ALIBABA CLOUD

阿里云

日志服务

应用中心（App）

文档版本：20210906

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

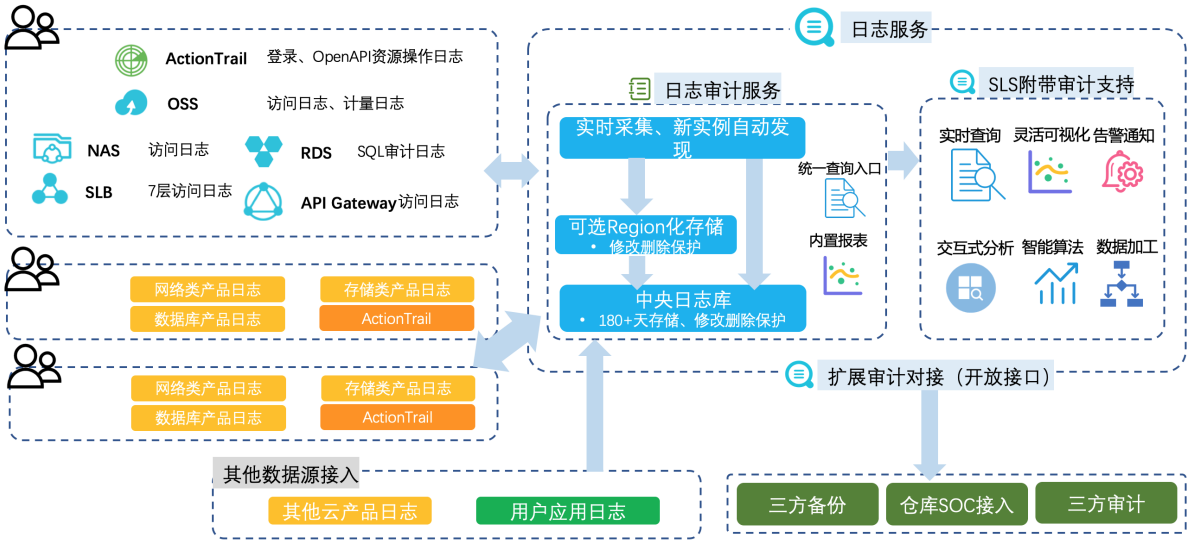
1.日志审计服务

1.1. 日志审计服务概述

本文介绍日志审计服务的功能特性、背景信息、应用场景、技术优势及覆盖的云产品。

功能特性

日志审计服务在继承现有日志服务所有功能外，还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。日志审计服务覆盖基础（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 1.0、云原生数据库PolarDB）、安全（WAF、DDoS防护、云防火墙、云安全中心）等产品，并支持自由对接其他生态产品或自有SOC中心。



背景信息

- 日志审计是法律刚性需求。

无论国内外，企业落实日志审计越来越迫切。尤其中国内地于2017年实施了《网络安全法》、于2019年12月实施《网络安全等保2.0标准》。

《网络安全法》（2017年6月1日 实施）

（三）采取监测、记录网络运行状态、网络安全事件的技术措施，并按照规定留存相关的网络日志不少于六个月

（第三章第一节第二十一条）

《GDPR》（2018年5月25日实施）

惩罚力度较强（2%/4%营收-可叠加），但有模糊空间
覆盖大部分国际业务的公司：
- 在欧盟境内拥有业务；或在欧盟境内没有业务，但是存储或处理欧盟公民的个人信息；
- 超过250名员工；或少于250名员工，但是其数据处理方式影响数据主体的权利和隐私，或是包含某些类型的敏感个人数据。

较明确规定了数据保护的一些细节：
- 网络数据（IP地址或cookie数据）等信息也被纳入保护范围
- 规定了数据是否需要离开信息拥有者所在地
- 规定了3年后客户敏感信息的脱敏要求等

《网络安全等保2.0标准》
（2019年5月13日发布，12月1日实施）

规定了哪些行为、事件需要审计：
- 网络边界、重要网络节点
- 覆盖每个用户、支持重要用户行为、重要安全事件
- 远程访问的用户行为、访问互联网用户行为等单独行为
- 网络系统操作、重要存储操作、计算系统操作、安全系统操作需详细记录
- 审计管理员的操作本身也需要记录并审计

对审计系统提出要求：
- 集中收集、集中分析，集中存储时长要求（180天以上）
- 对审计数据的保护（备份、防修改、覆盖、防止中断等）
- 需提供分析、监控报警等支持可疑行为发现
- 需要提供数据汇集接口，供第三方审计

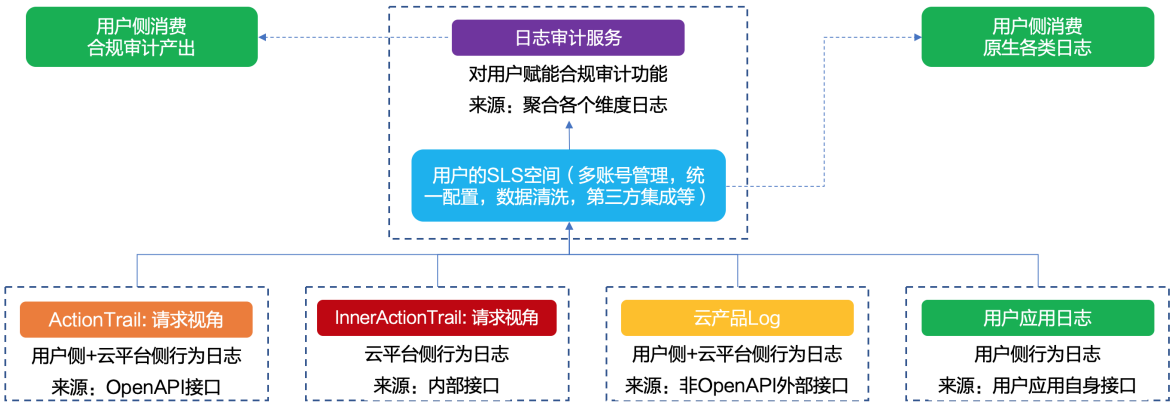
也覆盖云平台内部操作日志的审计

通过HIPAA、GLBA、PCI DSS、SOX、FISMA和ISO 27001/2等审计

- 日志数据保存180天
- 可以被溯源
- 无法篡改

- 日志审计是客户安全合规依赖的基础。

很多企业自身有成熟的法规条例以及合规审计团队，对账号设备的操作、网络行为、日志进行审计。客户可以直接消费原生各类日志，也可以使用日志审计服务提供的审计功能，构建并输出合规的审计信息。如果客户有安全中心（SOC），则可以直接消费日志审计中的日志，也可以使用阿里云安全中心消费日志。



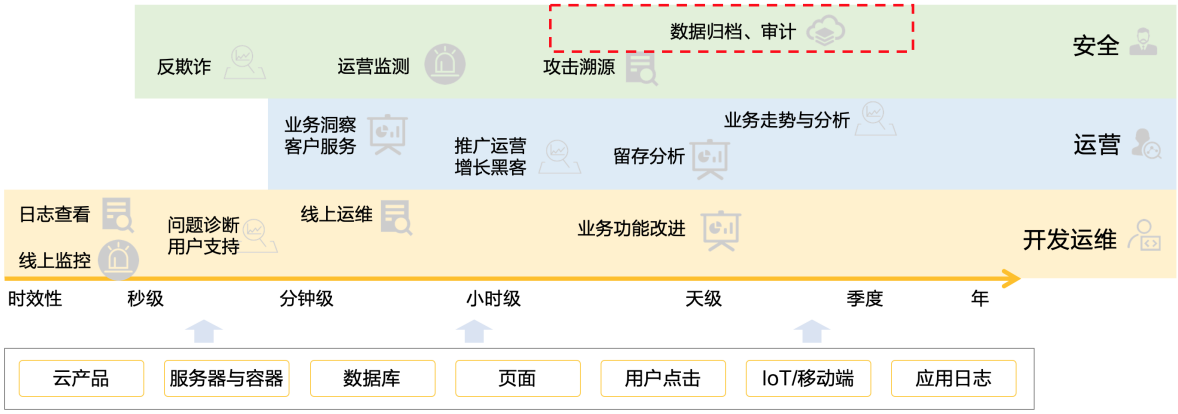
- 日志审计是安全防护的重要一环。

根据FireEye M-Trends 2018报告，企业安全防护管理能力薄弱，尤其是亚太地区。全球范围内企业组织的攻击从发生到发现所需时长平均101天，而亚太地域平均需要498天。企业需要长期、可靠、无篡改的日志记录与审计支持来持续缩短这个时间。

应用场景

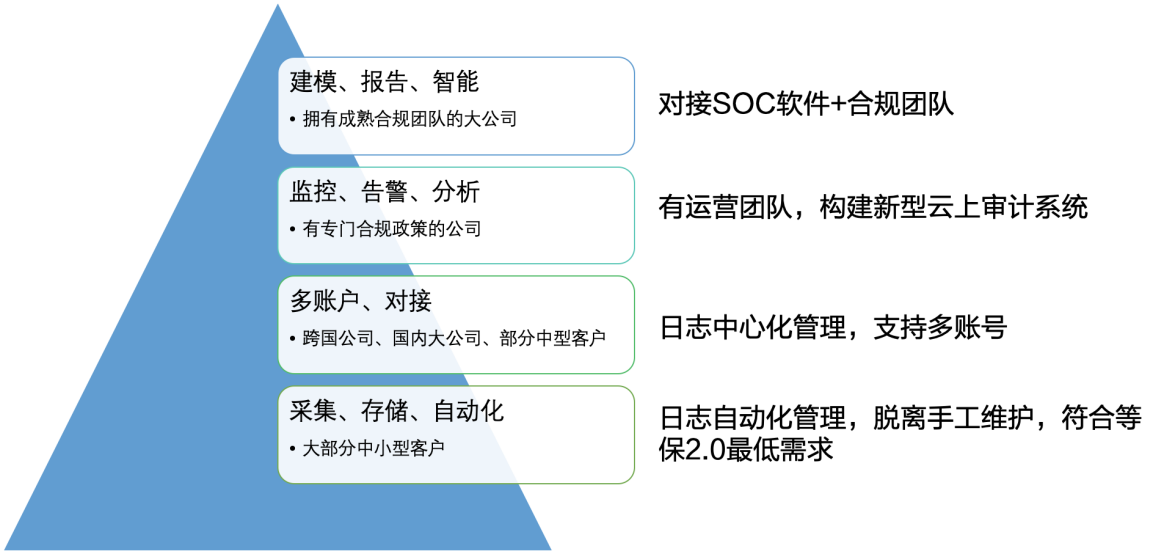
- 日志服务与审计场景

日志服务作为行业领先的日志大数据解决方案，提供一站式数据采集、清洗、分析、可视化和告警功能。支持日志服务相关场景：DevOps、运营、安全、审计。



- 典型日志审计场景

  日志审计一般分成如下4层需求。



  - 基础需求：大部分中小企业客户需要自动化采集存储日志。他们的主要诉求是满足《网络安全等保2.0标准》中的最低要求，并脱离手工维护。
  - 高级需求：跨国企业、大企业以及部分中型企业，存在多个部门之间独立结算并且在阿里云账号的使用上各自隔离，但是在审计的时候，需要自动化、统一采集相关日志。他们的主要诉求是除上述的基础诉求外，还希望中心化采集日志并支持多个账号的简单管理。这部分企业一般拥有审计系统，因此对日志审计的需求是能够实时、简单的对接。
  - 更上层的需求：拥有专门合规团队的大公司，他们需要对日志进行监控、告警和分析。一部分客户采集数据到审计系统中进行操作。另一部分客户（尤其是计划在云上搭建一套新审计系统的客户）可以使用日志服务提供的审计支持（查询、分析、告警、可视化等）进行审计操作。
  - 最顶端需求：拥有专业成熟审计合规团队的大企业，一般拥有自己的安全中心或审计系统，他们的核心需求是对接数据进行统一操作。

  针对以上4类客户需求，日志服务的日志审计服务都可以比较好的满足。

技术优势

- 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。
  - 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。

云产品覆盖及相关资源

日志审计服务支持采集基础（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 1.0、PolarDB MySQL云原生数据库）、安全（WAF、云防火墙、云安全中心、DDoS防护）等云产品日志。采集完成后，会自动存储到对应Logstore或Metricstore中，并生成对应的仪表盘。详细信息如下：

云产品	审计相关日志	采集地域	使用前提	日志服务资源
操作审计	• RAM登录日志 • 阿里云产品的资源操作日志 • 通过OpenAPI的操作行为日志	所有在售地域	无	• Logstore名称 actiontrail_log • 仪表盘名称 ◦ ActionTrail审计中心 ◦ ActionTrail核心配置中心 ◦ ActionTrail登录中心
负载均衡	HTTP或HTTPS侦听实例的7层网络日志	所有在售地域	无	• Logstore名称 slb_log • 仪表盘名称 ◦ SLB审计中心 ◦ SLB访问中心 ◦ SLB全局数据
API网关	访问日志	所有在售地域	无	• Logstore名称 apigateway_log • 仪表盘名称 API网关审计中心

云产品	审计相关日志	采集地域	使用前提	日志服务资源
Web应用防火墙	• 访问日志 • 攻击日志	所有在售地域	• 高级版本及以上 • 需在WAF控制台中购买**日志服务**模块。更多信息，请参见开通WAF日志服务。	• Logstore名称 waf_log • 仪表盘 ◦ WAF审计中心 ◦ WAF安全中心 ◦ WAF访问中心
云安全中心	• 主机日志（7种） • 网络日志（4种） • 安全日志（3种）	所有在售地域	• 企业版本 • 需在SAS控制台中开通**日志分析**功能。更多信息，请参见开通日志分析功能。	• Logstore名称 sas_log • 仪表盘名称 ◦ 主机 ▪ 账户快照 ▪ 进程快照 ▪ 网络连接中心 ◦ 网络 ▪ 网络会话 ▪ DNS中心 ◦ 安全 ▪ Web访问漏洞中心 ▪ 基线中心 ▪ 安全告警中心
云防火墙	互联网边界防火墙流量日志、VPC边界防火墙流量日志	不涉及	• 高级版本及以上 • 需在云防火墙控制台中购买**日志分析**模块。更多信息，请参见开通日志分析功能。	• Logstore名称 cloudfirewall_log • 仪表盘名称 云防火墙审计中心
堡垒机	操作命令日志	所有在售地域	V3.2版本及以上	• Logstore名称 bastion_log • 仪表盘名称 无

云产品	审计相关日志	采集地域	使用前提	日志服务资源
对象存储	• 资源操作日志 • 数据操作日志 • 数据访问日志、计量日志 • 过期文件删除日志 • CDN回流日志	所有在售地域	无	• Logstore名称 oss_log • 仪表盘名称 ◦ OSS审计中心 ◦ OSS访问中心 ◦ OSS运维中心 ◦ OSS性能中心 ◦ OSS全局数据
云数据库RDS	• RDS审计日志 • MySQL慢日志 • MySQL性能日志	除本地云以外的其他在售地域	• 审计日志 ◦ MySQL：不支持基础版 ◦ PostgreSQL、Microsoft SQL Server：无限制 ◦ 均需开启SQL洞察或审计功能，由日志审计服务自动开启。 • 慢日志、性能日志 只支持非基础版的MySQL实例。	• 审计日志 ◦ Logstore名称 rds_log ◦ 仪表盘名称 ■ RDS审计中心 ■ RDS审计安全中心 ■ RDS审计性能中心 ■ RDS全局数据 • 慢日志 ◦ Logstore名称 rds_log ◦ 仪表盘名称 无 • 性能日志 ◦ Metricstore名称 rds_metrics ◦ 仪表盘名称 RDS性能监控

云产品	审计相关日志	采集地域	使用前提	日志服务资源

云数据库PolarDB	• PolarDB审计日志
	• PolarDB MySQL慢日志
	• PolarDB MySQL性能日志
		所有在售地域	• 审计日志
			  ○ 支持MySQL集群和PostgreSQL集群。
			  ○ 需开启SQL洞察或审计功能。由日志审计服务自动开启。
			• 慢日志、性能日志
			只支持MySQL集群。
				• 慢日志、审计日志
				  ○ Logstore名称
				polardb_log
				  ○ 仪表盘名称
				无
				• 性能日志
				  ○ Metricstore名称
				polardb_metrics
				  ○ 仪表盘名称
				PolarDB性能监控

云原生分布式数据库PolarDB-X 1.0	PolarDB-X 1.0审计日志	华北1（青岛）、华南1（深圳）、华东2（上海）、华北2（北京）、华东1（杭州）、华北3（张家口）、西南1（成都）、中国（香港）	无	• Logstore名称
				drds_log
				• 仪表盘名称
				  ○ DRDS运营中心
				  ○ DRDS安全中心
				  ○ DRDS性能中心

文件存储	访问日志	所有在售地域	无	• Logstore名称
				nas_log
				• 仪表盘
				  ○ NAS概览
				  ○ NAS审计中心
				  ○ NAS运维中心

移动推送	推送回调事件	中国内地	无	• Logstore名称
				cps_log
				• 仪表盘名称
				  ○ Android回执中心
				  ○ iOS回执中心

云产品	审计相关日志	采集地域	使用前提	日志服务资源

容器服务Kubernetes版	• Kubernetes审计日志
	• Kubernetes事件中心
	• Ingress访问日志
		华东2（上海）、华北2（北京）、华东1（杭州）、华南1（深圳）、华北5（呼和浩特）、华北3（张家口）、西南1（成都）、中国（香港）	针对Kubernetes的采集，需要您先手动开通对应的日志采集功能。
			说明
			• 必须使用自动创建的专属Project（k8s-log-{ClusterID}），暂不支持自定义Project。
			• Kubernetes相关日志采集依赖于数据加工功能，会产生相应的加工费用。更多信息，请参见计费项。
			• 暂不支持跨账号采集K8s相关的日志。
			• Kubernetes审计日志的使用前提请参见通过日志服务采集Kubernetes容器日志。
			• Kubernetes事件中心的使用前提请参见创建并使用Kubernetes事件中心。
			• Ingress访问日志的使用前提请参见Ingress访问日志分析与监控。
				• Logstore名称
				  ◦ k8s_log
				  ◦ k8s_ingress_log
				• 仪表盘名称
				  ◦ Kubernetes审计中心概览
				  ◦ Kubernetes事件中心
				  ◦ Kubernetes资源操作概览
				  ◦ Ingress概览
				  ◦ Ingress访问中心

云产品	审计相关日志	采集地域	使用前提	日志服务资源
DDoS防护	• DDoS高防（新BGP）访问日志 • DDoS高防（国际）访问日志 • DDoS原生访问日志	不涉及	• DDoS高防（新BGP）：已在DDoS高防（新BGP）控制台上购买全量日志分析模块。更多信息，请参见开通全量日志分析功能。 • DDoS高防（国际）：已在DDoS高防（国际）控制台上购买全量日志分析模块。更多信息，请参见开通全量日志分析功能。 • DDDoS原生：已在DDoS原生控制台上购买全量日志分析模块。更多信息，请参见开通原生防护日志。	• Logstore名称 ddos_log • 仪表盘名称 ◦ DDoS高防（国际）访问中心 ◦ DDoS高防（国际）运营中心 ◦ DDoS高防（新BGP）访问中心 ◦ DDoS高防（新BGP）运营中心 ◦ DDoS原生防护事件报表 ◦ DDoS原生清洗分析报表
应用集成	操作日志	不涉及	无	• Logstore名称 appconnect_log • 仪表盘名称 无

金融云场景

在金融云场景中，日志审计服务在云产品覆盖和地域限制方面与公有云有所不同。

- 云产品覆盖

  日志审计服务支持采集ActionTrail、SLB、API网关、RDS、堡垒机、DDoS防护和云防火墙的日志。

云产品	审计相关日志	采集地域	使用前提	日志服务资源
操作审计	◦ RAM登录日志 ◦ 阿里云产品的资源操作日志 ◦ 通过OpenAPI的操作行为日志	华东2（上海）金融云	无	◦ Logstore名称 actiontrail_log ◦ 仪表盘名称 ▪ ActionTrail审计中心 ▪ ActionTrail核心配置中心 ▪ ActionTrail登录中心

云产品	审计相关日志	采集地域	使用前提	日志服务资源
负载均衡	HTTP或HTTPS侦听实例的7层网络日志	华东1（杭州）、华北1（青岛）、华东2（上海）金融云、华南1（深圳）金融云	无	◦ Logstore名称 slb_log ◦ 仪表盘名称 ▪ SLB审计中心 ▪ SLB访问中心 ▪ SLB全局数据
API网关	访问日志	华东1（杭州）金融云、华东2（上海）金融云、华南1（深圳）金融云	无	◦ Logstore名称 apigateway_log ◦ 仪表盘名称 API网关审计中心
云防火墙	互联网流量日志、边界防火墙流量日志	不涉及	◦ 高级版本及以上 ◦ 需在云防火墙控制台中购买**日志分析**模块。更多信息，请参见开通日志分析功能。	◦ Logstore名称 cloudfirewall_log ◦ 仪表盘名称 云防火墙审计中心

云产品	审计相关日志	采集地域	使用前提	日志服务资源
关系数据库RDS	○ MySQL审计日志 ○ SQL Server审计日志 ○ PostgreSQL审计日志 ○ MySQL慢日志	华东1（杭州）金融云、华东2（上海）金融云、华南1（深圳）金融云	○ 审计日志 ■ MySQL：不支持基础版 ■ PostgreSQL、Microsoft SQL Server：无限制 ■ 均需开启SQL洞察或审计功能。 由日志审计服务自动开启。 ○ 慢日志 只支持非基础版的MySQL实例。	○ 审计日志 ■ Logstore名称 rds_log ■ 仪表盘名称 ■ RDS审计中心 ■ RDS审计安全中心 ■ RDS审计性能中心 ■ RDS全局数据 ○ 慢日志 ■ Logstore名称 rds_log ■ 仪表盘名称 无
堡垒机	操作命令日志	华东1（杭州）	V3.2版本及以上	○ Logstore名称 bastion_log ○ 仪表盘名称 无
DDoS防护	DDoS高防（新BGP）访问日志	不涉及	已在DDoS高防（新BGP）控制台上购买全量日志分析模块。更多信息，请参见开通全量日志分析功能。	○ Logstore名称 ddos_log ○ 仪表盘名称 ■ DDoS高防（新BGP）访问中心 ■ DDoS高防（新BGP）运营中心

- 地域限制

采用中心化存储时，日志审计服务从各个阿里云账号的各个地域采集到的日志，会存储到中心阿里云账号下的一个中心化Project中，目前中心化存储可供选择的地域包括华东1（杭州）金融云、华东2（上海）金融云和华南1（深圳）金融云。更多信息，请参见使用限制。

1.2. 使用前须知

本文介绍日志审计服务的使用限制、费用说明等信息。

使用限制

- 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）
    - 海外：新加坡、日本（东京）、德国（法兰克福）、印尼（雅加达）
  - 区域化存储

    对于SLB、OSS、PolarDB-X 1.0的访问日志，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、OSS、PolarDB-X 1.0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

    对于SLB、OSS、PolarDB-X 1.0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，支持的地域：支持除华北1（青岛）外的所有地域。
- 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。
  - 对于SLB、OSS、PolarDB-X 1.0，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。
  - 对于SLB、OSS、PolarDB-X 1.0，如果开启了**同步到中心**功能，在对应的区域化Project中，会生成数据加工任务。
    - 数据加工任务名为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 DRDS。
    - 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任务。
    - 开启了**同步到中心**功能的区域化Logstore会变成同步专属的Logstore，您无法进行任何操作，如果需要查询等操作时，可以直接在中心化Logstore中操作。

费用说明

- 日志服务

  中心主账号需要开通日志服务与日志审计服务App，从其他主账号采集日志到中心主账号下。除特定云产品日志依赖外，其他主账号默认无需开通日志服务，也不会在其账号的日志服务下产生特定费用。目前日志审计服务免费，其涉及的数据存储、读写流量、数据加工等按量付费。更多信息，请参见计费项。

  > **说明** 特定云产品（例如负载均衡SLB、对象存储OSS、云原生分布式数据库PolarDB-X 1.0、容器服务Kubernetes版）的日志，在开启同步到中心后，会使用数据加工功能进行同步，其涉及的加工与跨网流量费用等按量付费。更多信息，请参见计费项。

  支持免费额度，支持用已购买的资源包抵扣相应的费用。

- 云产品

  开通日志审计服务与对应云产品的日志采集后，在云产品侧可能会产生额外的费用，如下所示。

云产品	额外费用
Web应用防火墙（WAF）	在Web应用防火墙控制台上购买**日志服务**模块，费用详情请参见计费方式。
云安全中心（SAS）	在云安全中心控制台开通**日志分析**功能，费用详情请参见计费模式。
云防火墙（Cloud Firewall）	在云防火墙控制台上购买**日志分析**模块，费用详情请参见日志分析计费方式。
关系数据库（RDS）	开启RDS MySQL审计日志采集功能后，会自动开启符合条件的RDS实例的SQL洞察（SQL审计）功能，费用详情请参见价格、收费项与计费方式。
PolarDB MySQL云原生数据库	开启PolarDB MySQL云原生数据库的审计日志采集功能后，会自动开启符合条件的PolarDB MySQL集群的SQL洞察（SQL审计）功能，费用详情请参见计费项概览。
DDoS防护	在DDoS高防（新BGP）控制台上购买**全量日志分析**模块，费用详情请参见概述。

1.3. 配置日志采集

本文介绍如何在日志审计服务中选择云产品进行日志采集。

前提条件

- 已注册阿里云账号。

  建议使用阿里云RAM用户，该RAM用户需具备RAM读权限（例如已被授权AliyunRAMReadOnlyAccess策略），且对日志服务有读写权限（例如被授权AliyunLogFullAccess策略）。

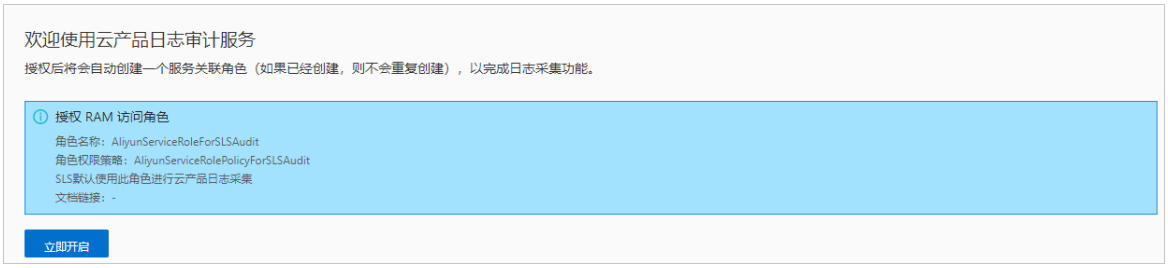
- 待采集日志的云产品已开启相应的服务。更多信息，请参见云产品覆盖及相关资源。

首次配置

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 根据页面提示完成授权。

   完成授权后，日志审计服务将使用服务关联角色AliyunServiceRoleForSLSAudit进行云产品的日志采集。更多信息，请参见管理服务关联角色AliyunServiceRoleForSLSAudit。

   > **注意**
   > - 执行该操作的账号具备AliyunRamFullAccess权限。
   > - 本操作只需执行一次。
   > - RDS审计中心和日志审计服务都需使用服务关联角色AliyunServiceRoleForSLSAudit进行日志采集，如果您已在RDS审计中心中执行此操作，则无需在日志审计服务中再次执行。



配置单账号采集

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在**云产品接入 > 全局配置**页面，开启日志采集功能。
   i. 在**中心项目Project所在区域**下拉列表中，选择日志中心化存储的目标地域。
      - 中国：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）
      - 海外：新加坡、日本（东京）、德国（法兰克福）、印尼（雅加达）

   ii. 在云产品列表中，选择需开启日志审计功能的云产品，并配置存储时间。

   如果是SLB 7层访问日志、OSS访问日志、PolarDB-X 1.0审计日志，还可以选择**同步到中心**。开启**同步到中心**后，区域化Project将作为中转，不需要存储很长时间，控制台会自动调整成推荐的时间。

   iii. 单击**保存**。
4. 在左侧导航栏，选择**云产品接入 > 接入状态**，查看日志接入状态。

配置完成后，需要2分钟左右完成初始同步。如果出现异常，请根据页面提示信息进行调整。更多信息，请参见常见问题及错误排查。

配置多账号采集

日志审计服务支持跨账号采集云产品日志到当前账号下的日志库中。在开始采集前，您需要先完成多账号配置。

> **注意** 暂不支持跨账号采集K8s相关的日志。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在**云产品接入 > 全局配置**页面，开启日志采集功能。
   i. 在**中心项目Project所在区域**下拉列表中，选择日志中心化存储的目标地域。
      - 中国：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）
      - 海外：新加坡、日本（东京）、德国（法兰克福）、印尼（雅加达）
   ii. 在云产品列表中，选择需开启日志审计功能的云产品，并配置存储时间。

   如果是SLB 7层访问日志、OSS访问日志、PolarDB-X 1.0审计日志，还可以选择**同步到中心**。开启**同步到中心**后，区域化Project将作为中转，不需要存储很长时间，控制台会自动调整成推荐的时间。
   iii. 单击**保存**。
4. 在**多账号配置 > 全局配置**页面，配置账号信息。

   日志审计服务支持手动授权和通过账号密钥服务授权。
   - 手动授权：输入主账号ID，可配置多个。对应的账号权限配置请参见操作步骤。
   - 通过账号密钥辅助授权：在**其他账号授权日志服务采集**文本框中输入其他账号的AK信息及其主账号ID。AK信息不会被保存，仅临时使用。

     此处AK对应的RAM用户需具备RAM读写权限（例如已被授权AliyunRAMFullAccess策略）。
5. 在左侧导航栏，选择**云产品接入 > 接入状态**，查看日志接入状态。

   配置完成后，需要2分钟左右完成初始同步。如果出现异常，请根据页面提示信息进行调整。更多信息，请参见常见问题及错误排查。

停止采集日志

如果您不再需要采集云产品日志但想要保留已采集的日志，可参见以下步骤。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在**云产品接入 > 全局配置**页面，单击右上角的**修改**。
4. 关闭目标日志选项，单击**确定**。

删除审计资源

如果您需要清理并删除日志审计服务相关的所有日志资源（如Logstore、仪表盘、告警等），可参见以下操作。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。

3. 在**云产品接入 > 全局配置**页面，单击右上角的**删除审计资源**。
4. 根据页面提示，完成删除。

常见问题及错误排查

- 如何查看接入状态？

  在**云产品接入 > 接入状态**中查看接入状态。

- 显示账号没有权限或密钥错误，怎么处理？

  请检查账号权限是否配置正确。如果是同一账号下的采集，请参见首次配置，如果是跨账号采集，请参见操作步骤。例如：账号中的sls-audit-service-monitor角色没有被授予**系统策略**下的**ReadOnlyAccess**策略。

- 显示账号没有开启特定服务，怎么处理？

  一般是由于某个云产品没有开启特定服务。更多信息，请参见云产品覆盖及相关资源。例如：已开通云安全中心，但未开通**日志分析**功能。

1.4. 生成威胁情报

日志审计服务支持对接入日志服务的云产品日志进行威胁情报检测，有效识别云产品使用过程中存在的潜在威胁。日志审计服务还支持以告警方式将检测到的异常及时通知给相关人员，提高威胁检查效率和响应速度。

限制与说明

- 对于SLB、OSS、PolarDB-X 1.0等支持区域化存储的云产品，必须开启中心化存储功能后，才支持威胁情报功能。如何开启中心化存储功能，请参见首次配置。
- 对于RDS、操作审计等仅支持中心化存储的云产品，开启威胁情报功能后，系统将在日志审计中心Project下自动创建transit_log Logstore及威胁情报富化的加工任务。关于威胁情报富化的加工任务的更多信息，请参见增值内容函数。
- 日志审计服务中的威胁情报功能是基于阿里云威胁情报服务提供最近30天的威胁情报信息，每天更新一次。如果您需要详细的威胁情报信息，请开通阿里云威胁情报服务进行查询。具体操作，请参见开通免费试用。
- 威胁情报功能所支持的云产品以控制台实际显示为准。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在**云产品接入 > 全局配置**页面，单击**修改**。
4. 在**云产品**列表中，找到目标云产品，打开**威胁情报**开关。
5. 单击**确定**。

威胁情报字段

开启威胁情报功能后，当云产品存在潜在威胁时，对应的云产品日志中将生成威胁情报相关的字段。

- IP地址威胁情报IP地址威胁情报

字段	说明
confidence	威胁情报数据置信度。取值范围为[0, 100]之间的整数值，值越大置信度越高。
severity	威胁级别。包括： ○ 0：无威胁 ○ 1：低危险 ○ 2：中危险 ○ 3：高危险 ○ 4：严重威胁
family	恶意家族，固定取值为空。
ioc_type	IP地址类型，目前仅支持IPv4类型。
ioc_raw	获取威胁情报信息的IP地址
intel_type	风险标签类型。包括： ○ web_attack：网络攻击的IP地址 ○ tor：TOR（Top of Rack）节点的IP地址 ○ mining：挖矿的IP地址 ○ c2：C2 IP地址 ○ malicious：恶意下载源的IP地址 ○ exploit：发起Exploit攻击的IP地址 ○ webshell：发起Webshell攻击的IP地址 ○ scan：网络服务扫描的IP地址 标签之间使用半角分号（;）分隔。
country	IP地址所属的国家
province	IP地址所属的省份
city	IP地址所属的城市
isp	IP地址所属网络的电信运营商

- 域名威胁情报域名威胁情报

字段	说明
confidence	威胁情报数据置信度，取值范围为[0, 100]之间的整数值，值越大置信度越高。

字段	说明
severity	威胁级别。包括： ○ 0：无威胁 ○ 1：低危险 ○ 2：中危险 ○ 3：高危险 ○ 4：严重威胁
family	恶意家族，固定取值为空。
ioc_type	域名，固定取值为domain。
ioc_raw	获取威胁情报信息的域名
intel_type	风险标签类型，标签之间使用半角分号（;）分隔。更多信息，请参见域名风险标签。
root_domain	扫描域名所属的根域名

域名风险标签

风险标签	说明
malware	恶意软件
spy_trojan	间谍木马
worm	蠕虫
ransomware	勒索
backdoor_trojan	后门木马
hacktool	黑客工具
infected_virus	感染型病毒
trojan_dropper	木马释放器
riskware	风险软件
apt	APT
rat_trojan	远控木马
hijack	劫持
macro_virus	宏病毒
porn	色情网站

风险标签	说明
js_miner	网页挖矿
compromised_host	失陷主机
gamble	博彩网站
dnslog_attack	DNSLOG攻击
infostealer	信息盗取
malicious	恶意站点
dga	DGA
botnet	僵尸网络
trojan	木马
bank_trojan	银行木马
adware	广告软件
exploit	漏洞利用
malicious_doc	恶意文档
bootkit_trojan	BootKit木马
script_trojan	脚本木马
virus	病毒
trojan_downloader	木马下载器
rat	远控
ddos_trojan	DDos木马
spam_email	垃圾邮件
miner_pool	矿池
rootkit_trojan	Rootkit木马
private_server	外挂私服
c2	中控
miner	挖矿
malicious_group	恶意团伙

风险标签	说明
sinkhole	Sinkhole

1.5. 审计操作

本文介绍日志审计服务在采集到日志后的审计操作。

前提条件

- 已完成日志审计配置。具体操作，请参见配置日志采集。
- 已有对应权限的账号，权限配置请参见配置权限助手。
  - 如果您需要查询日志、查看报表，则当前登录的账号需要对日志审计服务以及Project下的资源具有读权限。
  - 如果您需要创建报表、创建告警、二次对接，则当前登录的账号需要对日志审计服务以及Project下的资源具有读写权限。

使用审计报表

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在左侧导航栏中，单击**审计报表**。
4. 单击目标报表，进入审计中心。

   您可以在审计中心查看数据报表，仪表盘操作请参见仪表盘。

> **说明** 对于OSS、SLB和PolarDB-X 1.0，如果没有在**全局配置**中开启**同步到中心**功能，则只能在**区域化**页签中查看各个地域下的报表。如果开启了**同步到中心**功能，则可在**中心化**页签中查看除华北1（青岛）外的报表，地域限制请参见日志审计服务概述。

使用审计查询

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在左侧导航栏中，单击**审计查询**。
4. 单击目标云产品，进入查询与分析页面。

   具体的查询、分析操作请参见查询与分析。

> **说明** 对于OSS、SLB和PolarDB-X 1.0，如果没有在**全局配置**中开启**同步到中心**功能，则只能在**区域化**页签中查看各个地域下的日志。如果开启了**同步到中心**功能，则可在**中心化**页签中查看除华北1（青岛）外的日志，地域限制请参见日志审计服务概述。

操作Logstore

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 单击**审计配置 > 云产品接入 > 全局配置**。

4. 单击Project名称，进入日志库列表页面。

后续步骤

完成日志审计后，可通过数据消费、数据投递功能将日志与第三方系统进行对接。

- 数据投递

  使用数据投递与第三方系统对接，包括OSS、MaxCompute、AnalyticDB for MySQL、TSDB、Splunk或其他SIEM。更多信息，请参见数据投递。

- 数据消费

  使用第三方流计算系统实时消费日志，包括Storm、Flume、ARMS、Blink、Logstash、Spark streaming、Cloud Monitor或消费组等。更多信息，请参见实时消费。

1.6. 自定义授权日志采集与同步

在使用日志审计服务进行跨账号采集云产品日志时，需先授予日志服务采集相关云产品日志的权限以及授权多个主账号之间的数据同步。您可以直接使用具备特定权限的RAM用户的密钥或者参见本文进行自定义授权。

背景信息

日志审计服务支持采集同一主账号下的云产品日志，也支持跨主账号采集云产品日志。 进行跨账号采集云产品日志时，当前主账号和其他主账号需要进行双向授权。

> **说明** 当前主账号的授权在创建服务关联角色AliyunServiceRoleForSLSAudit时，自动完成。具体操作，请参见首次配置。其他主账号要使用自定义权限时，需参见本文完成授权。

- 当前主账号允许其他账号同步数据到当前主账号的审计Logstore。
- 其他主账号允许同步数据到当前主账号的审计Logstore。

使用日志审计服务涉及多个授权角色和策略，对应关系如下所示：

- 当前主账号

角色	权限策略
AliyunServiceRoleForSLSAudit	AliyunServiceRolePolicyForSLSAudit

- 其他账号

角色	权限策略
sls-audit-service-monitor	ReadOnlyAccess AliyunLogAuditServiceMonitorAccess

操作步骤

1. 使用其他账号登录RAM 控制台。

   建议使用RAM用户登录，且该RAM用户需具备RAM读写权限（例如已被授予*AliyunRAMFullAccess*策略）。

2. 创建权限策略AliyunLogAuditServiceMonitorAccess。

i. 在左侧导航栏中，选择**权限管理 > 权限策略管理**，单击**创建权限策略**。

ii. 在**新建自定义权限策略**页面，配置如下参数，并单击**确定**。

参数	说明
策略名称	配置为AliyunLogAuditServiceMonitorAccess。
配置模式	选择**脚本配置**。

策略内容	将配置框中的原有脚本替换为如下内容。
	{
	"Version": "1",
	"Statement": [
	{
	"Action": "log:*",
	"Resource": [
	"acs:log:*:*:project/slsaudit-*",
	"acs:log:*:*:app/audit"
	],
	"Effect": "Allow"
	},
	{
	"Action": [
	"rds:ModifySQLCollectorPolicy",
	"vpc:*FlowLog*",
	"drds:*SqlAudit*",
	"kvstore:ModifyAuditLogConfig",
	"polardb:ModifyDBClusterAuditLogCollector"
	],
	"Resource": "*",
	"Effect": "Allow"
	}
	]
	}

3. 创建*sls-audit-service-monitor*角色。

i. 在左侧导航栏中，选择**身份管理 > 角色**，然后单击**创建RAM角色**。

ii. 在**选择类型**配置向导中，选择**阿里云服务**，单击**下一步**。

iii. 在**配置角色**配置向导中，配置如下参数后，然后单击**完成**。

参数	说明
角色类型	选择**普通服务角色**。
角色名称	配置为sls-audit-service-monitor。
选择受信服务	选择**日志服务**。

iv. 在**创建完成**配置向导中，单击**为角色授权**。

4. 授予sls-audit-service-monitor角色AliyunLogAuditServiceMonitorAccess策略。

   在**添加权限**面板中，选择**自定义策略**下的**AliyunLogAuditServiceMonitorAccess**策略和**系统策略**下的**ReadOnlyAccess**策略。单击**确定**。

5. 修改sls-audit-service-monitor角色的信任策略。

   i. 在RAM角色列表中，单击**sls-audit-service-monitor**角色。

   ii. 在**信任策略管理**页签，将配置框中的原有脚本替换为如下内容，然后单击**确定**。

   其中，中心主账号ID请根据实际值替换，您可以在账号中心查看阿里云账号的ID。

```
{
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "中心主账号ID@log.aliyuncs.com",
          "log.aliyuncs.com"
        ]
      }
    }
  ],
  "Version": "1"
}
```

1.7. 日志字段详情

1.7.1. 操作审计

本文介绍操作审计的操作日志的字段详情。

日志字段	说明
__topic__	日志主题，固定为actiontrail_event。
owner_id	阿里云账号ID
event	事件主体内容，JSON格式。事件主体的内容随事件变化。
event.eventId	事件ID，事件的唯一标识。
event.eventName	事件名称
event.eventSource	事件来源
event.eventType	事件类型
event.eventVersion	ActionTrail的数据格式版本，固定为1。

日志字段	说明
event.acsRegion	事件所在的地域
event.requestId	操作云产品的请求ID
event.apiVersion	相关API的版本
event.errorMessage	事件失败的错误信息
event.serviceName	事件相关的服务名称
event.sourceIpAddress	事件相关的源IP地址
threat_sourceIpAddress	事件相关的源IP地址的威胁情报。更多信息，请参见威胁情报字段。
event.userAgent	事件相关的客户端
event.requestParameters.HostId	请求参数中的主机ID
event.requestParameters.Name	请求参数中的名称
event.requestParameters.Region	请求参数中的地域
event.userIdentity.accessKeyId	请求所使用的AccessKey ID
event.userIdentity.accountId	请求账号的ID
event.userIdentity.principalId	请求账号的凭证ID
event.userIdentity.type	请求账号的类型
event.userIdentity.userName	请求账号的名称
event.errorCode	事件失败的错误码
addionalEventData.isMFAChecked	登录账号是否开启MFA
addionalEventData.loginAccount	登录账号

1.7.2. 对象存储

本文介绍对象存储OSS相关日志的字段详情。

- 访问日志

  记录对应Bucket的所有访问日志，实时采集。

日志字段	说明
__topic__	日志主题，固定为oss_access_log。
owner_id	阿里云账号ID

日志字段	说明
region	Bucket所在地域
access_id	访问者的阿里云AccessKey ID
time	访问时间，即OSS收到请求的时间。如果需要时间戳，可使用__time__。
owner_id	Bucket拥有者的阿里云账号ID
User-Agent	HTTP的User-Agent头
logging_flag	是否开启了日志定期导出到OSS Bucket的功能
bucket	Bucket名称
content_length_in	请求头中Content-Length的值，单位：Byte。
content_length_out	响应头中Content-Length的值，单位：Byte。
object	OSS Object，URL编码。查询时，您可使用select url_decode(object)解码。
object_size	OSS Object大小，单位：Byte。
operation	访问类型。更多信息，请参见访问类型。
request_uri	请求URI，包括query-string，路径是URL编码。查询时，您可使用select url_decode(request_uri)解码。
error_code	OSS返回的错误码。更多信息，请参见错误响应。
request_length	HTTP请求的大小，包括header，单位：Byte。
client_ip	发起请求的IP地址，即客户端IP地址、其网络防火墙或者Proxy的IP地址。
threat_client_ip	请求客户端的IP地址的威胁情报。更多信息，请参见威胁情报字段。
response_body_length	响应Body大小，不包括header。
http_method	HTTP请求方法
referer	请求的HTTP Referer
requester_id	请求者的阿里云账号ID，匿名访问时为短划线（-）。
request_id	请求ID
response_time	请求响应时间，单位：毫秒。
server_cost_time	OSS服务器处理时间，即OSS服务器处理本次请求所花的时间，单位：毫秒。
http_type	HTTP请求类型，HTTP或HTTPS。

日志字段	说明
sign_type	签名类型。更多信息，请参见签名类型。
http_status	HTTP状态，即OSS请求返回的HTTP状态。
sync_request	同步请求类型。更多信息，请参见同步请求类型。
bucket_storage_type	Bucket存储类型。更多信息，请参见Bucket存储类型。
host	请求访问域名
vpc_addr	访问OSS的域名对应的VPC IP地址
vpc_id	VPC ID
delta_data_size	OSS Object大小的变化量。如果没有变化则为0；如果不是上传请求，则为短划线（-）。
acc_access_region	如果是传输加速请求，这个字段为请求接入点所在地域名，否则为短划线（-）。
restore_priority	Restore恢复优先级

- 批量删除日志

  记录批量删除Object时具体的删除信息，实时采集。

  > **说明** 当您调用DeleteObjects时，访问日志中会有一条请求记录。但因为删除的文件信息存放在请求的HTTP Body中，访问日志中的object字段显示为短划线（-）。如果需要查看具体的删除文件的列表，可查看批量删除的日志，通过request_id关联。

日志字段	说明
__topic__	日志主题，固定为oss_batch_delete_log。
owner_id	阿里云账号ID
region	Bucket所在地域
client_ip	发起请求的IP地址，例如客户端IP地址、网络防火墙或者Proxy的IP地址。
user_agent	HTTP的User-Agent头
bucket	Bucket名称
error_code	OSS返回的错误码。更多信息，请参见同步请求类型。
request_length	请求Body大小，包括header，单位：Byte。
response_body_length	响应Body大小，不包括header。

日志字段	说明
object	OSS Object，URL编码。查询时，您可使用select url_decode(object)解码。
object_size	请求对象的大小，单位：Byte。
operation	访问类型。更多信息，请参见访问类型。
bucket_location	Bucket所在集群
http_method	HTTP请求方法
referer	请求的HTTP Referer
request_id	请求ID
http_status	OSS请求返回的HTTP状态。
sync_request	同步请求类型。更多信息，请参见同步请求类型。
request_uri	请求URI，包括query-string，路径是URL编码。查询时，您可使用select url_decode(request_uri)解码。
host	请求访问域名
logging_flag	是否开启了日志定期导出到OSS Bucket的功能。
server_cost_time	OSS服务器处理时间，单位：毫秒。
owner_id	Bucket拥有者的阿里云账号ID
requester_id	请求者的阿里云账号ID，匿名访问为短划线（-）。
delta_data_size	OSS Object大小的变化量。如果没有变化则为0；如果不是上传请求，则为短划线（-）。

- 每小时计量日志

  记录特定Bucket每个小时累计的计量日志，延迟为数小时，用于辅助分析。

日志字段	说明
__topic__	日志主题，固定为oss_metering_log。
owner_id	Bucket拥有者的阿里云账号ID
bucket	Bucket名称
cdn_in	CDN流入量，单位：Byte。
cdn_out	CDN流出量，单位：Byte。
get_request	GET请求次数

日志字段	说明
intranet_in	内网流入量，单位：Byte。
intranet_out	内网流出量，单位：Byte。
network_in	外网流入量，单位：Byte。
network_out	外网流出量，单位：Byte。
put_request	PUT请求次数
storage_type	Bucket存储类型。更多信息，请参见Bucket存储类型。
storage	Bucket存储量，单位：Byte。
metering_datasize	非标准存储的计量数据大小
process_img_size	处理的图像大小，单位：Byte。
process_img	处理的图像
sync_in	同步流入量，单位：Byte。
sync_out	同步流出量，单位：Byte。
start_time	计量开始时间
end_time	计量截止时间
region	Bucket所在地域

Bucket存储类型

存储类型	描述
standard	标准存储类型
archive	归档存储类型
infrequent_access	低频访问存储类型

每个操作的具体信息，请参见API概览。

访问类型

操作值	描述
AbortMultiPartUpload	中止断点上传
AppendObject	追加上传文件
CompleteUploadPart	完成断点上传

操作值	描述
CopyObject	复制文件
DeleteBucket	删除Bucket
DeleteLiveChannel	删除LiveChannel
DeleteObject	删除文件
DeleteObjects	删除多个文件
GetBucket	列举文件
GetBucketAcl	获取Bucket权限
GetBucketCors	查看Bucket的CORS规则
GetBucketEventNotification	获取Bucket通知配置
GetBucketInfo	查看Bucket信息
GetBucketLifecycle	查看Bucket的生命周期规则
GetBucketLocation	查看Bucket地域
GetBucketLog	查看Bucket访问日志配置
GetBucketReferer	查看Bucket防盗链设置
GetBucketReplication	查看跨区域复制
GetBucketReplicationProgress	查看跨区域复制进度
GetBucketStat	获取Bucket的相关信息
GetBucketWebSite	查看Bucket的静态网站托管状态
GetLiveChannelStat	获取LiveChannel状态信息
GetObject	读取文件
GetObjectAcl	获取文件访问权限
GetObjectInfo	获取文件信息
GetObjectMeta	查看元信息
GetObjectSymlink	获取symlink文件的详细信息
GetPartData	获取断点文件块数据
GetPartInfo	获取断点文件块信息

操作值	描述
GetProcessConfiguration	获取Bucket图片处理配置
GetService	列举Bucket
HeadBucket	查看Bucket信息
HeadObject	查看文件信息
InitiateMultipartUpload	初始化断点上传文件
ListMultiPartUploads	列举断点事件
ListParts	列举断点块状态
PostObject	表单上传文件
PostProcessTask	提交相关的数据处理，例如截图等
PostVodPlaylist	创建LiveChannel点播列表
ProcessImage	图片处理
PutBucket	创建Bucket
PutBucketCors	设置Bucket的CORS规则
PutBucketLifecycle	设置Bucket的Lifecycle配置
PutBucketLog	设置Bucket访问日志
PutBucketWebSite	设置Bucket静态网站托管模式
PutLiveChannel	创建LiveChannel
PutLiveChannelStatus	设置LiveChannel状态
PutObject	上传文件
PutObjectAcl	修改文件访问权限
PutObjectSymlink	创建symlink文件
RedirectBucket	Bucket Endpoint重定向
RestoreObject	解冻文件
UploadPart	断点上传文件
UploadPartCopy	复制文件块
get_image_exif	获取图片的exif信息

操作值	描述
get_image_info	获取图片的长宽等信息
get_image_infoexif	获取图片的长宽以及exif信息
get_style	获取Bucket样式
list_style	列举Bucket的样式
put_style	创建Bucket样式

同步请求类型

同步请求类型	描述
短划线（-）	一般请求
cdn	CDN回源

关于签名的更多信息，请参见用户签名验证。

签名类型

签名类型	描述
NotSign	未签名
NormalSign	一般方式签名
UriSign	通过URL签名
AdminSign	管理员账号

1.7.3. 云数据库RDS

本文介绍云数据库RDS SQL审计日志、慢日志和性能日志的字段详情。

SQL审计日志

日志字段	说明
__topic__	日志主题，固定为rds_audit_log。
owner_id	阿里云账号ID
region	实例所在地域
instance_name	RDS实例名
instance_id	RDS实例ID

日志字段	说明
db_type	RDS实例类型
db_version	实例版本号
check_rows	扫描的行数
db	数据库名
fail	SQL执行是否出错。包括： • 0：成功 • 1：失败
client_ip	访问RDS实例的客户端IP地址
threat_client_ip	访问RDS实例的客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
latency	执行SQL操作后，多久返回结果，单位：微秒。
origin_time	执行操作的时间点
return_rows	返回的行数
sql	执行的SQL语句
thread_id	线程ID
user	执行SQL的用户名
update_rows	更新行数

慢日志

日志字段	说明
__topic__	日志主题，固定为rds_slow_log
owner_id	阿里云账号ID
region	实例所在地域
instance_name	RDS实例名
instance_id	RDS实例ID
db_type	RDS实例类型
db_version	实例版本号
db_name	数据库名

日志字段	说明
rows_examined	扫描的行数
rows_sent	返回的行数
start_time	开始执行的时间
query_time	执行的耗时，单位：秒。
lock_time	锁等待的耗时，单位：秒。
user_host	客户端信息
query_sql	慢日志SQL语句

性能日志

指标名称	说明
mysql_perf_active_session	活跃连接数，单位：个。
mysql_perf_com_delete	平均每秒Delete语句执行次数
mysql_perf_com_insert	平均每秒Insert语句执行次数
mysql_perf_com_insert_select	平均每秒Insert Select语句执行次数
mysql_perf_com_replace	平均每秒Replace语句执行次数
mysql_perf_com_replace_select	平均每秒Replace Select语句执行次数
mysql_perf_com_select	平均每秒Select语句执行次数
mysql_perf_com_update	平均每秒Update语句执行次数
mysql_perf_conn_usage	实例连接使用率，单位：百分比。
mysql_perf_cpu_usage	实例CPU使用率，单位：百分比。
mysql_perf_data_size	实例数据使用量，单位：MB。
mysql_perf_disk_usage	实例磁盘使用率，单位：百分比。
mysql_perf_ibuf_dirty_ratio	缓冲池脏块的百分率，单位：百分比。
mysql_perf_ibuf_read_hit	缓冲池的读命中率
mysql_perf_ibuf_request_r	平均每秒钟从InnoDB缓冲池的读次数
mysql_perf_ibuf_request_w	平均每秒钟向InnoDB缓冲池的写次数
mysql_perf_ibuf_use_ratio	缓冲池的利用率，单位：百分比。

指标名称	说明
mysql_perf_inno_data_read	InnoDB平均每秒钟读取的数据量，单位：KB。
mysql_perf_inno_data_written	InnoDB平均每秒钟写入的数据量，单位：KB。
mysql_perf_inno_row_delete	平均每秒从InnoDB表删除的行数
mysql_perf_inno_row_insert	平均每秒从InnoDB表插入的行数
mysql_perf_inno_row_readed	平均每秒从InnoDB表读取的行数
mysql_perf_inno_row_update	平均每秒从InnoDB表更新的行数
mysql_perf_innodb_log_write_requests	平均每秒日志写请求数
mysql_perf_innodb_log_writes	平均每秒向日志文件的物理写次数
mysql_perf_innodb_os_log_fsyncs	平均每秒向日志文件完成的fsync()写数量
mysql_perf_ins_size	实例磁盘使用量，单位：MB。
mysql_perf_iops	IOPS，单位：次/秒。
mysql_perf_iops_usage	实例IOPS使用率，单位：百分比。
mysql_perf_kbytes_received	平均每秒钟的输入流量，单位：KB。
mysql_perf_kbytes_sent	平均每秒钟的输出流量，单位：KB。
mysql_perf_log_size	实例binlog使用量，单位：MB。
mysql_perf_mem_usage	实例内存使用率，单位：百分比。
mysql_perf_open_tables	当前打开表数量
mysql_perf_other_size	实例其他空间使用量，单位：MB。
mysql_perf_qps	平均每秒SQL语句执行次数
mysql_perf_slow_queries	平均每秒慢查询数量
mysql_perf_tb_tmp_disk	MySQL执行语句时每秒在磁盘上自动创建的临时表的数量
mysql_perf_threads_connected	MySQL线程连接数
mysql_perf_threads_running	MySQL活跃线程
mysql_perf_tmp_size	实例临时空间使用量，单位：MB。
mysql_perf_total_session	总连接数，单位：个。

指标名称	说明
mysql_perf_tps	平均每秒事务数

1.7.4. PolarDB MySQL云原生数据库

本文介绍PolarDB MySQL云原生数据库审计日志、慢日志和性能日志的字段详情。

审计日志

日志字段	说明
__topic__	日志主题，固定为polardb_audit_log。
owner_id	阿里云账号ID
region	PolarDB MySQL集群所在地域
cluster_id	PolarDB MySQL集群ID
node_id	PolarDB MySQL节点ID
check_rows	扫描的行数
db	数据库名称
fail	SQL执行是否出错。包括： • 0：成功 • 1：失败
client_ip	访问PolarDB MySQL集群的客户端IP地址
threat_client_ip	访问PolarDB MySQL集群的客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
latency	执行SQL操作后，多久返回结果，单位：微秒。
origin_time	执行操作的时间
return_rows	返回的行数
sql	执行的SQL语句
thread_id	线程ID
user	执行SQL的用户名
update_rows	更新行数

慢日志

日志字段	说明
__topic__	日志主题，固定为rds_slow_log。
owner_id	阿里云账号ID
region	PolarDB MySQL集群所在地域
cluster_id	PolarDB MySQL集群ID
node_id	PolarDB MySQL节点ID
db_type	PolarDB数据库类型
db_name	PolarDB数据库名
version	PolarDB数据库版本号
rows_examined	扫描的行数
rows_sent	返回的行数
start_time	开始执行的时间
query_time	执行的耗时，单位：秒。
lock_time	锁等待的耗时，单位：秒。
user_host	客户端信息
query_sql	慢日志SQL语句

性能日志

指标名称	说明
mysql_perf_active_session	每秒活跃连接数
mysql_perf_binlog_size	本地Binlog使用量，单位：MB。
mysql_perf_com_delete	每秒DELETE语句执行次数
mysql_perf_com_delete_multi	每秒Multi-DELETE语句执行次数
mysql_perf_com_insert	每秒INSERT语句执行次数
mysql_perf_com_insert_select	每秒INSERT-SELECT语句执行次数
mysql_perf_com_replace	每秒REPLACE语句执行次数
mysql_perf_com_replace_select	每秒REPLACE-SELECT语句执行次数
mysql_perf_com_select	每秒SELECT语句执行次数

指标名称	说明
mysql_perf_com_update	每秒UPDATE语句执行次数
mysql_perf_com_update_multi	每秒Multi-UPDATE语句执行次数
mysql_perf_cpu_ratio	CPU使用率，单位：百分比。
mysql_perf_created_tmp_disk_tables	每秒创建临时表个数
mysql_perf_data_size	数据空间使用量，单位：MB。
mysql_perf_innodb_buffer_dirty_ratio	缓冲池脏块率，单位：百分比。
mysql_perf_innodb_buffer_read_hit	缓冲池读命中率，单位：百分比。
mysql_perf_innodb_buffer_use_ratio	缓冲池使用率，单位：百分比。
mysql_perf_innodb_data_read	每秒从存储引擎读取数据量，单位：Byte。
mysql_perf_innodb_data_reads	每秒缓冲池读取次数
mysql_perf_innodb_data_writes	每秒缓冲池写次数
mysql_perf_innodb_data_written	每秒往存储引擎写入数据量，单位：Byte。
mysql_perf_innodb_log_write_requests	每秒日志写请求数
mysql_perf_innodb_os_log_fsyncs	每秒向日志文件完成的fsync()写数量
mysql_perf_innodb_rows_deleted	每秒删除的行数
mysql_perf_innodb_rows_inserted	每秒插入的行数
mysql_perf_innodb_rows_read	每秒读取的行数
mysql_perf_iops	每秒IOPS
mysql_perf_iops_r	每秒读IOPS
mysql_perf_iops_throughput	每秒总IO吞吐量，单位：MB。
mysql_perf_iops_throughput_r	每秒读IO吞吐量，单位：MB。
mysql_perf_iops_throughput_w	每秒写IO吞吐量，单位：MB。

指标名称	说明
mysql_perf_iops_w	每秒写IOPS
mysql_perf_kbytes_received	每秒输入流量，单位：KB。
mysql_perf_kbytes_sent	每秒输出流量，单位：KB。
mysql_perf_mem_ratio	内存使用率，单位：百分比。
mysql_perf_mps	每秒数据操作数
mysql_perf_other_log_size	其他日志使用量，单位：MB。
mysql_perf_qps	每秒请求数
mysql_perf_redolog_size	本地Redolog使用量，单位：MB。
mysql_perf_slow_queries	平均每秒慢查询数量
mysql_perf_sys_dir_size	系统空间使用量，单位：MB。
mysql_perf_tmp_dir_size	临时空间使用量，单位：MB。
mysql_perf_total_session	当前平均总连接数
mysql_perf_tps	每秒事务数

1.7.5. 分布式关系型数据库PolarDB-X 1.0

本文介绍分布式关系型数据库PolarDB-X 1.0 SQL审计日志的字段详情。

字段名称	字段说明
__topic__	日志主题，固定为drds_audit_log。
instance_id	PolarDB-X 1.0实例ID
instance_name	PolarDB-X 1.0实例名
owner_id	阿里云账户ID
region	PolarDB-X 1.0实例所在地域
db_name	PolarDB-X 1.0数据库名
user	执行SQL的用户名
client_ip	访问PolarDB-X 1.0实例的客户端IP地址
threat_client_ip	访问PolarDB-X 1.0实例的客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。

字段名称	字段说明
client_port	访问PolarDB-X 1.0实例的客户端端口
sql	执行的SQL语句
trace_id	SQL执行的TRACE ID。如果是事务，则显示为跟踪ID、短划线（-）和数字，例如drdsabcdxyz-1。
sql_code	模板SQL的HASH值
hint	SQL执行的HINT
table_name	查询涉及的表名。多表之间以英文逗号（,）分隔。
sql_type	SQL类型。包括Select、Insert、Update、Delete、Set、Alter、Create、Drop、Truncate、Replace和Other。
sql_type_detail	SQL解析器名称
response_time	响应时间，单位：微秒。
affect_rows	SQL执行返回的行数。增删改时表示影响的行数，查询语句表示返回的行数。
fail	SQL执行是否出错。包括： • 0：成功 • 1：失败
sql_time	SQL开始执行的时间

1.7.6. 负载均衡

本文介绍负载均衡七层访问日志的字段详情。

日志字段	说明
owner_id	阿里云账号ID
region	实例所在地域
instance_id	实例ID
instance_name	实例名
network_type	网络类型，包括： • VPC：专有网络 • Classic：经典网络
vpc_id	VPC ID
body_bytes_sent	发送给客户端的Body字节数

日志字段	说明
client_ip	客户端IP地址
threat_client_ip	客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
client_port	客户端端口
host	优先从请求参数中获取host。如果获取不到，则从host header中取值。如果还是获取不到，则以处理请求的后端服务器IP地址作为host。
http_host	请求报文host header的内容
http_referer	Proxy收到的请求报文中HTTP referer header的内容
http_user_agent	Proxy收到的请求报文中HTTP user-agent的内容
http_x_forwarded_for	Proxy收到的请求报文中HTTP forwarded-for header的内容
http_x_real_ip	真实的客户端IP地址
read_request_time	Proxy读取请求的时间，单位：毫秒。
request_length	请求报文的长度，包括startline、HTTP header和HTTP Body。
request_method	请求方法
request_time	Proxy收到第一个请求报文的时间到proxy返回应答之间的间隔时间，单位：秒。
request_uri	Proxy收到的请求报文的URI
scheme	请求的Scheme
server_protocol	Proxy收到的HTTP协议的版本
slb_vport	SLB的监听端口
slbid	SLB实例ID
ssl_cipher	使用的cipher
ssl_protocol	建立SSL连接所使用的协议
status	Proxy应答报文的状态
tcpinfo_rtt	客户端的tcp rtt时间，单位：微秒。
time	日志记录时间
upstream_addr	后端服务器的IP地址和端口
upstream_response_time	从SLB向后端建立连接开始到接受完数据然后关闭连接为止的时间，单位：秒。

日志字段	说明
upstream_status	Proxy收到的后端服务器的响应状态码
vip_addr	VIP地址
write_response_time	Proxy写的响应时间，单位：毫秒。

1.7.7. 堡垒机

本文介绍堡垒机操作日志的字段详情。

日志字段	说明
__topic__	日志主题
owner_id	阿里云账号ID
content	日志内容
event_type	事件类型。更多信息，请参见事件类型。
instance_id	堡垒机实例ID
log_level	日志级别
resource_address	资源地址
resource_name	资源名称
result	操作结果
session_id	会话ID
user_client_ip	用户来源IP地址
threat_user_client_ip	用户来源IP地址的威胁情报。更多信息，请参见威胁情报字段。
user_id	用户ID
user_name	用户名称

事件类型

值	含义
cmd.Command	字符命令
file.Upload	上传文件
file.Download	下载文件

值	含义
file.Rename	重命名文件
file.Delete	删除文件
file.DeleteDir	删除目录
file.CreateDir	创建目录
graph.Text	图形文字
graph.Keyboard	键盘事件

1.7.8. Web应用防火墙

本文介绍Web应用防火墙访问日志的字段详情。

字段	说明
__topic__	日志主题，固定为waf_access_log。
owner_id	阿里云账号ID
acl_action	WAF精准访问控制规则行为，例如pass、drop、captcha。 空值或短划线（-）也表示pass。
block_action	触发拦截的WAF防护类型，详细说明如下： • tmd：CC攻击防护 • waf：Web应用攻击防护 • acl：精准访问控制 • geo：地域封禁 • antifraud：数据风控 • antibot：防爬封禁
body_bytes_sent	发送给客户端的HTTP Body字节数
cc_action	CC防护策略行为，例如none、challenge、pass、close、captcha、wait、login、n等。
cc_blocks	是否被CC防护功能拦截，包括： • 1表示拦截。 • 其他值均表示通过。
content_type	访问请求内容类型
host	源站服务器

字段	说明
http_cookie	访问请求头部中带有的访问来源客户端Cookie信息
http_referer	访问请求头部中带有的访问请求的来源URL信息。如果无来源URL信息，则显示为短划线（-）。
http_user_agent	访问请求头部中的User Agent字段，一般包含来源客户端浏览器标识、操作系统标识等信息。
http_x_forwarded_for	访问请求头部中带有的XFF头信息，用于识别通过HTTP代理或负载均衡方式连接到Web服务器的客户端最原始的IP地址。
https	访问请求是否为HTTPS请求，包括： • true：HTTPS请求 • false：HTTP请求
matched_host	匹配到的已接入WAF防护配置的域名，可能是泛域名。如果无法匹配到相关域名配置，则显示短划线（-）。
querystring	请求中的查询字符串
real_client_ip	访问的客户端的真实IP地址。如果无法获取，则显示为短划线（-）。
threat_real_client_ip	访问客户端的真实IP地址的威胁情报。更多信息，请参见威胁情报字段。
region	WAF实例地域信息
remote_addr	访问请求的客户端IP地址
remote_port	访问请求的客户端端口
request_length	访问请求长度，单位：字节。
request_method	访问请求的HTTP请求方法
request_path	请求的相对路径（不包含查询字符串）
request_time_msec	访问请求时间，单位：毫秒。
request_traceid	WAF记录的访问请求唯一ID标识
server_protocol	源站服务器响应的协议及版本号
status	WAF返回给客户端的HTTP响应状态信息
time	访问请求发生的时间
ua_browser	访问请求来源的浏览器信息
ua_browser_family	访问请求来源所属浏览器系列
ua_browser_type	访问请求来源的浏览器类型

字段	说明
ua_browser_version	访问请求来源的浏览器版本
ua_device_type	访问请求来源客户端的设备类型
ua_os	访问请求来源客户端的操作系统信息
ua_os_family	访问请求来源客户端所属操作系统系列
upstream_addr	WAF使用的回源地址列表，格式为IP:Port。 多个地址之间以英文逗号（,）分隔。
upstream_response_time	源站响应WAF请求的时间，单位：秒。 如果返回短划线（-），表示响应超时。
upstream_status	源站返回给WAF的响应状态。 如果返回短划线（-），表示没有响应，例如该请求被WAF拦截。
user_id	阿里云账号ID
waf_action	Web攻击防护策略行为。包括： • block表示拦截。 • by pass或其它值均表示放行。
bypass_matched_ids	客户端请求命中的WAF放行类规则的ID，具体包括白名单规则、设置了放行动作的自定义防护策略规则。 如果请求同时命中了多条放行类规则，该字段会记录所有命中的规则ID。多个规则ID间使用半角逗号（,）分隔。
final_plugin	WAF对客户端请求最终执行的防护动作（final_action）对应的防护模块。 如果一个请求未触发任何防护模块（包括命中了放行类规则、客户端完成滑块或JS校验后触发放行的情况），则不会记录该字段。 如果一个请求同时触发了多个防护模块，则仅记录最终执行的防护动作（final_action）对应的防护模块。
final_action	WAF对客户端请求最终执行的防护动作。 如果一个请求未触发任何防护模块（包括命中了放行类规则、客户端完成滑块或JS校验后触发放行的情况），则不会记录该字段。 如果一个请求同时触发了多个防护模块，则仅记录最终执行的防护动作。防护动作的优先级由高到低依次为：拦截（block）、严格滑块校验（captcha_strict）、普通滑块校验（captcha）和JS校验（js）。
final_rule_id	WAF对客户端请求最终应用的防护规则的ID，即final_action对应的防护规则的ID。

字段	说明
final_rule_type	WAF对客户端请求最终应用的防防护规则（final_rule_id）的子类型。例如在*final_plugin:waf*类型下有final_rule_type:sqli、final_rule_type:xss等细分的规则类型。
waf_rule_id	匹配的WAF的相关规则ID
waf_rule_type	WAF对客户端请求最终应用的防防护规则（final_rule_id）的子类型。 例如在`final_plugin:waf`类型下有`final_rule_type:sqli`、`final_rule_type:xss`等细分的规则类型。
acl_rule_type	客户端请求命中的IP黑名单、自定义防护策略（ACL访问控制）规则的类型。取值： • custom：表示自定义防护策略（ACL访问控制）规则。 • blacklist：表示IP黑名单规则。
cc_rule_id	CC攻击规则拦截ID。
cc_rule_type	客户端请求命中的CC安全防护、自定义防护策略（CC攻击防护）规则的类型。取值： • custom：表示自定义防护策略（CC攻击防护）规则。 • system：表示CC安全防护规则。
ssl_cipher	SSL加密套件
ssl_protocol	SSL协议版本

1.7.9. 云防火墙

本文介绍云防火墙的互联网边界防火墙流量日志和VPC边界防火墙流量日志的字段详情。

互联网边界防火墙流量日志

日志字段	说明
__topic__	日志主题，固定为cloudfirewall_access_log。
owner_id	阿里云账号ID
log_type	日志类型，固定为internet_log。
app_name	访问流量应用的协议名称，例如HTTPS、NTP、SIP、SMB、NFS、DNS等，未知时为Unknown。
direction	流量的方向，包括： • in：入方向 • out：出方向

日志字段	说明
domain	域名
dst_ip	目的IP地址
threat_dst_ip	目的IP地址的威胁情报。更多信息，请参见威胁情报字段。
dst_port	目的端口
end_time	会话结束时间，Unix时间戳格式，单位：秒。
in_bps	入流量大小，单位：bps。
in_packet_bytes	入流量总字节数
in_packet_count	入流量总报文数
in_pps	入流量大小，单位：pps。
ip_protocol	IP协议类型，支持TCP或UDP协议。
out_bps	出方向流量大小，单位：bps。
out_packet_bytes	出方向总流量字节数
out_packet_count	出方向报文数
out_pps	出方向流量大小，单位：pps。
region_id	访问流量所属的地域
rule_result	命中规则结果，包括： • pass：通过 • alert：告警 • drop：丢弃
src_ip	源IP地址
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口，流量数据发出的主机端口
start_time	会话开始时间，Unix时间戳，单位：秒。
start_time_min	会话开始时间的分钟取整，Unix时间戳，单位：秒。
tcp_seq	TCP序列号
total_bps	出入方向访问总流量的大小，单位：bps。
total_packet_bytes	出入方向的访问总流量，单位：字节。

日志字段	说明
total_packet_count	总流量，以报文数表示。
total_pps	出入方向访问总流量的大小，单位：pps。
src_private_ip	私网IP地址
vul_level	漏洞风险等级，包括： • 1：低危 • 2：中危 • 3：高危
url	URL地址
acl_rule_id	命中ACL的规则ID
ips_rule_id	命中IPS的规则ID
ips_ai_rule_id	命中AI的规则ID
ips_rule_name	命中IPS的规则名称（中文）
ips_rule_name_en	命中IPS的规则名称（英文）
attack_type_name	攻击类型的名称（中文）
attack_type_name_en	攻击类型的名称（英文）
proxy_acl_rule_id	命中正向代理ACL的规则ID

VPC边界防火墙流量日志

日志字段	说明
__topic__	主题，固定为cloudfirewall_vpc_log。
log_type	日志类型，固定为vpc_firewall_log。
aliuid	阿里云账号ID
app_name	应用名。值可能为HTTPS、NTP、SIP、SMB、NFS、DNS等，未知时为Unknown。
domain	域名
dst_ip	目的IP地址
dst_port	目的端口
dst_region	目的地域ID

日志字段	说明
dst_network_instance_id	目的网络实例ID，可能为VPC、VBR、CCN的网络实例ID。
end_time	会话结束时间，Unix时间戳格式，单位：秒。
firewall_id	VPC防火墙ID。 • 云企业网场景下，显示的是云企业网ID，例如cen-6srj4tvjjovhbc。 • 高速通道场景下，显示的是防火墙实例ID，例如vfw-123。
in_bps	入流量大小，单位：bps。
in_packet_bytes	入流量总子节数
in_packet_count	入流量总报文数
in_pps	入流量大小，单位：pps。
ip_protocol	IP协议类型，TCP或UDP。
out_bps	出方向流量大小，单位：bps。
out_packet_bytes	出方向总流量字节数
out_packet_count	出方向报文数
out_pps	出方向流量大小，单位：pps。
rule_result	命中规则结果。包括： • pass：通过 • alert：告警 • drop：丢弃
src_ip	源IP地址
src_port	源端口
src_region	源地域ID
src_network_instance_id	源网络实例ID，可能为VPC、VBR、CCN的网络实例ID。
start_time	会话开始时间，Unix时间戳格式，单位：秒。
start_time_min	会话开始时间的分钟取整，Unix时间戳格式，单位：秒。
tcp_seq	TCP序列号
total_bps	总流量大小，单位：bps。
total_packet_bytes	流量总字节数，单位：字节。
total_packet_count	流量总报文数

日志字段	说明
total_pps	总流量大小，单位：pps。
vul_level	漏洞风险等级，包括： • 1：低危 • 2：中危 • 3：高危
ips_rule_name	命中IPS规则中文名称
ips_rule_name_en	命中IPS规则英文名称
attack_type_name	攻击类型中文名称
attack_type_name_en	攻击类型英文名称

1.7.10. DDoS防护

本文介绍DDoS防护访问日志的字段详情。

DDoS高防（新BGP）

日志字段	说明
__topic__	日志主题，固定为ddoscoo_access_log。
owner_id	阿里云账号ID
body_bytes_sent	请求Body的大小，单位：字节。
cc_action	CC防护策略行为，例如none、challenge、pass、close、captcha、wait、login等。
cc_phase	CC防护策略，包括seccookie、server_ip_blacklist、static_whitelist、server_header_blacklist、server_cookie_blacklist、server_args_blacklist、qps_overmax等。
cc_blocks	是否被CC防护策略阻断。包括： • 1表示阻断。 • 其他内容表示通过。
content_type	内容类型
host	源网站
http_cookie	请求Cookie
http_referer	请求Referer。如果HTTP Header中没有Referer，则显示为短划线（-）。

