ALIBABA CLOUD

Alibaba Cloud

日志服务

数据采集

文档版本：20220711

阿里云

法律声明

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

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

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

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

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

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

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

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

目录

1.数据采集概述 12
2.采集加速 14
2.1. 步骤一：开启全球加速服务 14
2.2. 步骤二：配置Logtail采集加速 14
2.3. 关闭全球加速服务 14
3.Logtail采集 16
3.1. 简介 16
3.1.1. Logtail采集概述 16
3.1.2. Logtail采集原理 17
3.1.3. Logtail配置文件和记录文件 19
3.2. 选择网络 24
3.3. 安装 25
3.3.1. 安装Logtail（ECS实例） 25
3.3.2. 安装Logtail（Linux系统） 28
3.3.3. 安装Logtail（Windows系统） 38
3.3.4. 设置Logtail启动参数 41
3.4. 机器组 47
3.4.1. 简介 47
3.4.2. 配置用户标识 48
3.4.3. 创建IP地址机器组 50
3.4.4. 创建用户自定义标识机器组 51
3.4.5. 管理机器组 53
3.4.6. 管理Logtail采集配置 53
3.5. 采集文本日志 54
3.5.1. 概述 54
3.5.2. 使用极简模式采集日志 54

3.5.3. 使用完整正则模式采集日志 58
3.5.4. 使用Nginx模式采集日志 62
3.5.5. 使用分隔符模式采集日志 66
3.5.6. 使用JSON模式采集日志 71
3.5.7. 使用IIS模式采集日志 74
3.5.8. 使用Apache模式采集日志 79
3.5.9. 导入历史日志文件 83
3.5.10. 时间格式 85
3.5.11. 日志主题 86
3.6. 采集容器日志 88
3.6.1. 概述 88
3.6.2. 安装Logtail组件 89
3.6.3. 通过DaemonSet-控制台方式采集容器文本日志 93
3.6.4. 通过DaemonSet-控制台方式采集容器标准输出 100
3.6.5. 通过DaemonSet-CRD方式采集容器日志 110
3.6.6. 通过Sidecar-CRD方式采集容器文本日志 122
3.6.7. 通过Sidecar-控制台方式采集容器文本日志 131
3.6.8. 采集标准Docker容器日志 135
3.6.9. 采集Kubernetes事件 137
3.7. 使用Logtail插件采集数据 140
3.7.1. 概述 140
3.7.2. 采集MySQL Binlog 141
3.7.3. 采集MySQL查询结果 148
3.7.4. 采集HTTP数据 151
3.7.5. 采集Syslog 154
3.7.6. 采集Beats和Logstash数据源 157
3.7.7. 采集Systemd Journal日志 159
3.7.8. 采集Docker事件 163

3.7.9. 采集Windows事件日志 165
3.8. 使用Logtail插件处理数据 169
3.8.1. 概述 169
3.8.2. 提取字段 171
3.8.3. 添加字段 176
3.8.4. 丢弃字段 177
3.8.5. 重命名字段 177
3.8.6. 打包字段 178
3.8.7. 展开JSON字段 179
3.8.8. 过滤日志 180
3.8.9. 提取日志时间 181
3.8.10. 转换IP地址 184
3.8.11. 追加字段 185
3.9. 使用内置的Logtail告警监控规则 186
3.10. Logtail限制说明 190
3.11. Logtail发布历史 191
3.12. Logtail常见问题 198
4.云产品日志采集 199
4.1. 云产品日志概述 199
4.2. 云产品日志通用操作 201
4.3. 函数计算执行日志 202
4.3.1. 使用前须知 202
4.3.2. 开通日志功能 202
4.3.3. 日志字段详情 203
4.4. OSS访问日志 203
4.4.1. 使用前须知 203
4.4.2. 开通日志实时查询功能 204
4.4.3. 日志字段详情 204

4.5. NAS访问日志 210
4.5.1. 使用前须知 210
4.5.2. 开通日志分析功能 211
4.5.3. 日志字段详情 211
4.6. DDoS高防（旧版）日志 212
4.6.1. 使用前须知 212
4.6.2. 开通全量日志功能 213
4.6.3. 日志字段详情 213
4.7. DDoS高防（新BGP&国际）日志 214
4.7.1. 使用前须知 214
4.7.2. 开通全量日志分析功能 215
4.7.3. 管理日志存储空间 216
4.7.4. 日志字段详情 217
4.8. DDoS原生防护日志 218
4.8.1. 使用前须知 219
4.8.2. 开通原生防护日志 219
4.8.3. 日志字段详情 220
4.9. 云安全中心日志 221
4.9.1. 使用前须知 221
4.9.2. 开通日志分析功能 221
4.9.3. 日志字段详情 222
4.10. WAF日志 231
4.10.1. 使用前须知 231
4.10.2. 开通WAF日志服务 232
4.10.3. 管理日志存储空间 232
4.10.4. 日志字段详情 232
4.11. 云防火墙日志 237
4.11.1. 使用前须知 237

4.11.2. 开通日志分析功能 238
4.11.3. 管理日志存储空间 238
4.11.4. 日志字段详情 239
4.12. 负载均衡7层访问日志 240
4.12.1. 使用前须知 240
4.12.2. 开通访问日志功能 241
4.12.3. 日志字段详情 241
4.13. 负载均衡4层秒级监控指标 242
4.13.1. 使用前须知 242
4.13.2. 开通秒级监控 243
4.13.3. 秒级监控指标详情 243
4.14. VPC流日志 244
4.14.1. 使用前须知 244
4.14.2. 开通流日志功能 246
4.14.3. 日志字段详情 247
4.15. 弹性公网IP日志 248
4.15.1. 使用前须知 248
4.15.2. 开启秒级监控 249
4.15.3. 日志字段详情 249
4.16. API网关访问日志 250
4.16.1. 使用前须知 250
4.16.2. 开通日志管理功能 251
4.16.3. 日志字段详情 251
4.17. ActionTrail访问日志 252
4.17.1. 使用前须知 252
4.17.2. 开通日志功能 252
4.17.3. 日志字段详情 253
4.18. 平台操作日志 254

4.18.1. 使用前须知 254
4.18.2. 开通平台操作日志功能 255
4.18.3. 平台操作事件结构定义 256
4.19. PolarDB-X 1.0 SQL审计日志 257
4.19.1. 使用前须知 257
4.19.2. 开启SQL审计与分析功能 258
4.19.3. 日志字段详情 259
4.20. RDS SQL审计日志 260
4.20.1. 使用前须知 260
4.20.2. 采集RDS SQL审计日志 261
4.20.3. 日志字段详情 262
4.21. Redis日志 262
4.21.1. 使用前须知 262
4.21.2. 开通日志审计功能 263
4.21.3. 日志字段详情 263
4.22. MongoDB日志 265
4.22.1. 使用前须知 265
4.22.2. 开通日志审计功能 265
4.22.3. 日志字段详情 266
4.23. IoT日志 267
4.23.1. 使用前须知 267
4.23.2. 开通日志转储功能 268
4.23.3. 日志字段详情 269
4.24. DCDN实时日志 269
4.24.1. 使用前须知 269
4.24.2. 开启实时日志投递 271
4.24.3. 日志字段详情 272
5.数据导入 274

5.1. 导入OSS数据 274
5.2. 导入MaxCompute数据 279
5.3. 时间格式 281
6.其他采集方式 283
6.1. 使用Web Tracking采集日志 283
6.2. 使用Kafka协议上传日志 284
6.3. 使用Syslog协议上传日志 289
6.4. Logstash 292
6.4.1. 安装Logstash 292
6.4.2. 创建Logstash采集配置和处理配置 293
6.4.3. 设置Logstash为Windows服务 295
6.4.4. 进阶功能 296
6.4.5. Logstash 错误处理 296
6.5. SDK采集 296
7.采集常见日志 298
7.1. 采集Log4j日志 298
7.2. 采集Python日志 301
7.3. 采集Node.js日志 304
7.4. 采集WordPress日志 307
7.5. 采集Unity3D日志 310
8.最佳实践 312
8.1. 采集-IoT/嵌入式日志 312
8.2. 采集-通过WebTracking采集日志 317
8.3. 采集-搭建移动端日志直传服务 321
8.4. 采集Zabbix数据 324
8.5. 跨阿里云账号采集日志 325
8.6. 跨阿里云账号采集容器日志 328
9.常见问题 332

9.1. 数据采集常见问题 332
9.2. 日志管理 332
9.3. Logtail基本问题 333
9.4. 如何采集企业内网服务器日志? 334
9.5. 如何排查容器日志采集异常 336
9.6. 如何获取容器的Label和环境变量 338
9.7. 查询本地采集状态 340
9.8. Logtail采集日志失败的排查思路 348
9.9. Logtail机器组无心跳排查思路 349
9.10. 如何使用Logtail自动诊断工具 352
9.11. ECS经典网络切换为VPC后，如何更新Logtail配置 356
9.12. 如何查看Logtail采集错误信息 356
9.13. 日志服务采集数据常见的错误类型 357
9.14. 如何优化正则表达式的性能 362
9.15. 如何通过完整正则模式采集多种格式日志 362
9.16. SLB访问日志采集不到 363
9.17. 日志采集Agent对比 363
9.18. 日志服务采集功能与Kafka对比 366
9.19. 如何实现文件中的日志被采集多份 367

1.数据采集概述

日志服务支持采集服务器与应用、开源软件、物联网、移动端、标准协议、阿里云产品等多种来源的数据。本文列举了日志服务所支持的数据来源。

数据来源

日志服务支持的数据来源如下：

类别	来源	接入方式	更多
应用	程序输出	Logtail	无
	访问日志	Logtail	分析Nginx访问日志
语言	Java	• Log Service Java SDK • Java Producer Library	无
	Log4J Appender	• 1.x • 2.x	无
	LogBack Appender	LogBack	无
	C	Log Service C SDK	无
	Python	Log Service Python SDK	无
	Python Logging	无	无
	PHP	Log Service PHP SDK	无
	.Net	Log Service csharp SDK	无
	C++	Log Service C++ SDK	无
	Go	• Log Service Go SDK • Golang Producer Library	无
	NodeJS	NodeJs	无
	JS	JS/Web Tracking	无
OS	Linux	Logtail	无
	Windows	Logtail	无
	Mac/Unix	Native C	无
	Docker文件	Logtail文件采集	无
	Docker输出	Logtail容器输出	无
数据库	MySQL Binlog	采集MySQL Binlog	无
	JDBC Select	采集MySQL查询结果	无
移动端	iOS、Android	• Log Service Android SDK • Log Service iOS SDK	无
	网页	JS/Web Tracking	无
	智能IoT	C Producer Library	无
标准协议	HTTP 轮询	Logtail HTTP	采集及分析Nginx监控日志
	Syslog	Logtail插件-syslog输入源	无
	MaxCompute数据	导入MaxCompute数据	无

类别 数据导入	来源	接入方式	更多
	OSS数据	导入OSS数据	无
	Flink	通过Flink写入数据	注册日志服务SLS
第三方	Logstash	Logstash、使用Kafka协议上传日志	无
	Flume	Flume消费	无
	Beats	使用Kafka协议上传日志	无
	Fluentd	使用Kafka协议上传日志	无
	Telegraf	使用Kafka协议上传日志	无
阿里云云产品	ECS、OSS等阿里云产品日志	云产品日志采集	无

选择网络和接入点

日志服务提供以下网络类型的接入点：

- 阿里云内网（经典网络和专有网络VPC）：本地域内服务访问，带宽链路质量较好。
- 公网：可以被任意访问，访问速度取决于链路质量。为了保障传输安全建议使用HTTPS。

关于接入点的更多信息，请参见服务入口。

常见问题

- 专线方式接入应如何选择网络？

  请选择阿里云内网（经典网络或专有网络VPC）。

- 采集公网数据时能否采集公网IP地址？

  您可以在Logstore属性中开通记录外网IP功能。更多信息，请参见创建Logstore。

- 将地域A上的ECS日志采集到地域B下日志服务Project中，应如何选择网络？

  在地域A上的ECS中安装地域B公网的版本Logtail，进行公网传输。其他情况下的网络选择，请参见选择网络。

- 如何快速判断目标域名能否连接？

  执行以下命令，如果有返回信息则表示可以联通。

  ```
  curl $myproject.cn-hangzhou.log.aliyuncs.com
  ```

  - `$myproject`：Project名。
  - `cn-hangzhou.log.aliyuncs.com`：访问接入点。

2.采集加速

2.1. 步骤一：开启全球加速服务

本文介绍开启日志服务全球加速服务的操作步骤。

操作步骤

1. 提交工单联系日志服务技术支持人员为Project开启全球加速服务。

后续操作

开启全球加速服务后，您可以配置采集加速。

- 通过Logtail采集日志。
  - 如果开启全球加速后再安装Logtail，则在安装Logtail时将安装模式配置为**全球加速**，即可采用全球加速模式采集日志。具体操作，请参见安装Logtail（Linux系统）。
  - 如果开启全球加速前已安装Logtail，则需手动切换Logtail采集模式为**全球加速**。具体操作，请参见步骤二：配置Logtail采集加速。
- 通过SDK采集日志。

  通过SDK采集日志时可通过替换Endpoint获得加速效果，即将配置的Endpoint替换为`log-global.aliyuncs.com`。

2.2. 步骤二：配置Logtail采集加速

本文介绍在开启日志服务的全球加速服务前已安装了Logtail，则如何将Logtail采集模式切换为全球加速，实现采集加速。

前提条件

已开启日志服务的全球加速功能。更多信息，请参见步骤一：开启全球加速服务。

配置说明

- 如果开启全球加速后再安装Logtail，则在安装Logtail时将安装模式配置为**全球加速**，即可采用全球加速模式采集日志。更多信息，请参见安装Logtail（Linux系统）。
- 如果开启全球加速前已安装Logtail，则请参见本文档手动切换Logtail采集模式为**全球加速**。

操作步骤

1. 停止Logtail。
   - Linux系统

     以root用户执行`/etc/init.d/ilogtaild stop`命令。
   - Windows系统
     a. 选择**开始 > 控制面板 > 管理工具 > 服务**。
     b. 在**服务**对话框中，找到LogtailWorker，右键单击**停止**。
2. 修改Logtail启动配置文件*ilogtail_config.json*。

   将参数data_server_list中的endpoint一行替换为`log-global.aliyuncs.com`。更多信息，请参见启动参数配置文件（ilogtail_config.json）。
3. 启动Logtail。
   - Linux系统

     以root用户执行`/etc/init.d/ilogtaild start`命令。
   - Windows系统
     a. 选择**开始 > 控制面板 > 管理工具 > 服务**。
     b. 在**服务**对话框中，找到LogtailWorker，右键单击**启动**。

