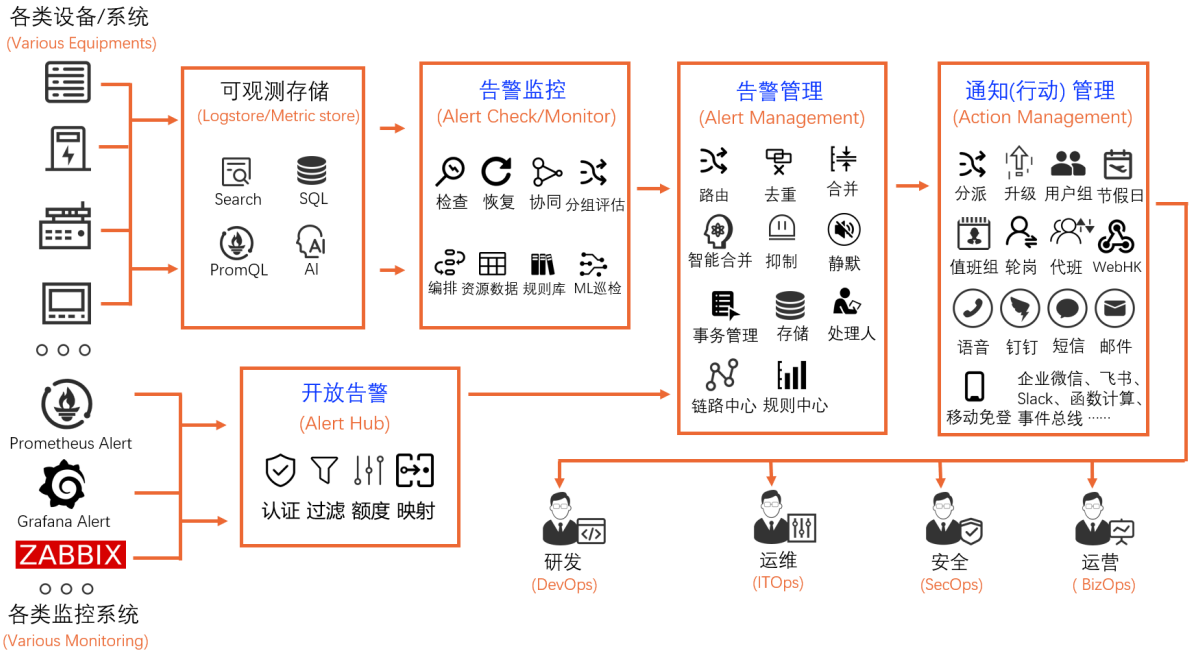
- 仪表盘内置图表：日志服务为您提供表格、线图、柱状图等多种统计图表，您可以根据分析需求选用合适的图表类型展示查询和分析结果，并将结果保存到仪表盘中。
- 第三方可视化工具：日志服务支持直接对接Grafana、DataV等第三方图表。

更多信息，请参见可视化概述。

告警

日志服务提供一站式的告警监控、降噪、事务管理、通知分派的智能运维平台。具体说明如下：

- 告警监控：支持通过告警监控规则定期检查评估查询和分析结果，触发告警或恢复通知，发送给告警管理系统。
- 告警管理：支持通过告警策略对所接收到的告警进行路由分派、抑制、去重、静默、合并等操作，然后发送给通知（行动）管理系统。
- 通知（行动）管理：支持通过行动策略将告警动态分派给特定的通知渠道，再通知给目标用户、用户组或值班组。
- 开放告警：支持通过Webhook方式接收外部监控系统中的告警消息（例如Grafana告警、Prometheus告警），并完成告警管理、告警通知等操作。

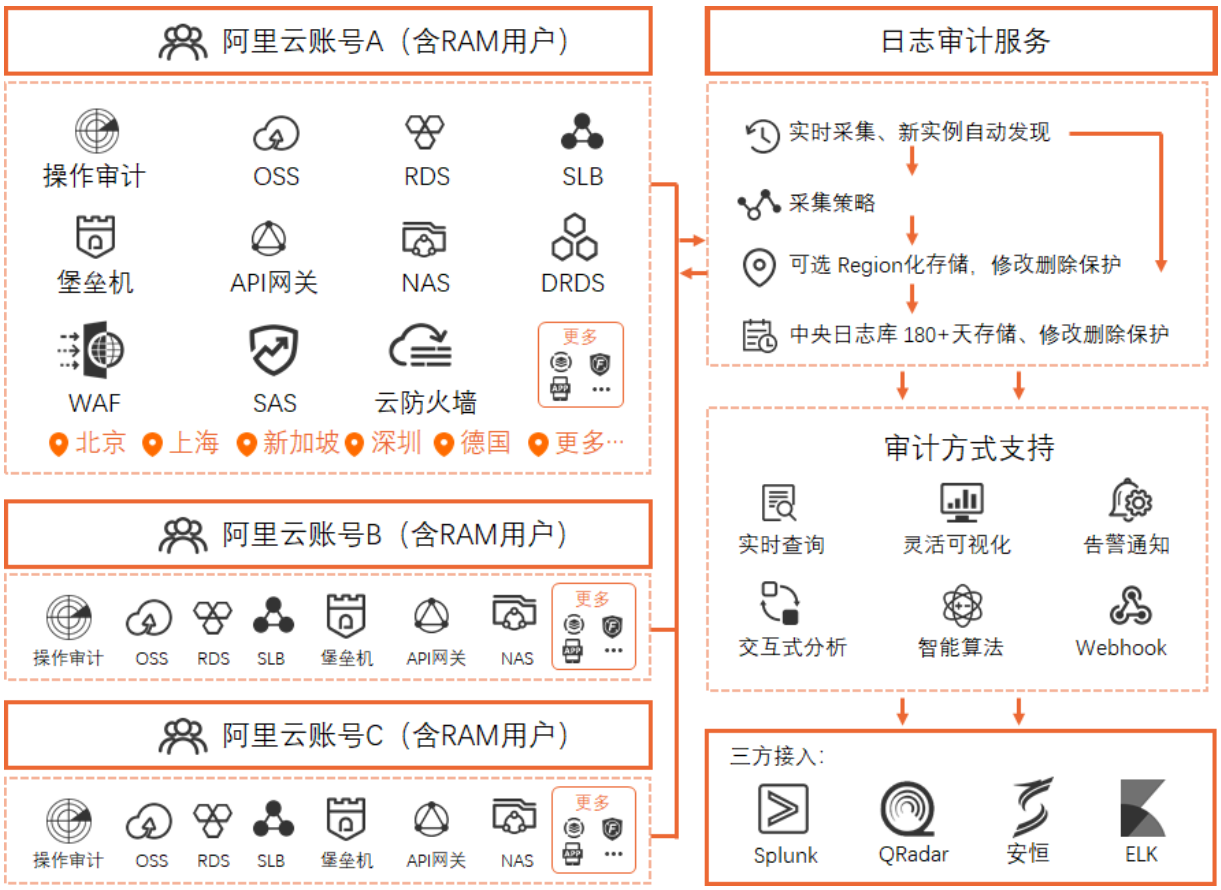


更多信息，请参见告警概述。

日志审计

日志审计服务在继承现有日志服务所有功能基础上还支持自动化采集、对接其他生态产品等功能。具体说明如下：

- 支持实时自动化、中心化采集多账号下的云产品日志并进行审计。
- 覆盖基础（ActionTrail、容器服务Kubernetes版）、存储（OSS、NAS）、网络（SLB、API网关）、数据库（关系型数据库RDS、云原生分布式数据库PolarDB-X、PolarDB MySQL云原生数据库）、安全（WAF、DDoS防护、云防火墙、云安全中心）等云产品。
- 支持自由对接其他生态产品或自有SOC中心。
- 内置百种告警规则，支持一键式开启，覆盖账户安全、权限管理、存储、主机、数据库、网络、日志等各个方面的合规监控。



更多信息，请参见日志审计概述。

3.产品架构

本文介绍日志服务的架构。

日志服务的架构如下图所示：

- 数据来源

  日志服务支持采集开源软件、服务器与应用、阿里云产品、标准协议、移动端、物联网等多种来源的数据。

- 日志服务
  - 数据类型

    日志服务为Log、Metric、Trace等数据提供大规模、低成本、实时的平台化服务。更多信息，请参见日志（Log）、时序数据（Metric）、链路数据（Trace）。

- 功能特性
  - 数据采集：日志服务支持通过Logtail、SDK、协议等多种方式采集数据。更多信息，请参见数据采集概述。
  - 数据加工：日志服务提供可托管、可扩展、高可用的数据加工服务。数据加工服务可用于数据的规整、富化、流转、脱敏和过滤。更多信息，请参见数据加工概述。
  - 查询与分析：日志服务支持PB级数据实时查询与分析，提供10多种查询运算符、10多种机器学习函数、100多个SQL函数，并支持Scheduled SQL和SQL独享版。更多信息，请参见查询概述和分析简介。
  - 可视化：日志服务支持可视化展示查询与分析结果，并支持基于统计图表自定义仪表盘。更多信息，请参见可视化概述。
  - 告警：日志服务提供一站式告警功能，包括告警监控、告警管理、通知（行动）管理等，适用于开发运维、IT运维、智能运维、安全运维、商务运维等多个场景。更多信息，请参见什么是日志服务告警。
  - 消费与投递：日志服务支持数据实时消费，适用于Storm消费、Flume消费、Flink消费等场景；支持数据实时投递，适用于将数据投递至OSS、TSDB等云产品。更多信息，请参见投递概述和消费概述。
  - 日志审计：日志服务支持实时自动化、中心化采集多账号下的云产品日志并进行审计。更多信息，请参见日志审计概述。
- 使用方式

  日志服务支持控制台、API、SDK、CLI等多种使用方式。
- 应用场景

  日志服务可服务于运营、运维、研发、安全等多种场景。更多信息，请参见应用场景。

- 数据目标

  日志服务支持通过消费或投递的方式将数据导出至云产品或第三方软件。

4.产品优势

本文介绍日志服务的优势。

统一接入

支持多种来源的多种类型数据接入。

智能

提供完整AIOps能力，支持智能异常检测与根因分析能力。

高效

提供百亿级数据实时采集和查询与分析能力。

一站式

提供一站式数据功能，包括数据采集、加工、查询与分析、可视化、告警等。

弹性

提供PB级别数据弹性伸缩能力。

低成本

支持按量付费。您仅需为实际用量付费，总拥有成本（TCO）降低50%以上。

5.应用场景

日志服务的典型应用场景包括：数据采集与消费、数据清洗与流计算 （ETL/Stream Processing）、数据仓库对接（Data Warehouse）、日志实时查询与分析。

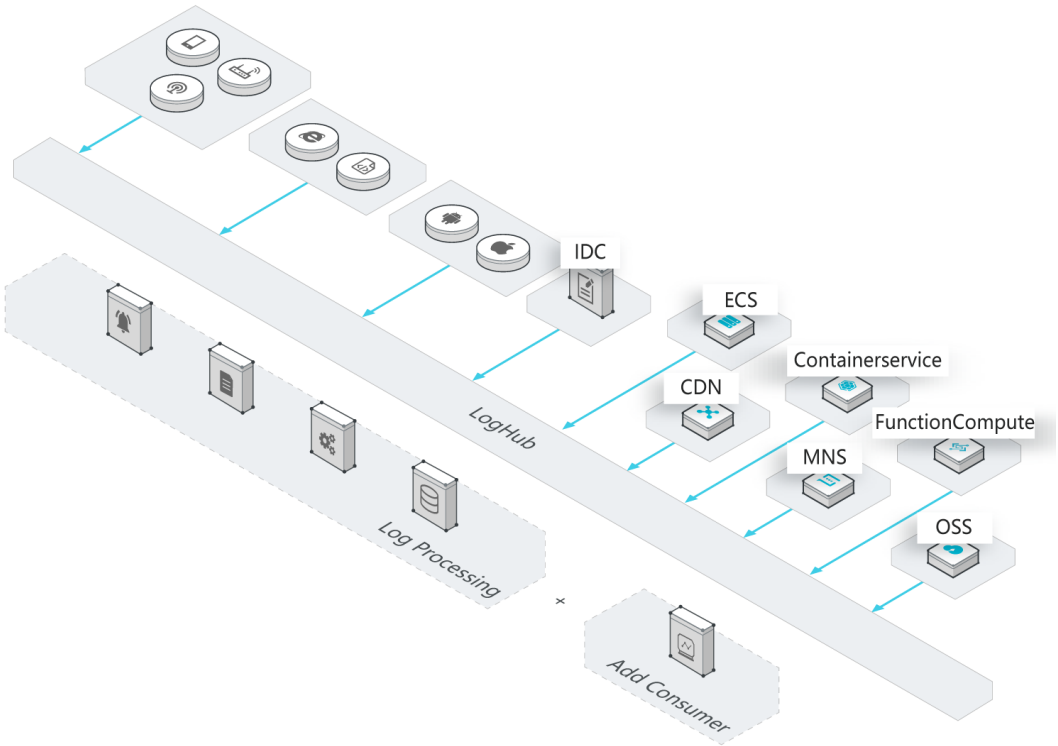
数据采集与消费

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50+实时数据采集方式，让您快速搭建平台；强大配置管理能力，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。

数据采集与消费

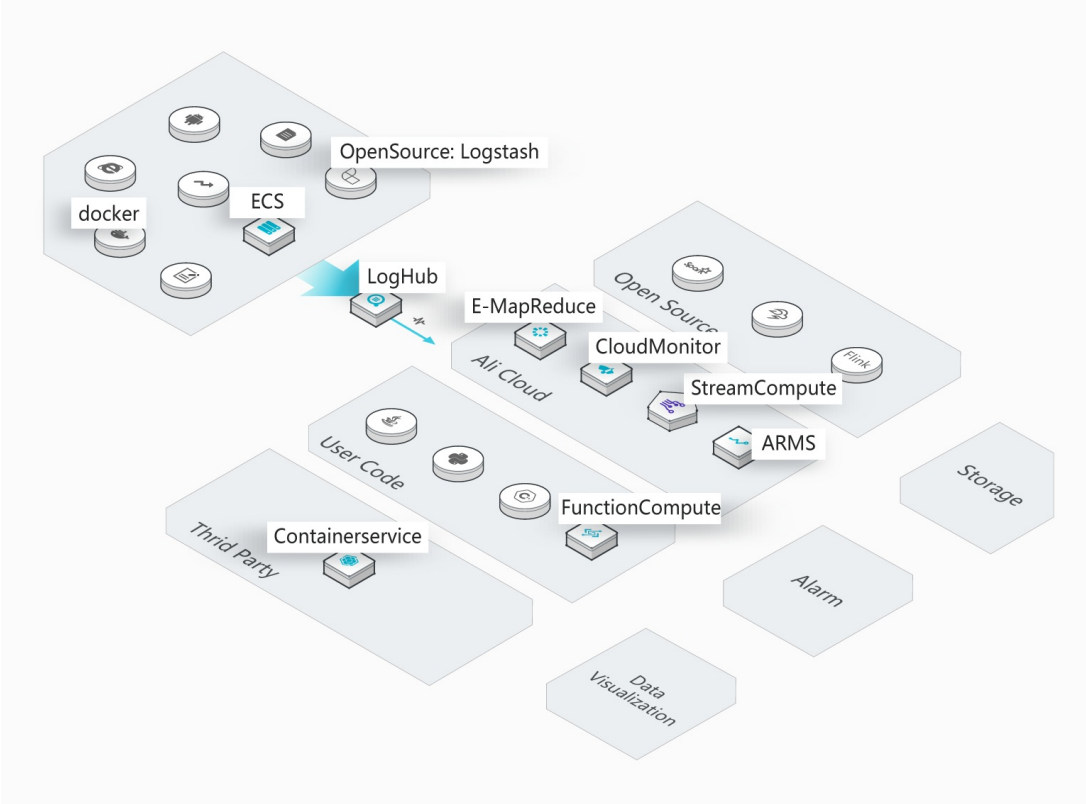


数据清洗与流计算 （ETL/Stream Processing）

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：提供丰富SDK以及编程框架，与各流计算引擎无缝对接。
- 监控报警：提供丰富监控数据，以及延迟报警机制。
- 弹性伸缩：PB级弹性能力，0延迟。

数据清洗与流计算

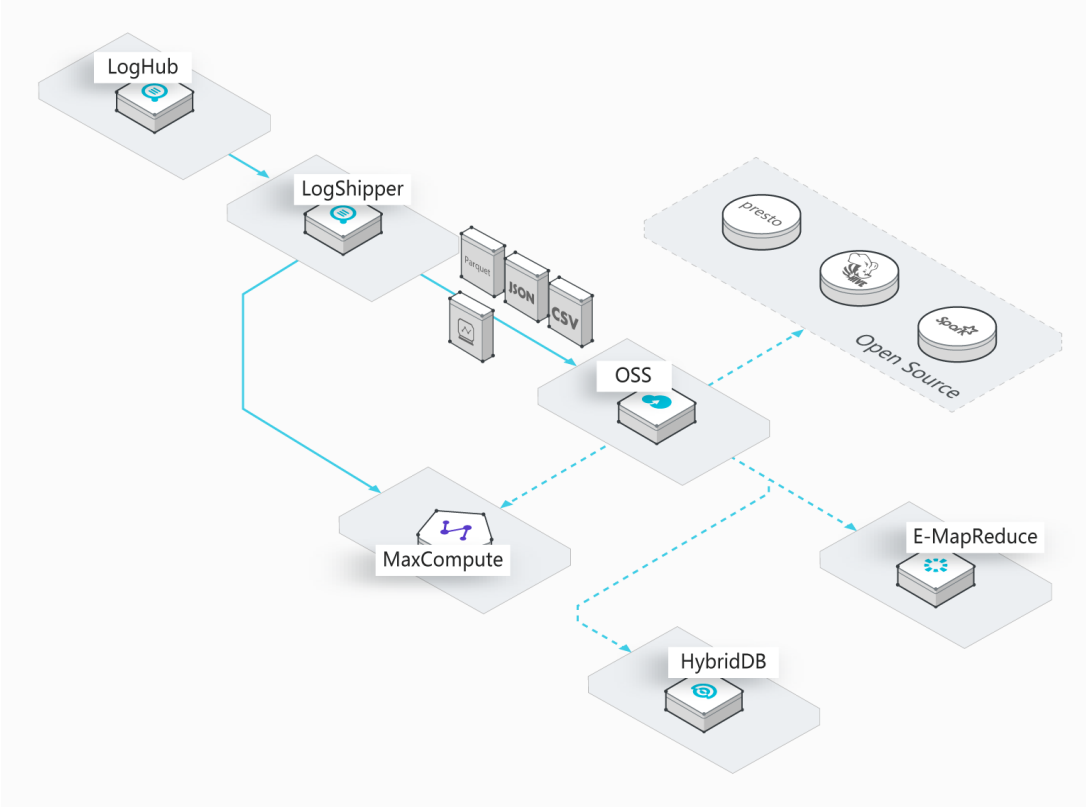


数据仓库对接（Data Warehouse）

日志投递（LogShipper）功能可以将日志中枢（LogHub）中数据投递至存储类服务，过程支持压缩、自定义Partition、以及行列等各种存储格式。

- 海量数据：对数据量不设上限。
- 种类丰富：支持行、列、TextFile等各种存储格式。
- 配置灵活：支持用户自定义Partition等配置。

数据仓库对接

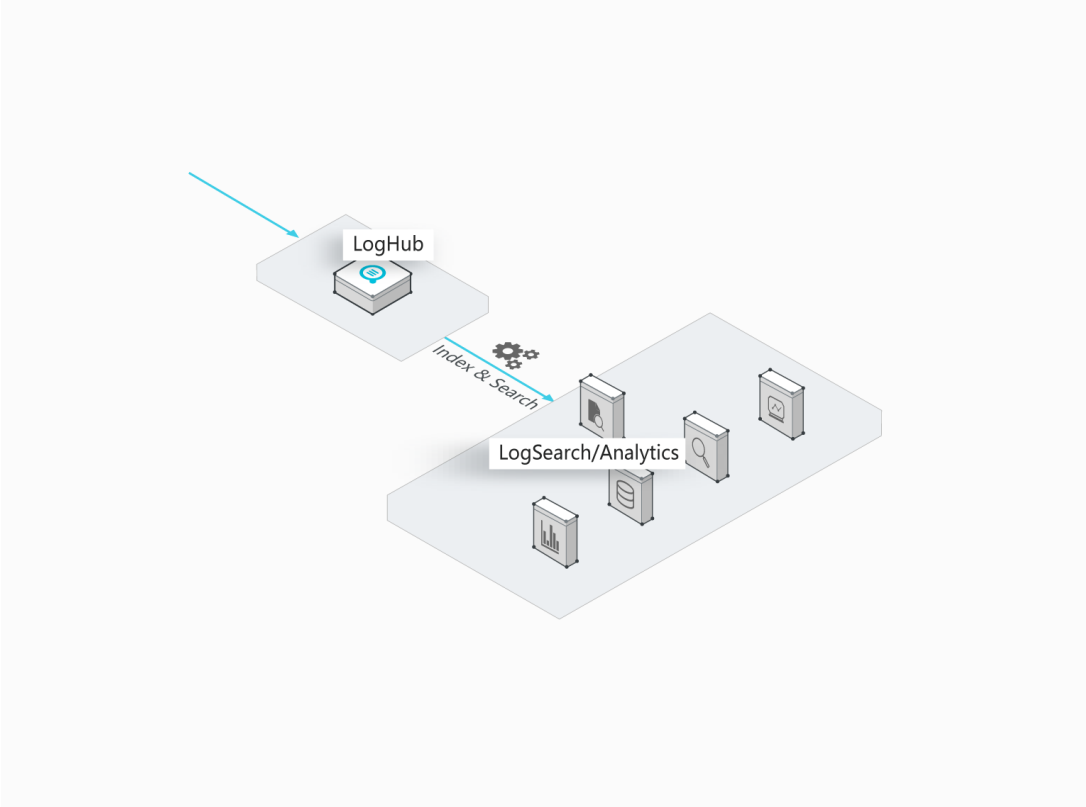


日志实时查询与分析

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。

日志实时查询与分析


LogHub
Index & Search
LogSearch/Analytics

6.基本概念

6.1. 术语表

本文介绍日志服务所涉及的术语。

基础资源

术语	说明
项目（Project）	项目是日志服务的资源管理单元，是进行多用户隔离与访问控制的主要边界。更多信息，请参见项目（Project）。
日志库（Logstore）	日志库是日志服务中日志数据的采集、存储和查询单元。更多信息，请参见日志库（Logstore）。
时序库（MetricStore）	时序库是日志服务中时序数据的采集、存储和查询单元。更多信息，请参见时序库（MetricStore）。
日志（Log）	日志是系统运行过程中变化的一种抽象数据，其内容为指定对象的操作和其操作结果按时间的有序集合。更多信息，请参见日志（Log）。
日志组（LogGroup）	日志组是一组日志的集合，是写入与读取日志的基本单位。一个日志组中的日志包含相同Meta信息（IP地址、Source等信息）。更多信息，请参见日志组（LogGroup）。
时序数据（Metric）	时序数据是指时间序列数据。更多信息，请参见时序数据（Metric）。
链路数据（Trace）	链路数据代表一个事务或者流程在（分布式）系统中的执行过程。更多信息，请参见链路数据（Trace）。
分区（Shard）	分区用于控制Logstore的读写能力，数据必定保存在某一个Shard中。每个Shard均有范围，为MD5左闭右开区间。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[00000000000000000000000000000000,ffffffffffffffffffffffffffffffff）。更多信息，请参见分区（Shard）。
日志主题（Topic）	日志主题是日志服务的基础管理单元。您可在采集日志时指定日志主题，日志服务将通过日志主题划分日志。更多信息，请参见日志主题（Topic）。
服务入口（Endpoint）	日志服务的服务入口是访问一个Project及其内部数据的URL。访问不同地域的Project时，所需的服务入口不同。通过内网和外网访问同一地域的Project时，所需的服务入口也是不同的。更多信息，请参见服务入口。
访问密钥（AccessKey）	访问密钥指的是访问身份验证中用到的AccessKey ID和AccessKey Secret。日志服务通过使用AccessKey ID和AccessKey Secret对称加密的方法来验证某个请求的发送者身份。AccessKey ID用于标识用户；AccessKey Secret是用户用于加密签名字符串和日志服务用来验证签名字符串的密钥，必须保密。更多信息，请参见访问密钥。
地域（Region）	地域是日志服务的数据中心所在物理位置。您可以在创建Project时指定地域，一旦指定之后就不允许更改。更多信息，请参见开服地域。