日志字段	说明
http_user_agent	请求User Agent
http_x_forwarded_for	通过代理跳转的上游用户的IP地址。
https	该请求是否为HTTPS请求。取值如下： • true：该请求是HTTPS请求。 • false：该请求是HTTP请求。
isp_line	线路信息，例如BGP、电信、联通等。
matched_host	匹配到的源站，可能是泛域名。如果未匹配，则显示为短划线（-）。
real_client_ip	客户端的真实IP地址。如果获取不到，则显示为短划线（-）。
threat_real_client_ip	客户端的真实IP地址的威胁情报。更多信息，请参见威胁情报字段。
remote_addr	请求连接的客户端IP地址
remote_port	请求连接的客户端端口号
request_length	请求长度，单位：字节。
request_method	请求的HTTP方法
request_time_msec	请求时间，单位：微秒。
request_uri	请求路径
server_name	匹配到的host名。如果未匹配，则显示为default。
status	HTTP状态
time	时间
ua_browser	浏览器
ua_browser_family	浏览器系列
ua_browser_type	浏览器类型
ua_device_type	客户端设备类型
ua_os	客户端操作系统
ua_os_family	客户端操作系统系列
upstream_addr	回源地址列表，格式为IP:Port。 多个地址之间以英文逗号（,）分隔。
upstream_ip	实际回源地址IP地址

日志字段	说明
upstream_response_time	回源响应时间，单位：秒。
upstream_status	回源请求HTTP状态

DDoS高防（国际）

日志字段	说明
__topic__	日志主题，固定为ddosdip_access_log。
owner_id	阿里云账号ID
body_bytes_sent	请求Body的大小，单位：字节。
cc_action	CC防护策略行为，例如none、challenge、pass、close、captcha、wait、login等。
cc_phase	CC防护策略，包括seccookie、server_ip_blacklist、static_whitelist、server_header_blacklist、server_cookie_blacklist、server_args_blacklist、qps_overmax等。
cc_blocks	是否被CC防护策略阻断。包括： • 1表示阻断。 • 其他内容表示通过。
content_type	内容类型
host	源网站
http_cookie	请求Cookie
http_referer	请求Referer。如果HTTP Header中没有Referer，则显示为短划线（-）。
http_user_agent	请求User Agent
http_x_forwarded_for	通过代理跳转的上游用户的IP地址。
https	该请求是否为HTTPS请求。取值如下： • true：该请求是HTTPS请求。 • false：该请求是HTTP请求。
isp_line	线路信息，例如BGP、电信、联通等。
matched_host	匹配到的源站，可能是泛域名。如果未匹配，则显示为短划线（-）。
real_client_ip	客户端的真实IP地址。如果获取不到，则显示为短划线（-）。
threat_real_client_ip	客户端的真实IP地址的威胁情报。更多信息，请参见威胁情报字段。
remote_addr	请求连接的客户端IP地址

日志字段	说明
remote_port	请求连接的客户端端口号
request_length	请求长度，单位：字节。
request_method	请求的HTTP方法
request_time_msec	请求时间，单位：微秒。
request_uri	请求路径
server_name	匹配到的Host名。如果未匹配，则显示为default。
status	HTTP状态
time	时间
ua_browser	浏览器
ua_browser_family	浏览器系列
ua_browser_type	浏览器类型
ua_device_type	客户端设备类型
ua_os	客户端操作系统
ua_os_family	客户端操作系统系列
upstream_addr	回源地址列表，格式为IP:Port。 多个地址之间以英文逗号（,）分隔。
upstream_ip	实际回源地址IP地址
upstream_response_time	回源响应时间，单位：秒。
upstream_status	回源请求HTTP状态

DDoS原生

字段	说明
__topic__	日志主题，固定为ddosbqp_access_log。
data_type	日志类型
event_type	事件类型
ip	事件发生的IP地址
subnet	代播的网段

字段	说明
event_time	事件发生时的时间，例如2020-01-01。
qps	事件发生时的每秒查询率
pps_in	事件发生时的入流量，单位：pps。
new_con	事件发生时的新连接
kbps_in	事件发生时的入流量，单位：bps。
instance_id	实例ID
time	日志时间，例如2020-07-17 10:00:30。
destination_ip	目的IP地址
port	目的端口
total_traffic_in_bps	总入流量，单位：bps。
total_traffic_drop_bps	总入流量的丢弃量，单位：bps。
total_traffic_in_pps	总入流量，单位：pps。
total_traffic_drop_pps	总入流量的丢弃量，单位：pps。
pps_types_in_tcp_pps	按协议统计的tcp类型入流量，单位：pps。
pps_types_in_udp_pps	按协议统计的udp类型入流量，单位：pps。
pps_types_in_icmp_pps	按协议统计的icmp类型入流量，单位：pps。
pps_types_in_syn_pps	按协议统计的syn类型入流量，单位：pps。
pps_types_in_ack_pps	按协议统计的ack类型入流量，单位：pps。
user_id	阿里云账号ID

1.7.11. 云安全中心

本文介绍云安全中心网络日志、安全日志和主机日志的字段详情。

网络日志

- DNS日志

日志字段	说明
__topic__	日志主题，固定为sas-log-dns。
owner_id	阿里云账号ID

日志字段	说明
additional	additional字段，各个值之间以竖线（|）分隔。
additional_num	additional字段数量
answer	DNS回答信息，各个值之间以竖线（|）分隔。
answer_num	DNS回答信息数量
authority	authority字段
authority_num	authority字段数量
client_subnet	客户端子网
dst_ip	目标IP地址
threat_dst_ip	目标IP地址的威胁情报。更多信息，请参见威胁情报字段。
dst_port	目标端口
in_out	数据的传输方向，包括： ○ in：入方式 ○ out：出方向
qid	查询ID
qname	查询域名
threat_qname	查询域名的威胁情报。更多信息，请参见威胁情报字段。
qtype	查询类型
query_datetime	查询时间戳，单位：毫秒。
rcode	返回代码
region	来源地域ID，包括： ○ 1：北京 ○ 2：青岛 ○ 3：杭州 ○ 4：上海 ○ 5：深圳 ○ 6：其它
response_datetime	返回时间
src_ip	源IP地址

日志字段	说明
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口

- 本地DNS日志

字段名	说明
__topic__	日志主题，固定为local-dns。
owner_id	阿里云账号ID
answer_rda	DNS回答信息，各个值之间以竖线（|）分隔。
answer_ttl	DNS回答的时间周期，各个值之间以竖线（|）分隔。
answer_type	DNS回答的类型，各个值之间以竖线（|）分隔。
anwser_name	DNS回答的名称，各个值之间以竖线（|）分隔。
dest_ip	目标IP地址
dest_port	目标端口
group_id	分组ID
hostname	主机名
id	主机IP地址
instance_id	实例ID
internet_ip	互联网IP地址
ip_ttl	IP地址的周期
query_name	查询域名
threat_query_name	查询域名的威胁情报。更多信息，请参见威胁情报字段。
query_type	查询类型
src_ip	源IP地址
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口
time	查询时间戳，单位：秒。
time_usecond	响应耗时，单位：微秒。

字段名	说明
tunnel_id	通道ID

- 网络会话日志

日志字段	说明
__topic__	日志主题，固定为sas-log-session。
owner_id	阿里云账号ID
asset_type	关联的资产类型，例如ECS、SLB、RDS等。
dst_ip	目标IP地址
threat_dst_ip	目标IP地址的威胁情报。更多信息，请参见威胁情报字段。
dst_port	目标端口
proto	协议类型，例如tcp、udp。
session_time	Session时间
src_ip	源IP地址
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口

- Web日志

日志字段	说明
__topic__	日志主题，固定为sas-log-http。
owner_id	阿里云账号ID
content_length	内容长度
dst_ip	目标IP地址
threat_dst_ip	目标IP地址的威胁情报。更多信息，请参见威胁情报字段。
dst_port	目标端口
host	访问主机名
jump_location	重定向地址
method	HTTP访问
referer	客户端向服务器发送请求时的HTTP referer

日志字段	说明
request_datetime	请求时间
ret_code	返回状态值
rqs_content_type	请求内容类型
rsp_content_type	响应内容类型
src_ip	源IP地址
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口
uri	请求URI
user_agent	向客户端发起的请求
x_forward_for	路由跳转信息

安全日志

- 漏洞日志

日志字段	说明
__topic__	日志主题，固定为sas-vul-log。
owner_id	阿里云账号ID
name	漏洞名称
alias_name	漏洞别名
op	操作信息，包括： ○ new：新增 ○ verify：验证 ○ fix：修复
status	状态。更多信息，请参见安全日志状态码。
tag	漏洞标签，例如oval、system、cms，主要用于区分EMG紧急漏洞。
type	漏洞类型，包括： ○ sys：windows漏洞 ○ cve：Linux漏洞 ○ cms：Web CMS漏洞 ○ EMG：紧急漏洞

日志字段	说明
uuid	客户端号

- 基线日志

日志字段	说明
__topic__	日志主题，固定为sas-hc-log。
owner_id	阿里云账号ID
level	风险级别
op	操作信息，包括： ○ new：新增 ○ verify：验证
risk_name	风险名称
status	状态。更多信息，请参见安全日志状态码。
sub_type_alias	子类型别名，中文。
sub_type_name	子类型名称
type_name	类型名称。更多信息，请参见基线type-sub-type列表。
type_alias	类型别名，中文。
uuid	客户端号
check_item	检查项名称
check_level	检查项级别
check_type	检查项类型

基线type-sub-type列表

type_name	sub_type_name
system	baseline
weak_password	postsql_weak_password
database	redis_check
account	system_account_security
account	system_account_security

type_name	sub_type_name
weak_password	mysq_weak_password
weak_password	ftp_anonymous
weak_password	rdp_weak_password
system	group_policy
system	register
account	system_account_security
weak_password	sqlserver_weak_password
system	register
weak_password	ssh_weak_password
weak_password	ftp_weak_password
cis	centos7
cis	tomcat7
cis	memcached-check
cis	mongodb-check
cis	ubuntu14
cis	win2008_r2
system	file_integrity_mon
cis	linux-httpd-2.2-cis
cis	linux-docker-1.6-cis
cis	SUSE11
cis	redhat6
cis	bind9.9
cis	centos6
cis	debain8
cis	redhat7
cis	SUSE12

type_name	sub_type_name
cis	ubuntu16

安全日志状态码

状态值	说明
1	未修复
2	修复失败
3	回滚失败
4	修复中
5	回滚中
6	验证中
7	修复成功
8	修复成功待重启
9	回滚成功
10	忽略
11	回滚成功待重启
12	已不存在
20	已失效

- 安全告警日志

日志字段	说明
__topic__	日志主题，固定为sas-security-log。
data_source	数据源。更多信息，请参见安全告警data_source列表。
level	告警级别
name	名称
op	操作信息，包括： ○ new：新增 ○ dealing：处理
status	状态。更多信息，请参见安全日志状态码。
uuid	客户端号

日志字段	说明
detail	告警详情
unique_info	告警的唯一标识

安全告警data_source列表

值	描述
aegis_suspicious_event	主机异常
aegis_suspicious_file_v2	Webshell
aegis_login_log	异常登录
security_event	安全中心异常事件

主机日志

- 进程启动日志

日志字段	说明
__topic__	日志主题，固定为aegis-log-process。
uuid	客户端号
ip	客户端主机的IP地址
cmdline	用户启动完整命令行
username	用户名
uid	用户ID
pid	进程ID
filename	进程文件名
filepath	进程文件完整路径
groupname	用户组
ppid	父进程ID
pfilename	父进程文件名
pfilepath	父进程文件完整路径
cmd_chain	进程链
containerhostname	容器主机名

日志字段	说明
containerpid	容器PID
containerimageid	镜像ID
containerimagename	镜像名称
containername	容器名称
containerid	容器ID
cwd	进程运行目录

- 进程快照日志

日志字段	说明
__topic__	日志主题，固定为aegis-snapshot-process。
owner_id	阿里云账号ID
uuid	客户端号
ip	客户端主机的IP地址
cmdline	用户启动完整命令行
pid	进程ID
name	进程文件名
path	进程文件完整路径
md5	进程文件进行MD5计算，超过1 MB的进程文件不进行计算。
pname	父进程文件名
start_time	进程启动时间，内置字段
user	用户名
uid	用户ID

- 登录日志

  1分钟内重复登录会被合并为1条日志。

日志字段	说明
__topic__	日志主题，固定为aegis-log-login。
owner_id	阿里云账号ID

日志字段	说明
uuid	客户端号
ip	客户端主机的IP地址
warn_ip	登录来源IP地址
threat_warn_ip	登录来源IP地址的威胁情报。更多信息，请参见威胁情报字段。
warn_port	登录端口
warn_type	登录类型，例如SSHLOGIN、RDPLOGIN、IPCLOGIN。
warn_user	登录用户名
warn_count	登录次数，例如3次表示这次登录前1分钟内还发送了2次。

- 暴力破解日志

字段名	说明
__topic__	日志主题，固定为aegis-log-crack。
owner_id	阿里云账号ID
uuid	客户端号
ip	客户端主机的IP地址
warn_ip	登录来源IP地址
threat_warn_ip	登录来源IP地址的威胁情报。更多信息，请参见威胁情报字段。
warn_port	登录端口
warn_type	登录类型，例如SSHLOGIN、RDPLOGIN、IPCLOGIN。
warn_user	登录用户名
warn_count	失败登录次数

- 主机网络连接日志

  主机上每隔10秒到1分钟会收集变化的网络连接。

日志字段	说明
__topic__	日志主题，固定为aegis-log-network。
owner_id	阿里云账号ID
uuid	客户端号

日志字段	说明
ip	客户端主机的IP地址
src_ip	源IP地址
threat_src_ip	源IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	源端口
dst_ip	目标IP地址
threat_dst_ip	目标IP地址的威胁情报。更多信息，请参见威胁情报字段。
dst_port	目标端口
proc_name	进程名
proc_path	进程路径
proto	连接协议
status	连接状态。更多信息，请参见网络连接状态描述列表。

网络连接状态描述列表

状态值	描述
1	closed
2	listen
3	syn send
4	syn recv
5	establisted
6	close wait
7	closing
8	fin_wait1
9	fin_wait2
10	time_wait
11	delete_tcb

- 端口监听快照

日志字段	说明
__topic__	日志主题，固定为aegis-snapshot-port。
owner_id	阿里云账号ID
uuid	客户端号
ip	客户端IP地址
proto	监听协议
src_ip	监听IP地址
threat_src_ip	监听IP地址的威胁情报。更多信息，请参见威胁情报字段。
src_port	监听端口
pid	进程ID
proc_name	进程名

- 账户快照

日志字段	说明
__topic__	日志主题，固定为aegis-snapshot-host。
owner_id	阿里云账号ID
name	漏洞名称
alias_name	漏洞别名
op	操作信息，包括： ○ new：新增 ○ verify：验证 ○ fix：修复
status	连接状态。更多信息，请参见网络连接状态描述列表。
tag	漏洞标签，例如oval、system、cms等，主要用于区分EMG紧急漏洞。
type	漏洞类型，包括： ○ sys：windows漏洞 ○ cve：Linux漏洞 ○ cms：Web CMS漏洞 ○ EMG：紧急漏洞
uuid	客户端号

1.7.12. API网关

本文介绍API网关访问日志的字段详情。

日志字段	说明
owner_id	API提供者的阿里云账号ID
apiGroupUid	API的分组ID
apiGroupName	API分组名称
apiUid	API ID
apiName	API名称
apiStageUid	API环境ID
apiStageName	API环境名称
httpMethod	HTTP请求方法
path	请求路径
domain	调用的域名
statusCode	HTTP状态码
errorMessage	错误信息
appId	调用者的应用ID
appName	调用者的应用名称
clientIp	调用者的客户端IP地址
threat_client_ip	调用者的客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
exception	返回的错误信息
region	地域
requestHandleTime	请求时间，格林威治时间
requestId	请求ID，全局唯一。
requestSize	请求大小，单位：字节。
responseSize	返回的数据大小，单位：字节。
serviceLatency	后端延迟，单位：毫秒。

1.7.13. 文件存储

本文介绍文件存储NAS访问日志的字段详情。

日志字段	说明
owner_id	阿里云账号ID
ArgIno	文件系统inode号
AuthRc	授权返回码
NFSProtocolRc	NFS协议返回码
OpList	NFSv4 Procedures编号
Proc	NFSv3 Procedures编号
RWSize	读写大小，单位：Byte。
RequestId	请求ID
ResIno	lookup的资源inode号
SourceIp	客户端IP地址
threat_source_ip	客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
Vers	NFS协议版本号
Vip	服务端IP地址
Volume	文件系统ID
microtime	请求发生时间，单位：微秒。

1.7.14. 移动推送

本文介绍移动推送的推送回执事件的字段详情。

日志字段	说明
__topic__	日志主题，固定为cps_callback_event。
owner_id	阿里云账号ID
app_key	AppKey
message_id	消息ID
event_time	回执事件时间

日志字段	说明
event_type	回执事件类型
device_id	设备ID
device_type	设备类型
last_active_time	设备最后活跃时间
app_version	应用版本号
client_ip	客户端IP地址
threat_source_ip	客户端IP地址的威胁情报。更多信息，请参见威胁情报字段。
brand	设备品牌
network_type	设备网络类型
os	设备操作系统
os_version	设备操作系统版本
isp	设备所属运营商
job_key	任务Key
event_channel	推送通道
vendor_message_id	厂商通道消息ID
reason	发送失败的原因

1.7.15. 应用集成

本文介绍应用集成操作日志的字段详情。

日志字段	说明
__topic__	日志主题，固定为appconnect_oplog。
uid	阿里云账号ID
execution_id	单次请求或者触发的唯一标识
status	本次集成流的执行状态，仅包含begin、done。
flow_name	集成流名称
step	集成流中步骤的名称，步骤的唯一标识。

日志字段	说明
id	步骤执行ID。集成流每次执行的唯一索引，可解码为stepTime时间戳字段。 在包含循环的业务场景中，同一步骤可执行多次，步骤名称相同，id不同。
type	步骤类型
duration	步骤执行持续时间，单位：纳秒。
message	步骤执行过程中，输出的信息，字符串文本格式。
step_time	集成流每次触发步骤开始执行的时间
container_ip	集成pod的IP地址
integration_name	集成pod的名称
failed	步骤运行是否成功

1.8. 查看全局数据

本文介绍如何在日志审计服务中查看从云产品接入的全局数据。

查看日志审计全局数据视图

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 单击**审计配置 > 云产品接入 > 全局数据**，查看日志审计全局数据视图。

查看云产品全局数据视图

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 单击**审计报表 > 中心化 > 云产品 > 云产品全局数据**，查看云产品全局数据视图。

> **说明** 目前仅支持查看RDS、SLB、OSS的全局数据视图。

报表详情

- 日志审计全局数据视图

仪表盘	描述	说明
活跃账户数	审计监控的账号总数	无
总日志量、小时日志量、天日志量	日志量统计	最多半小时延迟
日志审计全局数据视图、产品存量日志分布	所有采集云产品的日志全局数据汇总	最多半小时延迟

仪表盘	描述	说明
日志量整体趋势、产品日志量趋势	过去30天的日志量趋势	当天的统计有一小时延迟

- OSS全局数据

仪表盘	描述
总日志量、小时日志量、天日志量	OSS日志全局统计
访问日志总日志量、小时日志量、天日志量	访问日志统计
计量日志总志量、小时日志量、天日志量	计量日志统计
OSS全局信息	全局监控统计
日志量整体趋势、子类型日志量趋势	OSS过去30天的日志量趋势

- SLB全局数据

仪表盘	描述
总日志量、小时日志量、天日志量	SLB日志全局统计
经典网络日志总日志量、小时日志量、天日志量	经典网络日志统计
VPC网络日志总志量、小时日志量、天日志量	VPC网络日志统计
SLB全局信息	全局监控统计
日志量整体趋势、网路类型日志量趋势	SLB过去30天的日志量趋势

- RDS全局数据

仪表盘	描述
总日志量、小时日志量、天日志量	RDS日志全局统计
MySQL日志总日志量、小时日志量、天日志量	MySQL日志统计
PgSQL日志总志量、小时日志量、天日志量	PgSQL日志统计
MSSQL日志总志量、小时日志量、天日志量	MSSQL日志统计
RDS全局信息	全局监控统计
日志量整体趋势、子产品日志量趋势	RDS过去30天的日志量趋势

1.9. 使用Terraform配置日志审计

本文介绍如何使用Terraform调用接口配置日志审计服务。

前提条件

已安装和配置Terraform。具体操作，请参见在Cloud Shell中使用Terraform、在本地安装和配置Terraform。

背景信息

Terraform是一种开源工具，用于安全高效地预览、配置和管理云基础架构和资源。Terraform的命令行接口（CLI）提供了一种简单机制，用于将配置文件部署到阿里云或其他任意支持的云上，并对其进行版本控制。

阿里云是中国国内第一家与Terraform集成的云厂商。目前terraform-provider-alicloud已经提供了超过163个Resource和113个Data Source，覆盖计算、存储、网络、负载均衡、CDN、容器服务、中间件、访问控制和数据库等阿里云产品，满足大量大客户的自动化上云需求。

使用Terraform的优势

- 将基础结构部署到多个云

  Terraform适用于多云方案，将类似的基础结构部署到阿里云、其他云厂商或者本地数据中心。开发人员能够使用相同的工具和相似的配置文件同时管理不同云厂商的资源。

- 自动化管理基础结构

  您可以使用Terraform创建配置文件模板，用于重复、可预测的方式定义、预配和配置ECS资源，减少因人为因素导致的部署和管理错误。您可以多次部署同一模板，创建相同的开发、测试和生产环境。

- 基础架构即代码（Infrastructure as Code）

  Terraform支持通过代码来管理、维护资源，允许保存基础设施状态，从而使您能够跟踪对系统（基础设施即代码）中不同组件所做的更改，并与其他人共享这些配置 。

- 降低开发成本

  您通过按需创建开发和部署环境来降低成本。并且，您可以在系统更改之前进行评估。

步骤一：配置身份信息以及日志审计服务的中心化地域

在环境变量中配置用户身份信息以及日志审计服务的中心Project所在地域。

```
export ALICLOUD_ACCESS_KEY="LTAIUrZCw3****"
export ALICLOUD_SECRET_KEY="zfwwWAMWIAiooj14GQ2****"
export ALICLOUD_REGION="cn-huhehaote"
```

参数	说明
ALICLOUD_ACCESS_KEY	阿里云访问密钥AccessKey ID。更多信息，请参见访问密钥。
ALICLOUD_SECRET_KEY	阿里云访问密钥AccessKey Secret。更多信息，请参见访问密钥。
ALICLOUD_REGION	日志审计服务的中心Project所在地域。目前支持如下地域： • 中国：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） • 海外：新加坡、日本（东京）、德国（法兰克福）、印尼（雅加达）

步骤二：RAM授权

使用Terraform完成RAM授权。具体操作，请参见alicloud_ram_policy。在授权中所涉及的权限策略信息请参见自定义授权日志采集与同步。

步骤三：配置日志采集

1. 创建一个Terraform工作目录*sls*，并在该目录下创建一个名为*terraform.tf*的文件。
2. 在*terraform.tf*文件中，添加如下内容。

```
resource "alicloud_log_audit" "example" {
  display_name = "tf-audit-test"
  aliuid     = "12345678"
}
```

相关参数说明如下：

参数	说明
example	Resource名称。自定义配置。
display_name	采集配置名称。自定义配置。
aliuid	阿里云账号ID。

3. 在*sls*目录下，执行如下命令，初始化terraform工作目录。

```
terraform init
```

如果返回结果中提示 `Terraform has been successfully initialized!` ，表示初始化成功。

```
- Installed hashicorp/alicloud v1.125.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.


  Warning: Additional provider information from registry

  The remote registry returned warnings for registry.terraform.io/hashicorp/alicloud:
  - For users on Terraform 0.13 or greater, this provider has moved to aliyun/alicloud. Please up
  required_providers.


Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

4. 编辑*terraform.tf*文件，配置日志审计服务相关参数。

   配置示例如下。Terraform中日志审计采集配置的完整参数说明，请参见Terraform-Aliyun Log Audit。

   - 单账号采集

```
resource "alicloud_log_audit" "example" {
 display_name = "tf-audit-test"
 aliuid     = "12345678"
 variable_map = {
  "actiontrail_enabled" = "true",
  "actiontrail_ttl" = "180"
 }
}
```

- 多账号采集

```
resource "alicloud_log_audit" "example" {
 display_name = "tf-audit-test"
 aliuid     = "12345678"
 variable_map = {
  "actiontrail_enabled" = "true",
  "actiontrail_ttl" = "180"
 }
 multi_account = ["123456789123", "12345678912300123"]
}
```

参数	说明
actiontrail_enabled	是否开启操作审计（Actiontrail）日志的采集，取值： - *true*：开启。 - *false*：关闭。
actiontrail_ttl	设置操作审计日志的存储时间。
multi_account	多账号采集时，需配置多个阿里云账号ID。

5. 使*terraform.tf*文件中的采集配置生效。
   i. 执行如下命令。

   ```
   terraform apply
   ```

   ii. 输入*yes*。

   如果返回结果中提示 `Apply complete!` ，表示应用采集配置成功，日志审计服务将按照采集配置进行日志采集和存储。

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

alicloud_log_audit.example: Creating...
alicloud_log_audit.example: Creation complete after 3s [id=tf-audit-test]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

相关操作

您还可以通过Terraform完成如下相关操作。

- 导入已有审计配置。

  ```
  terraform import alicloud_log_audit.example tf-audit-test
  ```

  其中，*example*和*tf-audit-test*，请根据实际情况替换。

  ```
  lichang@B-QPJTMD6M-0104 terrafrom % terraform import alicloud_log_audit.example tf-audit-test
  alicloud_log_audit.example: Importing from ID "tf-audit-test"...
  alicloud_log_audit.example: Import prepared!
    Prepared alicloud_log_audit for import
  alicloud_log_audit.example: Refreshing state... [id=tf-audit-test]

  Import successful!

  The resources that were imported are shown above. These resources are now in
  your Terraform state and will henceforth be managed by Terraform.
  ```

  执行完毕后，您可以查看terraform工作目录下的*terraform.tfstate*文件内容。*terraform.tfstate*文件内容即为导入的采集配置。

  > **注意**
  >
  > - 如果您想要将导入的采集配置迁移到*terraform.tf*文件中，需要手动拷贝，并对格式做适当调整，满足*terraform.tf*文件的格式要求。
  > - 如果您已经在当前的terraform工作目录执行过terrraform apply或者terraform import命令，则此时再次执行terraform import命令会失败。您需要删除当前目录下的*terraform.tfstate*文件后再重新执行terraform import命令。

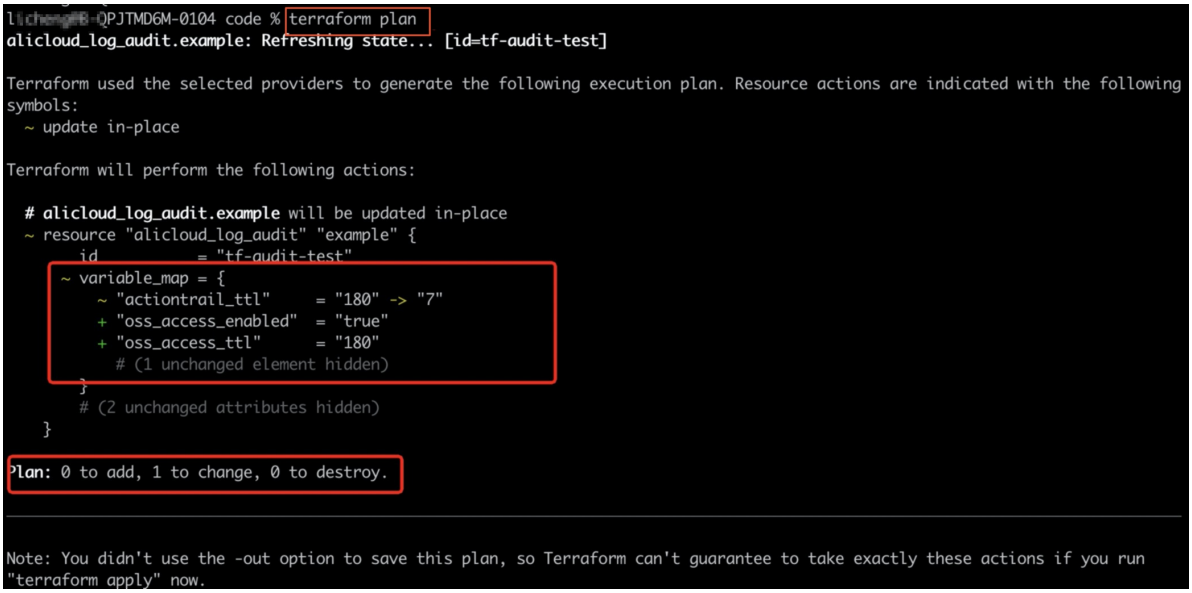
- 查看当前采集配置。

  ```
  terraform show
  ```

  ```
  li[redacted]0104 terrafrom % terraform show
  # alicloud_log_audit.example:
  resource "alicloud_log_audit" "example" {
      aliuid       = "1[redacted]0"
      display_name = "tf-audit-test"
      id           = "tf-audit-test"
      variable_map = {
          "actiontrail_enabled" = "true"
          "actiontrail_ttl"     = "180"
      }
  }
  ```

- 查看当前terraform工作目录下的*terraform.tf*文件与已生效的采集配置的差异。

  ```
  terraform plan
  ```



1.10. 采集策略

日志审计提供一键式跨账号采集云产品日志及中心化存储功能。对于已开通日志审计的阿里云产品，日志服务默认采集所有符合限定条件的云产品日志。而通过采集策略，可对账号、地域或实例等因素进行限制，实现精细化的日志采集目的。本文介绍如何配置采集策略。

产品支持

采集策略目前支持RDS、DRDS、PolarDB、SLB、Kubernetes容器，详细说明如下所示。

云产品	采集对象	属性	说明
RDS	RDS实例	账号：account.id	RDS实例所属的阿里云账号ID。
		地域：region	RDS实例所属的地域，例如：cn-shanghai。
		实例ID：instance.id	RDS实例ID。
		实例名：instance.name	RDS实例名。
		DB类型：instance.db_type	DB类型，可取值为mysql、pgsql、mssql。
		DB版本号：instance.db_version	DB版本号，例如：8.0。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。
		账号：account.id	PolarDB集群所属的阿里云账号ID。

云产品	采集对象	属性	说明
PolarDB	PolarDB集群	地域：region	PolarDB集群所属的地域，如cn-shanghai。
		集群ID：cluster.id	PolarDB集群ID。
		集群名：cluster.name	PolarDB集群名称。
		集群兼容的DB类型：cluster.db_type	PolarDB集群兼容的DB类型，目前只支持MySQL。
		集群兼容的DB版本：cluster.db_version	DB版本号，可选值为8.0、5.7和5.6。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。
DRDS	DRDS实例	账号：account.id	DRDS实例所属的阿里云账号ID。
		地域：region	DRDS实例所属的地域，例如：cn-shanghai。
		实例ID：instance.id	DRDS实例ID。
		实例名：instance.name	DRDS实例名。
SLB	SLB实例	账号：account.id	SLB实例所属的阿里云账号ID。
		地域：region	SLB实例所属的地域，例如：cn-shanghai。
		实例ID：instance.id	SLB实例ID。
		实例名：instance.name	SLB实例名。
		网络类型：instance.network_type	SLB网络类型，包括专有网络（VPC）和经典网络（Classic）。
		VPC ID：instance.vpc_id	SLB实例所属的专有网络VPC ID。
		地址类型：instance.address_type	SLB实例的地址类型，包括阿里云内网（intranet）和公网（internet）。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。

云产品	采集对象	属性	说明
Kubernetes容器（Kubernetes审计日志）	Kubernetes集群	地域：region	Kubernetes集群所属地域，例如：cn-shanghai。
		集群ID：cluster.id	Kubernetes集群ID。
		集群名：cluster.name	Kubernetes集群名称。
		集群类型：cluster.type	Kubernetes集群类型，包括专有版Kubernetes Kubernetes、托管版Kubernetes ManagedKubernetes、Serverless Kubernetes ASK。
		网络类型：cluster.network_mode	Kubernetes集群的网络类型，包括专有网络（VPC）和经典网络（Classic）。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。
Kubernetes容器（Kubernetes事件中心）	Kubernetes集群	地域：region	Kubernetes集群所属地域，例如：cn-shanghai。
		集群ID：cluster.id	Kubernetes集群ID。
		集群名：cluster.name	Kubernetes集群名称。
		集群类型：cluster.type	Kubernetes集群类型，包括专有版Kubernetes Kubernetes、托管版Kubernetes ManagedKubernetes、Serverless Kubernetes ASK。
		网络类型：cluster.network_mode	Kubernetes集群的网络类型，包括专有网络和经典网络。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。
		地域：region	Kubernetes集群所属地域，例如：cn-shanghai。
		集群ID：cluster.id	Kubernetes集群ID。
		集群名：cluster.name	Kubernetes集群名称。

云产品	采集对象	属性	说明
Kubernetes容器（Ingress访问日志）	Kubernetes集群	集群类型：cluster.type	Kubernetes集群类型，包括专有版Kubernetes Kubernetes、托管版Kubernetes ManagedKubernetes、Serverless Kubernetes ASK。
		网络类型：cluster.network_mode	Kubernetes集群的网络类型，包括专有网络（VPC）和经典网络（Classic）。
		标签：tag.*	用户自定义的标签名。 将tag.*中的星号（*）替换为您自定义的标签名。
		日志内容：log.*	日志内容。

配置采集策略

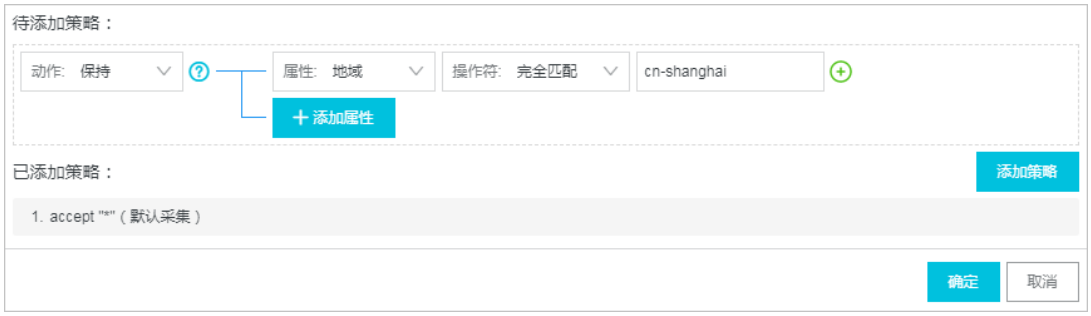
1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 选择**云产品接入 > 全局配置**，单击**修改**。
4. 单击目标云产品右侧的**采集策略**。
5. 配置采集策略。

   日志服务支持通过简易编辑模式或高级编辑模式配置采集策略。简易编辑模式配置简单，当简易编辑模式无法满足您的需求时，可开启高级编辑模式，灵活配置复杂的采集策略。

   > **说明**
   > - 您可以根据实际需求，配置多条采集策略。
   > - 在高级编辑模式下，您可以手动编辑策略语句，但在手动编辑策略语句后，无法返回到简易编辑模式。
   > - 在高级编辑模式下，清空策略语句并保存，再次打开可恢复到简易编辑模式。

   - 简易编辑模式

a. 在**待添加策略**区域，配置如下参数，并单击**添加策略**。



参数	说明
动作	通过您配置的采集策略，执行相应的动作。更多信息，请参见策略语法。
属性	选择采集对象的属性，不同采集对象对应的属性不同。更多信息，请参见产品支持。
操作符	选择操作符，例如选择**完全匹配**，则对应的操作符为==。更多信息，请参见策略语法。
属性取值	输入属性的值，支持配置多个值。

b. 在**已添加策略**区域，确认策略配置结果。

您也可以修改已添加的采集策略以及调整采集策略的顺序。

- 单击目标采集策略右侧的**编辑**，修改已添加的采集策略。
- 单击目标采集策略右侧的上下箭头，调整采集策略的顺序。

> **说明** 日志服务默认添加accept "*"策略，用于接受所有的采集项，不可编辑与删除。

c. 确认无误后，单击**确定**。

- 高级编辑模式

a. 开启**高级编辑模式**。

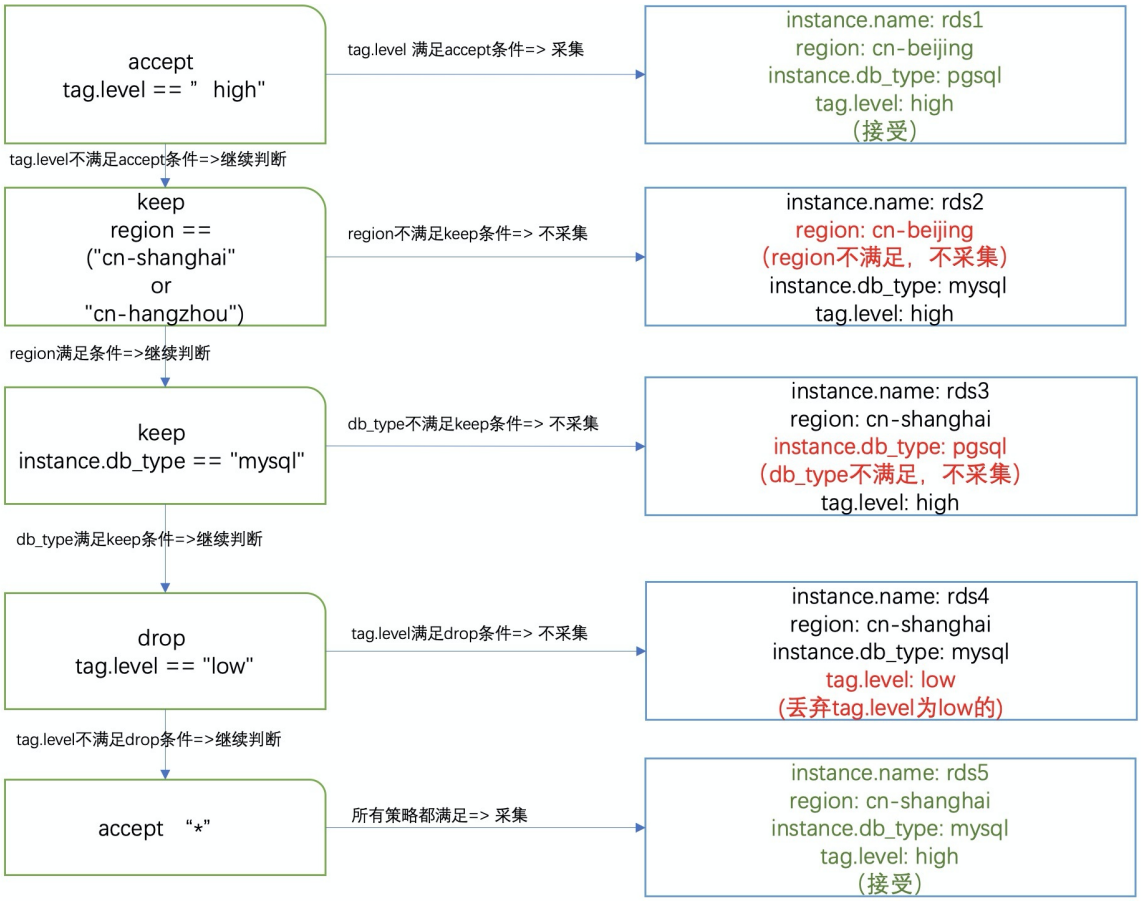
b. 在**规则**文本框中，配置采集策略，并单击**确定**。

详细的语法说明请参见策略语法。

6. 在**全局配置**页面，单击**保存**。

策略语法

- 动作
  - 保持（keep）：当采集对象满足采集策略时继续执行下一条策略，由后续策略判断是否采集日志。不满足则拒绝采集日志，不再做后续策略判断。
  - 拒绝（drop）：当采集对象满足采集策略时拒绝采集日志，不再执行下一条策略。不满足则继续执行下一条策略，由后续策略判断是否采集。
  - 接受（accept）：当采集对象满足采集策略时采集日志，不再执行下一条策略。不满足则继续执行下一条策略，由后续策略判断是否采集。



- 匹配模式

匹配模式	说明
完全匹配	通过字符串的完全匹配，进行采集策略的匹配。 ○ 操作符：== ○ 示例：**keep instance.db_type == "mysql"**表示MySQL类型的RDS实例通过当前判断。
通配符匹配	通过通配符星号（*）和问号（?）进行采集策略的匹配。星号（*） 表示0个或多个字符，半角问号（?）表示一个字符。 ○ 操作符：== ○ 示例： ■ **keep instance.name == "backend*"** 表示实例名以backend开头的实例，通过当前判断。 ■ **keep instance.name == "active?"**表示实例名以active开头且其后面还有一个任意字符的实例，通过当前判断。
正则表达式匹配	通过正则表达式进行采集策略的匹配。 ○ 操作符：~= ○ 示例：**keep instance.name ~= "^\d+$"**表示纯数字的实例名通过当前判断。 **说明** 默认为部分匹配，如果需要完全匹配，需要在开头和结尾加上^和$。
数值比较	对数值进行比较。 ○ 操作符： ■ 直接比较：>、>=、=、<=、< ■ 闭区间比较：：[*, 100]，支持用星号（*）表示无边界。 ○ 示例： ■ **keep tag.level >= 2**表示tag.level大于等于2的实例，通过当前判断。 ■ **keep tag.level：[*, 10]**表示tag.level小于等于10的实例，通过当前判断。 ■ **keep tag.level：[1, 10]**表示tag.level位于[1, 10]之间的实例，通过当前判断。

匹配模式	说明
逻辑关系	○ 关键字： ■ 且：使用and、AND、&&等关键词，不区分大小写。 ■ 或：使用or、OR等关键词，不区分大小写。 ■ 否：使用not，NOT，感叹号（!）等关键词，不区分大小写。 ○ 示例： ■ keep (tag.level > 10) and (region == "cn-shanghai")表示tag.level大于10且位于上海的实例，通过当前判断。 ■ keep (tag.level > 10) or (region == "cn-shanghai")表示tag.level大于10或位于上海的实例，通过当前判断。 ■ keep not region == "cn-shanghai"表示非上海的示例，通过当前判断。
全局匹配	如果策略中没有指定对象名，则表示全局匹配。例如： ○ keep "abc"表示含有abc字符的采集项都可以通过当前判断。 ○ accept "*"表示接受所有采集项。 说明 ○ 全局匹配，必须带双引号（" "）。 ○ 仅在高级编辑模式下，支持全局匹配。

- 字符转义

  采集策略中，需要对星号（*）、反斜线（\）等特殊字符进行转义，例如：keep instance.name == "abc*"表示实例名为abc*的实例通过当前判断。

常见案例

- 采集特定区域的实例日志

  例如：只采集中国区域的实例日志，采集策略如下所示。

```
# only scan cn region
keep region == "cn-*"
# accept by default
accept "*"
```

- 采集特定标签的实例日志

  例如：只采集所有标签打上type值是production（大小写不敏感）的实例日志，采集策略如下所示。

```
# only scan "production" instances
keep tag.type ~= "(?i)^production$"
# accept by default
accept "*"
```

- 复杂场景

例如：只采集RDS MySQL实例日志，但是如果标签打上level: high的实例，无论数据库类型是MySQL、SQL Server或PostgreSQL，都采集，采集策略如下所示。