2.3. 关闭全球加速服务

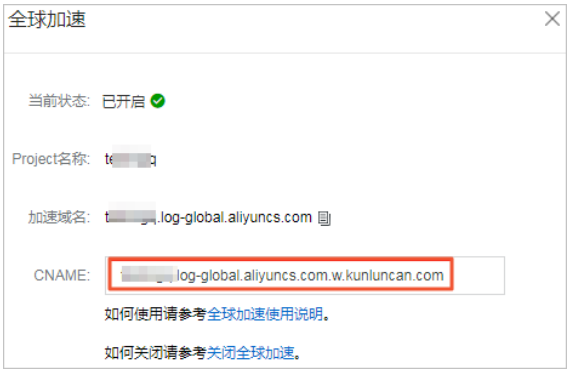
本文档介绍如何关闭日志服务全球加速功能。

操作步骤

1.
2.
3. 在**概览**页面中，单击**全球加速**后面的**设置**。

4. 在**全球加速**对话框中，填入加速域名对应的**CNAME**，并单击**关闭加速**。

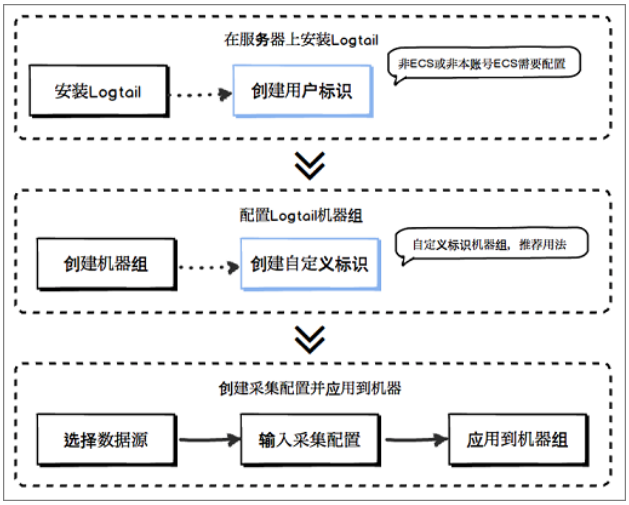
> **注意** 关闭全球加速前，请确保不再使用该域名上传或请求数据。



3.Logtail采集
3.1. 简介
3.1.1. Logtail采集概述

Logtail是日志服务提供的日志采集Agent，用于采集阿里云ECS、自建IDC、其他云厂商等服务器上的日志。本文介绍Logtail的功能、优势、使用限制及配置流程等信息。

配置流程



1. 在服务器上安装Logtail。
   - 如何为Linux系统安装Logtail，请参见安装Logtail（Linux系统）。
   - 如何为Windows系统安装Logtail，安装Logtail（Windows系统）。
2. 如果您的服务器是非本账号的ECS、本地IDC或其他云厂商服务器，您需要为服务器配置用户标识。

   具体操作，请参见配置用户标识。
3. 创建机器组。
   - 如何创建支持创建IP地址类型的机器组，请参见创建IP地址机器组
   - 如何创建用户自定义标识类型的机器组，请参见创建用户自定义标识机器组。
4. 创建Logtail采集配置，并应用到机器组。

   您可以通过日志服务控制台配置向导完成操作。具体操作，请参见概述。

完成上述操作后，Logtail开始采集您服务器上的日志，并发送到对应的Logstore中。您可以通过日志服务控制台、API、SDK或CLI查询日志。

功能优势

- 基于日志文件，无侵入式采集日志。您无需修改应用程序代码，且采集日志不会影响您的应用程序运行。
- 除采集文本日志外，还支持采集binlog、http数据、容器日志等。
- 对容器支持友好，支持标准容器、swarm集群、Kubernetes集群等容器集群的数据采集。
  - 阿里云容器服务Swarm：请参见集成日志服务。
  - 阿里云容器服务Kubernetes：请参见概述。
  - 自建Kubernetes：请参见概述。
  - 自建其他Docker集群：请参见采集标准Docker容器日志。
- 稳定处理日志采集过程中的各种异常。当遇到网络异常、服务端异常等问题时会采用主动重试、本地缓存数据等措施保障数据安全。
- 基于日志服务的集中管理能力。安装Logtail后，只需要在日志服务上配置机器组、Logtail采集配置等信息即可。
- 完善的自我保护机制。为保证运行在服务器上的Logtail，不会明显影响您服务器上其他服务的性能，Logtail在CPU、内存及网络使用方面都做了严格的限制和保护机制。

处理能力与限制

Logtail处理能力与限制，请参见Logtail限制说明。

核心概念

- 机器组：一个机器组包含一台或多台待采集同类日志的服务器。将Logtail采集配置应用到机器组上后，日志服务会根据Logtail采集配置采集机器组内所有服务器上的日志。

  日志服务通过机器组管理所有需要通过Logtail采集日志的服务器，支持通过IP地址或者用户自定义标识的方式定义机器组。您可以通过日志服务控制台管理机器组（包括创建、删除机器组，添加、移除机器等操作）。
- Logtail：日志服务提供的日志采集Agent，运行在待采集日志的服务器上。
  - Linux服务器：Logtail安装在*/usr/local/ilogtail*目录下，启动两个以ilogtail开头的独立进程，一个为采集进程，另外一个为守护进程，程序运行日志保存在*/usr/local/ilogtail/ilogtail.LOG*文件中。
  - Windows服务器：Logtail安装在*C:\Program Files\Alibaba\Logtail*目录下（32位系统）或*C:\Program Files (x86)\Alibaba\Logtail*目录下（64位系统）。您可以通过**控制面板 > 管理工具 > 服务**查看LogtailDaemon服务。程序运行日志保存在安装目录下的`ilogtail.LOG`文件中。
- Logtail采集配置：Logtail采集日志的策略集合。通过在创建Logtail采集配置时设置数据源、采集模式等参数，实现定制化的采集策略。Logtail采集配置定义了如何在服务器上采集同类日志并解析、发送到指定的日志服务Logstore上。

基本功能

功能	说明
实时采集日志	动态监控日志文件，实时读取、解析增量日志。日志从生成到发送到日志服务的延迟一般在3秒内。更多信息，请参见Logtail采集原理。 **说明** Logtail不支持采集历史日志，对于一条日志，读取该日志的时间减去日志中记录的时间，差值超过12小时会被丢弃。
自动处理日志轮转	很多应用会按照文件大小或者日期对日志文件进行轮转（rotation），把原日志文件重命名，并新建一个空日志文件等待写入。例如：*app.LOG*文件，通过日志轮转会生成 *app.LOG.1*、*app.LOG.2* 等。您可以指定采集日志写入的文件，如*app.LOG*，Logtail会自动检测到日志轮转过程，保证这个过程中不会丢失日志数据。
多种采集输入源	Logtail除支持采集文本日志外，还支持syslog、http、MySQL binlog等数据源。更多信息，请参见数据采集概述。
兼容开源采集Agent	Logtail支持Logstash、Beats等开源软件采集的数据作为数据源。更多信息，请参见数据采集概述。
自动处理采集异常	因为服务端错误、网络措施、配额超限等各种异常导致数据发送失败，Logtail会按场景主动重试。如果重试失败则将数据写入本地缓存，等待3秒自动重发。更多信息，请参见如何使用Logtail自动诊断工具。
灵活配置采集策略	可以通过Logtail采集配置非常灵活地采集日志。您可以根据实际场景指定日志目录和文件，支持精确匹配，也支持通配符模糊匹配。您也可以自定义提取日志的方式和提取字段的名称，日志服务支持通过正则表达式提取日志。 由于日志服务中的日志数据模型要求每条日志必须有精确的时间戳信息，Logtail提供了自定义的日志时间格式，方便您从不同格式的日志数据中提取必要的日志时间戳信息。
自动同步Logtail采集配置	您在日志服务控制台上新建或更新Logtail采集配置，一般情况下，Logtail在3分钟时间内即可接收并生效。更新过程中不会丢失日志数据。
自我监控状态	Logtail会实时监控自身CPU和内存消耗，避免Logtail消耗您太多资源而影响您的其他服务。Logtail在运行过程中，如果资源使用超出限制将会自动重启，避免影响服务器上的其它服务。同时，Logtail有主动的网络限流保护措施，防止过度消耗带宽。更多信息，请参见启动参数配置文件（ilogtail_config.json）。
签名数据发送	为保证您的数据在发送过程中不会被篡改，Logtail会通过可信通道从服务端获取私密Token，并对所有发送日志的数据包进行数据签名。 **说明** Logtail在获取私密Token时采用HTTPS通道，保障相关安全性。

数据采集可靠性

Logtail在采集日志时，定期将采集的点位（CheckPoint）信息保存到本地，如果遇到服务器意外关闭、进程崩溃等异常情况时，Logtail重启后会从上一次记录的位置开始采集数据，尽可能保证数据不丢失。Logtail会根据启动参数配置文件中配置进行工作，如果资源占用超过限定值5分钟以上，则Logtail会强制重启。重启后可能会产生一定的数据重复。

Logtail内部采用了很多机制提升日志采集可靠性，但并不能保证日志一定不会丢失。以下情况可能造成日志丢失：

- Logtail未运行且日志轮转多次。
- 日志轮转速度极快，例如1秒轮转1次。
- 日志采集速度长期无法达到日志产生速度。

3.1.2. Logtail采集原理

本文介绍Logtail的采集过程，包括监听文件、读取文件、处理日志、过滤日志、聚合日志和发送数据。

采集过程

Logtail采集数据的过程如下：

1. 监听文件
2. 读取文件
3. 处理日志
4. 过滤日志
5. 聚合日志
6. 发送日志

监听文件

在服务器上安装Logtail及在日志服务控制台上创建Logtail采集配置后，日志服务会实时下发Logtail采集配置到Logtail，Logtail根据Logtail采集配置开始监听文件。Logtail根据Logtail采集配置中的日志路径和最大监控目录深度，逐层扫描符合规则的日志目录和文件。

将Logtail采集配置应用到机器组后，对应服务器上没有发生修改事件的日志文件会被判定为历史日志文件，Logtail监听到历史日志文件，并不会采集。当日志文件产生了修改事件，才会触发采集流程，Logtail开始读取文件。如果您要采集历史日志文件，请参见导入历史日志文件。

为保证采集日志的时效性以及稳定性，Logtail会对待采集的目录注册事件监听（Linux下使用Inotify）以及定期轮询。

读取文件

Logtail监听到日志文件，并确认有更新后，开始读取。

- 首次读取日志文件时，日志服务默认首次读取大小为1024 KB。
  - 如果文件小于1024 KB，则从文件内容起始位置开始读取。
  - 如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始读取。

  > **说明** 日志服务支持自定义首次读取大小。
  >
  > - 控制台方式：在Logtail配置中修改**首次采集大小**参数。具体操作，请参见高级配置。
  > - API方式：在Logtail配置中修改**tail_size_kb**参数。具体操作，请参见advanced参数说明。

- 如果Logtail已读取过该日志文件，则从上次读取的Checkpoint处继续读取。
- 读取日志文件时，每次最多可以读取512 KB，因此每条日志的大小请控制在512 KB以内，否则无法正常读取。

> **说明** 如果您修改了服务器上的时间，请手动重启Logtail，否则会导致日志时间不正确、意外丢弃日志等现象。

处理日志

Logtail读取日志后，对日志内容进行分行、解析、设置时间字段。

- 分行

  如果Logtail采集配置中指定了行首正则表达式，则Logtail根据行首正则表达式对每次读取的日志进行分行，切分成多条日志；如果没有指定行首正则表达式，则将一行日志作为一条日志处理。

- 解析

  根据Logtail采集配置中配置的采集模式，对每条日志内容进行解析。

  > **说明** 如果您的正则表达式比较复杂，可能会导致CPU占用率过高，请使用合理高效的正则表达式。

  如果解析失败，会根据Logtail采集配置中是否开启丢弃解析失败日志的功能进行处理。

  - 开启**丢弃解析失败日志**，则直接丢弃该日志，并上报解析失败的报错信息。
  - 关闭**丢弃解析失败日志**，则上传解析失败的原始日志，其中Key为**raw_log**、Value为日志内容。

- 设置日志时间字段
  - 如果未配置时间字段，则日志时间为当前解析日志的时间。
  - 如果配置了时间字段：
    - 日志中记录的时间距离当前时间12小时以内，则从解析的日志字段中提取时间。
    - 日志中记录的时间距离当前时间12小时以上，则丢弃该日志并上传错误信息。

过滤日志

处理日志后，根据Logtail采集配置中的过滤器配置过滤日志。

- 在Logtail采集配置中未设置**过滤器配置**，则不过滤日志，执行下一个步骤。
- Logtail采集配置已设置**过滤器配置**，则对每条日志中的所有字段进行遍历并验证。

  只有符合过滤器配置的日志被采集。

聚合日志

为降低网络请求次数，在日志处理、过滤完毕后，会在Logtail内部缓存一段时间后进行聚合打包，再发送到日志服务。缓存数据后，触发打包日志发送到日志服务的条件如下：

- 日志聚合时间超过3秒。
- 日志聚合条数超过4096条。
- 日志聚合总大小超过512 KB。

发送日志

Logtail将采集到的日志聚合并发送到日志服务。如果数据发送失败，Logtail自动根据错误信息决定重试或放弃发送。

错误信息	说明	Logtail处理方式
401错误	Logtail没有权限采集数据。	直接丢弃日志包。
404错误	Logtail采集配置中指定的Project或Logstore不存在。	直接丢弃日志包。
403错误	Shard Quota超出限制。	等待3秒后重试。
500错误	服务端异常。	等待3秒后重试。

> **说明** 如果要调整数据的发送速度和最大并发数，您可以设置启动参数配置文件中的max_bytes_per_sec参数和send_request_concurrency参数。具体操作，请参见设置Logtail启动参数。

3.1.3. Logtail配置文件和记录文件

Logtail运行时依赖一系列的配置文件并产生部分信息记录文件，本文档介绍常见文件的基本信息及路径。

启动参数配置文件（ilogtail_config.json）

*ilogtail_config.json*文件用于配置Logtail的启动参数，文件类型为JSON，详情请参见设置Logtail启动参数。

> **说明**
> - 该文件必须为合法JSON，否则无法启动Logtail。
> - 修改该文件后需重启Logtail才能生效。

安装Logtail后，您可以在*ilogtail_config.json*文件进行如下操作。

- 修改Logtail的运行参数。
- 检验安装命令是否正确。

  *ilogtail_config.json*文件中的 `config_server_address` 参数和 `data_server_list` 参数的值取决于安装时选择的安装命令，如果其中的地域和日志服务所在地域不一致或地址无法联通，说明安装时选择了错误的命令。这时Logtail无法正常采集日志，需重新安装。
- 文件路径
  - Linux：*/usr/local/ilogtail/ilogtail_config.json*。
  - Windows：
    - 64位：*C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json*。
    - 32位：*C:\Program Files\Alibaba\Logtail\ilogtail_config.json*。

    > **说明** Windows 64位操作系统支持运行32/64位应用程序，但是出于兼容性考虑，在Windows 64位操作系统上，Windows会使用单独的x86目录来存放32位应用程序。
    >
    > Windows版本的Logtail是32位程序，所以在64位操作系统上的安装目录为Program Files (x86)。如果日志服务推出64位的Windows版本Logtail，会自动安装到Program Files目录下。
  - 容器：该文件存储在Logtail容器中，文件路径配置在Logtail容器的环境变量*ALIYUN_LOGTAIL_CONFIG*中，您可通过**docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_CONFIG**命令查看。例如：*/etc/ilogtail/conf/cn-hangzhou/ilogtail_config.json*。