数据采集

术语	说明
Logtail	Logtail是日志服务提供的日志采集工具。更多信息，请参见Logtail采集概述。
Logtail配置	Logtail配置是Logtail进行日志采集的策略集合，包括日志文件的位置、采集方式等。更多信息，请参见Logtail配置。
机器组	机器组是包含多台服务器的虚拟分组。日志服务通过机器组的方式管理所有需要通过Logtail采集日志的服务器。更多信息，请参见机器组。

查询与分析

术语	说明
查询	通过查询语句指定过滤规则，返回符合条件的日志。更多信息，请参见查询概述。
分析	在查询的基础上，使用SQL函数完成统计、分析，返回分析结果。 • 分析日志时，支持标准的SQL92语法。更多信息，请参见分析简介。 • 分析时序数据时，支持标准的SQL92语法和PromQL语法。更多信息，请参见时序数据查询分析简介。
查询和分析语句	查询和分析语句格式为 `查询语句|分析语句` 。查询语句可单独使用，分析语句必须与查询语句一起使用。即分析功能是基于查询结果或全量数据进行的。更多信息，请参见查询和分析。
索引	索引是一种存储结构，用于对数据中的一列或多列进行排序。您只有配置索引后，才能进行查询操作。日志服务提供如下两种索引类型： • 全文索引：日志服务根据您设置的分词符将整条日志拆分成多个词并构建索引。在查询时，字段名称（KEY）和字段值（Value）都是普通文本。 • 字段索引：配置字段索引后，您可以指定字段名称和字段值（Key:Value）进行查询，缩小查询范围。 更多信息，请参见配置索引。
SQL普通版	SQL普通版为免费资源，用于SQL分析。相对比SQL独享版，存在更多的资源限制。
SQL独享版	SQL独享版是日志服务提供的计费资源，用于SQL分析。当您对大规模（百亿到千亿级）数据有分析需求时，可使用SQL独享版。更多信息，请参见开启SQL独享版。

数据加工

术语	说明
DSL（Domain Specific Language）	DSL是日志服务数据加工使用的一种Python兼容的脚本语言。更多信息，请参见语言简介。

术语	说明
加工规则	数据加工脚本，日志服务DSL编排的逻辑代码的集合。更多信息，请参见语法简介。

消费与投递

术语	说明
消费组（ConsumerGroup）	日志服务支持通过消费组消费数据。一个消费组由多个消费者构成，同一个消费组中的消费者共同消费一个Logstore中的日志数据，消费者之间不会重复消费数据。更多信息，请参见通过消费组消费日志数据。

告警

术语	说明
告警	独立表达时，代表一个告警事件（Alert event）。例如告警监控规则触发一个或多个告警后，通过告警管理系统传递给通知管理系统。 当告警与其他词组合时，代表告警功能对应的子系统、功能、实体、模块等。例如告警监控系统、告警监控规则等。 更多信息，请参见什么是日志服务告警。
告警监控	告警子系统，负责产生告警。告警监控系统由告警监控规则和资源数据等组成。 通过告警监控规则定期检查评估，根据监控编排逻辑评估查询和分析结果，触发告警或恢复通知，发送给告警管理系统。
告警管理	告警子系统，负责管理告警降噪和告警状态。告警管理系统由告警策略、告警事务管理和告警态势大盘等组成。 告警管理系统通过告警策略对所接收到的告警进行路由分派、抑制、去重、静默、合并等操作，然后发送给通知（行动）管理系统。告警管理系统还支持设置告警事务阶段和处理人。
通知（行动）管理	告警子系统，负责管理告警的通知渠道和对象。通知（行动）管理系统由行动策略、内容模板、日历、用户、用户组、值班组和渠道额度等组成。 通知（行动）管理系统通过行动策略将告警动态分派给特定的通知渠道，再通知给目标用户、用户组或值班组。通知（行动）管理系统还支持告警通知升级、自定义告警通知内容等操作。
开放告警	告警子系统，负责接收外部的监控系统数据。开放告警系统由开放告警服务和开放告警应用等组成。 每个开放告警应用对外提供一个端口，接收来自于外部告警监控服务（例如Zabbix、Prometheus）触发的告警（包括恢复通知），预处理后发送给告警管理系统进行后续处理。

6.2. 日志（Log）

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

格式

文本日志（LogFile）、事件（Event）、数据库日志（BinLog）、时序数据（Metric）等数据都是日志的不同载体。日志服务采用半结构化的数据模式定义一条日志，包含日志主题（Topic）、时间（Time）、内容（Content）、来源（Source）和标签（Tags）五个数据域。日志服务对各个数据域的格式要求不同，详细说明如下：

数据域	说明	格式
日志主题（Topic）	自定义字段，用于标识日志的主题。例如您可以根据日志类型为网站相关日志设置不同的日志主题（access_log、operation_log）。更多信息，请参见日志主题（Topic）。	包括空字符串在内的任意字符串，大小为0~128字节。 该字段为空字符串时，表示未设置日志主题。
时间（Time）	日志服务保留字段，一般为日志中的时间信息（日志生成时间）或采集日志时Logtail所在主机的系统时间。	Unix时间戳，即从1970-1-1 00:00:00 UTC开始所经过的秒数。

内容（Content）	记录日志的具体内容，由一个或多个内容项组成，每一个内容项为一个键值对（Key:Value）。	Key:Value格式，详细说明如下：
		• Key为UTF-8编码字符串，可以为字母、下划线和数字但不以数字开头。字符串大小为1~128字节。不可使用如下字段：
		  ◦ __time__
		  ◦ __source__
		  ◦ __topic__
		  ◦ __partition_time__
		  ◦ _extract_others_
		  ◦ __extract_others__
		• Value为任意字符串，大小不超过1 MB。

来源（Source）	日志来源，例如产生日志的服务器IP地址。	任意字符串，大小为0~128字节。
标签（Tags）	日志标签。包括： • 自定义标签：通过PutLogs接口，在写入数据时添加标签。 • 系统标签：日志服务为您添加的标签，包括__client_ip__和__receive_time__。	字典格式，Key和Value均为字符串类型。在日志中以__tag__:为前缀展示。

示例

以下以一条网站访问日志为例，说明原始日志与日志服务中数据模型的映射关系。

- 原始日志

```
127.0.0.1 - - [01/Mar/2021:12:36:49  0800] "GET /index.html HTTP/1.1" 200 612 "-" "Mozill
a/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/6
8.0.3440.106 Safari/537.36
```

- 通过极简模式采集到日志服务后的日志样例

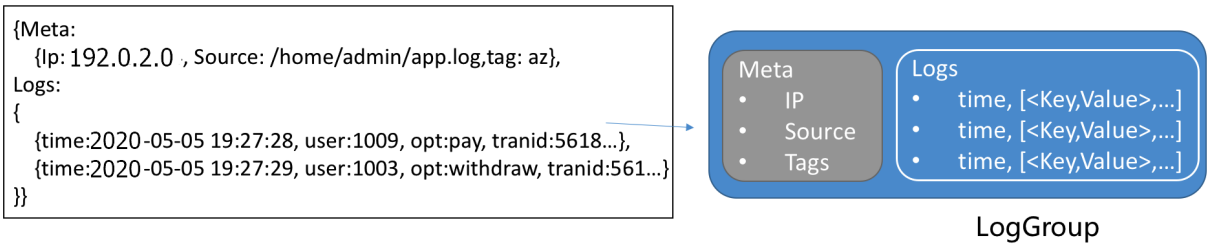
- 通过完整正则模式采集到日志服务后的日志样例

6.3. 日志组（LogGroup）

日志组（LogGroup）是一组日志的集合，是写入与读取日志的基本单元。一个日志组中的数据包含相同Meta（IP地址、Source等信息）。

写入日志到日志服务或从日志服务读取日志时，多条日志被打包为一个日志组，以日志组为单元进行写入与读取。该方式可减少读写次数，提高业务效率。每个日志组最大长度为5 MB。

日志服务的基本数据模型请参见Logstore数据模型。



LogGroup

6.4. 项目（Project）

项目（Project）是日志服务的资源管理单元，是进行多用户隔离与访问控制的主要边界。

Project中包含Logstore、MetricStore和机器组等资源，同时它也是您访问日志服务资源的入口。建议使用不同的Project管理不同的应用、产品或项目中的数据。具体说明如下：

- 组织、管理不同的Logstore或MetricStore。在实际使用中，您可能需要使用日志服务采集及存储不同项目、产品或者环境的日志。您可以把不同项目、产品或者环境中的日志分类管理在不同Project中，便于后

续的日志消费、导出或者分析。

- 用于访问控制隔离。您可以为RAM用户授予指定Project的操作权限。
- 提供日志服务资源的访问入口。日志服务为每个Project配置一个独立的访问入口。该访问入口支持通过网络写入、读取及管理日志。关于访问入口的更多信息，请参见服务入口。

6.5. 日志库（Logstore）

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Logstore隶属于一个Project，每个Project中可创建多个Logstore。您可以根据实际需求在目标Project中创建多个Logstore，一般是为同个应用中不同类型的日志创建独立的Logstore。例如您要采集App A所涉及的操作日志（operation_log）、应用程序日志（application_log）以及访问日志（access_log），您可以创建一个名为app-a的Project，并在该Project下创建名为operation_log、application_log和access_log的Logstore，用于分别存储操作日志、应用程序日志和访问日志。

您在执行写入日志、查询和分析日志、加工日志、消费日志、投递日志等操作时，都需要指定Logstore。具体说明如下：

- 以Logstore为采集单元，采集日志。
- 以Logstore为存储单元，存储日志以及执行加工、消费、投递等操作。
- 在Logstore中建立索引，用于查询和分析日志。

6.6. 时序库（MetricStore）

时序库（MetricStore）是日志服务中时序数据的采集、存储和查询单元。

每个MetricStore隶属于一个Project，每个Project中可创建多个MetricStore。您可以根据实际需求为某个项目创建多个MetricStore，一般是为不同类型的时序数据创建不同的MetricStore。例如您需要采集基础主机监控数据、云服务监控数据、业务应用监控数据，您可以创建一个名为demo-monitor的Project，然后在该Project下创建名为host-metrics、cloud-service-metrics和app-metrics的MetricStore，用于分类存储基础主机监控数据、云服务监控数据和业务应用监控数据。

您在执行写入、查询和分析、消费时序数据时，都需要指定MetricStore。具体说明如下：

- 以MetricStore为采集单元，采集时序数据。
- 以MetricStore为存储单元，存储时序数据以及执行消费操作。
- 查询和分析时序数据，支持SQL92语法和PromQL语法。

6.7. 时序数据（Metric）

日志服务的时序数据类型遵循Prometheus的定义规范，在时序库中所有的数据都按照时序类型存储。时序数据由时序标识和数据点组成，相同时序标识的数据组成时间线。

时序标识

每条时间线都有一个唯一的时序标识，由Metric name和Labels组成。

Metric name类型为字符串，一般用于标识指标类型，Metric name需遵循正则表达式：**[a-zA-Z_:][a-zA-Z0-9_:]*** 。例如http_request_total表示接收到的HTTP请求的总数。

Labels由一组Key-Value对组成，用于标识指标的相关属性，Label的Key需遵循正则表达式： **[a-zA-Z_][a-zA-Z0-9_]*** ，Label的Value不能包含竖线（|），其它不做限制。例如method为POST，URL为/api/v1/get 。

数据点

数据点代表时间线在具体某个时间点的值，每个数据点由时间戳和值组成。其中时间戳精度为纳秒，值的类型为double。

编码方式

时序数据的写入协议和日志写入协议一致，使用Protobuf的数据编码方式。时序标识和数据点都在content字段中，具体表示方式如下所示。

Key	说明	示例
__name__	Metric名称	nginx_ingress_controller_response_size
__labels__	label信息，分隔符形式：{key}#$#{value}\|{key}#$#{value}|{key}#$#{value} 说明 Label名称需按照字母顺序进行排序。	app#$#ingress-nginx\|controller_class#$#nginx|controller_namespace#$#kube-system\|controller_pod#$#nginx-ingress-controller-589877c6b7-hw9cj
__time_nano__	时间戳，单位为纳秒	1585727297293000000
__value__	值	36.0

6.8. 分区（Shard）

日志服务使用Shard控制Logstore或MetricStore的读写数据的能力，数据必定保存在某一个Shard中。

Shard范围

每个Shard均有范围，为MD5左闭右开区间[BeginKey,EndKey)。每个Shard范围不会相互覆盖，且属于整个MD5范围内[00000000000000000000000000000000,ffffffffffffffffffffffffffffffff）。您可以在创建Logstore或MetricStore时指定Shard个数，日志服务将自动平均划分整个MD5范围。

- BeginKey：指定Shard范围的起始值，Shard范围中包含该值。
- EndKey：指定Shard范围的结束值，Shard范围中不包含该值。

例如Logstore A中包含4个Shard，各个Shard范围如下：

Shard范围

Shard ID	范围
Shard0	[00000000000000000000000000000000,40000000000000000000000000000000)
Shard1	[40000000000000000000000000000000,80000000000000000000000000000000)
Shard2	[80000000000000000000000000000000,c0000000000000000000000000000000)

Shard ID	范围
Shard3	[c00000000000000000000000000000000,ffffffffffffffffffffffffffffffff)

在Shard读写数据过程中，读数据时必须指定Shard ID，写数据时可通过负载均衡模式或者指定Hash Key的模式。

- 负载均衡模式：每个数据包随机写入当前可用的Shard中。
- 指定Hash Key模式：指定MD5的Key值，数据将被写入包含该Key值的Shard中。

  例如Shard范围如Shard范围所示，当您写入数据时指定MD5的Key值为5F时，则数据将被写入包含5F的Shard1上；当您写入数据时指定MD5的Key值为8C时，则数据将被写入包含8C的Shard2上。

Shard的读写能力

每个Shard提供一定的服务能力，详细说明如下：

- 写入：5 MB/s或500次/s
- 读取：10 MB/s或100次/s

建议您根据实际数据流量规划Shard个数。当数据流量超出读写能力时，及时分裂Shard以增加Shard个数，从而达到更大的读写能力。当数据流量远未达到Shard的最大读写能力时，及时合并Shard以减少Shard个数，从而降低活跃Shard租用费用。

例如您有两个readwrite状态的Shard，最大可提供10 MB/s的数据写入服务。当您实时写入数据流量达到14 MB/s时，建议分裂其中一个Shard，使readwrite状态的Shard数量达到3个。当您实时写入数据流量仅为3 MB/s时，建议您合并两个Shard。

> 注意
>
> - 当写入数据的API持续报告403或者500错误时，您可以通过Logstore云监控查看流量和状态码判断是否需要增加Shard。
> - 超过Shard服务能力的读写，日志服务会尽可能服务，但不保证服务质量。

Shard状态

Shard状态包括readwrite（读写）和readonly（只读）。

创建Shard时，所有Shard状态均为readwrite状态。执行分裂或合并操作后，Shard状态变更为readonly，并生成新的readwrite状态的Shard。Shard状态不影响其数据读取的性能。readwrite状态的Shard可保证数据写入性能，readonly状态的Shard不提供数据写入服务。

分裂与合并

日志服务支持分裂和合并Shard。

- 分裂操作是指将一个Shard分裂为另外两个Shard，即分裂后Shard数量增加2。两个新生成的Shard的状态为readwrite，排列在原Shard之后且两个Shard的MD5范围覆盖原Shard的MD5范围。

  分裂Shard时，需指定一个处于readwrite状态的Shard。分裂完成后，原Shard状态由readwrite变为readonly，该Shard中的数据仍可被消费，但该Shard不支持写入新数据。
- 合并操作是指将两个Shard合并为一个Shard。新生成的Shard的状态为readwrite，排列在原Shard之后且其MD5范围覆盖原来两个Shard的MD5范围。

合并Shard时，需指定一个处于readwrite状态且未排列在最后一个的Shard，日志服务自动找到所指定Shard右侧相邻的Shard，并进行合并。合并完成后，原来两个Shard的状态由readwrite变为readonly，这两个Shard中的数据仍可被消费，但这两个Shard不支持写入新数据。

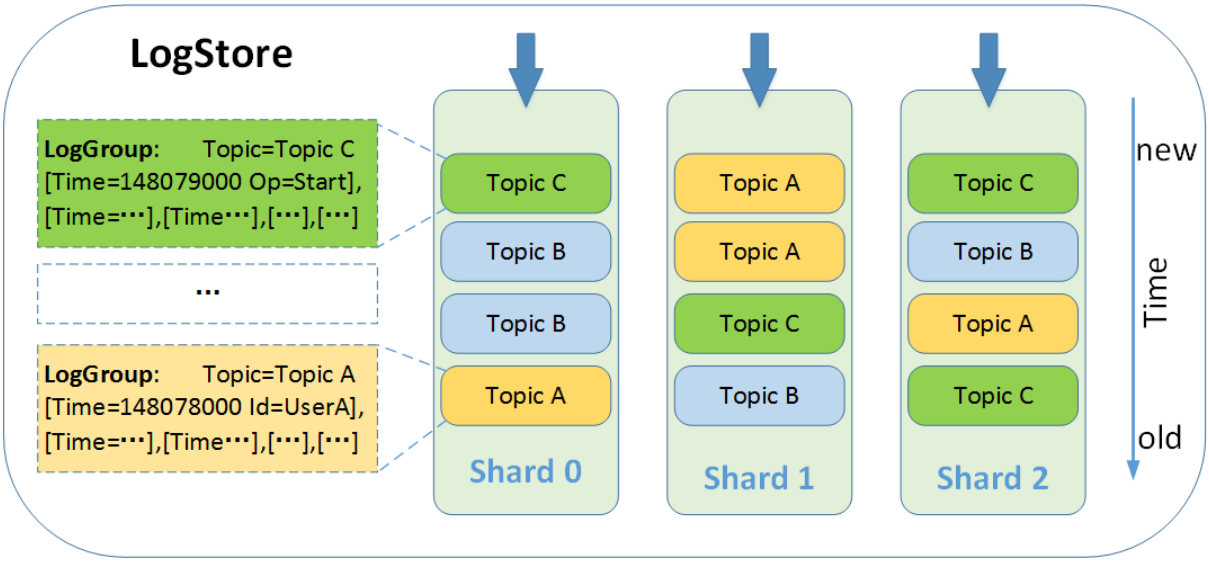
6.9. 日志主题（Topic）

日志主题（Topic）是日志服务的基础管理单元。您可在采集日志时指定Topic，用于区分日志。

Topic可用于区分不同服务、用户、实例等产生的日志。例如系统A由前端HTTP请求处理模块、缓存模块、逻辑处理模块和存储模块组成，您可设置前端HTTP请求处理模块日志的Topic为http_module，缓存模块日志的Topic为cache_module、逻辑处理模块日志的Topic为logic_module和存储模块日志的Topic为store_module。当日志被采集到的同一个Logstore中后，您可通过Topic进行区分。

如果不需要区分Logstore中的日志，则在采集日志时设置Topic为*空-不生成Topic*即可。空字符串是一个有效的Topic，即Topic的值为空字符串。

Logstore、Topic、Shard之间的关系如下：

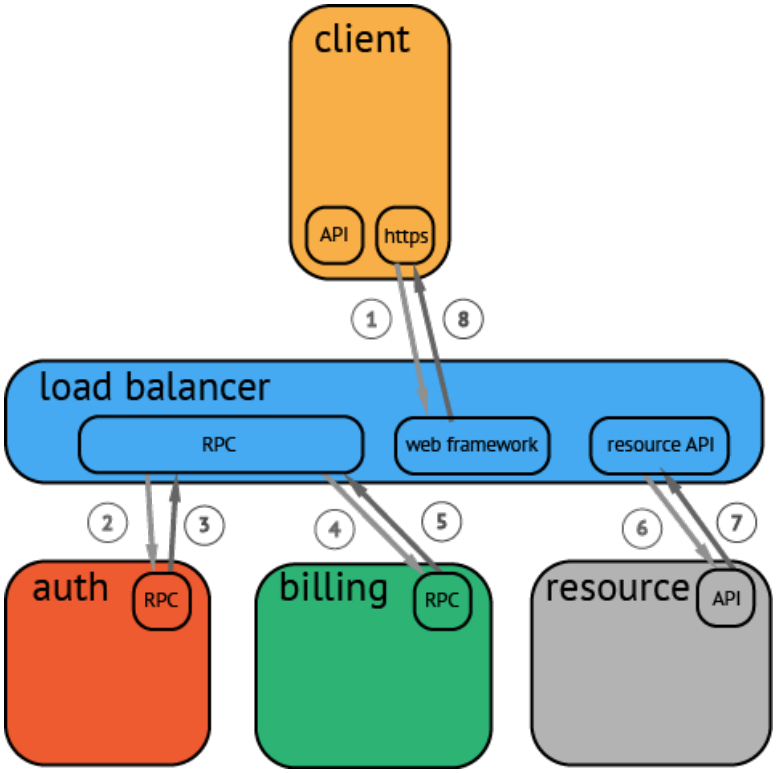


6.10. 链路数据（Trace）

链路数据（Trace）用于记录单次请求范围内的处理信息，其中包括服务调用和处理时长等数据。

一条链路数据对应一条调用链。在广义上，一个调用链代表一个事务或者流程在（分布式）系统中的执行过程。在OpenTracing标准中，调用链是多个Span组成的一个有向无环图（Directed Acyclic Graph，简称DAG），每一个Span代表调用链中被命名并计时的连续性执行片段。

下图是一个分布式调用的例子：客户端发起请求，请求首先到达负载均衡器，接着经过认证服务、计费服务，然后请求资源，最后返回结果。



7.使用限制

7.1. 基础资源

本文介绍日志服务基础资源的使用限制。

限制项	说明	备注
Project	您在1个阿里云账号下最多可创建50个Project。	如您有更大的使用需求，请提工单申请。
Logstore	您在1个Project中最多可创建200个Logstore。	如您有更大的使用需求，请提工单申请。
Shard	• 您在一个Project中最多可创建400个Shard。 • 使用控制台创建Logstore时，您在1个Logstore中最多可创建10个Shard；使用API创建Logstore时，您在1个Logstore中最多可创建100个Shard。两种方式都可以通过分裂操作来增加Shard。	如您有更大的使用需求，请提工单申请。
Logtail配置（LogtailConfig）	您在1个Project中最多可创建200个Logtail配置。	如您有更大的使用需求，请提工单申请。
日志保存时间	日志支持永久保存。 您也可以自定义日志保存时间。单位为天。取值范围为1~3000。	不涉及
机器组（MachineGroup）	您在1个Project中最多可创建200个机器组。	如您有更大的使用需求，请提工单申请。
消费组（ConsumerGroup）	您在1个Logstore中最多可创建30个消费组。	可以删除不使用消费组。
快速查询（SavedSearch）	您在1个Project中最多可创建100个快速查询。	不涉及
仪表盘（Dashboard）	• 您在1个Project中最多可创建100个仪表盘。 • 1个仪表盘最多可包含100张分析图表。	不涉及
LogItem	• 通过API采集：单个LogItem最大为1 MB。 • 通过Logtail采集：单个LogItem最大为512 KB。	不涉及