```
# accept all high level instances
accept tag.level == "high"
# only scan mysql
keep instance.db_type == "mysql"
# accept by default
accept "*"
```

1.11. 告警

1.11.1. 设置告警

日志审计服务已内置告警规则，您开启对应的告警实例即可实时监控日志审计服务。本文介绍设置告警的相关操作。

前提条件

已在**全局配置**页面中开启目标云产品的审计功能。具体操作，请参见配置日志采集。

背景信息

日志审计服务中已内置告警规则、SLS审计内置告警策略、SLS审计内置行动策略、SLS审计内置用户组和SLS审计内置内容模板。它们之间的关联如下：

- 通过告警规则指定SLS审计内置告警策略。

  > **说明** 日志审计服务中的告警规则已绑定SLS审计内置告警策略，无法解绑和更换绑定。

- 通过SLS审计内置告警策略指定SLS审计内置行动策略。
- 通过SLS审计内置行动策略指定SLS审计内置用户组和SLS审计内置内容模板。

配置流程

您可以直接使用内置的告警资源，也可以自定义告警资源，具体设置告警的流程如下：

- 使用内置的告警资源

  如果您希望快速完成告警设置，通过语音、短信或邮件接收到告警通知，您可以根据如下流程完成设置。

  i. 创建用户
  ii. 将用户添加到SLS审计内置用户组
  iii. 开启告警实例

- 自定义告警资源

  如果您希望根据实际场景自定义告警资源，您可以根据如下流程完成设置。

  i. 创建用户和用户组
  ii. 创建内容模板
  iii. 创建行动策略

iv. 修改内置告警策略所绑定的行动策略

v. 设置白名单

vi. 开启告警实例

日志服务提供的内置资源可满足大部分告警场景，在实际场景中，你可以综合上述两种方式设置告警。本文以内置的告警资源为例。

步骤一：创建用户

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**日志审计服务**。
3. 在左侧导航栏中，选择**审计告警 > 用户管理 > 用户管理**。
4. 创建用户。

   具体操作，请参见创建用户。

步骤二：将用户添加到SLS审计内置用户组

1. 在左侧导航栏中，选择**审计告警 > 用户管理 > 用户组管理**。
2. 在用户组列表中，单击**SLS审计内置用户组**对应的**修改**。
3. 在**修改用户组**中，将已创建的用户从**待添加成员**区域添加到**已添加成员**区域，然后单击**确认**。

步骤三：开启告警实例

1. 在左侧导航栏中，选择**审计告警 > 规则配置 > 告警规则**。
2. 在告警规则列表中，找到目标告警规则，单击**开启**。

   开启告警实例后，日志服务开始实时监控日志审计服务。如果您需要开启多个告警实例，可单击**添加**。

   告警规则的参数说明请参见告警规则总览。

相关操作

操作	说明
设置白名单	针对特定告警规则，如果您希望某些用户（或者实例ID、IP地址）进行操作时不触发告警，可将其设置为白名单。 不同告警规则对应的白名单配置不同。更多信息，请参见告警规则总览。
关闭告警实例	关闭告警实例后，告警实例不会再触发告警，**状态**变更为**未开启**。 该操作不会删除实例参数中已设置的信息。需要再次监控时，无需重新设置实例参数。
临时关闭告警实例	临时关闭告警实例后，在指定时间内不再触发告警。
恢复告警实例	处于临时关闭状态的告警实例，可随时恢复告警。
删除告警实例	删除告警实例，**状态**变更为**未创建**。 该操作会删除实例参数中已设置的信息（例如阿里云账号）。需要再次监控时，需要重新设置实例参数。

操作	说明
升级告警实例	当日志服务对告警规则进行较大的功能升级或升级后需要您额外配置时，系统会提示您升级告警规则。一般情况下，系统会自动完成升级。
手动初始化告警	如果误删除告警初始化产生的资产或者发生首次初始化告警资产失败的情况，可通过此操作强制重新初始化告警相关内容。
修改内置告警策略所绑定的行动策略	如果您要使用自定义的行动策略，则在创建行动策略后，需在**告警策略**页面，修改SLS审计内置告警策略的所绑定的行动策略。

1.11.2. 告警规则

1.11.2.1. 告警规则总览

本文介绍日志审计服务的内置告警规则，包括日志审计合规、账号安全、权限控制和流量安全等。了解告警规则，有助于您快速发现审计相关问题。

告警规则列表

支持的告警规则类型如下表所示。设置告警参数、设置白名单相关操作，请参见设置告警。

类型	告警规则
日志审计合规	云安全中心日志审计配置检测
	RDS日志审计配置检测
	PolarDB（DRDS）日志审计配置检测
	K8s日志审计配置检测
	应用防火墙（WAF）日志审计配置检测
	堡垒机日志审计配置检测
	API网关日志审计配置检测
	云防火墙日志审计配置检测
	日志审计状态检测
	ActionTrail日志审计配置检测
	RAM子账号无MFA登录告警
	RAM密码过期策略异常设置告警
	Root账号无MFA登录告警
	RAM密码登录重试策略异常设置告警
	Root账户连续登录告警

类型	告警规则
安全	RAM历史密码检查策略异常设置告警
	密钥配置变更告警
	账号连续登录失败告警
	Root账号AK使用检测
	RAM密码长度策略异常设置告警
权限控制	OSS Bucket权限变更告警
	RAM策略变更告警
	RAM策略异常添加告警
OSS操作合规	OSS Bucket加密关闭告警
	OSS新创建的Bucket加密未开启告警
	OSS Bucket访问日志记录关闭告警
	OSS新创建的Bucket访问日志记录未开启告警
RDS操作合规	RDS实例SQL洞察关闭告警
	RDS实例访问白名单异常设置告警
	新创建的RDS实例的SSL未开启告警
	新创建的RDS实例的TDE未开启告警
	RDS实例SSL关闭告警
	RDS实例配置变更告警
SLB操作合规	负载均衡修改保护关闭告警
	负载均衡健康检查关闭告警
ECS操作合规	ECS磁盘加密关闭告警
	ECS自动快照策略关闭告警
	安全组配置变更告警
	ECS网络类型检测
VPC操作合规	VPC网络路由变更告警
	VPC流日志配置异常变更告警

类型	告警规则
	VPC通用配置变更告警
云防火墙操作合规	云防火墙控制策略变更告警
API调用	未授权的API调用告警
TDI操作合规	云安全中心网页防篡改功能关闭告警
K8s安全	K8s Warning事件数过多告警
	K8s频繁删除事件告警
	K8s错误事件数过多告警
RDS安全	RDS慢SQL检测
	RDS大批量数据删除告警
	RDS外网访问检测
	RDS查询SQL平均执行时间监控告警
	RDS数据库更新峰值监控告警
	RDS数据库查询峰值监控告警
	RDS实例释放告警
	RDS高频访问IP检测
	RDS更新SQL平均执行时间监控告警
	RDS登录失败次数过多告警
	RDS大批量数据修改事件告警
	RDS危险的SQL执行告警
	RDS SQL执行错误数过多告警
SLB流量安全	负载均衡响应报文长度异常检测
	负载均衡请求报文长度异常检测
	负载均衡平均响应延迟过高告警
	负载均衡HTTP访问协议开启告警
	负载均衡访问UV异常检测
	负载均衡访问PV异常检测

类型	告警规则
API网关流量安全	API网关服务端平均延时过高告警
	API网关后端服务器错误率过高告警
	API网关请求成功率过低告警
OSS流量安全	OSS流入流量异常检测
	OSS Bucket有效请求率过低告警
	OSS外网访问检测
	OSS访问PV异常检测
	OSS流量异常检测
	OSS流出流量异常检测
	OSS访问UV异常检测
K8s流量安全	K8s非法访问次数过多告警
	K8s Ingress平均请求延迟过高告警
	K8s Ingress后端平均响应延迟过高告警
	K8s Ingress请求成功率过低告警
OSS数据安全	OSS Bucket账号访问控制
	OSS频繁删除对象告警
NAS数据安全	文件存储操作错误检测
	文件存储大批量删除文件告警
WAF安全事件	应用防火墙有效请求率过低告警
	应用防火墙防护网站被攻击次数过多告警
TDI安全事件	云安全中心高优先级告警数过多
	云安全中心新增漏洞数过多
	云安全中心有效请求率过低告警
	云安全中心新增告警数过多
	云安全中心外网DNS请求成功率过低告警
	云防火墙流出流量拦截告警

云防火墙安全事件类型	告警规则
	云防火墙流入流量拦截告警

1.11.2.2. 日志审计合规

本文介绍日志审计合规的告警规则，包括OSS、RDS、PolarDB、SLB、NAS、K8s等云产品的日志审计合规规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现日志审计合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 云安全中心日志审计配置检测
- RDS日志审计配置检测
- 日志审计状态检测
- PolarDB（DRDS）日志审计配置检测
- K8s日志审计配置检测
- ActionTrail日志审计配置检测
- OSS（对象存储）日志审计配置检测
- 应用防火墙（WAF）日志审计配置检测
- 堡垒机日志审计配置检测
- NAS（文件存储）日志审计配置检测
- API网关日志审计配置检测
- SLB日志审计配置检测
- 云防火墙日志审计配置检测

云安全中心日志审计配置检测

告警ID	sls_app_audit_cis_at_sas_audit_check
告警名称	云安全中心日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测云安全中心日志在日志审计服务中的配置是否正常。确保云安全中心日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无

消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启云安全中心日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

RDS日志审计配置检测

告警ID	sls_app_audit_cis_at_rds_audit_check
告警名称	RDS日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测RDS日志在日志审计服务中的配置是否正常。确保RDS日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启RDS日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

日志审计状态检测

告警ID	sls_app_audit_cis_at_audit_status_check
告警名称	日志审计状态检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	日志审计服务总体状态检测，总体状态异常时会触发告警。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	无
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 接入状态**中查看日志审计服务的状态，定位状态异常的原因。

前提条件	无

PolarDB（DRDS）日志审计配置检测

告警ID	sls_app_audit_cis_at_drds_audit_check
告警名称	PolarDB（DRDS）日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测PolarDB日志在日志审计服务中的配置是否正常。确保PolarDB（DRDS）日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启Polar（DRDS）日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

K8s日志审计配置检测

告警ID	sls_app_audit_cis_at_k8s_audit_check
告警名称	K8s日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测K8s相关日志（K8s审计日志、K8s事件中心和Ingress访问日志）在日志审计服务中的配置是否正常。确保K8s日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启K8s相关日志（K8s审计日志、K8s事件中心和Ingress访问日志）的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。

前提条件	无

ActionTrail日志审计配置检测

告警ID	sls_app_audit_cis_at_actiontrail_audit_check
告警名称	ActionTrail日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测操作审计（ActionTrail）日志在日志审计服务中的配置是否正常。确保ActionTrail日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启操作审计（ActionTrail）日志开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

OSS（对象存储）日志审计配置检测

告警ID	sls_app_audit_cis_at_oss_audit_check
告警名称	OSS（对象存储）日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测OSS相关日志（访问日志和计量日志）在日志审计服务中的配置是否正常。确保OSS（对象存储）日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无

消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启OSS相关日志（访问日志和计量日志）的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

应用防火墙（WAF）日志审计配置检测

告警ID	sls_app_audit_cis_at_waf_audit_check
告警名称	应用防火墙（WAF）日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测应用防火墙（WAF）日志在日志审计服务中的配置是否正常。确保应用防火墙（WAF）日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启应用防火墙（WAF）日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

堡垒机日志审计配置检测

告警ID	sls_app_audit_cis_at_bastion_audit_check
告警名称	堡垒机日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测堡垒机日志在日志审计服务中的配置是否正常。确保堡垒机日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无

消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启堡垒机日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

NAS（文件存储）日志审计配置检测

告警ID	sls_app_audit_cis_at_nas_audit_check
告警名称	NAS（文件存储）日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测NAS（文件存储）日志在日志审计服务中的配置是否正常。确保NAS（文件存储）日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启NAS（文件存储）日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

API网关日志审计配置检测

告警ID	sls_app_audit_cis_at_apigateway_audit_check
告警名称	API网关日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测API网关日志在日志审计服务中的配置是否正常。确保API网关日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无

消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启API网关日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

SLB日志审计配置检测

告警ID	sls_app_audit_cis_at_slb_audit_check
告警名称	SLB日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测SLB日志在日志审计服务中的配置是否正常。确保SLB日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无
消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启SLB日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

云防火墙日志审计配置检测

告警ID	sls_app_audit_cis_at_cloudfirewall_audit_check
告警名称	云防火墙日志审计配置检测
版本号	1
类别	云平台、阿里云、CIS、日志审计合规
作用	检测云防火墙日志在日志审计服务中的配置是否正常。确保云防火墙日志的审计开关已开启，且其存储时长大于等于规则参数中存储时长（ttl）最小值。
执行频率	固定时间间隔1分钟
查询范围	过去2分钟
参数配置	存储时长（ttl）最小值：存储时长最小值，默认为180天。
外部配置	无

消除方法	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中开启云防火墙日志的审计开关，并确保存储时长大于规则参数配置中设定的存储时长（ttl）最小值。
前提条件	无

1.11.2.3. 账号安全

本文介绍账号安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现账号安全相关问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- RAM子账号无MFA登录告警
- RAM密码过期策略异常设置告警
- Root账号无MFA登录告警
- RAM密码登录重试策略异常设置告警
- Root账户连续登录告警
- RAM历史密码检查策略异常设置告警
- 密钥配置变更告警
- 账号连续登录失败告警
- Root账号AK使用检测
- RAM密码长度策略异常设置告警

RAM子账号无MFA登录告警

告警ID	sls_app_audit_cis_at_ram_mfa
告警名称	RAM子账号无MFA登录告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控RAM用户无MFA（多登录因素验证）登录的行为。RAM用户登录控制台时需要开启MFA，且其登录次数小于等于规则参数配置中设定最大登录次数，否则会触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 最大登录次数：每5分钟内，允许未开启MFA的RAM用户登录的最大次数。默认值为0。
外部配置	无MFA登录的RAM用户白名单。白名单中RAM用户无MFA登录行为不会触发该告警。

消除方法	确保RAM用户5分钟内无MFA登录次数小于等于规则参数配置中设定的最大登录次数。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM密码过期策略异常设置告警

告警ID	sls_app_audit_cis_at_pwd_expire_policy
告警名称	RAM密码过期策略异常设置告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控RAM密码策略中的密码过期策略的设置是否正常。RAM密码策略中，RAM密码的有效期应该小于等于规则参数中设定的密码有效期最大值，否则会触发告警。
执行频率	固定时间间隔：5分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 密码有效期最大值：默认值为90天。根据阿里云CIS规则，该值建议设置为小于等于90。
外部配置	无
消除方法	确保RAM密码策略中密码有效期的值小于等于规则参数配置中设定的密码有效期最大值。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

Root账号无MFA登录告警

告警ID	sls_app_audit_cis_at_root_mfa
告警名称	Root账号无MFA登录告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控Root账号无MFA（多登录因素验证）登录的行为。Root账号登录控制台时需要开启MFA，且其登录次数小于等于规则参数配置中设定的最大登录次数，否则会触发告警。
执行频率	固定时间间隔：4分钟

查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 最大登录次数：Root账号每天未开启MFA登录的最大次数，默认值0。
外部配置	无MFA登录的Root账号白名单。白名单中的账号无MFA登录行为不会触发该告警。
消除方法	确保Root账号5分钟内无MFA登录次数小于等于规则参数配置中设定的最大登录次数。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM密码登录重试策略异常设置告警

告警ID	sls_app_audit_cis_at_pwd_login_attemp_policy
告警名称	RAM密码登录重试策略异常设置告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控RAM密码策略中的登录重试策略的设置是否正常。RAM密码登录重试策略中，允许一小时内使用错误密码尝试登录次数不能大于规则参数中设定的最大登录失败次数/h，否则会触发告警。
执行频率	固定时间间隔：5分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 最大登录失败次数/h：密码登录重试策略中，允许一小时内使用错误密码尝试登录次数的最大值。默认值为5。根据阿里云CIS规则，该值建议设置为5。
外部配置	无
消除方法	确保RAM密码登录重试策略中，允许一小时内最大连续失败登录次数的值小于等于规则参数配置中设定的最大登录失败次数/h。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

Root账户连续登录告警

告警ID	sls_app_audit_cis_at_root_login
告警名称	Root账户连续登录告警

版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控Root账号的连续登录行为。Root用户登录不能过于频繁，5分钟内登录次数超过规则参数中设定的最大登录次数会触发告警。
执行频率	固定时间间隔：5分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 最大登录次数：Root账号5分钟内的最大登录次数，默认值为2。
外部配置	Root账号登录白名单。白名单中账号的登录行为不会触发告警。
消除方法	确保Root账号每天登录次数小于等于规则参数配置中设定的最大登录次数。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM历史密码检查策略异常设置告警

告警ID	sls_app_audit_cis_at_pwd_reuse_policy
告警名称	RAM历史密码检查策略异常设置告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控RAM密码策略中的历史密码检查策略的设置是否正常。RAM历史密码检查策略中，禁止使用前N次密码。可在告警规则参数中配置N的最小值，小于该值会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为高。 • 密码重用最小值：历史密码检查策略中，**禁止使用前N次密码**中N的最小值。默认值为4。根据阿里云CIS规则，该值建议设为4。
外部配置	无
消除方法	确保RAM历史密码检查策略**禁止使用前N次密码**中N的值大于等于规则参数配置中设定密码重用最小值。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

密钥配置变更告警

告警ID	sls_app_audit_cis_at_ak_conf_change
告警名称	密钥配置变更告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控账号密钥配置变更事件。账号密钥的配置发生变更后（如删除或禁用等）会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许进行密钥配置变更的RAM用户白名单。使用白名单中的RAM用户进行密钥配置变更不会触发告警。
消除方法	禁止使用白名单以外的账号进行账号密钥配置变更。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

账号连续登录失败告警

告警ID	sls_app_audit_cis_at_abnormal_login_count
告警名称	账号连续登录失败告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控连续登录失败行为。5分钟内连续失败登录次数大于规则参数中设定的最大失败登录次数后触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为高。 • 最大失败登录次数：一个账号5分钟内的失败登录最大次数，默认值为5。
外部配置	无
消除方法	确保账号5分钟内的失败登录次数小于等于规则参数配置中设定的最大失败登录次数。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

Root账号AK使用检测

告警ID	sls_app_audit_cis_at_root_ak_usage
告警名称	Root账号AK使用检测
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控Root账号的密钥（AccessKey）使用行为。Root账号不应该创建和使用AccessKey密钥，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	Root账号密钥使用白名单。使用白名单中的Root账号密钥不会触发告警。
消除方法	确保Root账号密钥不被使用。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM密码长度策略异常设置告警

告警ID	sls_app_audit_cis_at_pwd_length_policy
告警名称	RAM密码长度策略异常设置告警
版本号	1
类别	云平台、阿里云、CIS、账号安全
作用	监控RAM密码策略中的密码长度策略的设置是否正常。RAM密码策略中，RAM密码的最小长度不能小于规则参数中设定的密码最小长度，否则会触发告警。
执行频率	固定时间间隔：5分钟
查询范围	过去5分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为高。 • 密码最小长度：密码策略中的密码最小长度的最小值。默认值为14。根据阿里云CIS规则，该值建议设置为14。
外部配置	无
消除方法	确保RAM密码策略中设置的密码最小长度大于等于规则参数配置中设定密码最小长度。

前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.4. 权限控制

本文介绍权限控制的告警规则，包括RAM用户策略变更、异常和OSS Bucket权限变更的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现权限控制相关问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- OSS Bucket权限变更告警
- RAM策略变更告警
- RAM策略异常添加告警

OSS Bucket权限变更告警

告警ID	sls_app_audit_cis_at_oss_policy_change
告警名称	OSS Bucket权限变更告警
版本号	1
类别	云平台、阿里云、CIS、权限控制
作用	监控OSS Bucket权限变更行为。OSS Bucket的权限发生变更后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许进行OSS Bucket权限变更的RAM用户白名单。使用白名单中的RAM用户进行Bucket权限变更不会触发告警。
消除方法	禁止使用白名单以外的账号进行OSS Bucket权限变更。
前提条件	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM策略变更告警

告警ID	sls_app_audit_cis_at_ram_policy_change
告警名称	RAM策略变更告警
版本号	1
类别	云平台、阿里云、CIS、权限控制

作用	监控RAM策略变更行为。RAM策略发生变更后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许进行RAM策略变更的RAM用户白名单。使用白名单中的RAM用户进行RAM策略变更不会触发告警。
消除方法	禁止使用白名单以外的账号进行RAM策略变更。
前提条件	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RAM策略异常添加告警

告警ID	sls_app_audit_cis_at_ram_policy_attach
告警名称	RAM策略异常添加告警
版本号	1
类别	云平台、阿里云、CIS、权限控制
作用	监控RAM策略是否存在异常添加行为。禁止将RAM策略添加到用户，只能添加到用户组或角色，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许添加RAM策略的RAM用户白名单。使用白名单中的RAM用户添加RAM策略不会触发告警。
消除方法	禁止将RAM策略添加到用户，只能添加到用户组或角色。
前提条件	在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.5. OSS操作合规

本文介绍OSS操作合规的告警规则，包括OSS Bucket加密关闭和新创建加密未开启等告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现OSS操作合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- OSS Bucket加密关闭告警

- OSS新创建的Bucket加密未开启告警
- OSS Bucket访问日志记录关闭告警
- OSS新创建的Bucket访问日志记录未开启告警

OSS Bucket加密关闭告警

告警ID	sls_app_audit_cis_at_oss_encry_config
告警名称	OSS Bucket加密关闭告警
版本号	1
类别	云平台、阿里云、CIS、OSS操作合规
作用	监控OSS Bucket加密关闭行为。所有OSS Bucket都应该在服务端开启加密，不建议关闭。关闭加密会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许OSS Bucket不开启加密的账号白名单。白名单账号下的OSS Bucket加密被关闭后，不会触发告警。
消除方法	禁止白名单以外的账号下的OSS Bucket关闭加密功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**的开关。

OSS新创建的Bucket加密未开启告警

告警ID	sls_app_audit_cis_at_oss_bucket_encry_off
告警名称	OSS新创建的Bucket加密未开启告警
版本号	1
类别	云平台、阿里云、CIS、OSS操作合规
作用	监控新创建的OSS Bucket加密未开启行为。OSS Bucket在创建时应该打开加密开关，或者在创建后（1小时内）尽快打开加密开关，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去1小时
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许OSS Bucket不开启加密的账号白名单。使用白名单中的账号下的OSS Bucket在创建后可以不开启加密。

消除方法	OSS Bucket在创建时打开加密开关，或者创建后尽快（1小时内）打开加密开关。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**的开关。

OSS Bucket访问日志记录关闭告警

告警ID	sls_app_audit_cis_at_oss_log_config
告警名称	OSS Bucket访问日志记录关闭告警
版本号	1
类别	云平台、阿里云、CIS、OSS操作合规
作用	监控OSS Bucket访问日志记录关闭行为。所有OSS Bucket都应该开启访问日志记录功能，不建议关闭。关闭日志记录后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许不开启OSS访问日志的账号白名单。使用白名单中账号下的OSS Bucket访问日志记录被关闭后，不会触发告警。
消除方法	禁止白名单以外的账号下的OSS Bucket关闭日志记录功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**的开关。

OSS新创建的Bucket访问日志记录未开启告警

告警ID	sls_app_audit_cis_at_oss_log_off
告警名称	OSS新创建的Bucket访问日志记录未开启告警
版本号	1
类别	云平台、阿里云、CIS、OSS操作合规
作用	监控新创建的OSS Bucket的访问日志记录未开启行为。OSS Bucket在创建后应该尽快开启访问日志记录功能。在Bucket创建1小时后还未打开访问日志记录会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去1小时
参数配置	严重度：严重、高、中、低、报告。默认值为中。

外部配置	允许不开启OSS访问日志的账号白名单。使用白名单中的账号下的OSS Bucket在创建后可以不开启访问日志记录功能。
消除方法	白名单以外的账号下的OSS Bucket在创建后尽快（1小时内）打开访问日志记录功能。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开OSS**访问日志**的开关。

1.11.2.6. RDS操作合规

本文介绍RDS操作合规的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现RDS操作合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- RDS实例SQL洞察关闭告警
- RDS实例访问白名单异常设置告警
- 新创建的RDS实例的SSL未开启告警
- 新创建的RDS实例的TDE未开启告警
- RDS实例SSL关闭告警
- RDS实例配置变更告警

RDS实例SQL洞察关闭告警

告警ID	sls_app_audit_cis_at_rds_sql_audit
告警名称	RDS实例SQL洞察关闭告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控RDS实例的SQL洞察关闭行为。RDS实例的SQL洞察功能应该保持开启，关闭后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许RDS SQL洞察功能关闭的账号白名单。白名单账号下RDS实例的SQL洞察功能关闭后，不会触发告警。
消除方法	禁止白名单以外的账号下的RDS实例关闭SQL洞察功能。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RDS实例访问白名单异常设置告警

告警ID	sls_app_audit_cis_at_rds_access_whitelist
告警名称	RDS实例访问白名单异常设置告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控RDS实例的访问白名单的异常设置行为。RDS实例的访问白名单不应该设置为0.0.0.0，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许RDS访问白名单设置为0.0.0.0的账号白名单。白名单账号下RDS实例的访问白名单设置为0.0.0.0后，不会触发告警。
消除方法	禁止白名单以外的账号下的RDS实例将访问白名单设置为0.0.0.0。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

新创建的RDS实例的SSL未开启告警

告警ID	sls_app_audit_cis_at_rds_ssl_off
告警名称	新创建的RDS实例的SSL未开启告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控新创建的RDS实例SSL未开启行为。RDS实例创建后应该尽快（1小时内）开启SSL，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去1小时
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许RDS不开启SSL的账号白名单。白名单账号下RDS实例在创建后可以不开启SSL。
消除方法	白名单以外账号下的RDS实例在创建后尽快（1小时内）打开SSL。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

新创建的RDS实例的TDE未开启告警

告警ID	sls_app_audit_cis_at_rds_tde_off
告警名称	新创建的RDS实例的TDE未开启告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控新创建的RDS实例TDE未开启行为。RDS实例在创建后应该尽快（1小时）打开TDE功能，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去1小时
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许RDS不开启TDE的账号白名单。白名单账号下的RDS实例在创建后可以不开启TDE。
消除方法	白名单以外账号下的RDS实例在创建后尽快（1小时内）打开TDE。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RDS实例SSL关闭告警

告警ID	sls_app_audit_cis_at_rds_ssl_config
告警名称	RDS实例SSL关闭告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控RDS实例的SSL关闭行为。RDS实例的SSL应该保持开启，关闭后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允允许RDS不开启SSL的账号白名单。白名单账号下RDS实例的SSL功能关闭后，不会触发告警。
消除方法	禁止白名单以外的账号下的RDS实例关闭SSL。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

RDS实例配置变更告警

告警ID	sls_app_audit_cis_at_rds_conf_change
告警名称	RDS实例配置变更告警
版本号	1
类别	云平台、阿里云、CIS、RDS操作合规
作用	监控RDS实例的配置变更行为。RDS实例的配置发生变更后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为低。
外部配置	RDS配置变更不会触发告警的账号白名单。白名单账号下RDS实例的配置发生变更后，不会触发告警。
消除方法	检查发生配置变更的RDS实例及其配置变更项是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.7. SLB操作合规

本文介绍负载均衡（SLB）操作合规的告警规则，包括SLB健康检测关闭和关闭修改保护告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现SLB操作合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 负载均衡修改保护关闭告警
- 负载均衡健康检查关闭告警

负载均衡修改保护关闭告警

告警ID	sls_app_audit_cis_at_slb_mod_protec
告警名称	负载均衡修改保护关闭告警
版本号	1
类别	云平台、阿里云、CIS、SLB操作合规
作用	监控负载均衡（SLB）实例的修改保护关闭行为。负载均衡（SLB）实例的修改保护功能应该保持开启，关闭后会触发告警。
执行频率	固定时间间隔：1分钟

查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许关闭修改保护的SLB实例白名单。白名单中SLB实例的修改保护功能关闭后，不会触发告警。
消除方法	禁止白名单以外的SLB实例关闭修改保护功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

负载均衡健康检查关闭告警

告警ID	sls_app_audit_cis_at_slb_health_check
告警名称	负载均衡健康检查关闭告警
版本号	1
类别	云平台、阿里云、CIS、SLB操作合规
作用	监控负载均衡（SLB）实例的健康检查关闭行为。负载均衡（SLB）实例的健康检查功能应该保持开启，关闭后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许关闭健康检查的SLB实例白名单。关闭白名单中SLB实例的健康检查功能后，不会触发告警。
消除方法	禁止白名单以外的SLB实例关闭健康检查功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.8. ECS操作合规

本文介绍ECS操作合规的告警规则，包括ECS磁盘加密、自动快照策略、安全组变更等告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现ECS操作合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- ECS磁盘加密关闭告警
- ECS自动快照策略关闭告警
- 安全组配置变更告警
- ECS网络类型检测

ECS磁盘加密关闭告警

告警ID	sls_app_audit_cis_at_ecs_disk_encry_detection
告警名称	ECS磁盘加密关闭告警
版本号	1
类别	云平台、阿里云、CIS、ECS操作合规
作用	监控ECS磁盘加密关闭行为。ECS磁盘应该在服务端开启加密，关闭加密会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许磁盘不加密的账号白名单。关闭白名单账号下磁盘的加密功能后，不会触发告警。
消除方法	禁止白名单以外账号下的磁盘关闭加密功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

ECS自动快照策略关闭告警

告警ID	sls_app_audit_cis_at_ecs_auto_snapshot_policy
告警名称	ECS自动快照策略关闭告警
版本号	1
类别	云平台、阿里云、CIS、ECS操作合规
作用	监控ECS自动快照策略的关闭行为。ECS磁盘建议使用自动快照策略进行自动备份，关闭自动快照策略会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许取消磁盘自动快照策略的账号白名单。白名单账号下磁盘的自动快照策略被关闭后，不会触发告警。
消除方法	禁止白名单以外账号下的磁盘关闭自动快照策略。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

安全组配置变更告警

告警ID	sls_app_audit_cis_at_securitygroup_change
告警名称	安全组配置变更告警
版本号	1
类别	云平台、阿里云、CIS、ECS操作合规
作用	监控安全组配置变更行为。ECS安全组的配置发生变更时会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许进行安全组配置变更的子账号白名单。白名单中的账号进行安全组配置变更时，不会触发告警。
消除方法	禁止白名单以外的账号进行安全组配置变更。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

ECS网络类型检测

告警ID	sls_app_audit_cis_at_ecs_network_type
告警名称	ECS网络类型检测
版本号	1
类别	云平台、阿里云、CIS、ECS操作合规
作用	监控ECS网络类型是否存在异常。ECS建议使用专有网络VPC，创建经典网络的ECS会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许ECS使用经典网络的账号白名单。白名单账号下创建使用经典网络的ECS，不会触发告警。
消除方法	禁止白名单以外的账号创建经典网络的ECS。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.9. VPC操作合规

本文介绍VPC操作合规的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现VPC操作合规问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- VPC网络路由变更告警
- VPC流日志配置异常变更告警
- VPC通用配置变更告警

VPC网络路由变更告警

告警ID	sls_app_audit_cis_at_vpc_route_change
告警名称	VPC网络路由变更告警
版本号	1
类别	云平台、阿里云、CIS、VPC操作合规
作用	监控VPC网络路由的变更行为。VPC网络路由的配置发生变更后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为低。
外部配置	允许VPC网络路由配置变更的账号白名单。白名单中的账号进行VPC网路路由配置变更时，不会触发告警。
消除方法	禁止白名单以外的账号进行VPC网络路由配置变更。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

VPC流日志配置异常变更告警

告警ID	sls_app_audit_cis_at_vpc_flowlog_detection
告警名称	VPC流日志配置异常变更告警
版本号	1
类别	云平台、阿里云、CIS、VPC操作合规
作用	监控VPC流日志的异常变更行为。所有VPC都应该开启流日志，关闭或者删除流日志会触发告警。
执行频率	固定时间间隔：1分钟

查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许不开启VPC流日志的账号白名单。白名单中的账号可以不开启VPC流日志。
消除方法	禁止白名单以外的账号关闭或删除VPC流日志。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

VPC通用配置变更告警

告警ID	sls_app_audit_cis_at_vpc_conf_change
告警名称	VPC通用配置变更告警
版本号	1
类别	云平台、阿里云、CIS、VPC操作合规
作用	监控VPC的配置变更行为。VPC配置发生变更后触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为低。
外部配置	允许VPC配置变更的账号白名单。白名单中的账号进行VPC配置变更时，不会触发告警。
消除方法	禁止白名单以外的账号进行VPC配置变更。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.10. TDI操作合规

本文介绍TDI操作合规的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现TDI操作合规问题。

云安全中心网页防篡改功能关闭告警

告警ID	sls_app_audit_cis_at_sas_webshell_detection
告警名称	云安全中心网页防篡改功能关闭告警
版本号	1
类别	云平台、阿里云、CIS、TDI操作合规

作用	监控云安全中心网页防篡改功能的关闭行为。云安全中心网页防篡改功能应该保持开启，关闭后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许关闭网页防篡改功能的账号白名单。白名单账号关闭云安全中心网页防篡改功能，不会触发告警。
消除方法	禁止白名单以外的账号关闭云安全中心的网页防篡改功能。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.11. 云防火墙操作合规

本文介绍云防火墙操作合规的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现云防火墙操作合规问题。

云防火墙控制策略变更告警

告警ID	sls_app_audit_cis_at_cloudfirewall_conf_change
告警名称	云防火墙控制策略变更告警
版本号	1
类别	云平台、阿里云、CIS、云防火墙操作合规
作用	监控云防火墙的控制策略变更行为。云防火墙的控制策略发生变更后会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为中。
外部配置	允许云防火墙控制策略变更的账号白名单。白名单中的账号进行云防火墙控制策略变更时，不会触发告警。
消除方法	禁止白名单以外的账号进行云防火墙控制策略变更。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.12. API调用

本文介绍API调用的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现API调用问题。

未授权的API调用告警

告警ID	sls_app_audit_cis_at_unauth_apicall
告警名称	未授权的API调用告警
版本号	1
类别	云平台、阿里云、CIS、API调用
作用	监控未授权的API调用行为。未授权API调用次数小于等于规则参数配置中设定的未授权调用的最大次数，否则会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	• 严重度：严重、高、中、低、报告。默认值为中。 • 未授权调用的最大次数：每2分钟，允许每个IP地址对同一个服务发起未授权API调用的最大次数。默认值为5。
外部配置	允许未授权API调用的IP地址白名单。白名单中的IP地址对服务发起未授权API调用时，不会触发告警。
消除方法	禁止白名单以外的IP地址发起过多的未授权API调用。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

1.11.2.13. K8s安全

本文介绍K8s安全的告警规则，包括K8s错误事件过多、频繁删除事件等。通过设置并开启告警规则，可及时触发告警，有助于您快速发现K8s安全问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- K8s Warning事件数过多告警
- K8s频繁删除事件告警
- K8s错误事件数过多告警

K8s Warning事件数过多告警

告警ID	sls_app_audit_container_at_k8s_warn
告警名称	K8s Warning事件数过多告警
版本号	1

类别	云平台、阿里云、容器安全、K8s安全
作用	监控K8s集群的Warning事件。K8s集群上的Warning事件大于等于规则参数Warning事件数的阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为K8s Warning事件数过多告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为中。
	• Warning事件数的阈值：每2分钟内，一个K8s集群上报Warning事件的最大次数。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• K8s集群名称：待监控的K8s集群名称（支持正则表达式）。默认值 `.*` ，表示监控该阿里云账号下的所有K8s集群。

外部配置	无
消除方法	检查Warning事件数过多的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **K8s事件中心**的开关。

K8s频繁删除事件告警

告警ID	sls_app_audit_container_at_k8s_del
告警名称	K8s频繁删除事件告警
版本号	1
类别	云平台、阿里云、容器安全、K8s安全
作用	监控K8s集群的频繁删除事件。K8s集群上的删除事件大于等于规则参数频繁删除的次数阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为K8s频繁删除事件告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 频繁删除的次数阈值：每2分钟内，一个K8s集群删除事件的最大次数。默认值为5。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• K8s集群名称：待监控的K8s集群名称（支持正则表达式）。默认值 `.*` ，表示监控该阿里云账号下的所有K8s集群。

外部配置	无
消除方法	检查发生频繁删除事件的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **K8s审计日志**的开关。

K8s错误事件数过多告警

告警ID	sls_app_audit_container_at_k8s_err
告警名称	K8s错误事件数过多告警
版本号	1
类别	云平台、阿里云、容器安全、K8s安全
作用	监控K8s集群的错误事件。K8s集群上的Error事件大于规则参数错误事件数的阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为K8s错误事件数过多告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 错误事件数的阈值：每2分钟内，一个K8s集群上报错误事件的最大次数。默认值为5。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• K8s集群名称：待监控的K8s集群名称（支持正则表达式）。默认值 `.*` ，表示监控该阿里云账号下的所有K8s集群。

外部配置	无
消除方法	检查错误事件数过多的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **K8s事件中心**的开关。

1.11.2.14. RDS安全

本文介绍RDS安全的告警规则。通过设置告警规则，可及时触发告警，有助于您快速发现RDS安全问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- RDS慢SQL检测
- RDS大批量数据删除告警
- RDS外网访问检测
- RDS查询SQL平均执行时间监控告警
- RDS数据库更新峰值监控告警
- RDS数据库查询峰值监控告警
- RDS实例释放告警
- RDS高频访问IP检测
- RDS更新SQL平均执行时间监控告警
- RDS登录失败次数过多告警
- RDS大批量数据修改事件告警
- RDS危险的SQL执行告警
- RDS SQL执行错误数过多告警

RDS慢SQL检测

告警ID	sls_app_audit_db_at_rds_slow_sql

告警名称	RDS慢SQL检测
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS SQL执行是否为慢SQL。RDS SQL执行时间大于等于规则参数慢SQL时间阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS慢SQL检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 慢SQL时间阈值：SQL执行时间大于该阈值时，判定为慢SQL。默认值为5000微妙。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控阿里云账号下的所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控阿里云账号下的所有数据库。

外部配置	无
消除方法	检查出现慢SQL的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS **SQL审计日志**的开关。

RDS大批量数据删除告警

告警ID	sls_app_audit_db_at_rds_batch_del_sql
告警名称	RDS大批量数据删除告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS是否大量删除数据。删除的RDS数据行数大于等于规则参数大批量删除界定阈值时，会触发告警。
执行频率	固定时间间隔：1分钟

查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS大批量数据删除告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 大批量删除界定阈值：删除数据行数的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控阿里云账号下的所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控阿里云账号下的所有数据库。

外部配置	无
消除方法	检查发生大批量删除事件的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS外网访问检测

告警ID	sls_app_audit_db_at_rds_internet_access
告警名称	RDS外网访问检测
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS是否被外网IP地址访问。RDS被外网IP地址访问时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许通过外网访问的RDS实例白名单。白名单中的RDS实例被外网IP地址访问时，不会触发告警。
消除方法	禁止白名单以外的RDS实例被外网IP地址访问。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS查询SQL平均执行时间监控告警

告警ID	sls_app_audit_db_at_rds_select_speed
告警名称	RDS查询SQL平均执行时间监控告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS每条查询SQL执行平均时间。RDS SQL查询语句平均执行时间大于等于规则参数SQL平均执行时间阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS查询SQL平均执行时间监控告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• SQL平均执行时间阈值：查询语句SQL平均执行时间的最大值。默认值为0.005秒/条。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控该阿里云账号下的所有数据库。

外部配置	无
消除方法	检查查询SQL的平均执行时间过长的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS数据库更新峰值监控告警

告警ID	sls_app_audit_db_at_rds_update_peak
告警名称	RDS数据库更新峰值监控告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全

作用	监控RDS更新（增删改）峰值。RDS更新（增删改）峰值大于等于规则参数更新峰值阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS数据库更新峰值监控告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 更新峰值阈值：RDS更新（增删改）峰值。默认值为100行/秒。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查更新峰值过高的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS数据库查询峰值监控告警

告警ID	sls_app_audit_db_at_rds_query_peak
告警名称	RDS数据库查询峰值监控告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS查询峰值。RDS查询峰值大于等于规则参数查询峰值阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS数据库查询峰值监控告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 查询峰值阈值：RDS查询峰值。默认值为1000行/秒。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查查询峰值过高的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS实例释放告警

告警ID	sls_app_audit_db_at_rds_instance_del
告警名称	RDS实例释放告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS实例释放异常。RDS实例被释放时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	无
消除方法	检查被释放的RDS实例是否属于正常释放。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS高频访问IP检测

告警ID	sls_app_audit_db_at_rds_visit

告警名称	RDS高频访问IP检测
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控同一个IP地址对RDS实例访问频率是否异常。同一个IP地址对RDS实例访问频率大于等于规则参数高频访问阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS高频访问IP检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 高频访问阈值：每2分钟内，同一个IP地址对一个RDS实例的访问次数最大值。默认值为30次。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。

外部配置	RDS高频访问IP地址白名单。白名单中的IP地址对RDS实例发起高频访问时，不会触发告警。
消除方法	检查高频访问RDS实例的IP地址是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS **SQL审计日志**的开关。

RDS更新SQL平均执行时间监控告警

告警ID	sls_app_audit_db_at_rds_update_speed
告警名称	RDS更新SQL平均执行时间监控告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS每条更新（增删改）SQL执行平均时间。RDS更新（增删改）SQL平均执行时间大于等于规则参数SQL平均执行时间阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS更新SQL平均执行时间监控告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• SQL平均执行时间阈值：更新（增删改）SQL平均执行时间的最大值。默认值为0.005秒/条。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查更新（增删改）SQL的平均执行时间过长的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS **SQL审计日志**的开关。

RDS登录失败次数过多告警

告警ID	sls_app_audit_db_at_rds_login_err_cnt
告警名称	RDS登录失败次数过多告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控登录RDS实例失败次数是否异常。一个RDS实例在5分钟内登录失败次数大于等于规则参数最大失败登录次数时，会触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS登录失败次数过多告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 最大失败登录次数：一个RDS实例5分钟内允许登录失败次数的最大值。默认值为3次。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。

外部配置	无
消除方法	检查登录失败次数过的RDS实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS大批量数据修改事件告警

告警ID	sls_app_audit_db_at_rds_batch_update_sql
告警名称	RDS大批量数据修改事件告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS大量修改数据是否异常。RDS大量修改数据行数大于等于规则参数大规模修改界定阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS大批量数据修改事件告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 大规模修改界定阈值：修改数据行数的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查发生大批量数据修改事件的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS危险的SQL执行告警

告警ID	sls_app_audit_db_at_rds_danger_sql
告警名称	RDS危险的SQL执行告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS是否存在执行危险SQL。RDS出现执行危险SQL时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS危险的SQL执行告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查执行危险SQL的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

RDS SQL执行错误数过多告警

告警ID	sls_app_audit_db_at_rds_sql_err_cnt
告警名称	RDS SQL执行错误数过多告警
版本号	1
类别	云平台、阿里云、数据库安全、RDS安全
作用	监控RDS SQL执行错误次数是否异常。一个RDS实例的SQL执行错误次数大于等于规则参数最大错误次数时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为RDS SQL执行错误数过多告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 最大错误次数：一个RDS实例2分钟内允许SQL执行错误的最大次数。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• RDS实例ID：待监控的RDS实例ID（支持正则表达式）。默认值 `.*` ，表示监控所有RDS实例。
	• 数据库名称：待监控的数据库名称（支持正则表达式）。默认值 `.*` ，表示监控所有数据库。

外部配置	无
消除方法	检查SQL执行错误次数过多的RDS数据库是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开RDS SQL审计日志的开关。

1.11.2.15. SLB流量安全

本文介绍SLB（阿里云负载均衡）流量安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现SLB流量安全问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 负载均衡响应报文长度异常检测
- 负载均衡请求报文长度异常检测
- 负载均衡平均响应延迟过高告警
- 负载均衡HTTP访问协议开启告警
- 负载均衡访问UV异常检测
- 负载均衡访问PV异常检测

负载均衡响应报文长度异常检测

告警ID	sls_app_audit_dataflow_at_slb_resp_detc
告警名称	负载均衡响应报文长度异常检测
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全

作用	检测负载均衡（SLB）响应报文长度异常。响应报文长度的异常点个数大于等于规则参数异常点个数的阈值时，会触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	• 告警名称：告警实例的名称，默认为负载均衡响应报文长度异常检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 异常点个数的阈值：每分钟统计一个平均的响应报文长度，4小时内响应报文长度的异常点个数的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• SLB实例名称：待监控的SLB实例名称（支持正则表达式）。默认值 `.*` ，表示监控您操作账号绑定的所有SLB实例。

外部配置	无
消除方法	检查响应报文长度异常点过多的负载均衡实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开SLB**7层访问日志**的开关。

负载均衡请求报文长度异常检测

告警ID	sls_app_audit_dataflow_at_slb_req_detc
告警名称	负载均衡请求报文长度异常检测
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全
作用	检测负载均衡（SLB）请求报文长度异常。请求报文长度的异常点个数大于等于规则参数异常点个数的阈值时，会触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	• 告警名称：告警实例的名称，默认为负载均衡请求报文长度异常检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 异常点个数的阈值：每分钟统计一个平均的请求报文长度，4小时内请求报文长度的异常点个数的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• SLB实例名称：待监控的SLB实例名称（支持正则表达式）。默认值 `.*` ，表示监控您操作账号绑定的所有SLB实例。

外部配置	无
消除方法	检查请求报文长度异常点过多的负载均衡实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开SLB**7层访问日志**的开关。

负载均衡平均响应延迟过高告警

告警ID	sls_app_audit_dataflow_at_slb_latency
告警名称	负载均衡平均响应延迟过高告警
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全
作用	检测负载均衡（SLB）实例平均响应延迟过高。负载均衡实例平均响应时长大于等于规则参数平均响应延迟阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为负载均衡平均响应延迟过高告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 平均响应延迟阈值：每2分钟内，负载均衡实例响应延迟的最大值。默认值为0.5秒。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• SLB实例名称：待监控的SLB实例名称（支持正则表达式）。默认值 `.*` ，表示监控您操作账号绑定的所有SLB实例。

外部配置	无
消除方法	检查平均响应延迟过高的负载均衡实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开SLB**7层访问日志**的开关。

负载均衡HTTP访问协议开启告警

告警ID	sls_app_audit_dataflow_at_slb_http
告警名称	负载均衡HTTP访问协议开启告警
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全
作用	检测负载均衡（SLB）是否通过HTTPS协议访问服务端。负载均衡通过HTTP协议访问服务端时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	严重度：严重、高、中、低、报告。默认值为高。
外部配置	允许开启HTTP访问协议的负载均衡实例白名单。白名单中的负载均衡实例开启HTTP访问协议后，不会触发告警。
消除方法	禁止白名单以外的负载均衡实例开启HTTP访问协议。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开操作审计（ActionTrail）**操作日志**的开关。

负载均衡访问UV异常检测

告警ID	sls_app_audit_dataflow_at_slb_uv_detc

告警名称	负载均衡访问UV异常检测
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全
作用	检测负载均衡（SLB）访问UV是否异常。负载均衡实例访问UV个数大于等于规则参数UV异常点个数的阈值时，会触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	• 告警名称：告警实例的名称，默认为负载均衡访问UV异常检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• UV异常点个数的阈值：每分钟统计1个UV值，每4小时内负载均衡访问UV异常的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• SLB实例名称：待监控的SLB实例名称（支持正则表达式）。默认值 `.*` ，表示监控您操作账号绑定的所有SLB实例。

外部配置	无
消除方法	检查UV异常点过多的负载均衡实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开SLB**7层访问日志**的开关。

负载均衡访问PV异常检测

告警ID	sls_app_audit_dataflow_at_slb_pv_detc
告警名称	负载均衡访问PV异常检测
版本号	1
类别	云平台、阿里云、流量安全、SLB流量安全
作用	检测负载均衡（SLB）访问PV是否异常。负载均衡实例访问PV个数大于等于规则参数PV异常点个数的阈值时，会触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	• 告警名称：告警实例的名称，默认为负载均衡访问PV异常检测。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• UV异常点个数的阈值：每分钟统计1个PV值，每4小时内负载均衡访问PV异常的最大值。默认值为10。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• SLB实例名称：待监控的SLB实例名称（支持正则表达式）。默认值 `.*` ，表示监控您操作账号绑定的所有SLB实例。

外部配置	无
消除方法	检查PV异常点过多的负载均衡实例是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开SLB**7层访问日志**的开关。

1.11.2.16. API网关流量安全

本文介绍API网关流量安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现API网关流量安全问题。

告警规则列表

支持的告警规则列表如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- API网关服务端平均延时过高告警
- API网关后端服务器错误率过高告警
- API网关请求成功率过低告警

API网关服务端平均延时过高告警

告警ID	sls_app_audit_dataflow_at_api_latency
告警名称	API网关服务端平均延时过高告警
版本号	1
类别	云平台、阿里云、流量安全、API网关流量安全
作用	监控API网关中的API请求的服务端平均延时。API网关中的API请求的服务端平均延时大于等于规则参数服务端平均延时阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为API网关服务端平均延时过高告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 服务端平均延时阈值：每2分钟内，API请求的服务端平均延时的最大值。默认值为100毫秒。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• API名称：待监控的API名称（支持正则表达式）。默认值 `.*` ，表示监控所有API。

外部配置	无
消除方法	检查服务端平均延时过高的API是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开API网关**访问日志**的开关。

API网关后端服务器错误率过高告警

告警ID	sls_app_audit_dataflow_at_api_err_rate
告警名称	API网关后端服务器错误率过高告警
版本号	1
类别	云平台、阿里云、流量安全、API网关流量安全
作用	监控API网关中API请求的后端服务器错误率。API网关中API请求的后端服务器错误率大于等于规则参数后端服务器错误率阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为API网关后端服务器错误率过高告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• 后端服务器错误率阈值：每2分钟内，API请求的后端服务器错误率最大值。默认值为0%。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• API名称：待监控的API名称（支持正则表达式）。默认值 `.*` ，表示监控所有API。

外部配置	无
消除方法	检查后端服务器错误率过高的API是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开API网关**访问日志**的开关。

API网关请求成功率过低告警

告警ID	sls_app_audit_dataflow_at_api_req_rate
告警名称	API网关请求成功率过低告警
版本号	1
类别	云平台、阿里云、流量安全、API网关流量安全
作用	监控API网关中API的请求成功率。API网关的API请求成功率低于规则参数API请求成功率阈值时，会触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	• 告警名称：告警实例的名称，默认为API网关请求成功率过低告警。您可以根据不同监控对象，命名不同的告警名称便于识别。
	• 严重度：严重、高、中、低、报告。默认值为高。
	• API请求成功率阈值：API请求的成功率最小值。默认值为95%。
	• 阿里云账号ID：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• API名称：待监控的API名称（支持正则表达式）。默认值 `.*` ，表示监控所有API。

外部配置	无
消除方法	检查请求成功率过低的API是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开API网关**访问日志**的开关。

1.11.2.17. OSS流量安全

本文介绍OSS流量安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现OSS流量安全问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- OSS流量异常检测
- OSS流入流量异常检测
- OSS流出流量异常检测
- OSS访问PV异常检测
- OSS访问UV异常检测
- OSS Bucket有效请求率过低告警
- OSS外网访问检测

OSS流量异常检测

告警ID	sls_app_audit_dataflow_at_oss_flow_detc
告警名称	OSS流量异常检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS的流入流量和流出流量。当流量的异常点个数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **流量异常点个数的阈值**：OSS流量异常点个数的阈值，默认值为10个。如果4小时内的流量异常点个数超过该阈值，则触发告警。
	每分钟统计一个流量值。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS流入流量异常检测

告警ID	sls_app_audit_dataflow_at_oss_inflow_detc
告警名称	OSS流入流量异常检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS的流入流量。当流入流量的异常点个数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	告警参数说明如下：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **入流量异常点个数的阈值**：OSS流入流量异常点个数的阈值，默认值为10个。如果4小时内的流入流量异常点个数超过该阈值，则触发告警。
	每分钟统计一个流入流量值。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS流出流量异常检测

告警ID	sls_app_audit_dataflow_at_oss_outflow_detc
告警名称	OSS流出流量异常检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS的流出流量。当流出流量的异常点个数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **出流量异常点个数的阈值**：OSS流出流量异常点个数的阈值，默认值为10个。如果4小时内的流出流量异常点个数超过该阈值，则触发告警。
	每分钟统计一个流量值。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS访问PV异常检测

告警ID	sls_app_audit_dataflow_at_oss_pv_detc
告警名称	OSS访问PV异常检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS的访问PV。当OSS访问PV的异常点个数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	告警参数说明如下所示： • **告警名称**：告警实例的名称，支持创建多个告警实例。 • **严重度**：告警严重度，包括严重、高、中、低、报告。 • **PV异常点个数阈值**：OSS访问PV异常点个数的阈值，默认值为10个。如果4小时内的PV异常点个数超过该阈值，则触发告警。 每分钟统计一个PV值。 • **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。 ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。 ◦ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。 • **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。 ◦ 您可以使用正则表达式 `.*` 进行配置。 ◦ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。
外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS访问UV异常检测

告警ID	sls_app_audit_dataflow_at_oss_uv_detc
告警名称	OSS访问UV异常检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS的访问UV。当OSS访问UV的异常点个数超过指定阈值时，触发告警。
执行频率	固定时间间隔：4小时
查询范围	过去4小时

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **UV异常点个数阈值**：OSS访问UV异常点个数的阈值，默认值为10个。如果4小时内的UV异常点个数超过该阈值，则触发告警。
	每分钟统计一个PV值。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS Bucket有效请求率过低告警

告警ID	sls_app_audit_dataflow_at_oss_req_rate
告警名称	OSS Bucket有效请求率过低告警
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS Bucket有效请求率。当OSS Bucket有效请求率低于指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **有效请求率阈值**：OSS Bucket的有效请求率的阈值，默认值为95%。如果OSS Bucket的有效请求率低于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
触发告警时的推荐**消除办法**	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开OSS**访问日志**开关。

OSS外网访问检测

告警ID	sls_app_audit_dataflow_at_oss_internet_access
告警名称	OSS外网访问检测
版本号	1
类别	云平台、阿里云、流量安全、OSS流量安全
作用	监控OSS Bucket外网访问情况。当OSS被外网访问时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	告警参数说明如下： **严重度**：告警严重度，包括严重、高、中、低、报告。
外部配置	添加阿里云账号和OSS Bucket白名单，白名单中的OSS Bucket被外网访问时，不会触发告警。
消除方法	请勿使用外网访问白名单以外的OSS Bucket。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开OSS**访问日志**开关。

1.11.2.18. K8s流量安全

本文介绍K8s流量安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现K8s流量安全问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- K8s Ingress后端平均响应延迟过高告警
- K8s Ingress请求成功率过低告警
- K8s Ingress平均请求延迟过高告警
- K8s非法访问次数过多告警

K8s Ingress后端平均响应延迟过高告警

告警ID	sls_app_audit_dataflow_at_ingress_resp
告警名称	K8s Ingress后端平均响应延迟过高告警
版本号	1
类别	云平台、阿里云、流量安全、K8s流量安全
作用	监控K8s Ingress的后端平均响应延迟。当K8s Ingress后端平均响应延迟高于指定阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **后端平均响应延迟阈值**：K8s Ingress后端平均响应延迟的阈值，默认值为500毫秒。如果2分钟内K8s Ingress的后端平均响应延迟高于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **K8s集群名称**：需要监控的K8s集群名称。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下所有的K8s集群名称。

外部配置	无
消除办法	检查触发告警的K8s集群是否存在异常。

前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **Ingress访问日志**的开关。

K8s Ingress平均请求延迟过高告警

告警ID	sls_app_audit_dataflow_at_ingress_latency
告警名称	K8s Ingress平均请求延迟过高告警
版本号	1
类别	云平台、阿里云、流量安全、K8s流量安全
作用	监控K8s Ingress的平均请求延迟。当K8s Ingress的平均请求延迟高于指定阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **平均请求延迟阈值**：K8s Ingress平均请求延迟的阈值，默认值为200毫秒。如果2分钟内K8s Ingress的平均响应延迟高于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **K8s集群名称**：需要监控的K8s集群名称。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*` ，表示监控目标阿里云账号下所有的K8s集群名称。

外部配置	无
消除办法	检查触发告警的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **Ingress访问日志**的开关。

K8s Ingress请求成功率过低告警

告警ID	sls_app_audit_dataflow_at_ingress_rate
告警名称	K8s Ingress请求成功率过低告警
版本号	1

类别	云平台、阿里云、流量安全、K8s流量安全
作用	监控K8s Ingress的请求成功率。当K8s Ingress请求成功率低于指定阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **后端平均响应延迟阈值**：K8s Ingress请求成功率的阈值，默认值为90%。如果2分钟内K8s Ingress的请求成功率低于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **K8s集群名称**：需要监控的K8s集群名称。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*`，表示监控目标阿里云账号下所有的K8s集群名称。

外部配置	无
消除办法	检查触发告警的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **Ingress访问日志**的开关。

K8s非法访问次数过多告警

告警ID	sls_app_audit_dataflow_at_k8s_visit
告警名称	K8s非法访问次数过多告警
版本号	1
类别	云平台、阿里云、流量安全、K8s流量安全
作用	监控K8s集群的访问情况。当K8s集群被非法访问的次数多于指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **非法访问次数阈值**：K8s集群被非法访问的次数的阈值，默认值为3次。如果2分钟内K8s集群被非法访问的次数超过该阈值时，触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **K8s集群名称**：需要监控的K8s集群名称。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*` ，表示监控目标阿里云账号下所有的K8s集群名称。

外部配置	无
消除办法	检查触发告警的K8s集群是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开Kubernetes **Ingress访问日志**的开关。

1.11.2.19. OSS数据安全

本文介绍OSS数据安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现OSS数据安全问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- OSS频繁删除对象告警
- OSS Bucket账号访问控制

OSS频繁删除对象告警

告警ID	sls_app_audit_storage_at_oss_obj_del
告警名称	OSS频繁删除对象告警
版本号	1
类别	云平台、阿里云、数据安全、OSS数据安全
作用	监控OSS Bucket的删除操作。当OSS Bucket中删除操作的次数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **频繁删除的阈值**：删除操作的阈值。默认值为10次。如果2分钟内某个OSS Bucket中删除操作的次数超过该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **Bucket名称**：需要监控的OSS Bucket名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值为 `.*` ，表示监控目标阿里云账号下的所有的OSS Bucket。

外部配置	无
消除办法	检查触发告警的OSS Bucket是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

OSS Bucket账号访问控制

告警ID	sls_app_audit_storage_at_oss_access_control
告警名称	OSS Bucket账号访问控制
版本号	1
类别	云平台、阿里云、数据安全、OSS数据安全
作用	监控OSS Bucket的访问控制。当目标OSS Bucket只能被指定的阿里云账号或RAM用户访问时，如果不在允许范围内的阿里云账号或RAM用户访问该OSS Bucket，则触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟
参数配置	告警参数说明如下所示： **严重度**：告警严重度，包括严重、高、中、低、报告。
外部配置	添加阿里云账号（RAM用户）和OSS Bucket白名单。白名单中的阿里云账号或RAM用户访问指定的OSS Bucket时，不会触发告警。
消除办法	请勿使用白名单以外的阿里云账号或RAM用户访问OSS Bucket。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开OSS**访问日志**开关。

1.11.2.20. NAS数据安全

本文介绍NAS数据安全的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现NAS数据安全问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 文件存储操作错误检测
- 文件存储大批量删除文件告警

文件存储操作错误检测

告警ID	sls_app_audit_storage_at_nas_err_op
告警名称	文件存储操作错误检测
版本号	1
类别	云平台、阿里云、数据安全、NAS数据安全
作用	监控NAS Volume的错误操作情况。当NAS Volume中错误操作的次数多于指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	参数如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **操作错误数的阈值**：操作错误的次数的阈值，默认值为5。如果2分钟内一个Volume中的操作错误次数大于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **Volume名称**：需要监控的Volume名称（支持正则）。
	  ◦ 您可以使用正则表达式 `.*` 进行配置。
	  ◦ 默认值为 `.*` ，表示监控目标阿里云账号下所有的Volume。

外部配置	无
消除办法	检查触发告警的NAS Volume是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置** > **云产品接入** > **全局配置**中打开NAS**访问日志**的开关。

文件存储大批量删除文件告警

告警ID	sls_app_audit_storage_at_nas_file_del
告警名称	文件存储大批量删除文件告警
版本号	1
类别	云平台、阿里云、数据安全、NAS数据安全
作用	监控NAS Volume的删除操作情况。当NAS Volume中删除操作的次数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	参数如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **大批量删除阈值**：删除操作的阈值。如果2分钟内某个NAS Volume中的删除操作次数超过该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **Volume名称**：需要监控的Volume名称（支持正则）。
	  ◦ 您可以使用正则表达式 `.*` 进行配置。
	  ◦ 默认值为 `.*` ，表示监控目标阿里云账号下所有的Volume。

外部配置	无
消除办法	检查触发告警的NAS Volume是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开NAS**访问日志**的开关。

1.11.2.21. WAF安全事件

本文介绍WAF安全事件的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现WAF安全事件问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 应用防火墙防护网站被攻击次数过多告警
- 应用防火墙有效请求率过低告警

应用防火墙防护网站被攻击次数过多告警

告警ID	sls_app_audit_secure_at_waf_attack
告警名称	应用防火墙防护网站被攻击次数过多告警
版本号	1
类别	云平台、阿里云、安全事件、WAF安全事件
作用	监控网站被攻击的情况。当应用防火墙所防护的网站被攻击的次数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **被攻击次数阈值**：网站被攻击次数的阈值，默认值为5次。如果2分钟内一个网站被攻击的次数超过该阈值时，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **网站（host）**：需要监控的网站名称。
	  ◦ 您可以使用正则表达式 `.*` 进行配置。
	  ◦ 默认值 `.*` 表示监控目标阿里云账号下所有被应用防火墙防护的网站。

外部配置	无
消除办法	检查触发告警的网站是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开应用防火墙（WAF）**访问日志**的开关。

应用防火墙有效请求率过低告警

告警ID	sls_app_audit_secure_at_waf_rate
告警名称	应用防火墙有效请求率过低告警
版本号	1
类别	云平台、阿里云、安全事件、WAF安全事件
作用	监控应用防火墙有效请求率。经应用防火墙（WAF）拦截过滤后，如果对网站的有效请求率低于指定的阈值，则触发告警。

执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **有效请求率阈值**：网站有效请求率的阈值，默认值为90%。过去2分钟内经应用防火墙拦截过滤后，如果对网站的有效请求率低于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **网站（host）**：需要监控的网站名称。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值 `.*` 表示监控目标阿里云账号下所有被应用防火墙防护的网站。

外部配置	无
消除办法	检查触发告警的网站是否存在异常，是否存在被攻击的事件。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开应用防火墙（WAF）**访问日志**的开关。

1.11.2.22. TDI安全事件

本文介绍TDI安全事件的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现TDI安全事件问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 云安全中心外网DNS请求成功率过低告警
- 云安全中心有效请求率过低告警
- 云安全中心新增告警数过多
- 云安全中心新增漏洞数过多
- 云安全中心高优先级告警数过多

云安全中心外网DNS请求成功率过低告警

告警ID	sls_app_audit_secure_at_sas_dns_rate
告警名称	云安全中心外网DNS请求成功率过低告警
版本号	1

类别	云平台、阿里云、安全事件、TDI安全事件
作用	监控云安全中心外网DNS请求成功率。当云安全中心的外网DNS请求成功率低于指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **请求成功率阈值**：请求成功率的阈值，默认值为90%。如果2分钟内云安全中心的外网DNS请求成功率低于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。

外部配置	无
消除办法	检查云安全中心的外网DNS请求事件是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云安全中心日志的开关。

云安全中心有效请求率过低告警

告警ID	sls_app_audit_secure_at_sas_rate
告警名称	云安全中心有效请求率过低告警
版本号	1
类别	云平台、阿里云、安全事件、TDI安全事件
作用	监控云安全中心的有效请求率。经云安全中心防护过滤后，如果对网站的有效请求率低于指定的阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **有效请求率阈值**：有效请求率的阈值，默认值为90%。如果过去2分钟内经云安全中心防护过滤后，对网站的有效请求率低于该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。
	• **网站（host）**：需要监控的网站名称（支持正则）。
	  ○ 您可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值 `.*` 表示监控目标阿里云账号下所有的网站。

外部配置	无
消除办法	检查云安全中心的请求事件是否存在异常，是否存在过多攻击事件。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云安全中心日志的开关。

云安全中心新增告警数过多

告警ID	sls_app_audit_secure_at_sas_new_alert
告警名称	云安全中心新增告警数过多
版本号	1
类别	云平台、阿里云、安全事件、TDI安全事件
作用	监控云安全中心告警情况。当云安全中心新增告警数超过指定的阈值时，则触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **新增告警数阈值**：新增告警数的阈值，默认值为2。如果5分钟内云安全中心新增告警数超过该阈值时，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。

外部配置	无
消除办法	检查云安全中心中新增的告警。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云安全中心日志的开关。

云安全中心新增漏洞数过多

告警ID	sls_app_audit_secure_at_sas_new_vul
告警名称	云安全中心新增漏洞数过多
版本号	1
类别	云平台、阿里云、安全事件、TDI安全事件
作用	监控云安全中心的漏洞情况。当云安全中心新增的漏洞数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **新增漏洞数阈值**：新增漏洞数的阈值，默认值为1。如果5分钟内云安全中心新增漏洞数超过该阈值时，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。

外部配置	无

消除办法	检查云安全中心中新增的漏洞。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云安全中心日志的开关。

云安全中心高优先级告警数过多

告警ID	sls_app_audit_secure_at_sas_ser_alert
告警名称	云安全中心高优先级告警数过多
版本号	1
类别	云平台、阿里云、安全事件、TDI安全事件
作用	监控云安全中心高优先级告警的情况。当云安全中心高优先级告警数超过指定的阈值时，触发告警。
执行频率	固定时间间隔：4分钟
查询范围	过去5分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **高优先级告警数阈值**：高优先级告警数的阈值，默认值为1。如果云安全中心内的高优先级告警数超过该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ◦ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ◦ 默认值为 `.*` ，表示监控审计服务下配置的所有阿里云账号。

外部配置	无
消除办法	检查云安全中心中的高优先级告警。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云安全中心日志的开关。

1.11.2.23. 云防火墙安全事件

本文介绍云防火墙安全事件的告警规则。通过设置并开启告警规则，可及时触发告警，有助于您快速发现云防火墙安全事件问题。

告警规则列表

支持的告警规则如下所示。设置告警参数、设置白名单等相关操作，请参见设置告警。

- 云防火墙流入流量拦截告警

- 云防火墙流出流量拦截告警

云防火墙流入流量拦截告警

告警ID	sls_app_audit_secure_at_cfw_in_block
告警名称	云防火墙流入流量拦截告警
版本号	1
类别	云平台、阿里云、安全事件、云防火墙安全事件
作用	监控云防火墙的流入流量拦截情况。当云防火墙对一个访问协议流入流量的拦截次数超过指定阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **流入流量拦截次数阈值**：流入流量拦截次数的阈值，默认值为10次。如果2分钟内云防火墙对一个访问协议的流入流量的拦截次数超过该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **访问协议名称**：需要监控的访问协议名称（支持正则）。
	  ○ 您还可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值 `.*` 表示监控目标阿里云账号下所有的访问协议。

外部配置	无
消除办法	检查云防火墙对流入流量的拦截事件，确认是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云防火墙**互联网访问日志**的开关。

云防火墙流出流量拦截告警

告警ID	sls_app_audit_secure_at_cfw_out_block
告警名称	云防火墙流出流量拦截告警
版本号	1
类别	云平台、阿里云、安全事件、云防火墙安全事件

作用	监控云防火墙的流出流量拦截情况。当云防火墙对一个访问协议流出流量的拦截次数超过指定阈值时，触发告警。
执行频率	固定时间间隔：1分钟
查询范围	过去2分钟

参数配置	告警参数说明如下所示：
	• **告警名称**：告警实例的名称，支持创建多个告警实例。
	• **严重度**：告警严重度，包括严重、高、中、低、报告。
	• **流出流量拦截次数阈值**：流出流量拦截次数的阈值，默认值为10次。如果2分钟内云防火墙对一个访问协议的流出流量的拦截次数超过该阈值，则触发告警。
	• **阿里云账号ID**：需要监控的阿里云账号ID（支持正则）。
	  ○ 多个阿里云账号ID之间可以使用竖线（|）分隔。您还可以使用正则表达式 `.*` 进行配置，例如156133.*，表示监控以156133开头的阿里云账号。
	  ○ 默认值为 `.*`，表示监控审计服务下配置的所有阿里云账号。
	• **访问协议名称**：需要监控的访问协议名称（支持正则）。
	  ○ 您还可以使用正则表达式 `.*` 进行配置。
	  ○ 默认值 `.*` 表示监控目标阿里云账号下所有的访问协议。

外部配置	无
消除办法	检查云防火墙对流出流量的拦截事件，确认是否存在异常。
前提条件	确保已在日志审计服务中的**审计配置 > 云产品接入 > 全局配置**中打开云防火墙**互联网访问日志**的开关。

1.12. 最佳实践

1.12.1. 使用资源目录进行跨账号日志采集与同步授权

日志审计服务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。在多账号场景下，您可以使用资源目录管理账号。本文介绍如何使用资源目录进行跨账号日志采集与同步授权。

前提条件

- 已创建成员，即待采集日志的云产品涉及的所有阿里云账号均已加入资源目录中。

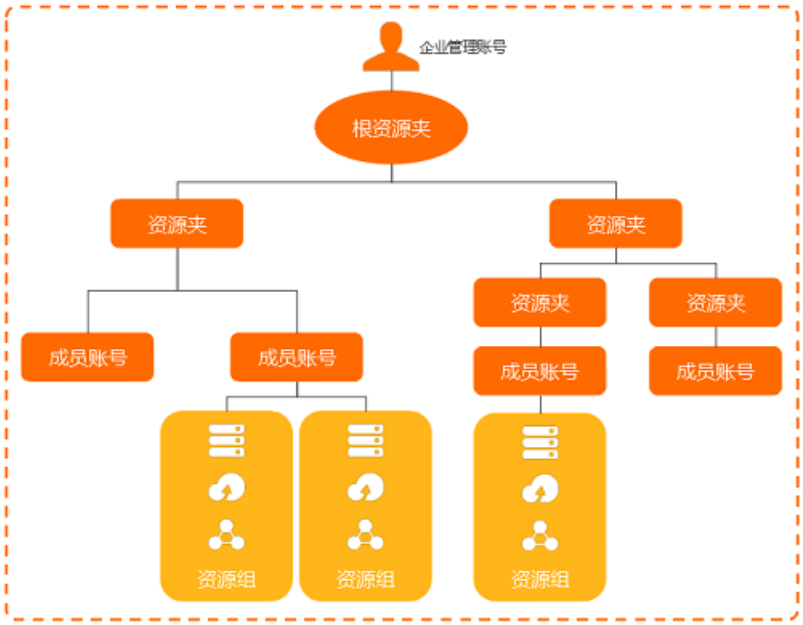
  您可通过创建或邀请成员的方式将阿里云账号加入到资源目录中。更多信息，请参见创建成员、邀请阿里云账号加入资源目录。
- 中心Project所在账号已开通日志服务。
- 待采集日志的云产品已开启相应的服务。更多信息，请参见云产品覆盖及相关资源。

背景信息

日志审计服务在继承现有日志服务所有功能外，还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。在使用日志审计服务时，您可以使用账号密钥辅助授权方式和手动授权方式完成授权，授予日志服务采集相关云产品日志的权限以及授权多个阿里云账号之间的同步汇集。在多账号场景下，您可以使用资源目录管理账号。更多信息，请参见日志审计服务。

资源目录是阿里云面向企业客户提供的一套多级资源（账号）关系管理服务。资源目录服务的本质：建立一套与您的企业相关的，基于资源使用的关系结构。资源目录具有全局一致性的特点，方便您基于此关系结构，对企业内多个应用服务所对应的各种资源进行高效的规划、构建和管理。是阿里云面向企业客户提供的一套多级资源（账号）关系管理服务。更多信息，请参见资源目录。

资源目录支持您基于企业的业务或生态环境，让您方便的构建出体现资源关系的目录结构，并将企业多个账号分布到这个目录结构中的相应位置，从而形成资源间的多层级关系。企业可依赖设定的组织关系进行资源的集中管理，满足企业资源在财资、安全、审计及合规方面的管控需要。下图展示了资源目录的基本结构。



- 企业管理账号是资源目录的超级管理员，也是开通资源目录的初始账号，对其创建的资源目录和成员账号拥有完全控制权限。每个资源目录有且只有一个企业管理账号。为了确保企业管理账号的安全，建议您创建一个新的阿里云账号作为企业管理账号，避免将已有用途的云账号作为企业管理账号。更多信息，请参见企业管理账号。
- 资源夹是资源目录内的组织单元，通常用于指代企业的分公司、业务线或产品项目。每个资源夹下可以放置成员账号，并允许嵌套子资源夹，最终形成树形的资源组织关系。更多信息，请参见资源夹。
- 成员账号是阿里云账号在资源目录中的一种称呼。在资源目录内，成员账号作为资源容器，是一种资源分组单位。成员账号通常用于指代一个项目或应用，每个成员账号中的资源相对其他成员账号中的资源是物理隔离的。更多信息，请参见成员账号。

操作步骤

1. 通过资源目录访问中心Project所在账号。

   在资源目录内创建或邀请成员后，您可以从资源目录的成员账号中选取一个账号作为日志审计服务中心Project所在的阿里云账号。然后通过RAM用户、RAM角色或根用户访问中心Project所在的阿里云账号。

   - 通过RAM角色访问成员
   - 通过RAM用户访问成员
   - 通过根用户访问成员

2. 登录日志服务控制台。
3. 在**日志应用**区域，单击**日志审计服务**。
4. 在中心Project所在账号内进行日志审计采集的首次配置。

   如果该账号已完成首次配置，可跳过此步骤。

   i. 在左侧导航栏，单击**云产品接入 > 全局配置**。

   ii. 在**中心项目Project所在区域**下拉列表中，选择日志中心化存储的目标地域。

   - 中国：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）
   - 海外：新加坡、日本（东京）、德国（法兰克福）、印尼（雅加达）

   iii. 配置采集同步授权。

   日志审计服务支持手动授权和通过账号密钥辅助授权。

   - 通过账号密钥辅助授权：输入账号的AccessKey信息，AccessKey信息不会被保存，仅临时使用。

     此处AccessKey对应的RAM用户需具备RAM读写权限（例如已被授权AliyunRAMFullAccess策略）。具体操作，请参见授权RAM用户。

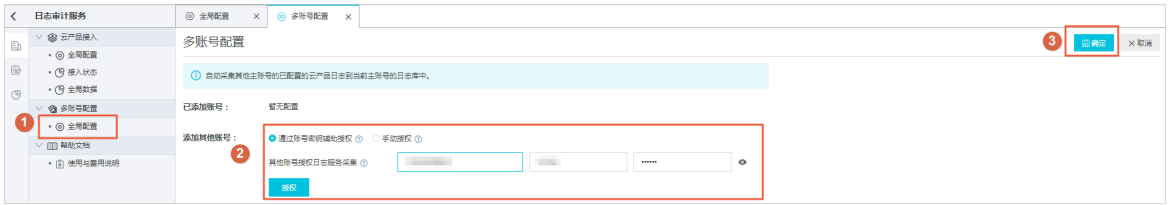
   - 手动授权：更多信息，请参见自定义授权日志采集与同步。

   iv. 在云产品列表中，选择需开启日志审计功能的云产品，并配置存储时间。

   如果是SLB 7层访问日志、OSS访问日志、DRDS审计日志，还可以选择**同步到中心**。开启**同步到中心**后，区域化Project将作为中转，不需要存储很长时间，控制台会自动调整成推荐的时间。

   v. 单击**保存**。

5. 在中心Project所在账号内进行多账号采集配置。



   i. 在左侧导航栏中，单击**多账号配置 > 全局配置**。

   ii. 在**多账号配置**页面，单击**修改**。

   iii. 配置采集同步授权。

   日志审计服务支持手动授权和通过账号密钥辅助授权。

   - 通过账号密钥辅助授权：在**其他账号授权日志服务采集**文本框中输入其他账号的AccessKey信息及其阿里云账号ID。AccessKey信息不会被保存，仅临时使用。

     此处AccessKey对应的RAM用户需具备RAM读写权限（例如已被授权AliyunRAMFullAccess策略）。

   - 手动授权：输入阿里云账号ID，可配置多个。对应的账号权限配置请参见操作步骤。

6. 通过资源目录依次访问其他需要被采集同步的阿里云账号，并进行手动授权。

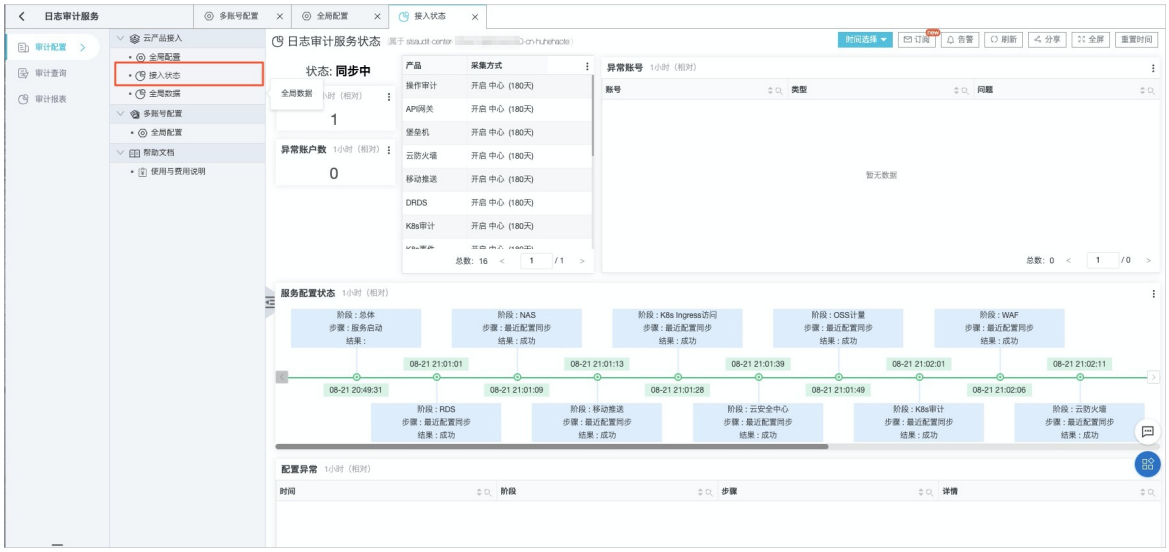
   在步骤5中，如果使用的是手动授权方式完成授权，则需要配置此步骤；如果使用的是通过账号密钥辅助授权方式完成授权，请跳过此步骤。

   i. 通过资源目录访问待授权的账号。更多信息，请参见步骤1。

   ii. 在该账号内进行跨账号采集配置手动授权。更多信息，请参见操作步骤中的步骤3。

7. 查看配置结果。

配置完成后，需要2分钟左右完成初始同步。如果出现异常，请根据页面提示信息进行调整。更多信息，请参见常见问题及错误排查。



2.数据实验室

2.1. 使用数据实验室

日志服务数据实验室为您提供各种场景的模拟日志数据及各种公共数据集数据，便于您熟悉日志服务的查询分析、查看报表等各种操作。本文介绍如何使用阿里云数据实验室。

前提条件

- 已开通日志服务。

  首次登录日志服务控制台时，根据页面提示开通日志服务。

- 已完成云资源访问授权。

  首次使用数据实验室时，在数据实验室的**首页**，单击**前往授权**完成云资源访问授权。

收费与限制

目前数据实验室提供网站访问日志、数据库审计日志、新冠疫情数据集等三个场景，均免费使用。

在数据实验室中Logstore仅用于存储模拟数据，数据保留7天，shard数量为1。您无法进行数据写入、编辑索引等操作。

场景选择

数据分为模拟数据和数据集。

- 模拟数据：根据模拟配置生成的数据。您可以根据需求修改部分字段值及时间范围等配置，数据模拟任务会根据您的配置产生数据。目前提供网站访问日志和数据库审计日志两种场景。
- 数据集：来自于各种场景的真实日志数据，目前提供新冠疫情数据集场景。

操作步骤

此处以数据库审计日志场景为例进行说明。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**数据实验室**中的**进入应用**。
3. 在**首页**页面中，单击**数据库审计日志**中的**初始化**。
4. 调整日志字段值、时间范围等配置。
   - 在**日志字段**页签中，包括字段名称、字段数据类型、字段随机参数和字段值等信息，您可以根据需

求，调整字段随机参数和字段值。

- 在**范围频率**页签中，可调整如下参数。

参数	说明
时间范围	指定模拟数据生成的起始时间与结束时间。如果不指定结束时间，会持续生成模拟数据。
分布模型	数据量分布模型，包括随机、周期与线性。您可以自定义变化周期、最小值、最大值和抖动值。 ■ 变化周期：数据量分布变化周期，支持秒、分钟、小时、天和周。 ■ 最小值：每秒最小数据量。 ■ 最大值：每秒最大数据量。 ■ 抖动值：例如设置**抖动**为0.1，模型数据量为count，则最终的数据量在[0.9*count, 1.1*count]之间随机。
异常点	在**分布模型**中所确定的数据量的基础上，随机选择某些周期，在这些周期内的数据量会出现异常变化，您可以指定出现异常点的概率和异常点数据量变化的倍率。 您可以配置多个异常选项，每个数据周期最多出现一个异常点，优先模拟排在第一个的异常选项。

5. 单击**开始导入**，数据实验室自动完成创建项目和Logstore、创建模拟任务、创建数据场景和数据模拟等任务。

   > **说明** 数据模拟任务需要大约1分钟的时间。

6. 数据导入完成后，单击**开始使用**。

   在**首页**页面的**数据库审计日志**区域，您可以执行查询日志、查看报表、重新导入数据、删除场景等操作。

   - 单击**查询分析**，进入**查询分析**页面，

     您可以进行查询分析操作，详情请参见查询与分析。

   - 单击**报表中心**，进入**报表中心**页面。

     默认提供RDS安全中心报表、RDS审计中心报表和RDS审计性能报表。单击对应的**查看报表**，可进入报表详情页面，进行仪表盘操作，详情请参见可视化概述。

   - 单击**数据配置**，进入**数据配置**页面。

     您可以调整日志的字段值、时间范围等配置，重新导入数据。

   - 单击**删除**，删除场景。

     在删除场景时，您可以选择是否要同步删除Logstore及相关报表。删除场景后，将停止数据模拟任务。

3.成本管家

3.1. 成本管家

日志服务推出成本管家功能，一键开通后自动导入账单，并提供可视化的账单分析报表，帮助您提高账单分析的效率。

背景信息

阿里云资源具备随时可用、规模弹性、规格丰富的特征，保证您在任意时刻都有足够的资源使用。在您使用云资源的同时，成本是个不容忽视的问题。阿里云的计费方式有按量付费和包年包月。对于按量付费方式，手工对账单进行统计分析不仅耗费时间和精力，准确性也没办法保证。日志服务的成本管家功能很好的解决了这个问题，将您从低效的账单获取和整理工作中解放出来，提高账单分析效率。

功能特点

日志服务提供的成本管家功能，一键开通后，会自动将账单从账单中心导入到日志库中。账单是一种时间序列的数据，而日志服务的主要功能就是对时间序列数据的采集、存储和分析，实现与账单数据的无缝对接，减少了账单分析人员80%的人力投入。成本管家的特点如下：

- 近实时采集：账单产生后一小时内上传到日志服务中。
- 定制报表：提供常见的账单分析场景，支持自动发送报告。
- 交互式分析：使用SQL分析账单数据，分析结果秒级可见。支持将分析规则保存到自定义报表中。
- 可视化：以图表的形式展示分析结果，更加直观。
- 机器学习算法：智能预测未来费用趋势，挖掘异常账单。
- 自定义告警：支持自定义告警功能，实时了解账单详情。
- 免费：账单分析涉及的数据存储和分析功能均不收费。

导入账单

1. 登录日志服务控制台。
2. 在**日志应用**中单击**成本管家**下的**进入应用**。
3. 在**成本管家**左侧，单击**设置**。
4. 导入账单设置。

   在导入账单步骤中进行如下设置。
   - **阿里云账单导入**：勾选后，会将本账号下所有的阿里云账单导入到日志服务中。
   - **首次导入历史账单**：首次导入您可以选择要导入历史账单的时间。
   - 访问账单权限：如果当前账号没有账单访问权限，请根据提示进行授权。
5. 订阅报告设置。

   在订阅报告步骤中进行如下设置。
   - **频率**：订阅后报告的发送频率。
   - **添加水印**：打开后会对账单中的敏感数据添加水印，以免关键信息泄露。
   - **通知列表**：可以选择**邮件**或者**WebHook-钉钉机器人**的方式发送订阅的报告。钉钉机器人的请求地址请参见WebHook-钉钉机器人进行获取。
6. （可选）设置告警。

您可以针对不同云产品设置不同的告警条件，当账单达到设置的告警条件，则触发告警，帮助您及时了解账单的使用量。

i. 单击**添加告警**。

ii. 设置告警条件。

根据需求配置以下参数：**选择产品**、**账单类型**、**判断条件**、**判断值类型**和**判断值大小**。

> **说明** 可以多次单击**添加告警**添加多个告警信息。

iii. 选择通知方式。

关于告警通知方式的操作及说明请参见通知方式。

7. 单击**创建/修改告警**完成账单设置。

功能说明

导入账单后，您可以单击**成本管家**下的**说明**，查看成本管家功能说明信息。包含产品说明、产品分析账单的使用、限制说明、账单字段说明等。

自定义分析

在自定义分析界面，您可以和操作其他日志库一样，对导入的账单进行查询分析，设置快速查询、保存仪表盘、设置告警等。

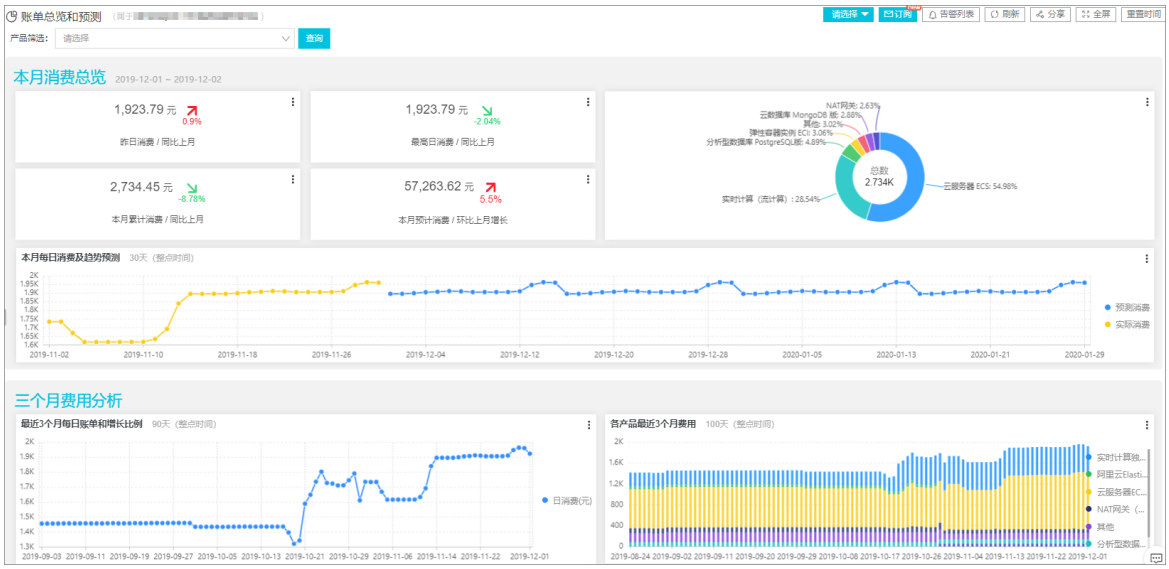
1. 单击左侧**成本管家**下的**自定义分析**。
2. 在**自定义分析**界面的查询分析输入框中，输入查询分析语句，对导入的账单进行查询分析。

该操作与其他日志库查询分析操作相同，具体请参见查询分析简介。

账单总览

成本管家提供内置的账单总览报表，展示当月及过去三个月的费用组成，并根据当前费用预测未来的费用趋势，帮助您合理的规划未来预算。该报表拥有和日志服务仪表盘相同的功能，详细介绍请参见可视化概述。

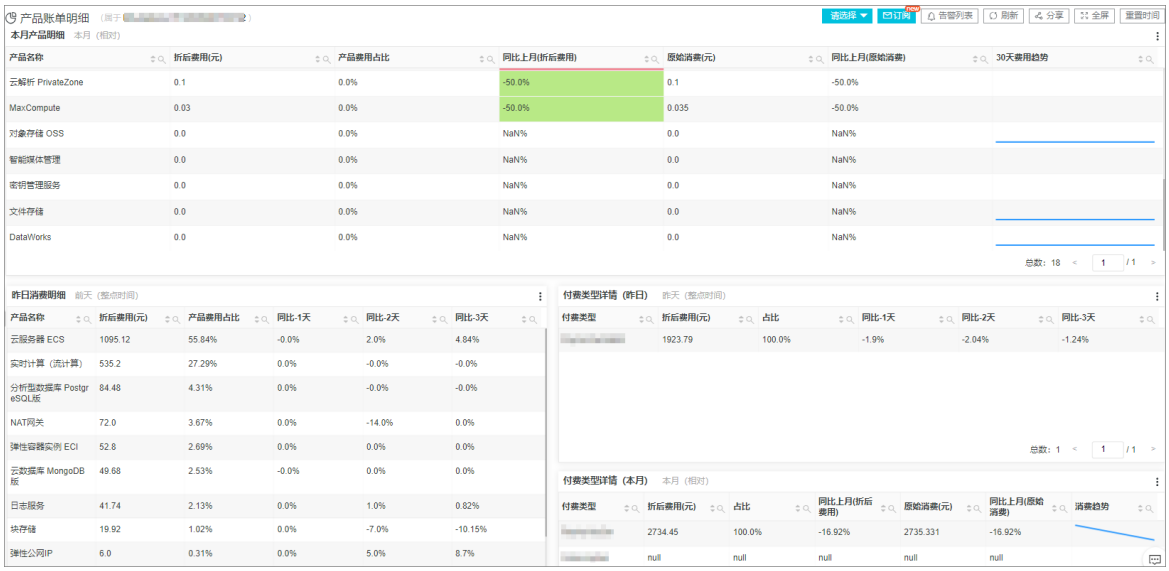
1. 单击左侧**成本管家**下的**总览**。
2. 在**总览**界面查看账单总览和预测信息。



账单明细

成本管家提供内置的账单明细报表，展示每个产品的账单明细和趋势，以及异常的账单信息。该报表拥有和日志服务仪表盘相同的功能，详细介绍请参见可视化概述。

1. 单击左侧**成本管家**下的**明细**。
2. 在**明细**界面查看产品消费明细。



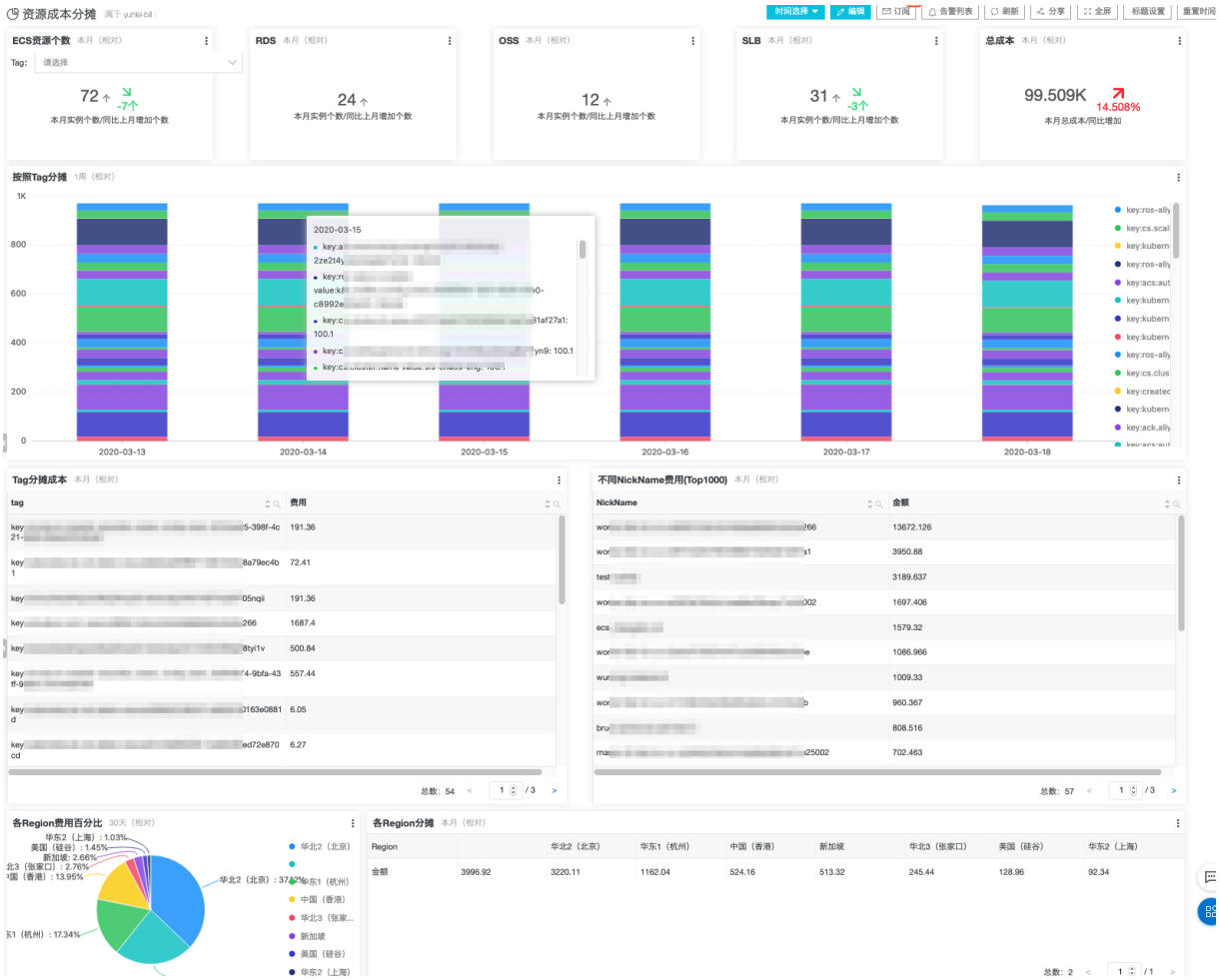
账单优化

成本管家提供内置的账单优化报表，根据产品账单详情，对按量付费产品自动推出包年包月的节省额度。

1. 单击左侧**成本管家**下的**优化**。
2. 在**优化**界面查看账单优化建议。

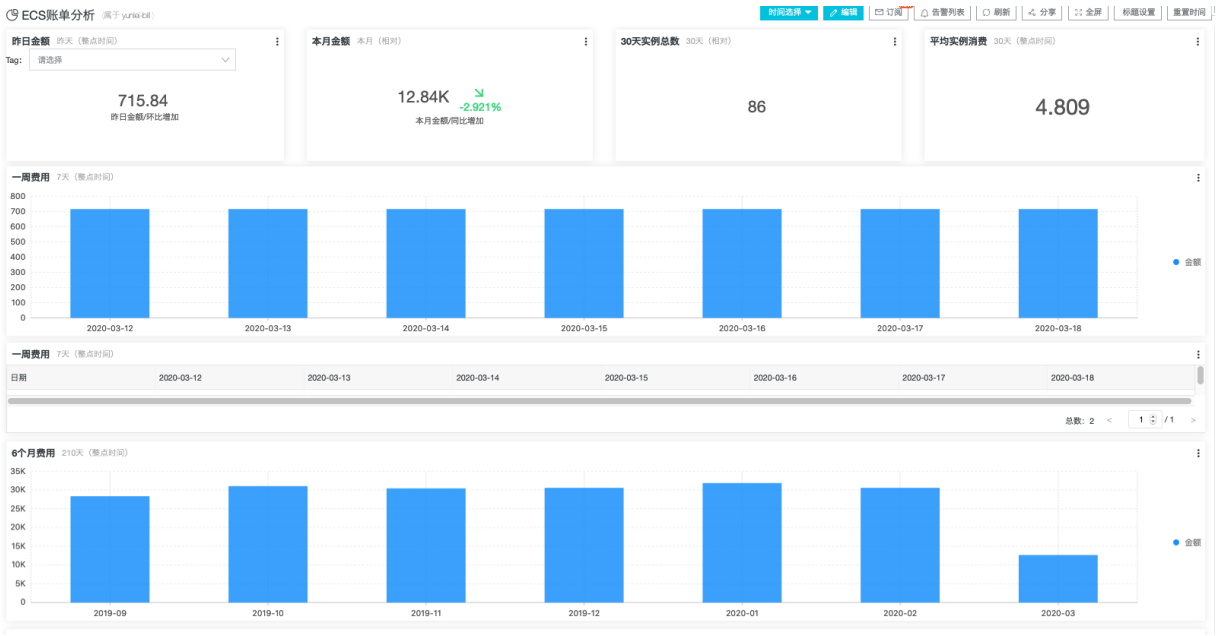
资源成本分摊

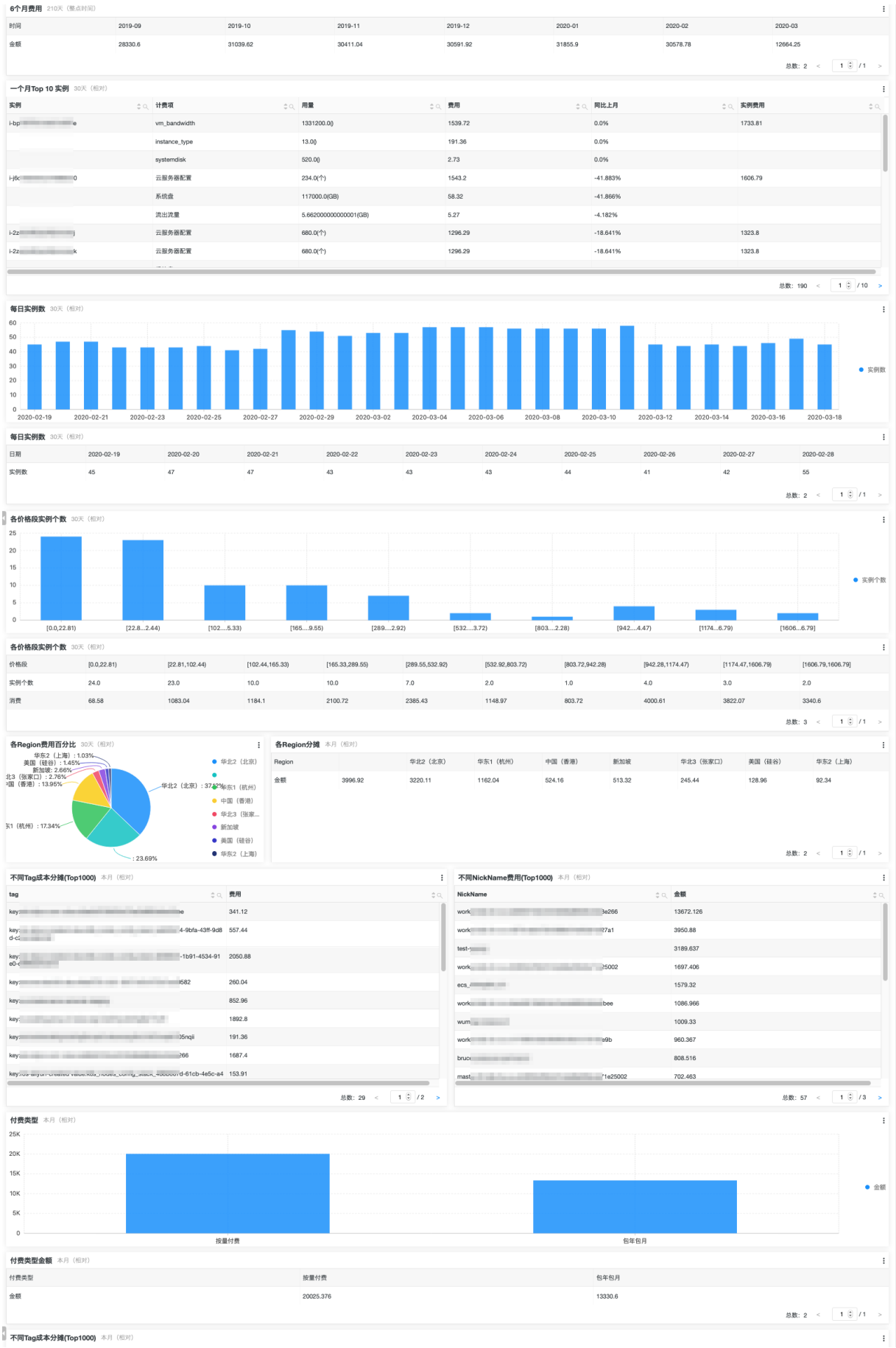
通过资源成本分摊报表，可以查看主要云资源的使用数目，以及按照tag、昵称等进行分账管理。

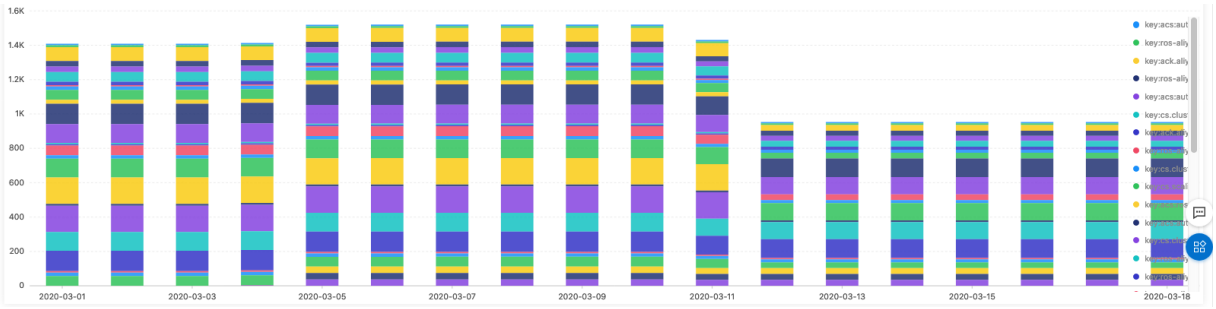


ECS 账单分析报表

通过ECS账单分析报表，可以查看ECS的使用情况，以及按照各个维度（region、Tag、昵称）进行分析。通过报表，可以整体把握ECS的使用有，适用于费用优化，成本分摊等场景。

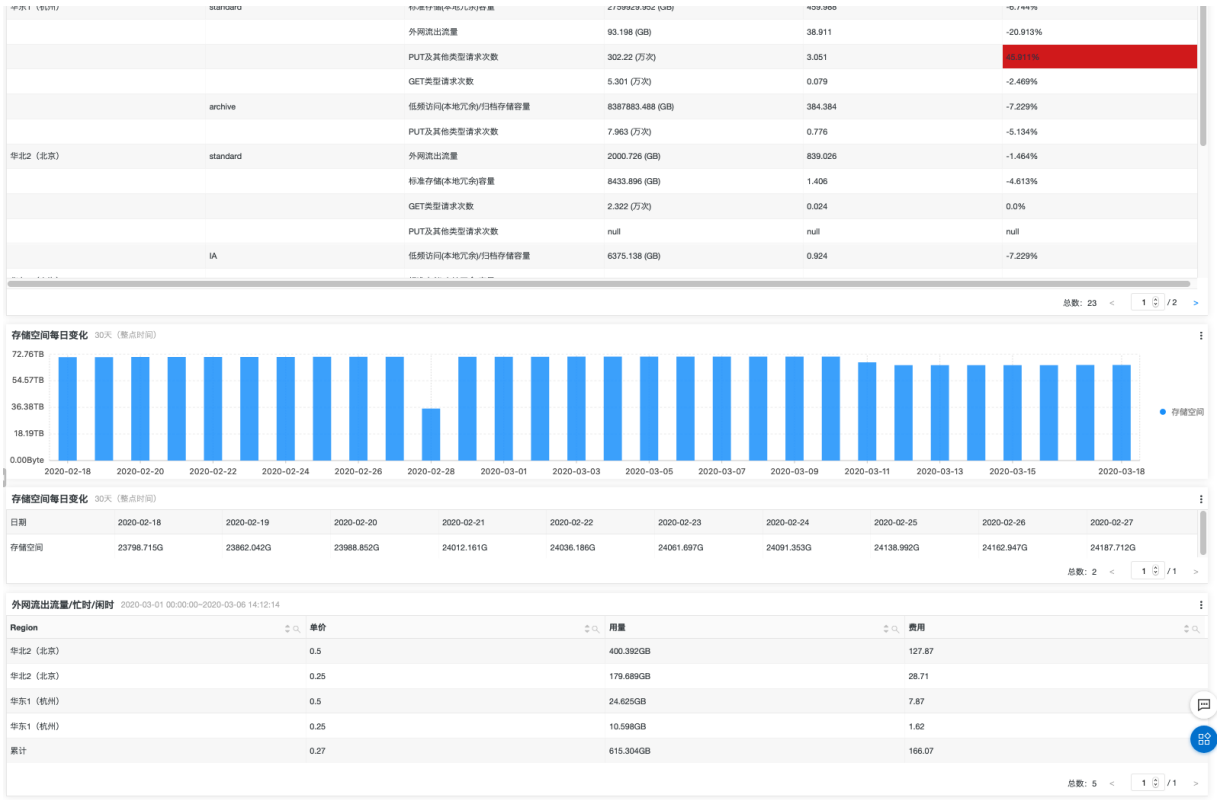



6个月费用 210天（整点时间）
时间 2019-09 2019-10 2019-11 2019-12 2020-01 2020-02 2020-03
金额 28330.6 31039.62 30411.04 30591.92 31855.9 30578.78 12664.25
一个月Top 10 实例 30天（相对）
实例 计费项 用量 费用 同比上月 实例费用
每日实例数 30天（相对）
实例数
日期 2020-02-19 2020-02-20 2020-02-21 2020-02-22 2020-02-23 2020-02-24 2020-02-25 2020-02-26 2020-02-27 2020-02-28
实例数 45 47 47 43 43 43 44 41 42 55
各价格段实例个数 30天（相对）
[0.0,22.81) [22.8...2.44) [102....5.33) [165....9.55) [289....2.92) [532....3.72) [803....2.28) [942....4.47) [1174...6.79) [1606...6.79]
各价格段实例个数 30天（相对）
价格段 [0.0,22.81) [22.81,102.44) [102.44,165.33) [165.33,289.55) [289.55,532.92) [532.92,803.72) [803.72,942.28) [942.28,1174.47) [1174.47,1606.79) [1606.79,1606.79]
实例个数 24.0 23.0 10.0 10.0 7.0 2.0 1.0 4.0 3.0 2.0
消费 68.58 1083.04 1184.1 2100.72 2385.43 1148.97 803.72 4000.61 3822.07 3340.6
各Region费用百分比 30天（相对）
华北2（北京）：37.22%
华东1（杭州）：17.34%
：23.69%
各Region分摊 本月（相对）
Region 华北2（北京） 华东1（杭州） 中国（香港） 新加坡 华北3（张家口） 美国（硅谷） 华东2（上海）
金额 3996.92 3220.11 1162.04 524.16 513.32 245.44 128.96 92.34
不同Tag成本分摊(Top1000) 本月（相对）
tag 费用
341.12
557.44
2050.88
260.04
852.96
1892.8
191.36
1687.4
153.91
不同NickName费用(Top1000) 本月（相对）
NickName 金额
13672.126
3950.88
3189.637
1697.406
1579.32
1086.966
1009.33
960.367
808.516
702.463
付费类型 本月（相对）
按量付费
包年包月
付费类型金额 本月（相对）
付费类型 按量付费 包年包月
金额 20025.376 13330.6
不同Tag成本分摊(Top1000) 本月（相对）



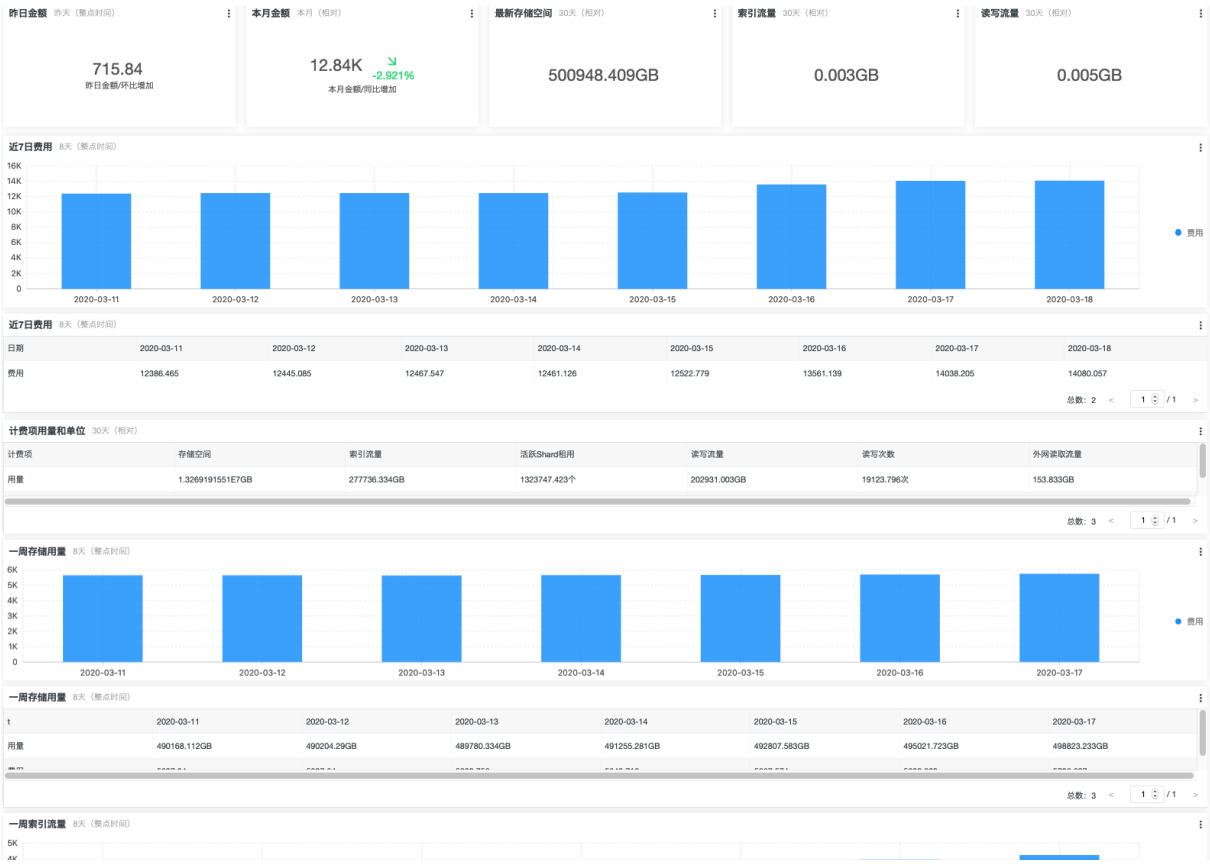
OSS账单分析

通过OSS账单分析报表，可以查看OSS整体费用，费用趋势，以及标准型存储、低频存储、归档型存储等不同类型的存储费用，各个计费项目的使用量和费用。您可根据实际使用情况调整存储类型，节省费用。



SLS账单分析

通过SLS账单分析报表，可以查看SLS的整体费用、费用趋势、各个计费项的用量以及存储空间和索引流量最多的Project和Logstore，可帮助客户优化SLS的使用成本。



3.2. 账单看板

日志服务成本管家提供账单看板，帮忙您更直观地查看云产品账单，包括每个云产品的每个计费项的使用量、原始金额、优惠金额、应付金额等信息。

操作步骤

1. 登录阿里云控制台首页。
2. 单击右上角的**自定义**。
3. 在左侧导航栏中，单击**成本管家.对账看板**后的**+**，把**成本管家.对账看板**模块添加到阿里云控制台首页。

4. 单击**保存**。
5. 通过**成本管家.对账看板**模块查看云产品账单。

   您还可以根据账期、产品筛选账单。

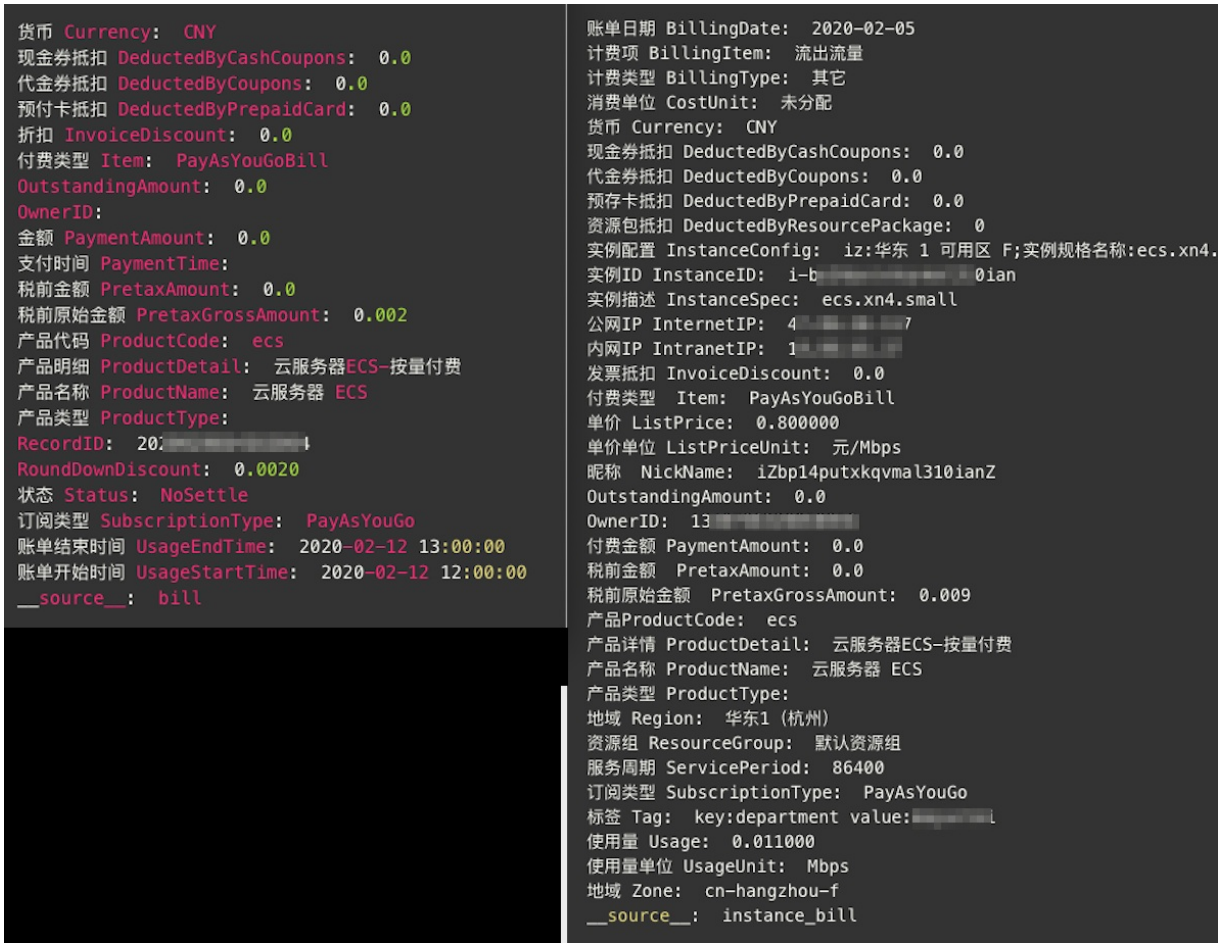
3.3. 使用SQL语句自定义分析账单

本文介绍在日志服务控制台上如何使用SQL语句自定义分析账单。

账单数据详情

账单数据包括以下两类数据：

- 左侧为账单数据，标识为 `source:bill` ，每个云产品在每个账单周期中产生一条记录。
- 右侧为实例账单数据，每个实例对应一条数据，包含实例的使用量、属性（TAG、NickName、名称等）、费用。标识为 `source:instance_bill` 。

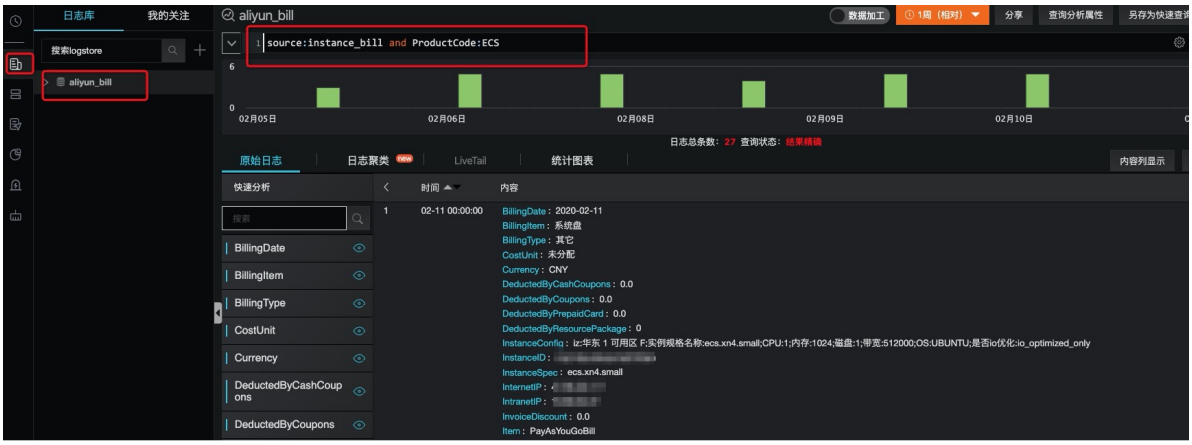


案例

成本管家中内置的报表仅是分析模板，提供分析案例。实际使用中，您可能有多种多样的需求，同一个模板无法满足。您可以通过SQL语句自定义分析账单，这里以ECS账单为例进行说明。

- 搜索关心的账单

  在所有的账单中，您可能只关心某些账单，例如：只想要获取ECS实例账单，那么只需要在名为aliyun_bill的Logstore中使用SQL语句 `source:instance_bill and ProductCode:ECS` 即可获取结果，如下图所示。更多搜索语法请参见查询语法。



- 简单聚合，获取总的账单费用

使用以下SQL语句获取ECS实例的总费用。在计算结果中单击**添加到仪表盘**，即可创建一个专属的仪表盘。

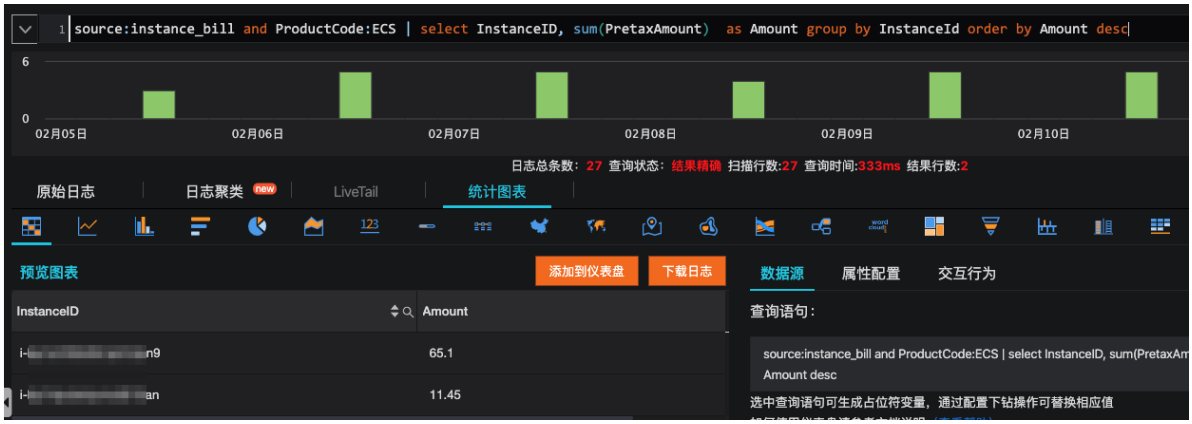
```
source:instance_bill and ProductCode:ECS | select sum(PretaxAmount)
```

- 分组聚合。

  使用以下SQL语句，获取每个ECS实例的账单总额。