- 文件示例

```
$cat /usr/local/ilogtail/ilogtail_config.json
{
    "config_server_address" : "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
    "data_server_list" :
    [
        {
            "cluster" : "cn-hangzhou",
            "endpoint" : "cn-hangzhou-intranet.log.aliyuncs.com"
        }
    ],
    "cpu_usage_limit" : 0.4,
    "mem_usage_limit" : 100,
    "max_bytes_per_sec" : 2097152,
    "process_thread_count" : 1,
    "send_request_concurrency" : 4,
    "streamlog_open" : false
}
```

用户标识配置文件

用户标识配置文件中包含阿里云主账号的ID信息，用于标识这台服务器有权限被该账号访问、采集日志，详情请参见配置用户标识。

> **说明**
> - 在采集非本账号ECS、自建IDC、其他云厂商服务器日志时需要配置用户标识。
> - 用户标识配置文件中必须配置阿里云主账号ID，不支持子账号。
> - 用户标识配置文件只需配置文件名，无需配置文件后缀。
> - 一台服务器上可配置多个用户标识，Logtail容器中仅支持配置一个用户标识。

- 文件路径
  - Linux：*/etc/ilogtail/users/*。
  - Windows：*C:\LogtailData\users*。
  - 容器：用户标识保存在Logtail容器的环境变量ALIYUN_LOGTAIL_USER_ID中，您可通过**docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_ID**命令查看。
- 文件示例

```
$ls /etc/ilogtail/users/
*************** ****************
```

用户自定义标识文件（user_defined_id）

*user_defined_id*文件用于配置用户自定义标识，详情请参见创建用户自定义标识机器组。

> **说明** 创建自定义标识机器组时需要配置*user_defined_id*文件。

- 文件路径
  - Linux：*/etc/ilogtail/user_defined_id*。
  - Windows：*C:\LogtailData\user_defined_id*。
  - 容器：用户自定义标识配置在Logtail容器的环境变量ALIYUN_LOGTAIL_USER_DEFINED_ID中，可通过**docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_DEFINED_ID**命令查看。
- 文件示例

```
$cat /etc/ilogtail/user_defined_id
aliyun-ecs-rs1e16355
```

Logtail采集配置文件（user_log_config.json）

*user_log_config.json*文件记录Logtail从日志服务获取的Logtail采集配置信息，文件类型为JSON，每次Logtail采集配置更新时会同步更新该文件。可通过*user_log_config.json*文件确认Logtail采集配置是否已经下发到服务器。Logtail采集配置文件存在，且内容与日志服务上的Logtail采集配置一致，表示Logtail采集配置已下发。

> **说明** 除手动配置AK信息、数据库密码等敏感信息外，不建议修改该文件。

- 文件路径

- Linux：*/usr/local/ilogtail/user_log_config.json*。
- Windows
  - 64位：*C:\Program Files (x86)\Alibaba\Logtail\user_log_config.json*。
  - 32位：*C:\Program Files\Alibaba\Logtail\user_log_config.json*。
- 容器：*/usr/local/ilogtail/user_log_config.json*。

- 文件示例

```
$cat /usr/local/ilogtail/user_log_config.json
{
   "metrics" : {
      "##1.0##k8s-log-c12ba2028*****939f0b$app-java" : {
         "aliuid" : "16542189*****50",
         "category" : "app-java",
         "create_time" : 1534739165,
         "defaultEndpoint" : "cn-hangzhou-intranet.log.aliyuncs.com",
         "delay_alarm_bytes" : 0,
         "enable" : true,
         "enable_tag" : true,
         "filter_keys" : [],
         "filter_regs" : [],
         "group_topic" : "",
         "local_storage" : true,
         "log_type" : "plugin",
         "log_tz" : "",
         "max_send_rate" : -1,
         "merge_type" : "topic",
         "plugin" : {
            "inputs" : [
               {
                  "detail" : {
                     "IncludeEnv" : {
                        "aliyun_logs_app-java" : "stdout"
                     },
                     "IncludeLable" : {
                        "io.kubernetes.container.name" : "java-log-demo-2",
                        "io.kubernetes.pod.namespace" : "default"
                     },
                     "Stderr" : true,
                     "Stdout" : true
                  },
                  "type" : "service_docker_stdout"
               }
            ]
         },
         "priority" : 0,
         "project_name" : "k8s-log-c12ba2028c*****ac1286939f0b",
         "raw_log" : false,
         "region" : "cn-hangzhou",
         "send_rate_expire" : 0,
         "sensitive_keys" : [],
         "tz_adjust" : false,
         "version" : 1
      }
   }
}
```

AppInfo记录文件（app_info.json）

*app_info.json*文件记录Logtail的启动时间、获取到的IP地址、主机名等信息。

如果已在服务器的*/etc/hosts*文件中设置了主机名与IP地址绑定，则自动获取绑定的IP地址。如果没有设置主机名绑定，会自动获取本机的第一块网卡的IP地址。

> **说明**
> - AppInfo记录文件仅用于记录Logtail内部信息。
> - 如果修改了服务器的主机名等网络配置，请重启Logtail，获取新的IP地址。

- 文件路径

- Linux：*/usr/local/ilogtail/app_info.json*。
- Windows
  - 64位：*C:\Program Files (x86)\Alibaba\Logtail\app_info.json*。
  - 32位：*C:\Program Files\Alibaba\Logtail\app_info.json*。
- 容器：*/usr/local/ilogtail/app_info.json*。

- 文件示例

```
$cat /usr/local/ilogtail/app_info.json
{
   "UUID" : "",
   "hostname" : "logtail-ds-slpn8",
   "instance_id" : "E5F93BC6-B024-11E8-8831-0A58AC14039E_1**.***.***.***_1536053315",
   "ip" : "1**.***.***.***",
   "logtail_version" : "0.16.13",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2018-09-04 09:28:36"
}
```

字段	说明
UUID	服务器序列号。
hostname	主机名。
instance_id	随机生成的Logtail唯一标识。
ip	Logtail获取到的IP地址。该字段为空时表示Logtail没有获取到IP地址，Logtail无法正常运行，请为服务器设置IP地址并重启Logtail。 **说明** 如果您创建了IP地址机器组，请确保机器组中配置的IP与此处显示的IP地址一致。如果不一致，请在日志服务控制台上修改机器组内IP地址，等待1分钟再查看。
logtail_version	Logtail客户端版本。
os	操作系统版本。
update_time	Logtail最近一次启动时间。

Logtail运行日志（ilogtail.LOG）

*ilogtail.LOG*文件记录了Logtail的运行日志，日志级别从低到高分别为INFO、WARN和ERROR，其中INFO类型的日志无需关注。

如果采集异常，请先诊断采集错误，根据具体的错误类型和Logtail运行日志排查问题，详情请参见如何查看Logtail采集错误信息。

说明 如果因Logtail采集异常提交工单时，请同时上传该日志。

- 文件路径
  - Linux：*/usr/local/ilogtail/ilogtail.LOG*。
  - Windows
    - 64位：*C:\Program Files (x86)\Alibaba\Logtail\ilogtail.LOG*。
    - 32位：*C:\Program Files\Alibaba\Logtail\ilogtail.LOG*。
  - 容器：*/usr/local/ilogtail/ilogtail.LOG*。
- 文件示例

```
$tail /usr/local/ilogtail/ilogtail.LOG
[2018-09-13 01:13:59.024679]    [INFO]    [3155]    [build/release64/sls/ilogtail/elogtail.cpp:123]    change working di
r:/usr/local/ilogtail/
[2018-09-13 01:13:59.025443]    [INFO]    [3155]    [build/release64/sls/ilogtail/AppConfig.cpp:175]    load logtail con
fig file, path:/etc/ilogtail/conf/ap-southeast-2/ilogtail_config.json
[2018-09-13 01:13:59.025460]    [INFO]    [3155]    [build/release64/sls/ilogtail/AppConfig.cpp:176]    load logtail con
fig file, detail:{
   "config_server_address" : "http://logtail.ap-southeast-2-intranet.log.aliyuncs.com",
   "data_server_list" : [
      {
         "cluster" : "ap-southeast-2",
         "endpoint" : "ap-southeast-2-intranet.log.aliyuncs.com"
      }
]
```

Logtail插件日志（logtail_plugin.LOG）

*logtail_plugin.LOG*文件记录Logtail插件的运行日志，日志级别从低到高分别为INFO、WARN和ERROR，其中INFO类型的日志无需关注。

如果在诊断采集错误时，提示**CANAL_RUNTIME_ALARM**等错误，可以通过*logtail_plugin.LOG*文件排查。

> **说明** 如果因插件异常提交工单时，请在工单中上传该文件。

- 文件路径
  - Linux：*/usr/local/ilogtail/logtail_plugin.LOG*。
  - Windows：
    - 64位：*C:\Program Files (x86)\Alibaba\Logtail\logtail_plugin.LOG*。
    - 32位：*C:\Program Files\Alibaba\Logtail\logtail_plugin.LOG*。
  - 容器：*/usr/local/ilogtail/logtail_plugin.LOG*。
- 文件示例

```
$tail /usr/local/ilogtail/logtail_plugin.LOG
2018-09-13 02:55:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 02:55:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:00:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:00:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##sls-zc-test-hz-pub$docker-stdout-config,k8s-stdout
]    open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31
bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379573    status:79
4354-64769-40379963
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-s
tdout-config,k8s-stdout]    open file for read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbf
a8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offse
t:40379573    status:794354-64769-40379963
2018-09-13 03:04:26 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##sls-zc-test-hz-pub$docker-stdout-config,k8s-stdou
t]    close file, reason:no read timeout    file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a7
0c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40
379963    status:794354-64769-40379963
2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$docker-
stdout-config,k8s-stdout]    close file, reason:no read timeout    file:/logtail_host/var/lib/docker/containers/7f46afec
6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d62
4-json.log    offset:40379963    status:794354-64769-40379963
2018-09-13 03:05:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:05:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
```

容器路径映射文件（docker_path_config.json）

*docker_path_config.json*文件只有在采集容器日志时才会创建，用于记录容器文件和宿主机文件的路径映射关系。文件类型为JSON。

如果在诊断采集错误时，如果提示**DOCKER_FILE_MAPPING_ALARM**错误，表示添加Docker文件映射失败，可以通过*docker_path_config.json*文件排查。

> **说明**
> - *docker_path_config.json*文件为记录文件，任何修改操作均不会生效。删除后会自动创建，不影响业务的正常运行。
> - 因采集容器日志异常而提交工单时，请在工单中上传此文件。

- 文件路径

/usr/local/ilogtail/docker_path_config.json

- 文件示例

```
$cat /usr/local/ilogtail/docker_path_config.json
{
   "detail" : [
      {
         "config_name" : "##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$nginx",
         "container_id" : "df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10",
         "params" : "{\n   \"ID\" : \"df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e754d8483fd330d10\",\n   \"Path\" :
\"/logtail_host/var/lib/docker/overlay2/947db346695a1f65e63e582ecfd10ae1f57019a1b99260b6c83d00fcd1892874/diff/var/log\",
\n   \"Tags\" : [\n      \"nginx-type\",\n      \"access-log\",\n      \"_image_name_\",\n      \"registry.cn-hangzhou.a
liyuncs.com/log-service/docker-log-test:latest\",\n      \"_container_name_\",\n      \"nginx-log-demo\",\n      \"_pod_
name_\",\n      \"nginx-log-demo-h2lzc\",\n      \"_namespace_\",\n      \"default\",\n      \"_pod_uid_\",\n      \"87e
56ac3-b65b-11e8-b172-00163f008685\",\n      \"_container_ip_\",\n      \"172.20.4.224\",\n      \"purpose\",\n      \"te
st\"\n   ]\n}\n"
      }
   ],
   "version" : "0.1.0"
}
```

3.2. 选择网络

本文介绍使用Logtail采集日志时，如何选择网络。

网络类型

- 阿里云内网：阿里云内网为千兆共享网络，日志数据通过阿里云内网传输比公网传输更快速、稳定，内网包括VPC和经典网络。
- 公网：使用公网传输日志数据，不仅会受到网络带宽的限制，还可能会因网络抖动、延迟、丢包等影响数据采集的速度和稳定性。
- 全球加速：利用阿里云CDN边缘节点进行日志采集加速，相对公网采集在网络延迟、稳定性上具有很大优势。

如何选择网络

- 内网

  仅有以下两种情况可以使用阿里云内网传输数据。

  - ECS和日志服务Project属于同一账号同一地域。
  - ECS和日志服务Project属于不同账号但同一地域的。

  推荐在ECS所在地域创建日志服务Project，日志服务通过阿里云内网采集ECS日志，不消耗公网带宽。

- 公网

  遇到以下两种情况时，您可以选择公网传输数据。

  - ECS和日志服务Project属于不同地域。
  - 服务器为其他云厂商服务器或自建IDC。

- 全球加速

  如果您的服务器分布在海外各地的自建机房或者来自海外云厂商，使用公网传输数据可能会出现网络延迟高、传输不稳定等问题，您可以选择全球加速传输数据。更多信息，请参见全球加速。

服务器类型	是否与Project在同一地域	是否需要手动配置用户标识①	网络类型
同一账号下的ECS	同一地域	不需要	阿里云内网
	不同地域	不需要	公网或全球加速
其他账号下的ECS	同一地域	需要	阿里云内网
	不同地域	需要	公网或全球加速
其他云厂商服务器、自建IDC	不涉及	需要	公网或全球加速

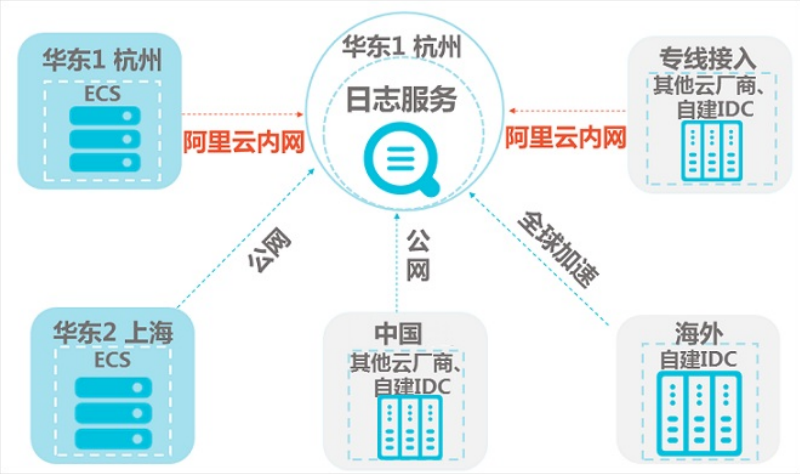
①如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您要通过Logtail采集该服务器日志，需要在服务器上安装Logtail后，手动配置日志服务所在阿里云账号ID为用户标识，表示该账号有权限通过Logtail采集该服务器日志。更多信息，请参见配置用户标识。

示例

以下是各种常见场景的网络选择示例，请根据您的实际场景选择网络类型。

> **说明** 全球加速场景中，在中国（香港）地域创建了日志服务Project，服务器为全球各地的自建IDC。建议您在类似场景下安装Logtail时选择中国（香港）的全球加速网络类型。通过全球加速传输日志，比公网传输的网络稳定性更高、性能更好。