限制项	说明	备注
LogItem（Key）	LogItem（Key）的长度最大为128字节。	不涉及
LogItem（Value）	LogItem（Value）的长度最大为1 MB。	不涉及
日志组（LogGroup）	1个日志组的长度最大为5 MB。	不涉及
告警	您在1个Project中最多可创建100个告警。	如果您有更大的使用需求，请提工单申请。

说明 通过日志服务命令行工具CLI可以查看当前的Logstore、Shard、仪表盘等基础资源使用情况。更多信息，请参见使用CLI查看基础资源使用状况。

7.2. 数据读写

本文介绍日志服务读写数据的限制。

类别	限制项	说明	备注
Project	写入流量	原始数据写入流量最大为30 GB/min。	如超过限制，返回状态码403，提示Inflow Quota Exceed，且您有更大的使用需求，请提工单申请。
	写入次数	写入次数最大为600000次/min。	如超过限制，返回状态码403，提示Write QPS Exceed，且您有更大的使用需求，请提工单申请。
	读取次数	读取次数最大为600000次/min。	如超过限制，返回状态码403，提示Read QPS Exceed，且如您有更大的使用需求，请提工单申请。
Shard	写入流量	• 日志库已配置索引，原始数据写入流量最大为5 MB/s。 • 日志库未配置索引，原始数据写入流量最大为10 MB/s。	非硬性限制，超过时系统会尽可能服务，但不保证服务质量。
	写入次数	写入次数最大为500次/s。	非硬性限制，超过时系统会尽可能服务，但不保证服务质量。
	读取流量	读取流量最大为10 MB/s。	非硬性限制，超过时系统会尽可能服务，但不保证服务质量。
	读取次数	读取次数最大为100次/s。	非硬性限制，超过时系统会尽可能服务，但不保证服务质量。

7.3. Logtail

本文介绍Logtail的使用限制。

文件采集限制

分类	限制说明
文件编码	支持UTF8或GBK的编码日志文件，建议使用UTF8编码以获得更好的处理性能。如果日志文件为其它编码格式则会出现乱码、数据丢失等错误。
日志文件大小	无限制。
日志文件轮转	支持，轮转文件名支持配置为 `.log*` 或者 `.log` 。
日志解析阻塞时采集行为	日志解析阻塞时，Logtail会将该日志文件FD保持打开状态；如果解析阻塞期间出现多次日志文件轮转，Logtail会尽可能保持各个轮转日志解析顺序。如果未解析的日志轮转超过20个，则后续文件不被处理。更多信息，请参见相关技术文章。
软链接	支持监控目录为软链接。
单条日志大小	单条日志大小限制为512 KB。多行日志按行首正则表达式划分后，每条日志大小限制仍为512 KB。如果日志超过512 KB后，会强制拆分多块进行采集。例如：单条日志大小为1025 KB，则第一次处理前512 KB，第二次处理512 KB，第三次处理1 KB。
正则表达式	正则表达式类型支持Perl兼容正则表达式。
同一文件对应多个Logtail配置	默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。
文件打开行为	Logtail会保持被采集文件处于打开状态，如果该文件超过5分钟未修改，则会关闭该文件（未发生轮转情况下）。
首次日志采集行为	Logtail只采集增量的日志文件。首次发现文件修改后，如果文件大小超过1 M，则从最后1M处开始采集，否则从开始位置采集；如果配置下发后日志文件一直无修改，则不采集该文件。
非标准文本日志	对于日志中包含\0的行，该条日志会被截断到第一个\0处。

Checkpoint管理

项目	能力与限制
Checkpoint超时时间	如果文件超过30天未修改，则会删除该Checkpoint。
Checkpoint保存策略	定期保存（15分钟），程序退出时会自动保存。您可以调整参数，更多信息，请参见设置Logtail启动参数。
Checkpoint保存位置	保存路径默认为 `/tmp/logtail_checkpoint` 。您可以调整参数，更多信息，请参见设置Logtail启动参数。

配置限制

项目	能力与限制
配置更新	配置更新生效的延时约30秒。
配置动态加载	支持，且其中某一配置更新不影响其他采集。
配置数	理论上无限制，建议一台服务器中的Logtail配置数不超过100个。
多租户隔离	各个Logtail配置隔离。更多信息，请参见相关技术文章。

资源、性能限制

项目	能力与限制
日志处理吞吐能力	原始日志流量默认限制为20 MB/s（数据会编码压缩后上传，一般压缩率为5-10倍）。超过该日志流量则有可能丢失日志，可调整参数，更多信息，请参见设置Logtail启动参数。
最大性能	单核能力：极简模式日志最大处理能力为100 MB/s，正则默认最大处理能力为20 MB/s（和正则复杂度有关），分隔符日志最大处理能力为40 MB/s，JSON日志最大处理能力为30 MB/s；开启多个处理线程性能可提高1.5-3倍左右。
监控目录数	主动限制监控的目录层深，避免出现过多消耗用户资源。如果监控上限已到，则放弃监控更多目录和日志文件。限制最多3000个目录（含子目录）。
监控文件数	每台服务器上的每个Logtail配置监控的最大文件数量为10,000个，每台服务器上的Logtail客户端最多可监控100,000个文件。超出限制的文件不监控。 达到限制时，您可以： • 在Logtail配置中提高监控目录的精度。 • 修改mem_usage_limit参数，提高Logtail内存。更多信息，请参见设置Logtail启动参数。 Logtail内存最大可调整至2 GB，表示每个Logtail配置可监控100,000个文件，每个Logtail客户端可监控的文件数对应提高至1,000,000个。
默认资源限制	默认Logtail最多会占用40%CPU、256 MB内存，如果日志产生速率较高，可调整参数，更多信息，请参见设置Logtail启动参数。
资源超限处理策略	如果Logtail占用相关资源超过最大限制的时间超过5分钟，则Logtail会强制重启，此时数据可能会丢失或重复。

错误处理限制

项目	能力与限制
网络错误处理	在出现网络异常时会主动重试并自动调整重试间隔。
资源配额超限处理	如果数据发送速率超出Logstore最大配额，Logtail会阻塞采集并自动重试。更多信息，请参见相关技术文章。
超时最大尝试时间	如果数据持续发送失败超过6小时，则丢弃该数据。

项目	能力与限制
状态自检	支持异常情况下自动重启，例如程序异常退出及使用资源超限等。

其他限制

项目	能力与限制
日志采集延迟	正常情况下从日志写入磁盘到Logtail采集的日志延迟不超过1秒（阻塞状态下除外）。
日志上传策略	Logtail会将同一文件的日志自动聚合上传，聚合条件为日志超过2000条、日志总大小超过2 M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。

7.4. 数据加工

本文介绍日志服务数据加工的使用限制。

作业配置

限制项	说明
作业数	1个Project中最多可创建100个数据加工作业。 注意 数据加工作业处于停止或者运行完成状态时，依然会占用配额，建议及时清理停止或者运行完成状态、并且确认不再使用的数据加工作业，以减少配额占用。更多信息，请参见管理数据加工作业。 如您有更大的使用需求，请提工单申请。
源数据Logstore消费组依赖	1个数据加工作业运行依赖源数据Logstore的一个消费组实例。 加工作业在运行中，不能针对该作业依赖的消费组实例执行**删除**和**重置消费点位**的操作，否则作业将重新从其配置的起始时间开始消费数据，可能会导致结果数据出现重复。 注意 为了优化数据加工的运行效率，作业的Shard消费进度会定时更新至其依赖的消费组实例，所以该消费组实例的GetCheckPoint接口结果不能反映最新的加工进度。准确的加工进度数据请参考数据加工仪表盘中的**Shard消费延迟**模块。 更多信息，请参见加工原理、术语表和消费组接口。

限制项	说明
源数据Logstore消费组数目	1个Logstore中最多可创建30个消费组，即1个源数据Logstore最多可创建30个数据加工作业。更多信息，请参见基础资源。 超出此限制时，加工作业启动后无法正常运行，作业通过其运行日志输出具体错误信息，详情请参考错误日志查看方式。 **注意** 数据加工作业处于停止或者运行完成状态时，日志服务并不会主动删除其依赖的消费组实例，建议及时清理处于停止或运行完成状态、并且确认不再使用的数据加工作业，减少无效消费组实例。更多信息，请参见管理数据加工作业。
作业时间范围修改	运行中的作业修改了时间范围后，将从新指定的起始时间点开始运行，并处理新时间范围内的所有数据。 1. 如果需要扩展时间范围：建议保留现有作业，新增作业补全需要扩展的时间范围。 2. 如果需要缩小时间范围：已经写到目标的数据将不会被删除，所以建议必要时清除已经写到目标的数据，再修改作业，以防数据重复。
输出目标数	在1个数据加工作业中配置独立静态输出目标数限制为20个。 在加工代码中使用1个静态输出目标配置，动态指定Project和Logstore输出数限制为200个。超出此限制时，写入到新增目标的数据将被丢弃。

加工处理

限制项	说明
快速预览	数据加工快速预览功能用于调试数据加工代码，使用有以下限制： • 不支持连接到外部资源（RDS/OSS/SLS等），可通过自定义输入测试维表数据。 • 单次请求测试原始数据和维表各自均不超过 1 MB 数据，超出时该请求返回错误。 • 单次请求最多返回前100条加工结果。 数据加工**高级预览**功能无此限制。

限制项	说明
运行时并发度	数据加工作业使用源数据Logstore的读写Shard数目作为作业的最大运行时并发度。更多信息，请参见加工原理。 Logstore的Shard使用限制请参考基础资源，Logstore的**分裂Shard**操作请参考管理Shard。 **注意** • 数据加工作业运行并发度不足不会触发源数据Logstore的**自动分裂Shard**功能，需要手动分裂源数据Logstore的Shard，才能提高加工作业的运行并发度。**自动分裂Shard**操作请参考管理Shard。 • 分裂源数据Logstore的Shard提升作业的最大运行时并发度，只对分裂Shard操作以后新写入的数据有效。在分裂Shard操作之前已写入的数据，其最大运行时并发度由数据写入时的源数据Logstore读写Shard数目决定。
并发单元数据负载	数据加工作业单个并发单元的数据负载，由其所运行的源数据Logstore Shard中存储数据量所决定。如果写入源数据Logstore的数据在其Shard间分布不均衡，会使得数据加工作业运行时出现热点并发单元，导致部分Shard处理延迟。 如果源数据通过**Key路由Shard模式（KeyHash）**写入，建议合理分配Key与Shard，尽可能减少数据分布不均衡。关于数据写入详情请参考PutLogs。
内存使用	数据加工作业单个并发单元的内存使用限制为6 GB，超过此限制时作业运行性能将变慢，导致处理延迟。 内存使用超限制的引起原因是单次拉取的LogGroup数量过大，需要修改高级参数 `system.process.batch_size` 来调整内存使用。 **注意** 高级参数 `system.process.batch_size` 的默认值（最大值）是1000，可以调整为1000以内的正整数。
CPU使用	数据加工作业单个并发单元的CPU使用限制为100%，可以根据上文扩展并发度上限实现更高的CPU要求。
维表数据量	维表数据条目数限制为一百万，数据内存使用限制为1 GB。数据量超过该限制时，将做截断处理，使用限制内的数据内容。涉及函数包括res_rds_mysql、res_log_logstore_pull和res_oss_file等。

结果数据写入

限制项	说明

限制项	说明
目标Logstore写入	将处理结果写入目标Logstore时，需要满足Lgstore的数据写入限制，具体请参考基础资源和数据读写。 如果在使用e_output、e_coutput函数时，指定 `hash_key_field` 或者 `hash_key` 参数且通过**Key路由Shard模式（KeyHash）**写入数据至目标Logstore，建议合理分配Key与Shard，尽可能减少数据分布不均衡。 此限制可通过数据加工作业日志进行定位，参考错误日志查看方式。 **注意** 数据加工在遇到目标Logstore写入限制错误时会执行无限重试，以保证数据完整，但是加工作业进度会因此受影响，导致负责当前源Shard的负载处理延迟。
跨地域传输	通过公网Endpoint完成跨地域数据传输时，由于公网网络质量无法保证，加工结果写入目标Logstore可能会出现网络错误，导致数据加工作业处理延迟。日志服务Endppint请参考服务入口。 建议为目标Project开启全球加速服务，并在数据加工中配置其全球加速Endpoint以提高网络传输稳定性。更多信息，请参见开启全球加速服务。

7.5. 查询和分析

本文介绍日志服务查询和分析的限制。

查询

限制项	说明	备注
关键词个数	关键词查询时，除布尔逻辑符外的条件个数。每次查询最多30个。	无
字段值大小	单个字段值最大为10 KB，超出部分不参与查询。	如果单个字段长度大于10 KB，有一定几率无法通过关键词查询到日志，但数据仍然是完整的。
操作并发数	单个Project支持的最大查询操作并发数为100个。	例如100个用户同时在同一个Project的各个Logstore中执行查询操作。
返回结果	每次查询时，每页最多显示100条查询结果，您可翻页读取完整的查询结果。	无
单条日志内容显示	由于网页浏览器性能原因，对于超过10,000个字符的日志，日志服务只会对前10,000个字符进行DOM切词处理。	如果超出10,000个字符，控制台会提示“该日志存在超过10,000个字符的日志数据，部分显示上会有降级处理”。
模糊查询	执行模糊查询时，日志服务最多查询到符合条件的100个词，并返回包含这100个词并满足查询条件的所有日志。更多信息，请参见模糊查询。	无

限制项	说明	备注
查询结果排序	默认按照分钟级时间从最新开始展示。	无

分析

限制项	普通实例	独享实例
操作并发数	单个Project支持的最大分析操作并发数为15个。 例如15个用户同时在同一个Project的各个Logstore中执行分析操作。	单个Project支持的最大分析操作并发数为100个。 例如100个用户同时在同一个Project的各个Logstore中执行分析操作。
数据量	单个Shard单次仅支持分析1 GB数据。	单次分析最大支持扫描2000亿行数据。
开启模式	默认开启。	通过开关开启。具体操作，请参见开启SQL独享版。
费用	免费。	根据实际使用的CPU时间付费。
数据生效机制	分析功能只对开启统计功能后写入的数据生效。 如果您需要分析历史数据，请对历史数据重建索引。更多信息，请参见重建索引。	分析功能只对开启统计功能后写入的数据生效。 如果您需要分析历史数据，请对历史数据重建索引。更多信息，请参见重建索引。
返回结果	执行分析操作后，默认最多返回100行数据。 如果您需要返回更多数据，请使用LIMIT语法。更多信息，请参见LIMIT子句。	执行分析操作后，默认最多返回100行数据。 如果您需要返回更多数据，请使用LIMIT语法。更多信息，请参见LIMIT子句。
字段值大小	单个字段值最大为16 KB，超出部分不参与分析。	单个字段值最大为16 KB，超出部分不参与分析。
超时时间	分析操作的最大超时时间为55秒。	分析操作的最大超时时间为55秒。
Double类型的字段值位数	Double类型的字段值最多52位。 如果浮点数编码位数超过52位，会造成精度损失。	Double类型的字段值最多52位。 如果浮点数编码位数超过52位，会造成精度损失。

7.6. Scheduled SQL

本文介绍Scheduled SQL的使用限制。

查询与分析

> **注意** Scheduled SQL仅支持SQL独享版引擎。

限制项	说明
操作并发数	单个Project支持的最大分析操作并发数为150个。 例如150个用户同时在同一个Project的各个Logstore中执行分析操作。
数据量	单次分析最大支持扫描2000亿行数据。
数据生效机制	分析功能只对开启统计功能后写入的数据生效。 如果您需要分析历史数据，请对历史数据重建索引。更多信息，请参见重建索引。
返回结果	• 执行分析操作后，默认最多返回100行数据，超出部分不会返回。 • 如果您需要返回更多数据，请使用LIMIT语法（最大支持返回100万行数据）。更多信息，请参见LIMIT子句。超出LIMIT语法限制的部分不会返回。 • 最大输出的数据量限制为20 GB，超出部分不会返回。
字段值大小	单个字段值默认为2048字节（2 KB），最大为16384字节（16 KB），超出部分不参与分析。 您可以在配置索引时，修改字段值的最大长度（64字节~16384字节）。具体操作，请参见配置索引。
超时时间	分析操作的最大超时时间为10分钟。
Double类型的字段值位数	Double类型的字段值最多52位。 如果浮点数编码位数超过52位，会造成精度损失。
模糊查询	执行模糊查询时，日志服务最多查询到符合条件的100个词，并返回包含这100个词并满足查询条件的所有日志。
查询不精确	结果不精确不会报错，会记录在实例状态以及作业执行记录（需手动开启）中。
数据延迟	当数据存在延迟时，可能存在数据漏查的风险。即如果某时间点的数据在对应的调度实例执行完成之后才到达，则在下一个调度实例中也不会被执行。更多信息，请参见如何保证SQL分析的数据准确性。
时间窗口	单次查询时间窗口最大为24小时，最小为1分钟。

数据写入

限制项	说明
目标Logstore写入阈值	如果写入数据时超过阈值，Scheduled SQL作业将重试10分钟以上。超过重试时间后，将返回错误信息。更多信息，请参见数据读写。
跨地域传输	中国内的跨地域传输数据时，网络较为稳定，但会有较高延迟（延迟大小随地域的不同而不同）。 国际网络无法保证。

作业执行

限制项	说明
超时时间	最大超时时间为1800秒，超过将视为本次作业执行失败。 建议添加告警监控任务，便于及时发现问题重试错误实例。更多信息，请参见为Scheduled SQL作业设置告警、重试Scheduled SQL作业实例。
重试次数	最大重试次数为100次，超过将视为本次作业执行失败。
延迟执行	延迟执行时间最大为120秒，延迟执行使用场景实例请参见调度与执行场景。
历史执行记录	单个作业的历史执行记录最多保存14天。 建议添加告警监控任务，便于及时发现问题重试错误实例。更多信息，请参见为Scheduled SQL作业设置告警、重试Scheduled SQL作业实例。

7.7. 投递

本文介绍数据投递的稳定性和使用限制。

MaxCompute投递（新版）

稳定性

- 读日志服务

稳定项	说明
可用性	可用性较高。 如果日志服务出错，无法读取数据，MaxCompute投递作业会在内部至少重试10次。如果仍然失败，作业执行会报错，然后作业重启。

- 写MaxCompute

稳定项	说明
并发度	按照日志服务Shard进行分区并创建投递实例，支持快速扩容。 如果日志服务源Logstore进行Shard分裂，可以在数秒以内完成投递实例的扩容，加快数据导出速度。
数据不丢失	MaxCompute投递作业基于消费组进行扩展，提供一致性保证。投递完成后，才会提交offset，因此可以保证数据写入MaxCompute之前，offset不被提交，即保证投递数据不丢失。
Schema变更	如果投递过程中在MaxCompute表中添加了新列，则新列只会被写入新分区，不会被写入旧分区以及当前分区。

- 处理脏数据

错误项	是否计入失败条数	说明
分区错误	是	常见场景为分区不合法或分区列不存在。该条数据不会写入MaxCompute。
数据列不合法	否	常见场景类型为不匹配或者类型转换失败。该列数据不会写入MaxCompute，其余列正常写入MaxCompute。
数据列过长	否	常见场景为数据超出string类型或者varchar长度限制。该列数据经过截断后写入MaxCompute，其余列正常写入MaxCompute。

- 监控告警

稳定项	说明
监控告警	数据投递有完善的监控，可实时追踪投递作业的延迟、流量等指标。您可以根据业务需求，配置自定义告警，及时发现投递问题（例如导出实例不足、网络Quota限制等）。具体操作，请参见为MaxCompute投递作业（新版）设置告警。

- 重启作业

稳定项	说明
分区数过多	作业重启时，因为分区数过多（5分钟仍未完成写入），可能导致数据重复。
数据写入失败	作业重启且数据写入MaxCompute失败（授权错误、网络错误）时，可能导致数据部分重复。

使用限制

- 网络

限制项	说明
同地域投递的网络	同地域投递时，数据通过阿里云内网传输，因此网络稳定性和速度更有保障。
跨地域投递的网络	跨地域投递时，数据通过公网传输，因此网络稳定性和速度没有保障。 网络出现问题时，内部会重试至少10次，如果仍然失败，作业执行会报错，导致作业重启。

- 读流量

限制项	说明

限制项	说明
读流量	单个Project以及单个Shard存在最高流量限制。更多信息，请参见数据读写。 如果超过最高流量限制，请分裂Shard或者申请扩容Project读流量限制。超过限制，会导致MaxCompute投递作业读取数据失败，并在内部至少重试10次，如果仍然失败，作业执行会报错，然后作业重启。

- 写入性能

限制项	说明
并发实例	支持最大的导出并发实例为64。 如果日志服务Shard数量超过64，则会合并多个Shard到一个实例中进行导出处理，并且尽量保证每个实例中的Shard数相同。
写入阈值	MaxCompute单分区写入上限为10 MB/s。 超过限制时，写数据到MaxCompute会不稳定。

- 权限管理

限制项	说明
写授权	MaxCompute写授权支持RAM用户以及RAM角色两种方式，并且需要在MaxCompute侧进行单独操作。

- 数据类型

- 普通列

类型	示例	说明
string	"hello"	最大长度为8 MB。
datetime	"2021-12-22 05:00:00.123456"	日志服务中的数据需满足MaxCompute的数据格式要求。
date	"2021-12-22"	日志服务中的数据需满足MaxCompute的数据格式要求。
timestamp	1648544867	毫秒级以及秒级精度。
decimal	1.2	日志服务中的数据需满足MaxCompute的数据格式要求。
char	"hello"	最大长度为255字节。
varchar	"hello"	最大长度为65535字节。
binary	"hello"	最大长度为8 MB。
bigint	123	最大支持int64。
boolean	1	■ 1、t、T、true、TRUE、True解析为True。 ■ 0、f、F、false、FALSE、False解析为False。
double	1.2	最大支持64位浮点数。
float	1.2	最大支持32位浮点数。
integer	123	最大支持int32。
smallint	12	最大支持int16。
tinyint	12	最大支持int8。

- 分区列

限制项	说明
分区列	按照字符串处理，需要满足MaxCompute分区列的格式要求。

7.8. 告警

本文介绍日志服务告警的限制。

告警（新版）

类别	限制项	说明
告警监控	最大告警监控规则数	每个Project中最多创建100个告警监控规则。 如果您有更大的使用需求，请提工单申请。单个Project中最大可扩容至200个。
	查询和分析操作一般性限制	查询和分析操作的限制项请参见查询和分析。
	查询和分析操作并发限制	如果在一个Project中，同时执行较多的查询和分析操作（例如通过SDK进行大量查询和分析操作），且创建了较多的告警监控规则，可能导致查询并发数超过Project限制从而使监控失败。建议在创建告警监控规则时，设置**独享SQL**为**自动**，以支持更高的并发数。具体操作，请参见创建日志告警监控规则。 使用SQL独享版时，需确保目标Project具备足够的独享SQL CU数。更多信息，请参见开启SQL独享版。
	单个查询和分析结果	• 只使用查询语句时，默认返回100条数据。基于数据条数的触发条件判断可能不准确，此时建议使用COUNT函数进行统计。 • 分析语句默认返回100行结果。如果您需要更多数据，可使用LIMIT语句。 如果一条分析语句的结果超过1000条，难么系统只选取前1000条用于集合操作。 • 当存在三个查询和分析操作且**集合操作**未选择**不合并**时，只选取每个查询和分析结果中的前100条数据。 • 当存在两个及以上的查询分析操作且**集合操作**为**不合并**时，如果开启了**无数据告警**，那么系统只取第一个查询分析语句的结果来判断是否无数据。
	查询和分析的组合个数	1~3个。
	字段值的长度	如果字段的值超过1024个字符，只截取前1024个字符用于分析。
	查询和分析的时间范围	每条查询和分析语句的时间跨度不能超过24小时。
	资源数据更新时效	资源数据更新非立即生效。生效时间在15分钟以内。
	告警策略变化等待	变化等待最小值为15秒，即使设置为更小的值，也是以15秒间隔进行检查。