  ```
  source:instance_bill and ProductCode:ECS | select InstanceID, sum(PretaxAmount) as Amount group by InstanceId order by Amount desc
  ```

  本案例通过实例维度进行分析，如果您想要通过其他维度（例如Region、昵称等）分析，只需更换SQL语句中 `group by` 后面的维度。



- 同比环比分析
  - 计算本月费用，同比上月的增长率。

    ```
    source:bill | select diff[1] as "本月费用", diff[2] as "上月费用", diff[3]*100-100 as "同比增加%" from(select compare(amount,604800) as diff from( select sum(PretaxAmount) as amount from log ))
    ```

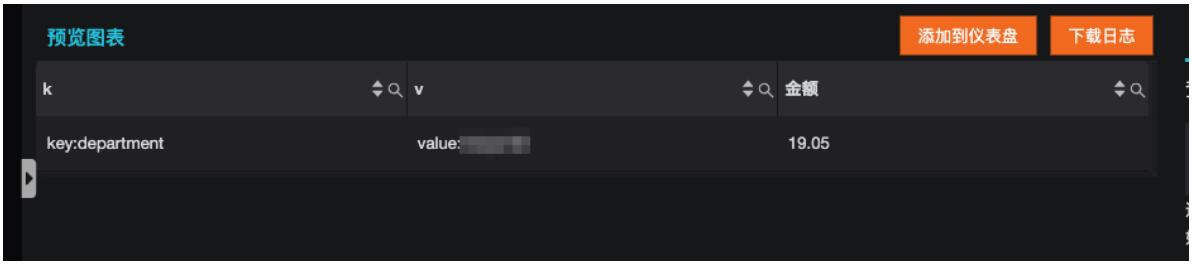
  - 按照产品，与上月进行同比分析。

    ```
    source:bill | select ProductCode, diff[1] as "本月费用", diff[2] as "上月费用", diff[3]*100-100 as "同比增加%" from(select productcode, compare(amount,604800) as diff from( select ProductCode, sum(PretaxAmount) as amount from log group by ProductCode ) group by productcode )
    ```

- 利用Tag做分账管理

  目前多种产品已支持Tag，您可以通过Tag完成分账。Tag中包含多个key-value，通过解析不同的key-value，计算每一对key-value的费用额度。

  ```
  source: instance_bill and ecs | select k,v , round(sum(PretaxAmount),3) "金额" from( select split_to_map(Tag,';',' ') as tags ,PretaxAmount from log where tag <>'' ),unnest(tags) as t(k,v) group by k,v order by "金额" desc limit 1000
  ```



3.4. 设置产品预算管理

日志服务成本管家从账号、财务单元、资源标签三个维度监控您的阿里云产品使用成本。您可以设定预算阈值并开启告警，便于您随时掌握成本达到阈值限制或超出阈值限制的情况。

创建预算管理任务

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**成本管家**。

   您也可以在阿里云控制台首页的**成本管家.对账看板**区域，单击**查看更多报表**，进入成本管家。具体操作，请参见账单看板。
3. 在左侧导航栏，选择**成本治理 > 预算管理**。
4. 在**预算管理**页签中，单击**创建预算**。
5. 选择预算类型，单击**下一步**。

   成本管家支持通过账号维度、标签维度和财务单元维度制定预算。
6. 设置成本预算相关参数，单击**下一步**。

参数	描述
预算名称	设置预算名称。 名称只能包含字母、汉字、短划线（-）和下划线（_），长度为4~20个字节。
预算周期	选择预算周期。您可以按每周、每月、每季度和自定义方式选择预算周期。
开始时间	设置该预算管理任务开始执行的时间。
财务单元	当您在步骤5中选择**按财务单元制定预算**时，需要添加财务单元。 您可以在阿里云用户中心的**企业财务 > 财务单元**中，添加财务单元。具体操作，请参见财务单元。
成本分配标签	当您在步骤5中选择**按标签维度制定预算**时，需要添加标签。 该标签为各个云产品资源的标签。每个标签均为一个键值对。对于每个资源，每个标签键都必须是唯一的，每个标签键只能有一个值。关于标签的设置，请参见各个云产品文档。

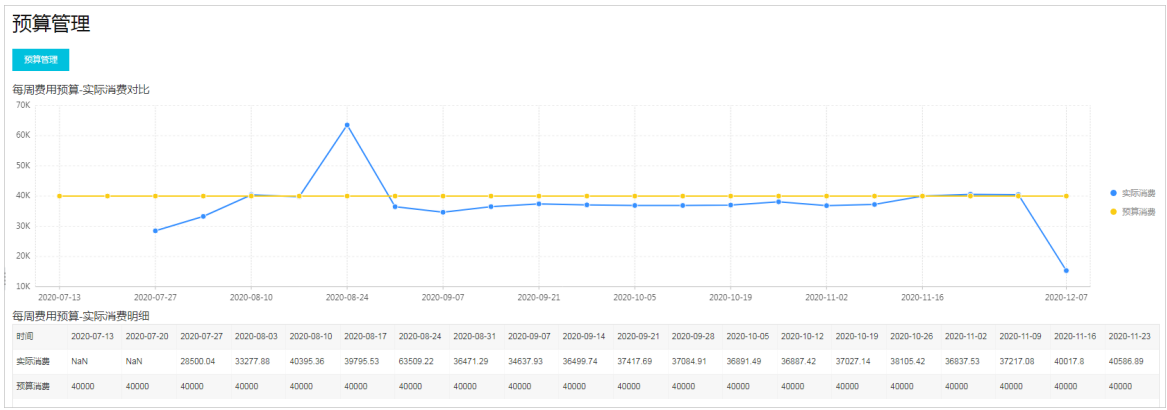
参数	描述
预算金额	设置预算金额，具体配置如下所示： ○ 固定金额：每个预算周期内，设置固定预算金额。 ○ 每周预算金额：当**预算周期**为**每周**时，可为开始时间后的12周分别设置预算金额。您可以手动输入预算金额，还可以设置起始预算和增长比例，系统将自动填充预算金额。 ○ 月度预算金额：当**预算周期**为**每月**时，可为开始时间后的12个月分别设置预算金额。您可以手动输入预算金额，还可以设置起始预算和增长比例，系统将自动填充预算金额。 ○ 季度预算金额：当**预算周期**为**每季度**时，可为开始时间后的4个季度分别设置预算金额。您可以手动输入预算金额，还可以设置起始预算和增长比例，系统将自动填充预算金额。

7. 设置预算告警提醒阈值，单击**下一步**。

   例如您设置**预算金额**为1000元，**提醒阈值**为80%预算金额，则表示当您的云产品使用费用达到800元时，会产生告警。

查看费用预算和实际消费的对比情况

1. 在**预算管理**页面，单击目标预算名称。
2. 查看费用预算和实际消费的对比情况。



设置产品预算告警

创建预算管理任务后，成本管家自动为您开启告警实例，您可以在**告警**页面查看该实例。如果您希望快速完成告警设置，接收告警通知，只需创建用户并在SLS成本管家内置用户组中添加用户即可，其他告警资源保持默认配置。如果您要自定义告警资源，请参见设置告警。

1. 在成本管家的左侧导航栏中，单击**告警**。
2. 在**告警管理 > 用户管理**页签中，单击**创建**或**批量添加**。
3. 在**添加用户**对话框中，配置相关参数，单击**确定**。

   相关配置参数说明，请参见创建用户。
4. 在SLS成本管家内置用户组中添加用户。

i. 在**告警管理 > 用户组管理**页签中，单击**SLS成本管家内置用户组**对应的**修改**。

ii. 在**修改用户组**对话框中，添加您已创建的用户，单击**确定**。

3.5. 设置告警

当您开启告警实例后，成本管家会根据对应的告警规则产生告警并按照行动策略进行告警通知。本文以自定义创建告警资源为例，介绍如何设置告警。

背景信息

成本管家支持产品账单告警和产品预算告警。产品账单告警用于监控单个阿里云产品的日账单和月账单，产品预算告警从账号、财务单元、实例标签这三个维度监控您的阿里云产品使用成本。

如果您要监控单产品账单，您可以参见本文自定义创建告警资源，然后开启对应的告警规则实例即可。如果您要从账号、财务单元、实例标签维度监控阿里云产品使用成本，您需要先创建预算管理任务。具体操作，请参见创建预算管理任务。

告警资源配置流程

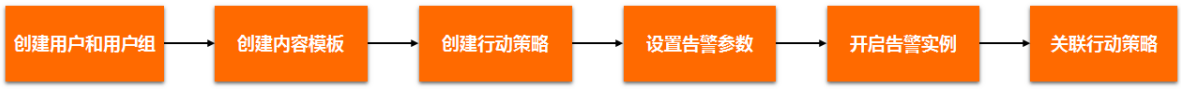
成本管家中已预设产品账单告警规则、SLS成本管家内置行动策略（sls.app.cost.builtin）、SLS成本管家内置用户组（sls.app.cost.builtin）和SLS成本管家内置内容模板（sls.app.cost.builtin.cn）等告警资源。您可以直接使用预设的告警资源，也可以自定义告警资源，具体配置流程如下：

- 使用预设的告警资源

  如果您希望快速完成告警设置，接收告警通知，只需完成如下配置。完成配置后，成本管家根据对应的告警规则产生告警并使用语音方式给您创建的用户发送告警通知。

- 自定义告警资源

  如果您希望根据实际场景自定义告警资源，您可以根据如下流程完成配置。完成配置后，成本管家根据对应的告警规则产生告警并根据您配置的告警渠道（语音、短信、邮件、钉钉WebHook、WebHook-自定义和通知中心）给您创建的用户发送告警通知。



步骤1：创建用户和用户组

当您选择短信、邮件或语音进行告警通知时，需要配置用户和用户组。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**成本管家**。

   您也可以在阿里云控制台首页的**成本管家.对账看板**区域，单击**查看更多报表**，进入成本管家。具体操作，请参见账单看板。
3. 在左侧导航栏，单击告警。

4. 创建用户。

   在**告警管理 > 用户管理**页签中，单击**添加**或**批量添加**，创建用户。相关配置参数说明，请参见创建用户。

5. 创建用户组。

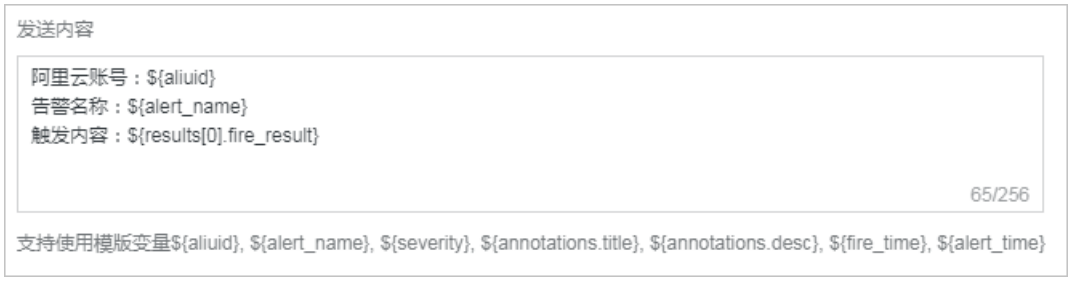
   在**告警管理 > 用户组管理**页签中，单击**创建**或**批量添加**，创建用户组。相关配置参数说明，请参见创建用户组。

步骤2：创建内容模板

成本管家中已预设SLS成本管家内置内容模板，当您需要自定义内容模板时，您可以新增内容模板或更新SLS成本管家内置内容模板。操作步骤如下：

1. 在**告警管理 > 内容模板**页签中，单击**添加**。
2. 在**添加内容模版**对话框中，配置**ID**、**名称**和**发送内容**，单击**确认**。

   发送内容支持使用模版变量，变量说明请参见创建内容模板。



步骤3：创建行动策略

成本管家中已预设SLS成本管家内置行动策略，当您需要自定义行动策略时，您可以新增行动策略或更新SLS成本管家内置行动策略。具体操作如下：

1. 在**告警管理 > 行动策略**页签中，单击**添加**。
2. 在**添加行动策略**对话框中，配置如下参数，单击**确认**。

参数	描述
ID	行动策略的唯一标识。
名称	行动策略的名称。
渠道	通知渠道包括短信、语音、邮件、钉钉（WebHook）、WebHook-自定义和通知中心，具体说明如下： ○ 当您使用短信、语音或邮件时，需设置接收人（用户或用户组）和内容模板。 ○ 当您使用钉钉（WebHook）时，需设置请求地址、提醒方式和内容模板。如何获取请求地址，请参见WebHook-钉钉机器人。 ○ 当您使用WebHook-自定义时，需设置请求地址、提醒方式和内容模板。如何获取请求地址，请参见WebHook-自定义。 ○ 当您使用通知中心时，需设置内容模板。

步骤4：设置告警参数

成本管家中已预设所有告警规则的默认值。当您需要自定义告警参数时，请参见如下步骤。此处以云数据库Redis-当日累计账单告警为例。

1. 在**告警规则**页签中，单击目标告警规则**云数据库Redis-当日累计账单**对应的图标。
2. 在**参数设置**对话框中，设置**严重度**和**当日累计账单**，单击**保存**。

   例如设置**严重度**为**高（High-8）**、**当日累计账单**为**10000**，表示当日累计账单超过10000时，产生高级别的告警。

步骤5：开启告警实例

在**告警规则**页签中，单击目标告警规则对应的图标。

> **说明** 开启告警实例后，系统将提示您前往**行动策略**页签进行配置。如果您已有可用的行动策略，可单击**不再提醒**。

步骤6：关联行动策略

1. 在**告警规则**页签中，选中目标告警规则，单击**配置行动策略**。
2. 在**参数设置**对话框中，选择您要关联的行动策略，单击**保存**。

3.6. 子账号授权

本文档为您介绍子账号使用成本管家所需的权限。

为子账号授权后，可以通过子账号来使用成本管家功能，详情请参见授权RAM用户。

权限策略内容如下。关于每个动作具体的说明请参见动作列表。

```
{
 "Version": "1",
 "Statement": [
  {
   "Action": "log:CreateLogStore",
   "Resource": "acs:log:*:*:project/bill-analysis-*/logstore/*",
   "Effect": "Allow"
  },
  {
   "Action": "log:CreateIndex",
   "Resource": "acs:log:*:*:project/bill-analysis-*/logstore/aliyun_bill",
   "Effect": "Allow"
  },
  {
   "Action": "log:UpdateIndex",
   "Resource": "acs:log:*:*:project/bill-analysis-*/logstore/aliyun_bill",
   "Effect": "Allow"
  },
  {
   "Action": "log:CreateDashboard",
   "Resource": "acs:log:*:*:project/bill-analysis-*/dashboard/*",
   "Effect": "Allow"
  },
  {
```

```
  {
    "Action": "log:UpdateDashboard",
    "Resource": "acs:log:*:*:project/bill-analysis-*/dashboard/*",
    "Effect": "Allow"
  },
 {
    "Action": "log:CreateSavedSearch",
    "Resource": "acs:log:*:*:project/bill-analysis-*/savedsearch/*",
    "Effect": "Allow"
  },
 {
    "Action": "log:UpdateSavedSearch",
    "Resource": "acs:log:*:*:project/bill-analysis-*/savedsearch/*",
    "Effect": "Allow"
  },
{
    "Action": "log:CreateJob",
    "Resource": "acs:log:*:*:project/bill-analysis-*/job/*",
    "Effect": "Allow"
  },
 {
    "Action": "log:UpdateJob",
    "Resource": "acs:log:*:*:project/bill-analysis-*/job/*",
    "Effect": "Allow"
  },
 {
    "Action": "log:CreateApp",
    "Resource": "acs:log:*:*:app/bill",
    "Effect": "Allow"
  },
{
    "Action": "log:UpdateApp",
    "Resource": "acs:log:*:*:app/bill",
    "Effect": "Allow"
  },
{
    "Action": "log:GetApp",
    "Resource": "acs:log:*:*:app/bill",
    "Effect": "Allow"
  },
{
    "Action": "log:DeleteApp",
    "Resource": "acs:log:*:*:app/bill",
    "Effect": "Allow"
  }
 ]
}
```

4.新冠病毒疫情分析

4.1. 简介

本文主要介绍新冠病毒疫情分析应用及其相关亮点。

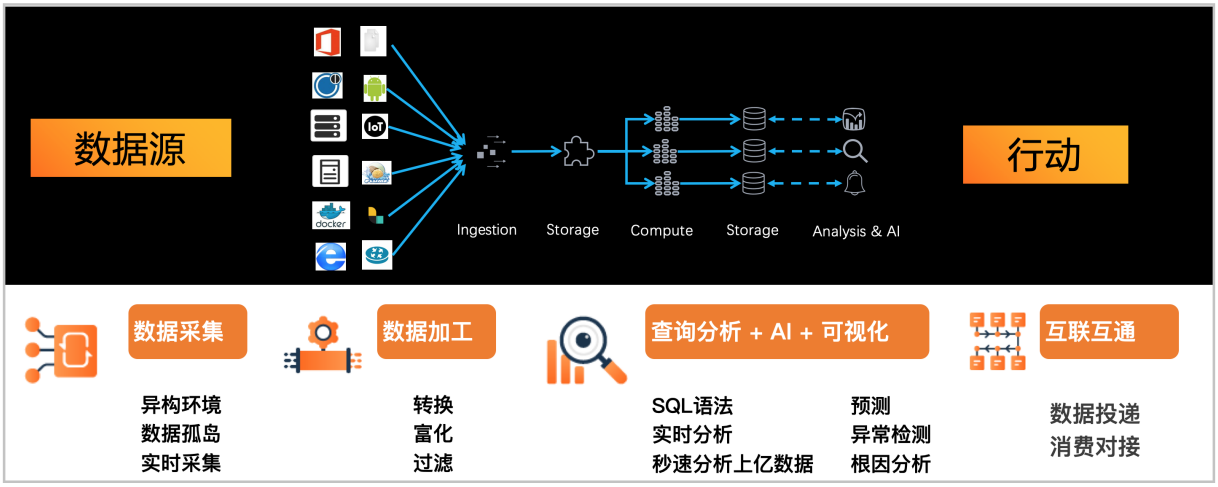
简介

新冠病毒疫情分析应用是基于阿里云日志服务中台提供的一站式的数据处理可视化分析系统。借助它，可以在全球范围内了解国家/地区、省份/州的疫情动态。目前该能力全面开放给政府、社区、第三方平台和开发者进行广泛应用，应用详情请参见详细说明。

关于日志服务

阿里云日志服务（Log Service）是针对日志类数据的一站式服务，无需开发就能快捷完成海量日志数据的采集、消费、投递以及查询分析等功能，提升运维、运营效率。日志服务主要包括实时采集与消费、数据投递、查询与实时分析等功能，适用于从实时监控到数据仓库的各种开发、运维、运营与安全场景。

作为日志分析中台，日志服务提供了一站式的数据采集、加工、查询分析、AI计算、可视化，并支持互联互通。

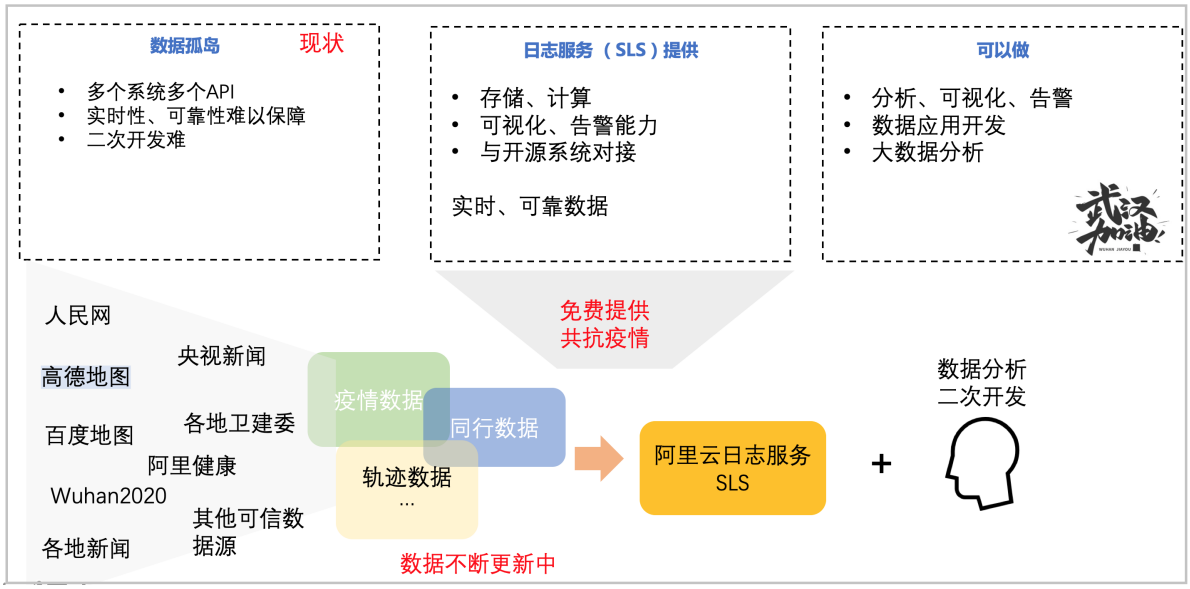


亮点

- 多份数据源、实时同步并形成可视化平台

  覆盖全球、省份、市区疫情信息，患者行程轨迹，路径信息，新闻公告等。

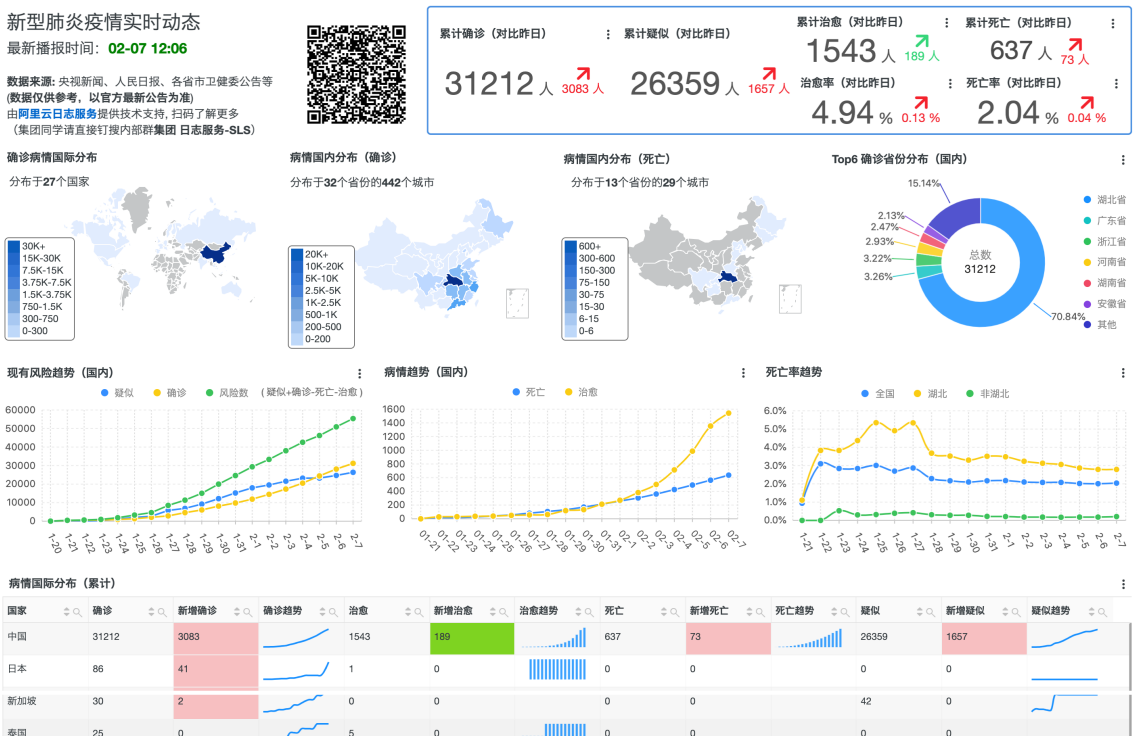
  > **注意** 图中各种数据源表示日志服务支持客户自行接入其他合法合规的多方面数据，目前App提供的数据来源于央视新闻、人民日报、各省市卫健委公告（实际信息以官方为准）。

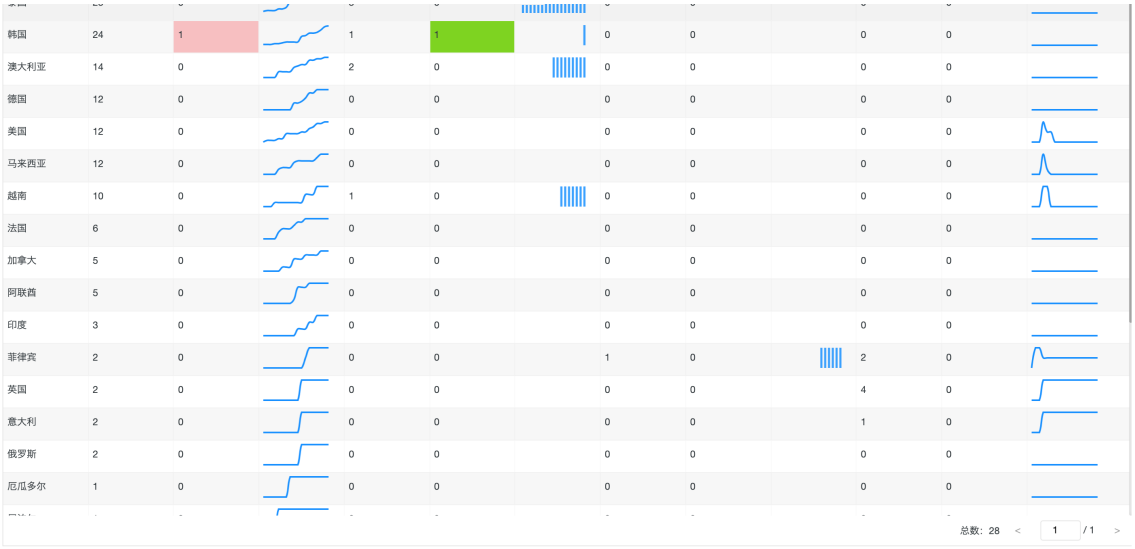


- 内置多份数据大盘并支持自定义

  提供全球态势、省市态势、确诊者踪迹态势、相关公告新闻等。支持交互式查询分析、自定义报表、深钻与告警等。

  - 全球态势





韩国	24	1		1	1		0	0	0	0
澳大利亚	14	0		2	0		0	0	0	0
德国	12	0		0	0		0	0	0	0
美国	12	0		0	0		0	0	0	0
马来西亚	12	0		0	0		0	0	0	0
越南	10	0		1	0		0	0	0	0
法国	6	0		0	0		0	0	0	0
加拿大	5	0		0	0		0	0	0	0
阿联酋	5	0		0	0		0	0	0	0
印度	3	0		0	0		0	0	0	0
菲律宾	2	0		0	0		1	0	2	0
英国	2	0		0	0		0	0	4	0
意大利	2	0		0	0		0	0	1	0
俄罗斯	2	0		0	0		0	0	0	0
厄瓜多尔	1	0		0	0		0	0	0	0

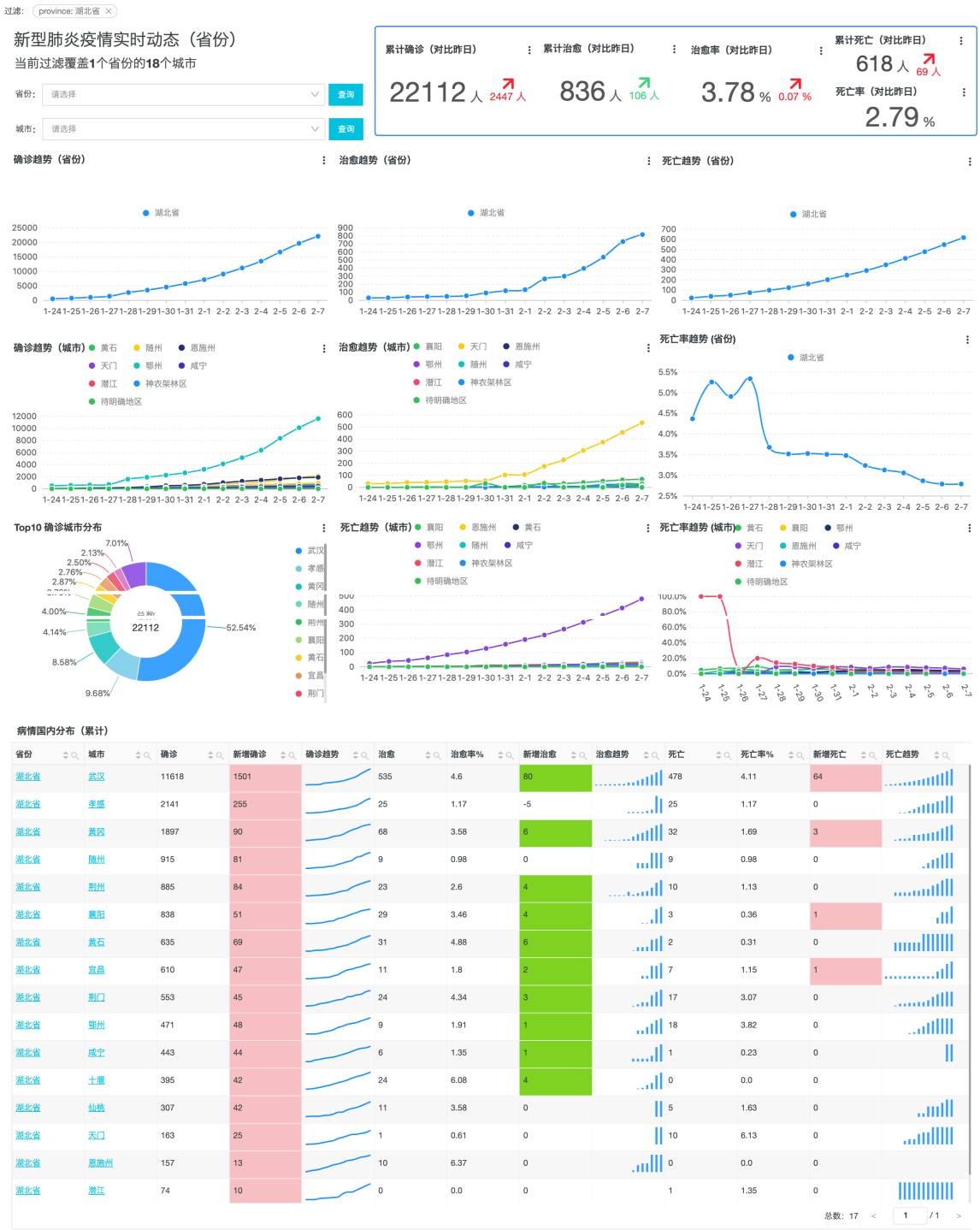
总数：28 < 1 / 1 >

病情国内分布（累计）

省份	确诊	新增确诊	确诊趋势	治愈	治愈率%	新增治愈	治愈趋势	死亡	死亡率%	新增死亡	死亡趋势
湖北省	22112	2447		818	3.7	106		618	2.79	69	
浙江省	1006	52		98	9.74	4		0	0.0	0	
河南省	914	63		70	7.66	14		3	0.33	1	
湖南省	772	61		91	11.79	10		0	0.0	0	
安徽省	665	74		34	5.11	2		0	0.0	0	
江西省	661	61		45	6.81	8		0	0.0	0	
重庆市	411	11		24	5.84	9		2	0.49	0	
江苏省	408	35		38	9.31	4		0	0.0	0	
山东省	379	32		31	8.18	4		0	0.0	0	
四川省	344	23		37	10.76	6		1	0.29	0	
北京市	297	23		33	11.11	2		1	0.34	0	
黑龙江省	277	50		8	2.89	0		3	1.08	0	
上海市	269	12		25	9.29	0		1	0.37	0	
陕西省	184	11		11	5.98	3		0	0.0	0	
广西壮族自治区	172	4		17	9.88	3		0	0.0	0	
河北省	171	14		16	9.36	4		1	0.58	0	
云南省	135	2		7	5.19	0		0	0.0	0	
海南省	114	8		8	7.02	0		2	1.75	1	
山西省	96	6		12	12.5	4		0	0.0	0	
辽宁省	94	3		5	5.32	0		0	0.0	0	
天津市	79	1		2	2.53	0		1	1.27	0	
贵州省	77	6		6	7.79	0		1	1.3	0	
甘肃省	67	5		9	13.43	3		0	0.0	0	
吉林省	65	6		4	6.15	1		1	1.54	1	
内蒙古自治区	50	4		4	8.0	0		0	0.0	0	
宁夏回族自治区	43	3		1	2.33	0		0	0.0	0	
新疆维吾尔自治区	39	3		0	0.0	0		0	0.0	0	
香港	24	3		0	0.0	0		1	4.17	0	
青海省	18	0		3	16.67	0		0	0.0	0	
台湾	16	3		1	6.25	0		0	0.0	0	
澳门	10	0		1	10.0	1		0	0.0	0	
西藏自治区	1	0		0	0.0	0		0	0.0	0	

总数：34 < 1 / 1 >

- 省市态势

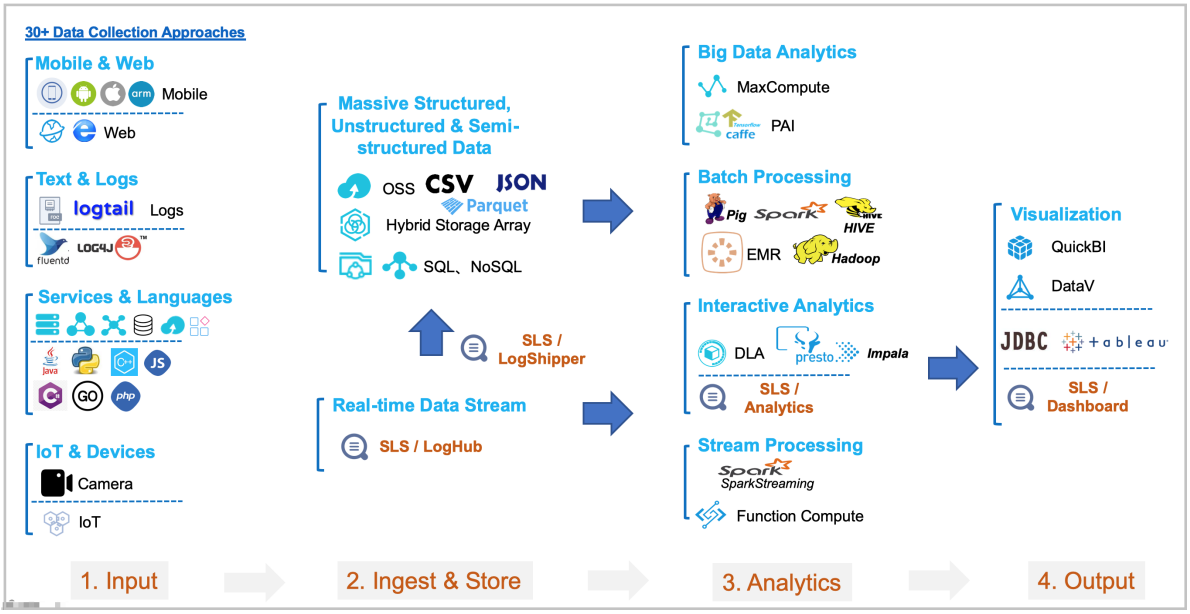

过滤：
province: 湖北省
新型肺炎疫情实时动态（省份）
当前过滤覆盖1个省份的18个城市
省份：
请选择
查询
城市：
请选择
查询
累计确诊（对比昨日）
22112 人 2447 人
累计治愈（对比昨日）
836 人 106 人
治愈率（对比昨日）
3.78 % 0.07 %
累计死亡（对比昨日）
618 人 69 人
死亡率（对比昨日）
2.79 %
确诊趋势（省份）
治愈趋势（省份）
死亡趋势（省份）
湖北省
确诊趋势（城市）
治愈趋势（城市）
死亡率趋势 (省份)
死亡率趋势 (城市)
黄石 随州 恩施州 天门 鄂州 咸宁 潜江 神农架林区 待明确地区
襄阳 天门 恩施州 鄂州 随州 咸宁 潜江 神农架林区 待明确地区
Top10 确诊城市分布
22112
52.54%
9.68%
8.58%
4.14%
4.00%
2.87%
2.76%
2.50%
2.13%
7.01%
武汉 孝感 黄冈 随州 荆州 襄阳 黄石 宜昌 荆门
死亡趋势（城市）
死亡率趋势 (城市)
病情国内分布（累计）
省份 城市 确诊 新增确诊 确诊趋势 治愈 治愈率% 新增治愈 治愈趋势 死亡 死亡率% 新增死亡 死亡趋势
湖北省 武汉 11618 1501 535 4.6 80 478 4.11 64
湖北省 孝感 2141 255 25 1.17 -5 25 1.17 0
湖北省 黄冈 1897 90 68 3.58 6 32 1.69 3
湖北省 随州 915 81 9 0.98 0 9 0.98 0
湖北省 荆州 885 84 23 2.6 4 10 1.13 0
湖北省 襄阳 838 51 29 3.46 4 3 0.36 1
湖北省 黄石 635 69 31 4.88 6 2 0.31 0
湖北省 宜昌 610 47 11 1.8 2 7 1.15 1
湖北省 荆门 553 45 24 4.34 3 17 3.07 0
湖北省 鄂州 471 48 9 1.91 1 18 3.82 0
湖北省 咸宁 443 44 6 1.35 1 1 0.23 0
湖北省 十堰 395 42 24 6.08 4 0 0.0 0
湖北省 仙桃 307 42 11 3.58 0 5 1.63 0
湖北省 天门 163 25 1 0.61 0 10 6.13 0
湖北省 恩施州 157 13 10 6.37 0 0 0.0 0
湖北省 潜江 74 10 0 0.0 0 1 1.35 0
总数：17
1 / 1

- 感染者行程信息

- 新闻公告信息

- 数据平台开放，互联互通

  日志服务是开放的，可以和大量其他环境的系统、三方应用或开源进行对接。提供易扩展的数据分析、存储、可视化平台能力，如DataV、Blink、OSS、流计算、Grafana、SOC等。



其他参考

- 新冠病毒疫情分析应用的资源说明
- 新冠病毒疫情分析应用的限制说明
- 新冠病毒疫情分析应用的日志格式说明
- 新冠病毒疫情分析应用的使用说明

4.2. 详细说明

本文介绍新冠病毒疫情分析应用的详细信息，包括应用说明、资源说明、限制说明、数据说明和使用说明等。

应用说明

- 首次使用该功能需要完成初始化配置（2分钟左右）。
- 每天自动更新同步数据，无需手动同步。
- 数据来源：央视新闻、人民日报、各省市卫健委公告等。
- 数据仅供参考，以官方最新公告为准。
- 数据存储和分析功能不收费。
- 如果本应用中相关免费资源长期无活跃操作，本服务保留回收的权利。您可以重启应用再次创建应用。
- 技术支持。

由阿里云日志服务提供技术支持，扫码了解更多。

资产说明

应用会创建以下日志服务项目资源，不会产生费用。

- 日志项目：ncp-{阿里云主账号UID}-cn-chengdu
- 日志库：ncp
- 仪表盘：ncp、ncp_detail、ncp_travel、ncp_news等。

限制说明

- 专属日志库，您无法修改删除Logstore、索引或写入数据。其他操作与一般日志库没有差别。
- 您可以在该项目中创建自己的Logstore并写入自己的数据，但这部分Logstore产生的费用不在免费范围内。
- 专属仪表盘，不推荐修改，可能在后续应用升级中自动覆盖任何改动。您可以在日志服务的项目中复制仪表盘再做修改。更多信息，请参见如何复制仪表盘。

仪表盘说明

提供如下多张内置仪表盘。仪表盘是基于日志库中的数据构建的，您也可以基于数据构建新的仪表盘。

仪表盘	ID	描述
全局态势	ncp	提供全球、中国各省的疫情指标、趋势与列表汇总。
省市态势	ncp_detail	提供中国各省市的疫情指标、趋势与列表汇总。 该仪表盘需要选择一个或多个省市查看（仪表盘全局态势中选择省份时会自动跳转到该页面并选择对应省份）。
患者行程	ncp_travel	提供公告中确诊患者的相关行程的汇总与查询。

仪表盘	ID	描述
公告新闻	ncp_news	提供疫情相关新闻公告的汇总与查询。

数据说明

- 数据版本与使用说明

  各种疫情相关数据均放在一个日志库ncp中，每天有多次版本自动同步到本地导入日志库中，通过字段version标示更新时间，例如：v2020-01-26T12:30:00。

  每个版本的数据都包含了全量数据，因此只需要使用最新版本的数据进行查询、分析统计即可。

  一般情况，可以在查询统计时指定一个版本，如下所示。

  ```
  version: "v2020-01-26T12:30:00" and type: province_detail | select .... from log
  ```

  但推荐将以上查询统计语句改成如下SQL模式，这样可以在版本更新后自动使用最新版本。

  ```
  type: province_detail | select .... from log l right join (select max(version) as version from log) r on  l.version =  r.version
  ```

  > 说明
  > - |前的是查询语句，一般用type过滤特定类型的日志，查询语法详情请参见查询语法。
  > - |后的是标准SQL92语法，其中from log表示从当前日志库中查询，也支持多库join等，并提供额外扩展，如IP地理库、外表OSS/MySQL协同查询功能。更多信息，请参见统计语法。
  > - 每天自动更新同步数据，因此查询统计的时间选择器，选择相对1天即可。

- 概览

  各种疫情相关数据均放在一个日志库ncp中，通过字段type作为类型区分。

日志类型（type值）	说明	更新频率
country_stat	国际疫情信息	一天多次
province_stat	中国各省疫情信息	一天多次
city_stat	中国各城市疫情信息	一天多次
travel_detail	确认患者行程信息	一天至少1次
news	新闻公告	一天至少1次

- 国际疫情详情

  > 说明 其中_hist会在表格的迷你图中使用，而_trend类数据会在各个趋势中使用。

字段名	说明	样例
type	数据类型	固定为country_stat

字段名	说明	样例
version	数据版本	v2020-01-26T12:30:00
news_time	来源新闻发布时间	2020-01-26 18:23
country	国家名称	中国， 泰国
quezhen	最新确诊病例累计数据	1058
quezhen_hist	确诊病例累计数据（从2020.01.21到当前的历史数据数组）	[270, 444, 444, 549, 729, 1058]
quezhen_trend	确诊病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
zhiyu	最新治愈病例累计数据	42
zhiyu_hist	治愈病例累计数据（从2020.01.21到当前的历史数据数组）	[0, 28, 28, 31, 32, 42]
zhiyu_trend	治愈病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
dead	最新死亡病例累计数据	52
dead_hist	死亡病例累计数据（从2020.01.21到当前的历史数据数组）	[3, 17, 17, 24, 39, 52]
dead_trend	死亡病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
yisi	最新疑似病例现有数据	127
yisi_hist	疑似病例现有数据（从2020.01.21到当前的历史数据数组）	[11, 0, 41, 0, 56, 127]
yisi_trend	疑似病例现有数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 7}

- 中国各省疫情信息

> **说明**
> - 直辖市（例如：北京市）也在此类型中提供。
> - 对于公告中未明确说明具体所属省份的数据，将放入到一个叫做未明确省份的省份中。
> - 其中_hist会在表格的迷你图中使用，而_trend类数据会在各个趋势中使用。

字段名	说明	样例
type	数据类型	固定为province_stat
version	数据版本	v2020-01-26T12:30:00
news_time	来源新闻发布时间	2020-01-26 18:23
country	国家名称	中国
province	省或直辖市名称	上海市， 浙江省
quezhen	最新确诊病例累计数据	1058
quezhen_hist	确诊病例累计数据（从2020.01.21到当前的历史数据数组）	[270, 444, 444, 549, 729, 1058]
quezhen_trend	确诊病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
zhiyu	最新治愈病例累计数据	42
zhiyu_hist	治愈病例累计数据（从2020.01.21到当前的历史数据数组）	[0, 28, 28, 31, 32, 42]
zhiyu_trend	治愈病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
dead	最新死亡病例累计数据	52
dead_hist	死亡病例累计数据（从2020.01.21到当前的历史数据数组）	[3, 17, 17, 24, 39, 52]
dead_trend	死亡病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
yisi	最新疑似病例现有数据	127
yisi_hist	疑似病例现有数据（从2020.01.21到当前的历史数据数组）	[11, 0, 41, 0, 56, 127]
yisi_trend	疑似病例现有数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 7}

- 中国各城市疫情信息

> 说明
> - 直辖市中的区（例如：海淀区）也在此类型中提供。
> - 某些公告中出现的分类如外来人员也会放入此类信息中。
> - 对于公告中未明确说明具体所属城市的数据，将放入到一个叫做未明确地区的省份中。
> - 其中_hist会在表格的迷你图中使用，而_trend类数据会在各个趋势中使用。

字段名	说明	样例
type	数据类型	固定为city_stat
version	数据版本	v2020-01-26T12:30:00
news_time	来源新闻发布时间	2020-01-26 18:23
country	国家名称	中国
province	省名称	上海市，浙江省
city	城市名称	杭州， 浦东新区
quezhen	最新确诊病例累计数据	1058
quezhen_hist	确诊病例累计数据（从2020.01.21到当前的历史数据数组）	[270, 444, 444, 549, 729, 1058]
quezhen_trend	确诊病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
zhiyu	最新治愈病例累计数据	42
zhiyu_hist	治愈病例累计数据（从2020.01.21到当前的历史数据数组）	[0, 28, 28, 31, 32, 42]
zhiyu_trend	治愈病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
dead	最新死亡病例累计数据	52
dead_hist	死亡病例累计数据（从2020.01.21到当前的历史数据数组）	[3, 17, 17, 24, 39, 52]
dead_trend	死亡病例累计数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 3}
yisi	最新疑似病例现有数据	127
yisi_hist	疑似病例现有数据（从2020.01.21到当前的历史数据数组）	[11, 0, 41, 0, 56, 127]

字段名	说明	样例
yisi_trend	疑似病例现有数据（从2020.01.21到当前的历史趋势数据字典）	{"01-21": 1, "01-22": 1, "01-23": 1, "01-24": 2, "01-25": 2, "01-26": 7}

- 确认患者行程信息

字段	说明	样例
type	数据类型	固定为travel_detail
version	数据版本	v2020-02-11T20:30:00
start_time	出行开始时间	2020/02/05 13:05:00
end_time	出行结束时间	2020/02/05 15:00:00
travel_date	出行日期	2020-02-05
travel_type	出行类型	火车
travel_no	车次、车牌、航班、场所名	D6315
travel_sub_no	车厢	8号车厢
travel_detail	出行描述	座位：9A
start_pos	出发站	古田北
end_pos	到达站	泉州
author	线索来源	福州卫生报
url	新闻页链接	https://weibo.com/abc.html

- 新闻公告信息

字段名	说明	样例
type	数据类型	固定为news
version	数据版本	v2020-01-26T12:30:00
news_date	新闻发布日期	2020-01-26
news_time	新闻发布时间	2020-01-26 18:23
author	新闻发布者	武汉市卫健委
title	新闻标题	-
summary	新闻摘要	-

字段名	说明	样例
source	新闻来源网站	www.weibo.com
url	新闻或者信息来源URL	https://weibo.com/abc.html

使用说明

1. 登录阿里云日志服务控制台。
2. 在**日志应用**区域，单击**新冠病毒疫情分析**。

3. 根据页面提示，完成初始化配置，开始使用新冠病毒疫情分析应用。

   只在首次使用时，需进行初始化配置。

常见问题

- 如何删除所属项目？

  如果您需删除所属项目，可直接打开Cloud Shell执行如下命令行删除项目。

  ```
  aliyunlog log delete_project --project_name=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --region-endpoint=cn-chengdu.log.aliyuncs.com
  ```

  > **注意** 如果项目中创建了自己的日志库，也会一并被删除，请谨慎操作。

- 如何从现有仪表盘复制新的仪表盘？

  i. 在阿里云控制台右上角，打开阿里云Cloud Shell。

  ii. 复制仪表盘配置到本地。

```
aliyunlog log get_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --entity=ncp --regi
on-endpoint=cn-chengdu.log.aliyuncs.com > ncp.json
aliyunlog log get_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --entity=ncp_detai
l --region-endpoint=cn-chengdu.log.aliyuncs.com > ncp_detail.json
aliyunlog log get_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --entity=ncp_trave
l --region-endpoint=cn-chengdu.log.aliyuncs.com > ncp_travel.json
aliyunlog log get_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --entity=ncp_news
--region-endpoint=cn-chengdu.log.aliyuncs.com > ncp_news.json
sed -i "s/\"dashboardName\": \"/\"dashboardName\": \"v2/g" ncp.json
sed -i "s/\"description\": \"\", \"displayName\": \"/\"description\": \"\", \"displayName\": \"v2/g" ncp.js
on
sed -i "s/\"dashboardName\": \"/\"dashboardName\": \"v2/g" ncp_detail.json
sed -i "s/\"description\": \"\", \"displayName\": \"/\"description\": \"\", \"displayName\": \"v2/g" ncp_
detail.json
sed -i "s/\"dashboardName\": \"/\"dashboardName\": \"v2/g" ncp_travel.json
sed -i "s/\"description\": \"\", \"displayName\": \"/\"description\": \"\", \"displayName\": \"v2/g" ncp_t
ravel.json
sed -i "s/\"dashboardName\": \"/\"dashboardName\": \"v2/g" ncp_news.json
sed -i "s/\"description\": \"\", \"displayName\": \"/\"description\": \"\", \"displayName\": \"v2/g" ncp_
news.json
aliyunlog log create_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --detail=file://./
ncp.json --region-endpoint=cn-chengdu.log.aliyuncs.com
aliyunlog log create_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --detail=file://./
ncp_detail.json --region-endpoint=cn-chengdu.log.aliyuncs.com
aliyunlog log create_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --detail=file://./
ncp_travel.json --region-endpoint=cn-chengdu.log.aliyuncs.com
aliyunlog log create_dashboard --project=ncp-$ALICLOUD_ACCOUNT_ID-cn-chengdu --detail=file://./
ncp_news.json --region-endpoint=cn-chengdu.log.aliyuncs.com
```

iii. 查看创建的仪表盘。

在**新冠病毒疫情分析**应用的**设置**页签中单击**跳转到Project控制台**，单击**仪表盘**，查看新建的仪表盘。

其他参考

- 本应用相关直播视频
- 日志服务文档
- 构建仪表盘
- 其他日志服务相关视频

5.K8S事件中心

5.1. 创建并使用Kubernetes事件中心

本文介绍如何创建Kubernetes事件中心及相关操作，包括查看事件总览、查询事件详情、查看Pod生命周期、配置告警和自定义查询等操作。

背景信息

Kubernetes事件中心记录了集群的状态变更，包括创建Pod、运行Pod、删除Pod、组件异常等。
Kubernetes事件中心实时汇聚Kubernetes中的所有事件并提供存储、查询、分析、可视化、告警等能力。

免费策略

Kubernetes事件中心关联的Logstore在90天内免费（每天允许免费写入256M数据，相当于25万条事件。默认一个Kubernetes线上集群每天产生的事件在1000条左右）。事件存储时间默认为90天，因此如果您不调整事件保存时间，可一直免费使用Kubernetes事件中心。例如：

- 不调整存储时间（默认90天），集群每天产生1000条事件，则事件中心永久免费。
- 调整存储时间为105天，集群每天产生1000条事件，则超过90天后，事件中心每天收取的费用约0.1元，费用详情请参见按量付费。

步骤一：创建事件中心

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**K8s事件中心**。
3. 在**事件中心管理**页面，单击**添加**。
4. 在**添加事件中心**页面，配置相关参数。
   - 选择**已有Project**，可从**Project**下拉框中选择已创建的Project。
   - 选择**从容器服务选择K8s集群**，可从**K8s集群**下拉框中选择已创建的K8s集群。通过此方式创建事件中心，默认创建一个名为k8s-log-{cluster-id}的Project。
5. 单击**下一步**，完成创建。

> **说明** 创建事件中心后，默认在您选择的日志服务Project中创建一个名为k8s-event的Logstore，并创建相关联的报表和告警等。

步骤二：部署Eventer和NodeProblemDetector

您需要在Kubernetes集群中配置事件采集和node-problem-detector后才能正常使用K8s事件中心。

- 阿里云Kubernetes配置方式

  阿里云Kubernetes应用市场中的ack-node-problem-detector已集成node-problem-detector和事件采集功能，您只需要部署该组件即可，该组件详细部署请参见事件监控。

  i. 登录容器服务控制台。
  ii. 在左侧导航栏中，选择**市场 > 应用目录**。
  iii. 在**阿里云应用**页签下，单击**ack-node-problem-detector**。
  iv. 在**参数**页签下，修改eventer节点中的相关信息。

- enabled：将**eventer > sinks > sls**下的enabled设置为true。
- topic：可选，设置为您的集群名称，只支持英文字母a-z、下划线（_）、连接号（-）。
- project：设置为您创建事件中心时的Project名称。
- logstore：只能设置为k8s-event。