场景类型	日志服务Project地域	服务器类型	ECS地域	安装Logtail时选择的地域	网络类型	是否需要手动配置用户标识
相同地域场景	华东1（杭州）	本账号ECS	华东1（杭州）	华东1（杭州）	内网	不需要
不同地域场景	华东2（上海）	本账号ECS	华北1（北京）	华东2（上海）	公网	不需要
不同账号场景	华东2（上海）	其他账号ECS	华北1（北京）	华东2（上海）	公网	需要
本地机房场景	华东5（深圳）	自建IDC	不涉及	华东5（深圳）	公网	需要
全球加速场景	中国（香港）	自建IDC	不涉及	中国（香港）	全球加速	需要



3.3. 安装

3.3.1. 安装Logtail（ECS实例）

日志服务支持在阿里云ECS实例中自动安装Logtail，本文介绍如何在数据采集配置向导中选择ECS实例并完成Logtail的安装。

前提条件

使用RAM用户操作时，该RAM用户需同时具备如下权限。

- AliyunOOSFullAccess权限：为RAM用户授予AliyunOOSFullAccess权限的具体操作，请参见为RAM用户授权。
- 自定义权限：为RAM用户授予如下自定义权限时，需要先创建自定义策略并为RAM用户授权。具体操作，请参见创建自定义权限策略、为RAM用户授权。

```
{
   "Version": "1",
   "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "ecs:DescribeTagKeys",
               "ecs:DescribeTags",
               "ecs:DescribeInstances",
               "ecs:DescribeInvocationResults",
               "ecs:RunCommand",
               "ecs:DescribeInvocations"
           ],
           "Resource": "*"
       },
       {
           "Effect": "Allow",
           "Action": [
               "oos:ListTemplates",
               "oos:StartExecution",
               "oos:ListExecutions",
               "oos:GetExecutionTemplate",
               "oos:ListExecutionLogs",
               "oos:ListTaskExecutions"
           ],
           "Resource": "*"
       }
   ]
}
```

背景信息

日志服务支持使用Logtail采集阿里云ECS实例、自建服务器、其他云厂商等服务器上的日志。在进行日志采集前，需要在服务器上安装Logtail。如果您是在自建服务器、其他云厂商服务器安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**分隔符-文本日志**。

   此处以分隔符-文本日志为例，您可以根据需求选择对应的数据源。
3. 选择目标Project和Logstore，单击**下一步**。
4. 在**创建机器组**步骤中，单击**ECS机器**。
5. 选择ECS实例。

   您可以通过如下方式选择ECS实例。

选择方式	说明

选择方式	说明
手动选择实例	在实例列表中，选中对应的ECS实例安装Logtail，支持多选。 在此方式中，还支持通过关键字、地域、标签、运行状态、付费方式和网络类型筛选实例。
指定实例标签	您可以给ECS实例绑定标签，用于资源管理和归类。更多信息，请参见创建或绑定标签。绑定标签后，您可以在此处选择对应的标签，系统自动在标签对应的所有ECS实例中安装Logtail。 标签由一对键值对（Key-Value）组成，如果您在此处只选择标签键，未选择标签值，则表示在该标签键绑定的所有ECS实例中安装Logtail。
指定实例资源组	您可以给ECS实例配置资源组，用于资源分级管理。更多信息，请参见资源组。绑定资源组后，您可以在此处选择对应的资源组，系统自动在该资源组下所有的ECS实例中安装Logtail。
上传CSV文件	您可以在ECS控制台中导出资源列表。导出后，您可以在此处上传已导出的CSV文件，系统自动在该列表涉及的所有ECS实例中安装Logtail。
选择全部	选择该选项后，系统自动在当前阿里云账号下的所有ECS实例中安装Logtail。
指定配置清单条件	您在运维编排服务中创建配置清单后，可以在此处通过配置清单指定待安装Logtail的ECS实例。更多信息，请参见通过OOS收集ECS实例的清单信息。

6. （可选）配置高级选项。

参数	说明
描述	执行任务的描述信息。
资源组的标签	选择对应的资源组后，系统会给执行记录添加资源组信息。
执行模式	包括两种模式，如下所示： ○ 自动执行：当安装失败时，继续执行下一台机器。 ○ 失败暂停：当安装失败时，等待安装任务重试。
速率控制类型	包括两种模式，如下所示： ○ 并发控制：并发执行安装任务。 ○ 批次控制：按批次执行安装任务。
并发速率	指定并发速率，可以是数值或者百分比。 例如配置为2目标，则表示在2台ECS上并发安装Logtail。 **说明** 当**速率控制类型**配置为**并发控制**时，需配置。
批次速率数组	把一个执行任务分成多个批次，一个批次运行完，再运行下一个批次。在上一个批次未全部完成的情况下，下一个批次不会开始，批次值可以是数值或者百分比。例如[1, 5%, 10%]表示第一个批次是1台ECS，第二个批次是总量的5%，第三个以后批次为10%。 **说明** 当**速率控制类型**配置为**批次控制**时，需配置。
最大错误次数	指定任务在停止前的最大错误次数，可以是数值或者百分比，默认值是0。 例如一共有4台ECS需安装Logtail，并发速率为1目标，最大错误次数为0，则表示在4台ECS上并发安装Logtail，当其中1台机器安装失败时，另外3台安装任务取消。

7. 单击**立即执行**。

自动跳转到如下页面，您可以查看执行结果，包括执行的基本信息、实例列表和日志等。

后续步骤

配置机器组和Logtail采集配置。更多信息，请参见采集文本日志。

3.3.2. 安装Logtail（Linux系统）

本文介绍如何在Linux服务器上安装、升级及卸载Logtail等操作。

前提条件

- 已拥有一台及以上的服务器。
- 已根据服务器类型和所在地域，确定采集日志时所需的网络类型。更多信息，请参见选择网络。

支持的系统

- 支持如下版本的Linux x86-64（64位）服务器：
  - Aliyun Linux 2
  - RedHat Enterprise 6、7、8
  - CentOS Linux 6、7、8
  - Debian GNU/Linux 8、9、10、11
  - Ubuntu 14.04、16.04、18.04、20.04
  - SUSE Linux Enterprise Server 11、12、15
  - OpenSUSE 15.1、15.2、42.3
  - 其他基于glibc 2.5及以上版本的Linux操作系统
- 支持如下版本的Linux ARM（64位）服务器：
  - Alibaba Linux 3.2 ARM版
  - Anolis OS 8.2 ARM版及以上版本
  - CentOS 8.4 ARM版
  - Ubuntu 20.04 ARM版
  - Debian 11.2 ARM版

注意事项

- 本文中的安装命令适用于Logtail 0.0版本（Logtail 0.16.x）。当您需要安装、升级Logtail 1.0版本时，需要在安装命令中添加版本号，例如 `sudo ./logtail.sh upgrade -v v1` 、 `./logtail.sh install cn-hangzhou -v v1` 。
- Logtail采用覆盖安装模式，如果您已安装过Logtail，那么重新安装Logtail时会先执行卸载、删除*/usr/local/ilogtail*目录操作。安装后默认启动Logtail并注册开机启动。
- 如果安装失败，请提工单。
- 安装Logtail后，如果ECS的网络由经典网络切换至VPC，则需要更新logtail配置。更多信息，请参见ECS经典网络切换为VPC后，如何更新Logtail配置。

安装方式

请根据您的网络类型选择对应的安装命令。

- 阿里云内网（经典网络、VPC）
- 公网
- 全球加速
- 离线安装

执行安装命令之前，您需要根据Project所在地域替换安装命令中的*${your_region_name}*参数，各地域对应的*${your_region_name}*参数如下所示。

Logtail安装参数

地域	${your_region_name}
华东1（杭州）	cn-hangzhou
华东2（上海）	cn-shanghai
华北1（青岛）	cn-qingdao
华北2（北京）	cn-beijing
华北3（张家口）	cn-zhangjiakou
华北5（呼和浩特）	cn-huhehaote
华北6（乌兰察布）	cn-wulanchabu
华南1（深圳）	cn-shenzhen
华南2（河源）	cn-heyuan
华南3（广州）	cn-guangzhou
西南1（成都）	cn-chengdu
中国（香港）	cn-hongkong
俄罗斯（莫斯科）	rus-west-1
美国（硅谷）	us-west-1
美国（弗吉尼亚）	us-east-1
新加坡	ap-southeast-1
澳大利亚（悉尼）	ap-southeast-2
马来西亚（吉隆坡）	ap-southeast-3
印度尼西亚（雅加达）	ap-southeast-5
菲律宾（马尼拉）	ap-southeast-6
泰国（曼谷）	ap-southeast-7
印度（孟买）	ap-south-1
日本（东京）	ap-northeast-1
韩国（首尔）	ap-northeast-2
德国（法兰克福）	eu-central-1
阿联酋（迪拜）	me-east-1
英国（伦敦）	eu-west-1

阿里云内网（经典网络、VPC）

- 如果您无法确定ECS所在地域，可使用Logtail安装脚本中的auto参数进行安装。

  在安装命令中指定auto参数后，Logtail安装脚本会通过ECS获取您的实例元数据，自动确定ECS所在地域，实例元数据介绍请参见 ECS实例元数据概述。

  i. 通过公网下载Logtail安装脚本。

  此下载消耗公网流量，约10 KB。

  ```
  wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh;chmod 755 logtail.sh
  ```

  ii. 使用auto参数安装logtail。

  此步骤自动下载对应地域的安装程序，不消耗公网流量。

```
./logtail.sh install auto
```

- 如果您已确定ECS所在地域，请根据地域选择安装命令。

  通过内网下载Logtail安装脚本，手动安装Logtail，不消耗公网流量。

  i. 根据日志服务Project所在地域，获取对应的*${your_region_name}*。

  各个地域对应的*${your_region_name}*请参见Logtail安装参数，例如华东 1（杭州）对应的*${your_region_name}*为cn-hangzhou。

  ii. 替换*${your_region_name}*后，执行安装命令。

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ${your_region_name}
```

您也可以根据日志服务Project所在的地域执行对应的命令进行安装。

Project所在地域	安装命令
华东1（杭州）	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou`
华东2（上海）	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai`
华北1（青岛）	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao`
华北2（北京）	`wget http://logtail-release-cn-beijing.oss-cn-beijing-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing`
华北3（张家口）	`wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou`
华北5（呼和浩特）	`wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote`
华北6（乌兰察布）	`wget http://logtail-release-cn-wulanchabu.oss-cn-wulanchabu-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-wulanchabu`
华南1（深圳）	`wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen`
华南2（河源）	`wget http://logtail-release-cn-heyuan.oss-cn-heyuan-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-heyuan`

Project所在地域	安装命令
华南3（广州）	`wget http://logtail-release-cn-guangzhou.oss-cn-guangzhou-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-guangzhou`
西南1（成都）	`wget http://logtail-release-cn-chengdu.oss-cn-chengdu-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu`
中国（香港）	`wget http://logtail-release-cn-hongkong.oss-cn-hongkong-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hongkong`
美国（硅谷）	`wget http://logtail-release-us-west-1.oss-us-west-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-west-1`
美国（弗吉尼亚）	`wget http://logtail-release-us-east-1.oss-us-east-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-east-1`
新加坡	`wget http://logtail-release-ap-southeast-1.oss-ap-southeast-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-1`
澳大利亚（悉尼）	`wget http://logtail-release-ap-southeast-2.oss-ap-southeast-2-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-2`
马来西亚（吉隆坡）	`wget http://logtail-release-ap-southeast-3.oss-ap-southeast-3-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-3`
印度尼西亚（雅加达）	`wget http://logtail-release-ap-southeast-5.oss-ap-southeast-5-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-5`
菲律宾（马尼拉）	`wget http://logtail-release-ap-southeast-6.oss-ap-southeast-6-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-6`
泰国（曼谷）	`wget http://logtail-release-ap-southeast-7.oss-ap-southeast-7-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-7`
日本（东京）	`wget http://logtail-release-ap-northeast-1.oss-ap-northeast-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-1`

Project所在地域	安装命令
韩国（首尔）	`wget http://logtail-release-ap-northeast-2.oss-ap-northeast-2-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-2`
印度（孟买）	`wget http://logtail-release-ap-south-1.oss-ap-south-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-south-1`
德国（法兰克福）	`wget http://logtail-release-eu-central-1.oss-eu-central-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-central-1`
阿联酋（迪拜）	`wget http://logtail-release-me-east-1.oss-me-east-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install me-east-1`
英国（伦敦）	`wget http://logtail-release-eu-west-1.oss-eu-west-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-west-1`
俄罗斯（莫斯科）	`wget http://logtail-release-rus-west-1.oss-rus-west-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install rus-west-1`

公网

1. 根据日志服务Project所在地域，获取对应的 `${your_region_name}` 。

   各个地域对应的*${your_region_name}*请参见Logtail安装参数，例如华东 1（杭州）对应的*${your_region_name}*为cn-hangzhou。

2. 替换*${your_region_name}*后，执行安装命令。

   ```
   wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ${your_region_name}-internet
   ```

您也可以根据日志服务Project所在地域直接执行对应的命令进行安装。

Project所在的地域	安装命令
华东1（杭州）	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou-internet`
华东2（上海）	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai-internet`
华北1（青岛）	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao-internet`
华北2（北京）	`wget http://logtail-release-cn-beijing.oss-cn-beijing.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing-internet`

Project所在的地域	安装命令
华北3（张家口）	wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou-internet
华北5（呼和浩特）	wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote-internet
华北6（乌兰察布）	wget http://logtail-release-cn-wulanchabu.oss-cn-wulanchabu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-wulanchabu-internet
华南1（深圳）	wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen-internet
华南2（河源）	wget http://logtail-release-cn-heyuan.oss-cn-heyuan.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-heyuan-internet
华南3（广州）	wget http://logtail-release-cn-guangzhou.oss-cn-guangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-guangzhou-internet
西南1（成都）	wget http://logtail-release-cn-chengdu.oss-cn-chengdu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu-internet
中国（香港）	wget http://logtail-release-cn-hongkong.oss-cn-hongkong.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hongkong-internet
美国（硅谷）	wget http://logtail-release-us-west-1.oss-us-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-west-1-internet
美国（弗吉尼亚）	wget http://logtail-release-us-east-1.oss-us-east-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-east-1-internet
新加坡	wget http://logtail-release-ap-southeast-1.oss-ap-southeast-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-1-internet
澳大利亚（悉尼）	wget http://logtail-release-ap-southeast-2.oss-ap-southeast-2.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-2-internet

Project所在的地域	安装命令
马来西亚（吉隆坡）	`wget http://logtail-release-ap-southeast-3.oss-ap-southeast-3.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-3-internet`
印度尼西亚（雅加达）	`wget http://logtail-release-ap-southeast-5.oss-ap-southeast-5.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-5-internet`
菲律宾（马尼拉）	`wget http://logtail-release-ap-southeast-6.oss-ap-southeast-6.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-6-internet`
泰国（曼谷）	`wget http://logtail-release-ap-southeast-7.oss-ap-southeast-7.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-7-internet`
日本（东京）	`wget http://logtail-release-ap-northeast-1.oss-ap-northeast-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-1-internet`
韩国（首尔）	`wget http://logtail-release-ap-northeast-2.oss-ap-northeast-2.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-2-internet`
德国（法兰克福）	`wget http://logtail-release-eu-central-1.oss-eu-central-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-central-1-internet`
阿联酋（迪拜）	`wget http://logtail-release-me-east-1.oss-me-east-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install me-east-1-internet`
印度（孟买）	`wget http://logtail-release-ap-south-1.oss-ap-south-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-south-1-internet`
英国（伦敦）	`wget http://logtail-release-eu-west-1.oss-eu-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-west-1-internet`
俄罗斯（莫斯科）	`wget http://logtail-release-rus-west-1.oss-rus-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install rus-west-1-internet`

全球加速

1. 根据日志服务Project所在地域选择安装参数。
   各个地域对应的*${your_region_name}*请参见Logtail安装参数。例如华东 1（杭州）对应的*${your_region_name}*为cn-hangzhou。
2. 替换*${your_region_name}*后，执行安装命令。

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail.sh -O logtail.sh;
chmod 755 logtail.sh; ./logtail.sh install ${your_region_name}-acceleration
```