类别	限制项	说明
告警管理	策略匹配条件	告警策略、行动策略等配置中，建议使用项目名、告警规则ID、告警名称、严重度、简短的标签或标注等作为条件。 • 如果匹配的是字符串，建议使用简短的普通字符串，例如 `foobar` 。 不支持匹配换行、双引号（""），例如 `foo "bar"` 无法被正常解析。 • 正则匹配不支持Glob表达式，例如 `*Error` 为Glob表达式， `.*Error` 为正则表达式。
	事务数量	30天内最多保留1000条事务，超过后自动覆盖旧的事务数据。
	事务注释	每个事务最多添加10条注释，超出后自动覆盖。
	策略配置更新时效	告警相关的策略配置，例如告警策略、行动策略、内容模板、用户、用户组、值班组等，更新后一般1分钟左右生效。

类别	限制项	说明
	通知渠道	各个通知渠道的使用限制如下所示。超出限制，可能导致您无法接收到告警通知。未收到告警通知时，您可在全局告警排障中心查看相关错误。更多信息，请参见全局告警排障中心。 • 语音 仅支持中国内地手机号码（+86）。 说明 ○ 告警语音未拨通时，不会重复拨打，将以短信方式发送一次告警通知。 ○ 无论告警语音是否拨通均按一次计费。未拨通的提示短信，不会额外产生短信费用。 • 钉钉 钉钉机器人限制每分钟最多20条消息。 • 企业微信 企业微信机器人限制每分钟最多20条消息。 • 飞书 ○ 飞书机器人限制每分钟最多20条消息。 ○ **提醒方式**仅支持设置为**不提醒**或者**所有人**，不支持设置为**指定成员**。 • 自定义Webhook ○ 必须为公网可访问的地址。 ○ Webhook调用返回的状态码为200时，表示调用成功，其余状态码都表示调用失败。 • 函数计算 仅支持以 `sls-ops-` 开头的函数。 更多信息，请参见通知渠道说明。

类别	限制项	说明
通知管理	通知内容	每个通知渠道都存在通知内容长度的限制。为了尽量保证告警通知成功，对于超长的内容，系统可能通过适当的内容截断来避免通知失败。内容截断无法保证内容的完整性以及百分百发送成功，这主要是受限于截断后的内容以及各个通知渠道的支持能力，例如截断后的内容是不合法的Markdown或者HTML，则可能导致通知失败。对于短信、语音等纯文本格式的内容，一般内容截断不会导致通知失败。 建议根据通知渠道的限制合理配置内容模板，避免内容超长导致通知失败。各个通知渠道的限制如下（中文、英文、数字或标点符号都算一个字符）： • 短信 通知内容限制为256个字符。 • 语音 通知内容限制为256个字符。 • 邮件 通知内容限制为8 KB。 • 钉钉 通知内容限制为8 KB。 • 企业微信 ◦ 通知内容限制为4 KB。 ◦ 当**提醒方式**为**所有人**或**指定成员**时，通知内容只能为普通文本格式，不支持Markdown格式。 • 飞书 通知内容限制为8 KB。 • Slack 通知内容限制为8 KB。 • 自定义Webhook 通知内容限制为16 KB。 • 通知中心 通知内容限制为8 KB。 • 函数计算 通知内容限制为16 KB。 • EventBridge 通知内容限制为16 KB。
	内容模板配置	内容模板配置错误时，可能导致模板渲染失败，从而返回报错信息。如果您所接收的告警通知中包含 `Template render error` 类似的报错信息，请根据内容模板语法（新版）及报错信息检查模板配置是否正确。
	内容模板变量	内容长度最多2 KB，超过2 KB部分会被截断。

类别	限制项	说明
	渠道通知额度	每个接收人每天最多可接收9999条邮件、短信或电话。更多信息，请参见配置渠道额度。

告警（旧版）

限制项	说明
组合查询	组合查询个数为1~3个。
字符串	如果日志字段长度超过1024个字符，只截取前1024个字符用于计算。
条件表达式	条件表达式限制说明如下： • 条件表达式长度为1~128个字符。 • 条件表达式只判断每次查询中的前100条查询结果。 • 条件表达式计算次数不超过1000次。
查询区间	每条查询语句的查询时间跨度不能超过24小时。
告警语音通知	告警语音未拨通时，不会重复拨打，将以短信方式发送一次通知。 无论告警语音是否拨通均按一次计费。未拨通的提示短信，不会额外产生短信费用。

7.9. 日志应用

本文介绍日志应用相关的使用限制。

日志审计服务

- 存储方式与地域限制
  - 中心化存储

    从各个阿里云账号、各个地域采集到的日志，会存储到中心账号下的一个中心Project中，目前中心化存储可供选择的地域如下所示。

    > **说明** 当您切换中心账号所在地域时，日志服务为您创建一个新的中心Project，原Project不会被删除。

    - 中国：华北1（青岛）、华北2（北京）、华北5（呼和浩特）、华东1（杭州）、华东2（上海）、华南1（深圳）、中国（香港）
    - 海外：新加坡、日本（东京）、德国（法兰克福）、印度尼西亚（雅加达）
  - 区域化存储

    针对SLB、ALB、OSS、PolarDB-X 1.0访问日志和VPC流日志，日志审计服务支持将各个主账号采集到的日志存储到中心主账号下的各个与SLB、ALB、OSS、PolarDB-X 1.0和VPC实例处于相同地域的日志服务Project中（例如：杭州的OSS访问日志，存储到杭州的日志服务Project中）。

- 同步到中心

  针对SLB、ALB、OSS、PolarDB-X 1.0和VPC的区域化存储，支持将各个地域的Logstore同步到一个中心化的Logstore中，以便做中心化查询、分析、告警、可视化、二次开发等。

  同步机制依赖日志服务数据加工。

- 资源限制
  - 中心主账号下对应的中心化Project只有一个，名为slsaudit-center-*中心化主账号ID-配置的地域*，例如：slsaudit-center-1234567890-cn-beijing。无法通过控制台删除中心化Project，只能通过命令行、API删除。
  - 针对SLB、ALB、OSS、PolarDB-X 1.0和VPC，可以有多个区域化Project，名为slsaudit-region-*中心化主账号ID-各个采集的地域*，例如：slsaudit-region-1234567890-cn-beijing。无法通过控制台删除区域化Project，只能通过命令行、API删除。
  - 配置云产品日志采集后，日志审计服务会创建专属Logstore，具备日志服务Logstore所有的功能，除以下操作限制。
    - 保护数据不被篡改，您无法自行写入数据，修改或删除索引。
    - 只能通过日志审计服务的配置页面或接口修改存储周期、删除Logstore。
    - 针对SLB、ALB、OSS、PolarDB-X 1.0和VPC，如果开启了**同步到中心**功能，在对应的区域化Project中，会生成数据加工作业。
      - 数据加工作业名为Internal Job: SLS Audit Service Data Sync for OSS Access、Internal Job: SLS Audit Service Data Sync for SLB、Internal Job: SLS Audit Service Data Sync for ALB、Internal Job: SLS Audit Service Data Sync for DRDS、Internal Job: SLS Audit Service Data Sync for VPC。
      - 您只能通过日志审计服务的配置页面或接口关闭该数据加工作业。
      - 开启了**同步到中心**功能的区域化Logstore会同步为专属的Logstore，您无法进行任何操作，如果需要进行查询等操作时，可以直接在中心化Logstore中操作。

- 权限限制

  通过日志审计服务采集Kubernetes日志（Kubernetes审计日志、Kubernetes事件中心、Ingress访问日志）时，您需要了解如下权限限制。

  - 日志审计服务仅支持采集中心账号下的Kubernetes日志，不支持采集多账号配置中的其他阿里云账号下的Kubernetes日志。
  - 日志审计服务采集Kubernetes日志依赖数据加工功能。如果您通过日志审计服务采集Kubernetes日志，则中心账号除了完成AliyunServiceRoleForSLSAudit授权外，还需具备sls-audit-service-monitor角色，且该角色具备AliyunLogAuditServiceMonitorAccess权限和如下自定义权限（AliyunLogAuditServiceK8sAccess）。具体操作，请参见自定义授权日志采集与同步。

    ```
    {
        "Version": "1",
        "Statement": [
            {
                "Action": "log:*",
                "Resource": [
                    "acs:log:*:*:project/k8s-log-*"
                ],
                "Effect": "Allow"
            }
        ]
    }
    ```

- 存储天数联动说明
  - 日志审计服务中的RDS MySQL审计日志、慢日志和错误日志都存储在同一个Logstore（rds_log）中，如果同时开启采集且设置的存储天数不同，则日志存储天数为开通者的中心化存储天数的最大值。
  - 日志审计服务中的PolarDB MySQL审计日志、慢日志和错误日志都存储在同一个Logstore（polardb_log）中，如果同时开启采集且设置的存储天数不同，则日志存储天数为开通者的中心化存储天数的最大值。
  - 日志审计服务中的云防火墙的互联网边界防火墙流量日志、VPC边界防火墙流量日志都存储在同一个Logstore（cloudfirewall_log）中，如果同时开启采集且设置的存储天数不同，则日志存储天数为开通者的中心化存储天数的最大值。
  - 日志审计服务中的DDoS高防（新BGP）访问日志、DDoS高防（国际）访问日志、DDoS原生防护访问日志都存储在同一个Logstore（ddos_log）中，如果同时开启采集且设置的存储天数不同，则日志存储天数为开通者的中心化存储天数的最大值。
  - 日志审计服务中的K8s审计日志、K8s事件中心都存储在同一个Logstore（k8s_log）中，如果同时开启采集且存储天数不同，则日志存储天数为开通者的中心化存储天数的最大值。

    > **说明** 针对上述存储天数存在联动的日志类型，如果同时开启日志采集及智能冷热分层存储功能，则热存储天数为开通者的热存储天数的最大值；如果开启了日志采集但没有同时开启智能冷热分层存储功能，则默认关闭智能冷热分层存储功能。
    >
    > 例如您开启了RDS MySQL审计日志和错误日志的采集，如果同时开启了两者的冷热分层存储功能，则热存储天数为二者中的最大值；如果您只开启RDS MySQL审计日志的冷热分层存储功能，没有开启RDS MySQL错误日志的冷热分层存储功能，则默认关闭其所在Logstore（rds_log）的冷热分层存储功能。

- 冷热分层存储

  日志审计服务的专属Logstore支持冷热分层存储功能。相对热存储而言，冷存储成本更低，其数据的查询与分析性能有所降低，其余功能（例如告警、可视化、加工、投递等）不受影响。更多信息，请参见智能冷热分层存储。

  > **说明** 目前已支持智能冷热分层存储的审计中心区域为华北1（青岛）、华北2（北京）、华北5（呼和浩特）、华东1（杭州）、华东2（上海）、华南1（深圳）。

  您可以在日志审计服务的**全局配置**页面开启冷热分层存储，其中热存储时间必须大于等于30天且不能超过当前存储天数。例如当前中心化存储时间为180天，开启30天热存储，则日志将在保存30天之后转为冷存储。

8.安全与合规

8.1. 概述

阿里云日志服务具有丰富的安全防护能力，支持服务器端加密、细粒度权限管控、详细服务日志等特性，同时提供日志审计功能，针对主流云产品提供多账号下实时自动化、中心化采集云产品日志并实现审计。日志服务为您提供丰富的数据洞察能力，同时满足您企业数据的安全与合规要求。

合规认证

日志服务已获得以下合规认证：

- ISO9001、ISO20000、ISO27001、ISO27017、ISO27018、ISO22301、ISO27701、ISO29151
- BS10012
- CSA STAR
- 等保三级
- SOC
- C5
- 中国香港金融审计
- 菲律宾金融审计
- MTCS
- OSPAR
- PCI DSS

安全能力

日志服务具有以下安全能力：

特性	说明
访问控制	日志服务提供授权策略、STS临时授权等功能，实现数据的访问控制和管理。
数据加密	日志服务提供服务器端加密能力，并支持基于SSL/TLS的HTTPS加密传输，有效防止数据在云端的潜在安全风险。
数据可靠性	日志服务底层存储采用三副本机制，提供高可靠性。
日志服务监控审计	日志服务提供服务日志功能，并支持云监控，满足您对企业数据的监控审计需求。
云产品日志审计	日志服务提供日志审计功能，针对云产品提供多账号下实时自动化、中心化采集云产品日志，帮助您实现对于云产品的审计需求。

8.2. 访问控制

日志服务提供授权策略、STS临时授权等功能，实现存储资源的访问控制和管理。

基于用户的授权策略RAM Policy

RAM（Resource Access Management）是阿里云提供的资源访问控制服务，RAM Policy是基于用户的授权策略。通过设置RAM Policy，您可以集中管理您的用户（例如员工、系统或应用程序），以及控制用户可以访问您名下哪些资源的权限，例如限制您的用户只拥有对某个Project或Logstore中的某些对象的读权限。

RAM Policy为JSON格式，您可以通过其中的Statement描述授权语义，每条语义包含对Action、Effect、Resource和Condition的描述。您可以根据业务场景设置多条语义，实现灵活的授权策略。

更多信息，请参见访问控制RAM。

STS临时授权

相对于RAM提供的长效控制机制，STS（Security Token Service）提供的是一种临时访问授权。通过STS可以返回临时的AccessKey和Token，这些信息可以直接发给临时用户用来访问日志服务。一般来说，从STS获取的权限会受到更加严格的限制，并且拥有时间限制，因此这些信息泄露之后对于系统的影响也很小。

日志服务可以通过阿里云STS进行临时授权访问。通过STS，您可以为第三方应用或子用户（即用户身份由您自己管理的用户）颁发一个自定义时效和权限的访问凭证。

更多信息，请参见通过STS实现跨账号访问日志服务资源。

8.3. 数据加密

日志服务支持通过密钥管理服务KMS（Key Management Service）对数据进行加密存储，提供数据静态保护能力。日志服务同时支持基于SSL/TLS的HTTPS加密传输，有效防止数据在云端的潜在安全风险。

服务器端加密

日志服务支持如下两种加密类型机制：

- 通过日志服务自带的服务密钥加密

  日志服务为每个Logstore生成独立的数据加密密钥，用于数据加密。该加密密钥永不过期。

  支持的数据加密算法为AES算法（默认）和国密算法M4。

- 通过用户自带密钥（BYOK）加密

  您可以在KMS控制台上创建主密钥CMK，并授权日志服务相应的权限。日志服务调用KMS接口时，使用该CMK创建用于数据加密的密钥。当您的主密钥CMK被删除或禁用后，BYOK密钥失效。

  > 注意 由KMS BYOK生成的主密钥（CMK）失效后，Logstore上的所有读写请求都会失败。

更多信息，请参见数据加密。

基于SSL/TLS的HTTPS加密传输

日志服务支持通过HTTP或HTTPS的方式访问。安全传输层协议（SSL/TLS）用于在两个通信应用程序之间提供保密性和数据完整性。

- Logtail加密传输

  Logtail是日志服务提供的日志采集Agent。为保证您的数据在发送过程中不会被篡改，Logtail会通过HTTPS通道从服务端获取私密Token，并对所有发送日志的数据包进行数据签名，以保障相关安全性。

- SDK加密传输

  为了能让您更高效地使用日志服务，日志服务提供Java、Python、.NET、PHP、C等多种语言SDK。多语言SDK均支持使用HTTPS协议向日志服务读写数据。

8.4. 数据可靠性

日志服务采用三副本机制为您提供高可靠性。

日志服务底层存储采用三副本机制来保证数据的可靠性，即每份数据都有3个副本，副本按照一定的分布式存储算法保存在集群中的不同机器。通过该机制，存储系统确保3个数据副本分布在不同服务器的不同物理磁盘上，单个硬件设备的故障不会造成数据丢失，同时确保3个数据副本之间的数据强一致性。

8.5. 日志服务监控审计

日志服务提供服务日志功能，并支持云监控，满足您对企业数据的监控审计需求。

服务日志

日志服务提供服务日志功能，支持记录Project内的用户操作日志等多种日志数据，并提供多种分析维度的仪表盘。您可以通过多种方式实时掌握日志服务的使用状况、提高运维效率。

当前Project产生的所有日志数据都会被分类保存到特定的Logstore中。默认为您创建以下2个Logstore：

- internal-operation_log：记录操作日志，每条日志对应一次请求。默认保存30天，计费方式与普通Logstore一致。
- internal-diagnostic_log：记录消费组延时、Logtail心跳日志、作业运行日志等，根据topic进行区分。默认保存30天。不产生费用。

更多信息，请参见服务日志。

云监控

您可以通过阿里云云监控服务来监控日志服务的写入流量、总体QPS、服务状态等指标，获取日志服务的使用情况。同时您可以通过创建报警规则，对日志采集、Shard资源使用等异常进行监控。

更多信息，请参见云监控。

8.6. 云产品日志审计

日志服务基于可观测性数据平台能力，为您提供云产品的日志审计能力。

日志审计对应的日志库为只读库，只能写入专属云产品日志数据，不允许通过自行写入或其他方式篡改数据。日志审计支持多账户下实时自动化、中心化采集云产品日志并进行审计，以及支持审计所需的存储、查询及信息汇总。日志审计的功能优势如下：

- 中心化采集
  - 跨账号：支持将多个阿里云账号下的日志采集到一个阿里云账号下的Project中。您可以通过自定义鉴权管理模式或资源目录管理模式（推荐）配置多账号采集。更多信息，请参见配置多账号采集。
  - 一键式采集：一次性配置采集策略后，即可完成跨账号自动实时发现新资源（例如新创建的RDS、SLB、OSS Bucket实例等）并实时采集日志。
  - 中心化存储：将采集到的日志存储到某个地域的中心化Project中，方便后续查询分析、可视化与告警、二次开发等。
- 支持丰富的审计功能
  - 继承日志服务现有的所有功能，包括查询分析、加工、报表、告警、导出等功能，支持审计场景下中心化的审计等需求。
  - 生态开放对接：与开源软件、阿里云大数据产品、第三方SOC软件无缝对接，充分发挥数据价值。

更多信息，请参见日志审计服务。

9.开服地域

地域是指物理的数据中心，Project创建成功后不能更换地域。本文介绍日志服务支持的地域。

日志服务开服的地域、城市、Region ID的对应关系如下表所示。

地域	城市	Region ID
华北1	青岛	cn-qingdao
华北2	北京	cn-beijing
华北3	张家口	cn-zhangjiakou
华北5	呼和浩特	cn-huhehaote
华北6	乌兰察布	cn-wulanchabu
华东1	杭州	cn-hangzhou
华东2	上海	cn-shanghai
华东5	南京（本地地域）	cn-nanjing
华南1	深圳	cn-shenzhen
华南2	河源	cn-heyuan
华南3	广州	cn-guangzhou
西南1	成都	cn-chengdu
中国香港	香港	cn-hongkong
亚太东南1	新加坡	ap-southeast-1
亚太东南2	悉尼	ap-southeast-2
亚太东南3	吉隆坡	ap-southeast-3
亚太东南5	雅加达	ap-southeast-5
亚太东南6	马尼拉	ap-southeast-6
亚太东南7	曼谷	ap-southeast-7
亚太南部1	孟买	ap-south-1
亚太东北1	东京	ap-northeast-1
亚太东北2	首尔	ap-northeast-2
美国西部1	硅谷	us-west-1

地域	城市	Region ID
美国东部1	弗吉尼亚	us-east-1
欧洲中部1	法兰克福	eu-central-1
欧洲西部1	伦敦	eu-west-1
中东东部1	迪拜	me-east-1
俄罗斯西部1	莫斯科	rus-west-1

10.常见问题

本文介绍使用日志服务前的常见问题。

什么是日志服务？

日志服务SLS是云原生观测与分析平台，为Log、Metric、Trace等数据提供大规模、低成本、实时的平台化服务。日志服务一站式提供数据采集、加工、查询与分析、可视化、告警、消费与投递等功能，全面提升您在研发、运维、运营、安全等场景的数字化能力。

日志服务可以做什么？

- 存储与分析：存储与分析基础设施日志、应用日志、网站访问日志、容器日志、云产品日志、时序数据和链路数据等。
- 智能运维：提供智能聚类、异常检测和异常预测能力。
- 告警监控：监控应用业务、用户体验等数据，具备可视化分析与告警能力。
- 问题排查：监控基础设施，提高问题定位排查效率。
- 日志审计：支持多账号日志的管理与审计。
- 统一存储中台：解决可观测后端存储异构、不稳定问题，支持对接流计算平台。
- 数据处理：通过数据加工功能清洗、脱敏、富化数据；通过数据投递功能将数据投递到数据仓库、数据湖中。

日志服务能为用户带来哪些价值？

- 提供全面的数据接入方案：具备Log、Metric、Trace数据的接入能力，全面覆盖IoT、移动端和服务端。支持接入云产品日志、开源系统日志、多云环境日志、本地服务器日志。
- 提供一站式平台服务：支持采集、分析、加工、可视化、投递、告警等一站式的数据生命周期管理功能。
- 具备智能、高效的数据分析能力：秒级分析百亿级数据能力、智能运维（AIOps）能力、智能异常检测与根因分析能力。
- 提供弹性、低成本的云服务：日志服务是全托管免运维的云服务，具备每天PB级别的弹性伸缩能力。支持按量付费，您仅需为实际用量付费，TCO（Total Cost of Ownership）降低50%以上。

日志服务支持采集哪些数据？

日志服务支持采集如下数据：

- 日志（Log）
- 时序数据（Metric）
- 链路数据（Trace）

日志服务支持哪些数据接入方式？

日志服务提供50多种接入方案，例如使用Logtail采集文本日志、使用Logtail插件采集移动端数据、通过标准协议（Kafka、Syslog等）上传日志、接入阿里云产品日志等。更多信息，请参见接入日志、接入时序数据、接入Trace数据。

阿里云会使用我在日志服务上存储的数据吗？

就用户业务数据，阿里云除执行您的服务要求或者法律法规要求外，不进行任何未获授权的使用及披露。更多信息，请参见服务条款。

阿里云是否会将自己的数据存储在日志服务上？

是的。日志服务是阿里巴巴内部自用的日志及监控平台，经历多次双十一的考验。阿里云内部的开发人员也在很多项目中使用日志服务。

如果数据量突然激增，日志服务如何保证服务不受影响？

日志服务提供弹性伸缩、灵活适配的数据基础框架，具备每天PB级别的弹性伸缩能力。无论是流量高峰还是业务增长都能轻松应对。

日志服务中的数据如何实现低成本存储？

您可以通过如下方式实现数据的低成本存储。

- 通过日志服务Scheduled SQL功能对全量、细粒度的数据进行聚合存储，汇总为存储大小、精度适合的数据，长期存储。更多信息，请参见Scheduled SQL。
- 通过日志服务投递功能将数据投递到阿里云OSS中进行低成本存储。OSS提供标准、低频访问、归档、冷归档四种存储类型，全面覆盖从热到冷的各种数据存储场景。