```
sinks:
  sls:
    enabled: true
    # If you want the monitoring results to be notified by sls, set enabled to true.
    topic: "my-cluster"
    project:  "{sls-project-name}"
    # You can view the project information by logging in to the
    # SLS console. Please fill in the name of the project here.
    # eg: your project name is k8s-log-cc18a5f3443dhdss22654da,
    # then you can fill k8s-log-cc18a5f3443dhdss22654da to project label.
    logstore: "k8s-event"
    # You can view the project information by logging in to the
    # SLS console. Please fill the logstore address in here.
```

v. 单击**创建**，完成部署。

- 自建Kubernetes配置方式
  i. 配置事件采集。更多信息，请参见采集Kubernetes事件。
  ii. 配置node-problem-detector，详情请参见Github。

步骤三：使用事件中心

创建K8s事件中心并部署Eventer和NodeProblemDetector后，即可使用K8s事件中心，包括查看事件总览、查询事件详情、查看Pod生命周期、配置告警和自定义查询等操作。

在**K8s事件中心**页面，找到目标事件中心实例，单击 图标，可进行如下操作。

操作	说明
查看事件总览	单击**事件总览**，查看核心事件的汇总统计信息。例如：总体错误数以及和昨天/上周的对比、告警项统计、重要事件趋势、Pod OOM详细信息等。 **说明** 目前Pod OOM信息不能精确到Pod，只能定位到事件发生的节点、进程名、进程号。您可以通过自定义查询查找Pod OOM发生时间点附近的Pod重启事件，以此定位到具体的Pod。
查询事件详情	单击**事件详情查询**，查看按照各种维度（事件等级、事件类型、事件目标、Host、Namespace、Name）过滤后的事件的统计信息以及详情。
查看Pod生命周期	单击**Pod生命周期**，以图形化方式展示Pod整个生命周期中的事件信息，还可通过事件等级筛选重要的Pod事件。
配置告警	单击**告警配置**，配置事件的告警，具体操作请参见表格下方的操作步骤。

操作	说明
自定义查询	单击**自定义查询**，自定义查询条件查询相关信息，查询条件请参见查询与分析语法规则。 事件中心的所有事件都保存在Logstore中，您可以使用Logstore中的所有功能，例如自定义查询、消费事件进行自定义处理、创建自定义报表、创建自定义告警等。 如果您要访问事件中心所在的Project，可通过以下两种方式获取Project名称。 • 通过**自定义查询**页面的URL定位到Project。URL规则为 `https://sls.console.aliyun.com/lognext/app/k8s-event/project/k8s-log-xxxx/logsearch/k8s-event` ，Project字段的后一个字段即为日志服务Project名称，例如k8s-log-xxxx。 • 在**集群管理**页签的事件中心列表中，查看目标事件中心对应的Project名称。
配置自定义告警	除了内置的告警外，事件中心还支持配置自定义告警。 在**自定义查询**页面，输入对应K8s事件的查询语句，单击**另存为告警**完成自定义告警配置。更多信息，请参见告警简介。 例如：创建一个FailedPreStopHook的告警，您可以在查询页面中输入 `* and FailedPreStopHook | SELECT "object-namespace", "object-name", "reason", "message"` ，单击**另存为告警**，配置参数后保存即可。 **说明** 如果您自定义配置的告警名称是前缀K8s，则该告警配置会在目标事件的**告警配置**页签的**全部告警事件**显示中，否则只显示在告警详情中。

配置告警具体操作如下所示。

1. 在**K8s事件中心**，找到目标事件中心实例，单击 图标。
2. 单击**告警配置**，进入**告警配置页面**。
3. 添加通知方式。
   i. 单击**添加通知方式**。
   ii. 在**添加通知方式**页面，配置相关参数。

参数	说明
通知方式名称	通知方式的名称。
告警间隔	两次告警通知之间的时间间隔，默认为5分钟。 **说明** 建议告警间隔最小设置为2分钟，防止收到过多的告警信息。
通知类型	包括短信、语音、邮件、钉钉机器人、WebHook自定义和通知中心，可选择一种或多种通知类型。更多信息，请参见通知方式。

   iii. 单击**确定**。
4. 开启告警通知
   i. 在**全部告警事件**区域，单击**修改**。

ii. 找到待开启的告警事件，单击开启图标，并选择合适的告警通知。

> **说明** 建议您先开启所有告警，若发现告警通知太多，可适当关闭告警或调整通知间隔。

iii. 单击**保存**。

删除事件中心

在**K8s事件中心 > 集群管理**页面中，找到目标事件中心实例，单击 图标，删除事件中心。

常见问题

- K8s事件中心无数据。

  部署好K8s事件中心后，新产生的事件会自动采集到K8s事件中心，您可以在**自定义查询**页面进行搜索（建议将右上角时间范围调整到1天）。如果无数据，一般有两个原因：

  - 部署K8s事件中心后，K8s集群还未产生事件。

    您可以通过 `kubectl get events --all-namespaces` 命令检查集群内是否有新事件产生。

  - 部署Eventer和NodeProblemDetectors时，参数填写错误。
    - 如果您使用的是阿里云Kubernetes集群，请在**容器服务控制台 > 应用 > 发布**中，找到对应的集群，单击**ack-node-problem-detector**后的**更新**，检查参数配置，详情配置请参见步骤二：部署Eventer和NodeProblemDetector。
    - 如果您使用的是自建Kubernetes集群，参数配置请参见采集Kubernetes事件。

- 如何查看事件对应容器的日志？
  - 如果您使用的是阿里云Kubernetes集群，请在**容器服务控制台 > 应用 > 容器组**中，找到目标集群，将**命名空间**选择为**kube-system**，在搜索框中输入eventer关键词找到目标容器，在其详情页面查看日志。
  - 如果您使用的是自建Kubernetes集群，请查看namespace为**kube-system**下文件名前缀为eventer-sls的Pod日志。

6.SLB日志中心

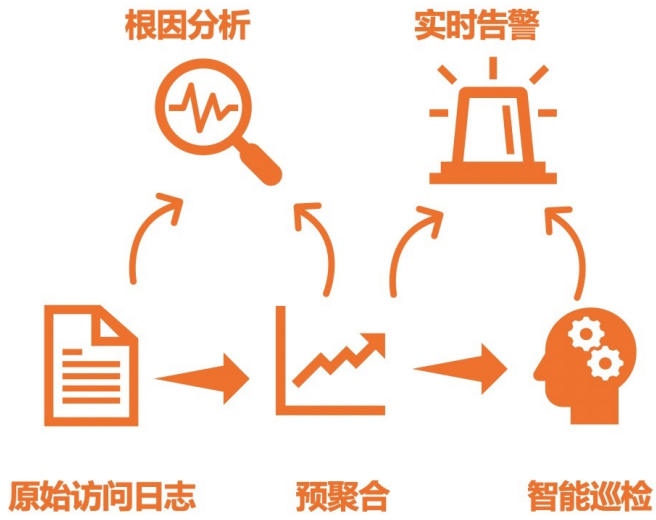
6.1. 使用前须知

阿里云日志服务和负载均衡（SLB）推出日志中心功能，日志中心提供负载均衡7层日志分析、秒级监控指标分析、实时告警等功能，并提供基于AIOps的自动异常巡检功能。您可以通过SLB日志中心了解客户端用户行为、客户端用户的地域分布、请求成功率、响应延迟等。本文介绍负载均衡7层日志中心相关的功能说明、功能优势、资产说明、费用说明、使用限制等信息。

功能说明

SLB日志中心基于实时访问日志进行自动聚合，并提供智能巡检、实时告警等功能，详细功能如下：

- 实时访问日志的存储、查询、分析。
- 基于原始访问日志实时提取各类指标信息，包括PV、请求成功率、平均延迟、P50/P99/P9999延迟、出入流量等。并支持多个维度组合，包括slbid、host、method、status。
- 提供丰富的可视化报表，包括监控中心、异常诊断等，并支持报表邮件、钉钉群订阅。
- 提供智能巡检功能，支持全局巡检和slbid粒度巡检，并支持在可视化报表中直接标注异常点。
- 自定义告警配置，告警通知直接对接消息中心、短信、邮件、语音（电话）、钉钉，并支持对接自定义WebHook。



功能优势

- 简单：一站式开通、中心化使用，无需关心日志收集、存储、计算、可视化等问题，将开发、运维人员从日志处理的繁琐耗时中解放出来，将更多的精力集中到业务开发和技术探索上去。
- 海量：访问日志与负载均衡实例请求PV成正比，数据规模很大，处理访问日志需要考虑性能和成本问题。日志中心可自定义配置预聚合功能，实时计算聚合指标，计算后的聚合结果可降低几个数量级，使查询速度大大提升。
- 实时：DevOps、监控、报警等场景要求日志数据的实时性。负载均衡结合日志服务强大的大数据计算能力，秒级分析处理实时产生的日志。
- 弹性：您可按负载均衡实例级别开通或关闭访问日志功能，可任意设置日志存储周期。Logstore容量可动态伸缩满足业务增长需求。
- 智能：基于达摩院智能AIOps算法，提供SLB指标自动巡检功能，有助于更快、更准确的发现并定位问题。

资产说明

所有资产都在您选择的Project下，Project内的资产如下：

- Logstore
  - 访问日志Logstore用于存储负载均衡7层访问日志，该Logstore为您自定义创建的Logstore。
  - 巡检结果Logstore用于存储巡检结果。开通日志中心功能后，自动生成该专属Logstore，其名称为*访问日志Logstore*-metrics-result。

> **说明**
> - 请勿删除负载均衡7层访问日志相关的Logstore，否则将无法正常采集日志到日志服务。
> - 请勿删除访问日志Logstore中的部分字段的索引，否则指标转换会失败。

- Metricstore

  监控指标Metricstore用于存储聚合后的指标信息。开通日志中心功能后，自动生成该专属Metricstore，其名称为*访问日志Logstore*-metrics。

> **说明** 监控指标Metricstore存储的是聚合后的指标，数据量相比原始访问日志大大降低，非常适用于长期存储。

- 聚合规则

规则名称	聚合时间粒度	聚合维度	生成指标名
total	10秒	total	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99 ◦ upstream_response_time_p9999 ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ request_time_p9999

规则名称	聚合时间粒度	聚合维度	生成指标名
slbid	10秒	slbid	◦ pv:slb ◦ body_bytes_sent_avg:slb ◦ body_bytes_sent_sum:slb ◦ request_length_avg:slb ◦ request_length_sum:slb ◦ upstream_response_time_avg:slb ◦ upstream_response_time_p50:slb ◦ upstream_response_time_p90:slb ◦ upstream_response_time_p99:slb ◦ upstream_response_time_p9999:slb ◦ request_time_avg:slb ◦ request_time_p50:slb ◦ request_time_p90:slb ◦ request_time_p99:slb ◦ request_time_p9999:slb
slbid_host_status	10秒	slbid+host+status	◦ pv:slbid:host:status ◦ body_bytes_sent_avg:slbid:host:status ◦ body_bytes_sent_sum:slbid:host:status ◦ request_length_avg:slbid:host:status ◦ request_length_sum:slbid:host:status ◦ upstream_response_time_avg:slbid:host:status ◦ upstream_response_time_p50:slbid:host:status ◦ upstream_response_time_p90:slbid:host:status ◦ upstream_response_time_p99:slbid:host:status ◦ upstream_response_time_p9999:slbid:host:status ◦ request_time_avg:slbid:host:status ◦ request_time_p50:slbid:host:status ◦ request_time_p90:slbid:host:status ◦ request_time_p99:slbid:host:status ◦ request_time_p9999:slbid:host:status

规则名称	聚合时间粒度	聚合维度	生成指标名
slbid+host+status+request_method+upstream_status+url	10秒	slbid+host+status+request_method+upstream_status+url	◦ pv:slbid:host:status:method:upstream_status ◦ body_bytes_sent_avg:slbid:host:status:method:upstream_status ◦ body_bytes_sent_sum:slbid:host:status:method:upstream_status ◦ request_length_avg:slbid:host:status:method:upstream_status ◦ request_length_sum:slbid:host:status:method:upstream_status ◦ upstream_response_time_avg:slbid:host:status:method:upstream_status ◦ upstream_response_time_p50:slbid:host:status:method:upstream_status ◦ upstream_response_time_p90:slbid:host:status:method:upstream_status ◦ upstream_response_time_p99:slbid:host:status:method:upstream_status ◦ upstream_response_time_p9999:slbid:host:status:method:upstream_status ◦ request_time_avg:slbid:host:status:method:upstream_status ◦ request_time_p50:slbid:host:status:method:upstream_status ◦ request_time_p90:slbid:host:status:method:upstream_status ◦ request_time_p99:slbid:host:status:method:upstream_status ◦ request_time_p9999:slbid:host:status:method:upstream_status

- 巡检规则

规则名称	开启状态	巡检算法	巡检指标

规则名称	开启状态	巡检算法	巡检指标
slb-patrol-total	默认开启	Time2Graph	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99 ◦ upstream_response_time_p9999 ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ request_time_p9999
slb-patrol-slb	默认开启	Time2Graph	◦ pv:slb ◦ body_bytes_sent_avg:slb ◦ body_bytes_sent_sum:slb ◦ request_length_avg:slb ◦ request_length_sum:slb ◦ upstream_response_time_avg:slb ◦ upstream_response_time_p50:slb ◦ upstream_response_time_p90:slb ◦ upstream_response_time_p99:slb ◦ upstream_response_time_p9999:slb ◦ request_time_avg:slb ◦ request_time_p50:slb ◦ request_time_p90:slb ◦ request_time_p99:slb ◦ request_time_p9999:slb

规则名称	开启状态	巡检算法	巡检指标
slb-patrol-slbid_host_status	默认关闭	Time2Graph	◦ pv:slbid:host:status ◦ body_bytes_sent_avg:slbid:host:status ◦ body_bytes_sent_sum:slbid:host:status ◦ request_length_avg:slbid:host:status ◦ request_length_sum:slbid:host:status ◦ upstream_response_time_avg:slbid:host:status ◦ upstream_response_time_p50:slbid:host:status ◦ upstream_response_time_p90:slbid:host:status ◦ upstream_response_time_p99:slbid:host:status ◦ upstream_response_time_p9999:slbid:host:status ◦ request_time_avg:slbid:host:status ◦ request_time_p50:slbid:host:status ◦ request_time_p90:slbid:host:status ◦ request_time_p99:slbid:host:status ◦ request_time_p9999:slbid:host:status

规则名称	开启状态	巡检算法	巡检指标
slb-patrol-slbid_host_status_request_method_upstream_status	默认关闭	Time2Graph	◦ pv:slbid:host:status:method:upstream_status ◦ body_bytes_sent_avg:slbid:host:status:method:upstream_status ◦ body_bytes_sent_sum:slbid:host:status:method:upstream_status ◦ request_length_avg:slbid:host:status:method:upstream_status ◦ request_length_sum:slbid:host:status:method:upstream_status ◦ upstream_response_time_avg:slbid:host:status:method:upstream_status ◦ upstream_response_time_p50:slbid:host:status:method:upstream_status ◦ upstream_response_time_p90:slbid:host:status:method:upstream_status ◦ upstream_response_time_p99:slbid:host:status:method:upstream_status ◦ upstream_response_time_p9999:slbid:host:status:method:upstream_status ◦ request_time_avg:slbid:host:status:method:upstream_status ◦ request_time_p50:slbid:host:status:method:upstream_status ◦ request_time_p90:slbid:host:status:method:upstream_status ◦ request_time_p99:slbid:host:status:method:upstream_status ◦ request_time_p9999:slbid:host:status:method:upstream_status

- 专属仪表盘

仪表盘名称	关联的Logstore或Metricstore	说明
监控概览	*访问日志Logstore名称*-metrics	展示SLB总体的监控信息，包括PV、失败率、5XX比例、状态码分布、流量等。

仪表盘名称	关联的Logstore或Metricstore	说明
访问概览	*访问日志Logstore名称*	展示用户请求相关的信息，包括PV、UV、移动端分布、国家/省/市分布等。 **说明** 此部分信息基于原始的访问日志全量计算，数据量超大的情况下会有一定延迟。
详细监控	*访问日志Logstore名称*-metrics	支持以slbib、host、url、method、status等维度过滤出实例详细的监控信息。
蓝绿对比	*访问日志Logstore名称*-metrics	用于对两个Host、URL的详细指标进行对比，通常应用于灰度发布、蓝绿发布等场景。
全局巡检结果	○ *访问日志Logstore名称*-metrics ○ *访问日志Logstore名称*-metrics-result	展示流式巡检算法检测出的全局异常信息，包括异常统计以及具体指标上异常的实时显示。
实例巡检结果	○ *访问日志Logstore名称*-metrics ○ *访问日志Logstore名称*-metrics-result	展示流式巡检算法检测出的slbid粒度异常信息，包括异常统计以及具体指标上异常的实时显示。

费用说明

- 目前，负载均衡不针对日志管理功能收取额外费用。
- 负载均衡将日志推送到日志服务后，日志服务根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费。更多信息，请参见日志服务产品定价。

使用限制

- 只有已配置7层监听的负载均衡实例才支持访问日志功能。
- 日志服务Project与负载均衡实例需处于同一地域。

6.2. 配置SLB日志中心

本文介绍如何配置SLB日志中心，将负载均衡7层访问日志采集到日志服务中并进行可视化分析。

前提条件

- 已创建负载均衡实例，详情请参见创建实例。
- 已为负载均衡实例配置7层监听，即配置HTTP监听或HTTPS监听，详情请参见添加HTTP监听或添加HTTPS监听。
- 在负载均衡实例所在地域，已创建日志服务Project和Logstore，详情请参见创建Project和Logstore。

步骤1：配置数据源

1. 登录负载均衡控制台。
2. 在页面左上角，选择地域。
3. 在左侧导航栏，选择**日志管理 > 访问日志**。
4. 根据页面提示，授权负载均衡使用AliyunLogArchiveRole角色访问日志服务。

   > **说明**
   > - 该操作仅在首次配置时需要，且需要由主账号进行授权。
   > - 如果您使用的是RAM用户，该RAM用户需具备相关权限，详情请参见RAM用户授权。
   > - 请勿取消授权或删除RAM角色，否则将导致日志无法正常推送到日志服务。

5. 在**访问日志（7层）**页面，单击目标实例右侧的**设置**。
6. 在**日志设置**页面，选择可用的项目Project和日志库Logstore，并单击**确定**。

   配置完成后，日志服务默认为该Logstore设置索引，如果该Logstore已经设置了索引，原有的索引配置将被覆盖。

步骤2：添加日志中心

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**SLB日志中心**中的**进入应用**。
3. 在**巡检管理**页面，单击**添加**。
4. 在**日志接入**页面中，配置如下参数，并单击**下一步**。

参数	说明
日志中心名称	配置日志中心名称。
项目Project	选择您已创建的Project，该Project需与步骤1：配置数据源中配置的Project保持一致。
日志库Logstore	选择您已创建的Logstore，该Logstore需与步骤1：配置数据源中配置的Logstore保持一致。

5. 在**注意**对话框中，单击**确定**。

   如果您还未开启SLB日志功能，即还未配置数据源，请单击**前往SLB配置**，完成配置，详情请参见步骤1：配置数据源。

6. 在**时序转换配置**中，保持默认配置，单击**下一步**。
7. 在**巡检配置**中，保持默认配置，单击**下一步**。
8. 单击**完成**。

后续步骤

配置完成后，您可在SLB日志中心查看相关的报表并进行日志的查询分析、下载、投递、加工、告警等操作，详情请参见云产品日志通用操作。您还可以执行监控数据的查询分析、告警等操作，详情请参见查询和分析时序数据。

6.3. 配置告警

SLB日志中心为您提供告警和通知功能。当日志数据满足某些条件时，您会收到告警通知，有助于及时发现SLB日志中心异常问题。本文以自定义创建告警实例为例，介绍如何配置告警。

背景信息

日志中心预设了基线告警、同环比告警、智能告警等告警策略，包含QPS、延迟、错误率、流量等，并支持短信、钉钉、邮件、语音、自定义Webhook等通知方式，您可以根据实际应用场景选择开启不同的告警。每种类型的告警特点如下：

- 基线告警：超过或低于某个基准线即触发告警，例如错误率高于1%触发告警。
- 同环比告警：当前的值相比历史某一时间点的变化率，例如流量相比昨天降低10%触发告警。
- 智能告警：日志中心预设了AIOps巡检策略，在发现QPS、延迟等异常的时候会自动记录异常事件，您可以直接设置发现某个异常时触发告警。

配置流程

日志中心中已预设各类告警规则、行动策略、用户组和内容模板等告警所需资源。您可以直接使用预设的告警资源，也可以自定义告警资源，具体配置流程如下。

- 使用预设的告警资源

  如果您希望快速完成告警设置，接收告警通知，只需完成如下配置。完成配置后，日志中心根据对应的告警规则产生告警并使用短信、邮件方式给您创建的用户发送告警通知，如果触发严重告警，则发送电话语音告警通知。

- 自定义告警资源

  如果您希望根据实际场景自定义告警资源，您可以根据如下流程完成配置。完成配置后，日志中心根据对应的告警规则产生告警并根据您配置的告警渠道（语音、短信、邮件、钉钉WebHook、WebHook-自定义和通知中心）给对应的用户或用户组发送告警通知。

本文以**SLB访问错误率监控**为例，介绍告警相关配置。当SLB访问错误率超过2%时，触发一般告警，短信通知运维人员；当SLB访问错误率超过8%时，触发严重告警，电话通知运维人员。

步骤1：创建用户和用户组

配置用户和用户组指定接收通知的*员工A*。请参见如下步骤。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**SLB日志中心**。
3. 在左侧导航栏中，展开目标日志中心实例，单击**告警配置**。
4. 选择**告警管理 > 用户管理**。
5. 创建用户。

   例如创建用户如下。更多信息，请参见创建用户。

标识符	姓名	状态	手机号	邮箱
user1	员工A	正常	86-1381	user1@example.com

6. 创建用户组。

   例如创建用户组如下，并将user1加入该组。更多信息，请参见创建用户组。

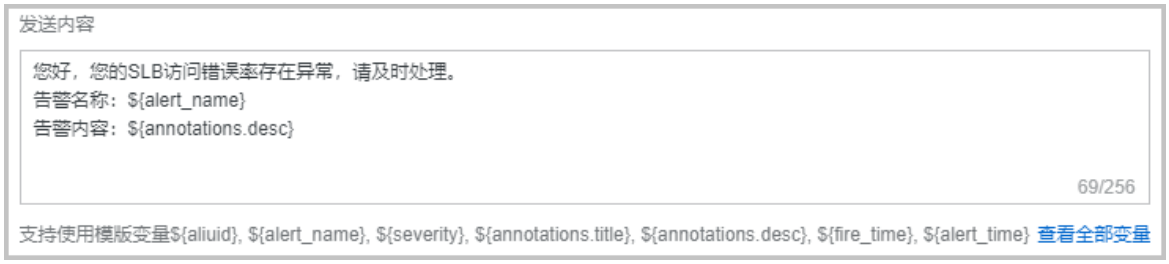
标识符	名称	状态	成员
slb.sls.app.slb.group	SLS SLB日志中心公司A用户组	正常	1个

步骤2：创建内容模板

配置短信和语音通知的内容。请参见如下步骤。

1. 选择**告警管理 > 内容模板**。
2. 在**内容模板**页面中，单击**添加**。
3. 在**添加内容模板**对话框中，配置**ID**、**名称**和**发送内容**，单击**确认**。

   例如短信和语音的发送内容配置如下。发送内容支持使用模板变量。更多信息，请参见创建内容模板。



步骤3：创建行动策略

配置通知员工A的行动策略，包括语音和短信渠道。请参见如下步骤。

1. 选择**告警管理 > 行动策略**。
2. 在**添加行动策略**对话框中，配置如下参数，单击**确认**。

   例如，对于一般告警和严重告警，配置行动策略如下。

参数	描述
ID	行动策略的唯一标识。
名称	行动策略的名称。

参数	描述
严重告警	严重告警行动策略。 ○ 渠道：语音 ○ 接收人：*员工A*。选择您在步骤1：创建用户和用户组中定义的告警通知人员。 ○ 内容模板：*公司A内容模板*。选择您在步骤2：创建内容模板中定义的告警通知内容模板。
一般告警	一般告警行动策略。 ○ 渠道：短信 ○ 接收人：*员工A*。选择您在步骤1：创建用户和用户组中定义的告警通知人员。 ○ 内容模板：*公司A内容模板*。选择您在步骤2：创建内容模板中定义的告警通知内容模板。

步骤4：自定义告警参数并开启实例

SLB日志中心预设的**SLB访问错误率监控**参数不满足告警通知条件，将一般告警阈值修改为2%，严重告警阈值修改为8%。请参见如下步骤。

1. 在**告警规则**页签中，单击**SLB访问错误率监控**对应的**添加**。
2. 在**参数设置**对话框中，按照对应的提示设置参数值，单击**设置并开启**。

   不同告警规则对应的配置参数不同，请根据界面提示填写配置信息即可。例如，配置如下。

   ○ SLB白名单：SLB实例白名单。该白名单中SLB实例才会触发告警。您可以通过SLB控制台获取实例ID。支持正则表达式，默认为 `.*` ，表示监控您阿里云账号下的所有SLB实例。支持配置多个SLB实例，多

个实例之间使用竖线分隔，例如 `lb-1cd34d1238976|lb-1cd34d1238978` 。

- SLB黑名单：SLB实例黑名单。该黑名单中SLB实例不会触发告警。您可以通过SLB控制台获取实例ID。支持正则表达式，默认为空。支持配置多个SLB实例，多个实例之间使用竖线分隔，例如 `lb-1cd34d1238976|lb-1cd34d1238978` 。
- 告警阈值（一般）：触发一般告警的阈值。
- 告警阈值（严重）：触发严重告警的阈值。

步骤5：关联行动策略

在**SLB访问错误率监控**中关联已创建的*公司A行动策略*，请参见如下步骤。

1. 在**告警规则**页签中，选中**SLB访问错误率监控**，单击**配置行动策略**。
2. 在**参数设置**对话框中，从**配置行动策略**列表中，选择**公司A行动策略**，单击**保存**。

相关操作

在**告警规则**页面，您还可以进行如下操作。

操作	说明
关闭告警实例	关闭告警实例，告警规则不会再触发告警，**状态**变更为**未开启**。该操作不会删除参数中定义的配置数据。需要再次开启时，无需重新配置规则参数，可以直接开启。
临时关闭告警实例	临时关闭告警实例后，在设置时长内不再触发告警。
恢复告警实例	处于临时关闭状态的监控实例，可随时恢复告警。
删除告警实例	删除告警实例，**状态**变更为**未创建**。该操作会删除参数中定义的例如SLB白名单、阈值等配置数据。需要再次开启时，需要重新配置参数，才能开启。

6.4. 指标说明

本文介绍基于负载均衡7层访问日志提取的指标详情，包括全局指标、slbid维度指标、status维度指标和upstream_status维度指标。

本文涉及的指标遵循时序数据格式，支持使用PromQL或SQL进行查询分析，详情请参见时序数据查询与分析简介。

全局指标

全局指标信息如下表所示。

指标	说明
pv	总访问次数
body_bytes_sent_avg	发送给客户端的http body平均字节数
body_bytes_sent_sum	发送给客户端的http body总字节数
request_length_avg	请求报文的平均长度
request_length_sum	请求报文的总长度
request_time_avg	请求时间的平均值
request_time_p50	请求时间的50分位值
request_time_p90	请求时间的90分位值
request_time_p99	请求时间的99分位值
request_time_p9999	请求时间的99.99分位值
upstream_response_time_avg	请求连接时长的平均值。 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50	请求连接时长的50分位值
upstream_response_time_p90	请求连接时长的90分位值
upstream_response_time_p99	请求连接时长的99分位值
write_response_time_avg	Proxy写的响应时间的平均值
write_response_time_p50	Proxy写的响应时间的50分位值
write_response_time_p90	Proxy写的响应时间的90分位值
write_response_time_p99	Proxy写的响应时间的99分位值

slbid维度

slbid维度指标的标签为slbid，指标详情如下表所示。

指标	说明
pv:slbid	SLB实例访问次数
body_bytes_sent_avg:slbid	发送给客户端的http body平均字节数

指标	说明
body_bytes_sent_sum:slbid	发送给客户端的http body总字节数
request_length_avg:slbid	请求报文的平均长度
request_length_sum:slbid	请求报文的总长度
request_time_avg:slbid	请求时间的平均值
request_time_p50:slbid	请求时间的50分位值
request_time_p90:slbid	请求时间的90分位值
request_time_p99:slbid	请求时间的99分位值
request_time_p9999:slbid	请求时间的99.99分位值
upstream_response_time_avg:slbid	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:slbid	请求连接时长的50分位值
upstream_response_time_p90:slbid	请求连接时长的90分位值
upstream_response_time_p99:slbid	请求连接时长的99分位值
write_response_time_avg:slbid	Proxy写的响应时间的平均值
write_response_time_p50:slbid	Proxy写的响应时间的50分位值
write_response_time_p90:slbid	Proxy写的响应时间的90分位值
write_response_time_p99:slbid	Proxy写的响应时间的99分位值

status维度

status维度指标的标签为slbid+host+status，指标详情如下表所示。

指标	说明
pv:slbid:host:status	每个slbid、host、status的访问次数
body_bytes_sent_avg:slbid:host:status	发送给客户端的http body平均字节数

指标	说明
body_bytes_sent_sum:slbid:host:status	发送给客户端的http body总字节数
request_length_avg:slbid:host:status	请求报文的平均长度
request_length_sum:slbid:host:status	请求报文的总长度
request_time_avg:slbid:host:status	请求时间的平均值
request_time_p50:slbid:host:status	请求时间的50分位值
request_time_p90:slbid:host:status	请求时间的90分位值
request_time_p99:slbid:host:status	请求时间的99分位值
request_time_p9999:slbid:host:status	请求时间的99.99分位值
upstream_response_time_avg:slbid:host:status	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:slbid:host:status	请求连接时长的50分位值
upstream_response_time_p90:slbid:host:status	请求连接时长的90分位值
upstream_response_time_p99:slbid:host:status	请求连接时长的99分位值
write_response_time_avg:slbid:host:status	Proxy写的响应时间的平均值
write_response_time_p50:slbid:host:status	Proxy写的响应时间的50分位值
write_response_time_p90:slbid:host:status	Proxy写的响应时间的90分位值
write_response_time_p99:slbid:host:status	Proxy写的响应时间的99分位值

upstream_status维度

维度指标的标签为slbid+host+status+request_method+upstream_status+url，指标详情如下表所示。

指标	说明
pv:slbid:host:status:method:upstream_status	每个slbid、host、status、method、url、upstream_status的访问次数
body_bytes_sent_avg:slbid:host:status:method:upstream_status	发送给客户端的http body平均字节数
body_bytes_sent_sum:slbid:host:status:method:upstream_status	发送给客户端的http body总字节数
request_length_avg:slbid:host:status:method:upstream_status	请求报文的平均长度
request_length_sum:slbid:host:status:method:upstream_status	请求报文的总长度
request_time_avg:slbid:host:status:method:upstream_status	请求时间的平均值
request_time_p50:slbid:host:status:method:upstream_status	请求时间的50分位值
request_time_p90:slbid:host:status:method:upstream_status	请求时间的90分位值
request_time_p99:slbid:host:status:method:upstream_status	请求时间的99分位值
request_time_p9999:slbid:host:status:method:upstream_status	请求时间的99.99分位值
upstream_response_time_avg:slbid:host:status:method:upstream_status	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:slbid:host:status:method:upstream_status	请求连接时长的50分位值
upstream_response_time_p90:slbid:host:status:method:upstream_status	请求连接时长的90分位值
upstream_response_time_p99:slbid:host:status:method:upstream_status	请求连接时长的99分位值
write_response_time_avg:slbid:host:status:method:upstream_status	Proxy写的响应时间的平均值

指标	说明
write_response_time_p50:slbid:host:status:method:upstream_status	Proxy写的响应时间的50分位值
write_response_time_p90:slbid:host:status:method:upstream_status	Proxy写的响应时间的90分位值
write_response_time_p99:slbid:host:status:method:upstream_status	Proxy写的响应时间的99分位值

6.5. 日志字段详情

本文介绍负载均衡7层访问日志的字段详情。

字段	说明
__topic__	日志主题，固定为slb_layer7_access_log。
body_bytes_sent	发送给客户端的http body字节数。
client_ip	请求客户端IP地址。
host	优先从请求参数中获取host，如果获取不到则从host header取值，如果还是获取不到则以处理请求的后端服务器IP地址作为host。
http_host	请求报文host header的内容。
http_referer	Proxy收到的请求报文中HTTP的referer header的内容。
http_user_agent	Proxy收到的请求报文中HTTP的user-agent header的内容。
http_x_forwarded_for	Proxy收到的请求报文中x-forwarded-for header的内容。
http_x_real_ip	真实的客户端IP地址。
read_request_time	Proxy读取请求的时间，单位：毫秒。
request_length	请求报文的长度，包括startline、http header和http body。
request_method	请求报文的方法。
request_time	Proxy收到第一个请求报文的时间到proxy返回应答之间的间隔时间，单位：秒。
request_uri	Proxy收到的请求报文的URI。
scheme	请求的schema，包括http、https。
server_protocol	Proxy收到的HTTP协议的版本，例如HTTP/1.0或HTTP/1.1。
slb_vport	负载均衡的监听端口。

字段	说明
slbid	负载均衡实例ID。
ssl_cipher	建立SSL连接使用的密码，例如ECDHE-RSA-AES128-GCM-SHA256等。
ssl_protocol	建立SSL连接使用的协议，例如TLSv1.2。
status	Proxy应答报文的状态。
tcpinfo_rtt	客户端TCP连接时间，单位：微秒。
time	日志记录时间。
upstream_addr	后端服务器的IP地址和端口。
upstream_response_time	从负载均衡向后端建立连接开始到接受完数据然后关闭连接为止的时间，单位：秒。
upstream_status	Proxy收到的后端服务器的响应状态码。
vip_addr	虚拟IP地址。
write_response_time	Proxy写的响应时间，单位：毫秒。

7.Kubernetes Ingress日志中心

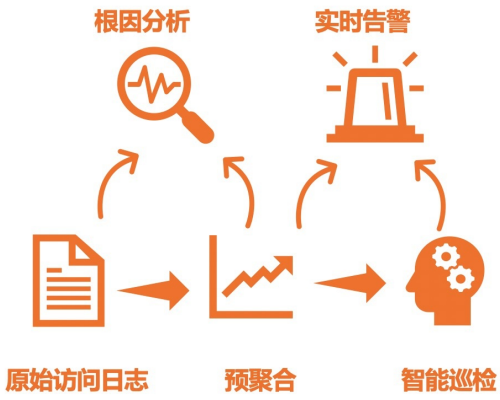
7.1. 使用前须知

阿里云Kubernetes Ingress组件除了提供外部可访问的URL、负载均衡、SSL、基于名称的虚拟主机外，还支持将所有您的HTTP请求日志记录到标准输出中。日志服务推出Ingress日志中心功能，用于分析和监控Ingress后端对接的服务状态。本文介绍Ingress日志中心相关的功能说明、功能优势、资产说明、费用说明、使用限制等信息。

功能说明

Ingress日志中心基于实时访问日志进行自动聚合，并提供智能巡检、实时告警等功能，详细功能如下：

- 实时访问日志的采集、存储、查询、分析。
- 基于原始访问日志实时提取出各类指标信息，包括PV、请求成功率、平均延迟、P50/P99/P9999延迟、出入流量等。并支持多个维度组合，包括host、status、url。
- 丰富可视化报表，包括监控大盘、异常事件、运营大盘等，支持报表邮件、钉钉群订阅。
- 提供智能巡检功能，支持全局以及Service粒度巡检，并支持可视化报表中直接标注异常点。
- 自定义告警配置，告警通知直接对接消息中心、邮件、短信、语音（电话）、钉钉，并支持对接自定义WebHook。



功能优势

- 简单：一站式开通、中心化使用，无需关心日志收集、存储、计算、可视化等问题，将开发、运维人员从日志处理的繁琐耗时中解放出来，将更多的精力集中到业务开发和技术探索上去。
- 海量：访问日志与Ingress请求PV成正比，数据规模很大，处理访问日志需要考虑性能和成本问题。日志中心可自定配置预聚和功能，实时计算聚合指标，计算后的聚合结果可降低几个数量级，使查询速度大大提升。
- 实时：DevOps、监控、报警等场景要求日志数据的实时性。结合日志服务强大的大数据计算能力，秒级分析处理实时产生的日志。
- 弹性：可任意设置日志存储周期。Logstore容量可动态伸缩满足业务增长需求。
- 智能：基于达摩院智能AIOps算法，提供各类指标自动巡检功能，有助于更快、更准确的发现并定位问题。

资产说明

所有资产都在您选择的Project下，Project内的资产如下：

- Logstore
  - 访问日志Logstore用于存储Kubernetes Ingress访问日志，该Logstore为您自定义创建的Logstore。
    - 该Logstore默认开启索引，并配置部分字段的索引。您可以增加索引字段，修改索引后只对新数据生效。
    - 您可以自定义修改日志存储时间，详情请参见修改Logstore配置。
  - 巡检结果Logstore用于存储巡检结果。开通日志中心功能后，自动生成该专属Logstore，其名称为*访问日志Logstore名称*-metrics-result。

> **说明**
> - 请勿删除Kubernetes Ingress访问日志相关的Logstore，否则将无法正常采集日志到日志服务。
> - 请勿删除访问日志Logstore中的部分字段的索引，否则指标转换会失败。

- Metricstore

  监控指标Metricstore用于存储聚和后的指标信息。开通日志中心功能后，自动生成该专属Metricstore，其名称为*访问日志Logstore名称*-metrics。

> **说明** 监控指标Metricstore存储的是聚合后的指标，数据量相比原始访问日志大大降低，非常适用于长期存储。

- 聚合规则

规则名称	聚合时间粒度	聚合维度	生成指标名
total	10秒	total	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99 ◦ upstream_response_time_p9999 ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ request_time_p9999

规则名称	聚合时间粒度	聚合维度	生成指标名
host	10秒	host	◦ pv:host ◦ body_bytes_sent_avg:host ◦ body_bytes_sent_sum:host ◦ request_length_avg:host ◦ request_length_sum:host ◦ upstream_response_time_avg:host ◦ upstream_response_time_p50:host ◦ upstream_response_time_p90:host ◦ upstream_response_time_p99:host ◦ upstream_response_time_p9999:host ◦ request_time_avg:host ◦ request_time_p50:host ◦ request_time_p90:host ◦ request_time_p99:host ◦ request_time_p9999:host
host_status	10秒	host+status	◦ pv:host:status ◦ body_bytes_sent_avg:host:status ◦ body_bytes_sent_sum:host:status ◦ request_length_avg:host:status ◦ request_length_sum:host:status ◦ upstream_response_time_avg:host:status ◦ upstream_response_time_p50:host:status ◦ upstream_response_time_p90:host:status ◦ upstream_response_time_p99:host:status ◦ upstream_response_time_p9999:host:status ◦ request_time_avg:host:status ◦ request_time_p50:host:status ◦ request_time_p90:host:status ◦ request_time_p99:host:status ◦ request_time_p9999:host:status

规则名称	聚合时间粒度	聚合维度	生成指标名
host+status+method+upstream_name+upstream_status+url	10秒	host+status+method+upstream_name+upstream_status+url	◦ pv:host:status:method:upstream_name:upstream_status:url ◦ body_bytes_sent_avg:host:status:method:upstream_name:upstream_status:url ◦ body_bytes_sent_sum:host:status:method:upstream_name:upstream_status:url ◦ request_length_avg:host:status:method:upstream_name:upstream_status:url ◦ request_length_sum:host:status:method:upstream_name:upstream_status:url ◦ upstream_response_time_avg:host:status:method:upstream_name:upstream_status:url ◦ upstream_response_time_p50:host:status:method:upstream_name:upstream_status:url ◦ upstream_response_time_p90:host:status:method:upstream_name:upstream_status:url ◦ upstream_response_time_p99:host:status:method:upstream_name:upstream_status:url ◦ upstream_response_time_p9999:host:status:method:upstream_name:upstream_status:url ◦ request_time_avg:host:status:method:upstream_name:upstream_status:url ◦ request_time_p50:host:status:method:upstream_name:upstream_status:url ◦ request_time_p90:host:status:method:upstream_name:upstream_status:url ◦ request_time_p99:host:status:method:upstream_name:upstream_status:url ◦ request_time_p9999:host:status:method:upstream_name:upstream_status:url

- 巡检规则

规则名称	开启状态	巡检算法	巡检指标
total	默认开启	Time2Graph	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ request_time_avg
host	默认开启	Time2Graph	◦ pv:host ◦ body_bytes_sent_avg:host ◦ body_bytes_sent_sum:host ◦ request_length_avg:host ◦ request_length_sum:host ◦ upstream_response_time_avg:host ◦ request_time_avg:host
host_status	默认关闭	Time2Graph	◦ pv:host:status ◦ body_bytes_sent_avg:host:status ◦ body_bytes_sent_sum:host:status ◦ request_length_avg:host:status ◦ request_length_sum:host:status ◦ upstream_response_time_avg:host:status ◦ request_time_avg:host:status

规则名称	开启状态	巡检算法	巡检指标
host_status_method_proxy_upstream_name_upstream_status_url	默认关闭	Time2Graph	○ pv:host:status:method:upstream_name:upstream_status:url ○ body_bytes_sent_avg:host:status:method:upstream_name:upstream_status:url ○ body_bytes_sent_sum:host:status:method:upstream_name:upstream_status:url ○ request_length_avg:host:status:method:upstream_name:upstream_status:url ○ request_length_sum:host:status:method:upstream_name:upstream_status:url ○ upstream_response_time_avg:host:status:method:upstream_name:upstream_status:url ○ request_time_avg:host:status:method:upstream_name:upstream_status:url

- 专属仪表盘

仪表盘名称	关联的Logstore、Metricstore	说明
运营大盘	*访问日志Logstore名称*	展示用户请求相关的信息，包括PV、UV、移动端分布、国家/省/市分布等。 **说明** 此部分信息基于原始访问日志全量计算，数据量超大的情况下会有一定延迟。
概览	*访问日志Logstore名称*-metrics	展示Kubernetes总体的监控信息，包括PV、失败率、5XX比例、状态码分布、流量等。
监控大盘	*访问日志Logstore名称*-metrics	支持以host、url、status等维度过滤出实例详细的监控信息。
异常事件	○ *访问日志Logstore名称*-metrics ○ *访问日志Logstore名称*-metrics-result	展示流式巡检算法检测出的Service粒度异常信息，包括异常统计以及具体指标上异常的实时显示。

费用说明

日志服务根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费，详情请参见日志服务产品定价。

使用限制

- 必须成功解析Ingress日志后才能进行时序转换规则配置和巡检配置。对于自定义日志格式的Ingress访问日志，需手动配置解析规则解析日志，对应的日志字段名称需要符合默认的字段命名规则。
- 日志中心配置完成后只对新产生的日志生效，存量日志并不会转换成指标信息。

7.2. 配置Ingress日志中心

本文介绍如何开通Ingress访问日志中心，将Ingress日志实时采集到日志服务中并进行可视化分析。

前提条件

已安装日志组件。具体操作，请参见安装Logtail日志组件。

默认情况下，在创建Kubernetes集群时自动安装日志组件。

步骤1：部署Ingress采集配置

日志服务采集配置针对Kubernetes进行了CRD扩展，alibaba-log-controller组件会根据您定义的AliyunLogConfig CRD自动创建日志服务相关采集配置和报表资源。

1. 在Kubernetes集群中，定义AliyunLogConfig CRD配置。

   > **说明**
   > - 请确保日志组件alibaba-log-controller版本不低于0.2.0.0-76648ee-aliyun。
   >
   >   如果您在应用了CRD配置后要更新组件版本，请在更新组件版本后，删除该CRD配置并重新应用。
   > - 此处的CRD配置只对ACK默认的Ingress Controller中的访问日志格式生效。如果您修改过Ingress Controller的访问日志格式，请修改此处CRD配置中的正则表达式提取processor_regex部分，具体修改内容请参见通过DaemonSet-CRD方式采集日志中的CRD配置。
   > - 如果您当前没有其他系统依赖访问日志，则推荐您将访问日志格式设置为日志服务推荐的格式。设置方式：执行**kubectl edit configmap -n kube-system nginx-configuration**命令修改configmap，将其中的log-format-upstream字段修改为如下内容：
   >
   >   ```
   >   log-format-upstream: $the_real_ip - [$the_real_ip] - $remote_user [$time_local] "$request
   >   " $status
   >   $body_bytes_sent "$http_referer" "$http_user_agent" $request_length $request_time [$
   >   proxy_upstream_name]
   >   $upstream_addr $upstream_response_length $upstream_response_time $upstream_stat
   >   us $req_id $host
   >   ```

   ```
   apiVersion: log.alibabacloud.com/v1alpha1
   kind: AliyunLogConfig
   metadata:
    # your config name, must be unique in you k8s cluster
    name: k8s-nginx-ingress
   spec:
   ```