您也可以根据日志服务Project所在地域执行对应的命令进行安装。

Project所在的地域	安装命令
华北2（北京）	`wget http://logtail-release-cn-beijing.oss-cn-beijing.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing-acceleration`
华北1（青岛）	`wget http://logtail-release-cn-qingdao.oss-cn-qingdao.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao-acceleration`
华东1（杭州）	`wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hangzhou-acceleration`
华东2（上海）	`wget http://logtail-release-cn-shanghai.oss-cn-shanghai.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai-acceleration`
华南1（深圳）	`wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen-acceleration`
华南2（河源）	`wget http://logtail-release-cn-heyuan.oss-cn-heyuan.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-heyuan-acceleration`
华南3（广州）	`wget http://logtail-release-cn-guangzhou.oss-cn-guangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-guangzhou-acceleration`
华北3（张家口）	`wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou-acceleration`
华北5（呼和浩特）	`wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote-acceleration`
华北6（乌兰察布）	`wget http://logtail-release-cn-wulanchabu.oss-cn-wulanchabu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-wulanchabu-acceleration`
西南1（成都）	`wget http://logtail-release-cn-chengdu.oss-cn-chengdu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu-acceleration`
中国（香港）	`wget http://logtail-release-cn-hongkong.oss-cn-hongkong.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hongkong-acceleration`

Project所在的地域	安装命令
美国（硅谷）	`wget http://logtail-release-us-west-1.oss-us-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-west-1-acceleration`
美国（弗吉尼亚）	`wget http://logtail-release-us-east-1.oss-us-east-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-east-1-acceleration`
新加坡	`wget http://logtail-release-ap-southeast-1.oss-ap-southeast-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-1-acceleration`
澳大利亚（悉尼）	`wget http://logtail-release-ap-southeast-2.oss-ap-southeast-2.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-2-acceleration`
马来西亚（吉隆坡）	`wget http://logtail-release-ap-southeast-3.oss-ap-southeast-3.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-3-acceleration`
印度尼西亚（雅加达）	`wget http://logtail-release-ap-southeast-5.oss-ap-southeast-5.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-5-acceleration`
菲律宾（马尼拉）	`wget http://logtail-release-ap-southeast-6.oss-ap-southeast-6.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-6-acceleration`
泰国（曼谷）	`wget http://logtail-release-ap-southeast-7.oss-ap-southeast-7.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-7-acceleration`
日本（东京）	`wget http://logtail-release-ap-northeast-1.oss-ap-northeast-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-1-acceleration`
韩国（首尔）	`wget http://logtail-release-ap-northeast-2.oss-ap-northeast-2.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-northeast-2-acceleration`
德国（法兰克福）	`wget http://logtail-release-eu-central-1.oss-eu-central-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-central-1-acceleration`
阿联酋（迪拜）	`wget http://logtail-release-me-east-1.oss-me-east-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install me-east-1-acceleration`

Project所在的地域	安装命令
印度（孟买）	`wget http://logtail-release-ap-south-1.oss-ap-south-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-south-1-acceleration`
英国（伦敦）	`wget http://logtail-release-eu-west-1.oss-eu-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install eu-west-1-acceleration`
俄罗斯（莫斯科）	`wget http://logtail-release-rus-west-1.oss-rus-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install rus-west-1-acceleration`

离线安装

1. 登录能通过公网访问的服务器。
2. 替换*${your_region_name}*后，执行下载命令，下载安装脚本和安装包。

   各个地域对应的*${your_region_name}*请参见Logtail安装参数。例如华东 1（杭州）对应的*${your_region_name}*为cn-hangzhou。

   - 下载安装脚本

     ```
     wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail.sh
     ```

   - 下载安装包（x86-64）

     ```
     wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail-linux64.tar.gz
     ```

   - 下载安装包（ARM）

     ```
     wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/aarch64/logtail-linux64.tar.gz
     ```

3. 将安装脚本和安装包拷贝至待安装Logtail的目标服务器上。
4. 替换*${your_region_name}*后，在目标服务器上执行安装命令。

   各个地域对应的*${your_region_name}*请参见Logtail安装参数。例如：

   - 华东 1（杭州）内网对应的*${your_region_name}*为cn-hangzhou。
   - 华东 1（杭州）公网对应的*${your_region_name}*为cn-hangzhou-internet。
   - 华东 1（杭州）全球加速对应的*${your_region_name}*为cn-hangzhou-acceleration。

   ```
   chmod +x logtail.sh; ./logtail.sh install-local ${your_region_name}
   ```

> **说明** 如果您要离线升级Logtail，可在下载最新版本的安装包后，执行 `chmod +x logtail.sh; ./logtail.sh upgrade-local` 命令。

查看Logtail版本

Logtail会将版本信息记录在*/usr/local/ilogtail/app_info.json*文件中的logtail_version字段。

- 命令

  ```
  cat /usr/local/ilogtail/app_info.json
  ```

- 返回结果

  ```
  {
     "UUID" : "0DF18E97-0F2D-486F-B77F-*********",
     "hostname" : "david*******",
     "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*********_1515129548",
     "ip" : "**********",
     "logtail_version" : "0.16.30",
     "os" : "Linux; 2.6.32-220.23.2.ali1113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013; x86_64",
     "update_time" : "2020-01-05 13:19:08"
  }
  ```

在线升级Logtail

您可以通过Logtail安装脚本（logtail.sh）升级Logtail，Logtail安装脚本会根据已经安装的Logtail配置信息自动选择合适的方式进行升级。

> **说明** 升级过程中会短暂停止Logtail。升级只覆盖必要的文件，配置文件以及Checkpoint文件将会被保留，升级期间日志不会丢失。

1. 执行以下命令升级Logtail。

```
wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh
sudo ./logtail.sh upgrade
```

2. 确认升级结果。

   显示类似信息表示升级成功。

```
Stop logtail successfully.
ilogtail is running
Upgrade logtail success
{
   "UUID" : "***",
   "hostname" : "***",
   "instance_id" : "***",
   "ip" : "***",
   "logtail_version" : "0.16.30",
   "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
   "update_time" : "2020-08-29 15:01:36"
}
```

离线升级Logtail

1. 登录能通过公网访问的服务器。
2. 替换*${your_region_name}*后，执行下载命令，下载安装脚本和安装包。

   各个地域公网对应的*${your_region_name}*请参见Logtail安装参数。例如华东 1（杭州）公网对应的*${your_region_name}*为cn-hangzhou-internet。

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail.sh
```

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail-linux64.tar.gz
```

3. 将安装脚本和安装包拷贝至待升级Logtail的目标服务器上。
4. 在目标服务器上执行升级命令。

```
chmod +x logtail.sh; ./logtail.sh upgrade-local
```

手动启动和停止Logtail

- 启动

  以root用户执行如下命令。

```
/etc/init.d/ilogtaild start
```

- 停止

  以root用户执行如下命令

```
/etc/init.d/ilogtaild stop
```

卸载Logtail

执行以下命令卸载Logtail。

```
wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh;chmod 755 logtail.sh;./logtail.sh uninstall
```

3.3.3. 安装Logtail（Windows系统）

本文介绍如何在Windows服务器上安装Logtail。

前提条件

- 已拥有一台及以上的服务器。

- 已根据服务器类型和所在地域，确定采集日志时所需的网络类型。更多信息，请参见选择网络。

支持的系统

支持如下Windows操作系统：

> **说明** 其中Microsoft Windows Server 2008和Microsoft Windows 7支持X86和X86_64，其他版本仅支持X86_64。

- Microsoft Windows Server 2008
- Microsoft Windows Server 2012
- Microsoft Windows Server 2016
- Microsoft Windows Server 2019
- Microsoft Windows 7
- Microsoft Windows 10
- Microsoft Windows Server Version 1909
- Microsoft Windows Server Version 2004

安装Logtail

1. 下载安装包。
   - 中国地域的下载地址：Logtail安装包。
   - 海外地域的下载地址：Logtail安装包。
2. 解压缩 `logtail_installer.zip` 到当前目录。
3. 根据网络类型和日志服务Project所在地域选择并执行安装命令。

   以管理员身份运行Windows Powershell或cmd，进入 `logtail_installer` 目录（您的安装包的解压目录），执行安装命令。

地域	阿里云内网（经典网络、VPC）	公网	全球加速
华北 1（青岛）	`.\logtail_installer.exe install cn-qingdao`	`.\logtail_installer.exe install cn-qingdao-internet`	`.\logtail_installer.exe install cn-qingdao-acceleration`
华北 2（北京）	`.\logtail_installer.exe install cn-beijing`	`.\logtail_installer.exe install cn-beijing-internet`	`.\logtail_installer.exe install cn-beijing-acceleration`
华北 3（张家口）	`.\logtail_installer.exe install cn-zhangjiakou`	`.\logtail_installer.exe install cn-zhangjiakou-internet`	`.\logtail_installer.exe install cn-zhangjiakou-acceleration`
华北 5（呼和浩特）	`.\logtail_installer.exe install cn-huhehaote`	`.\logtail_installer.exe install cn-huhehaote-internet`	`.\logtail_installer.exe install cn-huhehaote-acceleration`
华北6（乌兰察布）	`.\logtail_installer.exe install cn-wulanchabu`	`.\logtail_installer.exe install cn-wulanchabu-internet`	`.\logtail_installer.exe install cn-wulanchabu-acceleration`
华东 1（杭州）	`.\logtail_installer.exe install cn-hangzhou`	`.\logtail_installer.exe install cn-hangzhou-internet`	`.\logtail_installer.exe install cn-hangzhou-acceleration`
华东 2（上海）	`.\logtail_installer.exe install cn-shanghai`	`.\logtail_installer.exe install cn-shanghai-internet`	`.\logtail_installer.exe install cn-shanghai-acceleration`
华南 1（深圳）	`.\logtail_installer.exe install cn-shenzhen`	`.\logtail_installer.exe install cn-shenzhen-internet`	`.\logtail_installer.exe install cn-shenzhen-acceleration`
华南2（河源）	`.\logtail_installer.exe install cn-heyuan`	`.\logtail_installer.exe install cn-heyuan-internet`	`.\logtail_installer.exe install cn-heyuan-acceleration`

地域	阿里云内网（经典网络、VPC）	公网	全球加速
华南3（广州）	`.\logtail_installer.exe install cn-guangzhou`	`.\logtail_installer.exe install cn-guangzhou-internet`	`.\logtail_installer.exe install cn-guangzhou-acceleration`
西南 1（成都）	`.\logtail_installer.exe install cn-chengdu`	`.\logtail_installer.exe install cn-chengdu-internet`	`.\logtail_installer.exe install cn-chengdu-acceleration`
中国（香港）	`.\logtail_installer.exe install cn-hongkong`	`.\logtail_installer.exe install cn-hongkong-internet`	`.\logtail_installer.exe install cn-hongkong-acceleration`
美国 （硅谷）	`.\logtail_installer.exe install us-west-1`	`.\logtail_installer.exe install us-west-1-internet`	`.\logtail_installer.exe install us-west-1-acceleration`
美国（弗吉尼亚）	`.\logtail_installer.exe install us-east-1`	`.\logtail_installer.exe install us-east-1-internet`	`.\logtail_installer.exe install us-east-1-acceleration`
新加坡	`.\logtail_installer.exe install ap-southeast-1`	`.\logtail_installer.exe install ap-southeast-1-internet`	`.\logtail_installer.exe install ap-southeast-1-acceleration`
澳大利亚（悉尼）	`.\logtail_installer.exe install ap-southeast-2`	`.\logtail_installer.exe install ap-southeast-2-internet`	`.\logtail_installer.exe install ap-southeast-2-acceleration`
马来西亚（吉隆坡）	`.\logtail_installer.exe install ap-southeast-3`	`.\logtail_installer.exe install ap-southeast-3-internet`	`.\logtail_installer.exe install ap-southeast-3-acceleration`
印度尼西亚（雅加达）	`.\logtail_installer.exe install ap-southeast-5`	`.\logtail_installer.exe install ap-southeast-5-internet`	`.\logtail_installer.exe install ap-southeast-5-acceleration`
菲律宾（马尼拉）	`.\logtail_installer.exe install ap-southeast-6`	`.\logtail_installer.exe install ap-southeast-6-internet`	`.\logtail_installer.exe install ap-southeast-6-acceleration`
泰国（曼谷）	`.\logtail_installer.exe install ap-southeast-7`	`.\logtail_installer.exe install ap-southeast-7-internet`	`.\logtail_installer.exe install ap-southeast-7-acceleration`
印度（孟买）	`.\logtail_installer.exe install ap-south-1`	`.\logtail_installer.exe install ap-south-1-internet`	`.\logtail_installer.exe install ap-south-1-acceleration`
日本（东京）	`.\logtail_installer.exe install ap-northeast-1`	`.\logtail_installer.exe install ap-northeast-1-internet`	`.\logtail_installer.exe install ap-northeast-1-acceleration`
韩国（首尔）	`.\logtail_installer.exe install ap-northeast-2`	`.\logtail_installer.exe install ap-northeast-2-internet`	`.\logtail_installer.exe install ap-northeast-2-acceleration`
德国（法兰克福）	`.\logtail_installer.exe install eu-central-1`	`.\logtail_installer.exe install eu-central-1-internet`	`.\logtail_installer.exe install eu-central-1-acceleration`
阿联酋（迪拜）	`.\logtail_installer.exe install me-east-1`	`.\logtail_installer.exe install me-east-1-internet`	`.\logtail_installer.exe install me-east-1-acceleration`
英国（伦敦）	`.\logtail_installer.exe install eu-west-1`	`.\logtail_installer.exe install eu-west-1-internet`	`.\logtail_installer.exe install eu-west-1-acceleration`

地域	阿里云内网（经典网络、VPC）	公网	全球加速
俄罗斯（莫斯科）	`.\logtail_installer.exe install rus-west-1`	`.\logtail_installer.exe install rus-west-1-internet`	`.\logtail_installer.exe install rus-west-1-acceleration`

> **说明** 日志服务无法获取非本账号下ECS、自建IDC或其他云厂商服务器的属主信息，在安装Logtail后需手动配置用户标识。更多信息，请参见配置用户标识。

安装路径

执行Logtail安装命令后，默认安装Logtail到指定路径下，不支持修改。

- 32位Windows系统：*C:\Program Files\Alibaba\Logtail*
- 64位Windows系统：*C:\Program Files (x86)\Alibaba\Logtail*