日志服务中的数据可以保存多久？

日志服务支持永久保存您的数据，您也可以根据自己的业务需求设置数据保存时间。

如何开始使用日志服务？

在使用阿里云日志服务之前，请确保您已经注册了阿里云账号并完成实名认证。如果您还没有创建阿里云账号，系统会在您开通日志服务时提示您注册账号。具体操作，请参见注册账号。

准备好阿里云账号后，打开日志服务产品详情页，单击**立即开通/登录**。

开通日志服务后，默认的计费方式为按量计费，如果想进一步降低日志服务费用，建议您购买资源包。更多信息，请参见产品定价。

日志服务如何收费？

日志服务的所有计费项都是单独计费的，例如您存储日志会产生存储费用，您采集日志会产生写流量费用等。更多信息，请参见计费项。

如果有需求，如何联系？

如果您有日志服务相关的需求，可通过需求单或工单联系日志服务团队。

11.客户案例

11.1. 畅捷通

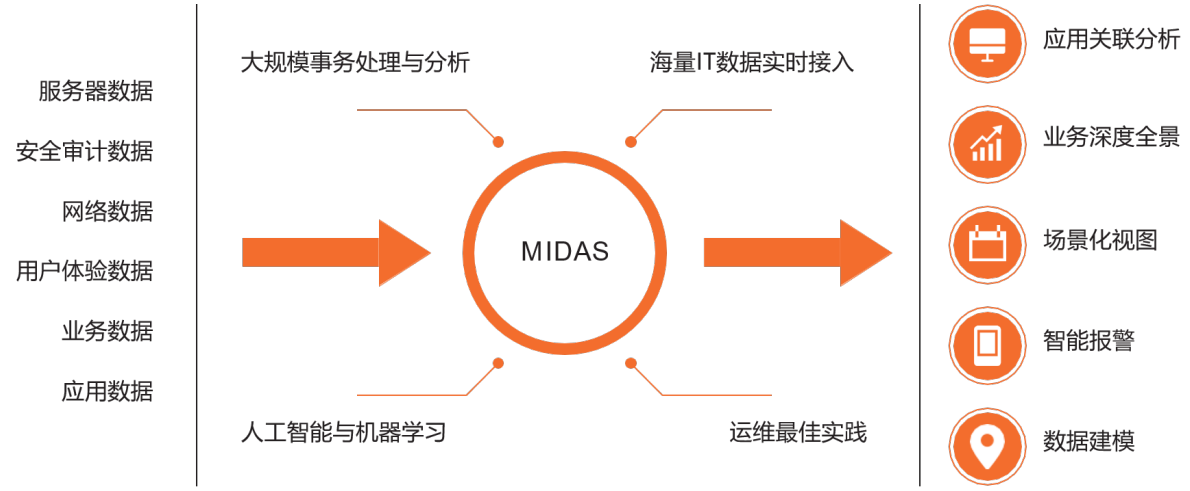
日志服务帮助畅捷通运维开发团队解决了误报频繁、 无法快速发现问题站点、无法快速定位异常的问题，实现了运维效率、运维成本、沟通成本等方面的改善。日志服务支撑了畅捷通所有云产品的健康稳定运行，在IT运维开发领域树立了一个标杆。

公司简介

畅捷通信息技术股份有限公司是用友旗下成员企业。畅捷通致力于为小微企业提供社交化、个性化、服务化、小量化的生意管理支持。畅捷通针对小微企业财务及管理转型问题，通过技术赋能，助力企业业务在线，改变传统的经营业态，实现利润持续增长。畅捷通充分利用SaaS业务与客户的高频互动的优势深挖客户的价值，从而多方面满足小微企业对云产品的需求。畅捷通未来业务将从SaaS市场拓展到企业业务运营服务的BaaS市场，并致力于成为中国最大的一站式小微企业服务平台。更多信息，请参见畅捷通。

业务场景

畅捷通IT运维开发部负责畅捷通所有云产品（包括好会计、好生意、易代账等）的生产及测试系统的运维、上线、发布等工作。该部门构建了一套MIDAS智能运维平台，提供了数据接入、数据处理、场景化分析等能力。



业务痛点

畅捷通在智能运维平台开发初期，底层使用了自建的ELK进行运维数据分析。随着畅捷通业务的增长接入的应用系统增多，畅捷通很快发现平台出现各种问题，各产品的稳定运行受到极大挑战。

- 并发量大

  几万个点并发发送数据，每天产生的各种日志与消息达到TB级。自建的ELK系统性能较差，优化性能需要耗费大量开发资源。

- 类型杂

  访问类、系统类、应用类、通知类、消息类等等，种类繁多、格式千奇百怪，为数据清洗增加了巨大的难度。

- 来源多

  网络、服务器、移动App、Web、Docker等各种来源的日志，接口繁多，并且要求实时性高，无法集中统一管理。

- 应用深入

  各产品部门对收集来的数据都有着自己个性化的需求，监控报警、问题诊断、分析挖掘、报表等，消费模式也多种多样。

解决方案

针对这些问题与调整，畅捷通选择日志服务作为基础来深度打造其智能运维平台。

- 高效消息采集和传输

  畅捷通利用日志服务的强大数据接入能力，将其混合云架构中网络、服务器、移动端、容器的各类访问类、系统类、应用类、消息类等各类日志统一汇入日志服务，实现每天TB级数据的快速处理。

- 灵活的数据处理和存储

  针对内部已经具备完善CMDB和关联规则的情况，畅捷通将原始日志进行语义切分和序列化后，对应到场景分析中。畅捷通在策略组中找到相应的执行策略，再发到外部服务中，用外部服务去调用Ansible或者消息转发等操作，实现数据投递的集中管理，为后续众多场景化分析提供支撑。

- 智能异常检测和定位

  畅捷通通过日志服务的时序数据分析与函数计算能力构建了智能运维平台，通过直接使用同环比函数，可以快速的得出监控指标的当前值，并且具有实时性。有了同环比后，报警的发送会变得准确，与原来的阈值相比准确性大大提高。畅捷通通过日志服务的异常预测函数，从海量指标中快速定位异常，将有问题的地方显示出来，快速发现系统故障。畅捷通通过日志服务将各块汇集过来的数据进行标记后，与应用的配置信息进行关联和整合，通过时序发现故障的根因，从而可以实现故障预测。

畅捷通基于日志服务打造的智能运维平台的架构如下图所示。

11.2. 米哈游

日志服务从内测期便伴随米哈游《原神》团队一同成长，从测试，到公测，到正式上线发布，到积累千万级用户。日志服务一如既往的高性能、高稳定得到了米哈游的广泛认可与赞扬。

公司简介

米哈游成立于2012年，业务主要集中在国产动漫文化下的移动游戏、漫画等领域。作为研发商，米哈游陆续推出了《崩坏学院》、《崩坏学院2》、《崩坏3》等国产动漫移动游戏领域的优秀作品，广受用户喜爱。米哈游于2020年重磅发布首个主打开放世界的二次元手游《原神》。上线仅一个月，《原神》移动端已成为同期全球收入最高的手游。更多信息，请参见米哈游。

业务场景

《原神》是由米哈游自研的一款全新开放世界冒险游戏，于2020年9月28日开启公测。《原神》一经上线，即引爆中国和海外的游戏市场，成为2020年的现象级手游。《原神》作为一款上线时间短，用户规模大，用户群体涵盖海内外的热门游戏， 需要有一个稳定、弹性、高性能的日志平台来满足运营团队日益增长的数据分析需求，以实现《原神》的精细化运营。

业务痛点

《原神》上线后面临的挑战主要表现为以下方面：

- 数据增长

  《原神》上线后，在短短一个月内，月活跃用户规模已经达到千万级别。随着用户数量的快速增长，以及游戏运营体系的逐步精细化，客户业务数据量也在快速的增长。如何实现大规模数据场景下的高性能查询分析成为《原神》团队迫切需要解决的问题。

- 集中采集

  除在中国市场广受欢迎外，《原神》在海外市场同样火爆，荣登美国、德国、加拿大、韩国、新加坡等国家APP Store畅销榜榜首。然而全球化也带来了数据日志集中采集困难，可靠性低的问题。

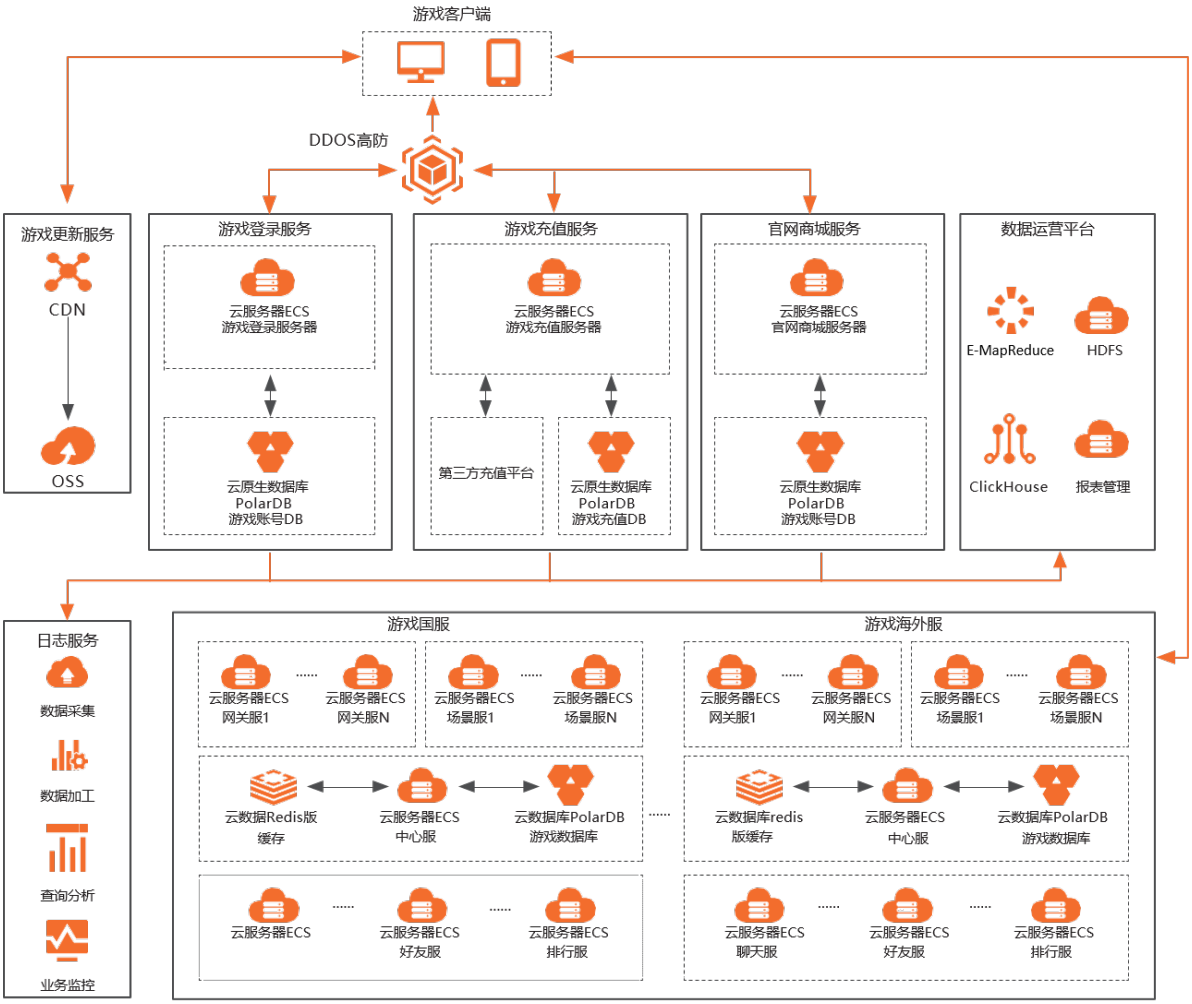
- 突增流量

  作为一款开放世界冒险游戏，《原神》结合游戏地图内每个地区不同的时节、文化背景创作了丰富的活动内容，同时假期活动及版本更新也将带来流量的突增。因此《原神》便需要有一款高弹性的日志平台来帮助产品实现流量突发期的快速扩容。

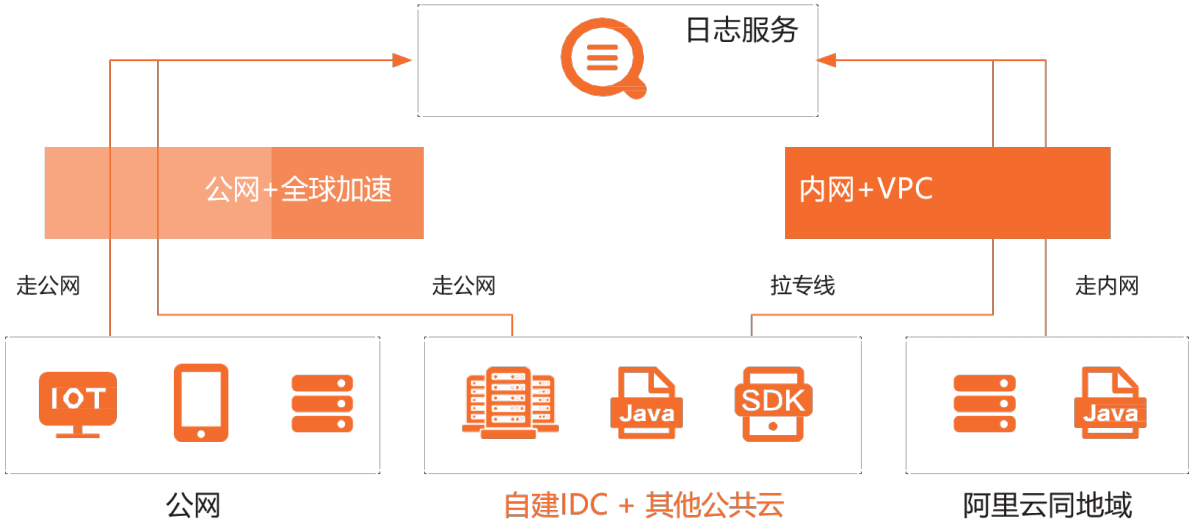
解决方案

针对《原神》业务监控平台对于性能、采集、弹性能力的诉求，阿里云提供了日志服务作为解决方案帮助客户解决上线后的数据采集、查询、监控等问题。

- 在性能方面，日志服务支持秒级处理十亿级数据，支持几十PB吞吐量，稳定性SLA达到99.95%。日志服务能够提供各种异构数据提取、聚合、可视化、告警、AI异常检测等功能，满足《原神》实际业务场景下需要采集分析包括业务服务监控日志，云产品运行及审计日志，游戏运营指标等多种维度数据需求。



- 在数据采集方面，日志服务支持客户端、网页、协议、SDK、API等50多种采集方式，支持断点续传以保证数据采集可靠性，根据不同业务环境支持内网+VPC、公网+全球加速等多种传输方式。日志服务帮助《原神》高效可靠地采集来自不同区域服务日志，实现统一管理。



- 在弹性能力方面，日志服务支持PB级每天的规模线性扩展，可根据流量弹性伸缩。日志服务很好地适应《原神》由于各类活动带来突发流量，克服了大规模流量突发带来的不稳定性。

11.3. 沙盒网络

日志服务帮助沙盒网络解决了多云部署全球化业务场景下数据集中采集统一管理问题，整体异常问题排查时间缩短30%，有效助力业务高速成长。

公司简介

沙盒网络致力于打造一个全球范围内的游戏UGC平台，帮助普通玩家将创意转变为游戏。当前沙盒网络以Blockman GO为代表的游戏已在线发行，并在全球获得了5000万用户的欢迎。更多信息，请参见沙盒网络。

业务场景

沙盒网络作为一款知名的游戏UGC平台，目前已经在全球发行多款游戏，并积累了5000万用户。随着自研游戏以及用户投稿的游戏产品不断上线，沙盒网络的业务规模仍在持续扩大。在此过程中，也对客户的业务平台提出了进一步精细化运维及运营的要求，需要在技术上打通上下游数据，通过多维度的系统监控体系进一步提高平台稳定性保障用户体验，通过用户行为数据分析来反哺产品的迭代优化。

业务痛点

沙盒网络面临的主要挑战如下：

- 全球业务多云部署场景下日志集中采集成为难点

  沙盒网络业务具有全球化特性，除中国集群外，分别在印度、中国香港等地均有部署业务集群。同时沙盒网络采用的是多云部署的方式，跨境以及跨云的架构带来了数据难以统一集中采集的问题。

- 多元化的数据来源导致统一管理困难

  沙盒网络的业务数据，除各云平台的系统数据外，也包括了各个游戏产品的用户行为数据，涉及到iOS、Android等不同客户端的日志。多元化的数据来源、不同的数据格式，导致了数据的统一管理困难。

- UGC游戏平台的业务特性要求具备快速的弹性能力

  新游戏发布、存量游戏活动期间、爆款诞生都可能带来流量高峰。作为游戏UGC平台，沙盒网络需要具备快速弹性能力以确保其整体业务的稳定性。

解决方案

针对沙盒网络对于采集、统一管理、弹性能力的诉求，阿里云提供了日志服务作为解决方案。

- 在数据集中采集方面，日志服务提供50+数据接入手段，其中Logtail采集方式可用于统一采集阿里云ECS、自建IDC、其他云厂商等服务器上的日志。在沙盒网络多云的部署场景下，可以通过Logtail作为统一的采集方式，并以日志服务作为统一的数据分析平台。同时日志服务根据不同业务环境支持内网+VPC、公网+全球加速等多种数据传输方式，能够很好地适应沙盒网络全球化的业务数据采集场景。
- 在多元化数据统一管理方面，日志服务提供可托管、高可用、可扩展的数据加工服务，广泛适用于数据的规整、富化、分发、汇总、重建索引等场景。日志服务可以帮助沙盒网络对于多元的数据进行统一的加工规整，便于后续投入分析应用。同时对于多云场景，日志服务提供在企业平台内嵌日志服务控制台内部管控能力可以进行免权限访问，规避跨云多账号问题。另外日志服务提供丰富的仪表盘能力，帮助沙盒网络不同业务团队根据不同的需求进行数据可视化。
- 在弹性能力方面，日志服务可随时随地进行弹性扩容，最大能够达到PB级弹性能力。无论对于沙盒网络的流量高峰还是业务增长场景，日志服务均可以完美应对。

11.4. 米连科技

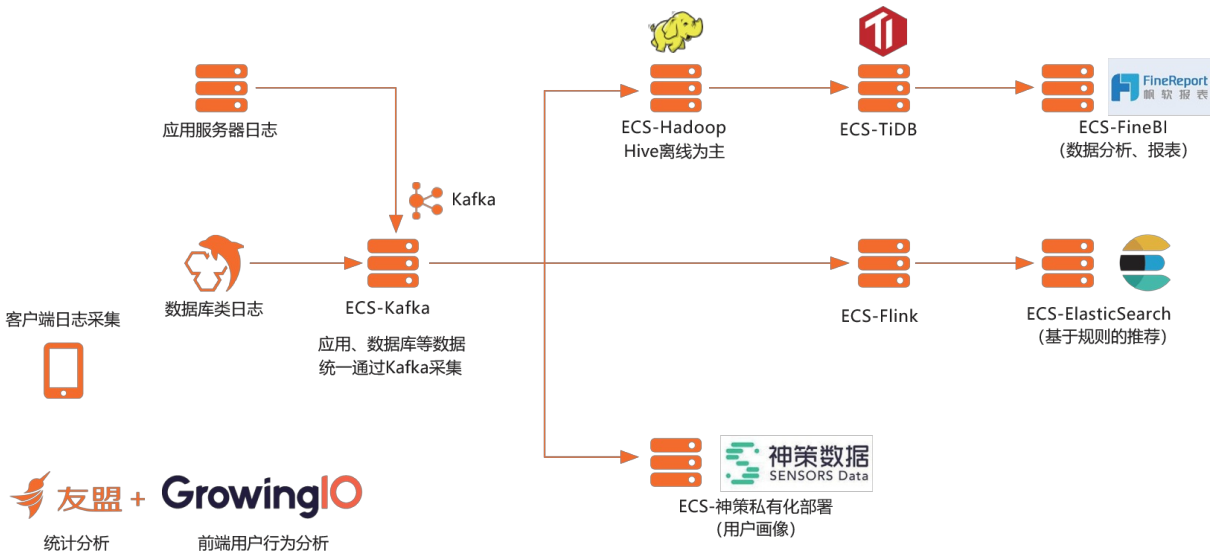
日志服务帮助米连科技解决了数据分散、问题排查效率低、数据分析手段少的问题，提升了IT运维、数据运营、风控等方面的能力。

公司简介

伊对是北京米连科技有限公司旗下品牌，公司成立于2015年，是国家高新技术企业和北京中关村高新技术企业。2019年公司营收近10亿人民币。2020年公司完成多家机构参与的近1亿美元的B轮投融资。伊对专注于移动端线上交友和相亲，将视频、直播和线上红娘创造性地融入该领域，开辟了视频恋爱社区的独立赛道，为单身人群提供了全新的社交体验，成为2019年互联网细分领域的亮点。更多信息，请参见伊对。

业务场景

伊对以恋爱为目的，提供实时视频互动和相亲场景，用户可以通过伊对认识喜欢的人，发送文字信息、语音、照片以及视频实时互动进行恋爱相亲。伊对会给新用户推荐最匹配的直播间，提供搜索功能，让男女双方都可以快速找到自己喜欢的人。在初期，伊对采用了ES和MySQL提供推荐和搜索业务。但是随着业务发展、架构的升级、数据量的增长，伊对需要寻找更强大的数据采集、处理和分析平台来满足运营团队日益增长的数据分析需求，保持伊对高速的用户增长率。



业务痛点

伊对面临的主要挑战如下：

- 数据来源分散

  客户使用不同的计算存储引擎，包括数据库类、大数据类、第三方服务等，需要统一规划和管理，避免产生数据孤岛；并且希望进一步提升开发和管理效率。

- 业务量迅猛增长

  随着业务和用户规模的提升，尤其是直播间相亲活动数量的成倍增长，系统复杂度和日志量也迅速地增长。但是由于自建的ES平台在高业务量下查询变的非常缓慢， 当出现系统问题时故障排查效率较低，用户体验无法得到充分和及时的保障。