```
 # logstore name to upload log
 logstore: nginx-ingress
 # product code, only for k8s nginx ingress
 productCode: k8s-nginx-ingress
 # logtail config detail
 logtailConfig:
  inputType: plugin
  # logtail config name, should be same with [metadata.name]
  configName: k8s-nginx-ingress
  inputDetail:
   plugin:
    inputs:
    - type: service_docker_stdout
     detail:
      IncludeLabel:
       io.kubernetes.container.name: nginx-ingress-controller
      Stderr: false
      Stdout: true
    processors:
    - type: processor_regex
     detail:
      KeepSource: false
      Keys:
      - client_ip
      - x_forward_for
      - remote_user
      - time
      - method
      - url
      - version
      - status
      - body_bytes_sent
      - http_referer
      - http_user_agent
      - request_length
      - request_time
      - proxy_upstream_name
      - upstream_addr
      - upstream_response_length
      - upstream_response_time
      - upstream_status
      - req_id
      - host
      - proxy_alternative_upstream_name
      NoKeyError: true
      NoMatchError: true
      Regex: ^(\S+)\s-\s\[([^]]+)]\s-\s(\S+)\s\[(\S+)\s\S+\s"(\w+)\s(\S+)\s([^"]+)"\s(\d+)\s(\d+)\s"([^"]*)"\s"([^"]*)"\s(\S+)\s(\S+)+\s\[([^]]*)]\s(\S+)\s(\S+)\s(\S+)\s(\S+)\s(\S+)\s*(\S*)\s*\[*([^]]*)\]*.*
      SourceKey: content
```

2. 部署Ingress采集配置。

   您可以选择如下任意一种方式进行部署：

   - 方式1：执行kubectl命令完成部署。

- 方式2：将步骤1中的AliyunLogConfig CRD配置保存为nginx-ingress.yaml文件，执行**kubectl apply -n kube-system -f**命令完成部署。
- 方式3：使用编排模板完成部署。
  a. 登录容器服务管理控制台。
  b. 将步骤1中的AliyunLogConfig CRD配置保存为编排模板。具体操作，请参见创建编排模板。
  c. 基于您所创建的模板创建应用。具体操作，请参见通过编排模板创建Linux应用。

  其中**命名空间**选择为您所在集群的默认命名空间。

步骤2：添加日志中心

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**Ingress日志中心**。
3. 在**巡检管理**页签中，单击**添加**。
4. 在**添加日志中心**面板中，配置如下参数，并单击**确定**。

参数	说明
日志中心名称	配置日志中心名称。
项目Project	选择您已创建的Project。
日志库Logstore	选择您已创建的Logstore，该Logstore需与步骤1：部署Ingress采集配置中配置的Logstore保持一致。

后续步骤

配置完成后，您可在Ingress日志中心查看相关的报表并进行日志的查询分析、下载、投递、加工、告警等操作。具体操作，请参见云产品日志通用操作。您还可以执行监控数据的查询分析、告警等操作。具体操作，请参见查询和分析时序数据。

7.3. 指标说明

本文介绍基于Kubernetes Ingress访问日志提取的指标详情，包括全局指标、host维度指标、status维度指标和url维度指标。

本文涉及的指标遵循时序数据格式，支持使用PromQL或SQL进行查询分析，详情请参见时序数据查询分析简介。

全局指标

全局指标信息如下表所示。

指标	说明
pv	总访问次数
body_bytes_sent_avg	发送给客户端的http body平均字节数
body_bytes_sent_sum	发送给客户端的http body总字节数
request_length_avg	请求报文的平均长度

指标	说明
request_length_sum	请求报文的总长度
request_time_avg	请求时间的平均值
request_time_p50	请求时间的50分位值
request_time_p90	请求时间的90分位值
request_time_p99	请求时间的99分位值
request_time_p9999	请求时间的99.99分位值
upstream_response_time_avg	请求连接时长的平均值
upstream_response_time_p50	请求连接时长的50分位值
upstream_response_time_p90	请求连接时长的90分位值
upstream_response_time_p99	请求连接时长的99分位值

host维度

host维度指标的标签为host，指标详情如下表所示。

指标	说明
pv:host	每个host访问次数
body_bytes_sent_avg:host	发送给客户端的http body平均字节数
body_bytes_sent_sum:host	发送给客户端的http body总字节数
request_length_avg:host	请求报文的平均长度
request_length_sum:host	请求报文的总长度
request_time_avg:host	请求时间的平均值
request_time_p50:host	请求时间的50分位值
request_time_p90:host	请求时间的90分位值
request_time_p99:host	请求时间的99分位值
request_time_p9999:host	请求时间的99.99分位值
upstream_response_time_avg:host	请求连接时长的平均值
upstream_response_time_p50:host	请求连接时长的50分位值

指标	说明
upstream_response_time_p90:host	请求连接时长的90分位值
upstream_response_time_p99:host	请求连接时长的99分位值

status维度

status维度指标的标签为host+status，指标详情如下表所示。

指标	说明
pv:host:status	每个host、status的访问次数
body_bytes_sent_avg:host:status	发送给客户端的http body平均字节数
body_bytes_sent_sum:host:status	发送给客户端的http body总字节数
request_length_avg:host:status	请求报文的平均长度
request_length_sum:host:status	请求报文的总长度
request_time_avg:host:status	请求时间的平均值
request_time_p50:host:status	请求时间的50分位值
request_time_p90:host:status	请求时间的90分位值
request_time_p99:host:status	请求时间的99分位值
request_time_p9999:host:status	请求时间的99.99分位值
upstream_response_time_avg:host:status	请求连接时长的平均值。
upstream_response_time_p50:host:status	请求连接时长的50分位值
upstream_response_time_p90:host:status	请求连接时长的90分位值
upstream_response_time_p99:host:status	请求连接时长的99分位值

url维度

维度指标的标签为host+status+method+upstream_name+upstream_status+url，指标详情如下表所示。

指标	说明
pv:host:status:method:upstream_name:upstream_status:url	每个host、status、method、upstream_name、url、upstream_status的访问次数
body_bytes_sent_avg:host:status:method:upstream_name:upstream_status:url	发送给客户端的http body平均字节数

指标	说明
body_bytes_sent_sum:host:status:method:upstream_name:upstream_status:url	发送给客户端的http body总字节数
request_length_avg:host:status:method:upstream_name:upstream_status:url	的请求报文的平均长度
request_length_sum:host:status:method:upstream_name:upstream_status:url	请求报文的总长度
request_time_avg:host:status:method:upstream_name:upstream_status:url	请求时间的平均值
request_time_p50:host:status:method:upstream_name:upstream_status:url	请求时间的50分位值
request_time_p90:host:status:method:upstream_name:upstream_status:url	请求时间的90分位值
request_time_p99:host:status:method:upstream_name:upstream_status:url	请求时间的99分位值
request_time_p9999:host:status:method:upstream_name:upstream_status:url	请求时间的99.99分位值
upstream_response_time_avg:host:status:method:upstream_name:upstream_status:url	请求连接时长的平均值
upstream_response_time_p50:host:status:method:upstream_name:upstream_status:url	请求连接时长的50分位值
upstream_response_time_p90:host:status:method:upstream_name:upstream_status:url	请求连接时长的90分位值
upstream_response_time_p99:host:status:method:upstream_name:upstream_status:url	请求连接时长的99分位值
upstream_response_time_p9999:host:status:method:upstream_name:upstream_status:url	请求连接时长的99.99分位值

7.4. 日志字段详情

本文介绍Kubernetes Ingress访问日志的字段详情。

字段	说明
body_bytes_sent	发送给客户端的http body字节数。
client_ip	请求客户端IP地址。
host	优先从请求参数中获取host，如果获取不到则从host header取值，如果仍获取不到则以处理请求的后端服务器IP地址作为host。

字段	说明
http_referer	Proxy收到的请求报文中HTTP的referer header的内容。
http_user_agent	Proxy收到的请求报文中HTTP的user-agent header的内容。
x_forwarded_for	Proxy收到的请求报文中x-forwarded-for的内容。
request_length	请求报文的长度，包括startline、http header和http body。
method	请求报文的方法。
request_time	Proxy收到第一个请求报文的时间到proxy返回应答之间的间隔时间，单位：秒。
url	收到的请求报文的URI。
version	Proxy收到的HTTP协议的版本，例如HTTP/1.0或HTTP/1.1。
status	Proxy应答报文的状态。
time	日志记录时间。
upstream_addr	后端服务器的IP地址和端口。
upstream_response_time	从负载均衡向后端建立连接开始到接受完数据然后关闭连接为止的时间，单位：秒。
upstream_status	Proxy收到的后端服务器的响应状态码。
proxy_upstream_name	Proxy转发请求的后端服务名，在Kubernetes中的命名规则为*namespace-service-port*。
proxy_alternative_upstream_name	可选的Proxy转发请求的后端服务名，在Kubernetes中的命名规则为*namespace-service-port*。通常存在该值时可以忽略proxy_upstream_name字段。

8.ALB日志中心

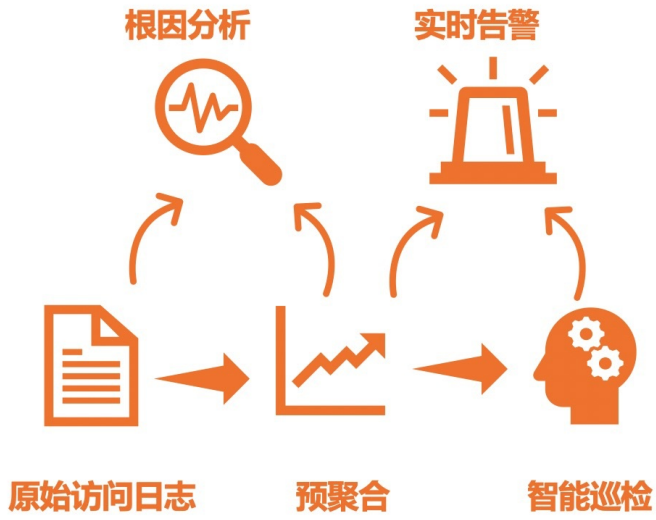
8.1. 使用前须知

阿里云日志服务ALB（Application Load Balancer）日志中心提供负载均衡7层日志分析、秒级监控指标分析、实时告警等功能，并提供基于AIOps的自动异常巡检功能。您可以通过ALB日志中心了解客户端用户行为、客户端用户的地域分布、请求成功率、响应延迟等。本文介绍ALB日志中心相关的功能说明、功能优势、资产说明、费用说明、使用限制等信息。

功能说明

ALB日志中心基于实时访问日志进行自动聚合，并提供智能巡检、实时告警等功能，详细功能如下：

- 实时访问日志的存储、查询、分析。
- 基于原始访问日志实时提取各类指标信息，包括PV、请求成功率、平均延迟、P50延迟、P99延迟、P9999延迟、出入流量等。并支持多个维度组合，包括app_lb_id、host、method、status。
- 提供丰富的可视化报表，包括监控大盘、异常事件、运营大盘等，并支持报表邮件、钉钉群订阅。
- 提供智能巡检功能，支持全局巡检和app_lb_id粒度巡检，并支持在可视化报表中直接标注异常点。
- 自定义告警配置，告警通知直接对接消息中心、短信、邮件、语音（电话）、钉钉，并支持对接自定义WebHook。



功能优势

- 简单：一站式开通、中心化使用，无需关心日志收集、存储、计算、可视化等问题，将开发、运维人员从日志处理的繁琐耗时中解放出来，将更多的精力集中到业务开发和技术探索上去。
- 海量：访问日志与负载均衡实例请求PV成正比，数据规模很大，处理访问日志需要考虑性能和成本问题。日志中心可自定配置预聚和功能，实时计算聚合指标，计算后的聚合结果可降低几个数量级，使查询速度大大提升。
- 实时：DevOps、监控、报警等场景要求日志数据的实时性。负载均衡结合日志服务强大的大数据计算能力，秒级分析处理实时产生的日志。
- 弹性：您可按负载均衡实例级别开通或关闭访问日志功能，可任意设置日志存储周期。Logstore容量可动态伸缩满足业务增长需求。
- 智能：基于达摩院智能AIOps算法，提供ALB指标自动巡检功能，有助于更快、更准确的发现并定位问题。

资产说明

所有资产都在您选择的Project下，Project内的资产如下：

- Logstore
  - 访问日志Logstore用于存储负载均衡7层访问日志，该Logstore为您自定义创建的Logstore。
  - 巡检结果Logstore用于存储巡检结果。开通日志中心功能后，自动生成该专属Logstore，其名称为*访问日志Logstore名称*-metrics-result。

> **说明**
> - 请勿删除负载均衡7层访问日志相关的Logstore，否则将无法正常采集日志到日志服务。
> - 请勿删除访问日志Logstore中的部分字段的索引，否则指标转换会失败。

- Metricstore

  监控指标Metricstore用于存储聚和后的指标信息。开通日志中心功能后，自动生成该专属Metricstore，其名称为*访问日志Logstore名称*-metrics。

> **说明** 监控指标Metricstore存储的是聚合后的指标，数据量相比原始访问日志大大降低，非常适用于长期存储。

- 聚合规则

规则名称	聚合时间粒度	聚合维度	生成指标名
total	10秒	total	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99 ◦ upstream_response_time_p9999 ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ request_time_p9999

规则名称	聚合时间粒度	聚合维度	生成指标名
app_lb_id	10秒	app_lb_id	◦ pv:alb ◦ body_bytes_sent_avg:alb ◦ body_bytes_sent_sum:alb ◦ request_length_avg:alb ◦ request_length_sum:alb ◦ upstream_response_time_avg:alb ◦ upstream_response_time_p50:alb ◦ upstream_response_time_p90:alb ◦ upstream_response_time_p99:alb ◦ upstream_response_time_p9999:alb ◦ request_time_avg:alb ◦ request_time_p50:alb ◦ request_time_p90:alb ◦ request_time_p99:alb ◦ request_time_p9999:alb

规则名称	聚合时间粒度	聚合维度	生成指标名
app_lb_id_host_status	10秒	app_lb_id+host+status	◦ pv:app_lb_id:host:status ◦ body_bytes_sent_avg:app_lb_id:host:status ◦ body_bytes_sent_sum:app_lb_id:host:status ◦ request_length_avg:app_lb_id:host:status ◦ request_length_sum:app_lb_id:host:status ◦ upstream_response_time_avg:app_lb_id:host:status ◦ upstream_response_time_p50:app_lb_id:host:status ◦ upstream_response_time_p90:app_lb_id:host:status ◦ upstream_response_time_p99:app_lb_id:host:status ◦ upstream_response_time_p9999:app_lb_id:host:status ◦ request_time_avg:app_lb_id:host:status ◦ request_time_p50:app_lb_id:host:status ◦ request_time_p90:app_lb_id:host:status ◦ request_time_p99:app_lb_id:host:status ◦ request_time_p9999:app_lb_id:host:status

规则名称	聚合时间粒度	聚合维度	生成指标名
app_lb_id+host+status+request_method+upstream_status+url	10秒	app_lb_id+host+status+request_method+upstream_status+url	◦ pv:app_lb_id:host:status:method:upstream_status ◦ body_bytes_sent_avg:app_lb_id:host:status:method:upstream_status ◦ body_bytes_sent_sum:app_lb_id:host:status:method:upstream_status ◦ request_length_avg:app_lb_id:host:status:method:upstream_status ◦ request_length_sum:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_avg:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p50:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p90:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p99:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p9999:app_lb_id:host:status:method:upstream_status ◦ request_time_avg:app_lb_id:host:status:method:upstream_status ◦ request_time_p50:app_lb_id:host:status:method:upstream_status ◦ request_time_p90:app_lb_id:host:status:method:upstream_status ◦ request_time_p99:app_lb_id:host:status:method:upstream_status ◦ request_time_p9999:app_lb_id:host:status:method:upstream_status

- 巡检规则

规则名称	开启状态	巡检算法	巡检指标

规则名称	开启状态	巡检算法	巡检指标
alb-patrol-total	默认开启	Time2Graph	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99 ◦ upstream_response_time_p9999 ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ request_time_p9999
alb-patrol-alb	默认开启	Time2Graph	◦ pv:alb ◦ body_bytes_sent_avg:alb ◦ body_bytes_sent_sum:alb ◦ request_length_avg:alb ◦ request_length_sum:alb ◦ upstream_response_time_avg:alb ◦ upstream_response_time_p50:alb ◦ upstream_response_time_p90:alb ◦ upstream_response_time_p99:alb ◦ upstream_response_time_p9999:alb ◦ request_time_avg:alb ◦ request_time_p50:alb ◦ request_time_p90:alb ◦ request_time_p99:alb ◦ request_time_p9999:alb

规则名称	开启状态	巡检算法	巡检指标
alb-patrol-app_lb_id_host_status	默认关闭	Time2Graph	◦ pv:app_lb_id:host:status ◦ body_bytes_sent_avg:app_lb_id:host:status ◦ body_bytes_sent_sum:app_lb_id:host:status ◦ request_length_avg:app_lb_id:host:status ◦ request_length_sum:app_lb_id:host:status ◦ upstream_response_time_avg:app_lb_id:host:status ◦ upstream_response_time_p50:app_lb_id:host:status ◦ upstream_response_time_p90:app_lb_id:host:status ◦ upstream_response_time_p99:app_lb_id:host:status ◦ upstream_response_time_p9999:app_lb_id:host:status ◦ request_time_avg:app_lb_id:host:status ◦ request_time_p50:app_lb_id:host:status ◦ request_time_p90:app_lb_id:host:status ◦ request_time_p99:app_lb_id:host:status ◦ request_time_p9999:app_lb_id:host:status

规则名称	开启状态	巡检算法	巡检指标
alb-patrol-app_lb_id_host_status_request_method_upstream_status	默认关闭	Time2Graph	◦ pv:app_lb_id:host:status:method:upstream_status ◦ body_bytes_sent_avg:app_lb_id:host:status:method:upstream_status ◦ body_bytes_sent_sum:app_lb_id:host:status:method:upstream_status ◦ request_length_avg:app_lb_id:host:status:method:upstream_status ◦ request_length_sum:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_avg:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p50:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p90:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p99:app_lb_id:host:status:method:upstream_status ◦ upstream_response_time_p9999:app_lb_id:host:status:method:upstream_status ◦ request_time_avg:app_lb_id:host:status:method:upstream_status ◦ request_time_p50:app_lb_id:host:status:method:upstream_status ◦ request_time_p90:app_lb_id:host:status:method:upstream_status ◦ request_time_p99:app_lb_id:host:status:method:upstream_status ◦ request_time_p9999:app_lb_id:host:status:method:upstream_status

- 专属仪表盘

仪表盘名称	关联的Logstore或Metricstore	说明
概览	*访问日志Logstore名称*-metrics	展示ALB总体的监控信息，包括PV、失败率、5XX比例、状态码分布、流量等。

仪表盘名称	关联的Logstore或Metricstore	说明
运营大盘	*访问日志Logstore名称*	展示用户请求相关的信息，包括PV、UV、移动端分布、国家/省/市分布等。 **说明** 此部分信息基于原始的访问日志全量计算，数据量超大的情况下会有一定延迟。
监控大盘	*访问日志Logstore名称*-metrics	支持以app_lb_id、host、request_uri、request_method、status等维度过滤出实例详细的监控信息。
蓝绿对比	*访问日志Logstore名称*-metrics	用于对两个host、url的详细指标进行对比，通常应用于灰度发布、蓝绿发布等场景。
秒级监控	◦ *访问日志Logstore名称*-metrics ◦ *访问日志Logstore名称*-metrics-result	以秒级粒度展示监控信息，便于发现瞬时抖动的异常情况。
异常事件	◦ *访问日志Logstore名称*-metrics ◦ *访问日志Logstore名称*-metrics-result	展示流式巡检算法检测出的app_lb_id粒度异常信息，包括异常统计以及具体指标上异常的实时显示。

费用说明

- 阿里云负载均衡不针对日志管理功能收取额外费用。
- 阿里云负载均衡将日志推送到日志服务后，日志服务根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费。更多信息，请参见日志服务产品定价。

使用限制

日志服务Project与负载均衡实例需处于同一地域。

8.2. 配置ALB日志中心

本文介绍如何配置ALB日志中心，将负载均衡7层访问日志采集到日志服务中并进行可视化分析。

前提条件

- 已创建应用型负载均衡实例。更多信息，请参见创建实例。
- 在负载均衡实例所在地域，已创建日志服务Project和Logstore。更多信息，请参见创建Project和创建Logstore。

步骤1：配置数据源

1. 登录负载均衡控制台。
2. 在左侧导航栏，选择**应用型负载均衡ALB > 实例**。
3. 单击目标实例。
4. 在**实例详情**页签的**实例属性**区域，打开**访问日志**开关。
5. 在**访问日志**面板中，选择您已创建的Project和Logstore，单击**确定**。

   > **说明** 开通访问日志功能后，系统将为您自动创建AliyunServiceRoleForAlbLogDelivery服务角色。

6. 在弹出的对话框中，单击**确定**。

   配置完成后，日志服务默认为该Logstore设置索引，如果该Logstore已经设置了索引，原有的索引配置将被覆盖。

步骤2：添加日志中心

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**ALB日志中心**。
3. 在**巡检管理**页面，单击**添加**。
4. 在**添加日志中心**面板中，配置如下参数，并单击**确定**。

参数	说明
日志中心名称	配置日志中心名称。
项目Project	选择您已创建的Project，该Project需与步骤1：配置数据源中配置的Project保持一致。
日志库Logstore	选择您已创建的Logstore，该Logstore需与步骤1：配置数据源中配置的Logstore保持一致。

5. 单击**确定**。

后续步骤

配置完成后，您可在ALB日志中心查看相关的报表并进行日志的查询分析、下载、投递、加工、告警等操作。更多信息，请参见云产品日志通用操作。您还可以执行监控数据的查询分析、告警等操作。更多信息，请参见查询和分析时序数据。

8.3. 指标说明

本文介绍基于负载均衡7层访问日志提取的指标详情，包括全局指标、app_lb_id维度指标、status维度指标和upstream_status维度指标。

本文涉及的指标遵循时序数据格式，支持使用PromQL或SQL进行查询分析。更多信息，请参见时序数据查询分析简介。

全局指标

全局指标信息如下表所示。

指标	说明
pv	总访问次数
body_bytes_sent_avg	发送给客户端的HTTP Body平均字节数
body_bytes_sent_sum	发送给客户端的HTTP Body总字节数
request_length_avg	请求报文的平均长度
request_length_sum	请求报文的总长度
request_time_avg	请求时间的平均值
request_time_p50	请求时间的50分位值
request_time_p90	请求时间的90分位值
request_time_p99	请求时间的99分位值
request_time_p9999	请求时间的99.99分位值
upstream_response_time_avg	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50	请求连接时长的50分位值
upstream_response_time_p90	请求连接时长的90分位值
upstream_response_time_p99	请求连接时长的99分位值
write_response_time_avg	Proxy写的响应时间的平均值
write_response_time_p50	Proxy写的响应时间的50分位值
write_response_time_p90	Proxy写的响应时间的90分位值
write_response_time_p99	Proxy写的响应时间的99分位值

app_lb_id维度

app_lb_id维度指标的标签为app_lb_id，指标详情如下表所示。

指标	说明
pv:app_lb_id	ALB实例访问次数
body_bytes_sent_avg:app_lb_id	发送给客户端的HTTP Body平均字节数

指标	说明
body_bytes_sent_sum:app_lb_id	发送给客户端的HTTP Body总字节数
request_length_avg:app_lb_id	请求报文的平均长度
request_length_sum:app_lb_id	请求报文的总长度
request_time_avg:app_lb_id	请求时间的平均值
request_time_p50:app_lb_id	请求时间的50分位值
request_time_p90:app_lb_id	请求时间的90分位值
request_time_p99:app_lb_id	请求时间的99分位值
request_time_p9999:app_lb_id	请求时间的99.99分位值
upstream_response_time_avg:app_lb_id	请求连接时长的平均值 说明 upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:app_lb_id	请求连接时长的50分位值
upstream_response_time_p90:app_lb_id	请求连接时长的90分位值
upstream_response_time_p99:app_lb_id	请求连接时长的99分位值
write_response_time_avg:app_lb_id	Proxy写的响应时间的平均值
write_response_time_p50:app_lb_id	Proxy写的响应时间的50分位值
write_response_time_p90:app_lb_id	Proxy写的响应时间的90分位值
write_response_time_p99:app_lb_id	Proxy写的响应时间的99分位值

status维度

status维度指标的标签为app_lb_id+host+status，指标详情如下表所示。

指标	说明
pv:app_lb_id:host:status	每个app_lb_id、host、status的访问次数

指标	说明
body_bytes_sent_avg:app_lb_id:host:status	发送给客户端的HTTP Body平均字节数
body_bytes_sent_sum:app_lb_id:host:status	发送给客户端的HTTP Body总字节数
request_length_avg:app_lb_id:host:status	请求报文的平均长度
request_length_sum:app_lb_id:host:status	请求报文的总长度
request_time_avg:app_lb_id:host:status	请求时间的平均值
request_time_p50:app_lb_id:host:status	请求时间的50分位值
request_time_p90:app_lb_id:host:status	请求时间的90分位值
request_time_p99:app_lb_id:host:status	请求时间的99分位值
request_time_p9999:app_lb_id:host:status	请求时间的99.99分位值
upstream_response_time_avg:app_lb_id:host:status	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:app_lb_id:host:status	请求连接时长的50分位值
upstream_response_time_p90:app_lb_id:host:status	请求连接时长的90分位值
upstream_response_time_p99:app_lb_id:host:status	请求连接时长的99分位值
write_response_time_avg:app_lb_id:host:status	Proxy写的响应时间的平均值
write_response_time_p50:app_lb_id:host:status	Proxy写的响应时间的50分位值
write_response_time_p90:app_lb_id:host:status	Proxy写的响应时间的90分位值

指标	说明
write_response_time_p99:app_lb_id:host:status	Proxy写的响应时间的99分位值

upstream_status维度

维度指标的标签为app_lb_id+host+status+request_method+upstream_status+url，指标详情如下表所示。

指标	说明
pv:app_lb_id:host:status:method:upstream_status	每个app_lb_id、host、status、method、url、upstream_status的访问次数
body_bytes_sent_avg:app_lb_id:host:status:method:upstream_status	发送给客户端的HTTP Body平均字节数
body_bytes_sent_sum:app_lb_id:host:status:method:upstream_status	发送给客户端的HTTP Body总字节数
request_length_avg:app_lb_id:host:status:method:upstream_status	的请求报文的平均长度
request_length_sum:app_lb_id:host:status:method:upstream_status	请求报文的总长度
request_time_avg:app_lb_id:host:status:method:upstream_status	请求时间的平均值
request_time_p50:app_lb_id:host:status:method:upstream_status	请求时间的50分位值
request_time_p90:app_lb_id:host:status:method:upstream_status	请求时间的90分位值
request_time_p99:app_lb_id:host:status:method:upstream_status	请求时间的99分位值
request_time_p9999:app_lb_id:host:status:method:upstream_status	请求时间的99.99分位值
upstream_response_time_avg:app_lb_id:host:status:method:upstream_status	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:app_lb_id:host:status:method:upstream_status	请求连接时长的50分位值

指标	说明
upstream_response_time_p90:app_lb_id:host:status:method:upstream_status	请求连接时长的90分位值
upstream_response_time_p99:app_lb_id:host:status:method:upstream_status	请求连接时长的99分位值
write_response_time_avg:app_lb_id:host:status:method:upstream_status	Proxy写的响应时间的平均值
write_response_time_p50:app_lb_id:host:status:method:upstream_status	Proxy写的响应时间的50分位值
write_response_time_p90:app_lb_id:host:status:method:upstream_status	Proxy写的响应时间的90分位值
write_response_time_p99:app_lb_id:host:status:method:upstream_status	Proxy写的响应时间的99分位值

8.4. 日志字段详情

本文介绍ALB负载均衡7层访问日志的字段详情。

字段	说明
__topic__	日志主题，固定为alb_layer7_access_log。
body_bytes_sent	发送给客户端的HTTP Body字节数。
client_ip	请求客户端IP地址。
host	域名或IP地址。优先从请求参数中获取host，如果获取不到则从host header取值，如果还是获取不到则以处理请求的后端服务器IP地址作为host。
http_host	请求报文host header的内容。
http_referer	URL跳转来源。
http_user_agent	客户端浏览器等信息。
http_x_forwarded_for	经过HTTP代理后的客户端IP地址。
http_x_real_ip	真实的客户端IP地址。
read_request_time	Proxy读取请求的时间，单位：毫秒。
request_length	请求的长度，包括请求行、请求头和请求正文。
request_method	请求方法。
request_time	Proxy收到第一个请求报文的时间到Proxy返回应答之间的间隔时间，单位：秒。

字段	说明
request_uri	Proxy收到的请求报文的URI。
scheme	请求的schema，包括HTTP、HTTPS。
server_protocol	Proxy收到的HTTP协议的版本，例如HTTP/1.0或HTTP/1.1。
slb_vport	负载均衡的监听端口。
app_lb_id	ALB负载均衡实例ID。
ssl_cipher	建立SSL连接使用的密码，例如ECDHE-RSA-AES128-GCM-SHA256等。
ssl_protocol	建立SSL连接使用的协议，例如TLSv1.2。
status	Proxy应答报文的状态。
tcpinfo_rtt	客户端TCP连接时间，单位：微秒。
time	日志记录时间。
upstream_addr	后端服务器的IP地址和端口。
upstream_response_time	从负载均衡向后端建立连接开始到接受完数据然后关闭连接为止的时间，单位：秒。
upstream_status	Proxy收到的后端服务器的响应状态码。
vip_addr	虚拟IP地址。
write_response_time	Proxy写的响应时间，单位：毫秒。

9.Nginx日志中心

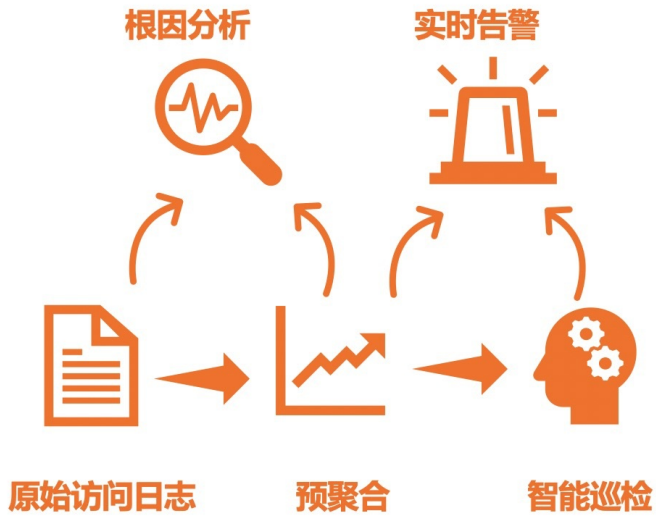
9.1. 使用前须知

阿里云日志服务Nginx日志中心提供日志分析、秒级监控指标分析、实时告警等功能，并提供基于AIOps的自动异常巡检功能。您可以通过Nginx日志中心了解客户端用户行为、客户端用户的地域分布、请求成功率、响应延迟等。本文介绍Nginx日志中心相关的功能说明、功能优势、资产说明、费用说明等信息。

功能说明

Nginx中心基于实时访问日志进行自动聚合，并提供智能巡检、实时告警等功能，详细功能如下：

- 实时访问日志的存储、查询、分析。
- 基于原始访问日志实时提取各类指标信息，包括PV、请求成功率、平均延迟、P50延迟、P90延迟、P99延迟、出入流量等。并支持多个维度组合，包括request_uri、request_method、host、status。
- 提供丰富的可视化报表，包括Nginx监控中心、异常事件、秒级监控等，并支持报表邮件、钉钉群订阅。
- 提供智能巡检功能，支持全局巡检和host粒度巡检，并支持在可视化报表中直接标注异常点。
- 自定义告警配置，告警通知直接对接消息中心、短信、邮件、语音（电话）、钉钉，并支持对接自定义WebHook。



功能优势

- 简单：一站式开通、中心化使用，无需关心日志收集、存储、计算、可视化等问题，将开发、运维人员从日志处理的繁琐耗时中解放出来，将更多的精力集中到业务开发和技术探索上去。
- 海量：访问日志数据规模很大，处理访问日志需要考虑性能和成本问题。日志中心可自定配置预聚和功能，实时计算聚合指标，计算后的聚合结果可降低几个数量级，使查询速度大大提升。
- 实时：秒级分析处理实时产生的日志。
- 弹性：可任意设置日志存储周期，Logstore容量可动态伸缩满足业务增长需求。
- 智能：基于达摩院智能AIOps算法，提供Nginx指标自动巡检功能，有助于更快、更准确的发现并定位问题。

资产说明

所有资产都在您选择的Project下，Project内的资产如下：

- Logstore
  - 访问日志Logstore用于存储Nginx访问日志，该Logstore为您自定义创建的Logstore。
  - 巡检结果Logstore用于存储巡检结果。开通日志中心功能后，自动生成该专属Logstore，其名称为*访问日志Logstore名称*-metrics-result。

    说明
    - 请勿删除Nginx访问日志相关的Logstore，否则将无法正常采集日志到日志服务。
    - 请勿删除访问日志Logstore中的部分字段的索引，否则指标转换会失败。

- Metricstore

  监控指标Metricstore用于存储聚和后的指标信息。开通日志中心功能后，自动生成该专属Metricstore，其名称为*访问日志Logstore名称*-metrics。

  说明 监控指标Metricstore存储的是聚合后的指标，数据量相比原始访问日志大大降低，非常适用于长期存储。

- 聚合规则

规则名称	聚合时间粒度	聚合维度	生成指标名
total	10秒	total	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ request_time_avg ◦ request_time_p50 ◦ request_time_p90 ◦ request_time_p99 ◦ upstream_response_time_avg ◦ upstream_response_time_p50 ◦ upstream_response_time_p90 ◦ upstream_response_time_p99

规则名称	聚合时间粒度	聚合维度	生成指标名
host	10秒	host	◦ pv:host ◦ body_bytes_sent_avg:host ◦ body_bytes_sent_sum:host ◦ request_length_avg:host ◦ request_length_sum:host ◦ upstream_response_time_avg:host ◦ upstream_response_time_p50:host ◦ upstream_response_time_p90:host ◦ upstream_response_time_p99:host ◦ request_time_avg:host ◦ request_time_p50:host ◦ request_time_p90:host ◦ request_time_p99:host
host+status	10秒	host+status	◦ pv:host:status ◦ body_bytes_sent_avg:host:status ◦ body_bytes_sent_sum:host:status ◦ request_length_avg:host:status ◦ request_length_sum:host:status ◦ request_time_avg:host:status ◦ request_time_p50:host:status ◦ request_time_p90:host:status ◦ request_time_p99:host:status ◦ upstream_response_time_avg:host:status ◦ upstream_response_time_p50:host:status ◦ upstream_response_time_p90:host:status ◦ upstream_response_time_p99:host:status

规则名称	聚合时间粒度	聚合维度	生成指标名
host+status+request_method+request_uri	10秒	host+status+request_method+request_uri	◦ pv:host:status:request_method:request_uri ◦ body_bytes_sent_avg:host:status:request_method:request_uri ◦ body_bytes_sent_sum:host:status:request_method:request_uri ◦ request_length_avg:host:status:request_method:request_uri ◦ request_length_sum:host:status:request_method:request_uri ◦ request_time_avg:host:status:request_method:request_uri ◦ request_time_p50:host:status:request_method:request_uri ◦ request_time_p90:host:status:request_method:request_uri ◦ request_time_p99:host:status:request_method:request_uri ◦ upstream_response_time_avg:host:status:request_method:request_uri ◦ upstream_response_time_p50:host:status:request_method:request_uri ◦ upstream_response_time_p90:host:status:request_method:request_uri ◦ upstream_response_time_p99:host:status:request_method:request_uri

- 巡检规则

规则名称	开启状态	巡检算法	巡检指标
total	默认开启	Time2Graph	◦ pv ◦ body_bytes_sent_avg ◦ body_bytes_sent_sum ◦ request_length_avg ◦ request_length_sum ◦ request_time_avg ◦ upstream_response_time_avg

规则名称	开启状态	巡检算法	巡检指标
host	默认开启	Time2Graph	○ pv:host ○ body_bytes_sent_avg:host ○ body_bytes_sent_sum:host ○ request_length_avg:host ○ request_length_sum:host ○ request_time_avg:host ○ upstream_response_time_avg:host

- 专属仪表盘

仪表盘名称	关联的Logstore或Metricstore	说明
Nginx监控概览	*访问日志Logstore名称*-metrics	展示总体的监控信息，包括访问PV、成功率、流量等。
Nginx访问日志分析	*访问日志Logstore名称*	展示用户请求相关的信息，包括PV、UV、移动端占比、访问地域分布等。 **说明** 此部分信息基于原始的访问日志全量计算，数据量超大的情况下会有一定延迟。
Nginx监控中心	*访问日志Logstore名称*-metrics	支持以host、request_uri、request_method、status等维度过滤出实例详细的监控信息。
蓝绿对比	*访问日志Logstore名称*-metrics	用于对两个host、uri的详细指标进行对比，通常应用于灰度发布、蓝绿发布等场景。
秒级监控	○ *访问日志Logstore名称*-metrics ○ *访问日志Logstore名称*-metrics-result	以秒级粒度展示监控信息，便于发现瞬时抖动的异常情况。
异常事件	○ *访问日志Logstore名称*-metrics ○ *访问日志Logstore名称*-metrics-result	展示流式巡检算法检测host粒度的异常信息，包括异常统计以及具体指标上异常的实时显示。

费用说明

日志服务根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费。更多信息，请参见日志服务产品定价。

9.2. 配置Nginx日志中心

本文介绍如何配置Nginx日志中心，将采集到Nginx日志进行多维度的可视化分析。

前提条件

已通过Logtail Nginx模式采集到Nginx日志。更多信息，请参见使用Nginx模式采集日志。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**Nginx日志中心**。
3. 在**巡检管理**页面，单击**添加**。
4. 在**添加日志中心**面板中，配置如下参数，并单击**确定**。

参数	说明
日志中心名称	配置日志中心名称。
项目Project	选择您已创建的Project。
日志库Logstore	选择您已创建的Logstore，该Logstore已采集到Nginx日志。

后续步骤

配置完成后，您可在Nginx日志中心查看相关的报表并进行日志的查询分析、下载、投递、加工、告警等操作。更多信息，请参见云产品日志通用操作。您还可以执行监控数据的查询分析、告警等操作。更多信息，请参见查询和分析时序数据。

9.3. 指标说明

本文介绍基于Nginx访问日志提取的指标详情，包括全局指标、host维度指标、status维度指标、host+status+request_method+request_ur维度指标。

本文涉及的指标遵循时序数据格式，支持使用PromQL或SQL进行查询分析。更多信息，请参见时序数据查询分析简介。

全局指标

全局指标信息如下表所示。

指标	说明
pv	总访问次数
body_bytes_sent_avg	发送给客户端的请求体的平均字节数
body_bytes_sent_sum	发送给客户端的请求体的总字节数
request_length_avg	请求报文的平均长度
request_length_sum	请求报文的总长度
request_time_avg	请求时间的平均值
request_time_p50	请求时间的50分位值

指标	说明
request_time_p90	请求时间的90分位值
request_time_p99	请求时间的99分位值
upstream_response_time_avg	请求连接时长的平均值 说明 upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50	请求连接时长的50分位值
upstream_response_time_p90	请求连接时长的90分位值
upstream_response_time_p99	请求连接时长的99分位值

host维度

host维度指标的标签为host，指标详情如下表所示。

指标	说明
pv:host	每个host的访问次数
body_bytes_sent_avg:host	发送给客户端的请求体的平均字节数
body_bytes_sent_sum:host	发送给客户端的请求体的总字节数
request_length_avg:host	请求报文的平均长度
request_length_sum:host	请求报文的总长度
request_time_avg:host	请求时间的平均值
request_time_p50:host	请求时间的50分位值
request_time_p90:host	请求时间的90分位值
request_time_p99:host	请求时间的99分位值
upstream_response_time_avg:host	请求连接时长的平均值 说明 upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:host	请求连接时长的50分位值

指标	说明
upstream_response_time_p90:host	请求连接时长的90分位值
upstream_response_time_p99:host	请求连接时长的99分位值

status维度

status维度指标的标签为host+status，指标详情如下表所示。

指标	说明
pv:host:status	每个host、status的访问次数
body_bytes_sent_avg:host:status	发送给客户端的请求体的平均字节数
body_bytes_sent_sum:host:status	发送给客户端的请求体的总字节数
request_length_avg:host:status	请求报文的平均长度
request_length_sum:host:status	请求报文的总长度
request_time_avg:host:status	请求时间的平均值
request_time_p50:host:status	请求时间的50分位值
request_time_p90:host:status	请求时间的90分位值
request_time_p99:host:status	请求时间的99分位值
upstream_response_time_avg:host:status	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:host:status	请求连接时长的50分位值
upstream_response_time_p90:host:status	请求连接时长的90分位值
upstream_response_time_p99:host:status	请求连接时长的99分位值

host+status+request_method+request_uri维度

host+status+request_method+request_uri维度指标的标签为host+status+request_method+request_uri，指标详情如下表所示。

指标	说明
pv:host:status:request_method:request_uri	每个host、status、request_method、request_uri的访问次数
body_bytes_sent_avg:host:status:request_method:request_uri	发送给客户端的请求体的平均字节数
body_bytes_sent_sum:host:status:request_method:request_uri	发送给客户端的请求体的总字节数
request_length_avg:host:status:request_method:request_uri	请求报文的平均长度
request_length_sum:host:status:request_method:request_uri	请求报文的总长度
request_time_avg:host:status:request_method:request_uri	请求时间的平均值
request_time_p50:host:status:request_method:request_uri	请求时间的50分位值
request_time_p90:host:status:request_method:request_uri	请求时间的90分位值
request_time_p99:host:status:request_method:request_uri	请求时间的99分位值
upstream_response_time_avg:host:status:request_method:request_uri	请求连接时长的平均值 **说明** upstream_response_time表示请求连接时长，该时长包括从负载均衡向后端建立连接开始到接收数据，然后关闭连接为止的时间。
upstream_response_time_p50:host:status:request_method:request_uri	请求连接时长的50分位值
upstream_response_time_p90:host:status:request_method:request_uri	请求连接时长的90分位值
upstream_response_time_p99:host:status:request_method:request_uri	请求连接时长的99分位值

9.4. 日志字段详情

本文介绍Nginx访问日志的字段详情。

字段	说明
__topic__	日志主题，固定为nginx_access_log。
body_bytes_sent	发送给客户端的字节数，不包括响应头的大小。
host	请求地址，IP地址或域名。
http_referer	URL跳转来源。
http_user_agent	客户端浏览器等信息。
http_x_forwarded_for	经过HTTP代理后的客户端IP地址。
remote_addr	客户端IP地址。
remote_user	客户端用户名。
request_length	请求的长度，包括请求行、请求头和请求正文。
request_method	请求方法。
request_time	整个请求的总时间，单位为秒。
request_uri	请求的URI。
status	请求状态。
time_local	服务器时间。
upstream_response_time	从负载均衡向后端建立连接开始到接受完数据然后关闭连接为止的时间，单位：秒。

10.Flowlog日志中心

10.1. 使用前须知

阿里云日志服务和专有网络联合推出Flowlog日志中心，用于VPC的策略统计、弹性网卡、流量统计以及网段间流量统计，帮助您快速、有效地分析VPC流日志。

功能说明

Flowlog日志中心包括监控中心和域间分析，详细说明如下：

- 监控中心

  监控中心用于分析与监控VPC流日志。

  - 提供概览、策略统计、ENI流量和ECS间流量仪表盘。更多信息，请参见专属仪表盘。
  - 提供自定义查询页面，便于您查询和分析VPC流日志。具体操作，请参见查询和分析日志。

- 域间分析

  开启域间分析功能后，系统将自动创建数据加工任务，生成具有网段信息的VPC流日志，用于分析不同网段之间的流量情况。

  - 提供域间流量、ECS到区间流量、威胁情报仪表盘。更多信息，请参见专属仪表盘。
  - 提供自定义查询页面，便于您查询和分析具有网段信息的VPC流日志。具体操作，请参见查询和分析日志。

资产说明

- Project和Logstore

  您需要自定义Project和Logstore，用于存储VPC流日志。当您配置域间网段后，系统将自动生成数据加工任务，并创建一个名为*自定义Logstore名-lowlogID*的Logstore，用于存储经过数据加工后的VPC流日志。

- 专属仪表盘专属仪表盘

仪表盘名称	关联的Logstore	说明
概览	自定义的Logstore	展示VPC流日志的整体信息。
策略统计	自定义的Logstore	展示Accept趋势、Reject趋势、Accept次数统计（由五元组构成）、Reject次数统计（由五元组构成）等信息。五元组是由源网段、源端口、协议类型、目标网段和目标端口组成的集合。 ○ ACCEPT：安全组和网络ACL允许记录的流量。 ○ REJECT：安全组和网络ACL拒绝记录的流量。
ENI流量	自定义的Logstore	展示弹性网卡ENI传入和传出的流量信息。
ECS间流量	自定义的Logstore	展示ECS实例之间的流量情况。
域间流量	名为*自定义Logstore名-lowlogID*的Logstore	展示不同网段之间的流量情况。

仪表盘名称	关联的Logstore	说明
ECS到区间流量	名为*自定义Logstore名-lowlogID*的Logstore	展示ECS实例到目标网段的流量情况。
威胁情报	名为*自定义Logstore名-lowlogID*的Logstore	展示源IP地址与目标IP地址的威胁情报信息。

费用说明

目前，VPC流日志仅支持将提取到的网络日志投递到日志服务，流日志的费用=网络日志提取费+日志服务的费用。

- 网络日志提取费

  VPC按照提取的日志收取网络日志提取费。

  > 说明
  > - 公测期间，免收网络日志提取费。
  > - 在VPC产品侧获取账单。

- 日志服务的费用

  日志服务采集到VPC流日志后，根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费。更多信息，请参见日志服务产品定价。

使用限制

日志服务Project与VPC实例需处于同一地域。

10.2. 配置Flowlog日志中心

日志服务支持通过Flowlog日志中心可视化分析VPC流日志。本文介绍配置Flowlog日志中心的操作步骤。

前提条件

已在VPC控制台上创建流日志。具体操作，请参见创建流日志。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**展开**，然后单击**Flowlog日志中心**。
3. 在**Flowlog管理**页面，单击**添加**。
4. 在**创建实例**面板中，配置如下参数，然后单击**确定**。

参数	说明
实例ID	日志服务默认提供实例ID。您也可以自定义实例ID。
实例名称	配置实例名称。

参数	说明
Project	选择您已创建的Project。该Project需与创建流日志中配置的Project保持一致。
Logstore	选择您已创建的Logstore。该Logstore需与创建流日志中配置的Logstore保持一致。

后续步骤

- 查看策略统计、ENI流量、ECS间流量等仪表盘。
- 执行自定义查询与分析。更多信息，请参见查询和分析日志。
- 开启域间分析。更多信息，请参见开启域间分析。

10.3. 开启域间分析

开启域间分析后，系统将自动创建数据加工任务，生成具有网段信息的VPC流日志，用于分析不同网段之间的流量情况。

前提条件

已配置Flowlog日志中心。具体操作，请参见配置Flowlog日志中心。

背景信息

日志服务已预设多个网段，如下图所示。当您需要分析不同网段之间的流量情况时，只需一键开启域间分析功能即可。

如果预设网段未满足您的需求，您可以自定义添加网段。

网段名称	IP段
私有网络	10.0.0.0/8,172.16.0.0/12,192.168.0.0/16
阿里云内部云服务	100.64.0.0/10
其他	

开启域间分析

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**展开**，然后单击**Flowlog日志中心**。
3. 在**Flowlog日志中心列表**区域，单击目标实例。
4. 在左侧导航栏中，单击**网段设置**。
5. 如果预设网段未满足您的需求，您可以自定义添加网段。
    i. 在**网段设置**页面，单击**添加**。

ii. 在**网络设置-添加**面板中，配置如下参数，然后单击**确定**。

参数	说明
网段名称	自定义网段名称。
IP段	配置您要分析的IP网段。多个网段之间使用半角逗号（,）分隔。例如： ■ 单个网段：192.168.0.0/16 ■ 多个网段：192.168.0.0/16,10.0.0.0/8
备注	添加备注信息。

6. 在**网段设置**页面，单击**开启域间分析**。
7. 如果您还未完成云资源访问授权，请根据页面提示，完成授权。

   此处必须使用阿里云账号完成AliyunLogETLRole授权。授权完成后，日志服务将使用AliyunLogETLRole来读取源Logstore中的数据以及将数据加工结果写入目标Logstore。

   如果您要使用RAM用户操作，则阿里云账号完成AliyunLogETLRole授权后，还需要授予RAM用户数据加工操作权限。更多信息，请参见授予RAM用户数据加工操作权限。

后续步骤

- 查看域间流量、ECS到区间流量、威胁情报等仪表盘。
- 执行自定义查询与分析。更多信息，请参见查询和分析日志。

10.4. 日志字段详情

本文介绍VPC流日志的字段详情。

字段	说明
__topic__	日志主题，固定为flow_log。
version	流日志版本
vswitch-id	弹性网卡所在交换机ID
vm-id	弹性网卡绑定的云服务器ID
vpc-id	弹性网卡所在专有网络ID
account-id	阿里云账号ID
eni-id	弹性网卡ID
srcaddr	源IP地址
srcport	源端口
dstaddr	目的IP地址

字段	说明
dstport	目的端口
protocol	流量的IANA协议编号。更多信息，请参见Internet 协议编号。
direction	流量方向。包括： • in：入方向流量。 • out：出方向流量。
packets	数据包数量
bytes	数据包大小
start	捕捉窗口开始时间
end	捕捉窗口结束时间
log-status	流日志的日志记录状态。包括： • OK：数据记录正常。 • NODATA：捕获窗口中没有传入或传出网络接口的网络流量。 • SKIPDATA：捕获窗口中跳过了一些流日志记录。
action	与流量关联的操作。包括： • ACCEPT：安全组和网络ACL允许记录的流量。 • REJECT：安全组和网络ACL拒绝记录的流量。
srctype	开启域间分析后，源IP地址所对应的网段信息。 **说明** 只有您开启域间分析后，才会有该字段。
dsttype	开启域间分析后，目标IP地址所对应的网段信息。 **说明** 只有您开启域间分析后，才会有该字段。

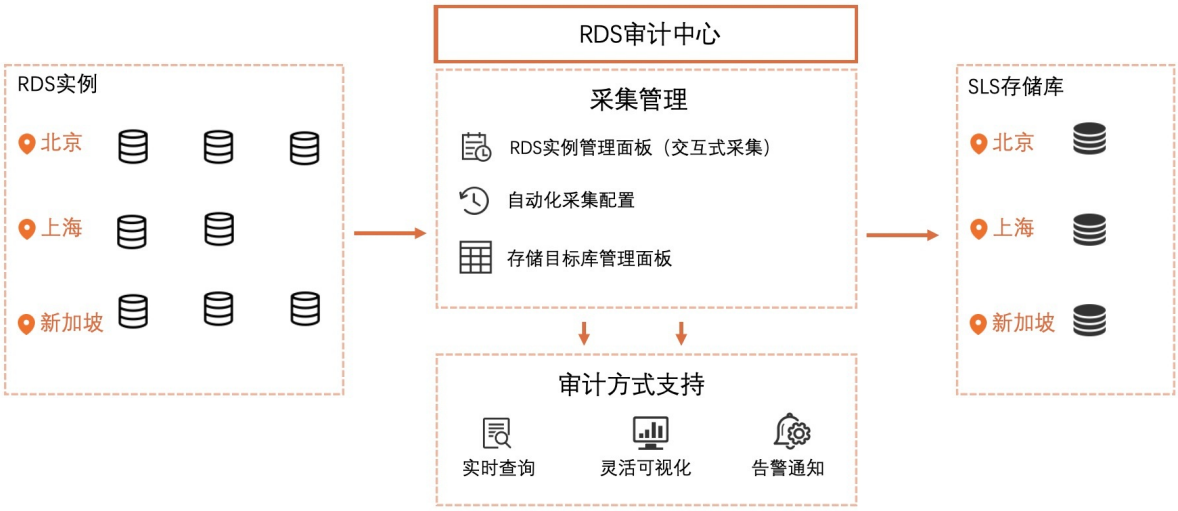
11.RDS审计中心

11.1. 使用前须知

阿里云日志服务与云数据库RDS联合推出RDS审计中心。您可以通过RDS审计中心实时查看RDS SQL审计日志的采集状态，集中管理采集配置，并可基于采集到的日志进行后续的审计、分析、告警等操作。

功能说明

RDS审计中心支持如下功能：



- 采集管理
  - 支持集中管理RDS SQL审计日志的采集状态。
  - 支持自动采集现有或未来新增RDS实例的SQL审计日志。
  - 支持集中管理存储目标库（Project、Logstore）。
- 日志审计
  - 提供RDS SQL审计日志的实时存储、查询与分析。
  - 提供丰富的可视化报表，支持报表邮件、钉钉群订阅。
  - 提供丰富的内置告警规则，支持灵活配置告警策略，及时精准地发送告警消息。

支持的日志类型

RDS SQL审计日志记录了对数据库执行的所有操作，这些信息是系统通过网络协议分析所得，对系统CPU消耗极低，不影响SQL执行效率。RDS SQL审计日志包括但不限于如下操作：

- 数据库的登录和退出操作。
- DDL（Data Definition Language）操作：对数据库结构定义的SQL语句，包括CREATE、ALTER DROP、TRUNCATE、COMMENT等。
- DML（Data Manipulation Language）操作：SQL操作语句，包括SELECT、INSERT、UPDATE、DELETE等。
- 其他SQL执行操作，包括任何其他通过SQL执行的控制，例如回滚、控制等。
- SQL执行的延迟、执行结果、影响的行数等信息。

资产详情

- 自定义日志服务Project和Logstore

  > **注意** 请勿删除RDS SQL审计日志对应的日志服务Project和Logstore，否则将无法正常推送日志到日志服务。

- 专属仪表盘

  默认生成3个仪表盘。

  > **说明** 专属仪表盘可能随时进行升级与更新，建议您不要修改专属仪表盘。您可以自定义仪表盘用于查询结果展示。更多信息，请参见创建仪表盘。

仪表盘	说明
RDS审计运营中心	展示整体访问情况、活跃数据库等信息，包括操作的数据库数量、操作表格数、执行错误、累计插入行数、累计更新行数、累计删除行数、累计查询行数等。
RDS审计性能中心	展示运维可靠性相关指标，包括SQL执行峰值、查询带宽峰值、插入开端峰值、更新带宽峰值、删除带宽峰值、SQL平均时间、查询SQL平均时间、更新SQL平均时间、删除SQL平均时间等。
RDS审计安全中心	展示数据库安全相关指标，包括错误数、登录失败次数、大批量删除事件、大批量修改事件数、危险SQL执行次数、错误操作类型分布、出错客户端外网分布、错误最多的客户端等。

费用说明

- RDS审计中心中的日志采集功能依赖于RDS实例的SQL审计（PostgreSQL、SQL Server）或SQL洞察（MySQL）功能。SQL审计或SQL洞察功能，在RDS产品侧产生相关费用。更多信息，请参见价格、收费项与计费方式。

  > **说明** 三节点企业版（原金融版）实例的SQL洞察功能免费。

- 采集RDS SQL审计日志到日志服务后，日志服务根据存储空间、读取流量、请求数量、数据加工、数据投递等进行收费，费用说明请参见按量付费。

限制说明

- 目前支持投递RDS SQL审计日志到日志服务的RDS类型如下所示。
  - MySQL：基础版本不支持，其他在售版本均支持。
  - PostgreSQL、SQL Server：所有在售版本均支持。
- RDS审计中心中的日志采集功能依赖于RDS实例的SQL审计（PostgreSQL、SQL Server）或或SQL洞察（MySQL）功能。

  在RDS审计中心开启RDS SQL审计日志采集功能后，系统自动开启对应RDS实例的SQL审计（PostgreSQL、SQL Server）或SQL洞察（MySQL）功能。
- RDS实例和日志服务Project需处于同一地域。
- 除本地云以外的其他地域都支持。

RDS审计日志采集方式比较？

目前，日志服务支持通过如下三种方式采集RDS SQL审计日志。

> **说明** RDS审计中心方式和接入数据-RDS审计方式中的采集配置是互通的。日志审计服务中的RDS SQL审计日志采集配置为独立的采集渠道，不受另外两种采集方式影响。

- RDS审计中心
  - 入口：在日志服务控制台首页的**日志应用**区域，单击**RDS审计中心**。
  - 推荐场景：建议在单账号采集场景下使用。
- 日志审计服务
  - 入口：在日志服务控制台首页的**日志应用**区域，单击**日志审计服务**。
  - 推荐场景：建议在跨账号、跨地域采集场景下使用。
- 接入数据-RDS审计
  - 入口：在日志服务控制台首页的**接入数据**区域，单击**RDS审计**。
  - 推荐场景：无，可由RDS审计中心代替。

属性	接入数据-RDS审计	RDS审计中心	日志审计服务
指定RDS实例粒度	支持	支持	支持
灵活指定存储目标库	支持	支持	不支持
跨地域采集	不支持	不支持	支持
跨账号采集	不支持	不支持	支持
自动采集	不支持	支持	支持
手动采集	支持	支持	不支持
查看采集状态视图	不支持	支持	不支持

11.2. 授予RAM用户操作权限

本文介绍如何授予阿里云RAM用户操作RDS审计中心的权限。

前提条件

已创建RAM用户。具体操作，请参见创建RAM用户。

背景信息

您可以通过如下两种方式给RAM用户授予RDS审计中心的操作权限。

- 极简授权：权限较大，操作简单。
- 自定义权限策略：权限精细，配置复杂。

极简授权

使用阿里云账号登录RAM控制台，为RAM用户授予全部管理权限（AliyunLogFullAccess、AliyunRAMFullAccess）。具体操作，请参见为RAM用户授权。

自定义权限策略

1. 使用阿里云账号登录RAM控制台。
2. 创建权限策略。
   i. 在左侧导航栏中，选择**权限管理 > 权限策略管理**。
   ii. 单击**创建权限策略**。
   iii. 在**新建自定义权限策略**页面中，配置如下参数，并单击**确定**。

参数	说明
策略名称	配置策略名称。
配置模式	选择**脚本配置**。
	将配置框中的原有脚本替换为如下内容。 您可以授予RAM用户使用RDS审计中心的只读权限或读写权限，具体权限策略说明如下： ■ 只读权限（只允许查看RDS审计中心中的各个页面。） { "Version": "1", "Statement": [{ "Action": ["rds:DescribeSqlLogInstances", "rds:DisableSqlLogDistribution"], "Resource": "*", "Effect": "Allow" }, { "Effect": "Allow", "Action": ["log:CreateLogStore", "log:CreateIndex", "log:UpdateIndex", "log:ListLogStores", "log:GetLogStore", "log:GetLogStoreLogs", "log:CreateDashboard", "log:CreateChart", "log:UpdateDashboard"], "Resource": ["acs:log:*:*:project/sls-alert-*/logstore/*", "acs:log:*:*:project/sls-alert-*/dashboard/*"] }, {

参数	说明

策略内容	"Effect": "Allow",
	"Action": [
	"log:CreateProject"
	],
	"Resource": [
	"acs:log:*:*:project/sls-alert-*"
	]
	},
	{
	"Effect": "Allow",
	"Action": [
	"log:GetLogStore",
	"log:ListLogStores",
	"log:GetIndex",
	"log:GetLogStoreHistogram",
	"log:GetLogStoreLogs",
	"log:GetDashboard",
	"log:ListDashboard",
	"log:ListSavedSearch",
	"log:GetProjectLogs"
	],
	"Resource": [
	"acs:log:*:*:project/*/logstore/*",
	"acs:log:*:*:project/*/dashboard/*",
	"acs:log:*:*:project/*/savedsearch/*"
	]
	},
	{
	"Action": [
	"ram:GetRole"
	],
	"Resource": "acs:ram:*:*:role/aliyunlogarchiverole",
	"Effect": "Allow"
	}
	]
	}
	■ 读写权限（允许操作RDS审计中心中的各个功能。）
	{
	"Version": "1",
	"Statement": [
	{
	"Action": [
	"rds:DescribeSqlLogInstances",
	"rds:DisableSqlLogDistribution",
	"rds:DisableSqlLogDistribution",
	"rds:EnableSqlLogDistribution",
	"rds:ModifySQLCollectorPolicy"
	],
	"Resource": "*",
	"Effect": "Allow"
	},
	{
	"Effect": "Allow",

参数	说明
	"Action": ["log:CreateLogStore", "log:CreateIndex", "log:UpdateIndex", "log:ListLogStores", "log:GetLogStore", "log:GetLogStoreLogs", "log:CreateDashboard", "log:CreateChart", "log:UpdateDashboard"], "Resource": ["acs:log:*:*:project/sls-alert-*/logstore/*", "acs:log:*:*:project/sls-alert-*/dashboard/*"] }, { "Effect": "Allow", "Action": ["log:CreateProject"], "Resource": ["acs:log:*:*:project/sls-alert-*"] }, { "Effect": "Allow", "Action": ["log:GetLogStore", "log:ListLogStores", "log:GetIndex", "log:GetLogStoreHistogram", "log:GetLogStoreLogs", "log:GetDashboard", "log:ListDashboard", "log:ListSavedSearch", "log:CreateLogStore", "log:CreateIndex", "log:UpdateIndex", "log:ListLogStores", "log:GetLogStore", "log:GetLogStoreLogs", "log:CreateDashboard", "log:CreateChart", "log:UpdateDashboard", "log:UpdateLogStore", "log:GetProjectLogs"], "Resource": ["acs:log:*:*:project/*/logstore/*", "acs:log:*:*:project/*/dashboard/*", "acs:log:*:*:project/*/savedsearch/*"] },

参数	说明

	{
	"Action": [
	"log:SetGeneralDataAccessConfig"
	],
	"Resource": [
	"acs:log:*:*:resource/sls.general_data_access.rds.global_conf.single_account_channel/record"
	],
	"Effect": "Allow"
	},
	{
	"Action": "ram:CreateServiceLinkedRole",
	"Resource": "*",
	"Effect": "Allow",
	"Condition": {
	"StringEquals": {
	"ram:ServiceName": "audit.log.aliyuncs.com"
	}
	}
	},
	{
	"Action": [
	"ram:*"
	],
	"Resource": [
	"acs:ram:*:*:role/aliyunlogarchiverole",
	"acs:ram:*:*:policy/AliyunLogArchiveRolePolicy"
	],
	"Effect": "Allow"
	}
	]
	}

3. 为RAM用户授权。
   i. 在左侧导航栏中，选择**身份管理 > 用户**。
   ii. 找到目标RAM用户，单击**添加权限**。
   iii. 在**添加权限**面板的**选择权限**区域，单击**自定义策略**，选中步骤中创建的策略。
   iv. 单击**确定**。

11.3. 开启日志采集功能

RDS审计中心支持手动开启采集功能和自动化采集功能。手动开启采集功能针对单个RDS实例，自动化采集功能支持多个RDS实例，自动采集符合条件的RDS实例（包括未来创建的）的审计日志。本文介绍开启采集功能的操作步骤及相关操作。

前提条件

- 如果是手动开启采集功能，则需要在RDS实例所在地域创建日志服务Project和Logstore。具体操作，请参见创建Project和Logstore。
- 如果您使用的是RAM用户，则需要先授予RAM用户RDS审计中心操作权限。具体操作，请参见授予RAM用

户操作权限。

首次配置

> 注意
> - 执行该操作的账号具备AliyunRamFullAccess权限。
> - 本操作只需执行一次。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**RDS审计中心**。
3. 根据页面提示，完成AliyunLogArchiveRole角色授权。

   完成此操作后，阿里云自动为您创建一个系统角色AliyunLogArchiveRole，并授予RDS审计中心使用该角色访问其他云产品中的资源。

4. 根据页面提示，完成AliyunServiceRoleForSLSAudit角色授权。

   完成此操作后，阿里云自动为您创建一个服务关联角色AliyunServiceRoleForSLSAuditRDS，并授予RDS审计中心使用该角色采集RDS审计日志。更多信息，请参见管理服务关联角色AliyunServiceRoleForSLSAudit。

> 注意 RDS审计中心和日志审计服务都需使用服务关联角色AliyunServiceRoleForSLSAudit进行日志采集，如果您已在日志审计服务中执行此操作，则无需在RAN审计中心中再次执行。

手动开启采集功能

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**RDS审计中心**。
3. 在**数据接入**页签中，单击目标RDS实例对应的**开启**。
4. 在**选择投递目标**对话框中，选择目标Project和Logstore，然后单击**确认**。

开启采集功能后，日志服务开始采集目标RDS实例的审计日志。

设置自动化采集

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**RDS审计中心**。
3. 在**数据接入**页签中，单击**自动化采集配置**。
4. 单击图标。
5. 设置采集条件。

   您可以使用阿里云账号ID、地域、实例ID、实例名、DB类型、DB版本号、标签等属性设置采集条件。

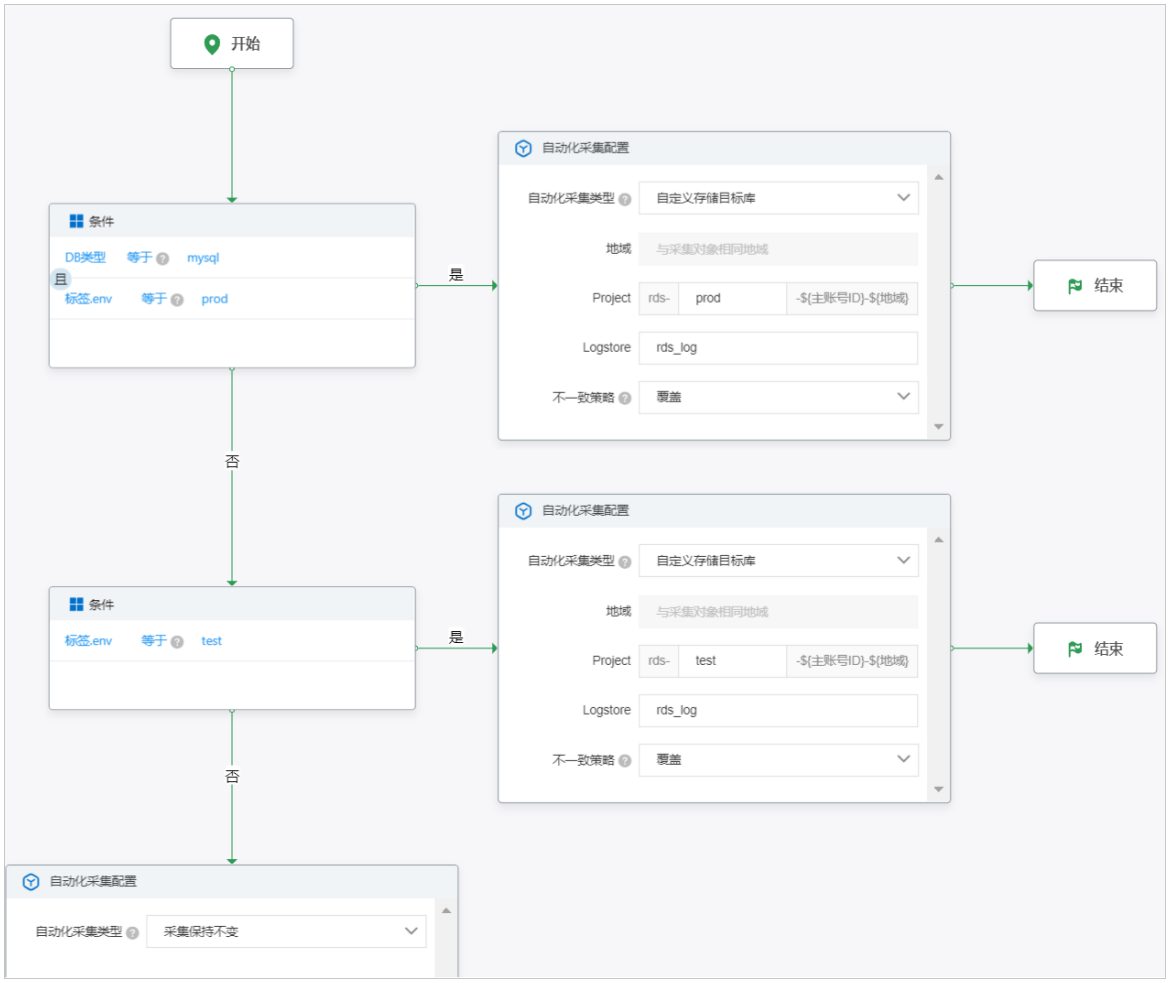
   标准模式下各个条件之间为且关系。高级模式下，您可以灵活组合与嵌套条件。
6. 设置自动化采集配置。

参数	说明
自动化采集类型	选择自动化采集类型，具体说明如下： ○ **自定义存储目标库**：自动采集符合条件的RDS实例的审计日志到目标Logstore中。 如果存储目标库（Project、Logstore）不存在，会自动创建对应的日志库目标。 ○ **采集保持不变**：选择**采集保持不变**时，无需设置**地域**、**Project**、**Logstore**和**不一致策略**参数。 ■ 符合条件的RDS实例，如果未开启采集，则不会自动开启。 ■ 符合条件的RDS实例，如果已开启采集，则不会改变其目标日志库。
地域	系统自动默认选择目标RDS实例所在地域，无法修改。
Project	在RDS实例所在地域，自动创建一个名为`rds-xxx-${主账号ID}-${地域}`的Project。例如rds-test-12345674523-cn-hangzhou。
Logstore	在名为`rds-xxx-${主账号ID}-${地域}`的Project下，自动创建一个名为`rds_log`的Logstore。

参数	说明
不一致策略	当此次设置的存储目标库与当前已生效的存储目标库不一致时，系统将根据如下选择进行判断，具体说明如下： ○ **忽略**：以当前已生效的存储目标库为准。 ○ **覆盖**：以此次设置的存储目标库为准。

例如：

- 绑定 `env==prod` 标签的RDS MySQL实例的审计日志投递到名为 `rds-prod-${主账号ID}-${地域}` 的Project下的 `rds_log` Logstore中。
- 绑定 `env==test` 标签的RDS MySQL实例的审计日志投递到名为 `rds-test-${主账号ID}-${地域}` 的Project下的 `rds_log` Logstore中。
- 其他RDS实例的审计日志的存储目标库以当前已生效的存储目标库为准。



7. 单击图标。
8. 在页面右上角，单击**保存**。

相关操作

操作	说明
管理RDS实例	您可以在**数据接入**页签的**RDS实例**区域，查看您阿里云账号所拥有的所有RDS实例、RDS所在地域、RDS实例的采集状态等。
关闭采集功能	您可以在**数据接入**页签的**RDS实例**区域，单击目标RDS实例对应的**关闭**，关闭采集功能。
修改存储目标库（Project、Logstore）	您可以在**数据接入**页签的**RDS实例**区域，单击目标RDS实例对应的**变更**，修改该RDS实例的审计日志所要投递的Project和Logstore。
管理存储目标库（Project、Logstore）	您可以在**数据接入**页签的**存储目标库**区域，查看用于存储RDS审计日志的Logstore、修改目标logstore中数据的保存时长。

后续步骤

采集到RDS审计日志后，您可以执行如下操作：

- 在**查询**页签中，选择目标Logstore，执行查询和分析操作。更多信息，请参见查询和分析日志。
- 在**审计运营中心**页签、**审计安全中心**页签或**审计性能中心**页签中，选择目标Logstore，查看对应的仪表盘。

11.4. 设置告警

RDS审计中心已内置告警规则，您开启对应的告警实例即可实时监控RDS审计中心。本文介绍设置告警的相关操作。

前提条件

已完成数据接入配置。具体操作，请参见开启日志采集功能。

背景信息

RDS审计中心中已内置告警规则、SLS审计内置告警策略、SLS审计内置行动策略、SLS审计内置用户组和SLS审计内置内容模板。它们之间的关联如下：

- 通过告警规则指定SLS审计内置告警策略。

  > **说明** RDS审计中心中的告警规则已绑定SLS审计内置告警策略，无法解绑和更换绑定。

- 通过SLS审计内置告警策略指定SLS审计内置行动策略。
- 通过SLS审计内置行动策略指定SLS审计内置用户组和SLS审计内置内容模板。

您可以直接使用内置的告警资源，也可以自定义告警资源，本文以使用内置告警资源为例。自定义告警资源的操作，请参见日志审计服务。

步骤一：创建用户

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**RDS审计中心**。
3. 在左侧导航栏中，单击**告警**。
4. 在**告警**页签中，选择**告警管理 > 用户管理**。
5. 创建用户。

具体操作，请参见创建用户。

步骤二：将用户添加到SLS审计内置用户组

1. 在**告警**页签中，选择**告警管理 > 用户组管理**
2. 在用户组列表中，单击**SLS审计内置用户组**对应的**修改**。
3. 在**修改用户组**中，将已创建的用户从**待添加成员**区域添加到**已添加成员**区域，然后单击**确认**。

步骤三：开启告警实例

1. 在**告警**页签中，单击**规则/事务**。
2. 在告警规则列表中，找到目标告警规则，单击**开启**。

   开启告警实例后，日志服务开始实时监控RDS审计中心。如果您需要开启多个告警实例，可单击**添加**。

   告警规则的参数说明请参见RDS安全。

相关操作

操作	说明
设置白名单	针对特定告警规则，如果您希望某些用户（或者实例ID、IP地址）进行操作时不触发告警，可将其设置为白名单。 不同告警规则对应的白名单配置不同。更多信息，请参见RDS安全。
关闭告警实例	关闭告警实例，告警规则不会再触发告警，**状态**变更为**未开启**。 该操作不会删除规则参数中已设置的信息。需要再次监控时，无需重新设置规则参数。
临时关闭告警实例	临时关闭告警实例后，在指定时间内不再触发告警。
恢复告警实例	处于临时关闭状态的监控实例，可随时恢复告警。
删除告警实例	删除告警实例，**状态**变更为**未创建**。 该操作会删除规则参数中已设置的信息（例如阿里云账号）。需要再次监控时，需要重新设置规则参数。
升级告警实例	当日志服务对告警规则进行较大的功能升级或升级后需要您额外配置时，系统会提示您升级告警规则。一般情况下，系统会自动完成升级。
手动初始化告警	如果误删除告警初始化产生的资产或者发生首次初始化告警资产失败的情况，可通过此操作强制重新初始化告警相关内容。

11.5. 日志字段详情

本文介绍RDS SQL审计日志字段详情。

字段名称	说明
__topic__	日志主题，固定为rds_audit_log。

字段名称	说明
instance_id	RDS实例ID。
check_rows	扫描的行数。
db	数据库名。
fail	SQL执行是否出错。 • 0：成功 • 1：失败
client_ip	访问RDS实例的客户端IP地址。
latency	执行SQL操作后，多久返回结果，单位：微秒。
origin_time	执行操作的时间点。
return_rows	返回的行数。
sql	执行的SQL语句。
thread_id	线程ID。
user	执行操作的用户名。
update_rows	更新的行数。

12.移动运维监控

12.1. 移动运维监控概述

日志服务移动运维监控用于实时监控移动应用、前端页面、小程序的运行，并且支持智能分析，帮助您低成本、高效率地发现各类隐患。

> **注意** 目前仅限白名单用户使用移动运维监控。如果您需要使用移动运维监控，请提交提交工单申请。

功能说明

日志服务移动运维监控提供如下功能：

功能	说明
数据接入	• 接入Android应用监控数据到移动运维监控服务。更多信息，请参见接入Android App监控数据。 • 接入iOS应用监控数据到移动运维监控服务。更多信息，请参见接入iOS App监控数据。 • 接入前端监控数据到移动运维监控服务。更多信息，请参见接入前端监控数据。 • 接入小程序监控数据到移动运维监控服务。更多信息，请参见接入小程序监控数据。
移动监控	• 实时大盘：实时更新并展示最近一小时或今天App异常（崩溃、ANR）的关键指标数据。更多信息，请参见实时大盘。 • 历史趋势：展示App异常（崩溃、ANR）的关键指标的历史数据。更多信息，请参见历史趋势。 • 崩溃分析：展示崩溃相关的用户影响趋势、异常趋势、异常问题等信息。更多信息，请参见崩溃分析。 • ANR分析：展示ANR相关的用户影响趋势、异常趋势、异常问题等信息。更多信息，请参见ANR分析。 • 高级查询：适用于查询条件复杂的分析场景，您可以自定义组合多个查询条件。更多信息，请参见高级查询。 • 自定义查询：日志服务提供专属Logstore，用于存储经过数据加工后的数据。您可以在该Logstore中执行查询和分析操作。更多信息，请参见自定义查询。 • 版本管理：当您的项目代码需要做打包混淆时，需要上传符号表，进行版本管理。更多信息，请参见版本管理。

功能	说明
前端监控	• 实时大盘：实时更新并展示最近一小时或今天访问页面的关键指标数据。更多信息，请参见实时大盘。 • JS异常：展示JS异常次数、异常次数PV比、影响用户、异常问题等信息。更多信息，请参见JS异常。 • API请求：展示成功率、成功请求的耗时、失败请求的耗时以及失败请求的影响用户数等信息。更多信息，请参见API请求。 • 页面性能：展示首字节、DOM Ready、页面完全加载、采样PV、2s快开比等信息。更多信息，请参见页面性能。 • 资源异常：展示失败资源数、异常次数PV比、资源异常所影响的用户数、异常域名等信息。更多信息，请参见资源异常。 • 页面访问：展示当前站点的访问次数、用户数分布等信息。更多信息，请参见页面访问。 • 自定义查询：日志服务提供专属Logstore，用于存储经过数据加工后的数据。您可以在该Logstore中执行查询和分析操作。更多信息，请参见自定义查询。
小程序监控	• 实时大盘：实时更新并展示最近一小时或今天小程序的关键指标数据。更多信息，请参见实时大盘。 • JS异常：展示JS异常次数、异常次数PV比、影响用户、异常问题等信息。更多信息，请参见JS异常。 • API请求：展示成功率、成功请求的耗时、失败请求的耗时以及失败请求的影响用户数等信息。更多信息，请参见API请求。 • 页面性能：展示onReady、业务可用、采样PV等信息。更多信息，请参见页面性能。 • 启动性能：展示onLand、onReady、业务可用、采样PV等信息。更多信息，请参见启动性能。 • 页面访问：展示当前小程序的访问次数、用户数分布等信息。更多信息，请参见页面访问。 • 自定义查询：日志服务提供专属Logstore，用于存储经过数据加工后的数据。您可以在该Logstore中执行查询和分析操作。更多信息，请参见自定义查询。

资产说明

所有资产都在您选择的Project下，Project下的Logstore说明如下：

- sls-alysls-track-base：移动端、前端页面、小程序上报的所有数据存储到该Logstore中，经数据加工后分别存储到目标Logstore中。
- sls-alysls-track-android：Android端上报的数据，经过日志服务加工功能的富化处理后存储到该Logstore中。
- sls-alysls-track-ios：iOS端上报的数据，经过日志服务加工功能的富化处理后存储到该Logstore中。
- sls-alysls-track-stat：Android、iOS端上报的崩溃数据存储到该Logstore中。
- sls-alysls-track-h5：
  - 小程序、前端页面上报的崩溃数据存储在该Logstore中。
  - 前端页面、小程序上报的数据，经过日志服务加工功能的富化处理后存储到该Logstore中。
- alysls-etl-mobile-track：移动端、前端页面、小程序上报数据到日志服务后，日志服务默认生成一个加

工任务，对数据进行富化。该Logstore用于存储数据加工过程所涉及的数据。

费用说明

移动运维监控目前处于公测阶段，移动运维监控应用本身免费。移动运维监控应用所涉及的数据存储、索引、加工等操作，按照正常收费。更多信息，请参见计费项。

12.2. 添加应用

移动运维监控应用用于管理已接入的监控数据。本文介绍添加应用的操作步骤。

前提条件

已创建Project。具体操作，请参见创建Project。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 单击**添加应用**。
4. 在**添加应用**面板中，配置如下参数，然后单击**确定**。

参数	描述
应用名称	移动运维监控应用的名称。
应用描述	移动运维监控应用的描述信息。
项目Project	用于存储监控数据的Project。
地域	显示您所选择的Project的所在地域。
选择类型	监控数据的来源。
角色权限	日志服务使用AliyunLogETLRole访问Logstore中的数据。 如果您使用的阿里云账号还未完成AliyunLogETLRole授权，请单击**您需要授权系统角色AliyunLogETLRole**，根据页面提示完成授权。 如果您要使用RAM用户操作，则使用阿里云账号完成AliyunLogETLRole授权后，还需要使用阿里云账号授予RAM用户数据加工操作权限。更多信息，请参见授予RAM用户数据加工操作权限。

后续步骤

接入数据：

- 接入Android App监控数据
- 接入iOS App监控数据
- 接入前端监控数据
- 接入小程序监控数据

12.3. 数据接入

12.3.1. 接入Android App监控数据

本文介绍如何通过Android SDK接入Android应用数据到日志服务移动运维监控。移动运维监控用于实时监控App崩溃、ANR等问题，并且支持智能分析，帮助您低成本、高效率地发现App应用中的各类隐患。

前提条件

已创建移动监控应用。具体操作，请参见添加应用。

步骤一：引入库文件

您需要添加如下依赖。

```
dependencies {
  // Gradle 3.0之后版本，请使用implementation。
  // 指定最新的SDK版本号。
  compile 'com.aliyun.openservices:aliyun-log-android-sdk:bricks_1.0.1'
  compile 'com.aliyun.openservices:alysls-android-crashreporter:1.1.1'
}
```

接入Android应用数据所涉及的依赖包说明如下表所示。

库文件	说明
aliyun-log-android-sdk	核心SDK，用于采集Android应用的数据到日志服务。
alysls-android-crashreporter	稳定性数据采集SDK，支持采集Java、native、ANR异常信息。 CrashReporter支持Maven中央仓库。

步骤二：配置权限

在*AndroidManifest.xml*文件中加上如下权限申明：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

步骤三：混淆配置

如果您的项目代码进行了打包混淆，则您需要进行混淆配置。在打包混淆规则中，需要保留*com.uc.crashsdk*包名下所有的类名和方法名。例如在*progaurd.cfg*文件中添加如下配置：

```
-keep class com.uc.crashsdk.** { *; }
-keep interface com.uc.crashsdk.** { *; }
-keep class com.aliyun.sls.android.producer.* { *; }
-keep interface com.aliyun.sls.android.producer.* { *; }
```

步骤四：配置接入服务

1. 添加Application类，即在*$PROJECT/app/src/main/AndroidManifest.xml*文件中增加Application类。

   例如添加MyApplication类，配置示例如下：

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.aliyun.sls.android.demo">
  ...
  <application
    android:icon="@mipmap/ic_launcher"
    ...
    android:name="com.aliyun.sls.android.demo.SLSDemoApplication"
    ...
    android:theme="@style/AppTheme">
      ...
  </application>
</manifest>
```

IDE将根据Android Studio提示，自动创建一个名为MyApplication的类添加到当前项目。

2. 在MyApplication.onCreate方法中，增加如下初始化代码。

```
public class MyApplication extends Application {
  @Override
  public void onCreate() {
    super.onCreate();
    SLSConfig config = new SLSConfig(this);
    config.debuggable = true;
    config.endpoint = endpoint;
    config.accessKeyId = accesskeyid;
    config.accessKeySecret = accesskeysecret;
    config.pluginAppId = pluginAppId;
    config.pluginLogproject = pluginLogproject;
    // 根据集成情况，自主开启日志开关。
    // config.debuggable = true;
    final SLSAdapter slsAdapter = SLSAdapter.getInstance();
    slsAdapter.addPlugin(new SLSCrashReporterPlugin());
    slsAdapter.init(config);
  }
}
```

- SLSConfig

  SLSConfig类定义了关键的配置字段。

类型	字段	说明
调试参数	debuggable	是否调试日志信息。 **说明** 发布时，建议关闭，即配置为config.debuggable = false。
配置参数	appVersion	App版本号。建议保持默认配置。
	appName	App名称。建议保持默认配置。
	endpoint	日志服务Project所属的Endpoint。如何获取，请参见服务入口。

类型	字段	说明
配置参数	accessKeyId	日志服务Project的AccessKey ID。如何获取，请参见访问密钥。
	accessKeySecret	日志服务Project的AccessKey Secret。如何获取，请参见访问密钥。
	securityToken	日志服务Project的访问秘钥Token。使用STS方式接入时，需要配置。
	pluginAppId	插件所属应用的ID，即您在日志服务移动运维监控平台上添加的应用。更多信息，请参见添加应用。
	pluginLogproject	插件所属应用所绑定的日志服务Project。更多信息，请参见添加应用。
自定义参数	channel	自定义参数，App渠道号。
	channelName	自定义参数，App渠道名称。
	userNick	自定义参数，用户昵称。
	longLoginNick	自定义参数，用户昵称，最后一次登录的用户昵称。
	userId	自定义参数，用户ID。
	longLoginUserId	自定义参数，用户ID，最后一次登录的用户ID。
	loginType	自定义参数，用户登录类型。
业务参数	slsConfig.addCustom("customKey", "customValue");	用于添加业务参数，键值对形式。 ■ customKey：参数名。 ■ customValue：参数值。

- SLSAdapter

  SLSAdapter类是插件的管理类。

方法	说明
getInstance()	返回SLSAdapter单例。
addPlugin(plugin)	增加一个插件。
init(slsConfig)	初始化所有插件。 **注意** init()方法会初始化所有的插件，请勿单独调用插件的init()方法。调用init()方法前，需先调用addPlugin()方法。

方法	说明
resetSecurityToken(accessKeyId, accessKeySecret, securityToken)	更新STS token。 **注意** resetSecurityToken()或updateConfig()需要在调用initWithSLSConfig方法后再调用。
updateConfig(slsConfig)	更新SLSConfig配置信息，仅限自定义参数。

3. 通过STS方式配置config.accessKeyId、config.accessKeySecret和config.accessKeySecret。

```
public class MyApplication extends Application {
  @Override
  public void onCreate() {
    super.onCreate();
    SLSConfig config = new SLSConfig(this);
    config.endpoint = endpoint;
    // 初始化时，传入STS AccessKey信息。
    config.accessKeyId = accesskeyid;
    config.accessKeySecret = accesskeysecret;
    config.accessKeySecret = securityToken;
    config.pluginAppId = pluginAppId;
    config.pluginLogproject = pluginLogproject;
    final SLSAdapter slsAdapter = SLSAdapter.getInstance();
    slsAdapter.addPlugin(new SLSCrashReporterPlugin());
    slsAdapter.init(config);
    // token过期后，请及时更新SLSAdapter的token。
    slsAdapter.resetSecurityToken(accesskeyid, accesskeysecret, securityToken);1
  }
}
```

步骤五：接入验证

1. 在MyApplication.onCreate方法中，配置 `config.debuggable = true` ，打开插件的日志开关。

   更多信息，请参见步骤四：配置接入服务。

2. 编写测试代码，模拟或触发移动端崩溃。

   常见的崩溃模式方式如下：

   - 空指针

     ```
     private void crashInJavaNull() {
       String nullStr = "1";
       if (nullStr.equals("1")) {
         nullStr = null;
       }
       nullStr.equals("");
     }
     ```

   - 类型转换异常

```
private void crashInJavaClassCast() {
  View view = new View(this);
  TextView text = (TextView)view;
}
```

- 越界异常

```
private void crashInJavaOutOfBounds() {
  new ArrayList<>(10).get(11);
}
```

- native crash

```
JNIBridge.nativeCrash(0, 0);
```

- native abort

```
JNIBridge.nativeCrash(2, 0);
```

- ANR

```
while (true) {
  try {
    Thread.sleep(1);
  } catch (InterruptedException e) {
    e.printStackTrace();
  }
}
```

3. 重启移动端，然后等待大概2分钟后，您可以在控制台查看是否显示崩溃信息。

   如果显示崩溃数据，则表示SDK接入成功。

后续步骤

- 实时大盘
- 历史趋势
- 崩溃分析
- ANR分析
- 高级查询
- 自定义查询

12.3.2. 接入iOS App监控数据

本文介绍如何使用Pod集成方式接入iOS应用数据到日志服务移动运维监控。移动运维监控用于实时监控App崩溃等问题，并且支持智能分析，帮助您低成本、高效率地发现App应用中的各类隐患。

前提条件

已创建移动监控应用。具体操作，请参见添加应用。

步骤一：通过CocoaPods集成

1. 在工程的Podfile中添加如下内容。

```
pod 'AliyunLogProducer', '2.3.0', :subspecs => ['Bricks', 'CrashReporter']
```

接入iOS应用数据所涉及的依赖包说明如下表所示。

库文件	说明
AliyunLogProducer	核心SDK，用于采集iOS应用的数据到日志服务。

2. 保存并执行 `pod install` 命令。
3. 使用后缀为*.xcworkspace*的文件打开工程。

步骤二：配置接入服务

1. 在工程的*AppDelegate.m*文件导入头文件。

```
#import <AliyunLogProducer/AliyunLogProducer.h>
```

2. 在*AppDelegate.m*文件的application:(UIApplication *)application didFinishLaunchingWithOptions方法中初始化SDK。

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  // Override point for customization after application launch.
  SLSConfig *config = [[SLSConfig alloc] init];
  [config setDebuggable:YES];
  [config setEndpoint:endpoint];
  [config setAccessKeyId:accessKeyId];
  [config setAccessKeySecret:accessKeySecret];
  [config setPluginAppId:pluginAppId];
  [config setPluginLogproject:pluginLogproject];
  SLSAdapter *slsAdapter = [SLSAdapter sharedInstance];
  [slsAdapter addPlugin:[[SLSCrashReporterPlugin alloc]init]];
  [slsAdapter initWithSLSConfig:config];
  return YES;
}
```

其中，SLSConfig类和SLSAdapter类说明如下：

- SLSConfig

  SLSConfig类定义了关键的配置字段。

类型	字段	说明
调试参数	debuggable	是否调试日志信息。 **说明** 发布时，建议关闭，即配置为[config setDebuggable:NO]。
配置参数	appVersion	App版本号。建议保持默认配置。
	appName	App名称。建议保持默认配置。

类型	字段	说明
配置参数	endpoint	日志服务Project所属的Endpoint。如何获取，请参见服务入口。
	accessKeyId	日志服务Project的AccessKey ID。如何获取，请参见访问密钥。
	accessKeySecret	日志服务Project的AccessKey Secret。如何获取，请参见访问密钥。
	securityToken	日志服务Project的访问秘钥Token。使用STS方式接入时，需要配置。
	pluginAppId	插件所属应用的ID，即您在日志服务移动运维监控平台上添加的应用。更多信息，请参见添加应用。
	pluginLogproject	插件所属应用所绑定的日志服务Project。更多信息，请参见添加应用。
自定义参数	channel	自定义参数，App渠道标识。
	channelName	自定义参数，App渠道名称。
	userNick	自定义参数，用户昵称。
	longLoginNick	自定义参数，用户昵称，最后一次登录的用户昵称。
	userId	自定义参数，用户ID。
	longLoginUserId	自定义参数，用户ID，最后一次登录的用户ID。
	loginType	自定义参数，用户登录类型。
业务参数	[config addCustomWithKey:@"customKey" andValue:@"testValue"];	用于添加业务参数，键值对形式。 ■ customKey：具体的参数名。 ■ testValue：具有的参数值。

- SLSAdapter

  SLSAdapter类是插件的管理类。

方法	说明
addPlugin	增加一个插件。
removePlugin	移除一个插件。

方法	说明
initWithSLSConfig	初始化所有插件。 **注意** initWithSLSConfig方法会初始化所有的插件，请勿单独调用插件的initWithSLSConfig方法。调用initWithSLSConfig方法前，需先调用addPlugin方法。
resetSecurityToken(accessKeyId, accessKeySecret, securityToken)	更新STS token。 **注意** resetSecurityToken或updateConfig需要在调用initWithSLSConfig方法后再调用。
updateConfig(slsConfig)	更新SLSConfig配置信息，仅限自定义参数。

3. 通过STS方式配置setAccessKeyId、setAccessKeySecret和setSecurityToken信息。

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
  // Override point for customization after application launch.
  SLSConfig *config = [[SLSConfig alloc] init];
  [config setEndpoint:endpoint];
  // 初始化时，传入STS AccessKey信息。
  [config setAccessKeyId:accessKeyId];
  [config setAccessKeySecret:accessKeySecret];
  [config setSecurityToken:token];
  [config setPluginAppId:pluginAppId];
  [config setPluginLogproject:pluginLogproject];
  SLSAdapter *slsAdapter = [[SLSAdapter alloc] init];
  [slsAdapter addPlugin:[[SLSCrashReporterPlugin alloc]init]];
  [slsAdapter initWithSLSConfig:config];
  // token过期后，需要及时更新SLSAdapter的token。
  [slsAdapter resetSecurityToken:accessKeyId secret:accessKeySecret token:token];
  return YES;
}
```

步骤三：接入验证

1. 编写测试代码，模拟或触发移动端崩溃。

```
[self performSelector:@selector(die_die)];
```

2. 重启移动端，然后等待大概2分钟后，您可以在控制台查看是否显示崩溃信息。

   如果显示崩溃数据，表示数据接入成功。

后续步骤

- 实时大盘

- 历史趋势
- 崩溃分析
- ANR分析
- 高级查询
- 自定义查询

12.3.3. 接入前端监控数据

本文介绍如何通过SDK接入前端监控数据到日志服务移动运维监控。移动运维监控用于实时监控JS异常、页面性能等问题，并且支持智能分析，帮助您低成本、高效率地发现前端页面中的各类隐患。

前提条件

已创建移动监控应用。具体操作，请参见添加应用。

步骤1：开通Web Tracking

1. 登录日志服务控制台。
2. 在Project列表区域，单击目标Project。
3. 在**日志存储 > 日志库**页签中，选择目标Logstore右侧的**图标 > 修改**。
4. 在**Logstore属性**页面，单击右上方的**修改**。
5. 打开**WebTracking**开关，并单击**保存**。

步骤二：安装SDK包

1. 安装依赖包。

```
npm i sls-wpk-reporter --save
```

2. 导入SDK核心包。

```
import SlsReporter from 'sls-wpk-reporter'
```

3. 导入插件安装包。

```
import {
 wpkglobalerrorPlugin, // JS异常监控
 wpkperformancePlugin, // 性能监控
} from 'sls-wpk-reporter'
```

> **说明** 当您需要添加采集插件，用于扩展SDK的打点能力时，需要导入wpkinterfacePlugin包，导入命令为 `import wpkinterfacePlugin from 'sls-wpk-reporter/src/plugins/interface'` 。详细的采集插件说明，请参见采集插件扩展说明。

步骤三：上报数据

常见的SDK Demo示例如下。

```
const wpk = new SlsReporter({
 bid: 'sls-f****bfa4573',
 project: 'my-project',
 endpoint: 'cn-hangzhou-intranet.log.aliyuncs.com',
 slsParams: {
  app_name: '',
  user_nick: '',
  logon_type: '',
 },
 rel: '1.0',
 uid: '123456',
 plugins: [
  [wpkperformancePlugin],
  [
   wpkglobalerrorPlugin,
   {
    jsErr: true, // 是否开启JS异常监控。
    resErr: true, // 是否开启资源加载异常监控。
   },
  ],
 ],
});
wpk.install(); // 初始化。
// 上报一条数据。
wpk.logReport({
 key1: '1',
 key2: '2'
})
```

参数名称	是否必填	说明
bid	是	您在日志服务移动运维监控平台上所添加的应用的ID。更多信息，请参见添加应用。
project	是	您在添加应用时所选的日志服务Project。
endpoint	是	日志服务Project所属的Endpoint。如何获取，请参见服务入口。
slsParams	是	上报扩展数据。参数值为Object类型。具体格式，请参见logdata说明。
rel	否	前端资源版本号。推荐配置。 支持字符串和函数两种方式。使用函数设置时，最终需要返回的也是字符串。
uid	否	浏览当前页面的用户的唯一标识，默认使用uuid。推荐配置。 支持字符串和函数两种方式。使用函数设置时，最终需要返回的也是字符串。

参数名称	是否必填	说明
plugins	否	采集插件。 没有设置任何插件时，默认使用wpkflowPlugin插件，用于上报站点流量数据。 **说明** 当您需要添加插件，扩展SDK打点能力时，可添加插件配置。详细的采集插件说明，请参见采集插件扩展说明。
sampleRate	否	采样率。 默认情况下，性能数据的采样率为10%，其他类型的数据采样率为100%。
beforeSend	否	用于数据上报时的前置处理，添加 `return false` 可阻止日志上报。参数值为Object类型。具体格式，请参见logdata说明。
spa	否	是否使用SPA应用。 • （默认值）false：不使用。 • true：使用SPA应用，可配合wpkflowPlugin插件更好的打点，自动监听hashchange事件。
debug	否	是否开启debug模式。 • （默认值）false：关闭。 • true：开启。开启后，可以输出更详细的打点过程日志。一般用于接入时的联调分析。 **说明** 生产环境中，建议关闭。

采集插件扩展说明

当您需要添加插件，扩展SDK打点能力时，可在SDK代码中添加如下插件配置。

插件	说明	示例

插件	说明	示例

wpkglobalerrorPlugin	用于监控全局错误。	插件配置说明如下：
		{
		jsErr: true, // 是否开启JS异常监控。
		jsErrSampleRate: 1, // JS异常数据的采样率。
		jsErrFilter: fn, // JS异常数据过滤的回调函数，参数为Error对象。如果返回false，则不上报JS异常数据。
		resErr: false, // 是否开启资源异常监控。
		resErrSampleRate: 1 // 资源异常监控采样率。
		resErrFilter: fn // 资源异常数据过滤的回调函数，参数为Error对象。如果返回false，则不上报资源异常数据。
		}

wpkinterfacePlugin	用于监控API请求异常。	插件配置说明如下：
		{
		enable: true, // 是否开启API请求监控。
		sampleRate: 1, // API请求数据的采样率。
		withBody: false, // 是否上报请求的Body参数。
		errorFilter: function(params) {
		// API请求异常数据过滤的回调函数，参数包含请求地址、响应码及响应内容。如果返回false，则不上报API请求异常数据。
		// params格式为{url: '', status: '', response: '', body: '', reqHeaders: {}, resHeaders: {},queryString: ''}
		// 返回结果为{bizCode: '', msg: '', traceId: ''}，SDK会将该结果一起上报。
		}
		}

插件	说明	示例

wpkperformancePlugin	用于自动上报性能数据。	插件配置说明如下：
		{
		enable: true, // 是否开启自动上报性能数据功能。
		sampleRate: 0.1 // 性能数据的采样率。
		}

wpkflowPlugin	用于监控站点流量。
	开启后，默认在onload阶段自动上报站点流量数据。
	说明 如果在SDK构造参数中将spa设置为true，则在hashchange阶段也会自动上报站点流量数据。
		插件配置说明如下：
		{
		enable: true // 是否开启站点流程监控。
		}

API说明

您可以在SDK中调用如下API。

API	说明
.logReport(*logdata*)	用于主动上报一条数据。配置内容，请参见logdata说明。
.setConfig(*opts*)	用于更新SDK实例的配置。配置内容，请参见采集插件扩展说明。
.install()	用于安装SDK，即初始化所有的设置，包含各个插件的初始化。 注意 调用此方法后，SDK的初始化工作才算完成，后续才能成功调用其他API。
.installAll()	用于安装SDK，即初始化所有的设置，包含内置的所有插件及安装全家桶。适合不接内核的场景。
.diagnose()	诊断API，用于调试。 调用该API后，会在浏览器控制台上输出诊断信息，并重定向到一个新页面，显示上报结果。

logdata说明

logdata为扩展数据。SDK在上报数据时，会将您自定义的扩展数据一起上报。

```
{
 category: 100, // [必填]指定监控项，0-99为系统预留。
 msg: '',    // [必填]自定义内容。
 c1: '',     // [必填]预留的扩展字段1，即上报的日志可以带上该属性用于聚合分析。
 c2: '',     // [必填]预留的扩展字段2，即上报的日志可以带上该属性用于聚合分析。
 c3: '',     // [必填]预留的扩展字段3，即上报的日志可以带上该属性用于聚合分析。
 c4: '',     // [必填]预留的扩展字段4，即上报的日志可以带上该属性用于聚合分析。
 c5: '',     // [必填]预留的扩展字段5，即上报的日志可以带上该属性用于聚合分析。
 wl_avgv1: 100, // [必填]用于监控耗时、性能等指标的均值，参数值必须为数字。
}
```

后续步骤

- 实时大盘
- JS异常
- API请求
- 页面性能
- 资源异常
- 页面访问
- 自定义查询

12.3.4. 接入小程序监控数据

日志服务支持通过SDK接入支付宝小程序、微信小程序、钉钉小程序、百度智能小程序和头条小程序的监控数据。移动运维监控用于实时监控小程序问题，并且支持智能分析，帮助您低成本、高效率地发现小程序中的各类隐患。

前提条件

已创建移动监控应用。具体操作，请参见添加应用。

步骤一：安装SDK

目前大部分小程序都支持通过npm方式安装SDK，推荐您使用npm方式。当您安装的npm依赖包无法编译时，您可以使用源码依赖方式来安装SDK。

> **说明** SDK的构建产物支持直接使用ES6语法import或require。

- （推荐）npm方式
  i. 安装依赖包。

  ```
  npm i sls-mini-app-reporter --save
  ```

  ii. 导入业务核心SDK包。

  ```
  import AliPayReporter from 'sls-mini-app-reporter/alipay'
  import WechatReporter from 'sls-mini-app-reporter/wechat'
  import DDReporter from 'sls-mini-app-reporter/dingtalk'
  import BaiduReporter from 'sls-mini-app-reporter/baidu'
  import TTReporter from 'sls-mini-app-reporter/tt'
  ```

- 源码依赖方式

  此处以微信小程序为例。

  i. 获取文件*sls-mini-app-reporter/wechat.js*和*sls-mini-app-reporter/app/wechat.js*。

  ii. 拷贝文件至项目目录中，例如*wechatDemo/utils*。

```
|- wechatDemo
|----utils
|------app
|--------wechat.js
|------wechat.js
```

步骤二：初始化SDK

1. 在*utils*目录下，新增*miniapp-reporter.js*文件。
2. 在*miniapp-reporter.js*文件中，添加如下内容。

```
// miniapp-reporter.js
import MiniAppReporter from 'sls-mini-app-reporter/alipay'
const reporter = new MiniAppReporter({
 bid: 'sls-f51****fa4573',
 project: 'my-project',
 endpoint: 'cn-hangzhou.log.aliyuncs.com',
 uid: '123456',
 rel: '1.0',
 debug: false,
 checkHidden: false,
 slsParams: {
  app_name: '',
  user_nick: '',
  logon_type: '',
 }
})
// 导出单例
export default reporter
```

参数名称	是否必填	说明
bid	是	您在日志服务移动运维监控平台上所添加的应用的ID。更多信息，请参见添加应用。
project	是	您添加应用时所选择的日志服务Project。
endpoint	是	日志服务Project所属的Endpoint。如何获取，请参见服务入口。
uid	否	浏览当前页面的用户的唯一标识，默认使用uuid。推荐配置。 支持字符串和函数两种方式。使用函数设置时，最终需要返回的也是字符串。

参数名称	是否必填	说明
rel	否	小程序版本号。推荐配置 支持字符串和函数两种方式。使用函数设置时，最终需要返回的也是字符串。
debug	否	是否开启debug模式。 ○ （默认值）false：关闭。 ○ true：开启。开启后，可以输出更详细的打点过程日志。一般用于接入时的联调分析。 说明 生产环境中，建议关闭。
checkHidden	否	是否上报onHide阶段的数据。 ○ （默认值）false：上报。 ○ true：不上报。
slsParams	否	上报扩展数据。参数值为Object类型。具体格式，请参见logdata说明。

步骤三：上报数据

常用的SDK Demo示例如下：

- 全局监听

```
// 根目录下的app.js。
import reporter from '/utils/miniapp-reporter';
App(
 reporter.wrapApp({
  onLaunch () {},
  onShow () {},
  onHide () {},
  onError (error) {}
 })
);
// reporter.wrapApp的参数与原App的参数保持一致。
// 本步骤使用reporter.wrapApp()来包裹原来App的入参，降低使用成本。
```

- 页面监听

```
// 页面的入口JS。
import reporter from '/utils/miniapp-reporter';
const app = getApp();
Page(
 reporter.wrapPage({
  data: {},
  onLoad() {},
  onShow() {},
  onHide () {},
  onUnload () {},
  onReady () {} ,
  onTodoChanged(e) {},
  addTodo() {}
 })
);
```

进阶场景

除了监控常规的PV、UV、JS异常、API请求等指标外，您还可以自定义上报功能，轻松实现各种场景的监控分析。

> **注意** 使用进阶功能时，需要先导入itrace实例。
>
> ```
> import reporter from '/utils/miniapp-reporter';
> ```

- 手动上报
  - 通用的手动上报接口report

    ```
    reporter.report(logdata) //logdata表示扩展数据。上报数据时，会将该部分数据一起上报。格式说明，请参见logdata说明。
    ```

  - SDK内置的API

API	说明

API	说明

reportJSError reporter.reportJSError(*error*error, *logdata*)	手动上报JS异常数据。
	■ error必须是一个标准的JS JSON对象。JSON对象格式说明，请参见JavaScript参考。
	■ *logdata*表示扩展数据，示例如下：
	{
	url: 'https://your.gateway.domain/api/user', // 完整请求地址，包括协议。
	method: 'GET', // HTTP请求方法。
	queryString: '', // 支持字符串或者Object格式：p1=v1&p2=v2 or { p1: v1, p2: v2 }
	headers: {}, // 请求头，Object格式。
	body: '', // 请求内容。
	response: '', // 响应内容。
	respHeaders: {}, // 响应头。
	status: 200, // HTTP响应码。
	spent: 100 // 请求RT，整数。单位：毫秒。
	}

reporter.reportApi(*logdata*)	手动上报API请求数据。

- 其他

- 设置采样率：当日志量较大时，建议设置采样率，避免日志全量上报。

```
//全量数据，默认采样率是100%。
const reporter = new MiniAppReporter({
// ...
 sampleRate: 0.5 // 所有数据都会应用此采样率。
// ...
})
//单独对JS异常数据进行采样。
const reporter = new MiniAppReporter({
// ...
 jsErrorSampleRate: 0.5 //JS异常数据应用此采样率。
// ...
})
//单独对API请求数据进行采样。
const reporter = new MiniAppReporter({
// ...
 apiSampleRate: 0.5 // API请求数据应用此采样率。
// ...
})
//单独对性能数据进行采样。
const reporter = new MiniAppReporter({
// ...
 launchPerfSampleRate: 0.5, //启动性能数据应用此采样率。
 pagePerfSampleRate: 0.5 //页面性能数据应用此采样率。
// ...
})
```

- 设置面向全量数据的过滤器。

```
const reporter = new MiniAppReporter({
// ...
 beforeSend: function (logdata) {
   // 您还可以添加自定义字段，对日志数据进行加工。
   // 添加return false，可阻止本条数据上报。
 }
// ...
})
```

- 设置面向JS异常数据的过滤器。

```
const reporter = new MiniAppReporter({
// ...
 jsErrorFilter: function (error) {
   // error为标准的JS Error对象。
   // 添加return false，可阻止本条数据上报。
 }
// ...
})
```

- 设置API请求数据相关的配置：API请求相关的SDK默认开启API请求监控功能，同时由于安全方面的考虑，SDK默认不会上报请求体、请求头、响应体以及响应头等信息。您可以修改配置，实现请求体、请求头、响应体以及响应头等数据上报。

```
const reporter = new MiniAppReporter({
// ...
 disableApi: true,      // 是否关闭API监控功能。默认为false，表示不关闭。
 apiBody: false,        // 是否上报请求体。默认为false，表示不上报。
 apiResponse: false,    // 是否上报响应体。默认为false，表示不上报。
 apiRequestHeader: false, // 是否上报请求头。默认为false，表示不上报。
 apiResponseHeader: false, // 是否上报响应头。默认为false，表示不上报。
 apiFilter: function (logdata) {
  // ...
  //添加return false，可阻止本条数据上报。
 // ...
  }
// ...
})
```

- 设置性能数据相关的配置。
  - 配置

```
const reporter = new MiniAppReporter({
// ...
 launchPerfSampleRate: 0.5, //启动性能数据应用此采样率。
 pagePerfSampleRate: 0.5, // 页面性能日志应用此采样率。
 disableLaunchPerf: true,// 是否关闭启动性能监控。
 disablePagePerf: true,// 是否关闭页面性能监控。
 //默认将在onReady钩子回调5000 ms后或onHide、onUnload钩子回调时上报性能日志。
 perfUploadAfter: 5000,
 launchPerfFilter: function (logdata) {
  // ...
  // 添加return false，可阻止本条数据上报。
 // ...
  }
 pagePerfFilter: function (logdata) {
    // 添加return false，可阻止本条数据上报。
  }
// ...
})
```

- 性能标记

```
reporter.perfMark(name, type)
```

```
setData({...}, function () {
  itrace.perfMark('bizAvailable', 'end')// 设置业务可用结束时间点。
})
itrace.perfMark('wl_avgv1', 'start')// 设置性能指标1的时间起点。
// 指标1相关代码开启执行。
...
// 指标1相关代码执行结束。
itrace.perfMark('wl_avgv1', 'end')
```

参数	说明
name	自定义指标的名称。例如： ■ bizAvailable： 业务可用。 ■ wl_avgv1 - wl_avgv5：自定义指标。
type	时间点。包括： ■ start：标记指标的起始时间点。 ■ end：标记指标的结束时间点。

logdata说明

logdata为扩展数据。SDK在上报数据时，会将您自定义的扩展数据一起上报。

```
{
 category: 100, // [必填]指定监控项，0-99为系统预留。
 msg: '',    // [必填]自定义内容。
 c1: '',     // [必填]预留的扩展字段1，即上报的日志可以带上该属性用于聚合分析。
 c2: '',     // [必填]预留的扩展字段2，即上报的日志可以带上该属性用于聚合分析。
 c3: '',     // [必填]预留的扩展字段3，即上报的日志可以带上该属性用于聚合分析。
 c4: '',     // [必填]预留的扩展字段4，即上报的日志可以带上该属性用于聚合分析。
 c5: '',     // [必填]预留的扩展字段5，即上报的日志可以带上该属性用于聚合分析。
 wl_avgv1: 100, // [必填]用于监控耗时、性能等指标的均值，参数值必须为数字。
}
```

12.4. 移动监控

12.4.1. 基本概念

本文介绍移动监控相关的基本概念。

名词	说明
应用	不同平台的App。
异常	App客户端在运行过程中发生崩溃和ANR，统称为异常。

名词	说明
崩溃	App发生崩溃。
ANR	App出现无响应现象，ANR仅适用于Android应用。
启动次数	启动App的次数。 用户启动App计为一次APP启动，退出App重新启动也计为一次启动，前后台切换不计为启动。
活跃用户	启动App的用户数量，通常为启动App的设备排重计数。
异常次数	App发生一个异常问题且被记录上报，计为一次异常。
异常率（%）	异常率=异常次数/启动次数。
影响用户	一台设备发生异常，计为一个影响用户。 在指定时间范围内，如果一个设备发生多次异常，只算一个影响用户。
影响用户率（%）	影响用户率=影响用户/活跃用户。
人均异常次数	人均异常次数=异常次数/影响用户。

12.4.2. 实时大盘

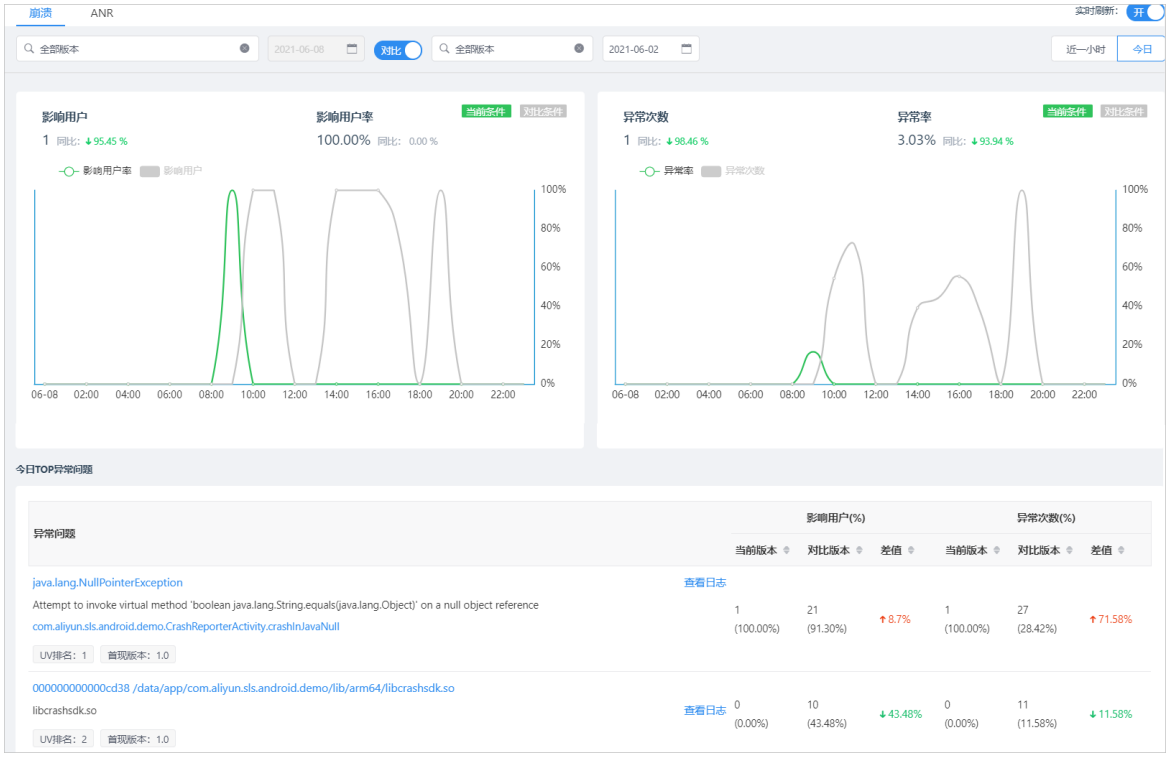
实时大盘实时更新并展示最近一小时或今天App异常（崩溃、ANR）的关键指标数据。实时大盘用于在发布版本、重要活动等关键时间点实时监控App。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**实时大盘**。
5. 在**实时大盘**页面，选择版本和时间，查看异常信息。