> **说明** Windows 64位操作系统支持运行32/64位应用程序，但是出于兼容性考虑，在Windows 64位操作系统上，Windows会使用单独的x86目录来存放32位应用程序。
>
> Windows版本的Logtail是32位程序，所以在64位操作系统上的安装目录为*Program Files (x86)*。如果日志服务推出64位的Windows版本Logtail，会自动安装到*Program Files*目录下。

查看Logtail版本

您可以通过安装路径下的*app_info.json*文件中的 `logtail_version` 字段查看Logtail版本。

例如，以下内容表示Logtail的版本号为1.0.0.0。

```
{
    "logtail_version" : "1.0.0.0"
}
```

升级Logtail

升级的操作和安装Logtail的操作相同，您只需要下载并解压最新的安装包，然后按照步骤执行安装即可。更多信息，请参见安装Logtail。

> **说明** 升级相当于自动卸载并重新安装，会删除掉您原先安装目录中的内容，请您在执行升级前做好备份工作。

手动启动和停止Logtail

1. 选择**开始 > 控制面板 > 管理工具 > 服务**。
2. 在**服务**对话框中，选择对应的服务。
   - 如果是0.x.x.x版本，选择LogtailWorker服务。
   - 如果是1.0.0.0及以上版本，选择LogtailDaemon服务。
3. 右键选择对应的操作：**启动**、**停止**、**重新启动**。

卸载Logtail

以管理员身份运行Windows Powershell或cmd进入 `logtail_installer` 目录（安装包的解压目录），执行如下命令。

```
.\logtail_installer.exe uninstall
```

卸载成功后，您的Logtail的安装目录会被删除，但仍有部分配置被保留在*C:\LogtailData*目录中，您可以根据实际情况进行手动删除。遗留信息包括：

- *checkpoint*：存放所有Logtail插件的Checkpoint信息。只有您使用了Logtail插件后，才会出现此文件。
- *user_config.d*：存放本地采集配置的目录。

  其中以.json结尾的文件会被视为采集配置，格式类似于*/usr/local/ilogtail/user_log_config.json*。
- *logtail_check_point*：存放Logtail主体部分的Checkpoint信息。
- *users*：存放您所配置的用户标识文件。

3.3.4. 设置Logtail启动参数

为防止Logtail消耗过多服务器资源，影响其他服务运行，日志服务对Logtail采集性能做了限制。当您需要提升Logtail采集性能时，可修改Logtail启动参数。

设置场景

遇到以下场景时，可修改Logtail启动参数。

- 需要采集的日志文件数目大（同时采集的文件数超过100个或所监控的目录下的文件数超过5000个），占用大量内存。
- 日志数据流量大（例如极简模式下超过2 MB/s，正则模式下超过1 MB/s），导致CPU占用率高。
- Logtail发送数据到日志服务的速率超过10 MB/s。

推荐参数值

根据实际经验推荐如下参数配置，适用于普通JSON文件的采集场景。完整正则模式和分隔符模式的性能与JSON模式相近，极简模式性能为JSON模式的5倍。由于数据、规则的复杂度、采集目录和文件的数量都会对CPU和MEM消耗带来影响，请参照下述表格并结合实际情况按需调整。

- 主机环境

参数	默认的采集速率	采集速率大于10 MB/s	采集速率大于20 MB/s	采集速率大于40 MB/s
cpu_usage_limit	0.4	1	2	4
mem_usage_limit	384	1024	2048	4096
max-bytes-per-sec	20971520	209715200	209715200	209715200
process_thread_count	1	2	4	8
send_request_concurrency	4	20	40	80

- 容器或Kubernetes环境

环境变量	默认的采集速率	采集速率大于10 MB/s	采集速率大于20 MB/s	采集速率大于40 MB/s
cpu_usage_limit	2	3	5	9
mem_usage_limit	512	1024	2048	4096
max_bytes_per_sec	209715200	209715200	209715200	209715200
process_thread_count	1	2	4	8
send_request_concurrency	20	20	40	80
resources.limits.cpu	500M	1000M	2000M	4000M
resources.limits.memory	1 Gi	2 Gi	3 Gi	5 Gi

在容器或Kubernetes环境下，您需要通过修改daemonset环境变量来修改Logtail启动参数。部分环境引用configmap，configmap路径为**configmap > kube-system > alibaba-log-configuration**。同时还需调整**daemonset > kube-system > logtail-ds**中的resources.limits.cpu和resources.limits.memory，避免Container资源超限。

按照上述表格中的采集速率大于40 MB/s列配置Logtail启动参数时，Logtail的采集性能接近极限，继续增加线程对性能提升效果不显著。采集端的性能极限说明如下表所示。

> **说明** 因测试环境与生产环境不同，实际采集性能可能存在差异。

采集模式	性能极限
极简模式	440 MB/s
完整正则模式	70 MB/s
分隔符模式	75 MB/s
JSON模式	75 MB/s

设置启动参数

1. 在安装Logtail的服务器上，打开*/usr/local/ilogtail/ilogtail_config.json*文件。

此步骤适用于主机环境。

在容器或Kubernetes环境下，您需要通过修改daemonset环境变量来修改Logtail启动参数。部分环境引用configmap，configmap路径为**configmap > kube-system > alibaba-log-configuration**

2. 根据需求设置启动参数。

启动参数示例如下：

```
{
    ...
    "cpu_usage_limit" : 0.4,
    "mem_usage_limit" : 384,
    "max_bytes_per_sec" : 20971520,
    "process_thread_count" : 1,
    "send_request_concurrency" : 4,
    "buffer_file_num" : 25,
    "buffer_file_size" : 20971520,
    "buffer_file_path" : "",
    ...
}
```

说明

- 下表中只列出您需要关注的常用启动参数，未列出的启动参数，保持默认配置即可。
- 您可以根据需要新增或修改指定启动参数。

Logtail启动参数

参数	类型	说明	示例
cpu_usage_limit	double	CPU使用阈值，以单核计算。取值如下： - 取值范围：0.1~当前机器的CPU核心数 - 默认值：2 **警告** cpu_usage_limit为软限制，实际Logtail占用的CPU可能超过限制值，超限5分钟后将触发熔断保护，Logtail自动重启。 例如设置为0.4，表示日志服务将尽可能限制Logtail的CPU使用为CPU单核的40%，超出后Logtail自动重启。 一般情况下，通过极简模式采集日志时，单核处理能力约100 MB/s；通过完整正则模式采集日志时，单核处理能力约20 MB/s 。	"cpu_usage_limit" : 0.4
mem_usage_limit	int	内存使用阈值。取值如下： - 取值范围：128（MB）~ 2048（MB） - 默认值：384（MB） **警告** mem_usage_limit为软限制，实际Logtail占用的内存可能超过限制值，超限5分钟后将触发熔断保护，Logtail自动重启。 mem_usage_limit与监控的文件数的关系如下： - 使用默认值时，每台服务器上的每个Logtail采集配置最多可监控19,200个文件，每台服务器上的Logtail客户端最多可监控192,000个文件。 - 使用最大值时，每台服务器上的每个Logtail采集配置最多可监控100,000个文件，每台服务器上的Logtail客户端最多可监控1,000,000个文件。 相关计算公式如下： - 每个Logtail采集配置可监控的最大文件数=mem_usage_limit/100×5000 - 每个Logtail客户端可监控的最大文件数=mem_usage_limit/100×50,000	"mem_usage_limit" : 384

参数	类型	说明	示例
max_bytes_per_sec	int	每秒钟Logtail发送原始数据的流量限制。取值如下： ○ 取值范围：1024（Byte/s）~52428800（Byte/s） ○ 默认值：20971520（Byte/s） 例如设置为2097152，表示Logtail发送数据的速率为2 MB/s。 注意 设置的值超过20971520 Byte/s（20MB/s），表示不限速。	"max_bytes_per_sec"：2097152
process_thread_count	int	Logtail处理数据的线程数。 取值如下： ○ 取值范围：1~64 ○ 默认值：1 一般情况下，可以处理极简模式下24 MB/s的数据写入或完整正则模式12 MB/s的数据写入。默认情况下无需调整该参数取值。	"process_thread_count"：1
send_request_concurrency	int	异步并发的个数。取值如下： ○ 取值范围：1~1000 ○ 默认值：20 如果写入TPS很高，可以设置更高的异步并发个数。可以按照一个并发支持0.5 MB/s~1 MB/s网络吞吐来计算，实际根据网络延时而定。 说明 设置异步并发个数过高容易导致网络端口占用过多，需调整TCP相关参数。	"send_request_concurrency"：4
buffer_file_num	int	限制缓存文件的最大数目。取值如下： ○ 取值范围：1~100 ○ 默认值：25 遇到网络异常、写入配额超限等情况时，Logtail将实时解析后的日志写入本地文件（安装目录下）缓存起来，等待恢复后尝试重新发送。	"buffer_file_num"：25
buffer_file_size	int	单个缓存文件允许的最大字节数。取值如下： ○ 取值范围：1048576（Byte）~104857600（Byte） ○ 默认值：20971520（Byte） buffer_file_size*buffer_file_num是缓存文件可以实际使用的最大磁盘空间。	"buffer_file_size"：20971520
buffer_file_path	String	缓存文件存放目录。 默认值为空，即缓存文件存放于logtail安装目录*/usr/local/ilogtail*下。 当您设置此参数后，需手动将原目录下名为*logtail_buffer_file_*的文件移动到此目录，以保证Logtail可以读取到该缓存文件并在发送后进行删除。	"buffer_file_path"：""
bind_interface	String	本机绑定的网卡名。默认值为空，自动绑定可用的网卡。 如果设置为指定的网卡（例如eth1），则表示Logtail将强制使用该网卡上传日志。 只支持Linux版本。	"bind_interface"：""
check_point_filename	String	Logtail的checkpoint文件的保存路径， 默认值：*/tmp/logtail_check_point*。	"check_point_filename"：/tmp/logtail_check_point
check_point_dump_interval	int	Logtail更新Checkpoint文件的周期，默认值：900，单位：秒。即默认情况下每15分钟更新一次Checkpoint文件。 仅支持Logtail 1.0.19及以上版本。	"check_point_dump_interval"：900
user_config_file_path	String	Logtail配置文件的保存路径，默认为进程binary所在目录，文件名为*user_log_config.json*。	"user_config_file_path"：user_log_config.json

参数	类型	说明	示例
docker_file_cache_path	String	该文件记录了容器文件到宿主机文件的路径映射，默认为*/usr/local/ilogtail/docker_path_config.json*。 仅支持Logtail 0.16.54及以上版本。	"docker_file_cache_path": /usr/local/ilogtail/docker_path_config.json
discard_old_data	Boolean	是否丢弃历史日志。默认值：true，表示丢弃距离当前时间超过12小时的日志。	"discard_old_data" : true
ilogtail_discard_interval	int	丢弃历史日志距离当前时间的阈值。默认值：43200（12小时），单位：秒。	"ilogtail_discard_interval": 43200
working_ip	String	Logtail上报本服务器的IP地址。默认值为空，表示自动从本服务器获取IP地址。	"working_ip" : ""
working_hostname	String	Logtail上报的本服务器的主机名。默认值为空，表示自动从本服务器获取主机名。	"working_hostname" : ""
max_read_buffer_size	long	每条日志读取的最大值。默认值：524288（512 KB），最大值：4194304（4MB）。单位：Byte。 如果您的单条日志超过524288 Byte，可修改此参数。	"max_read_buffer_size" : 524288
oas_connect_timeout	long	Logtail发起获取Logtail配置、访问密钥等请求时，连接阶段的超时时间。默认值：5，单位：秒。 网络条件较差，建立连接时间过长时可修改此参数。	"oas_connect_timeout" : 5
oas_request_timeout	long	Logtail发起获取Logtail配置、访问密钥等请求时，整个请求阶段的超时时间。默认值：10，单位：秒。 网络条件较差，建立连接时间过长时可修改此参数。	"" : 10
data_server_port	long	设置data_server_port为443后，Logtail将通过HTTPS协议传输数据到日志服务。 仅支持Logtail 1.0.10及以上版本。	"data_server_port": 443
enable_log_time_auto_adjust	Boolean	设置enable_log_time_auto_adjust为true后，日志时间可自适应服务器本地时间。 出于数据安全考虑，日志服务会对请求（包括Logtail发起的请求）所携带的时间进行校验，拒绝与日志服务端时间相差超过15分钟的请求。Logtail发起请求时所携带的时间为服务器本地时间，当服务器本地时间被修改后（例如某些测试场景下需要调整本地时间为未来时间），Logtail请求将被拒绝，导致写入数据失败。您可以使用该参数实现日志时间自适应服务器本地时间。 仅支持Logtail 1.0.19及以上版本。 注意 ◦ 开启该功能后，日志时间将被加上日志服务端的时间与服务器本地时间的偏移量。由于偏移量只在请求被日志服务端拒绝时更新，因此可能出现日志服务端所查询到的日志的时间和日志实际的写入时间不一致的情况。 ◦ Logtail的部分逻辑依赖于系统时间的递增，建议在每次机器时间调整后重启Logtail。	"enable_log_time_auto_adjust": true
accept_multi_config	Boolean	是否允许多个Logtail配置采集同一个文件。默认值：false，表示不允许。 默认情况下，一个文件只能被一个Logtai配置采集，您可以通过该参数消除限制。每个Logtail配置的处理过程是独立的，当允许多个Logtai配置采集同一个文件时，需要消耗多倍的CPU、内存开销。 仅支持Logtail 0.16.26及以上版本。	"accept_multi_config": true

参数	类型	说明	示例
enable_checkpoint_sync_write	Boolean	是否开启sync写功能。默认值：false，表示不开启。 sync写功能主要用于搭配ExactlyOnce写入功能。开启ExactlyOnce写入功能后，Logtail会在本地磁盘记录细粒度的Checkpoint信息（文件级别）。但出于性能考虑，默认写入Checkpoint时不会调用sync落盘，所以如果机器重启导致buffer数据来不及写入磁盘时，可能导致Checkpoint丢失。此时，您可以设置enable_checkpoint_sync_write为true，开启sync写功能。更多信息，请参见附录：ExactlyOnce写入功能说明。 仅支持Logtail 1.0.20及以上版本。	"enable_checkpoint_sync_write": false
enable_env_ref_in_config	Boolean	是否启用采集配置环境变量替换功能。默认值：false。 开启该功能后，您可以在控制台的Logtail采集配置中使用 `${xxx}` 作为环境变量 `xxx` 的占位符。例如设置采集路径为 `/${xxx}/logs` ，环境变量为 `xxx=root` ，则生效的采集路径为 `/root/logs` 。 如果配置中需要使用 `${` 、 `}` ，则您可以使用 `$${` 、 `$}` 进行转义。 仅支持Logtail 1.0.31及以上版本。	"enable_env_ref_in_config": false
docker_config_update_interval	int	容器路径更新的最小时间间隔。默认值：3（1.0.32及以上版本）、10（1.0.32之前版本）。单位：秒。 与max_docker_config_update_times配合使用，任意一个参数达到阈值则不再更新容器路径。	"docker_config_update_interval": 3
max_docker_config_update_times	int	3分钟内更新容器路径最大次数。默认值：10（1.0.32及以上版本）、3（1.0.32之前版本）。默认情况下，3分钟内容器路径更新次数超过3次则不再更新容器路径。	"max_docker_config_update_times": 10