- 数据分析能力缺乏

  伊对终坚持以数据驱动产品运营，从最早的统计报表类需求，逐步扩展到基于算法的推荐、风控、运营交互式查询、用户行为分析等领域，但是与之对应的数据能力较为薄弱。

解决方案

针对数据来源分散、业务量迅猛增长、数据分析能力缺乏等挑战，阿里云提供了日志服务作为解决方案。

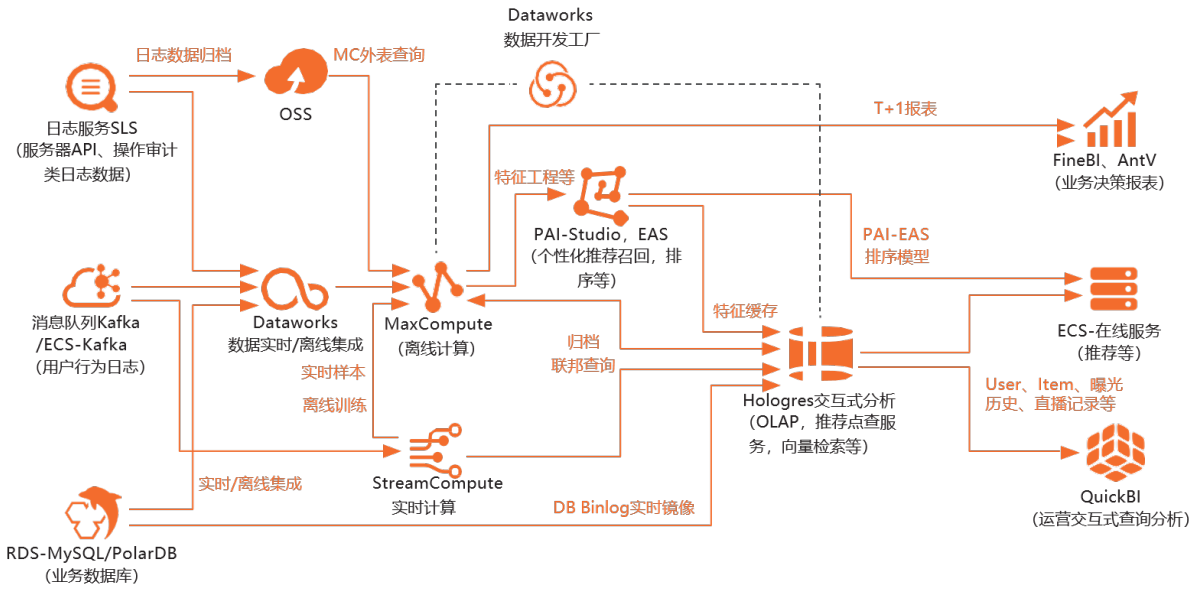
- 统一日志采集
  - Web前端日志：通过日志服务WebTracking方式采集上报到日志服务。
  - APP前端日志：通过日志服务iOS和Android SDK采集上报到日志服务。
  - 服务端日志：通过日志服务Logtail采集方式上报到日志服务。
- 统一智能运维平台

  通过日志服务，客户构建了统一的智能运维平台。通过对各应用系统、服务器、数据库及网络安全产品等服务的访问日志做统一的采集，并利用日志服务的秒级查询、日志聚类、AI异常检测能力、多种告警方式，伊对构建了异常事件的快速分析与响应平台，确保了线上系统的稳定。以日志服务为基础的伊对智能运维平台，也应用在用户体验改善上，通过多维度的指标分析和图形化展现，为持续改善用户体验提供了数据基础。

- 统一数据服务

  通过以阿里云日志服务采集上来的数据为基础，结合离线计算、实时计算引擎和PAI机器学习平台，为客户提供统一的数据分析服务。

- API网关：作为统一数据服务出口。
- Quick BI：交互式报表制作，拖拽形式快速制作各种报表。
- DataV：固定格式的数据大屏。



11.5. 哈啰出行

哈啰出行通过把日志数据迁移到日志服务，替代原有的Kafka、ES、ClickHouse，累积节省成本30%，同时满足了稳定性、扩展性以及日志查询与分析的需求。

公司简介

哈啰出行是本地出行及生活服务平台，致力于应用数字技术的红利，为人们提供更便捷的出行以及更好的普惠生活服务。更多信息，请参见哈啰出行。

业务场景

哈啰出行为用户提供哈啰单车、哈啰助力车分时租赁的服务。共享单车服务致力于解决最后一公里的出行难题。哈啰出行以技术创新赋能智能终端，推动运维高效执行与自营管理体系相结合，依托搭载定位芯片的智能锁，辅以后台智能规划调度、运维人员智能端口精细化运营。哈啰单车累积注册4亿多用户，入驻400多座城市，累积骑行237亿公里。依托于智能锁，赋能了在线的实时调度。单车数据、APP数据无缝打通，因而催生了数据的实时采集、分析、存储需求。

业务痛点

哈啰出行原有架构是将数据采集到Kafka，然后将日志写入ELK做查询，同时写入ClickHouse做分析。由于每天增量数据在TB级别，对ES稳定性压力比较大。当查询数据操作，会影响ES的写入延时。由于写入量大，查询基本处于不可用状态。因此，当天数据采用单副本，隔天再生成多副本。这种方式对数据的可靠性带来很大的挑战。此外，自建Kafka、ES、ClickHouse成本较高，急需降低成本。

解决方案

日志服务为哈啰出行提供了TB级别日志的实时采集、弹性扩容、实时查询的能力。

- 实时采集

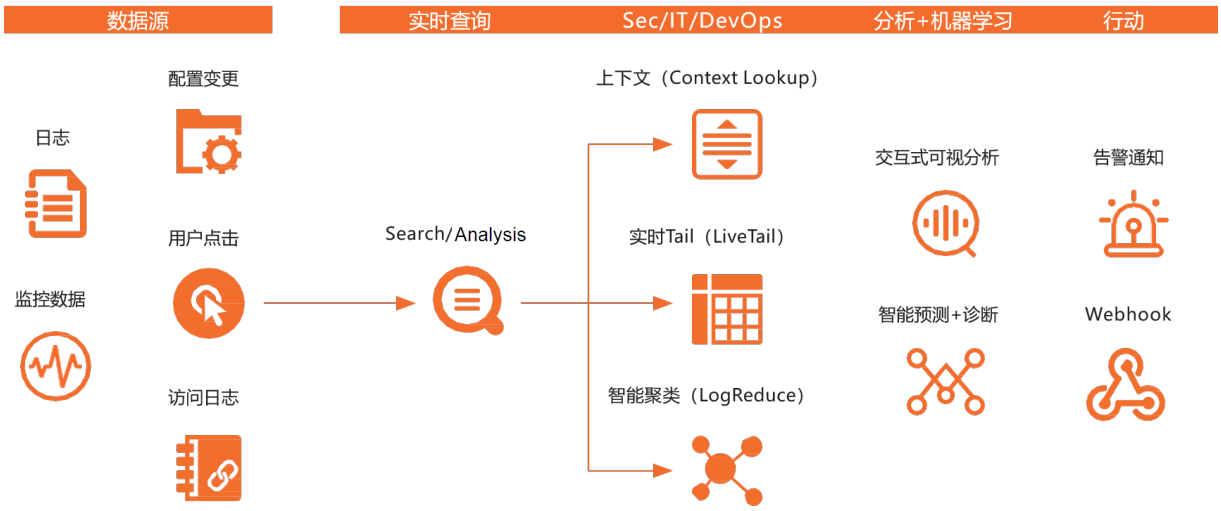
日志服务原生支持Kafka协议。哈啰出行的各个客户端只需把Kafka的地址设置成日志服务的Kafka协议地址即实现了无缝迁移。

- 弹性扩容

  日志服务采用Shard模型，当流量发生上涨时，可以手动分裂Shard，实现写入带宽的扩容，也可以设置成自动分裂，当流量达到上限时，自动扩容出新的Shard。

- 查询与分析

  日志服务同时提供了查询和分析能力。在查询方面，日志服务支持关键字检索、数值范围查询、JSON字段的递归查询、多条件组合查询。在分析方面，日志服务支持以SQL 92语法分析日志，秒级分析数百亿条日志。SQL语法支持200多种函数，以及支持join计算，可与OSS、MySQL数据源做关联分析。



12.竞品对比
12.1. 成本优势

成本优势

日志服务产品在日志处理的三种场景下具有以下成本优势：

- LogHub：
  - 与购买云主机 + 云磁盘搭建 Kafka 相比，对于 98% 场景下用户价格有优势。对小型网站而言，成本为 kafka 的30% 以下。
  - 提供 RESTful API，可以直接针对移动设备提供数据收集功能，节省了日志收集网关服务器的费用。
  - 免运维，随时随地弹性扩容使用。
- LogShipper：
  - 无需任何代码/机器资源，灵活配置与丰富监控数据。
  - 规模线性扩展 （PB级/Day），功能当前免费。
- LogSearch/Analytics：
  - 与购买云主机 + 自建 ELK 相比，成本为自建的 15% 以下，并且查询能力与数据规模有极大提升。与日志管理软件相比，能无缝支持各种流行流计算 + 离线计算框架，日志流动畅通无阻。

成本对比

以下是在计费模型下，日志服务功能与自建方案的对比，仅供参考。

日志中枢（LogHub vs Kafka）

-	关注点	LogHub	自建中间件（如Kafka）
使用	新增	无感知	需要运维动作
	扩容	无感知	需要运维动作
	增加备份	无感知	需要运维动作
	多租户使用	隔离	可能会相互影响
费用	公网采集（10GB/天）	2 元/天	16.1 元/天
	公网采集（1TB/天）	162 元/天	800 元/天
	内网采集（数据量小）	-	-
	内网采集（数据量中）	-	-
	内网采集（数据量大）	-	-

日志存储与查询引擎

关注点		LogSearch	ES（Lucene Based）	NoSQL	Hive
规模	规模	PB	TB	PB	PB
成本	存储（元/GB*天）	0.0115	3.6	0.02	0.035
	写入（元/GB）	0.35	5	0.4	0
	查询（元/GB）	0	0	0.2	0.3
	速度-查询	毫秒级-秒级	毫秒级-秒级	毫秒级	分钟级
	速度-统计	弱+	较强	弱	强
延时	写入->可查询	实时	分钟级	实时	十分钟级

> **说明**
>
> 此处价格对比主要基于ECS上部署软件，并且设置为3份副本后的计算结果。
>
> 更多内容请参考：查询分析全方位对比（ELK）。

12.2. 查询分析全方位对比（ELK）

通过将阿里云日志服务与ELK Stack进行全面对比，帮助您更好的了解阿里云日志服务的主要功能和优势。

背景信息

提到日志实时分析，很多人都会想到基于ELK Stack（Elastic/Logstash/Kibana）来搭建。ELK方案开源，在社区中有大量的内容和使用案例。

阿里云日志服务是阿里巴巴集团对日志场景的解决方案产品，前身是2012年初阿里云在研发飞天操作系统过程中用来监控和问题诊断的产物，但随着用户增长与产品发展，慢慢开始向面向Ops（DevOps，Market Ops，SecOps）日志分析领域发展，期间经历双十一、蚂蚁双十二、新春红包、国际业务等场景挑战，成为同时服务全球的产品。

面向日志分析场景

Apache Lucene是Apache软件基金会一个开放源代码的全文检索引擎工具包，一个全文检索引擎的架构，提供了完整的查询引擎和索引引擎、部分文本分析引擎。2012年Elastic把Lucene基础库包成了一个更好用的软件，并且在2015年推出ELK Stack（Elastic Logstash Kibana）解决集中式日志采集、存储和查询问题。Lucene设计场景是Information Retrival，面对是Document类型，因此对于Log这种数据有一定限制，例如规模、查询能力、以及智能聚类LogReduce等定制化功能。

Log Service提供日志存储引擎是阿里内部自研究技术，经过3年万级应用锤炼，每日索引数据量达PB级，服务万级开发者每天亿次查询分析。在阿里集团内阿里云全站，SQL审计、鹰眼、蚂蚁云图、飞猪Tracing、阿里云谛听等都选择Log Service作为日志分析引擎。

而日志查询是DevOps最基础需求，业界的调研50 Most Frequently Used Unix Command也验证了这一点，tar排名第一、Grep排名第二，由此可见日志查询对程序员的重要性。

在日志查询分析场景，以如下点对ELK与Log Service做一个全方位比较。

- 易用：上手和使用的便利程度。
- 功能：主要针对查询与分析。
- 性能：对于单位大小数据量查询与分析需求，延时如何。
- 规模：能够承担的数据量、扩展性等。
- 成本：同样功能和性能，使用分别花多少钱。

易用性

对日志分析系统而言，有如下使用过程。

- 采集：将数据稳定写入。
- 配置：如何配置数据源。
- 扩容：接入更多数据源，更多机器，对存储空间，机器进行扩容。
- 使用：这部分在功能这一节介绍。
- 导出：数据能否方便导出到其他系统，例如做流计算、放到对象存储中进行备份。
- 多租户：如何将数据分享给其他人使用，使用是否安全等。

以下是比较结果：

项目	分项	自建ELK	Log Service
采集	协议	RESTful API	• RESTful API • JDBC
	客户端	Logstash、Beats和FluentD，生态十分丰富。	• Logtail • 其他（例如Logstash）
配置	单元	提供Index概念用以区分不同日志。	• Project • Logstore 提供两层概念，Project相当于命名空间，可以在Project下建立多个Logstore。
	属性	AP+Kibana	• API+SDK • 控制台
扩容	存储	• 增加机器 • 购买云盘	无需操作
	计算	新增机器	无需操作
	配置	• 通过配管系统应用机器。 • Logstash在Beta版本中已经提供中心化配置功能。	控制台或API操作，无需配管系统。
	采集点	通过配管系统控制，将配置和Logstash安装到机器组。	控制台或API操作，无需配管系统。
	容量	不支持动态扩容	动态扩容、弹性伸缩。

项目	分项	自建ELK	Log Service
导出	方式	• API • SDK	• API • SDK • 类Kafka接口消费 • 各流计算引擎消费（Spark，Storm，Flink） • 流计算类库消费（Python、Java）
多租户	安全	商业版	• HTTPS • 传输签名 • 多租户隔离 • 访问控制
	流控	无流控	• Project级 • Shard级
	多租户	Kibana支持	原生提供账号与权限级管理。

整体而言：

- ELK有非常多的生态和写入工具、安装、配置等都有较多工具可以使用。
- Log Service是托管服务，从接入、配置、使用上集成度非常高，普通用户5分钟就可以接入。
- Log Service是SaaS化服务，在过程中不需要担心容量、并发等问题。弹性伸缩，免运维。

功能（查询+分析）

查询主要是将符合条件的日志快速命中，分析功能是对数据进行统计与计算。

例如我们需要所有状态码大于200的读请求，根据IP统计次数和流量。这样的分析请求可以转化为两种操作。

- 查询到指定结果，对结果进行统计分析。
- 不进行查询，直接对所有日志进行分析。

```
1. Status in (200,500] and Method:Get* | select count(1) as c, sum(inflow) as sum_inflow, ip group by Ip
2. * | select count(1) as c, sum(inflow) as sum_inflow, ip group by Ip
```

- **查询基础对比**

  该对比基于Elastic 6.5 Indices。

类型	子类	ELK	Log Service
	索引查询	支持	支持

类型	子类	ELK	Log Service
文本	分词	支持	支持
	中文分词	支持	支持
	前缀	支持	支持
	后缀	支持	不涉及
	模糊	支持	可通过SQL支持
	Wildcard	支持	可通过SQL支持
数值	long	支持	支持
	double	支持	支持
Nested	Json	支持	不涉及
Geo	Geo	支持	可通过SQL支持
IP	IP查询	支持	可通过SQL支持

对比结论如下：

- ES支持的数据类型丰富度，原生查询能力比Log Service更完整。
- Log Service能够通过SQL方式（如下）来代替字符串模糊查询，Geo函数等，但性能会比原生查询稍差。

子串命中

```
* | select content where content like '%substring%' limit 100
```

正则表达式匹配

```
* | select content where regexp_like(content, '\d+m') limit 100
```

JSON内容解析与匹配

```
* | select content where json_extract(content, '$.store.book')='mybook' limit 100
```

如果设置JSON类型索引也可以使用：

```
field.store.book='mybook'
```

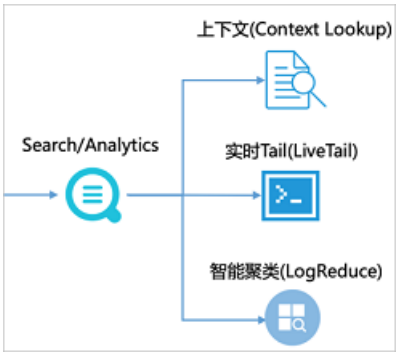
- **查询扩展能力**

  在日志查询场景中，光有检索还不够，需要能够围绕查询做进一步的工作。

  - 定位到错误日志后，想通过上下文查看是什么参数引起了错误。
  - 定位到错误后，想看看之后有没有类似错误，类似tail -f原始日志文件，并进行grep。
  - 通过关键词搜索到大量日志（例如百万条），其中90%都是已知问题干扰调查线索。

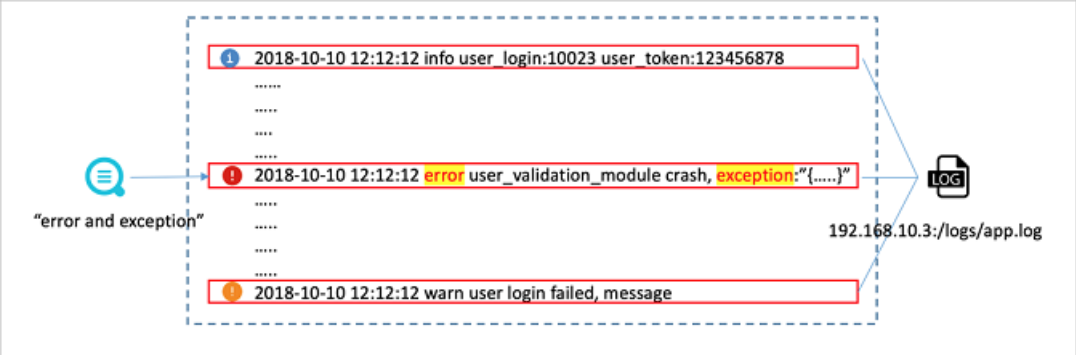
  Log Service针对以上问题提供闭环解决方案。

  - 上下文查询（Context Lookup）：原始上下文翻页，免登服务器。
  - LiveTail功能（Tail-f）：原始上下文tail-f，更新实时情况。
  - 智能聚类（LogReduce）：根据日志Pattern动态归类，合并重复模式，洞察异常。



- LiveTail功能

  在传统的运维方式中，如果需要对日志文件进行实时监控，需要到服务器上对日志文件执行 `tail -f` 命令，如果实时监控的日志信息不够直观，可以加上 `grep` 或者 `grep -v` 进行关键词过滤。Log Service在控制台提供了日志数据实时监控的交互功能LiveTail，针对线上日志进行实时监控分析，减轻运维压力。



  Livetail特点如下：

  - 智能支持Docker、K8s、服务器、Log4J Appender等来源数据。
  - 监控日志的实时信息，标记并过滤关键词。
  - 日志字段做分词处理，以便查询包含分词的上下文日志。

- 智能聚类（LogReduce）

  随着业务的高速发展，各个系统每天产生大量的日志，存在如下担忧：

  - 系统有潜在异常，但被淹没在海量日志中。
  - 机器被入侵，有异常登录，却后知后觉。
  - 新版本上线，系统行为有变化，却无法感知这些问题，是因为信息太多、太杂，不能良好归类，同时记录信息的日志，往往还都是无主题，格式多样，归类难度更大。Log Service提供实时日志**智能聚类（LogReduce）**功能根据日志的相似性进行归类，快速掌握日志全貌。
    - 支持任意格式日志：Log4J、Json、单行（syslog）。
    - 日志经任意条件过滤后再Reduce；对日志Reduce后Pattern，根据signature反查原始数据。
    - 不同时间段Pattern比较。
    - 动态调整Reduce精度。
    - 亿级数据，秒级出结果。

分析能力对比

ES在docvalue之上提供一层聚合（Aggregation）语法，并且在6.x版本中提供SQL语法能够对数据进行分组聚合运算。 Log Service支持完整SQL92标准（提供RESTful和JDBC两种协议），除基本聚合功能外，支持完整的SQL计算，并支持外部数据源联合查询（Join）、机器学习、模式分析等函数。

> **说明** 以下分析基于ES 6.5 Aggregation和log分析语法。更多信息，请参见ES 6.5 Aggregation和LOG分析语法。

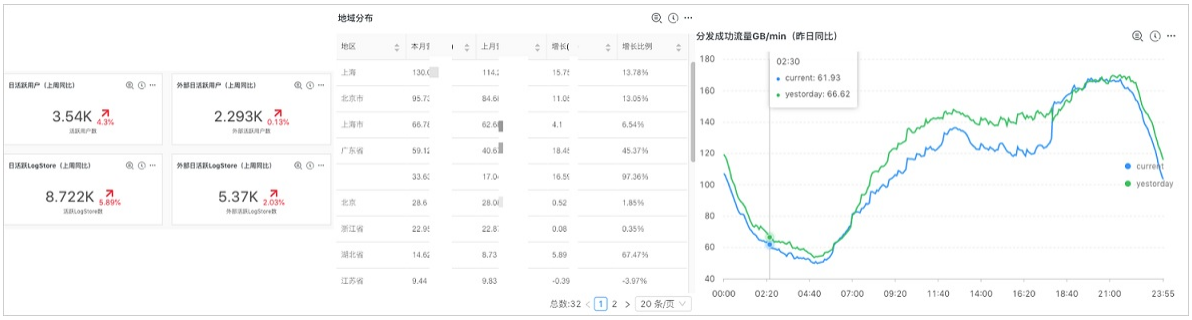
除SQL92标准语法外，我们根据实际日志分析需求，研发一系列实用的功能。

- 同比和环比函数

  同比环比函数能够通过SQL嵌套对任意计算（单值、多值、曲线）计算同环比（任意时段），以便洞察增长趋势。

```
* | select compare( pv , 86400) from (select count(1) as pv from log)
```

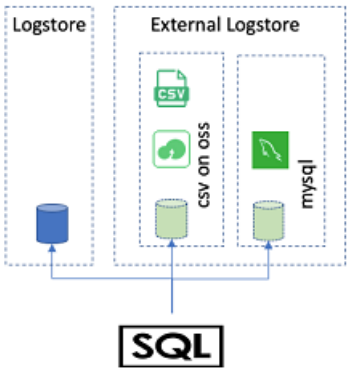
```
*|select t, diff[1] as current, diff[2] as yestoday, diff[3] as percentage from(select t,
compare( pv , 86400) as diff from (select count(1) as pv, date_format(from_unixtime(__tim
e__), '%H:%i') as t from log group by t) group by t order by t) s
```



- 外部数据源联合查询（Join）

  可以在查询分析中关联外部数据源。

- 支持Logstore、MySQL、OSS（CSV）等数据源。
- 支持left、right、full outer join、innerjoin。
- SQL查询外表，SQL Join外表。



Join外表的样例：

```
sql
创建外表：
* | create table user_meta ( userid bigint, nick varchar, gender varchar, province varcha
r, gender varchar,age bigint) with ( endpoint='oss-cn-hangzhou.aliyuncs.com',accessid='LT
A288',accesskey ='EjsowA',bucket='testossconnector',objects=ARRAY['user.csv'],type='oss')
使用外表
* | select u.gender, count(1) from chiji_accesslog l join user_meta1 u on l.userid = u.us
erid group by u.gender
```

- 地理位置函数

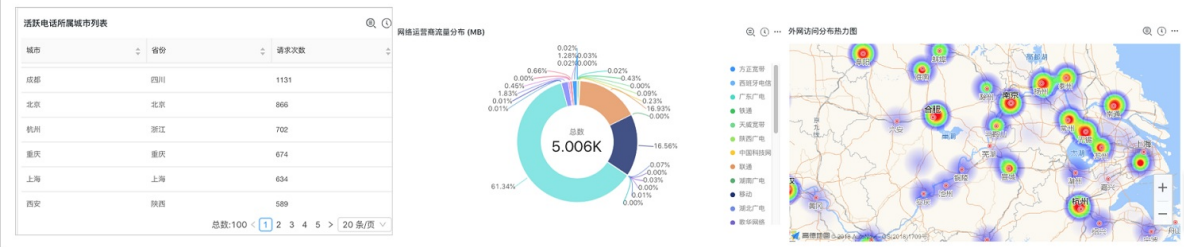
  针对IP地址、手机等内容，内置地理位置函数方便分析用户来源。

  - IP：国家、省市、城市、经纬度、运营商。
  - Mobile：运营商、省市。
  - GeoHash：Geo位置与坐标转换。

  查询结果分析样例：

```
sql
* | SELECT count(1) as pv, ip_to_province(ip) as province WHERE ip_to_domain(ip) != 'intr
anet' GROUP BY province ORDER BY pv desc limit 10
* | SELECT mobile_city(try_cast("mobile" as bigint)) as "城市", mobile_province(try_cast("
mobile" as bigint)) as "省份", count(1) as "请求次数" group by "省份", "城市" order by "请求
次数" desc limit 100
```