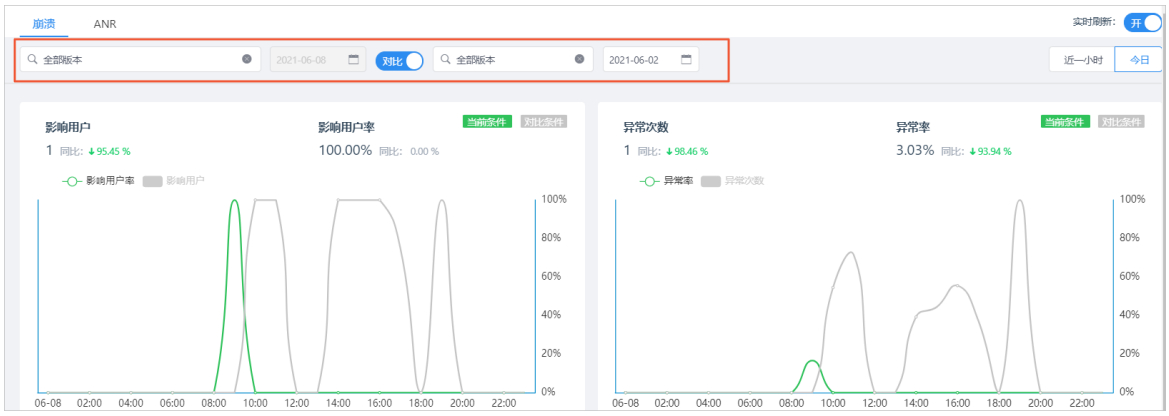


日志服务展示影响用户和用户率、异常次数和异常率、TOP异常问题列表等数据。详细的异常分析，请参见崩溃分析、ANR分析。

- 影响用户和用户率
  - 通过柱状图展示最近一小时或今天App崩溃所影响的用户数量。
  - 通过折线图展示最近一小时或今天App崩溃所影响的用户率及趋势。
- 异常次数和异常率
  - 通过柱状图展示最近一小时或今天App异常次数。
  - 通过折线图展示最近一小时或今天App异常率。
- TOP异常问题列表

  展示最近1小时或今日的TOP异常问题。

6. 如果您需要对比数据，您可以选择版本和时间后，单击**对比**。



日志服务移动运维监控支持按小时或天对比影响用户、影响用户率、异常次数和异常率。

- 选择**近一小时**，则可对比最近一小时与过去（6个月内）同一时段的数据。

- 选择**今日**，则可对比今天与过去（6个月内）某一天的数据。

12.4.3. 历史趋势

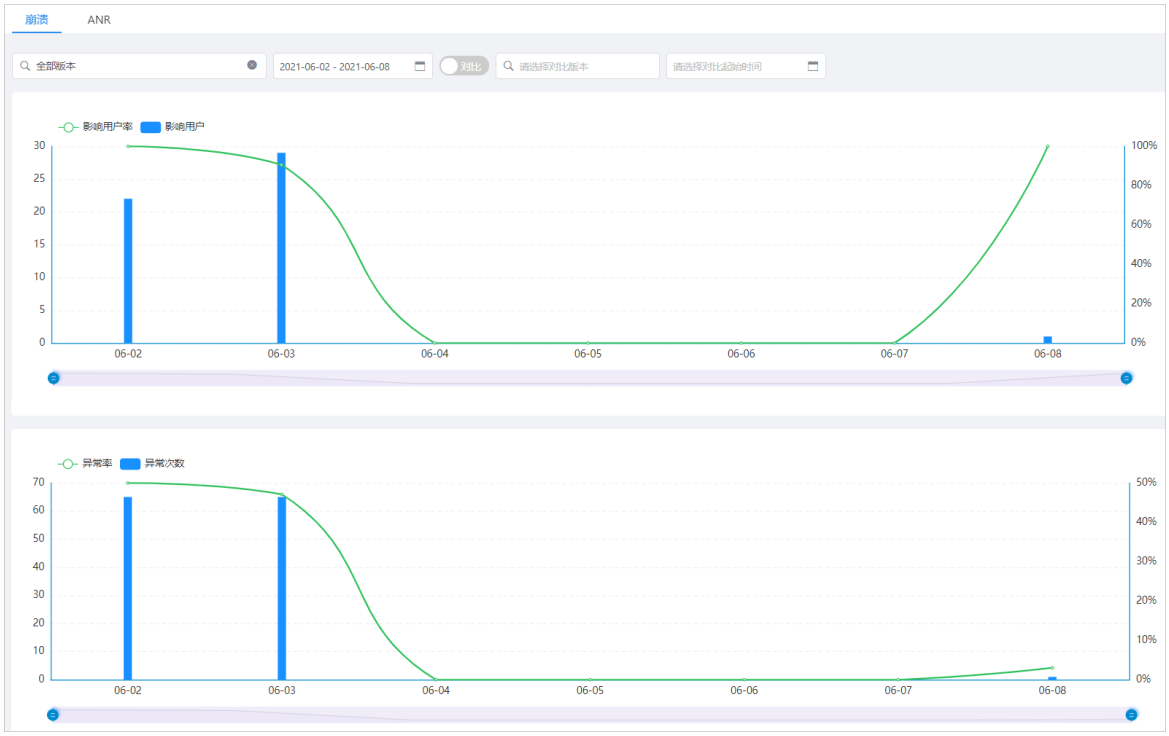
历史趋势大盘展示App异常（崩溃、ANR）的关键指标的历史数据。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

操作步骤

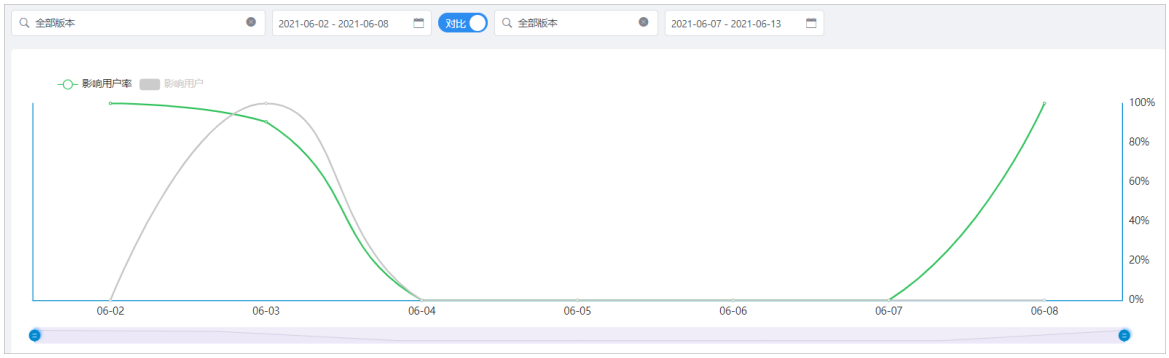
1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**历史趋势**。
5. 在**历史趋势**页面，选择版本和时间，查看异常信息。



日志服务展示影响用户和影响用户率、异常次数和异常率等数据。请参见崩溃分析、ANR分析。

- 影响用户和影响用户率
  - 通过柱状图展示指定时间范围内App崩溃或ANR所影响的用户数量。
  - 通过折线图展示指定时间范围内App崩溃或ANR所影响的用户率。
- 异常次数和异常率
  - 通过柱状图展示指定时间范围内App崩溃或ANR出现的次数。
  - 通过折线图展示指定时间范围内App崩溃或ANR出现的频率。

6. 如果您需要对比数据，您可以选择版本和时间后，单击**对比**。



日志服务移动运维监控支持对比最近92天内的异常数据。

> **说明** 对比的时间区间需一致，例如都选择一天，或者都选择一个月。

12.4.4. 崩溃分析

崩溃指标是反应用户对App质量满意度的核心指标。日志服务崩溃分析大盘展示了崩溃相关的用户影响趋势、异常趋势、异常问题等信息，帮助您快速分析出现崩溃问题所涉及的影响以及主要原因。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

功能入口

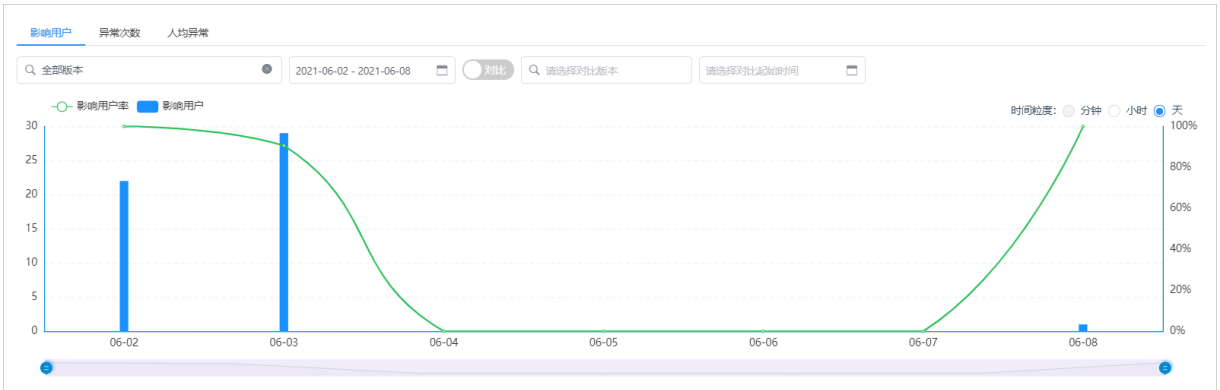
1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**崩溃分析**。

数据趋势与对比

崩溃分析大盘中，以折线图、柱状图形式展示影响用户、影响用户率、异常次数、异常率、人均异常等指标数据。

当您选择**时间粒度**为**天**时，单击折线图或状态图，系统自动跳转至**异常问题**列表区域，展示当天所涉及的异常问题。

您还可以指定版本和时间，对崩溃数据进行对比。对比的时间区间需一致，例如都选择一天，或者都选择一个月。



TOP异常问题

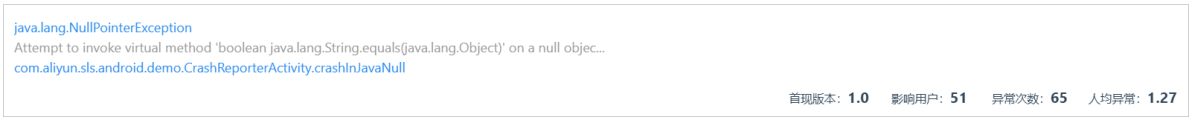
当您选择版本和时间后，系统自动展示异常问题并按照**影响用户**排序，帮助您一目了然找到TOP异常问题。

您还可以通过聚合维度、状态、责任人等条件筛选异常问题。

异常问题详情

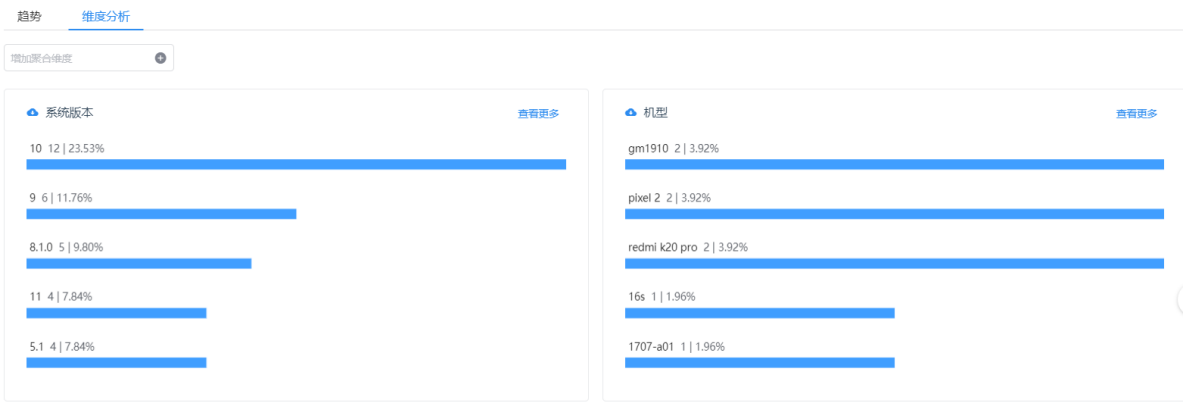
您在**异常问题**列表中，单击目标异常问题，可查询异常问题详情。

- 展示主要的堆栈信息以及App版本号、影响用户、异常次数、人均异常等信息。

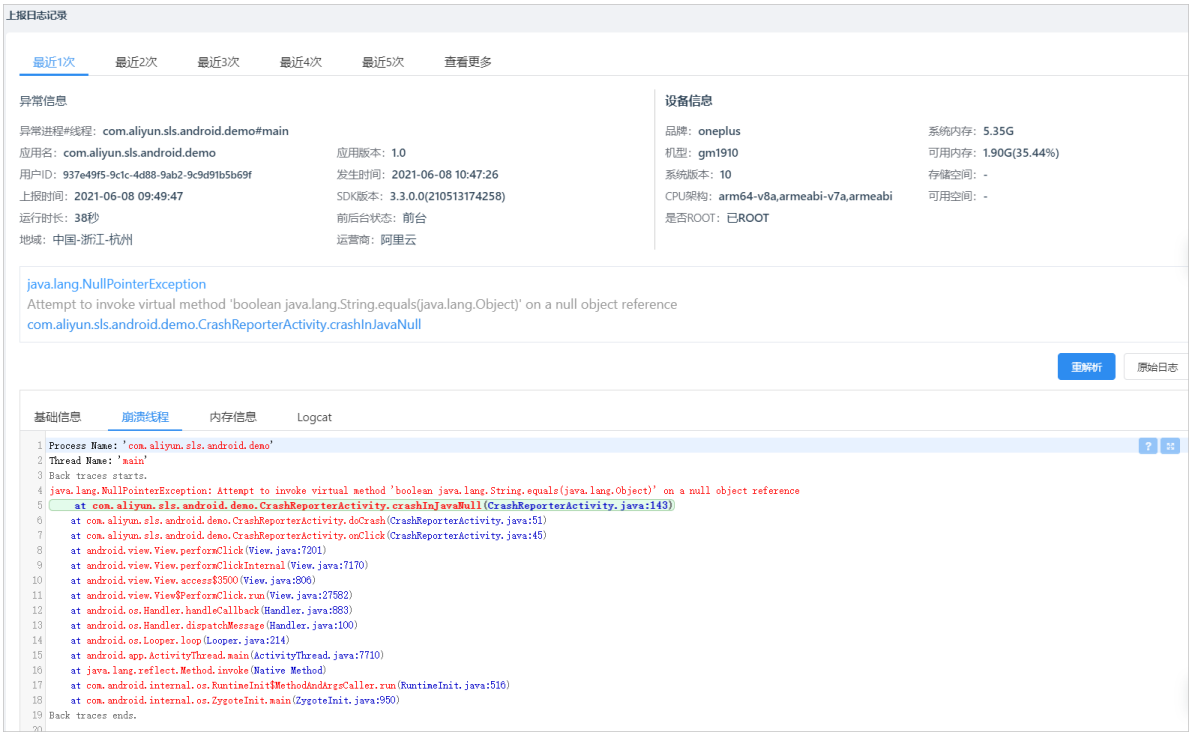


- 查看指定时间范围内App崩溃所涉及的影响用户和异常次数。

- 支持多个维度排行，查看发生崩溃的TOP手机机型、系统版本、App版本，快速定位共性问题。



- 通过崩溃上报日志，查看崩溃的设备信息、App运行信息以及核心的崩溃堆栈，分析崩溃代码找到问题根源。



12.4.5. ANR分析

ANR是指使用Android App过程中出现应用无响应的现象。日志服务ANR分析大盘展示了ANR相关的用户影响趋势、异常趋势、异常问题等信息，帮助您快速分析出现崩溃问题所涉及的影响以及主要原因。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

功能入口

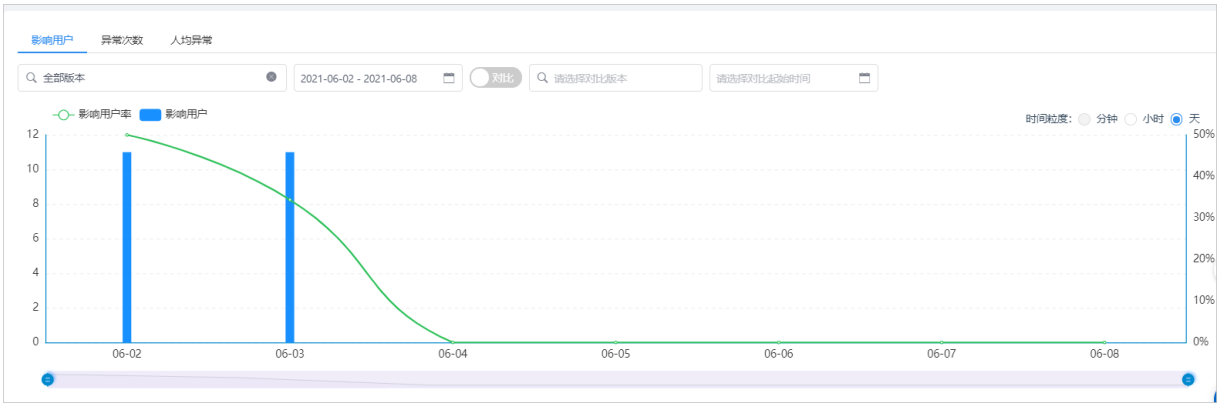
1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**ANR分析**。

数据趋势与对比

ANR分析大盘中，以折线图、柱状图形式展示影响用户、影响用户率、异常次数、异常率、人均异常等指标数据。

当您选择**时间粒度**为**天**时，单击折线图或状态图，系统自动跳转至**异常问题**列表区域，展示当天所涉及的异常问题。

您还可以指定版本和时间，对ANR数据进行对比。对比的时间区间需一致，例如都选择一天，或者都选择一个月。



TOP异常问题

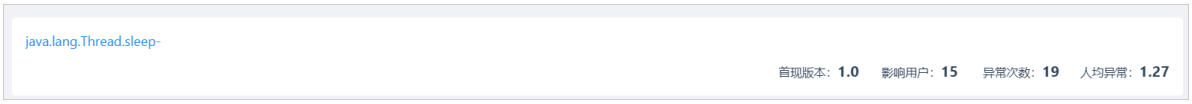
当您选择版本和时间后，系统自动展示异常问题并按照**影响用户**排序，帮助您一目了然找到TOP异常问题。

您还可以通过聚合维度、状态、责任人等条件筛选异常问题。

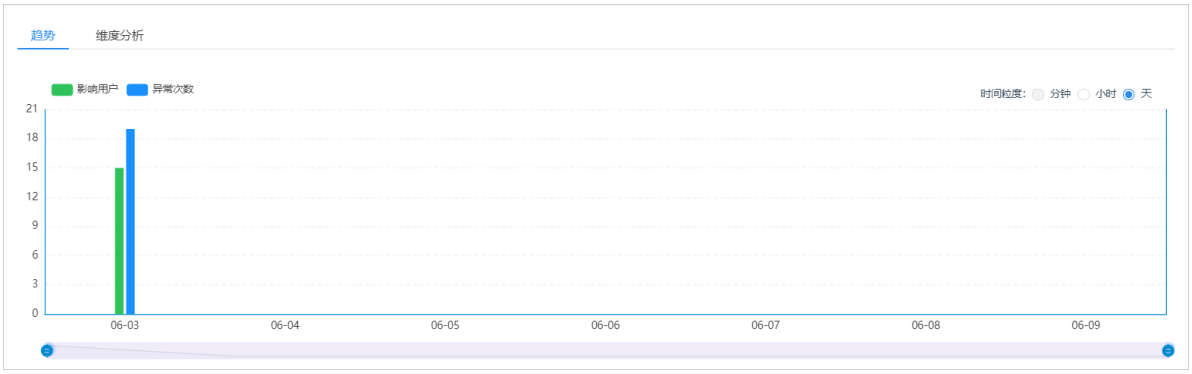
异常问题详情

您在**异常问题**列表中，单击目标异常问题，可查看异常问题详情。

- 展示主要的堆栈信息以及App版本号、影响用户、异常次数、人均异常等信息。

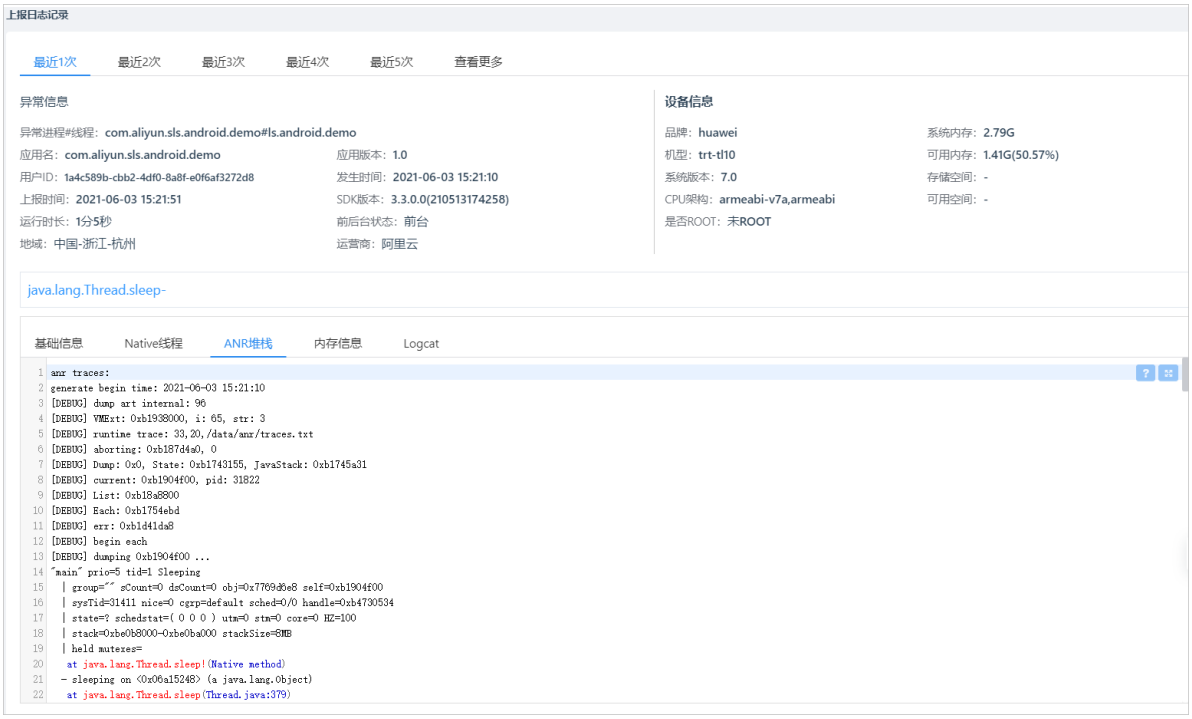


- 查看指定时间范围内App崩溃所涉及的影响用户和异常次数。



- 支持多个维度排行，查看发生崩溃的TOP手机机型、系统版本、App版本，快速定位共性问题。

- 通过崩溃上报日志，查看崩溃的设备信息、App运行信息以及核心的崩溃堆栈，分析崩溃代码找到问题根源。



12.4.6. 高级查询

高级查询适用于查询条件复杂的分析场景，您可以自定义组合多个查询条件。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**高级查询**。
5. 在**高级查询**页面，选择查询条件，单击**查询**。

您可以通过日志分类、版本、时间、前后台状态等查询条件筛选数据。

查询结果中包括聚合分析、趋势分析和日志列表。更多信息，请参见崩溃分析、ANR分析。

6. 如果您需要保存常用的查询条件用于持续追踪一系列问题，可单击**保存为查询模板**。

添加查询模板后，您直接单击模板查询模板即可查询相应的数据。

12.4.7. 自定义查询

日志服务提供专属Logstore，用于存储接入到日志服务的移动运维监控数据。您可以在该Logstore中执行查询和分析操作。

前提条件

已接入数据。具体操作，请参见接入Android App监控数据或接入iOS App监控数据。

背景信息

移动端应用相关的专属Logstore说明如下：

- iOS应用：sls-alysls-track-ios
- Android应用：sls-alysls-track-android

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**自定义查询**。
5. 在**自定义查询**页面，执行查询和分析日志、创建告警、加工数据等操作。

   自定义查询页面支持Logstore相关的所有功能。具体操作，请参见查询和分析日志。

12.4.8. 版本管理

当您的项目代码需要做打包混淆时，需要上传符号表，进行版本管理。本文介绍符号表的操作步骤、格式和示例。

前提条件

已添加应用。具体操作，请参见添加应用。

上传符号表

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 创建版本。
   i. 在左侧导航栏中，单击**版本管理**。
   ii. 在**版本管理**页签中，单击**新建版本**。
   iii. 在**新建版本**对话框中，设置**版本**、**子版本**和**版本说明**，然后单击**确定**
5. 上传符号表。
   i. 在版本列表中，单击目标版本对应的**上传**。
   ii. 选择保存时长，然后选择符号表文件。

符号表文件示例

- Android

```
xxx.zip
|-- mapping.txt     --多个mapping文件，需手动合并。
|-- libxx1.so       --要求与发布的so同名。
|-- libxx2.so
|-- armv7
|-- libxx3.so       --同文件名，多个架构放不同的目录。
|-- x86
|-- libxx3.so       --同文件名，多个架构放不同的目录。
```

- iOS

```
xxx.zip
|-- demo.ios.app.dSYM
|--Contents
|--Resources
|--DWARF
|--ios                --主应用符号，要求与xcode模块同名，路径不限。
|-- sdk1               --sdk1符号，要求与xcode模块同名，路径不限。
|-- sdk2               --sdk2符号，要求与xcode模块同名，路径不限。
```

符号表格式

- Java符号

将多个mapping文件内容添加到一个文件中，命名为*mapping.txt*，并打包为.zip文件。如果包含so符号，可将so符号文件与*mapping.txt*文件一起打包。

- Android so库

  > 注意 保证符号表的so文件与发布的so文件同名。

  如果一个版本里包含同名但不同架构的so库，您可将其压缩到不同的目录中，反符号化时需要通过buildid去关联。

  建议编译时使用-g参数，并加上debug信息，崩溃堆栈可以解析到代码行级。实际发布时，再使用strip命令去掉调试信息。如果不带debug信息，则只能定位到函数名级。

  下列情况下，需要so文件带有buildid。

  - 不同CPU架构的so文件名一样，放在不同的路径下。
  - 同一个应用版本里，可能有多个版本的so库，使用了动态加载的技术。

  检查编译参数，确定没有`--build-id=none`，即表示生成带buildid的so库。如果没有buildid，可以添加编译参数`ld_flags += -Wl,--build-id=sha1`。您还可以通过file命令检查so文件是否带buildid，调试信息如下：

  ```
  file libmytest.so
  libmytest.so: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
  interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 2.6.24,
  BuildID[sha1]=63f643875228a281430e69123def59c9b6894803,
  with debug_info, not stripped
  ```

- iOS OC符号

  将编译生成的dSYM文件夹打包为.zip文件即可。如果存在第三方库，可将第三方库与dSYM文件夹一起打包，目录层次不作要求。

  如果未保存dSYM文件，可上传未加密的ipa文件作为符号表。此方式可定位函数名，无法定位文件名及行号。

  iOS符号表通过UUID做关联，您可通过如下命令查看符号表的UUID。

  ```
  xcrun dwarfdump --uuid 符号表文件路径
  ```

  检查与日志中image块中可执行模块的uuid是否一致。

12.5. 前端监控

12.5.1. 基本概念

本文介绍前端监控相关的基本概念。

名词	说明
页面性能	页面被完全加载所需的时间。 移动运维监控服务中统计的页面性能为页面被加载所需的平均耗时。
2s快开比	页面完全加载的时间≤2s的页面的占比，即完全加载时间≤2s的页面PV/页面总PV×100%。

名词	说明
JS异常	JavaScript异常。
异常PV	在一次页面访问过程中，如果发生过JS异常，则此PV记为一次异常PV。 例如用户在一次访问过程中出现了三种不同的JS异常，则此次访问计为三次异常PV。
PV	页面被访问的次数。
UV	访问页面的用户数量。
采样PV	上报页面性能的PV。
采样率	监控指标数据的采样率。
影响用户	JS异常等情况所影响的用户数。
网络类型	用户访问页面所使用的网络类型。
运营商	用户访问页面所使用的网络运营商。
操作系统	用户访问页面所使用的设备操作系统。
客户端	用户访问页面所使用的客户端应用，例如系统浏览器、微信等。
前端版本	用户所访问的页面的发布版本。
探针版本	用于前端监控的JS SDK的版本，JS SDK统称探针。

12.5.2. 实时大盘

实时大盘实时更新并展示最近一小时或今天访问页面的关键指标数据。实时大盘用于在发布版本、重要活动等关键时间点实时监控访问页面。

前提条件

已接入数据。具体操作，请参见接入前端监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**实时大盘**。
5. 在**实时大盘**页面，查看各个监控指标的实时动态。
   - **JS异常**图表：展示当前时间范围内发生JS异常的次数。更多信息，请参见JS异常。
   - **API成功率**图表：展示当前时间范围内的API请求成功率。更多信息，请参见API请求。
   - **页面性能**图表：展示当前时间范围内的页面性能。更多信息，请参见页面性能。

- **资源异常**图表：展示当前时间范围内资源加载的异常情况。更多信息，请参见资源异常。
- **页面访问**图表：展示当前时间范围内页面的访问次数和访问用户数。更多信息，请参见页面访问。

12.5.3. JS异常

JS异常大盘用于监控用户访问页面过程中所遇到的JS异常。JS异常大盘通过JS异常次数、异常次数PV比、影响用户等信息展示页面运行的健康情况，并通过JS异常堆栈帮您快速分析页面发生JS异常的原因。

前提条件

已接入数据。具体操作，请参见接入前端监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**JS异常**。

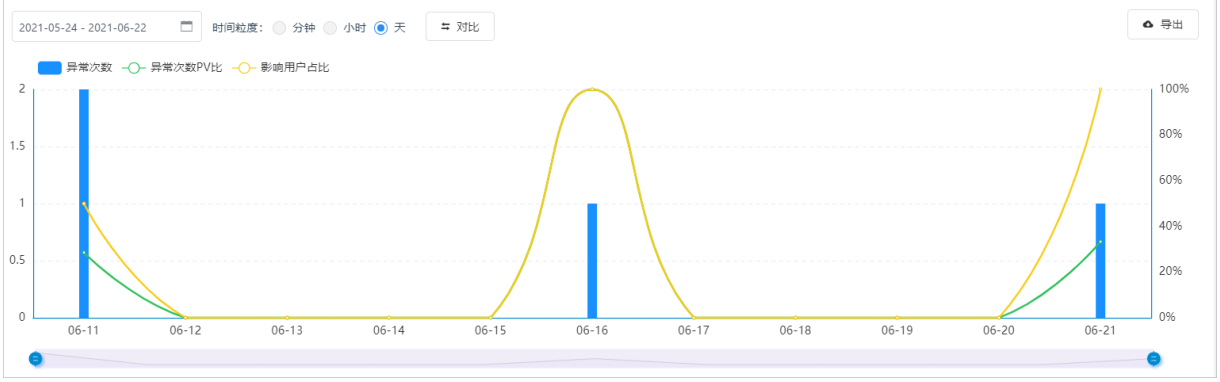
今日指标

本区域展示今天页面发生JS异常的次数、异常次数PV比、影响用户和影响用户占比等数据。

数据趋势与对比

JS异常大盘以折线图、柱状图形式展示异常次数、异常次数PV比、影响用户占比等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对JS异常数据进行对比。

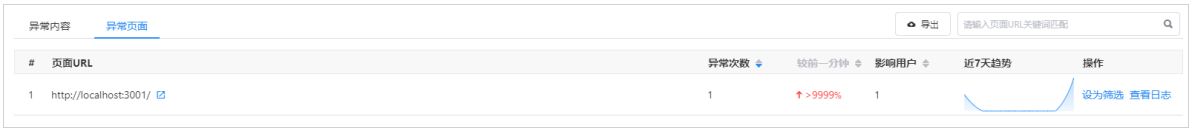
单击折线图或柱状图，系统自动跳转至**异常内容**区域，展示当前时间范围内所涉及的异常内容。



异常详情

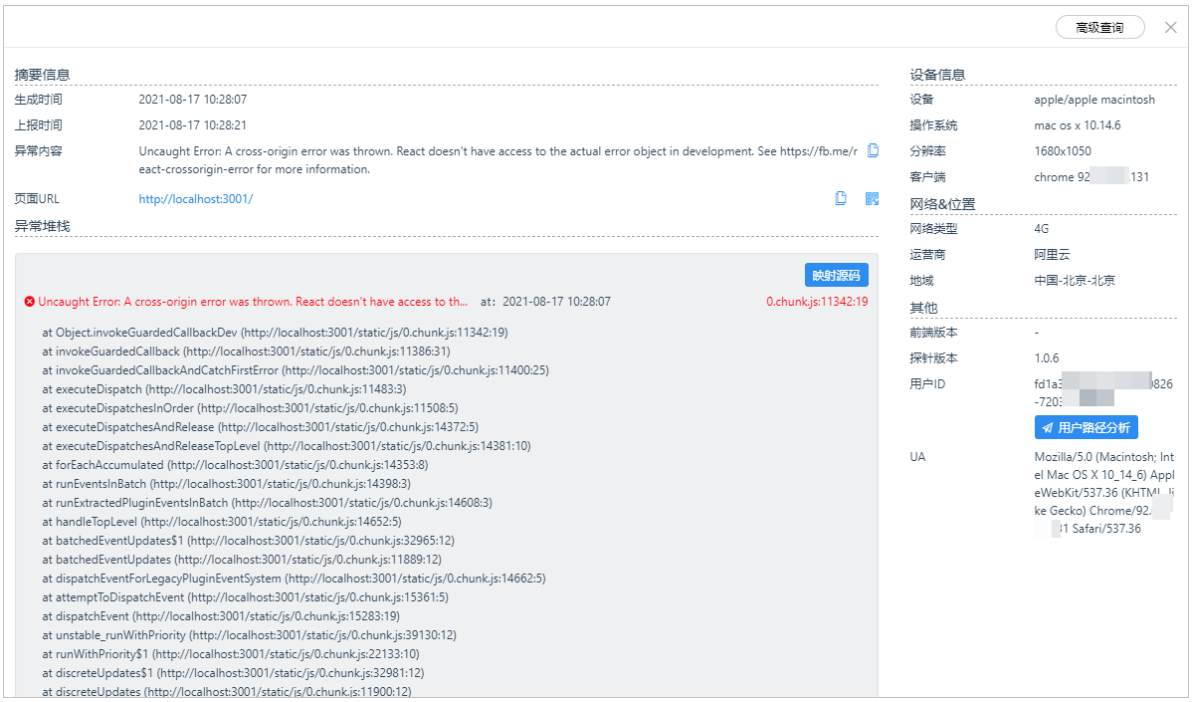
- **异常内容**展示异常问题的详情。

- **异常页面**展示发生JS异常的页面。



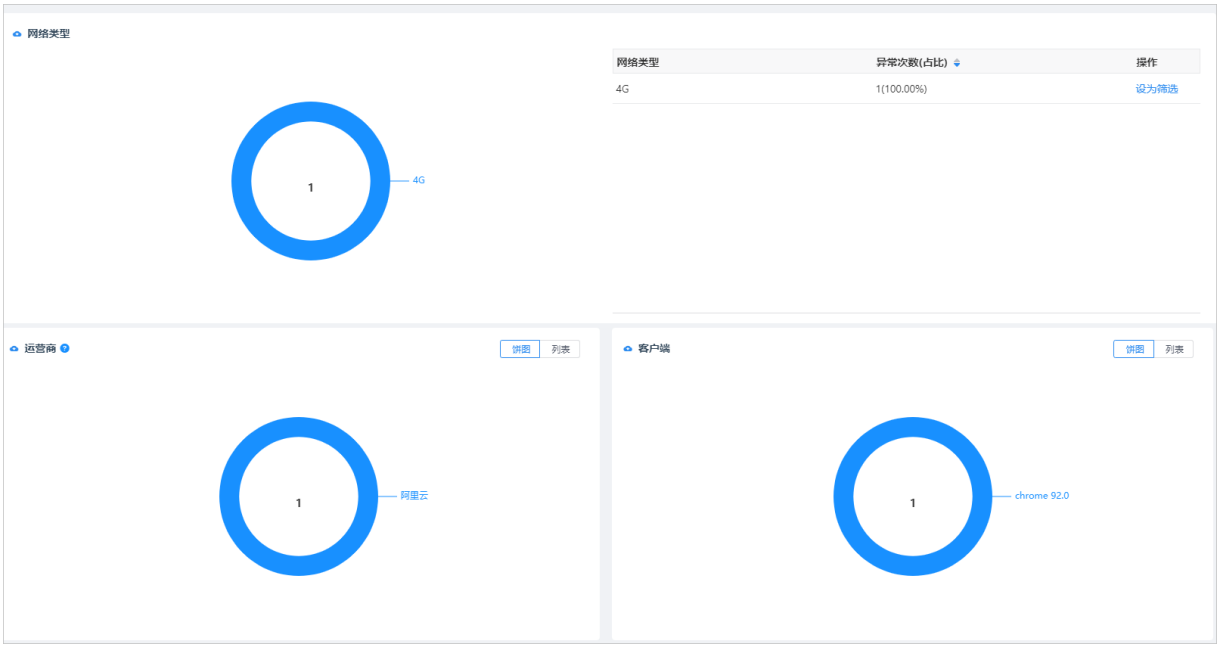
- 日志

  每一条异常内容或异常URL都有相应的日志，您可以单击**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息以及核心的异常堆栈等。您可以通过分析核心的异常堆栈找到问题根源。



多维度聚合分析

移动运维监控服务支持通过网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合分析。



12.5.4. API请求

API请求大盘主要通过成功率、成功请求的耗时、失败请求的耗时以及失败请求的影响用户数等指标监控API请求情况，并通过多维度聚合分析，帮您直观地了解API异常的原因。

前提条件

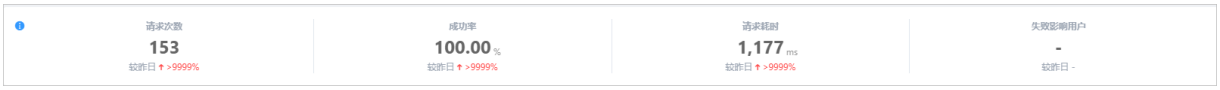
已接入数据。具体操作，请参见接入前端监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**API请求**。

今日指标

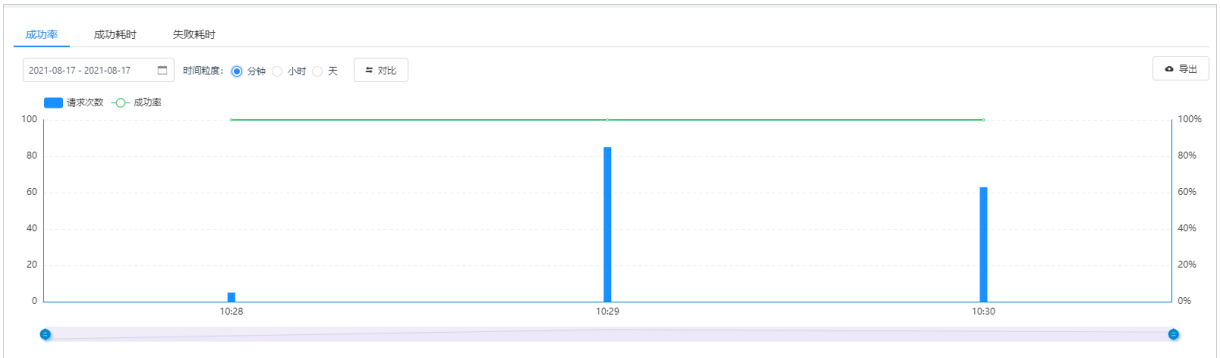
本区域展示今天API请求次数、API请求成功率、API请求的平均耗时和API请求失败所影响用户数等监控指标数据。



数据趋势与对比

API请求大盘以折线图、柱状图形式展示请求成功率、请求成功耗时、请求失败耗时等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

单击折线图或柱状图，系统自动跳转至**全部URL**区域，展示当前时间范围内API请求所涉及的URL地址。



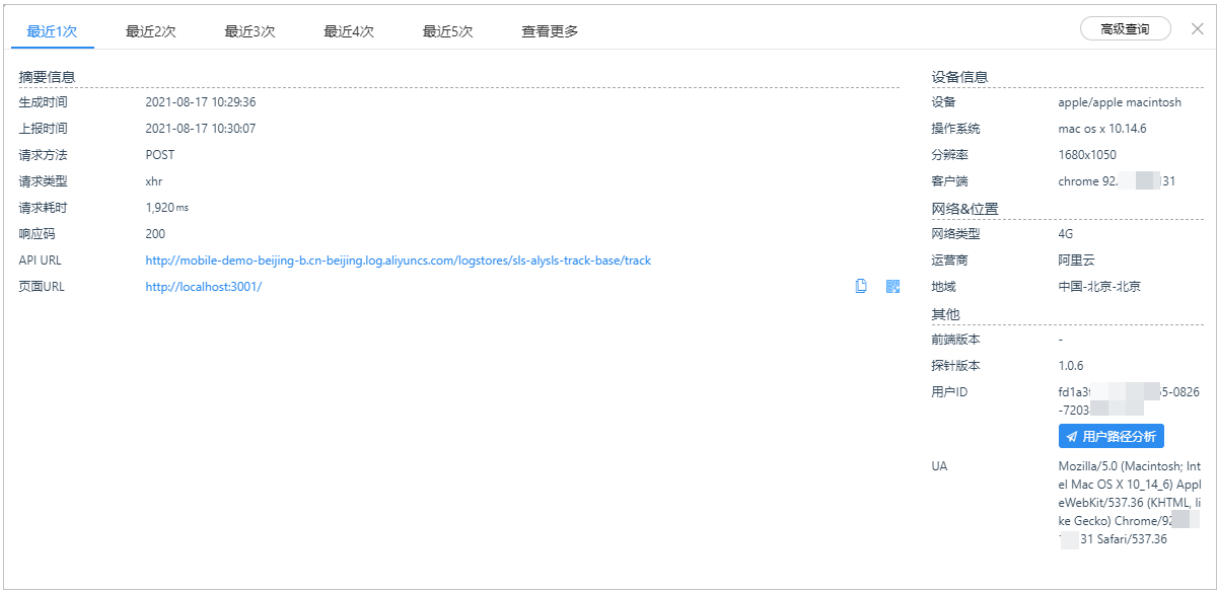
URL详情

- **全部URL**展示API请求所涉及的URL地址。

- **失败URL**展示API请求失败所涉及的URL地址。

日志详情

每一条URL都有相应的日志，您可以单击目标URL对应的**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息等。



多维度聚合分析

移动运维监控服务支持通过响应码、网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合分析。

12.5.5. 页面性能

页面性能大盘主要通过首字节、DOM Ready、页面完全加载、采样PV、2s快开比等指标监控页面性能，并通过多维度聚合分析，帮您快速分析出现慢页面的原因。

前提条件

已接入数据。具体操作，请参见接入前端监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**页面性能**。

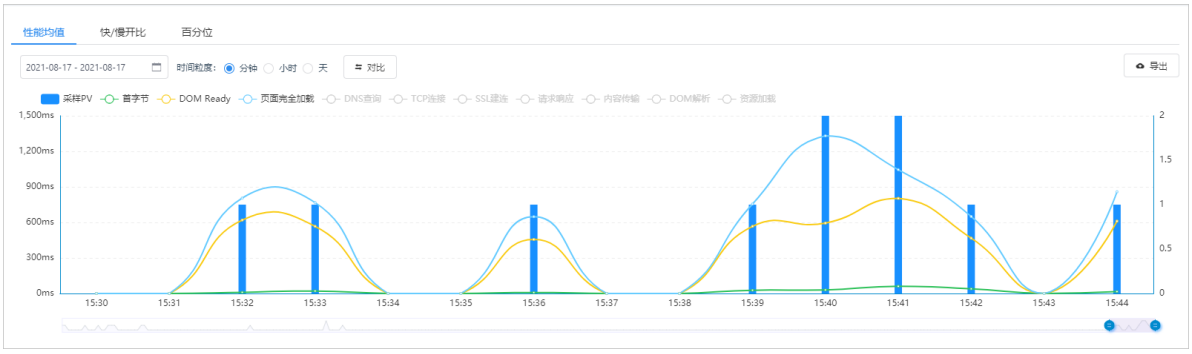
今日指标

本区域展示今天首字节、DOM Ready、页面完全加载、采样PV、2s快开比等监控指标数据。

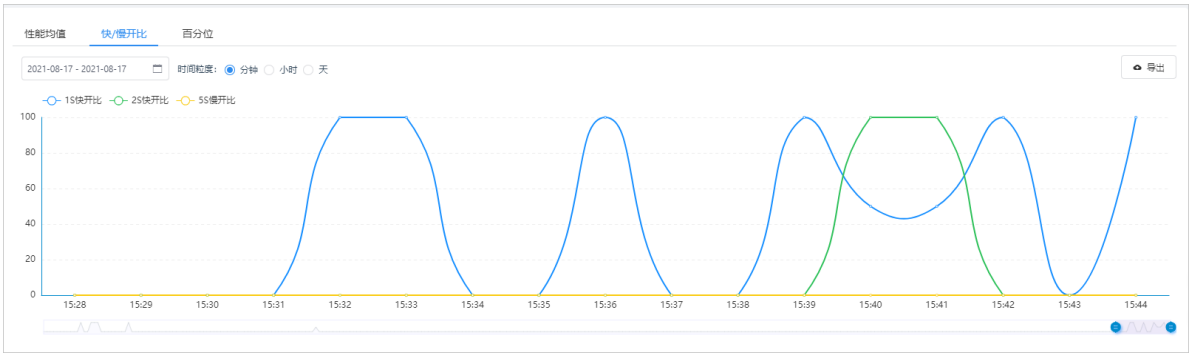
数据趋势与对比

页面性能大盘以折线图、柱状图形式展示采样PV、首字节、DOM Ready、页面完全加载、1s快开比、2s快开比、5s慢开比、百分位等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

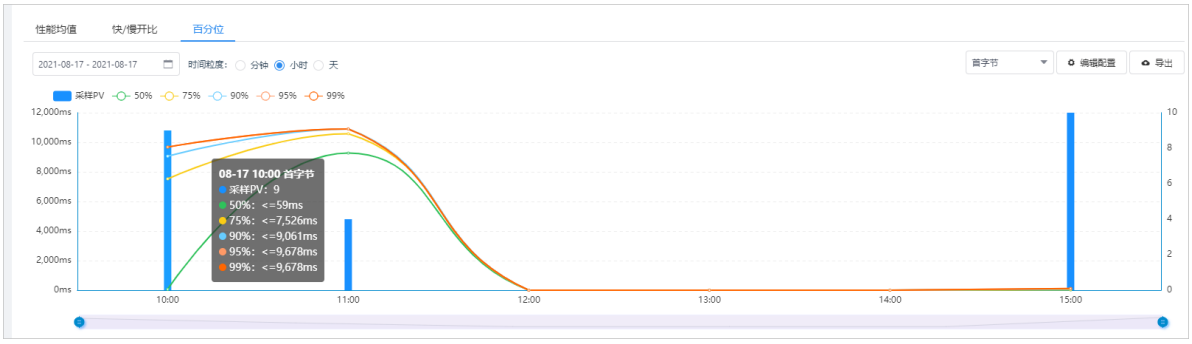
- **性能均值**图表：展示采样PV、首字节、DOM Ready、页面完全加载等监控指标数据。



- **快/慢开比**图表：展示1s快开比、2s快开比和5s慢开比的变化趋势。



- **百分位**图表：展示处于不同百分位的监控数据。例如下图中的50%表示当前时间，处于50分位的首字节时间为59 ms。



单击折线图或柱状图，系统自动跳转至**页面URL**区域，展示当前时间范围内目标页面所涉及的URL地址。

URL详情

页面URL展示访问页面的URL地址。

日志详情

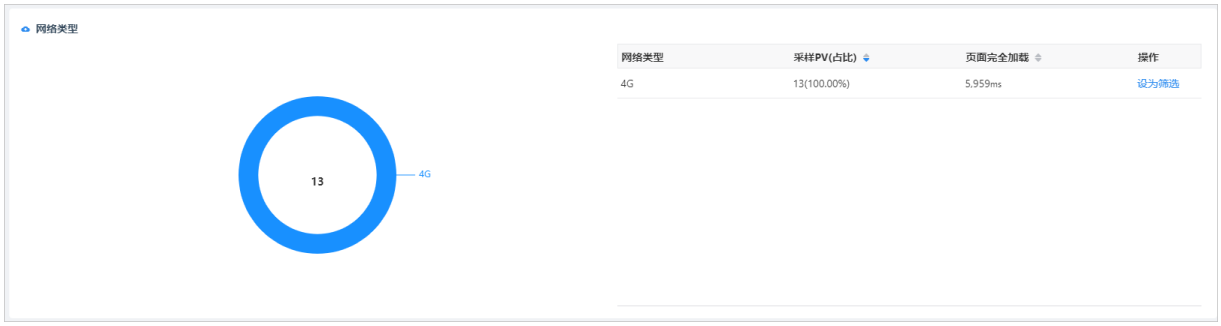
每一条URL都有相应的日志，您可以单击**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息、页面性能关键指标、页面加载瀑布图等。

页面加载瀑布图

页面加载瀑布图按照页面加载的顺序，直观地展示页面加载过程中发生重要事件的位置及对应的加载时间。

多维度聚合分析

移动运维监控服务支持通过网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合分析。



12.5.6. 资源异常

资源异常是指页面加载过程中JS、CSS、图片等资源发生异常。资源异常大盘通过失败资源数、异常次数PV比、资源异常所影响的用户数等指标监控资源异常情况，并通过多维度聚合分析，帮您快速分析资源异常的原因。

前提条件

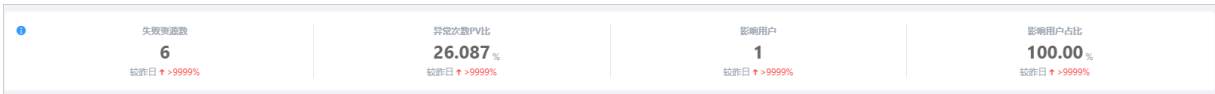
已接入数据。具体操作，请参见接入前端监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**资源异常**。

今日指标

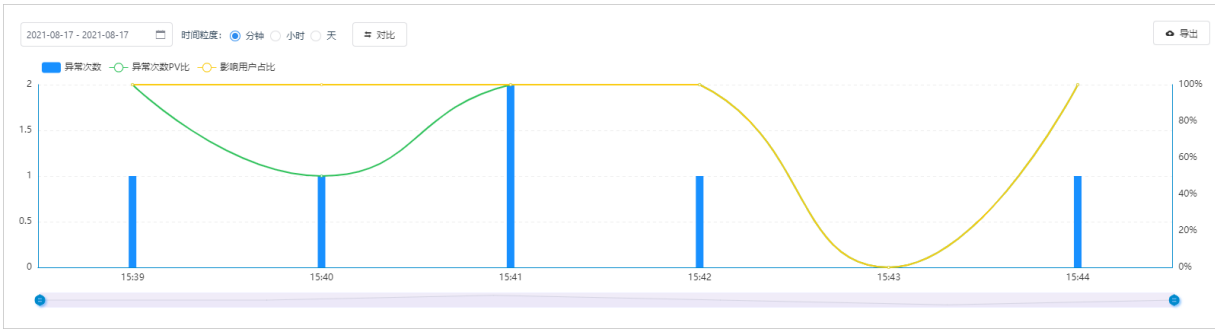
本区域展示今天失败资源数、异常次数PV比、影响用户和影响用户占比等监控指标数据。



数据趋势与对比

资源异常大盘以折线图、柱状图形式展示异常次数、异常次数PV比、影响用户占比等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

单击折线图或柱状图，系统自动跳转至**异常域名**区域，展示当前时间范围内发生资源异常的域名。



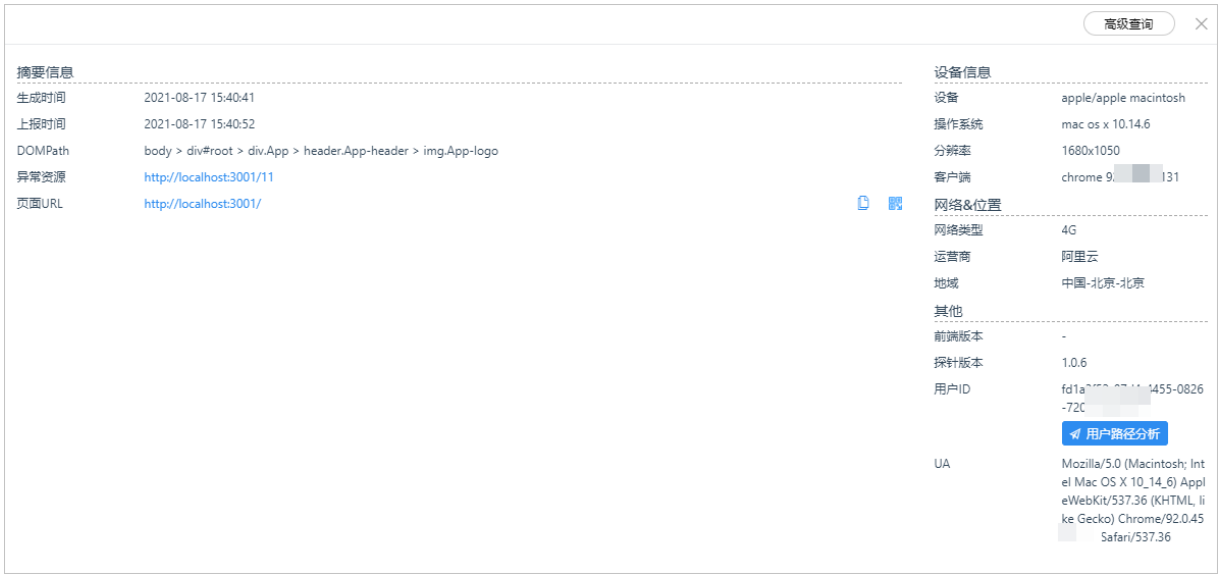
异常详情

- **异常域名**展示发生资源异常的域名。

- **异常页面**展示发生资源异常的页面。

日志详情

每一条异常信息都有相应的日志，您可以单击目标域名或URL对应的**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息等。



多维度聚合分析

移动运维监控服务支持通过资源类型、前端版本、网络类型、运营商以及地域分布等维度进行聚合分析。

12.5.7. 页面访问

页面访问主要展示当前站点的访问次数、用户数分布情况。

前提条件

已接入数据。具体操作，请参见接入前端监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**页面访问**。

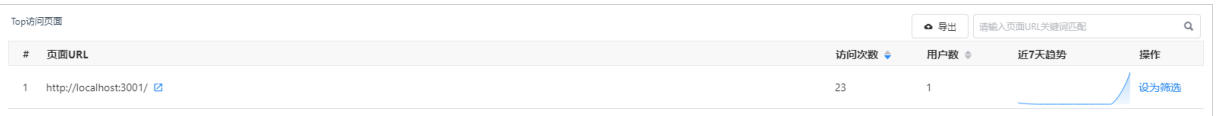
数据趋势与对比

页面访问大盘以折线图、柱状图形式展示请求次数、用户数等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

单击折线图或柱状图，系统自动跳转至**页面URL**区域，展示当前时间范围内页面访问所涉及的URL地址。

Top访问页面

系统自动按照访问次数对URL地址进行排序，帮助您一目了然找到Top访问页面。



多维度聚合分析

移动运维监控服务支持通过客户端、前端版本、网络类型、运营商以及地域分布等维度进行聚合分析。

12.5.8. 自定义查询

日志服务提供专属Logstore，用于存储接入到日志服务的前端监控数据。您可以在该Logstore中执行查询和分析操作。

前提条件

已接入数据。具体操作，请参见接入前端监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**自定义查询**。
5. 在**自定义查询**页面，执行查询和分析日志、创建告警、加工数据等操作。

   自定义查询页面支持Logstore相关的所有功能。具体操作，请参见查询和分析日志。

12.6. 小程序监控

12.6.1. 基本概念

本文介绍小程序监控相关的基本概念。

名词	说明
页面性能	页面被完全加载所需的时间。 移动运维监控服务中统计的页面性能为页面被加载所需的平均耗时。
启动性能	小程序从冷启动到首页首次渲染完成所需要的时间。
业务可用	业务可使用的时间。
采样PV	上报页面性能数据的PV。
onLand	页面加载所需的时间。
onReady	页面首次渲染完成所需的时间。
JS异常	JavaScript异常。
异常PV	在一次小程序访问过程中，如果发生过JS异常，则此PV记为一次异常PV。 例如用户在一次访问过程中出现了三种不同的JS异常，则此次访问计为三次异常PV。
PV	小程序被访问的次数。
UV	访问小程序的用户数量。
采样率	平台规定的监控指标的采样率。
影响用户	JS异常等情况所影响的用户数。
网络类型	用户访问小程序所使用的网络类型。
运营商	用户访问小程序所使用的网络运营商。
操作系统	用户访问小程序所使用的设备操作系统。
客户端	用户访问小程序所使用的客户端应用，例如钉钉。
小程序版本	用户所访问的小程序的发布版本。
探针版本	用于小程序监控的JS SDK的版本，JS SDK统称探针。

12.6.2. 实时大盘

实时大盘实时更新并展示最近一小时或今天小程序运行的关键指标数据。实时大盘用于在发布版本、重要活动等关键时间点实时监控小程序。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**实时大盘**。
5. 在**实时大盘**页面，选择平台和时间，然后查看各个监控指标的实时动态。
   - **JS异常**图表：展示当前时间范围内发生JS异常的次数。更多信息，请参见JS异常。
   - **API成功率**图表：展示当前时间范围内的API请求成功率。更多信息，请参见API请求。
   - **启动性能**图表：展示当前时间范围内的启动性能。更多信息，请参见启动性能。
   - **页面性能**图表：展示当前时间范围内的页面性能。更多信息，请参见页面性能。
   - **页面访问**图表：展示当前时间范围内小程序的访问次数和访问用户数。更多信息，请参见页面访问。

12.6.3. JS异常

JS异常大盘用于监控用户访问小程序过程中所遇到的JS异常。JS异常大盘通过JS异常次数、异常次数PV比、影响用户等信息展示小程序运行的健康情况，并通过JS异常堆栈帮您快速分析小程序发生JS异常的原因。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**JS异常**。

今日指标

本区域展示今天小程序发生JS异常的次数、异常次数PV比、影响用户和影响用户占比等数据。

数据趋势与对比

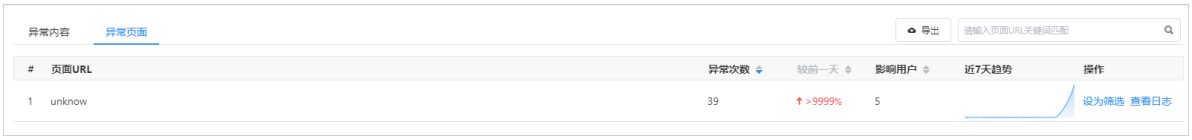
JS异常大盘以折线图、柱状图形式展示异常次数、异常次数PV比、影响用户占比等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对JS异常数据进行对比。

单击折线图或柱状图，系统自动跳转至**异常内容**区域，展示当前时间范围内所涉及的异常内容。

异常详情

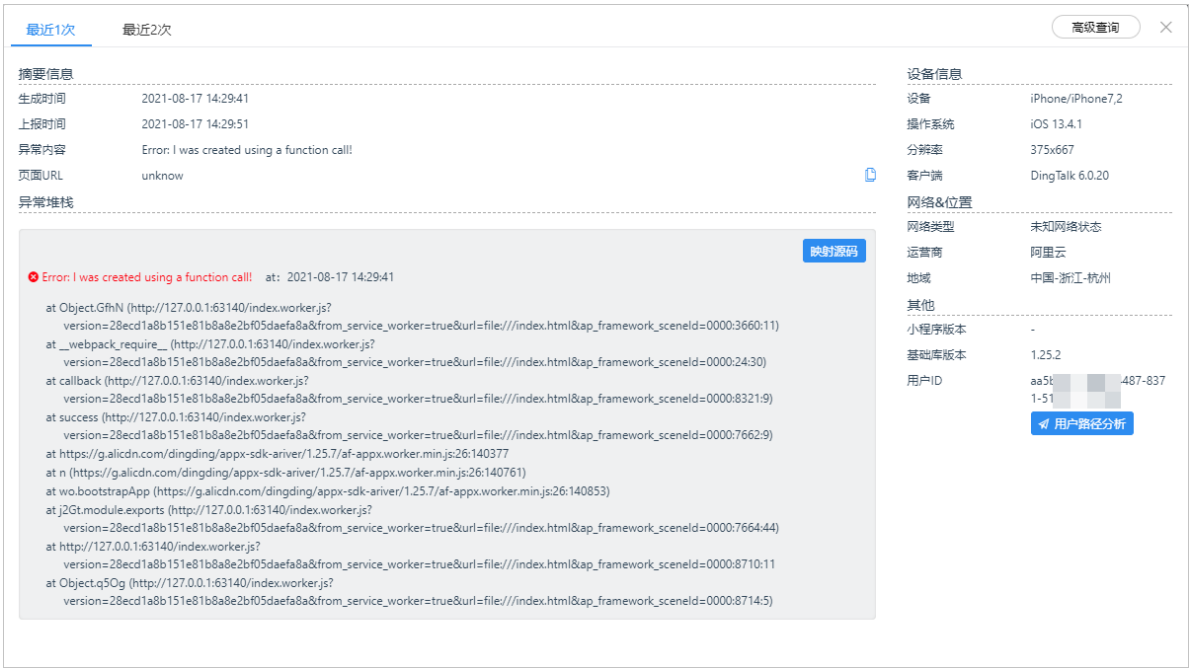
- **异常内容**展示异常问题的详情。

- **异常页面**展示发生JS异常的页面。



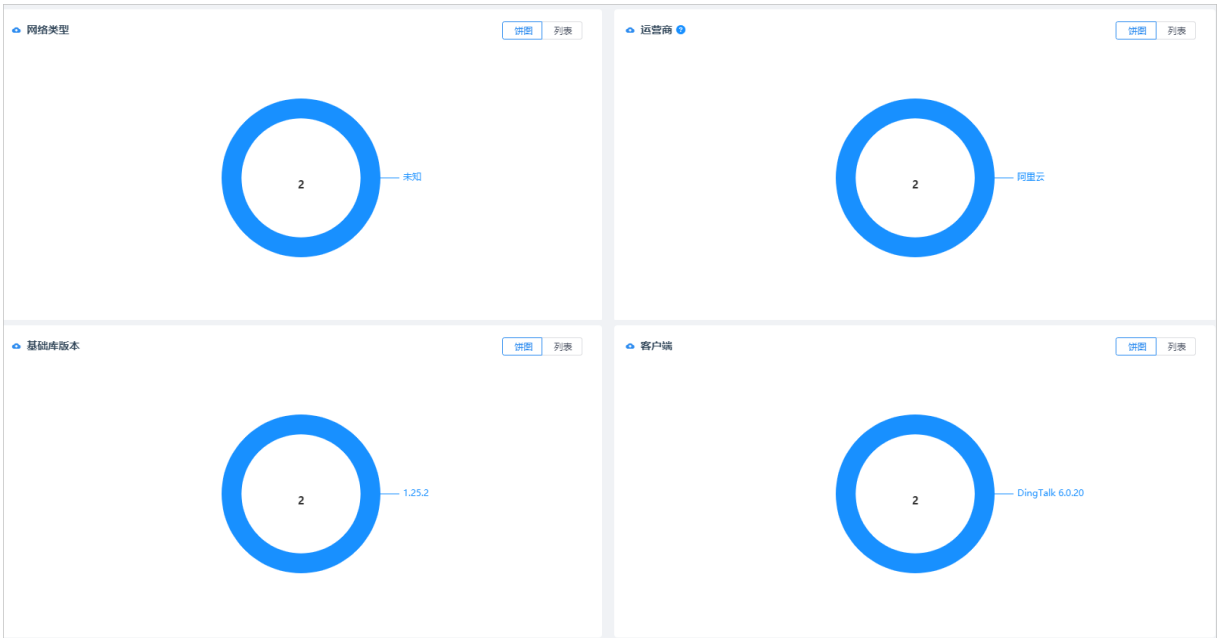
- 日志

  每一条异常内容或异常URL都有相应的日志，您可以单击**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息以及核心的异常堆栈等。您可以通过分析核心的异常堆栈找到问题根源。



多维度聚合查询

移动运维监控服务支持通过网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合查询。



12.6.4. API请求

API请求大盘主要通过API请求次数、成功率、请求耗时和失败影响用户等指标监控API请求情况，并通过多维度聚合分析，帮您直观地了解API请求失败的原因。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**API请求**。

今日指标

本区域展示今天API请求次数、成功率、请求耗时和失败影响用户等监控指标数据。

数据趋势与对比

API请求大盘以折线图、柱状图形式展示请求成功率、请求成功耗时、请求失败耗时等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

单击折线图或柱状图，系统自动跳转至**全部URL**区域，展示当前时间范围内API请求所涉及的URL地址。

URL详情

- **全部URL**展示API请求所涉及的URL地址。

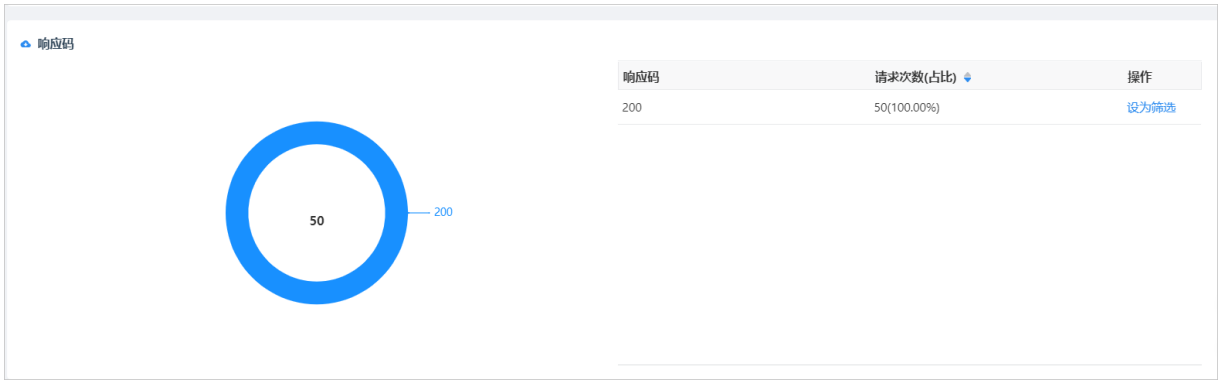
- **失败URL**展示API请求失败所涉及的URL地址。

日志详情

每一条URL都有相应的日志，您可以单击目标URL对应的**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、前端版本、网络信息等。

多维度聚合分析

移动运维监控服务支持通过响应码、网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合分析。



12.6.5. 页面性能

页面性能大盘主要通过onReady、业务可用、采样PV等指标监控页面性能，并通过多维度聚合分析，帮您快速分析出现慢页面的原因。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**页面性能**。

今日指标

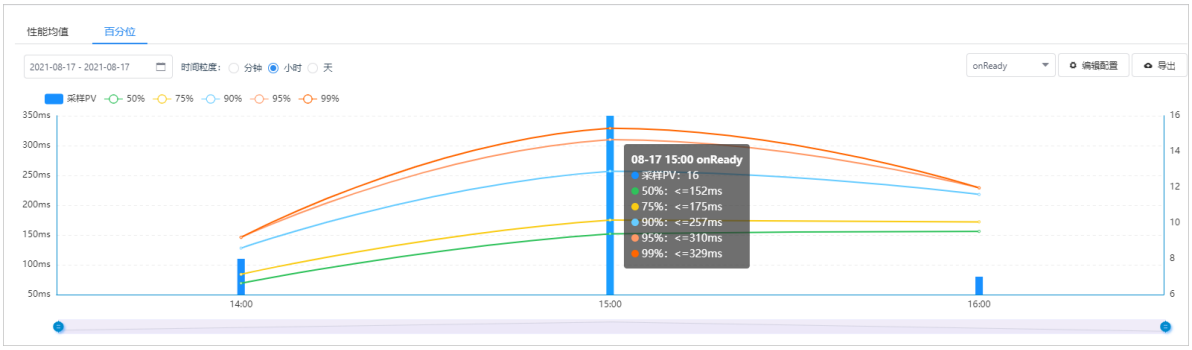
本区域展示今天onReady、业务可用、采样PV等监控指标数据。

数据趋势与对比

页面性能大盘以折线图、柱状图形式展示采样PV、onReady、业务可用、百分位等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对API请求数据进行对比。

- **性能均值**图表：展示采样PV、onReady、业务可用等监控指标数据。

- **百分位**图表：展示处于不同百分位的监控数据。例如下图中的50%表示当前时间，处于50分位的onReady时间为152ms。



单击折线图或柱状图，系统自动跳转至**页面URL**区域，展示当前时间范围内小程序中访问页面的URL地址。

URL详情

页面URL展示小程序中访问页面的URL地址。

日志详情

每一条URL都有相应的日志，您可以单击**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、小程序版本、网络信息、页面性能关键指标等。

多维度聚合分析

移动运维监控服务支持通过网络类型、运营商、客户端、操作系统、前端版本以及地域分布等维度进行聚合分析。

12.6.6. 启动性能

启动性能大盘主要通过onLand、onReady、业务可用、采样PV等指标监控页面性能，并通过多维度聚合分析，帮您快速分析出现小程序启动慢的原因。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**启动性能**。

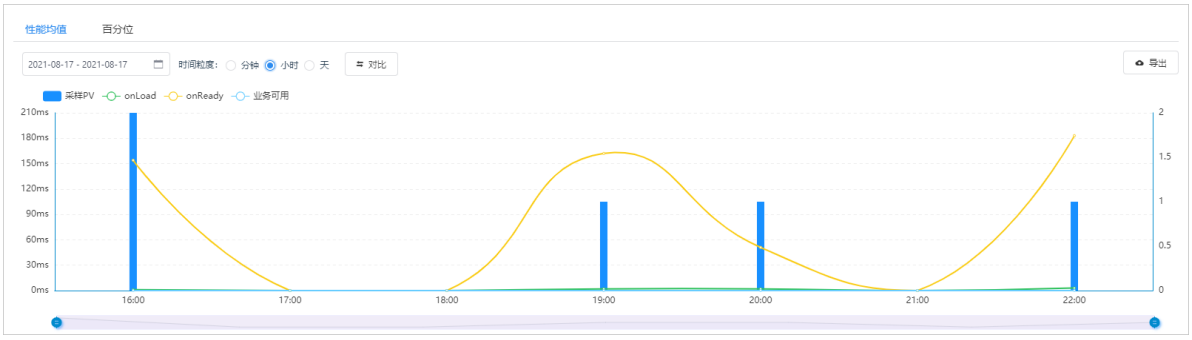
今日指标

本区域展示今天onLand、onReady、业务可用、采样PV等监控指标数据。

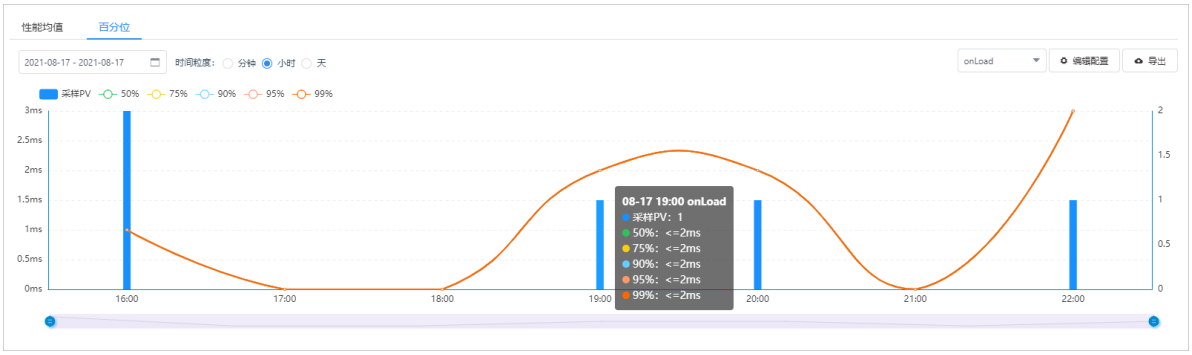
数据趋势与对比

启动性能大盘以折线图、柱状图形式展示采样PV、onLand、onReady、业务可用、百分位等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对启动性能数据进行对比。

- **性能均值**图表：展示采样PV、onLand、onReady、业务可用等监控指标数据。



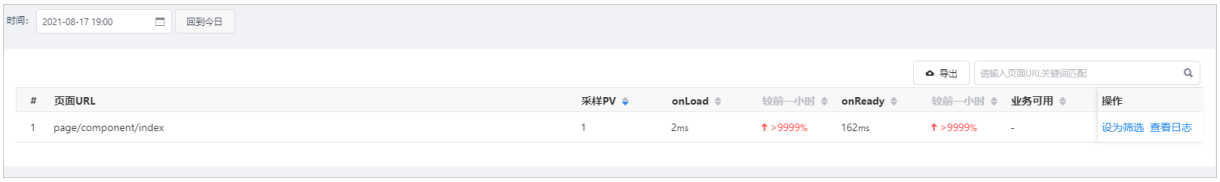
- **百分位**图表：展示处于不同百分位的监控数据。例如下图中的99%表示当前时间，处于99分位的onLand时间为2 ms。



单击折线图或柱状图，系统自动跳转至**页面URL**区域，展示当前时间范围内目标页面所涉及的URL地址。

URL详情

页面URL展示小程序中访问页面的URL地址。



日志详情

每一条URL都有相应的日志，您可以单击**查看日志**，查看日志详情。日志中包括摘要信息、设备信息、小程序版本、网络信息、页面性能关键指标等。

多维度聚合分析

移动运维监控服务支持通过网络类型、运营商、基础库版本、客户端、操作系统、小程序版本以及地域分布等维度进行聚合分析。

12.6.7. 页面访问

页面访问主要展示当前小程序的访问次数、用户数分布情况。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

功能入口

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**页面访问**。

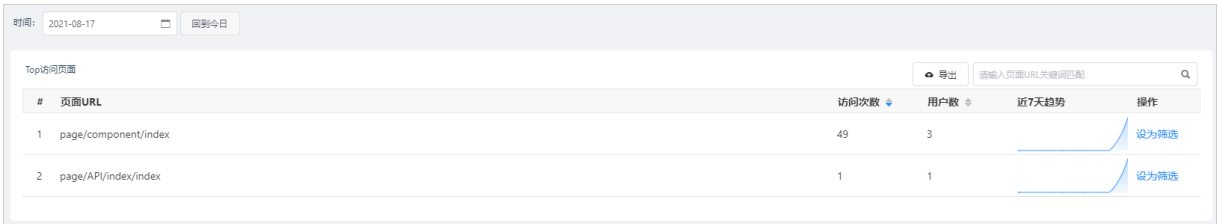
数据趋势与对比

页面访问大盘以折线图、柱状图形式展示请求次数、用户数等监控指标。您还可以指定时间、客户端版本、探针版本或前端版本，对页面访问数据进行对比。

单击折线图或柱状图，系统自动跳转至**页面URL**区域，展示当前时间范围内小程序中页面的URL地址。

Top访问页面

系统自动按照访问次数对URL地址进行排序，帮助您快速找到Top访问页面。



多维度聚合分析

移动运维监控服务支持通过客户端、小程序版本、网络类型、运营商以及地域分布等维度进行聚合分析。

12.6.8. 自定义查询

日志服务提供专属Logstore，用于存储接入到日志服务的小程序监控数据。您可以在该Logstore中执行查询和分析操作。

前提条件

已接入数据。具体操作，请参见接入小程序监控数据。

操作步骤

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**移动运维监控**。
3. 在应用列表中，单击目标应用。
4. 在左侧导航栏中，单击**自定义查询**。
5. 在**自定义查询**页面，执行查询和分析日志、创建告警、加工数据等操作。