3. 重启Logtail使配置生效。

```
/etc/init.d/ilogtaild stop && /etc/init.d/ilogtaild start
```

重启后，您可以执行 `/etc/init.d/ilogtaild status` 命令检查Logtail状态。

附录：环境变量说明

环境变量与Logtail启动参数的对应关系如下，具体的参数说明请参见Logtail启动参数。

环境变量与Logtail启动参数对应关系

参数	环境变量	优先级	支持版本
cpu_usage_limit	cpu_usage_limit	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.32及以上版本
mem_usage_limit	mem_usage_limit	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.32及以上版本
max_bytes_per_sec	max_bytes_per_sec	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.32及以上版本
process_thread_count	process_thread_count	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.32及以上版本
send_request_concurrency	send_request_concurrency	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.32及以上版本
check_point_filename	check_point_filename或ALIYUN_LOGTAIL_CHECK_POINT_PATH	如果您通过环境变量和配置文件修改了Logtail启动参数，以环境变量为准。	Logtail 0.16.36及以上版本
docker_file_cache_path	docker_file_cache_path	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.54及以上版本

参数	环境变量	优先级	支持版本
user_config_file_path	user_config_file_path	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
discard_old_data	discard_old_data	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
working_ip	working_ip或ALIYUN_LOGTAIL_WORKING_IP	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
working_hostname	working_hostname或ALIYUN_LOGTAIL_WORKING_HOSTNAME	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
max_read_buffer_size	max_read_buffer_size	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
oas_connect_timeout	oas_connect_timeout	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
oas_request_timeout	oas_request_timeout	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
data_server_port	data_server_port	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
accept_multi_config	accept_multi_config	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 0.16.56及以上版本
enable_log_time_auto_adjust	enable_log_time_auto_adjust	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 1.0.19及以上版本
check_point_dump_interval	check_point_dump_interval	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 1.0.19及以上版本
enable_checkpoint_sync_write	enable_checkpoint_sync_write	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 1.0.20及以上版本
docker_config_update_interval	docker_config_update_interval或ALIYUN_LOGTAIL_DOCKER_CONFIG_UPDATE_INTERVAL	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 1.0.29及以上版本
max_docker_config_update_times	max_docker_config_update_times或ALIYUN_LOGTAIL_MAX_DOCKER_CONFIG_UPDATE_TIMES	如果您通过环境变量和配置文件修改了Logtail启动参数，以配置文件为准。	Logtail 1.0.29及以上版本

3.4. 机器组

3.4.1. 简介

机器组是包含多台服务器的虚拟分组，日志服务通过机器组的方式管理所有需要通过Logtail采集日志的服务器。

日志服务支持通过一个Logtail采集配置来采集多台服务器上的日志，您可以将这些服务器加入到同一个机器组，并将Logtail采集配置应用到该机器组。

您可以通过如下两种方法定义一个机器组。

- IP地址：在机器组中添加服务器的IP地址，通过IP地址识别服务器。
- 自定义标识：定义属于机器组的一个标识，在对应服务器上配置对应标识进行关联。

> **说明** 如果您的服务器为其他云厂商服务器、自建IDC、其他账号下的ECS，则在添加到机器组前，需先在服务器上配置用户标识，详情请参见配置用户标识 。

IP地址机器组

您可以通过添加服务器IP地址的方式，将多台服务器添加到一个机器组中。

- 如果您使用ECS服务器，且没有绑定过主机名、没有更换过网络类型，则可以在机器组中配置ECS服务器的私网IP地址。
- 其他情况下，请在机器组中配置Logtail自动获取到的IP地址。该IP地址记录在服务器*app_info.json*文件中的ip字段中。Logtail自动获取服务器IP地址的方式如下所示。
  - 如果已在服务器*/etc/hosts*文件中设置了主机名与IP地址绑定，则自动获取绑定的IP地址。
  - 如果未在服务器*/etc/hosts*文件中设置主机名与IP地址绑定，则自动获取本机第一块网卡的IP地址。

> **说明** 采集日志时是否使用阿里云内网，与机器组中填写的IP地址是否为私网IP地址无关。如果您的服务器是阿里云ECS云服务器，并且安装Logtail时选择**阿里云内网（经典网络/VPC）**模式，才会通过阿里云内网采集日志到日志服务。

自定义标识机器组

使用自定义标识动态定义机器组，在以下场景中具有明显优势。

- VPC等自定义网络环境中，可能出现不同服务器IP地址冲突的问题，导致日志服务无法管理Logtail。使用自定义标识可以避免此类情况的发生。
- 多台服务器通过同一个自定义标识实现机器组弹性伸缩。您只需为新增的服务器配置相同的自定义标识，日志服务可自动识别，并将其添加至机器组中。

通常情况下，系统由多个模块组成，每个模块都可以进行独立的水平扩展，即支持添加多台服务器。为每个模块分别创建机器组，可以实现日志分类采集。因为需要为每个模块分别定义自定义标识，即在各个模块的服务器上配置各自所属的自定义标识。例如常见网站分为前端HTTP请求处理模块、缓存模块、逻辑处理模块和存储模块，其自定义标识可以分别定义为http_module、cache_module、logic_module和store_module。

3.4.2. 配置用户标识

本文介绍如何在服务器上配置阿里云账号ID为用户标识。

前提条件

- 已有可用的服务器。

  此处的服务器是指与日志服务属于不同账号的ECS、其他云厂商的服务器或自建IDC。
- 已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）、安装Logtail（Windows系统）。

背景信息

如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您要通过Logtail采集该服务器日志，需要在服务器上安装Logtail后，配置日志服务所在阿里云账号ID为用户标识，表示该账号有权限通过Logtail采集该服务器日志。否则在机器组中会显示服务器心跳失败，导致Logtail无法采集日志到日志服务。

步骤一：获取日志服务所在的阿里云账号ID

1. 使用日志服务所在的阿里云账号登录阿里云。
2. 打开云命令行。
3. 执行以下命令，获取日志服务所在的阿里云账号ID。

```
echo $ALIBABA_CLOUD_ACCOUNT_ID
```

步骤二：配置用户标识

1. 登录服务器。
2. 配置用户标识。

> 注意
> - 如果*/etc/ilogtail/users*目录不存在，请手动创建目录。
> - 新增、删除用户标识后，1分钟之内即可生效。

- Linux系统

  在*/etc/ilogtail/users*目录下，创建账号ID同名文件。

  ```
  touch /etc/ilogtail/users/17****745
  ```

- Windows系统

  在*C:\LogtailData\users*目录下，创建账号ID同名文件。

  - 使用Windows PowerShell

    ```
    ni C:\LogtailData\users\17******745
    ```

  - 使用命令提示符（cmd）

    ```
    type nul > C:\LogtailData\users\17******745
    ```

多账号场景

当您使用多个阿里云账号下的日志服务对同一台服务器进行日志采集时，您可以在同一台服务器上创建多个用户标识文件。例如：

```
touch /etc/ilogtail/users/17****742
touch /etc/ilogtail/users/17****743
```

删除用户标识

> 注意 请及时删除服务器上多余的用户标识文件，回收不再使用的采集权限。

- Linux系统

  执行如下命令删除用户标识文件，即可删除对应的用户标识。

  ```
  rm /etc/ilogtail/users/17****745
  ```

- Windows系统

  执行如下命令删除用户标识文件，即可删除对应的用户标识。

  ```
  del C:\LogtailData\users\17******745
  ```

后续步骤

配置阿里云账号ID为用户标识后，您可以创建机器组。更多信息，请参见创建IP地址机器组或创建用户自定义标识机器组。

3.4.3. 创建IP地址机器组

日志服务支持使用服务器IP地址定义机器组。本文介绍如何在日志服务控制台上创建IP地址机器组。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和Logstore。
- 已有一台及以上的服务器。
  - 如果是阿里云ECS，请确保ECS和日志服务属于同账号且同地域。
  - 如果是与日志服务属于不同账号的ECS、其他云厂商的服务器或自建IDC，请先配置用户标识。更多信息，请参见配置用户标识。
- 已在服务器上安装Logtail。更多信息，请参见安装Logtail（ECS实例）。

操作步骤

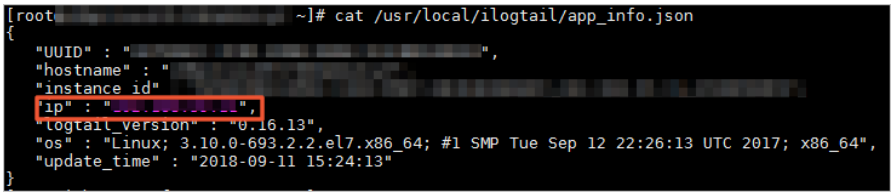
1. 获取服务器IP地址。

   Logtail自动获取的IP地址记录在*app_info.json*文件的ip字段中。

   在已安装Logtail的服务器上查看*app_info.json*文件，路径为：

   - Linux：*/usr/local/ilogtail/app_info.json*
   - Windows x64：*C:\Program Files (x86)\Alibaba\Logtail\app_info.json*
   - Windows x32：*C:\Program Files\Alibaba\Logtail\app_info.json*

   例如，在Linux中查看服务器IP地址，如下图所示。

2.
3. 在**Project列表**区域，单击目标Project。
4. 在左侧导航栏中，选择**资源 > 机器组**。
5. 选择**机器组**右侧的 **> 创建机器组**。
6. 在**创建机器组**对话框中，配置如下参数，单击**确定**。

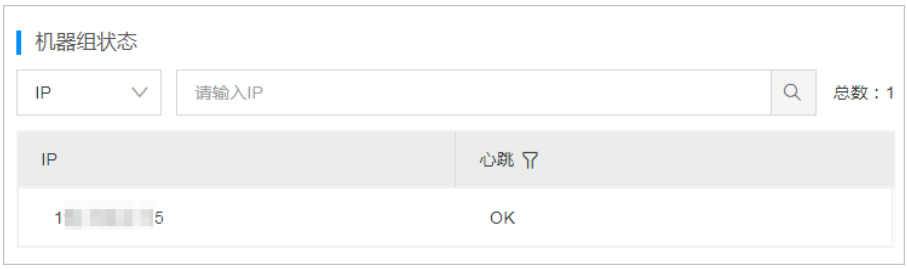
参数	说明
名称	名称只能包含小写字母、数字、连字符（-）和下划线（_）且必须以小写字母或数字开头和结尾，长度为3~128字节。 **注意** 创建后，不支持修改机器组名称，请谨慎填写。
机器组标识	选择**IP地址**。
机器组Topic	机器组Topic用于区分不同服务器产生的日志数据。更多信息，请参见日志主题。
IP地址	填写步骤1中获取到的服务器IP地址。 **说明** - 机器组中存在多台服务器时，IP地址之间请用换行符分割。 - 请勿将Windows服务器和Linux服务器添加到同一机器组中。

   完成创建后，机器组大约2分钟后生效。

7. 查看机器组状态。
   i. 在机器组列表中，单击目标机器组。

ii. 在**机器组配置**页面，查看服务器及其状态。

心跳为**OK**表示服务器与日志服务的连接正常，如果显示**FAIL**请参见Logtail机器组无心跳。



3.4.4. 创建用户自定义标识机器组

日志服务支持使用用户自定义标识动态定义机器组，本文介绍如何创建自定义标识机器组。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和Logstore。
- 已有一台及以上的服务器。
  - 如果是阿里云ECS，请确保ECS和日志服务属于同账号且同地域。
  - 如果是与日志服务属于不同账号的ECS、其他云厂商的服务器或自建IDC，请先配置用户标识。更多信息，请参见配置用户标识。
- 已在服务器上安装Logtail。更多信息，请参见安装Logtail（ECS实例）。

背景信息

用户自定义标识机器组在以下场景中具有明显优势：

- 在VPC等自定义网络环境中，可能出现不同服务器IP地址冲突的问题，导致日志服务无法管理Logtail。使用自定义标识可以避免此类情况的发生。
- 多台服务器通过同一个自定义标识实现机器组弹性伸缩。您只需为新增的服务器配置相同的自定义标识，日志服务可自动识别，并将其添加至机器组中。

操作步骤

1. 在指定目录下创建*user_defined_id*文件。
   - Linux服务器：*/etc/ilogtail/user_defined_id*
   - Windows服务器：*C:\LogtailData\user_defined_id*
2. 在服务器上配置用户自定义标识。

   > **说明**
   > - 同一机器组中不允许同时存在Linux服务器、Windows服务器，请勿在Linux和Windows服务器上配置相同的用户自定义标识。
   > - 一个服务器可配置多个用户自定义标识，标识之间以换行符分割。
   > - 如果目录*/etc/ilogtail/*、*C:\LogtailData*或文件*/etc/ilogtail/user_defined_id*、*C:\LogtailData\user_defined_id*不存在，请手动创建。

   - Linux服务器

     在*/etc/ilogtail/user_defined_id*文件中配置用户自定义标识。例如：您要配置用户自定义标识为`userdefined`，则执行如下命令编辑文件，在文件中输入`userdefined`，并保存。

     ```
     vim /etc/ilogtail/user_defined_id
     ```

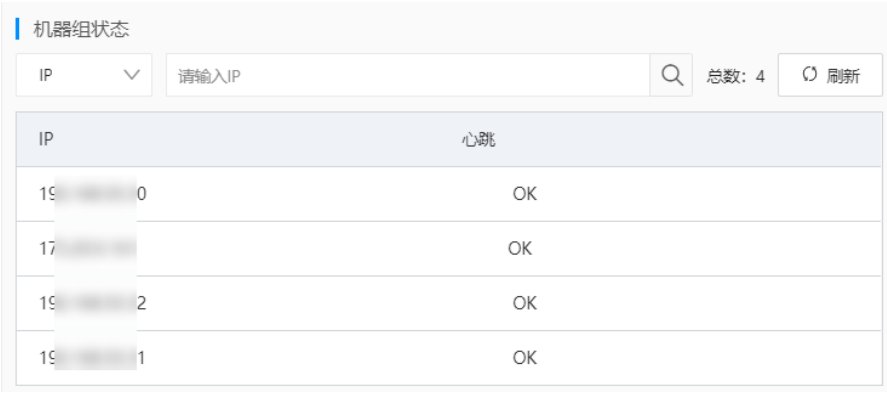
   - Windows服务器

     在*C:\LogtailData\user_defined_id*文件中配置用户自定义标识。例如：您要配置用户自定义标识为`userdefined_windows`，则在*C:\LogtailData\user_defined_id*文件中输入`userdefined_windows`，并保存。
3.
4. 在**Project列表**区域，单击目标Project。
5. 在左侧导航栏中，选择**资源** > **机器组**。
6. 选择**机器组**右侧的 > **创建机器组**。
7. 在**创建机器组**对话框中，配置如下参数，单击**确定**。

参数	说明
名称	名称只能包含小写字母、数字、连字符（-）和下划线（_）且必须以小写字母或数字开头和结尾，长度为2~128字符。 **注意** 创建后，不支持修改机器组名称，请谨慎填写。
机器组标识	选择**用户自定义标识**。
机器组Topic	机器组Topic用于区分不同服务器产生的日志数据。更多信息，请参见日志主题。
用户自定义标识	配置为中配置的用户自定义标识。