查询结果：

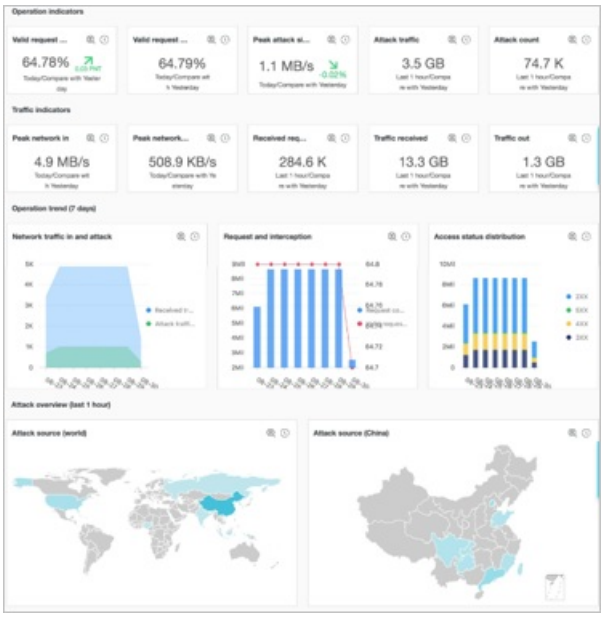


- GeoHash
- 电话号码
- IP函数

- 安全分析函数

  依托全球白帽子共享安全资产库，提供安全检测函数，您只需要将日志中任意的IP、域名或者URL传给安全检测函数，即可检测是否安全。

  - security_check_ip
  - security_check_domain
  - security_check_url



- 机器学习与时序检测函数

  新增机器学习与智能诊断系列函数。

  - 根据历史自动学习其中规律，并对未来的走势做出预测。
  - 实时发现不易察觉的异常变化，并通过分析函数组合推理导致异常的特征。
  - 结合环比、告警功能智能发现/巡检。该功能适用在智能运维、安全、运营等领域，帮助更快、更有效、更智能洞察数据。

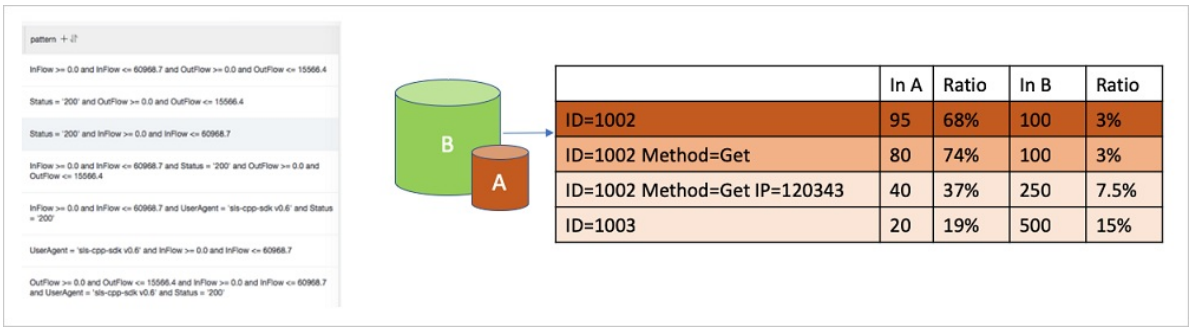
  提供如下功能：

  - 预测：根据历史数据拟合基线。
  - 异常检测、变点检测和折点检测：找到异常点。
  - 多周期检测：发现数据访问中的周期规律。
  - 时序聚类：找到形态不一样的时序。

- 模式分析函数

  模式分析函数能够洞察数据中的特征与规律，帮助快速、准确推断问题。

  - 定位频繁集。例如：错误请求中90%由某个用户ID构成。
  - 定位两个集合中最大支持因素。
    - 延时>10S请求中某个ID构成比例远远大于其他维度组合。
    - 并且该ID在对比集合（B）中的比例较低。
    - A和B中差异明显。



	In A	Ratio	In B	Ratio
ID=1002	95	68%	100	3%
ID=1002 Method=Get	80	74%	100	3%
ID=1002 Method=Get IP=120343	40	37%	250	7.5%
ID=1003	20	19%	500	15%

性能

针对相同数据集，分别对比写入数据及查询，和聚合计算能力。

- 实验环境

- 测试配置

类别	自建ELK	Log Service
环境	ECS 4核16 GB * 4台 + 高效云盘或SSD	无
Shard	10	10
拷贝数	2	3（默认配置，对用户不可见）

- 测试数据
  - 5列double、5列long、5列text，字典大小分别是256、512、768、1024、1280。
  - 以上字段完全随机（测试日志样例如下）。
  - 原始数据大小：50 GB。
  - 日志行数：162640232 （约为1.6亿条）。

  以上字段完全随机，如下为一条测试日志样例：

```
timestamp:August 27th 2017, 21:50:19.000
long_1:756,444 double_1:0 text_1:value_136
long_2:-3,839,872,295 double_2:-11.13 text_2:value_475
long_3:-73,775,372,011,896 double_3:-70,220.163 text_3:value_3
long_4:173,468,492,344,196 double_4:35,123.978 text_4:value_124
long_5:389,467,512,234,496 double_5:-20,10.312 text_5:value_1125
```

- 写入测试结果

  ES采用bulk api批量写入，LogSearch或Analytics用PostLogstoreLogs API批量写入，结果如下。

类型	项目	自建ELK	Log Service
延时	平均写入延时	40 ms	14 ms
存储	单拷贝数据量	86 GB	58 GB
	膨胀率：数据量/原始数据大小	172%	116%

> **说明** 日志服务产生计费的存储量包括压缩的原始数据写入量（23 GB）和索引流量27 GB，共50 GB存储费用。

  从测试结果来看

  - 日志服务写入延时好于ES，40 ms vs 14 ms。
  - 空间：原始数据50 GB，因为测试数据比较随机，所以存储空间会有膨胀（大部分真实场景下，存储因压缩会比原始数据小）。ES胀到86 GB，膨胀率为172%，在存储空间超出日志服务58%。这个数据与ES推荐的存储大小为原始大小2.2倍比较接近。

- 读取（查询+分析）测试

- 测试场景

  选取两种比较常见的场景：日志查询和聚合计算。分别统计并发度为1、5、10时，两种case的平均延时。

  - 针对全量数据，对任意text列计算group by，计算5列数值的avg/min/max/sum/count，并按照count排序，取前1000个结果，例如：

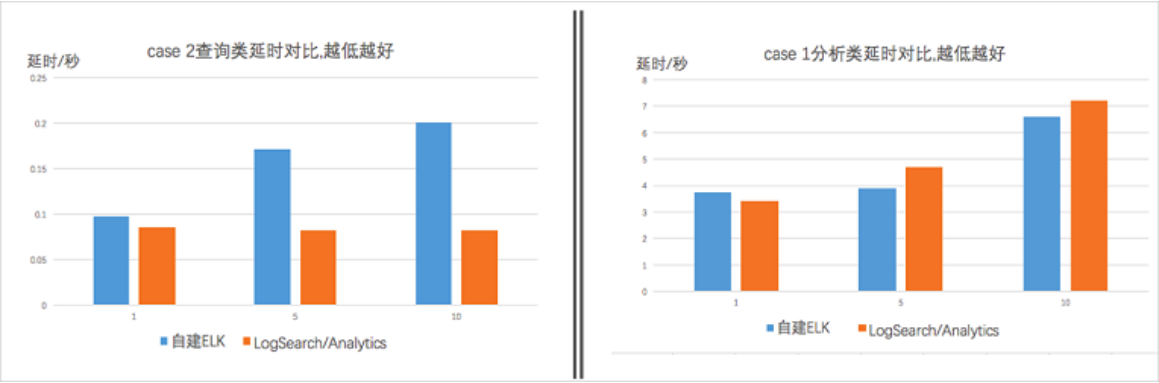
    ```
    select count(long_1) as pv,sum(long_2),min(long_3),max(long_4),sum(long_5)
      group by text_1 order by pv desc limit 1000
    ```

  - 针对全量数据，随机查询日志中的关键词，例如查询value_126，获取命中的日志数目与前100行。

    ```
    value_126
    ```

- 测试结果

类型	并发数	ES延时（单位为秒）	日志服务延时（单位为秒）
case1：分析类	1	3.76	3.4
	5	3.9	4.7
	10	6.6	7.2
case2：查询类	1	0.097	0.086
	5	0.171	0.083
	10	0.2	0.082



- 结果分析
  - 从结果看，对于1.6亿数据量这个规模，两者都达到了秒级查询与分析能力。
  - 针对统计类场景（case 1），ES和日志服务延时处同一量级。ES采用SSD云盘，在读取大量数据时IO优势比较高。
  - 针对查询类场景（case 2），LogAnalytics在延时明显优于ES。随着并发的增加，ELK延时对应增加，而LogAnalytics延时保持稳定甚至略有下降。

规模与成本

- 规模能力

- 日志服务一天可以索引PB级数据，一次查询可以在秒级过几十TB规模数据，在数据规模上可以做到弹性伸缩与水平扩展。
- ES比较适合服务场景为：写入GB-TB/Day、存储在TB级。原因如下：
  - 单集群规模：比较理想为20台左右，据了解业界比较大为100节点一个集群，为了应对业务往往拆成多个集群。
  - 写入扩容：shard创建后便不可再修改，当吞吐率增加时，需要动态扩容节点，最多可使用的节点数便是shard的个数。
  - 存储扩容：主shard达到磁盘的上限时，要么迁移到更大的一块磁盘上，要么只能分配更多的shard。一般做法是创建一个新的索引，指定更多shard，并且rebuild旧的数据。

用户案例（规模带来的问题）

客户A是中国的最大资讯类网站之一，有数千台机器与百号开发人员。运维团队原先负责一套ELK集群用来处理Nginx日志，但始终处于无法大规模使用状态：

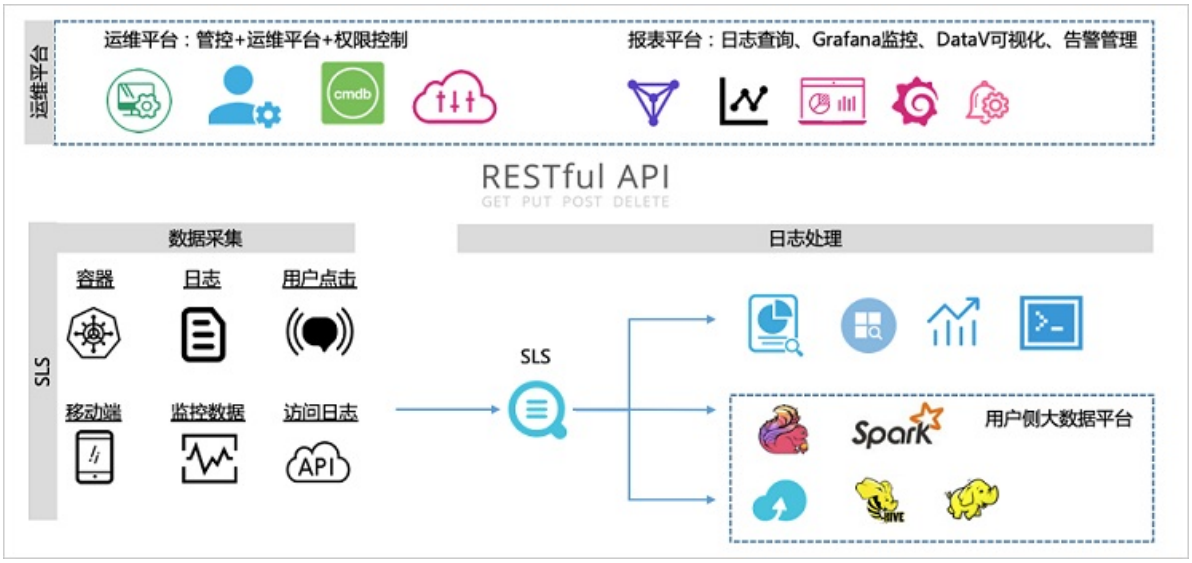
- 一个大Query容易把集群打爆，导致其他用户无法使用。
- 在业务高峰期间，采集与处理能力打满集群，造成数据丢失，查询结果不准确。
- 业务增长到一定规模，因内存设置、心跳同步等节点经常内存失控导致OOM 不能保证可用性与准确性，开发最终没有使用起来，成为一个摆设。

在2018年6月份，运维团队开始运行日志服务方案。

i. 使用Logtail来采集线上日志，将采集配置、机器管理等通过API集成进客户自己运维与管控系统。
ii. 将日志服务查询页面嵌入统一登录与运维平台，进行业务与账户权限隔离。
iii. 通过控制台内嵌方案满足开发查询日志需求，通过Grafana插件调用日志服务统一业务监控，通过DataV连接日志服务进行大盘搭建。

> **说明** 详细说明请参见文档：
>
> - 控制台内嵌及分享
> - 对接Grafana
> - 对接DataV
> - 对接Jaeger

整体架构如下图：



平台上线2个月后：

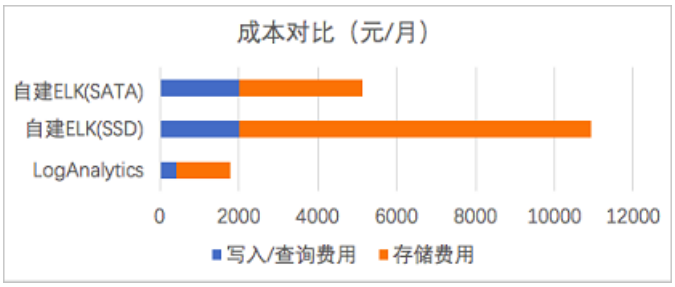
- 每天查询的调用量大幅上升，开发逐步开始习惯在运维平台进行日志查询与分析，提升了研发的效率，运维部门也回收了线上登录的权限。
- 除Nginx日志外，把App日志、移动端日志、容器日志也进行接入，规模是之前10倍。
- 除查询日志外，也衍生出很多新的玩法，例如通过Jaeger插件与控制台基于日志搭建了Trace系统，将线上错误配置成每天的告警与报表进行巡检。
- 通过统一日志接入管理，规范了各平台对接总线，不再有一份数据同时被采集多次的情况，大数据部门Spark、Flink等平台可以直接去订阅实时日志数据进行处理。

- 成本

  以上述测试数据为例，一天写入50 GB数据（其中23 GB为实际的内容），保存90天，平均一个月的耗费。

  - 日志服务（LogSearch/LogAnalytics）计费规则参见按量付费，包括读写流量、索引流量、存储空间等计费项，查询功能免费。

计费项目	值	单价	费用（元）
读写流量	23 GB * 30	0.2元/GB	138
存储空间（保存90天）	50 GB * 90	0.3元/GB*Month	1350
索引流量	27 GB * 30	0.35元/GB	283
总计	无	无	1771



- ES费用包括机器费用，及存储数据SSD云盘费用。
  - 云盘一般可以提供高可靠性，因此我们这里不计费副本存储量。
  - 存储盘一般需要预留15%剩余空间，防止空间写满，因此乘以一个1.15系数。

计费项目	值	单价	费用（元）
服务器	4台4核16GB（三个月）（ecs.mn4.xlarge）	包年包月费用：675元/Month	2025
存储	86 * 1.15 * 90 （这里只计算一个副本）	SSD：1元/GB*M	8901
	无	SATA：0.35元/GB*M	3115.35
总计			10926 （SSD）
			5140.35 （SATA）

同样性能，使用LogSearch/Analytics与ELK（SSD）费用比为13.6%。在测试过程中，我们也尝试把SSD换成SATA以节省费用（LogAnalytics与SATA版费用比为34%），但测试发现延时会从40 ms上升至150 ms，在长时间读写下，查询和读写延时变得很高，无法正常工作了。

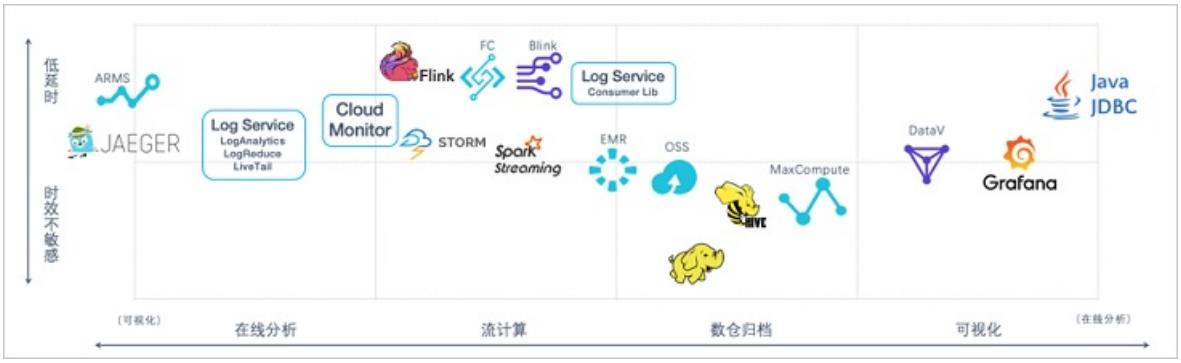
- 时间成本（Time to Value）

  除硬件成本外，日志服务在新数据接入、搭建新业务、维护与资源扩容成本基本为0。

  - 支持各种日志处理生态，可以和Spark、Hadoop、Flink、Grafana等系统无缝对接。
  - 在全球化部署（有20+Region），方便拓展全球化业务。
  - 提供30+日志接入SDK，与阿里云产品无缝打通集成。

  日志服务采集和可视化可以参见如下文章，非核心功能不展开做比较。

  - 采集
  - 可视化



总结

ES支撑更新、查询、删除等更通用场景，在搜索、数据分析、应用开发等领域有广泛使用，ELK组合在日志分析场景上把ES灵活性与性能发挥到极致；日志服务是纯定位在日志类数据分析场景的服务，在该领域内做了很多定制化开发。一个服务更广，一个场景更具针对性。当然离开了场景纯数字的比较没有意义，找到适合自己场景的才重要。

深入了解日志服务：

- 日志服务产品
- 了解更多

12.3. 监控分析平台对比

本文从运维和SRE团队角度介绍监控分析平台的建设与选择。

背景信息

运维和SRE团队承载着重要的职责，其工作内容复杂而广泛，从应用部署、性能和可用性监控、告警、值班，到容量规划、业务支撑等都有涉及。随着云原生、容器化和微服务的快速发展，迭代节奏愈发加快，运维和SRE团队面临更多挑战，运维和SRE团队面临常见的困境如下：

- 业务线广泛
  - 业务线分布广泛，包括客户端、前端Web、应用后端。
  - 同时支持几条甚至数十条业务线。
- 人力严重短缺

  相对开发人员，不少公司的运维和SRE团队人员不到1%，甚至更低。
- 线上稳定性压力大
  - 经常扮演救火队员的角色。
  - 业务复杂、组件众多，快速排障和业务恢复的难度陡增。
- 缺乏统一而有效监控分析平台
  - 从不同的维度对各类数据进行监控，脚本泛滥、工具众多、烟囱林立。
  - 各类数据落在不同的系统中，欠缺关联分析，无法快速进行根因定位。
  - 阈值告警缺乏灵活性，一个系统可能出现数千条告警规则，管理成本高昂，并且容易造成告警泛滥，引起误判、漏判。

因此，一套简单易用、高效、分析能力强的监控分析平台，对于提高运维和SRE团队的工作效率、快速而准确进行根因定位、保证业务连续性至关重要。

监控分析平台需要解决的数据问题

运维和SRE团队为了保证业务稳定和支持业务发展，需要对大量的数据进行采集和分析，包括机器硬件、网络指标、用户行为等多方面的数据。在完成数据采集后，还需要有一套合适的系统进行转换、存储、处理、分析，满足多样的需求。数据问题主要包括：

- 数据多样
  - 各类系统数据：cpu、mem、net、disk等通用硬件指标，系统日志。
  - 业务黄金指标：延时、流量、错误、饱和度。
  - 业务访问日志：Access Log。
  - 应用日志：Java应用日志、错误日志。
  - 用户行为数据：Web click。
  - App埋点数据：Android、iOS App中埋点统计。
  - 各类框架数据：被广泛使用的K8s框架产生的数据。
  - 服务调用链：各类Tracing数据。
- 需求多样

对于各类数据，运维和SRE团队不仅需要保障业务稳定，还需要支持其他业务团队进行数据的使用，对于数据的使用也是多样的，常见需求如下：

- 监控、报警：实时处理（流式，小批量），秒级~分钟级延时。
- 客服、问题排查：快速检索，例如通过关键词过滤，秒级延时。
- 风控：实时流量处理，秒级延时。
- 运营、分析：大规模数据分析，如OLAP场景，秒级到小时级延时。

- 资源需求估算难

  对于快速发展的业务，各类数据的规模在一开始是很难准确估算的，经常遇到：

  - 新业务接入，数据量无准确估算参考。
  - 业务快速发展，数据暴增。
  - 数据使用需求变动，原有存储方式，保存时间不符合使用需求。

构建监控分析平台方案选择

由于数据来源广、样式杂，需求多，运维和SRE团队往往需要使用和维护多套系统，才能满足多样的监控和业务需求，常见的开源组合如下：

- Telegraf+Influxdb+Grafana

  Telegraf是一个轻量级的采集框架，通过丰富的插件采集操作系统、数据库、中间件等各类指标，配合Influxdb对时序数据进行高效读写、存储和分析，然后在Grafana上进行可视化展示和交互式查询。

- Prometheus

  在云原生K8s的生态中，Prometheus基本上作为时序数据的标配，配合灵活的exporter可以非常方便地采集Metric数据，同时Prometheus也可以和Grafana集成。

- ELK

  在日志数据多维度查询和分析上，ELK套件是常用的开源组件，提供快速、灵活、强大的查询能力，可满足研发、运维、客服团队的大部分查询需求。

- Tracing类工具

  在微服务、分布式的系统中，请求调用链路复杂，没有一套合适的Tracing系统，很难进行高效的问题根因定位，从Zipkin、Jeager到逐渐形成行业标准的OpenTelemetry、SkyWalking都是不错的Tracing系统，而这些Tracing系统并未提供数据存储组件，需要配合ES或Cassandra来存储Tracing数据。