   自定义查询页面支持Logstore相关的所有功能。具体操作，请参见查询和分析日志。

13.Jupyter Lab

13.1. 简介

日志服务Jupyter Lab是日志服务与机器学习PAI联合推出的开箱即用的云上开发IDE，集成开源Jupyter Lab和日志服务Python SDK。您无需任何运维配置，即可编写、调试及运行日志服务Python代码。

在PAI-DSW功能的基础上，日志服务Jupyter Lab进行深度定制化开发，为您提供多种场景开发模板（包括配置管理、任务管理、探索性数据分析和AIops功能体验场景）。您可以根据实际业务需求，参考日志服务提供的模板样例快速实现功能开发。

功能优势

- 开箱即用，免运维，无需部署即可进行对应的开发及调试工作。
- 预置日志服务Python SDK，全面实现开发工具与日志服务功能的无缝衔接。
- 内置丰富的业务场景模板和样例，引用便捷，可以快速构建功能需求。
- Jupyter Lab实例资源支持启用和停止操作，实例停止后不再产生费用，有效降低使用成本。
- 支持数据永久化保存。您可以将开发代码保存到Jupyter Lab实例挂载的NAS文件系统中，一次开发多次使用。

支持的日志服务功能场景

类别	子类	支持功能
资源管理	批量创建Project和Logstore	支持通过模板批量创建多个Project和Logstore。
	查询Project和Logstore列表	通过模板创建Logtail配置，同时支持跨Project复制Logtail配置。
	索引配置管理	支持Logstore的索引配置管理，同时支持批量查询账号下各个Logstore的索引配置。
任务管理类	投递任务管理	快速创建投递任务，无须通过人工配置即可实现投递任务的快速编排。
查询分析类	探索性数据分析	基于Jupyter Lab中内置阿里云成本分析和全球疫情分析Demo，可以快速获取所需的成本分析和疫情分析数据。
机器学习类	ECS指标预测	通过Jupyter Lab中内置的模板，模拟ECS指标数据，并结合机器学习预测函数对指标趋势进行预测。您可以快速直观地体验日志服务的机器学习预测功能，也可以根据实际业务将机器学习能力结合到业务场景。
	ECS指标异常检测	通过Jupyter Lab中内置的模板，模拟ECS指标数据，并结合机器学习预测函数对异常指标进行检测发现。您可以快速直观地体验日志服务的机器学习异常检测功能，也可以根据实际业务将日志服务异常检测能力结合到业务场景，并结合告警能力对异常进行通知。

类别	子类	支持功能
	ECS指标流式智能巡检	通过Jupyter Lab中内置的模板，将真实的ECS数据写入到日志服务，并调用机器学习流式巡检函数对数据进行建模，从而快速直观地体验日志服务的机器学习流式智能巡检功能。

实例说明

Jupyter Lab后端功能由PAI-DSW提供，因此使用Jupyter Lab前您需要创建对应的PAI-DSW个人版实例。PAI-DSW个人版实例基于阿里云Docker和Kubernetes等云原生技术，为您提供灵活且开放的AI开发环境。该版本的功能特点、实例规格及可用区、支持的镜像列表如下：

- 华东1（杭州）地域的CPU类型实例规格如下表所示。

规格	vCPU数量	内存（GiB	带宽（Gbps）	系统盘（GB）
ecs.c6.large	2	4	1	128
ecs.g6.large	2	8	1	128
ecs.g6.xlarge	4	16	1.5	256
ecs.g6.2xlarge	8	32	2.5	500
ecs.g6.4xlarge	16	64	5	500
ecs.g6.8xlarge	32	128	10	500

- 华东1（杭州）地域的GPU类型的实例规格如下表所示。

规格	vCPU数量	内存（GiB）	GPU	带宽（Gbps）	系统盘（GB）
ecs.gn6e-c12g1.12xlarge	48	368	4*NVIDIA V100	16	500
ecs.gn5-c4g1.xlarge	4	30	1*NVIDIA P100	3	256
ecs.gn5-c8g1.2xlarge	8	60	1*NVIDIA P100	3	500
ecs.gn5-c8g1.4xlarge	16	120	2*NVIDIA P100	5	500
ecs.gn5-c28g1.7xlarge	28	112	1*NVIDIA P100	5	500
cs.gn6v-c10g1.20xlarge	82	336	8 * NVIDIA V100	35	500

规格	vCPU数量	内存（GiB）	GPU	带宽（Gbps）	系统盘（GB）
ecs.gn6v-c8g1.16xlarge	64	256	8 * NVIDIA V100	20	500
ecs.gn6v-c8g1.2xlarge	8	32	1 * NVIDIA V100	2.5	500
ecs.gn6v-c8g1.8xlarge	32	128	4 * NVIDIA V100	10	500

- CPU类型的镜像列表如下表所示。

镜像名称	描述
py27_cpu_tf1.12_ubuntu	支持TensorFlow 1.12（CPU）版本
py36_cpu_tf2.1_torch1.4_ubuntu	支持TensorFlow 2.1和PyTorch 1.4（CPU）版本
ubuntu18.04-py3.6-paitf1.12	支持PAI-TensorFlow 1.12（CPU）版本
py36_cpu_tf1.15_ubuntu	支持TensorFlow 1.15（CPU）版本

- GPU类型的镜像列表如下表所示。

镜像名称	描述
py27_cuda90_tf1.12_ubuntu	支持TensorFlow 1.12（GPU）版本
py36_cuda101_tf2.1_torch1.4_ubuntu	支持TensorFlow 2.1和PyTorch 1.4（GPU）版本
ubuntu18.04-py3.6-cuda10.0-paitf1.12	支持PAI-TensorFlow 1.12（GPU）版本
py36_cuda100_tf1.15_ubuntu	支持TensorFlow 1.15（GPU）版本

操作流程

快速使用Jupyter Lab功能，基本操作流程如下：

1. 完成授权。具体操作，请参见授权。
2. 创建和启动Jupyter Lab实例。具体操作，请参见Jupyter Lab实例相关操作。
3. 登录Jupyter Lab，进行交互式开发。详情请参见开始编程。

   也可以直接引用日志服务内置的Demo库进行快速开发。详情请参见场景案例。
4. 保存并下载您的代码至本地，以备下次编写使用。

   如果创建Jupyter Lab实例时，已挂载NAS存储，则可以将代码直接存储到NAS。
5. 完成开发编程后，退出Jupyter Lab，停止Jupyter Lab实例。具体操作，请参见Jupyter Lab实例相关操作。

费用说明

Jupyter Lab由PAI-DSW提供服务，日志服务不收取额外费用。费用遵循PAI-DSW定价规则，更多信息，请参见PAI-DSW计费说明。

- 目前仅支持华东1（杭州）地域使用Jupyter Lab。当您使用Jupyter Lab跨地域读取日志时，将产生日志服务外网读取流量费用。更多信息，请参见计费项。
- 您通过挂载NAS实现数据永久化保存，将产生NAS费用。更多信息，请参见文件存储NAS产品定价。

Jupyter Lab使用PAI-DSW个人版，目前仅支持按量计费（后付费），折算规则如下：

```
账单金额=（定价/60）×使用时长（分钟）
```

13.2. 授权

使用Jupyter Lab前，您需要参考本文提供的指导进行相关授权。

步骤一：为阿里云账号授予PAI-DSW服务角色权限

为确保PAI-DSW能够正常提供服务，您需要为阿里云账号授予PAI-DSW服务角色权限。PAI-DSW使用此角色来访问您在其他云产品中的资源。请单击授权，按照提示完成授权操作。

步骤二：为关联角色授权

首次使用Jupyter Lab进行编程时，需要授予Jupyter Lab读取您云资源信息（例如读取日志库）的访问权限。

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**Jupyter Lab**。
3. 在**交互式建模（DSW）**页面，单击**创建实例**。
4. 在**角色授权**对话框，单击**去授权**。
5. 在**云资源访问授权**页面，单击**同意授权**。

   在**云资源访问授权**页面，系统自动配置Jupyter Lab需要的关联角色。

步骤三：为RAM用户授权

如果您使用RAM用户，需要使用阿里云账号授予RAM用户管理PAI-DSW实例的权限，包括创建、启动、停止及删除实例。

1. 登录RAM 控制台。
2. 创建自定义权限策略。
   i. 在左侧导航栏，选择**权限管理 > 权限管理策略**。
   ii. 在**权限管理策略**页面，单击**创建权限策略**。
   iii. 在**新建自定义权限策略**页面，配置如下参数。

参数	描述
策略名称	输入SLS_Notebook_Access。
备注	输入日志服务Jupyter Notebook访问策略。
配置模式	选择**脚本配置**。

参数	描述

策略内容	将配置框中的原有脚本替换为如下内容。
	`{ "Statement": [ { "Action": [ "notebook:CreateInstance", "notebook:StartInstance", "notebook:StopInstance", "notebook:EditInstance", "notebook:ListInstance" ], "Effect": "Allow", "Resource": "*" } ], "Version": "1" }`
	其中**Action**表示赋予的操作权限，可以包括以下权限：
	■ notebook:CreateInstance：创建Jupyter Lab实例。
	■ notebook:StartInstance：开启Jupyter Lab实例。
	■ notebook:StopInstance：停止Jupyter Lab实例。
	■ notebook:EditInstance：编辑Jupyter Lab实例。
	■ notebook:ListInstance：查看所有Jupyter Lab实例。
	Resource表示资源权限，配置方式包括：
	■ 指定实例的地域权限。
	`"Resource": "acs:notebook:cn-beijing:*:notebook/*"`
	■ 为特定实例（例如hhdemo）赋予Jupyter Lab的使用权限。
	`"Resource": "acs:notebook:*:*:notebook/hhdemo"`
	■ 为所有实例赋予Jupyter Lab的使用权限。
	`"Resource": "*"`
	授权策略语言的结构和语法请参见权限策略语法和结构。

iv. 单击**确定**。

3. 为RAM用户授权。
   i. 在左侧导航栏中，单击**人员管理 > 用户**。
   ii. 找到目标RAM用户，单击**添加权限**。
   iii. 在**添加权限**页面，选中自定义策略下的SLS_Notebook_Access，单击**确定**。

iv. 单击**完成**。

13.3. Jupyter Lab实例相关操作

Jupyter Lab由PAI-DSW提供服务，在使用Jupyter Lab进行交互式开发前，您需要创建PAI-DSW实例。本文介绍Jupyter Lab实例的相关操作。

前提条件

- 首次使用PAI-DSW，需要对相关资源进行访问授权。具体操作，请参见授权。
- 如果使用RAM用户创建实例，则需要阿里云账号为其授权。具体操作，请参见为RAM用户授权。

创建Jupyter Lab实例

1. 登录日志服务控制台。
2. 在**日志应用**区域，单击**Jupyter Lab**。
3. 在**交互式建模（DSW）**页面，单击**创建实例**。
4. 在**机器学习PAI DSW**页面，配置如下参数。

参数	描述
实例名称	实例名称。满足如下规则： ○ 以字母开头 ○ 长度不超27位 ○ 包含字母、数字和下划线（_）的字符串。
实例版本	选择**DSW个人版**。
地域及可用区	华东1（杭州）
付费模式	**DSW个人版**仅支持按量付费模式。
实例资源	根据您的实际业务需求选择对应实例资源。 详细CPU实例和GPU实例规格，请参见实例说明。
存储	实例自带系统盘存储为临时存储，停止或删除实例后，该存储清空。长期存储时，需要选择已创建的NAS文件系统进行挂载。如何创建NAS文件系统，请参见创建文件系统。 **说明** 挂载NAS文件系统，PAI-DSW将默认使用该NAS存储数据，不再使用临时存储。
实例镜像	支持的镜像包括Python、TensorFlow和PyTorch多个版本，详细的镜像列表请参见镜像列表。

参数	描述
专有网络	创建位于华东1（杭州）的专有网络。在VPC内使用PAI-DSW，您必须同时配置**专有网络**、**交换机**及**安全组**。 您可以直接选择已经创建的专有网络，或单击**专有网络**后的**创建专有网络**进行创建。如何创建专有网络，请参见使用专有网络。
交换机	配置**交换机**。 您可以直接选择已经创建的交换机，或单击**创建交换机**进行创建。如何创建交换机，请参见使用交换机。
安全组	配置**安全组**。 您可以直接选择已经创建的安全组，或单击**创建安全组**进行创建。如何创建安全组，请参见创建安全组。

5. 单击**确认订单**。
6. 核对订单信息，选中**《机器学习PAID SW服务条款》**复选框，并单击**创建实例**。

   创建完成后，实例默认处于启动状态。您可以打开实例进入编程环境。

相关操作

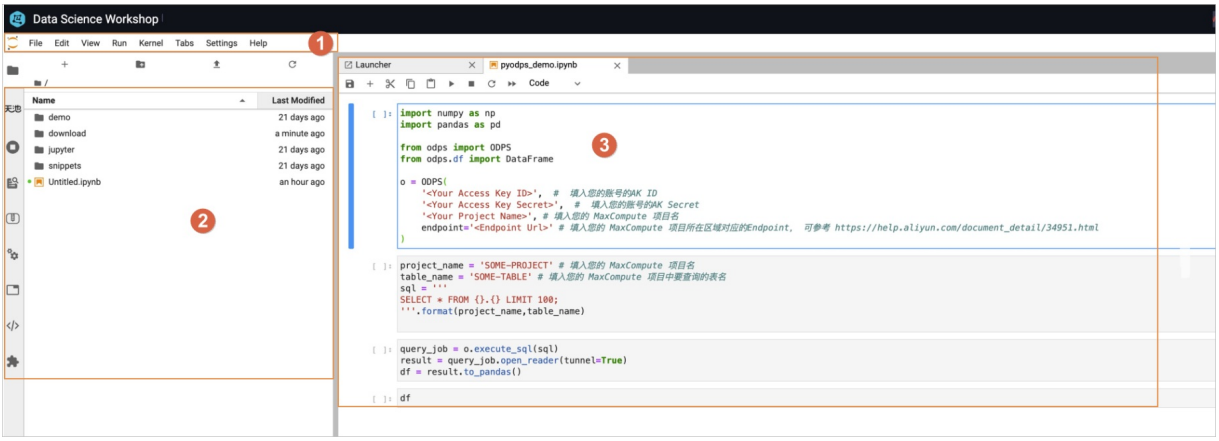
您还可以对Jupyter Lab实例进行如下操作。

操作	说明
启动	处于停止状态的实例，可启动实例。启动实例后，您可以打开实例进入编程环境，系统开始计费。
删除	您不再使用实例进行开发编程，则可以删除实例。 警告 实例删除后，其数据无法恢复，请谨慎操作。
停止	开发编程完成，退出Jupyter Lab后，停止实例，系统停止计费。支持如下两种停止方式： • **保存环境后停止** 如果您对默认环境进行了修改（例如安装了软件包或pip包），建议选择该方式。 • **直接停止** 如果未修改默认环境，通常选择该方式。

13.4. 界面介绍

日志服务DSW（Data Science Workshop）是日志服务为开发者推出的云端交互式编程环境IDE。开发者无需任何安装和运维配置，可快速编写、调试、运行Python代码，体验和使用日志服务的功能。

DSW界面由菜单栏、左边栏和主工作区组成。界面如下图所示。更多信息，请参见JupyterLab Doc。



序号	区域	说明
①	菜单栏	顶部菜单栏包含如下功能： • **File**：操作文件和目录。 • **Edit**：编辑文档和其他活动相关的操作。 • **View**：改变JupyterLab的外观视图。 • **Run**：运行Notebook和代码控制台的代码。 • **Kernel**：管理内核的操作。内核是运行代码的独立进程。 • **Tabs**：操作面板中已打开的文档和活动。 • **Settings**：常用设置和高级设置。 • **Help**：帮助参考信息。
②	左边栏	左边栏包含常用的选项卡，例如文件浏览器、正在运行的内核和终端列表、命令调色板以及主工作区中的选项卡列表。 •  ：文件浏览器。可以快速浏览、操作文件和文件夹，也支持一些操作文件的快捷操作。 •  ：正在运行的内核和终端列表。可以快速浏览正在运行的会话，也支持关闭会话。 •  ：支持命令列表。 •  ：参考教程。 •  ：管理内核的操作。内核是运行代码的独立进程。 •  ：处于打开状态的页签列表。可以快速浏览已打开的页签，也可以关闭页签。 如果您需要关注单个文档而同时不关闭主工作区的其他选项卡时，可以选择**View > Single-Document Mode**，开启选项卡简单界面模式。

序号	区域	说明
③	主工作区	您可以在主工作区创建编辑各类文档，例如Notebook、Text文件、Markdown文件等。也可创建命令行终端和Console等。新创建页签支持随意拖动布局。活动页签默认只有一个，在左边栏中默认蓝色底纹显示。 上下文菜单：提供丰富的快捷操作功能，在编辑Notebook、Text文件、Markdown等文件时，您随时可以单击鼠标右键获得快捷功能访问入口。如果需要使用浏览器的上下文菜单功能，您可以按住键盘Shift同时单击右键来获取访问入口。 键盘快捷键：您可以通过键盘快捷键操作Notebook。在菜单栏选择**Settings > Advanced Settings Editor**，单击**Keyboards Shortcuts**，即可自定义键盘快捷键。

日志服务内置机器学习、任务配置、数据分析和资源管理等场景案例，便于您体验、快速开发。在左边栏单击，可浏览、获取和下载这些场景案例。

13.5. 开始编程

开启PAI-DSW实例后，您可以直接进入交互式编程环境，开启编程。

步骤一：创建日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。

> **说明** 如果您要使用HTTPS连接，则需在endpoint中加入https://前缀，例如https://cn-hangzhou.log.aliyuncs.com。

```
# 设置LogClient
from aliyun.log.logclient import LogClient
# 日志服务的服务入口。更多信息，请参见服务入口。
# 此处以杭州为例，其它地域请根据实际情况填写。
endpoint = "cn-hangzhou.log.aliyuncs.com"
#阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
#创建LogClient。
client = LogClient(endpoint, accessId, accessKey)
```

步骤二：写入日志

```
# 向Logstore中写入日志。
import time
from aliyun.log.logitem import LogItem
from aliyun.log.putlogsrequest import PutLogsRequest
# Project和Logstore名称。
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
# 日志内容。
one_log_contents = []
one_log_contents.append(("msg1", "Hello"))
one_log_contents.append(("msg2", "World"))
#日志增加时间戳，并作为一条日志。
one_log = LogItem(
  timestamp=int(time.time()),
  contents=one_log_contents,
)
#日志
logs = [one_log]
#使用PutLogs方法写入日志库。
putReq = PutLogsRequest(
    project=project,
    logstore=logstore,
    logitems=logs,
)
res = client.put_logs(putReq)
#打印
print(res.get_body())
```

使用内置场景案例进行开发

日志服务已内置机器学习、任务配置、数据分析和资源管理等场景案例，便于您体验、快速开发。更多操作，请参见场景案例。

13.6. 场景案例

13.6.1. ECS指标流式智能巡检

通过Jupyter Lab中内置的模板，模拟ECS指标数据，并结合机器学习预测函数对指标趋势进行预测。本文介绍日志服务机器学习流式智能巡检的使用方法。

前提条件

- 已创建保存时序库结果的日志库。更多信息，请参见管理Logstore。
- 已创建接入模拟数据的时序库。更多信息，请参见管理MetricStore。

步骤一：接入模拟数据

接入模拟主机监控数据，用于ECS指标流式智能巡检。

1. 登录日志服务控制台。
2. 在Project列表区域，单击目标Project。
3. 在左侧导航栏中，单击图标。
4. 选择已创建时序库对应的**数据接入 > 模拟接入**，单击。
5. 在**接入数据**对话框中，单击**模拟**。
6. 在**主机监控模拟接入**页面，设置**时间间隔**为1s。
7. 单击**开始导入**。

步骤二：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

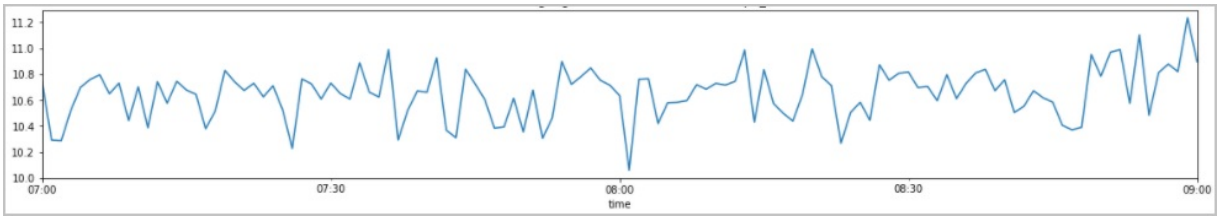
```
# Setup basic client
# !pip install -U matplotlib
import time
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import aliyun.log as sls
# SLS Endpoint列表。日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-beijing.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
#Project名称
project  = "YOUR_SLS_PROJECT"
# MetricStore名称。
metricstore = "YOUR_SLS_METRICSTORE"
#保存巡检结果的Logstore。
sink_logstore = 'YOUR_SLS_LOGSTORE_FOR_RESULTS_WRITE'
 #设置任务名称。
task_name = "YOUR_TASK_NAME"
#创建LogClient。
client = sls.LogClient(endpoint, accessId, accessKey)
```

步骤三：数据可视化

展示采集到的数据。需指定开始时间、结束时间和展示的指标名称。示例代码如下所示：

```
startTime = "2021-03-19 07:00:00"  # 设置模拟数据开始的时间，请根据当前模拟时间设置。
endTime = "2021-03-19 09:00:00"   # 设置可视化结束时间。
metric_name = 'cpu_util'       # 选择需要可视化的指标。更多信息，请参见指标说明。
request = sls.GetLogsRequest(project, metricstore, fromTime=startTime, toTime=endTime, topic='',
                  query="* | select promql_query_range('{}') from metrics limit 10000".format(metric_name),
                  line=500, offset=0, reverse=False)
res = client.get_logs(request)
df = []
for x in res.get_logs():
  item = {}
  for k, v in x.get_contents().items():
    if k == 'labels':
      tmp = json.loads(v)
      for k_, v_ in tmp.items():
        item[k_] = v_
    else:
      item[k] = v
  df.append(item)
df = pd.DataFrame(df)
df['time'] = pd.to_datetime(df['time'], unit='ms', utc=True).dt.tz_convert('Asia/Shanghai')
df = df.set_index('time')
df['value'] = df['value'].astype(np.float64)
for name, group in df.groupby(['hostname', 'ip']):
  group['value'].plot(title='{} - {} - {}'.format(name[0], name[1], metric_name), figsize=(20, 3))
  plt.show()
```

数据可视化后，如下图所示。



步骤四：流式智能巡检

对采集数据进行流式智能巡检。示例代码如下所示：

```
fromTime = "2021-3-19 00:00:00"
fromStamp = int(time.mktime(time.strptime(fromTime, "%Y-%m-%d %H:%M:%S")))
# 配置数据
data_meta = {
  "query": "",                      # the query to load time series data / no query to consume data
  "granularity": 60,                  # the granularity of time series. unit: second
  "columns": ["__time_nano__", "hostname", "ip",  # the keywords in time series data
        "cpu_util", "mem_util", "disk_util",
        "net_err_util", "system_load1"],
  "timestamp_name": "__time_nano__",        # a keyword in columns to indicate timestamp
  "entity_names": ["hostname", "ip"],       # a group of keywords in columns to indicate a curve entity
  "parent_names": ["hostname", "ip"],       # a group of keywords in entity_names to indicate a parent ent
ity, for trace analysis (optional)
  "child_names": [],                  # a group of keywords in entity_names to indicate a child entity, for trace a
nalysis (optional)
  "numeric_names": [                  # a category for each metric value, contain: name, upper value, down val
ue
    {
      "numeric_name": "cpu_util",
      "upper_limit": 1e64,
      "down_limit": -1e64
    },
    {
      "numeric_name": "mem_util",
      "upper_limit": 1e64,
      "down_limit": -1e64
    },
    {
      "numeric_name": "disk_util",
      "upper_limit": 1e64,
      "down_limit": -1e64
    },
    {
      "numeric_name": "net_err_util",
      "upper_limit": 1e64,
      "down_limit": -1e64
    },
    {
      "numeric_name": "system_load1",
      "upper_limit": 1e64,
      "down_limit": -1e64
    }
```

```
  ],
  "is_sparse": False,                    # a sign to indicate data structure
}
# 配置算法
algo_meta = {
  "algo_items":[                         # a group of algorithm configuations, one algorithm at least
    {
      "algo_type":7,                     # algorithm ID
      "refer_win_size":1200,             # a window size of time series for algorithm training
      "delay_interval":60,               # the interval for algorithm inference
      "parameter": json.dumps({          # algorithm paremeter
        "num_step": 10,                  # the number of value segmentation in time series
        "cycle": 2880                    # the cycle length of time series
      })
    }
  ]
}
# 配置计算资源
schedule_meta = {
  "from_stamp":fromStamp,                # the start timestamp for time series data
  "to_stamp":-1,                         # the end timestamp for time series data, default: -1
  "update_period":60,                    # the time duration for algorithm synchronization. unit:minute
  "tick_worker_number":1,                # the number of tick worker for fetching data
  "model_worker_number":1,               # the number of model worker for traning algorithm
  "cron_worker_number":1,                #
  "only_show_anomaly":True               # the sign to indicate whether only anomalies will be outputted or
not
}
# ETL配置
configuration = {
  'accessKeyId': accessId,
  'accessKeySecret': accessKey,
  'fromTime': fromStamp,
  'toTime': 0,
  'logstore': metricstore,
  'parameters': {
    "sls.config.job_mode": json.dumps({"type":"ml"}),
    "config.ml.data_meta": json.dumps(data_meta),
    "config.ml.algo_meta": json.dumps(algo_meta),
    "config.ml.schedule_meta": json.dumps(schedule_meta)
  },
  'roleArn': '',
  'script': '',
  'sinks': [
    {
      'accessKeyId': accessId,
      'accessKeySecret': accessKey,
      'endpoint': '',
      'logstore': sink_logstore,
      'name': 'test',
      'project': project,
      'roleArn': '',
    }
  ],
```

```
  'version': 2
}
schedule = {
  'type': 'Resident'
}
# 为sinklogstore设置索引
request_json = {'line':{'token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'caseSensitive':False,'chn':False},'keys':{'__tag__:__job_name__':{'type':'text','token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'caseSensitive':False,'alias':'job_name','doc_value':True,'chn':False},'entity':{'type':'json','token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'caseSensitive':False,'alias':'','doc_value':True,'chn':False,'index_all':True,'max_depth':-1,'json_keys':{}},'meta':{'type':'json','token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'caseSensitive':False,'alias':'','doc_value':True,'chn':False,'index_all':True,'max_depth':-1,'json_keys':{'logstore_name':{'alias':'','caseSensitive':False,'chn':False,'doc_value':True,'token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'type':'text'},'project_name':{'alias':'','caseSensitive':False,'chn':False,'doc_value':True,'token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'type':'text'}}},'result':{'type':'json','token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/','\n','\t','\r'],'caseSensitive':False,'alias':'','doc_value':True,'chn':False,'index_all':True,'max_depth':-1,'json_keys':{'dim_name':{'alias':'','caseSensitive':False,'chn':False,'doc_value':True,'token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'type':'text'},'is_anomaly':{'alias':'','caseSensitive':False,'chn':False,'doc_value':True,'token':[',',' ','"',"'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],'type':'text'},'score':{'alias':'','doc_value':True,'type':'double'}}}},'log_reduce':False,'max_text_len':2048}
request = sls.IndexConfig()
request.from_json(request_json)
res = client.create_index(project, sink_logstore, request)
job_name = "111-" + task_name
job_display_name = job_name
job_discription = task_name
# 创建ETL任务
client.create_etl(project, job_name, configuration, schedule, job_display_name, job_discription)
```

步骤五：获取智能巡检结果

获取流式智能巡检结果。示例代码如下所示：

```
startTime = "2021-3-19 12:00:00"
endTime = "2021-3-19 23:00:00"
query = "* | select date_trunc('second', __time__) as time, json_extract(entity, '$.hostname') as hostname, json_extract(entity, '$.ip') as ip, json_extract(result, '$.dim_name') as metric_name, json_extract(result, '$.score') as anomaly_score, json_extract(result, '$.anomaly_type') as anomaly_type from log order by time desc limit 10000"
request = sls.GetLogsRequest(project, sink_logstore, fromTime=startTime, toTime=endTime, topic='',
                             query=query, line=500, offset=0, reverse=False)
res = client.get_logs(request)
df = []
for x in res.get_logs():
  item = {}
  for k, v in x.get_contents().items():
    item[k] = v
  df.append(item)
df = pd.DataFrame(df)
```

获取巡检结果如下图所示。

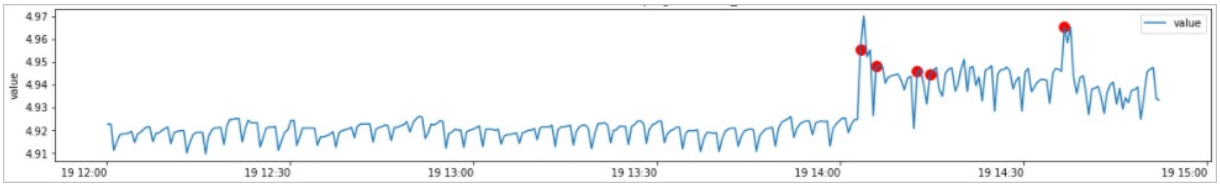
	time	hostname	ip	metric_name	anomaly_score	anomaly_type
0	2021-03-19 14:05:50.000	"i[redacted]Z"	"172.17.82.156"	"mem_util"	0.6773103436688586	"Variance"
1	2021-03-19 14:03:48.000	"i[redacted]Z"	"172.17.82.156"	"cpu_util"	0.7522210783779193	"Shift"
2	2021-03-19 14:03:15.000	"i[redacted]Z"	"172.17.82.156"	"mem_util"	0.8012835328429566	"Variance"
3	2021-03-19 13:56:02.000	[redacted]Z"	"192.168.0.33"	"system_load1"	0.6868585527924442	"Variance"
4	2021-03-19 13:31:00.000	[redacted]Z"	"10.1.138.110"	"cpu_util"	0.6392962733998253	"Stab"
5	2021-03-19 13:12:37.000	[redacted]Z"	"10.1.138.113"	"cpu_util"	0.6609754236695644	"Stab"
6	2021-03-19 13:00:20.000	[redacted]Z"	"10.1.138.110"	"cpu_util"	0.6489994550536586	"Stab"
7	2021-03-19 12:31:02.000	[redacted]Z"	"10.1.138.110"	"cpu_util"	0.7354376138180156	"Stab"

（可选）步骤六：获取智能巡检异常结果

获取流式智能巡检异常结果。示例代码如下所示：

```
hostname = "iZ2ze931vw5cx6kunqnhgdZ"
metric_name = "mem_util"
anomaly_score = 0.5
query = "* and result.score > "+str(anomaly_score)+" and entity.hostname: "+hostname+" and result.dim_name: "+metric_name+" | select time, value, case when score is null then 0 else score end as score, case when score is null then 0 else 1 end as label from (select A.time, A.value, B.score from (( SELECT __time_nano__/1000000 as time, __value__ as value FROM \"metric-test.prom\" WHERE element_at(__labels__, 'hostname')='"+hostname+"' and __name__='"+metric_name+"') as A left join (select __time__ as time, \"result.score\" as score from log) as B on A.time = B.time)) limit 10000"
request = sls.GetLogsRequest(project, sink_logstore, fromTime=startTime, toTime=endTime, topic='',
                             query=query, line=500, offset=0, reverse=False)
res = client.get_logs(request)
df = []
for x in res.get_logs():
  item = {}
  for k, v in x.get_contents().items():
    item[k] = v
  df.append(item)
df = pd.DataFrame(df)
df['time'] = pd.to_datetime(df['time'], unit='s', utc=True).dt.tz_convert('Asia/Shanghai')
df[['value', 'score', 'label']] = df[['value', 'score', 'label']].astype(np.float64)
figs, ax = plt.subplots(figsize=(20, 3))
df.plot(x='time', y='value', title='{} - {}'.format(hostname, metric_name), ax=ax)
df_ = df[df['label'] > 0]
df_.plot.scatter(x='time', y='value', s=100, c='red', ax=ax)
plt.show()
```

获取巡检异常结果如下图所示。



13.6.2. ECS指标异常检测

通过Jupyter Lab中内置的模板，模拟ECS指标数据，并结合机器学习预测函数，检测和发现ECS异常指标。本文介绍日志服务机器学习检测ECS异常指标的使用方法。

前提条件

- 已创建保存时序库结果的日志库。更多信息，请参见管理Logstore。
- 已创建接入模拟数据的时序库。更多信息，请参见管理MetricStore。

步骤一：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
# !pip install -U matplotlib
import time
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import aliyun.log as sls
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-beijing.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
# Project名称。
project  = "YOUR_SLS_PROJECT"
# MetricStore名称。
metricstore = "YOUR_SLS_METRICSTORE"
# 保存巡检结果的Logstore。
sink_logstore = 'YOUR_SLS_LOGSTORE_FOR_RESULTS_WRITE'
# 设置任务名称。
task_name = "YOUR_TASK_NAME"
# 创建LogClient。
client = sls.LogClient(endpoint, accessId, accessKey)
```

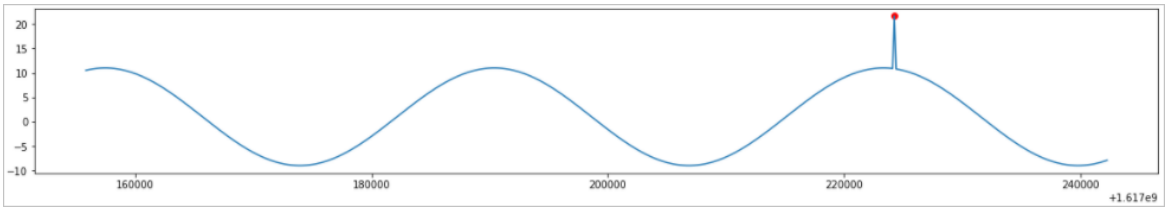
步骤二：写入ECS指标数据到日志服务

1. 通过代码生成ECS指标模拟数据。

   示例代码如下所示：

```
#定义生成数据方法。
class MockParam:
  def __init__(self, a, b, omega, phi, n_anomaly=1, is_show=False):
    self.a = a
    self.b = b
    self.omega = omega
    self.phi = phi
    self.n_anomaly = n_anomaly
    self.is_show = is_show
  def run(self, x_data):
    y_data = self.a * np.sin(self.omega / (2.0 * np.pi) * x_data + self.phi) + self.b
    a_idx = np.random.randint(len(x_data) // 2, len(x_data))
    y_data[a_idx] = 2.0 * y_data[a_idx]
    if self.is_show:
      plt.figure(figsize=(20, 3))
      plt.plot(x_data, y_data)
      plt.scatter(x_data[a_idx], y_data[a_idx], c='red')
      plt.show()
    return y_data
def gen_mock_cpu_data(st_time, ed_time, step=150, mock_param=None):
  st_time = st_time // step * step
  ed_time = st_time + (ed_time - st_time) // step * step
  n = (ed_time - st_time) // step + 1
  x_data = np.linspace(st_time, ed_time, n)
  if mock_param is not None:
    y_data = mock_param.run(x_data)
    return x_data, y_data
  return None, None
#调用并显示生成数据。
end_time = int(time.mktime(time.localtime()))
start_time = end_time - 24 * 60 * 60
mock_param = MockParam(10, 1, 10, 1, is_show=True)
x_data, y_data = gen_mock_cpu_data(start_time, end_time, mock_param=mock_param)
print(len(x_data))
print(len(y_data))
```

生成的ECS指标模拟数据，如下图所示。



2. 获取已创建的Project和Logstroe。

   示例代码如下所示：

```
import time
from aliyun.log.logitem import LogItem
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log import IndexConfig
# hostname: iZhp3fqc8ciryj43wtn76xZ
# metricname: cpu_sys_util
# value: 0.83
```

```
# value: 0.85
hostname = "iZhp3fqc8ciryj43wtn76xZ"
metric_name = "cpu_sys_util"
# 创建Project和Logstore。
print("create project & logstore")
try:
  res = client.get_project(project)
except:
  res = client.create_project(project, "a simple demo project")
try:
  res = client.get_logstore(project, logstore)
except:
  res = client.create_logstore(project, logstore, ttl=30, shard_count=2)
# 开启索引。
print("create index")
request_json = {
   "keys": {
    "hostname": {
     "caseSensitive": False,
     "token": [
      ",", " ", "\"", "\'", ";", "=", "(", ")", "[", "]",
      "{", "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
     ],
     "type": "text",
     "doc_value": True
    },
    "metricname": {
     "caseSensitive": False,
     "token": [
      ",", " ", "\"", "\'", ";", "=", "(", ")", "[", "]",
      "{", "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
     ],
     "type": "text",
     "doc_value": True
    },
    "value": {
     "doc_value": True,
     "type": "long"
    }
   },
   "storage": "pg",
   "ttl": 2,
   "index_mode": "v2",
   "line": {
    "caseSensitive": False,
    "token": [
     ",", " ", "\"", "\'", ";", "=", "(", ")", "[", "]", "{",
     "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
    ]
   }
 }
request = IndexConfig()
request.from_json(request_json)
res = client.create_index(project, logstore, request)
res.log_print()
```

```
print("wait for 1 minute")
time.sleep(60)
```

3. 将ECS指标模拟数据写入日志库。

   示例代码如下所示：

```
# upload data
print("upload data")
log_items = []
for x, y in zip(x_data, y_data):
  log_time = int(x)
  log_content = list()
  log_content.append(("hostname", "{}".format(hostname)))
  log_content.append(("metricname", metric_name))
  log_content.append(("value", "{}".format(y)))
  log_item = LogItem(timestamp=log_time, contents=log_content)
  log_items.append(log_item)
putReq = PutLogsRequest(project=project, logstore=logstore, topic=topic, logitems=log_items)
res = client.put_logs(putReq)
res.log_print()
```

步骤三：调测并输出异常结果

完成数据准备后，您可以编写代码调用机器学习函数，对数据进行分析，并输出异常检测结果。示例如下：

1. 查询日志。

   示例代码如下：

```
# Query SLS Logstore
query = '''metricname: cpu_sys_util and __topic__: ANOMALY_DETECTION_DEMO | select __time__ as t,
value as v from log limit 10000'''
datas = []
for i in client.get_log_all(project, logstore, start_time, end_time, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
df_ret = pd.DataFrame(datas)
print(df_ret)
```

   查询ECS指标异常数据，如下图所示。

	t	v
0	1617156000	10
1	1617156150	10
2	1617156300	10
3	1617156450	10
4	1617156600	10
..	...	..
570	1617241500	-8
571	1617241650	-8
572	1617241800	-8
573	1617241950	-8
574	1617242100	-8

2. 使用机器学习函数ts_predicate_ar进行检测，并输出。

   示例代码如下：

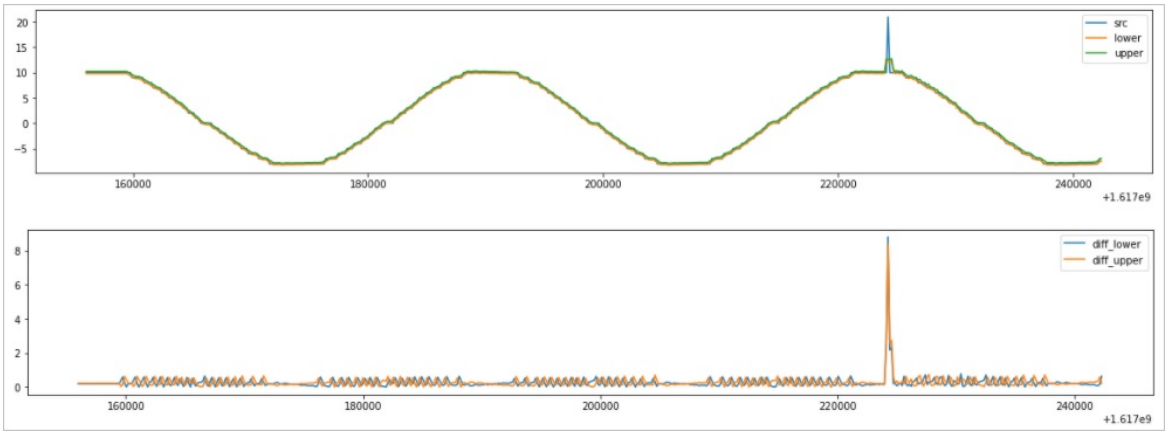
```
# Query SLS Logstore
query = '''metricname: cpu_sys_util and __topic__: ANOMALY_DETECTION_DEMO | select ts_predicate_
ar(t, v, 40, 1) from ( select __time__ as t, value as v from log ) limit 10000'''
datas = []
for i in client.get_log_all(project, logstore, start_time, end_time, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
df_ret = pd.DataFrame(datas)
print(df_ret)
# Visualize predicted values
# df_ret.set_index("unixtime", inplace=True)
df_ret = df_ret.astype("double")
df_ret["diff_lower"] = np.abs(df_ret["lower"] - df_ret["src"])
df_ret["diff_upper"] = np.abs(df_ret["upper"] - df_ret["src"])
print(df_ret)
```

3. 可视化展示。

   示例代码如下：

```
plt.figure(figsize=(20, 3))
plt.plot(df_ret["unixtime"], df_ret["src"], label='src')
plt.plot(df_ret["unixtime"], df_ret["lower"], label='lower')
plt.plot(df_ret["unixtime"], df_ret["upper"], label='upper')
plt.legend()
plt.show()
plt.figure(figsize=(20, 3))
plt.plot(df_ret["unixtime"], df_ret["diff_lower"], label='diff_lower')
plt.plot(df_ret["unixtime"], df_ret["diff_upper"], label='diff_upper')
plt.legend()
plt.show()
```

   数据可视化后，如下图所示。



13.6.3. ECS指标预测

通过Jupyter Lab中内置的模板，模拟ECS指标数据，并结合机器学习预测函数，预测指标趋势。本文介绍如何通过日志服务和机器学习来预测ECS指标的使用方法。

前提条件

- 已创建保存时序库结果的日志库。更多信息，请参见管理Logstore。
- 已创建接入模拟数据的时序库。更多信息，请参见管理MetricStore。

步骤一：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
# !pip install -U matplotlib
import time
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import aliyun.log as sls
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-beijing.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
# Project名称。
project  = "YOUR_SLS_PROJECT"
# MetricStore名称。
metricstore = "YOUR_SLS_METRICSTORE"
# 保存巡检结果的Logstore。
sink_logstore = 'YOUR_SLS_LOGSTORE_FOR_RESULTS_WRITE'
# 设置任务名称。
task_name = "YOUR_TASK_NAME"
# 创建LogClient。
client = sls.LogClient(endpoint, accessId, accessKey)
```

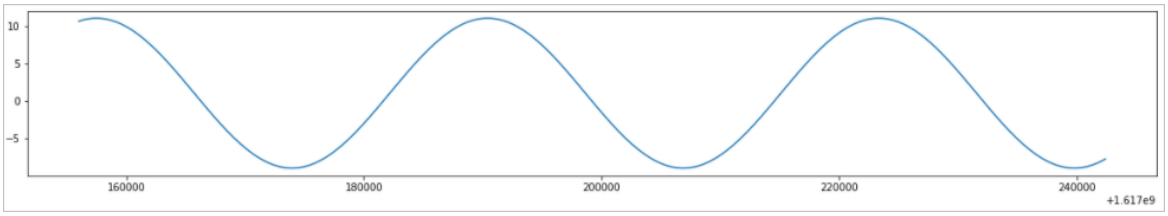
步骤二：写入ECS指标数据到日志服务

1. 通过代码生成ECS指标模拟数据。

   示例代码如下所示：

```
#定义生成数据方法。
class MockParam:
  def __init__(self, a, b, omega, phi, is_show=False):
    self.a = a
    self.b = b
    self.omega = omega
    self.phi = phi
    self.is_show = is_show
  def run(self, x_data):
    y_data = self.a * np.sin(self.omega / (2.0 * np.pi) * x_data + self.phi) + self.b
    if self.is_show:
      plt.figure(figsize=(20, 3))
      plt.plot(x_data, y_data)
      plt.show()
    return y_data
def gen_mock_cpu_data(st_time, ed_time, step=150, mock_param=None):
  st_time = st_time // step * step
  ed_time = st_time + (ed_time - st_time) // step * step
  n = (ed_time - st_time) // step + 1
  x_data = np.linspace(st_time, ed_time, n)
  if mock_param is not None:
    y_data = mock_param.run(x_data)
    return x_data, y_data
  return None, None
#调用并显示生成数据。
end_time = int(time.mktime(time.localtime()))
start_time = end_time - 24 * 60 * 60
mock_param = MockParam(10, 1, 10, 1, is_show=True)
x_data, y_data = gen_mock_cpu_data(start_time, end_time, mock_param=mock_param)
print(len(x_data))
print(len(y_data))
```

生成的ECS指标模拟数据，如下图所示。



2. 获取已创建的Project和Logstroe。

   示例代码如下所示：

```
import time
from aliyun.log.logitem import LogItem
from aliyun.log.putlogsrequest import PutLogsRequest
from aliyun.log import IndexConfig
# hostname: iZhp3fqc8ciryj43wtn76xZ
# metricname: cpu_sys_util
# value: 0.83
hostname = "iZhp3fqc8ciryj43wtn76xZ"
metric_name = "cpu_sys_util"
# 获取Project和Logstore。
print("create project & logstore")
```

```
print("create project & logstore")
try:
  res = client.get_project(project)
except:
  res = client.create_project(project, "a simple demo project")
try:
  res = client.get_logstore(project, logstore)
except:
  res = client.create_logstore(project, logstore, ttl=30, shard_count=2)
# 开启索引。
print("create index")
request_json = {
   "keys": {
    "hostname": {
     "caseSensitive": False,
     "token": [
      ",", " ", "\"", "\"", ";", "=", "(", ")", "[", "]",
      "{", "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
     ],
     "type": "text",
     "doc_value": True
    },
    "metricname": {
     "caseSensitive": False,
     "token": [
      ",", " ", "\"", "\"", ";", "=", "(", ")", "[", "]",
      "{", "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
     ],
     "type": "text",
     "doc_value": True
    },
    "value": {
     "doc_value": True,
     "type": "long"
    }
   },
   "storage": "pg",
   "ttl": 2,
   "index_mode": "v2",
   "line": {
    "caseSensitive": False,
    "token": [
     ",", " ", "\"", "\"", ";", "=", "(", ")", "[", "]", "{",
     "}", "?", "@", "&", "<", ">", "/", ":", "\n", "\t"
    ]
   }
 }
request = IndexConfig()
request.from_json(request_json)
res = client.create_index(project, logstore, request)
res.log_print()
print("wait for 1 minute")
time.sleep(60)
```

3. 将ECS指标数据写入日志库。

   示例代码如下所示：

```
# 写入日志数据。
print("upload data")
log_items = []
for x, y in zip(x_data, y_data):
  log_time = int(x)
  log_content = list()
  log_content.append(("hostname", "{}".format(hostname)))
  log_content.append(("metricname", metric_name))
  log_content.append(("value", "{}".format(y)))
  log_item = LogItem(timestamp=log_time, contents=log_content)
  log_items.append(log_item)
putReq = PutLogsRequest(project=project, logstore=logstore, topic=topic, logitems=log_items)
res = client.put_logs(putReq)
res.log_print()
```

步骤三：调测并输出预测结果

完成数据准备后，您可以编写代码调用机器学习函数，对数据进行预测分析，并输出预测结果。

1. 查询日志。

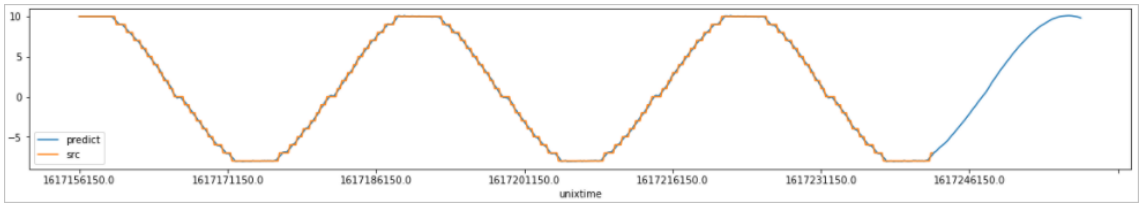
   示例代码如下：

```
# Query SLS Logstore
query = '''__topic__: PREDICT_DEMO and metricname: cpu_sys_util | select __time__ as t, value, hostna
me from log order by t limit 10000'''
datas = []
for i in client.get_log_all(project, logstore, start_time, end_time, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
df_ret = pd.DataFrame(datas)
print(df_ret)
```

2. 使用ts_predicate_ar或者ts_predicate_arma函数进行预测，并输出结果。
   - 使用机器学习函数ts_predicate_ar进行预测，示例代码如下：

```
query = '''__topic__: PREDICT_DEMO and metricname: cpu_sys_util | select ts_predicate_ar(t, value,
40, 100) from ( select __time__ as t, value from log order by t ) limit 10000'''
datas = []
for i in client.get_log_all(project, logstore, start_time, end_time, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
df_ret = pd.DataFrame(datas)
# Visualize predicted values
df_ret.set_index("unixtime", inplace=True)
df_ret = df_ret.astype("double")
print(df_ret)
df_ret[["predict", "src"]].plot(figsize=(20, 3))
```

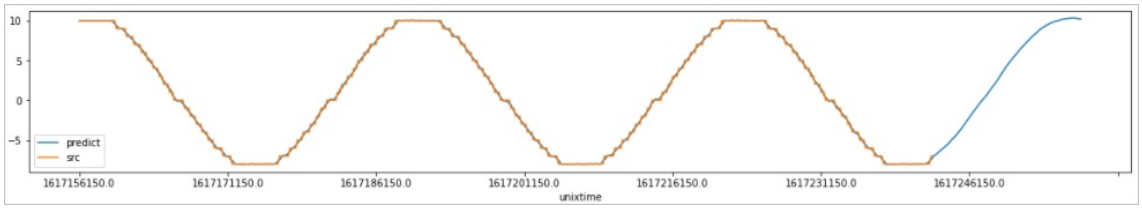
   ECS指标预测趋势，如下图所示。



- 使用机器学习函数ts_predicate_arma进行预测，示例代码如下。

```
query = '''__topic__: PREDICT_DEMO and metricname: cpu_sys_util | select ts_predicate_arma(t, valu
e, 50, 1, 100) from ( select __time__ as t, value from log order by t ) limit 10000'''
datas = []
for i in client.get_log_all(project, logstore, start_time, end_time, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
df_ret = pd.DataFrame(datas)
# Visualize predicted values
df_ret.set_index("unixtime", inplace=True)
df_ret = df_ret.astype("double")
print(df_ret)
df_ret[["predict", "src"]].plot(figsize=(20, 3))
```

ECS指标预测趋势，如下图所示。



13.6.4. 创建投递任务

通过内置的投递任务模板，快速创建投递任务。本文介绍创建投递任务的操作方法。

步骤一：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
# !pip install -U matplotlib
import time
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import aliyun.log as sls
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-beijing.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
# Project名称。
project  = "YOUR_SLS_PROJECT"
# MetricStore名称。
metricstore = "YOUR_SLS_METRICSTORE"
# 保存巡检结果的Logstore。
sink_logstore = 'YOUR_SLS_LOGSTORE_FOR_RESULTS_WRITE'
# 设置任务名称。
task_name = "YOUR_TASK_NAME"
# 创建LogClient。
client = sls.LogClient(endpoint, accessId, accessKey)
```

步骤二：查询当前Logstore的投递任务及配置

在创建投递任务前，查询已存在的投递任务及配置。示例代码如下所示：

```
# 查询投递任务配置。
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
ret = client.list_shipper(project, logstore)
for shipper_name in ret.get_body()['shipper']:
  ret = client.get_shipper(project, logstore, shipper_name)
  print("----------------------------------------")
  print(shipper_name)
  print(ret.get_body())
```

步骤三：创建投递任务

- 创建MaxCompute投递任务

  示例代码如下所示：

```
# 创建投递任务。
from aliyun.log.logexception import LogException
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
# 投递参数的具体配置。更多信息，请参见通过日志服务投递日志到MaxCompute。
shipper_config = {
    'shipperName': 'test_ship22',
    'targetType': 'odps',
    'targetConfiguration': {
      'bufferInterval': 1800,
      'enable': True,
      'fields': ['__time__', '__source__', '__topic__', 'content'],
      'odpsEndpoint': 'http://odps-ext.aliyun-inc.com/api',
      'odpsProject': 'TS_DL',
      'odpsTable': 'test_odps',
      'partitionColumn': ['__time__'],
      'partitionTimeFormat': 'yyyy_MM_dd_HH_mm'
    }
}
try:
  client.create_shipper(project, logstore, shipper_config)
except LogException as ex:
  if 'shipperName already exists' in ex.get_error_message():
    # create index if index not exists
    ret = client.update_shipper(project, logstore, shipper_config)
  else:
    raise ex
```

- 创建OSS投递任务

  示例代码如下：

```
# 创建投递任务。
# Create Logstore Shipper
from aliyun.log.logexception import LogException
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
# 投递参数的具体配置。更多信息，请参见将日志服务数据投递到OSS。
shipper_config = {
    'shipperName': 'to-oss',
    'targetConfiguration': {
        'bufferInterval': 300,
        'bufferSize': 256,
        'compressType': 'snappy',
        'enable': True,
        'ossBucket': 'YOUR_oss-bucket',
        'ossPrefix': 'prefix',
        'pathFormat': '%Y/%m/%d/%H/%M',
        'roleArn': 'acs:ram::YOUR_ALIYUN_UID:role/aliyunlogdefaultrole',
        'storage': {'detail': {'enableTag': False}, 'format': 'json'}
    },
    'targetType': 'oss'}
# 用户角色标识（ARN）。更多信息，请参见常见问题。
try:
  client.create_shipper(project, logstore, shipper_config)
except LogException as ex:
  if 'shipperName already exists' in ex.get_error_message():
    # create index if index not exists
    ret = client.update_shipper(project, logstore, shipper_config)
  else:
    raise ex
```

13.6.5. 阿里云成本分析

通过内置阿里云成本分析模板，可以快速获取所需的成本分析数据。本文介绍日志服务阿里云成本分析的使用方法。

前提条件

已开通日志服务成本管家。更多操作，请参见成本管家。

步骤一：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
# !pip install -U matplotlib
import time
import json
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import aliyun.log as sls
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-beijing.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
# Project名称。
project  = "YOUR_SLS_PROJECT"
# MetricStore名称。
metricstore = "YOUR_SLS_METRICSTORE"
# 保存巡检结果的Logstore。
sink_logstore = 'YOUR_SLS_LOGSTORE_FOR_RESULTS_WRITE'
# 设置任务名称。
task_name = "YOUR_TASK_NAME"
# 创建LogClient。
client = sls.LogClient(endpoint, accessId, accessKey)
```

步骤二：获取账单数据

示例代码如下所示：

```
import time
import pandas as pd
import matplotlib.pyplot as plt
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
def query_logstore(query, stime, etime):
  datas = []
  for i in client.get_log_all(project, logstore, stime, etime, query=query):
    for log in i.logs:
      datas.append(log.get_contents())
  # Convert datas to pandas dataframe
  df_ret = pd.DataFrame(datas)
  return df_ret
```

步骤三：统计和分析账单

- 查看账单总体趋势

  示例代码如下：

```
%matplotlib inline
etime = int(time.time())
stime = etime - 86400 * 7
query = '''
source:bill |
select date_format(__time__,'%Y-%m-%d') as ds, round(sum(PretaxAmount),3) as cost group by ds order by ds
'''
df_bill = query_logstore(query, stime, etime)
df_bill['cost'] = df_bill['cost'].astype(float)
#  Plot Total Cost Bar
figure = plt.figure(figsize=(18,5),dpi=98)
plt.title(u"Total Cost", fontproperties='SimHei',fontsize = 15)
plt.bar(df_bill['ds'], df_bill['cost'],label='Cost(CNY)')
plt.legend(loc='best')
```

- 统计云产品费用

  示例代码如下：

```
%matplotlib inline
etime = int(time.time())
stime = etime - 86400 * 7
query = '''
source:bill |
select ProductName,ProductCode, round(sum(PretaxAmount),3) as cost from log group by ProductCode, ProductName
'''
df_product_bill = query_logstore(query, stime, etime)
df_product_bill['cost'] = df_product_bill['cost'].astype(float)
df_product_bill.sort_values(by=['cost'], ascending=False).head(10)
```

- 查看ECS一周账单趋势

  示例代码如下：

```
%matplotlib inline
etime = int(time.time())
stime = etime - 86400 * 7
query = '''
source:bill and ProductCode:ecs |
select date_format(__time__,'%Y-%m-%d') as ds, round(sum(PretaxAmount),3) as cost group by ds order by ds
'''
df_ecs_bill = query_logstore(query, stime, etime)
df_ecs_bill['cost'] = df_ecs_bill['cost'].astype(float)
# Plot ECS Cost Bar
figure = plt.figure(figsize=(18,5),dpi=98)
plt.title(u"ECS Cost", fontproperties='SimHei',fontsize = 15)
plt.bar(df_ecs_bill['ds'], df_ecs_bill['cost'],label='Cost(CNY)')
plt.legend(loc='best')
```

- 查看ECS一周账单明细

  示例代码如下：

```
etime = int(time.time())
stime = etime - 86400 * 7
query = '''
ProductCode:ECS and source: instance_bill |
select BillingDate,Region,instanceid,Nickname,BillingItem,InstanceConfig,IntranetIP,PretaxAmount limit
10000
'''
df_ecs_bill_detail = query_logstore(query, stime, etime)
df_ecs_bill_detail['PretaxAmount'] = df_ecs_bill_detail['PretaxAmount'].astype(float)
df_ecs_bill_detail.sort_values(by=['PretaxAmount'],ascending=False)[['BillingDate','Region','Nickname','I
nstanceConfig','PretaxAmount']].head(10)
```

13.6.6. 查询账号下各Logstore索引配置

查看已开启索引的Logstore列表和索引字段信息，有助于您梳理日志库索引的配置情况，便于您对索引进行增减配置。本文介绍如何批量查询索引配置的操作方法。

步骤一：获取服务入口

示例代码如下所示：

```
# Setup basic client
from aliyun.log.logclient import LogClient
# 日志服务的服务入口。更多信息，请参见服务入口。
ALL_ENDPOINTS = ["https://ap-northeast-1.log.aliyuncs.com",
        "https://ap-south-1.log.aliyuncs.com",
        "https://ap-southeast-1.log.aliyuncs.com",
        "https://ap-southeast-2.log.aliyuncs.com",
        "https://ap-southeast-3.log.aliyuncs.com",
        "https://ap-southeast-5.log.aliyuncs.com",
        "https://cn-beijing.log.aliyuncs.com",
        "https://cn-chengdu.log.aliyuncs.com",
        "https://cn-guangzhou.log.aliyuncs.com",
        "https://cn-hangzhou.log.aliyuncs.com",
        "https://cn-heyuan.log.aliyuncs.com",
        "https://cn-hongkong.log.aliyuncs.com",
        "https://cn-huhehaote.log.aliyuncs.com",
        "https://cn-north-2-gov-1.log.aliyuncs.com",
        "https://cn-qingdao.log.aliyuncs.com",
        "https://cn-shanghai.log.aliyuncs.com",
        "https://cn-shenzhen.log.aliyuncs.com",
        "https://cn-wulanchabu.log.aliyuncs.com",
        "https://cn-zhangjiakou.log.aliyuncs.com",
        "https://eu-central-1.log.aliyuncs.com",
        "https://eu-west-1.log.aliyuncs.com",
        "https://me-east-1.log.aliyuncs.com",
        "https://rus-west-1.log.aliyuncs.com",
        "https://us-east-1.log.aliyuncs.com",
        "https://us-west-1.log.aliyuncs.com"]
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
```

步骤二：查询日志库

示例代码如下：

```
import time
import pandas as pd
from aliyun.log.logexception import LogException
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
index_config = None
def get_logstore_index(endpoint):
  client = LogClient(endpoint, accessId, accessKey)
  # 获取Project清单。
  res = client.list_project()
  projects = res.get_projects()
  # 获取每个Project的日志库Logstore。
  datas = []
  for i in projects:
    project = i['projectName']
    res = client.list_logstores(ListLogstoresRequest(project=project))
    for logstore in res.get_logstores():
      try:
        index_config = client.get_index_config(project , logstore)
        index_config_json = index_config.get_index_config().to_json()
      except LogException as ex:
        if ex.get_error_code() == 'IndexConfigNotExist':
          continue
      if 'keys' in index_config_json:
        for column in index_config_json['keys'].keys():
          datas.append({
            'region' : i['region'],
            'project' : project,
            'logstore' : logstore,
            'index_type' : 'column',
            'index' : column
          })
      if 'log_reduce' in index_config_json and index_config_json['log_reduce'] == True:
        datas.append({
            'region' : i['region'],
            'project' : project,
            'logstore' : logstore,
            'index_type' : 'logreduce',
            'index' : "True"
        })
      if 'line' in index_config_json:
        datas.append({
            'region' : i['region'],
            'project' : project,
            'logstore' : logstore,
            'index_type' : 'fullindex',
            'index' : "True"
        })
  return datas
datas = []
for endpoint in ALL_ENDPOINTS:
  datas.extend(get_logstore_index(endpoint))
```

```
    datas.extend(get_logstore_index(endpoint))
df_all_logstore_index_detail = pd.DataFrame(datas)
df_all_logstore_with_index = df_all_logstore_index_detail.groupby(['region',
                             'project']).agg({'logstore':'max'}).reset_index()
# 展示具体Logstore的所有索引配置。
df_all_logstore_index_detail
# 已开启索引的Logstore列表。
df_all_logstore_with_index
```

步骤三：将索引配置信息导出到文件

示例代码如下：

```
# 将已开启索引的Logstore列表导出到文件。
df_all_logstore_with_index.to_csv("all_logstore_with_index.csv",sep=';',index=False)
# Logstore中已开启索引的字段详情导出到文件。
df_all_logstore_index_detail.to_csv("all_logstore_index_detail.csv",sep=';',index=False)
```

13.6.7. 查询Project和Logstore列表

本文介绍批量查询Project和Logstore的操作方法。

步骤一：获取服务入口

示例代码如下所示：

```
# Setup basic client
from aliyun.log.logclient import LogClient
# 日志服务的服务入口。更多信息，请参见服务入口。
ALL_ENDPOINTS = ["https://ap-northeast-1.log.aliyuncs.com",
            "https://ap-south-1.log.aliyuncs.com",
            "https://ap-southeast-1.log.aliyuncs.com",
            "https://ap-southeast-2.log.aliyuncs.com",
            "https://ap-southeast-3.log.aliyuncs.com",
            "https://ap-southeast-5.log.aliyuncs.com",
            "https://cn-beijing.log.aliyuncs.com",
            "https://cn-chengdu.log.aliyuncs.com",
            "https://cn-guangzhou.log.aliyuncs.com",
            "https://cn-hangzhou.log.aliyuncs.com",
            "https://cn-heyuan.log.aliyuncs.com",
            "https://cn-hongkong.log.aliyuncs.com",
            "https://cn-huhehaote.log.aliyuncs.com",
            "https://cn-north-2-gov-1.log.aliyuncs.com",
            "https://cn-qingdao.log.aliyuncs.com",
            "https://cn-shanghai.log.aliyuncs.com",
            "https://cn-shenzhen.log.aliyuncs.com",
            "https://cn-wulanchabu.log.aliyuncs.com",
            "https://cn-zhangjiakou.log.aliyuncs.com",
            "https://eu-central-1.log.aliyuncs.com",
            "https://eu-west-1.log.aliyuncs.com",
            "https://me-east-1.log.aliyuncs.com",
            "https://rus-west-1.log.aliyuncs.com",
            "https://us-east-1.log.aliyuncs.com",
            "https://us-west-1.log.aliyuncs.com"]
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
```

步骤二：查询Project和Logstore列表

示例代码如下：

```
import time
import pandas as pd
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
def get_projects(endpoint):
  client = LogClient(endpoint, accessId, accessKey)
  # Get Project List
  res = client.list_project()
  projects = res.get_projects()
  # 查询每个Project下的Logstore列表。
  datas = []
  for i in projects:
    res = client.list_logstores(ListLogstoresRequest(project=i['projectName']))
    for logstore in res.get_logstores():
      datas.append({
        'region' : i['region'],
        'project' : i['projectName'],
        'logstore' : logstore,
      })
      time.sleep(0.01)
  return datas
datas = []
for endpoint in ALL_ENDPOINTS:
  print("get projects from %s" % endpoint)
  datas.extend(get_projects(endpoint))
df_project_logstores = pd.DataFrame(datas)
# 可视化展示所有Project和Logstore列表。
df_project_logstores
```

13.6.8. 批量创建Project和Logstore

本文介绍批量创建Project和Logstore的操作方法。

步骤一：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
from aliyun.log.logclient import LogClient
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-huhehaote.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
client = LogClient(endpoint, accessId, accessKey)
```

步骤二：创建Project和Logstore

示例代码如下：

1. 批量配置Project和Logstore信息。

```
import time
from aliyun.log.listlogstoresrequest import ListLogstoresRequest
# config project and logstore to create
project_logstores = {
  "your-project-name" : {
    "project_description" : "project description",
    "logstores" : ["logstore1","logstore2"]
  }
}
```

2. 批量创建Project和Logstore。

```
# Get Project List
def get_project_names():
  res = client.list_project()
  projects = res.get_projects()
  return [i['projectName'] for i in projects]
# Get Logstore under project
def get_project_logstores(project):
  res = client.list_logstores(ListLogstoresRequest(project=project))
  return res.get_logstores()
project_names = get_project_names()
for i in project_logstores:
  project = i
  project_description = project_logstores[i]['project_description']
  logstores = project_logstores[i]['logstores']
  if project not in project_names:
    client.create_project(project, project_description)
    time.sleep(0.1)
    print("create project %s" % project)
  else:
    print("project %s is already exists" % project)
  logstores_now = get_project_logstores(project)
  for logstore in logstores:
    if logstore not in logstores_now:
      print("create logstore %s" % logstore)
      client.create_logstore(project, logstore)
    else:
      print("logstore %s is already exists" % logstore)
```

13.6.9. 常见资源管理

本文介绍常用的日志服务资源管理代码操作示例。

示例1：初始化日志服务Client

LogClient是日志服务的Python客户端，用于管理Project、Logstore等日志服务资源。使用Python SDK发起日志服务请求，您需要初始化一个Client实例。示例代码如下所示：

```
# Setup basic client
from aliyun.log.logclient import LogClient
# 日志服务的服务入口。更多信息，请参见服务入口。
endpoint = "cn-huhehaote.log.aliyuncs.com"
# 阿里云访问密钥AccessKey ID和AccessKey Secret。更多信息，请参见访问密钥。
accessId = "YOUR_ACCESS_ID"
accessKey= "YOUR_ACCESS_KEY"
client = LogClient(endpoint, accessId, accessKey)
```

示例2：写入日志数据

示例代码如下：

```
# Write Log to Logstore
import time
from aliyun.log.logitem import LogItem
from aliyun.log.putlogsrequest import PutLogsRequest
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
one_log_contents = []
one_log_contents.append(("msg1", "Hello")) # log key value in one logitem
one_log_contents.append(("msg2", "World"))
one_log = LogItem(
  timestamp=int(time.time()),
  contents=one_log_contents,
)
logs = [one_log]
putReq = PutLogsRequest(
    project=project,
    logstore=logstore,
    logitems=logs,
)
res = client.put_logs(putReq)
print(res.get_body())
```

示例3：查询日志

示例代码如下：

```
# Query SLS Logstore
import time
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
etime = int(time.time())
stime = etime - 3600
query = '*'
datas = []
for i in client.get_log_all(project, logstore, stime, etime, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
print(datas)
```

可视化展示查询日志，代码示例如下：

```
# Query SLS Logstore to Pandas
import time
import pandas as pd
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
etime = int(time.time())
stime = etime - 3600
query = '* | select msg1 from log'
datas = []
for i in client.get_log_all(project, logstore, stime, etime, query=query):
  for log in i.logs:
    datas.append(log.get_contents())
# Convert datas to pandas dataframe
df_ret = pd.DataFrame(datas)
print(df_ret)
```

示例4：查询Project列表

示例代码如下：

```
# List SLS Project
ret = client.list_project()
ret.get_projects()
```

示例5：查询Logstore列表

示例代码如下：

```
# List SLS Logstore
project  = "YOUR_SLS_PROJECT"
ret = client.list_logstore(project)
print(ret.get_logstores())
```

示例6：创建Logstore

示例代码如下：

```
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
ret = client.create_logstore(project, logstore)
ret.log_print()
```

示例7：创建或更新Logstore索引配置

示例代码如下：

```
# Create or Update SLS Logstore Index
from aliyun.log.logexception import LogException
from aliyun.log.index_config import IndexConfig
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
request_json = {
  'index_mode': 'v2',
  'line': {
    'caseSensitive': False,
    'chn': False,
    'token': [',',' ','"","'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r']
  },
  'log_reduce': False,
}
req = IndexConfig()
req.from_json(request_json)
try:
  client.get_index_config(project_b, logstore_b)
  # update index config if index config is already exists
  ret = client.update_index(project_b, logstore_b, req)
except LogException as ex:
  if ex.get_error_code() in ("IndexConfigNotExist"):
    # create index if index not exists
    ret = client.create_index(project_b, logstore_b, req)
ret.log_print()
```

创建或更新Logstore列索引，代码示例如下：

```
# Create or Update SLS Logstore Index With Column Index
from aliyun.log.logexception import LogException
from aliyun.log.index_config import IndexConfig
project  = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
request_json = {
  'index_mode': 'v2',
  'line': {
    'caseSensitive': False,
    'chn': False,
    'token': [',',' ',"""","'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r']
  },
  'log_reduce': False,
  'keys' : {
    "msg1" : {
      "type": "text",
      'token': [',',' ',"""","'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],
      "doc_value": True
    },
    "long_column" : {
      "type": "long",
      "doc_value": True
    },
    "double_column" : {
      "type": "double",
      "doc_value": True
    },
    "json_column" : {
      "type": "json",
      'token': [',',' ',"""","'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],
      "doc_value": True,
      'json_keys': {
        "key1" : {
          "type": "text",
          'token': [',',' ',"""","'",';','=','(',')','[',']','{','}','?','@','&','<','>','/',':','\n','\t','\r'],
          "doc_value": True
        }
      }
    },
  }
}
req = IndexConfig()
req.from_json(request_json)
try:
  client.get_index_config(project_b, logstore_b)
  # update index config if index config is already exists
  ret = client.update_index(project_b, logstore_b, req)
except LogException as ex:
  if ex.get_error_code() in ("IndexConfigNotExist"):
    # create index if index not exists
    ret = client.create_index(project_b, logstore_b, req)
ret.log_print()
```

示例8：跨Logstore复制索引配置

示例代码如下：

```
# Copy SLS Logstore A's index config to Logstore B
from aliyun.log.logexception import LogException
from aliyun.log.index_config import IndexConfig
project_a = "YOUR_SLS_PROJECT_A"
logstore_a = "YOUR_SLS_LOGSTORE_A"
project_a = "YOUR_SLS_PROJECT_B"
logstore_a = "YOUR_SLS_LOGSTORE_B"
ret = client.get_index_config(project_a, logstore_a)
req = IndexConfig()
req.from_json(ret.get_body())
try:
  client.get_index_config(project_b, logstore_b)
  # update index config if index config is already exists
  ret = client.update_index(project_b, logstore_b, req)
except LogException as ex:
  if ex.get_error_code() in ("IndexConfigNotExist"):
    # create index if index not exists
    ret = client.create_index(project_b, logstore_b, req)
```

示例9：跨Project复制Logtail配置

示例代码如下：

```
# Copy logtail configs between projects
project_a = "YOUR_SLS_PROJECT_A"
project_b = "YOUR_SLS_PROJECT_B"
logtail_config_name = "YOUR_LOGTAIL_CONFIG_NAME"
ret = client.get_logtail_config(project_a, logtail_config)
ret = client.create_logtail_config(project_b, ret.logtail_config)
ret.log_print()
```

示例10：跨Project复制Dashboard配置

示例代码如下：

```
# Copy dashboard between projects
project_a = "YOUR_SLS_PROJECT_A"
project_b = "YOUR_SLS_PROJECT_B"
dashboard_name = "YOUR_DASHBOARD_NAME"
ret = client.get_dashboard(project_a, dashboard_name)
ret = client.create_dashboard(project_b, ret.get_body())
ret.log_print()
```

示例11：跨Project复制告警配置

示例代码如下：

```
# Copy alert between projects
project_a = "YOUR_SLS_PROJECT_A"
project_b = "YOUR_SLS_PROJECT_B"
alert_name = "YOUR_ALERT_NAME"
ret = client.get_alert(project_a, alert_name)
ret = client.create_alert(project_b, ret.get_body())
ret.log_print()
```

示例12：创建日志投递任务

示例代码如下：

```
# Create Logstore Shipper
from aliyun.log.logexception import LogException
project = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
shipper_config = {
    'shipperName': 'test_ship22',
    'targetType': 'odps',
    'targetConfiguration': {
      'bufferInterval': 1800,
      'enable': True,
      'fields': ['__time__', '__source__', '__topic__', 'content'],
      'odpsEndpoint': 'http://odps-ext.aliyun-inc.com/api',
      'odpsProject': 'TS_DL',
      'odpsTable': 'test_odps',
      'partitionColumn': ['__time__'],
      'partitionTimeFormat': 'yyyy_MM_dd_HH_mm'
    }
}
try:
  client.create_shipper(project, logstore, shipper_config)
except LogException as ex:
  if 'shipperName already exists' in ex.get_error_message():
    # create index if index not exists
    ret = client.update_shipper(project, logstore, shipper_config)
```

示例13：查询日志投递任务配置

示例代码如下：

```
# List and get Logstore Shipper Config
project = "YOUR_SLS_PROJECT"
logstore = "YOUR_SLS_LOGSTORE"
ret = client.list_shipper(project, logstore)
for shipper_name in ret.get_body()['shipper']:
  ret = client.get_shipper(project, logstore, shipper_name)
  print("----------------------------------------")
  print(shipper_name)
  print(ret.get_body())
```