- Kafka+Flink

  对于数据清洗、风控等需求，需要构建一套实时数据通道和流式系统，支撑数据的全量实时消费，一般使用Kafka和Flink组合。

- ClickHouse、Presto、Druid

  在运营分析、报表等场景中，为了追求更高的实时响应性，通常还会将数据导入OLAP引擎，在秒级到分钟级内完成海量数据分析需求，以及各类Adhoc的查询。

不同组件面向不同的数据类型和处理需求，数据需要在其中流转，有时候同一份数据需要同时保存在多个系统中，增加系统复杂度和使用成本。

当数据越来越多，使用需求越来越广时，保障这些组件的稳定性、满足多种业务性能需求、进行有效的成本控制，又要对大量业务线进行高效支撑，都是非常繁重而又有挑战的工作。

监控分析平台的挑战

能够维护好多套系统又能有效支持众多业务线，这是一个巨大的挑战。

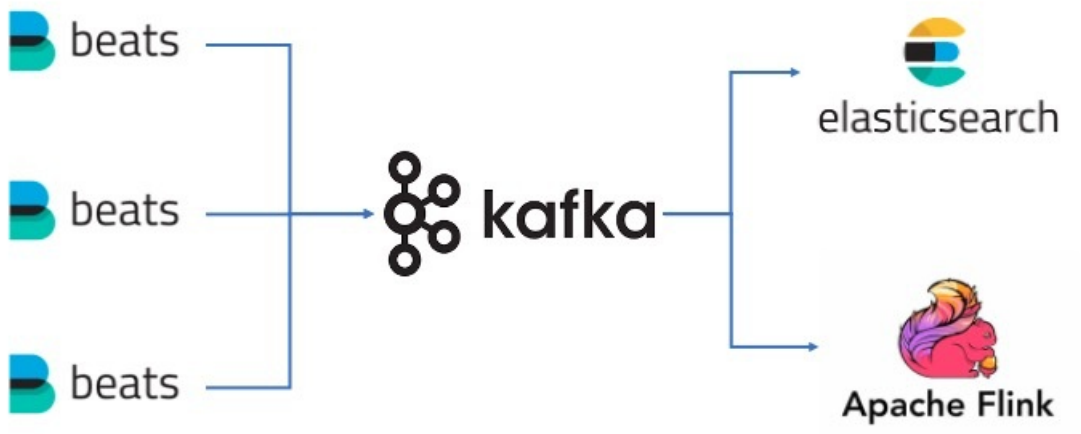
- 稳定性保障
  - 依赖系统：数据在多套系统中流转，系统之间又存在依赖关系，当某系统出现问题时，对其他系统造成影响。例如下游ES系统写入变慢后，用于缓存数据的Kafka集群存储水位变高，可能导致集群写满。
  - Burst问题：在互联网环境下，流量Burst是非常常见的情况。对于监控分析平台也一样，当大量数据需要写入系统时，保证系统不被压垮，同时保证读取功能正常运转，是一项巨大的挑战。
  - 资源隔离：不同数据的优先级有高低，如果过分依赖资源物理隔离将导致集群资源严重浪费和运维成本极大提高，而当数据共享资源时，需要尽可能保证相互之间不受干扰。例如某些系统中，一次超大规模的查询，可能拖垮整个集群。
  - 技术门槛：各类系统都有大量参数需要调优，面对不同的场景和需求，调优模式也不尽相同，需要投入大量的时间和精力，根据实际情况进行对比和优化。
- 性能可预期
  - 数据规模：对系统的性能有非常大的影响。例如时序数据在千万级到亿级时间线下读写，ES在10亿到100亿行数据中的查询性能保证，都非常有挑战。
  - QoS控制：任意一个系统的硬件资源都是有限的，需要对不同数据的QPS、并发进行合理的分配和管理，必要时进行降级处理，否则某个业务的使用可能导致其他业务性能受损。而开源组件一般很少考虑QoS的控制。
- 成本控制
  - 资源成本：各类组件的部署都需要消耗硬件资源，特别是当数据同时存在多个系统中的时候，硬件的资源消耗将更加严重。另外一个常见问题是很难准确估算业务的数据量。很多时候，采用相对保守手段来降低系统水位，这又将造成资源浪费。
  - 接入成本：支持各业务线数据接入也是一个繁重的工作，涉及到数据格式的适配、环境管理、配置设置和维护、资源估算等一系列工作，需要有工具或平台帮助业务线自主完成，否则运维和SRE团队将陷入大量的琐事中。
  - 支持成本：使用各种系统难免会遇到各类问题，必要的技术支持必不可缺，但问题种类多样。例如使用模式不合适、参数配置不合理等。遇到开源软件本身BUG导致的问题，又是一笔额外的成本。
  - 运维成本：各系统的软硬件难免会出故障，硬件替换、缩扩容、软件版本升级，都需要投入不小的人力和精力。
  - 费用分摊：只有将资源消耗清晰准确地分摊到实际业务线中，才能更有效利用资源，制定合理的预算和规划。这也需要监控分析平台能提供有效的计量数据进行费用分摊。

实际场景模拟

业务背景

- 公司有100多应用，每个应用都有Nginx访问日志和Java应用服务日志。
- 各应用日志规模变化巨大，单日1 GB到1 TB不等，每天新增10 TB数据，需保存7天~90天，平均15天。
- 日志数据主要用于业务监控和报警、线上问题排查以及实时风控使用。

业务架构选型



- Beats：实时采集数据发送至Kafka。
- Kafka：数据临时存储，用于Flink实时消费和导入Elasticsearch。
- Flink：对业务数据实时分析，进行实时监控、风控。
- Elasticsearch：日志查询与分析，问题排查。

在以上看似简单的架构中，也隐藏了大量细节需要关注，以ES为例：

- 容量规划：原始数据*膨胀系数*（1+副本数）*（1+预留空间），一般膨胀系数取1.1~1.3，1个副本，25%的预留（剩余空间，临时文件等），实际磁盘空间是原始空间的2.75~3.5倍。如果需要开启_all参数设置，数据膨胀会更严重，也需要预留更多空间。
- 冷热分离：所有数据全部保存到SSD上，成本过高。需要根据数据的重要程度和时间因素，将部分索引数据直接保存至HDD磁盘或使用Rollover功能迁移索引数据。
- 索引设置：每个应用的两类日志，分别按照时间周期性创建索引，根据数据大小合理设置Shard数，单Shard以30~50 GB为宜，但是各应用的日志量很难准确估计，常在遇到写入或查询问题后再调整，然而重建索引的消耗又非常大。
- Kafka消费设置：使用Logstash消费Kafka数据再写入到ES，需要Kafka topic的patition数和logconsumer_threads相匹配，否则容易导致各partition消费不均。
- ES参数调优：对写入吞吐、可见性延时、数据安全性以及查询性能等多方面因素进行综合评估和权衡后，结合集群CPU、内存，对ES一些列参数进行调优，才能更好发挥ES的特性。常见的参数包括线程数、内存控制、translog设置、队列长度、各类操作的间隔interval、merge参数等。
- 内存：通常JVM堆内存大小在32 GB以内，剩余的留给OS缓存使用，如果频繁GC会严重影响性能，甚至直接导致服务不可用。
  - master节点内存占用和集群中Shard数直接相关，一般集群Shard需要控制在10,000个以内，ES默认配置中，单节点Shard数上限为1000个，需要合理控制索引和Shard数量。
  - data节点的内存由索引数据规模决定，如ES的FST会长期驻留在内存，虽然在7.3及之后版本中，提供了堆外内存方式（mmap），但缓存被系统回收又会导致查询性能下降，如果使用的是更低版本，则只能控制单节点数据大小。
- 查询与分析：影响查询与分析性能的因素非常多，需要花费大量时间不断试错和积累。
  - 合理设置mapping，例如text和keyword的选择，尽量避免无必要的nested mapping。
  - 避免过大的查询范围和复杂度（过深的Group by语句等），以免急剧消耗系统资源。对结果集大小进行限制，否则复杂的聚合查询或模糊查询等，在过大数据集上甚至直接导致内存溢出（OOM）。
  - 控制segment数量，必要时进行force merge，也需要评估force merge带来的大量IO和资源消耗。

- 合理选择Filter和Query。在无需计算的场景中，Filter可以更好使用Query Cache，速度要明显快于Query。
- script脚本带来的性能和稳定性问题。
- 合理使用好routing可以使得单次查询只扫描某个Shard数据，提升性能。

- 数据损坏：如果遇到异常的crash，可能导致文件损坏。在segment或translog文件受损时，Shard可能无法加载，需要使用工具或手动将有问题的数据清理掉，但这也会导致部分数据丢失。

以上是在使用和运维ES集群中，经常会遇到和需要注意的问题，稳定维护好ES集群可真不是一件容易的事情，特别是当数据逐步扩大到数百TB，又有大量使用需求的情况下。同样的问题也存在其他系统中，这对于平时工作极其繁忙的运维和SRE同学是不小的负担。

云上一体化服务选择

针对运维和SRE团队工作中的监控分析平台需求，以及平台搭建过程中遇到的种种问题，阿里云日志服务团队希望在云上提供一套简单易用、稳定可靠、高性能而又具有良好性价比的解决方案，以支持运维和SRE团队更高效地工作。日志服务从原本只支持阿里巴巴集团和蚂蚁集团内部日志系统开始，逐步完善，演进成为同时支持Log、Metric、Trace的PB级云原生观测分析平台。

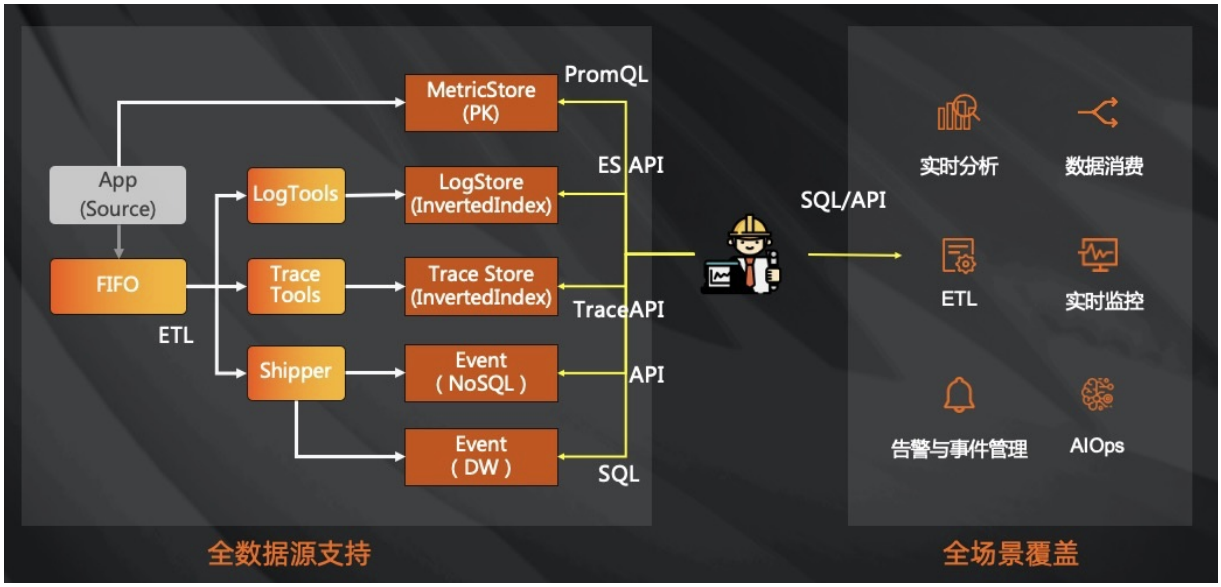
- 接入数据极其简便
  - Logtail：经过多年百万级服务器锤炼，简便、可靠、高性能，界面化管理。
  - SDK/Producer：接入各类移动端Jave、C、GO、iOS、Android、Web Tracking数据。
  - 云原生：云上ACK原生支持，自建CRD一键接入。
- 实时消费和生态对接
  - 秒级扩容能力，支持PB级数据实时写入和消费。
  - 原生支持Flink、Storm、Spark Streaming等主流系统。
- 海量数据查询分析力
  - 百亿规模秒级查询。
  - 支持SQL92语法，支持交互式查询，支持机器学习、安全检测等函数。
- 数据加工
  - 对比传统的ETL，可节省90%的开发成本。
  - 纯托管、高可用、高弹性扩展。
- Metric数据

  云原生Metric数据接入，支持亿级时间线的Prometheus存储。
- 统一的Tracing方案

  支持OpenTelemetry协议，兼容Jaeger、Zipkin等OpenTracing协议，支持OpenCensus、SkyWalking等方案。
- 完善的监控和报警

  一站式完成告警监控、降噪、事务管理、通知分派。
- 异常智能诊断

  高效的无监督流式诊断和人工打标反馈机制，大大提高了监控效率和准确率。



相比开源多套系统的方案，日志服务采用All in one模式。在一个系统中，完整支持运维和SRE团队工作中的监控分析平台需求，可以直接替代搭建Kafka、ES、Prometheus、OLAP等多套系统的组合，具有如下优势：

- 降低运维复杂度
  - 云上服务、开箱即用、零运维成本、无需再维护和调优多套系统。
  - 可视化管理、5分钟完成接入、业务支持成本大大降低。
- 成本优化
  - 数据只保留一份，无需将数据在多套系统中流转。
  - 按量使用，无预留资源的浪费。
  - 提供完善的技术支持，人力成本大大降低。
- 完善的资源权限管理
  - 提供完整的消费数据，助力完成内部分账和成本优化。
  - 完整的权限控制和资源隔离，避免重要信息泄露。

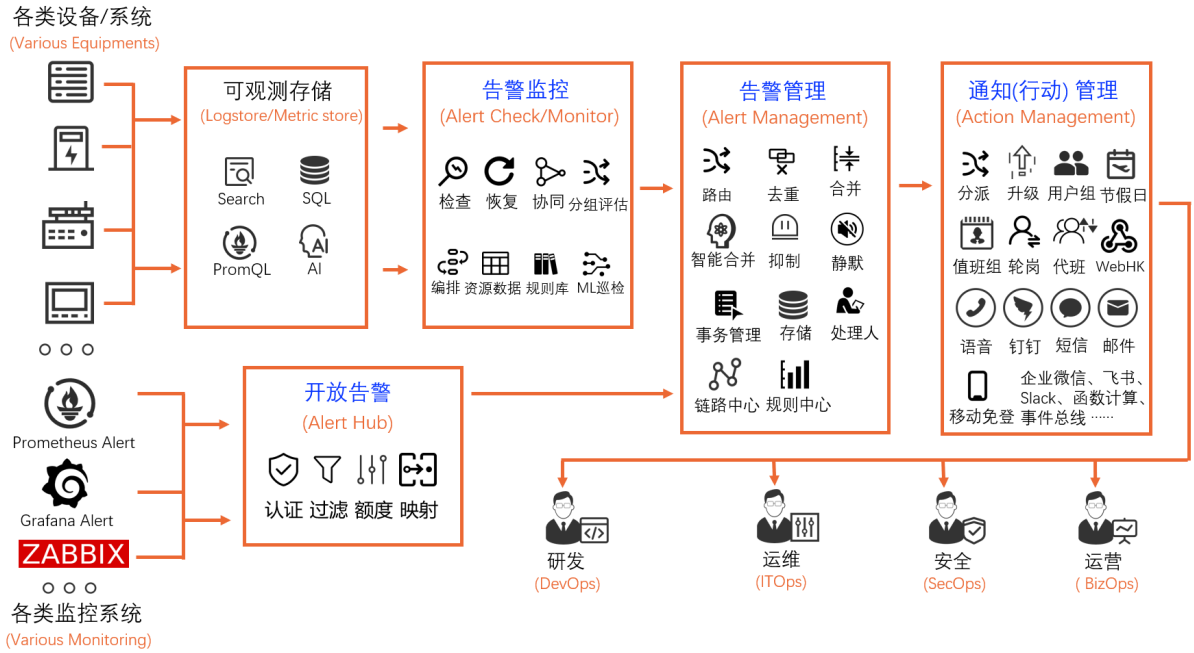
日志服务希望通过自身的不断进步，为Log、Metric、Trace等数据提供大规模、低成本、实时平台化服务，助力运维和SRE团队更高效工作，更有效支持业务快速发展。

12.4. 可观测告警运维系统对比

日志服务新版告警是一站式的告警监控、降噪、事务管理、通知分派的智能运维平台。本文介绍日志服务新版告警与各个开源告警系统的对比信息。

日志服务告警

日志服务新版告警支持监控日志、时序等各类数据，支持接收三方告警，支持对告警进行降噪、事件管理、通知管理等，新增40+功能场景，充分考虑研发、运维、安全以及运营人员的告警监控运维需求。更多信息，请参见什么是日志服务告警。



新版告警具备如下五大优势：

与ELK X-Pack告警（ElasticSearch Watcher、Kibana 7.x+Alert）对比

自建ELK使用开源的ElasticSearch+Logstash+Kibana组合，其不包括告警功能。如果您要为自建ELK配置告警，需额外购买X-Pack商业包，其包含两个告警功能（ElasticSearch Watcher和Kibana 7.x+Alert）。这两个告警功能互相独立，不能协同与关联。

类别	对比项	日志服务告警	ELK X-Pack告警
持久性	告警服务可用性	服务可用性>99.9%、存储持久性>99.99999999%。	商业版采用分布式，存储数据需要手动配置。

类别	对比项	日志服务告警	ELK X-Pack告警
成本	费用	无订阅费用、免运维、监控与告警管理免费、通知渠道仅短信和语音按照条数收取少量费用。	商业订阅费用、人工运维费用、自购的机器费用、三方短信和语音费用。
告警监控	监控日志和时序数据的规模	PB级别。	TB级别。
	监控查询分析语法	支持SQL92语法（含扩展）、PromQL语法、告警语法扩展。	• ElasticSearch Watcher：支持ES DSL。 • Kibana 7.x+Alert：支持有限的过滤聚集操作。
	机器学习能力	支持十多种预测、异常检测、根因分析等AI算法。	支持X-Pack ML算法。
	数据协同能力	支持跨存储库、跨Project、跨地域、跨账号协同监控。	支持同一集群下的同构索引合并分析。
	无数据告警	支持。	不支持。
	告警恢复	支持。	不支持。
	标签与标注	支持。	Kibana 7.x+Alert支持自定义标签。
	动态严重度	支持。	不支持。
	分组评估	支持，可自定义配置。	• ElasticSearch Watcher：固定不分组。 • Kibana 7.x+Alert：固定自动分组。
	监控侧控制	• 支持配置持续触发阈值。 • 支持暂停和自动恢复（基于时间）监控。	ElasticSearch Watcher支持暂停和自动恢复（基于ACK）。
告警管理	告警管理	• 支持告警去重、合并、抑制、静默。 • 支持事务管理、责任人设置。	不支持。
通知管理	通知管理	支持通知渠道动态分派、告警级别提升、接收组管理、渠道日历设置、值班表设置、渠道额度控制。	不支持。

类别	对比项	日志服务告警	ELK X-Pack告警
	常用渠道	支持短信、语音、钉钉、邮件、WebHook、阿里云消息中心等通知渠道。 其中通过WebHook，还支持企业微信、飞书、Slack等渠道。	支持邮件、WebHook等通知渠道，不支持短信和语音。 • Watcher支持PagerDuty、JIRA、Slack。 • Kibana Alert支持IBM Resilient、MS Teams、Service Now。

与Prometheus&Loki（含AlertManager）告警对比

自建Prometheus&Loki使用开源的Prometheus+Loki+Alert Manager组合搭建告警监控系统，其中Prometheus Alert对时序数据进行告警监控，Loki对日志进行告警监控，两者共同将告警发送给Alert Manager进行告警管理。

类别	对比项	日志服务告警	Prometheus+Loki 2.0告警
持久性	告警服务可用性	服务可用性>99.9%、存储持久性>99.99999999%。	部分服务采用分布式、部分服务采用单机可用性；存储采用单机可用性。
成本	费用	无订阅费用、免运维、监控与告警管理免费、通知渠道仅短信和语音按照条数收取少量费用。	人工运维费用、自购的机器费用、三方短信和语音费用。
告警监控	监控日志和时序数据的规模	PB级别。	• 日志：百GB级别。 • 时序数据：TB级别。
	监控查询分析语法	支持SQL92语法（含扩展）、PromQL语法、告警语法。	• 日志：LogQL语法。 • 时序数据：PromQL语法。
	机器学习能力	支持十多种预测、异常检测、根因分析等AI算法。	不支持。
	数据协同能力	支持跨存储库、跨Project、跨地域、跨账号协同监控。	支持同一集群下跨指标PromQL Join。
	无数据告警	支持。	不支持。
	告警恢复	支持。	支持。
	标签与标注	支持。	支持。
	动态严重度	支持。	不支持。
	分组评估	支持，可自定义配置。	支持按标签固定分组。
	监控侧控制	• 支持配置持续触发阈值。 • 支持暂停和自动恢复（基于时间）监控。	支持设置持续触发阈值，不支持暂停与恢复监控。

类别	对比项	日志服务告警	Prometheus+Loki 2.0告警
告警管理	告警管理	• 支持告警去重、合并、抑制、静默。 • 支持事务管理、责任人设置。	支持告警去重、合并、抑制、静默，不支持事务管理、责任人管理。
通知管理	通知管理	支持通知渠道动态分派、告警级别提升、接收组管理、渠道日历设置、值班表设置、渠道额度控制。	仅支持渠道动态分派，其他不支持。
	常用渠道	支持短信、语音、钉钉、邮件、WebHook、阿里云消息中心等通知渠道。 其中通过WebHook，还支持企业微信、飞书、Slack等渠道。	支持邮件、企业微信、WebHook（不支持自定义Body）、PagerDuty、PushOver、Slack、OpsGenie、VictorOps。不支持短信、语音服务。 通过三方插件，也可以支持钉钉、飞书、Slack等渠道。

与InfluxDB 2.0告警（含Kapacitor）告警对比

自建InfluxDB使用开源的InfluxDB OSS 2.0+Kapacitor组合搭建告警监控系统。如果您需要集群部署功能，还需要购买InfluxDB商业版。该方案仅适用于时序数据的告警监控。

类别	对比项	日志服务告警	InfluxDB 2.0 告警（含Kapacitor）
持久性	告警服务可用性	服务可用性>99.9%、存储持久性>99.99999999%。	商业版采用分布式，支持存储配置。开源采用单机版。
成本	费用	无订阅费用、免运维、监控与告警管理免费、通知渠道仅短信和语音按照条数收取少量费用。	商业版订阅费用、人工运维费用、自购的机器费用、三方短信和语音费用。
告警监控	监控日志和时序数据的规模	PB级别。	• 日志：不支持。 • 时序数据：TB级别。
	监控查询分析语法	支持SQL92语法（含扩展）、PromQL语法、告警语法扩展。	支持Flux语法。
	机器学习能力	支持十多种预测、异常检测、根因分析等AI算法。	支持Loud ML算法。
	数据协同能力	支持跨存储库、跨Project、跨地域、跨账号协同监控。	支持单集群下跨Bucket Flux Join。
	无数据告警	支持。	不支持。
	告警恢复	支持。	不支持。
	标签与标注	支持。	支持设置简单的标签。
	动态严重度	支持。	支持。

类别	对比项	日志服务告警	InfluxDB 2.0 告警（含Kapacitor）
	分组评估	支持，可自定义配置。	不支持。
	监控侧控制	• 支持配置持续触发阈值。 • 支持暂停和自动恢复（基于时间）监控。	不支持。
告警管理	告警管理	• 支持告警去重、合并、抑制、静默。 • 支持事务管理、责任人设置。	仅支持告警抑制，其他不支持。
通知管理	通知管理	支持通知渠道动态分派、告警级别提升、接收组管理、渠道日历设置、值班表设置、渠道额度控制。	仅支持通知渠道动态分派，其他不支持。
	常用渠道	支持短信、语音、钉钉、邮件、WebHook、阿里云消息中心等通知渠道。 其中通过WebHook，还支持企业微信、飞书、Slack等渠道。	支持邮件、WebHook（不支持灵活自定义Body）、exec、PagerDuty、PushOver、Slack、OpsGenie、VictorOps、HipChat等通知渠道。不支持短信、语音服务。