8. 查看机器组状态。
   i. 在机器组列表中，单击目标机器组。
   ii. 在**机器组配置**页面，可看到使用相同用户自定义标识的服务器及其心跳状态。
   - 机器组状态中的IP列表，即为使用相同用户自定义标识的服务器的IP地址。例如：

     假设当前为用户自定义标识机器组，用户自定义标识为userdefined，机器组状态中的IP分别为10.10.10.10、10.10.10.11、10.10.10.12。则表示您在这三个服务器上创建了相同的用户自定义标识userdefined。如果您需要新增10.10.10.13服务器，则只需要在该服务器上创建用户自定义标识userdefined，即可在机器组状态中看到该服务器。
   - **心跳**为OK表示服务器与日志服务的连接正常，如果显示**FAIL**请参见Logtail机器组无心跳。



禁用用户自定义标识

如果您要恢复服务器IP地址作为标识，请删除user_defined_id文件，1分钟之内即可生效。

- Linux系统

```
rm -f /etc/ilogtail/user_defined_id
```

- Windows系统

```
del C:\LogtailData\user_defined_id
```

生效时间

新增、删除、修改user_defined_id文件后，默认情况下，1分钟之内即可生效。如果需要立即生效，请执行以下命令重启Logtail。

- Linux系统

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
```

- Windows系统
  i. 选择**开始 > 控制面板 > 管理工具 > 服务**。
  ii. 在**服务**对话框中，选择对应的服务。
  - 如果是0.x.x.x版本，选择LogtailWorker服务。
  - 如果是1.0.0.0及以上版本，选择LogtailDaemon服务。

  iii. 右键单击**重新启动**使配置生效。

3.4.5. 管理机器组

本文介绍如何在日志服务控制台上查看机器组、查看机器列表、修改机器组、查看机器组状态、应用Logtail采集配置、删除机器组等操作。

前提条件

已创建机器组。具体操作，请参见创建IP地址机器组或创建用户自定义标识机器组。

查看机器组

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，选择**资源 > 机器组**。
4. 在机器组列表中，查看当前Project下的所有机器组。

查看机器列表

1. 在机器组列表中，单击目标机器组。
2. 在**机器组状态**区域，查看该机器组下的所有机器。

修改机器组

1. 在机器组列表中，单击目标机器组。
2. 在**机器组配置**页面，单击**修改**。
3. 修改机器组的配置信息。
4. 单击**保存**。

查看机器组状态

您可以通过机器组的心跳状态确认安装在机器上的Logtail实例与日志服务是否连接成功。

> **说明** 创建机器组后，请耐心等待2分钟再查看心跳状态。

1. 在机器组列表中，单击目标机器组。
2. 在**机器组状态**区域，查看机器组状态。
   - 心跳状态为OK表示安装在机器上的Logtail实例与日志服务连接正常。
   - 心跳状态为FAIL表示安装在机器上的Logtail实例与日志服务连接异常，请根据页面提示进行排查。更多信息，请参见Logtail机器组无心跳排查思路。如果无法解决问题，请提交工单。

删除机器组

1. 在机器组列表中，单击目标机器组对应的图标，然后单击**删除**。
2. 在弹出的对话框中，单击**确认**。

应用Logtail采集配置

您可以通过日志服务创建Logtail采集配置，并将Logtail采集配置应用到机器组上。

1. 在机器组列表中，单击目标机器组。
2. 在**机器组配置**页面，单击**修改**。
3. 在**管理配置**区域，添加或移除Logtail采集配置到机器组，然后单击**保存**。

   添加Logtail配置到机器组后，该Logtail配置会被下发到机器组内的服务器的Logtail上。从机器组移除Logtail配置后，该Logtail配置会从机器组内的服务器的Logtail上移除。

3.4.6. 管理Logtail采集配置

本文介绍如何在日志服务控制台上创建、查看、修改及删除Logtail采集配置等操作。

创建Logtail采集配置

在日志服务控制台上创建Logtail采集配置，详情请参见采集文本日志。

查看Logtail采集配置

1.
2. 在**Project列表**区域，单击目标Project。

3. 在**日志存储 > 日志库**页签中，单击目标日志库前面的**>**，依次选择**数据接入 > Logtail配置**。
4. 单击目标Logtail采集配置，查看Logtail采集配置详情。

修改Logtail采集配置

1.
2. 在**Project列表**区域，单击目标Project。
3. 在**日志存储 > 日志库**页签中，单击目标日志库前面的**>**，依次选择**数据接入 > Logtail配置**。
4. 在**Logtail配置**列表中，单击目标Logtail采集配置。
5. 在**Logtail配置**页面，单击**修改**。
6. 根据需求，修改相关配置，并单击**保存**。

   详细参数说明请参见采集文本日志。

删除Logtail采集配置

1. 在**Logtail配置**列表中，单击目标Logtail采集配置右侧的图标，选择**删除**。
2. 在删除确认框中，单击**确认**。

   删除成功后，该Logtail采集配置与机器组解除绑定，Logtail停止采集该Logtail采集配置对应的日志。

> **说明** 删除logstore前，必须删除其对应的所有Logtail采集配置。

3.5. 采集文本日志

3.5.1. 概述

本文介绍通过Logtail采集服务器文本日志的配置流程和采集模式。

配置流程

日志服务提供配置向导，帮助您快速完成采集配置。

> **说明** 在创建Logtail配置前，建议先了解Logtail的使用限制。更多信息，请参见Logtail限制说明。
>
> 如果Logtail的默认设置不满足您的采集需求，您可以修改Logtail的启动参数。更多信息，请参见设置Logtail启动参数。

采集模式

Logtail支持通过极简模式、正则模式、分隔符模式、JSON模式、Nginx模式 、IIS模式、Apache模式采集文本日志。

- 使用极简模式采集日志
- 使用完整正则模式采集日志
- 使用分隔符模式采集日志
- 使用JSON模式采集日志
- 使用Nginx模式采集日志
- 使用IIS模式采集日志
- 使用Apache模式采集日志

3.5.2. 使用极简模式采集日志

极简模式不对日志内容进行解析，每条日志都被作为一个整体被采集到日志服务中，极大简化了日志采集流程。本文介绍如何通过日志服务控制台创建极简模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。

背景信息

极简模式支持以下类型的文本日志：

- 单行文本日志

  默认一行日志内容为一条日志，即在日志文件中，以换行符分隔两条日志。该模式下，您只需指定文件目录和文件名称即可，Logtail会按照每行一条日志进行采集。

- 多行文本日志

  默认一条日志有多行内容。该模式下，您除了指定文件目录和文件名称外，还需配置日志样例和行首正则表达式，Logtail通过行首正则表达式去匹配一条日志的行首，未匹配部分为该条日志的一部分。

> **说明** 通过极简模式采集日志时，日志时间为采集日志时Logtail所在主机的系统时间。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**多行-文本日志**。

   此处以采集多行文本日志为例，如果您要采集单行文本日志，请选择**单行-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。

     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

     日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 创建Logtail配置，单击**下一步**。

参数	描述
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_*模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** • 目录通配符只支持星号（*）和半角问号（?）。 • 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 • 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集，请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**极简模式-多行**，可修改为其它模式。

日志样例	请务必使用实际场景的日志，便于日志服务自动提取行首正则表达式。例如：
	`[2020-10-01T10:30:01,000] [INFO] java.lang.Exception: exception happened`
	`at TestPrintStackTrace.f(TestPrintStackTrace.java:3)`
	`at TestPrintStackTrace.g(TestPrintStackTrace.java:7)`
	`at TestPrintStackTrace.main(TestPrintStackTrace.java:16)`
	如果您是使用极简模式采集单行文本日志，无需配置此参数。

参数	描述
行首正则表达式	Logtail通过行首正则表达式去匹配一条日志的行首，未匹配部分为该条日志的一部分。日志服务支持自动生成和手动输入行首正则表达式。 ○ 自动生成行首正则表达式 填写日志样例后，单击**自动生成**，生成行首正则表达式。 ○ 手动输入行首正则表达式 填写日志样例后，单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否正确。更多信息，请参见如何调试正则表达式。 如果您是使用极简模式采集单行文本日志，无需配置此参数。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。

参数	描述
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

3.5.3. 使用完整正则模式采集日志

如果您需要对日志内容做更多个性化的字段提取设置，可选择完整正则模式。本文介绍如何通过日志服务控制台创建完整正则模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**正则-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 创建Logtail配置，单击**下一步**。

参数	描述
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	日志所在的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** • 目录通配符只支持星号（*）和半角问号（?）。 • 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 • 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器的创建和销毁，并根据Tag采集指定容器的日志。关于容器文本日志采集，请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**完整正则模式**，可修改为其它模式。
单行模式	• 如果待采集的日志是单行日志，请打开**单行模式**开关，日志服务将逐行采集日志。 • 如果待采集的日志是多行日志（例如Java程序日志），请关闭**单行模式**开关，使用多行正则模式采集。
日志样例	请务必使用实际场景的日志，便于日志服务自动提取其中的正则表达式。日志样例请参见单行日志采集案例、多行日志采集案例。

参数	描述
行首正则表达式	如果您要采集多行日志，在关闭**单行模式**开关后，还需配置行首正则表达式。日志服务支持自动生成和手动输入行首正则表达式。 ○ 自动生成行首正则表达式 填写日志样例后，单击**自动生成**，生成行首正则表达式。 ○ 手动输入行首正则表达式 填写日志样例后，单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否正确。更多信息，请参见如何调试正则表达式。
提取字段	打开**提取字段**开关后，可通过正则表达式将日志内容提取为Key-Value对。
正则	打开**提取字段**开关后，需要配置。 ○ 自动生成正则表达式 在**日志样例**文本框中，选中需要提取的日志内容，单击**生成正则**，自动生成正则表达式。 ○ 手动输入正则表达式 单击**手动输入正则表达式**，手动配置正则表达式。配置完成后，单击**验证**即可验证您输入的正则表达式是否可以解析、提取日志样例。更多信息，请参见如何调试正则表达式。
日志抽取内容	打开**提取字段**开关后，需要配置。 通过正则表达式将日志内容提取为Value后，您需要为每个Value设置对应的Key。
使用系统时间	打开**提取字段**开关后，需要配置。 具体说明如下： ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要在日志抽取内容中指定time字段，并根据time字段的值配置时间转换格式。时间格式详情请参见时间格式。 **注意** ○ 当使用DaemonSet方式采集容器日志时，由于Logtail容器的时区为UTC，如果业务容器的时区为非UTC（例如设置了**容器与节点使用相同时区**），则必须在本Logtail配置的**高级选项**中，设置**时区属性**为**自定义时区**，并选择时区为业务容器的时区，否则日志时间将错误偏移。 ○ 默认情况下，日志服务中的日志时间戳精确到秒。如果原始日志中的时间字段具备更高的时间精度（毫秒、微秒或纳秒），并希望在日志服务中保留该时间精度，可在Logtail采集配置的扩展配置中添加enable_precise_timestamp参数完成设置。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。取值范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。

参数	描述
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为.***^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

单行日志采集案例

- 日志样例

```
127.0.0.1 - - [10/Sep/2018:12:36:49 +0800] "GET /index.html HTTP/1.1" 200 612 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36"
```

- 正则表达式

```
(\S+)\s-\s(\S+)\s\[([^]]+)]\s"(\w+)\s(\S+)\s([^"]+)"\s(\d+)(\s\d+)[^-]+([^"]+)"\s"([^"]+).*
```

多行日志采集案例

- 日志样例

```
[2018-10-01T10:30:01,000] [INFO] java.lang.Exception: exception happened
    at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
    at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
    at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
```

- 行首正则表达式

```
\[\d+-\d+-\w+:\d+:\d+,\d+]\s\[\w+]\s.*
```

- 正则表达式

```
\[(\S+)]\s\[(\S+)]\s(.*)
```

3.5.4. 使用Nginx模式采集日志

Nginx日志是运维网站的重要信息，日志服务支持通过Nginx模式快速采集Nginx日志并进行多维度分析。本文介绍如何通过日志服务控制台创建Nginx模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**Nginx-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

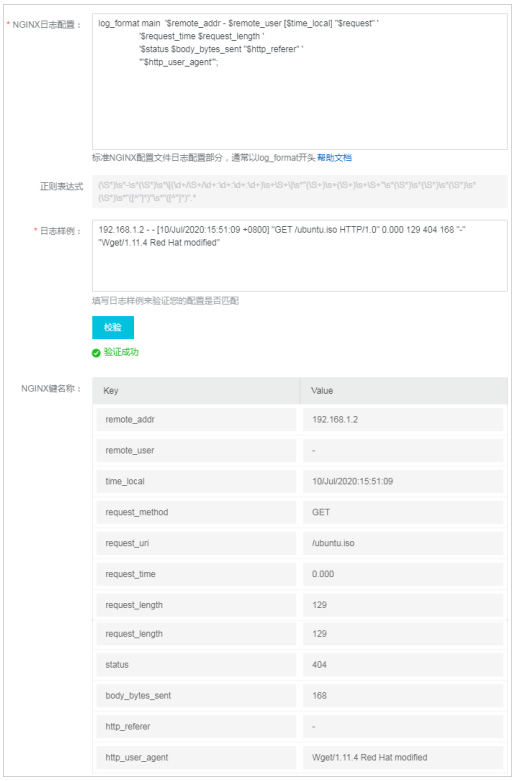
        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 创建Logtail配置，单击**下一步**。



参数	说明
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。

参数	说明
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： ○ 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** ○ 目录通配符只支持星号（*）和半角问号（?）。 ○ 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 ○ 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**NGINX配置模式**，可修改为其它模式。
NGINX日志配置	Nginx配置文件中的日志配置部分，以**log_format**开头。例如： `log_format main '$remote_addr - $remote_user [$time_local] "$request" ' '$request_time $request_length ' '$status $body_bytes_sent "$http_referer" ' '"$http_user_agent"';` 更多信息，请参见附录：日志格式和样例。
正则表达式	日志服务根据**NGINX日志配置**中的内容自动生成正则表达式。
日志样例	请根据实际场景，输入Nginx日志样例。例如： `192.168.1.2 - - [10/Jul/2020:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1.11.4 Red Hat modified"` 日志服务通过日志样例验证您的**NGINX日志配置**是否匹配自动生成的正则表达式。输入日志样例后，单击**验证**。如果校验成功，自动将日志样例提取为**NGINX键名称**参数中NGINX键对应的值。
NGINX键名称	根据NGINX日志配置和日志样例自动生成NGINX键名称和对应的值。
丢弃解析失败日志	○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击下一步即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

说明 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

附录：日志格式和样例

Nginx访问日志相关指令主要有两条：log_format和access_log，通常在配置文件*/etc/nginx/nginx.conf*中配置。log_format用来定义日志格式；access_log用来指定日志文件的存放路径。

- 日志格式

  log_format和access_log的默认值如下所示。

```
log_format main  '$remote_addr - $remote_user [$time_local] "$request" '
                 '$request_time $request_length '
                 '$status $body_bytes_sent "$http_referer" '
                 '"$http_user_agent"';
access_log /var/logs/nginx/access.log main
```

  日志字段说明如下所示。

字段名称	说明
remote_addr	客户端IP地址。
remote_user	客户端用户名。
time_local	服务器时间，前后必须加上中括号（[]）。
request	请求的URI和HTTP协议。
request_time	整个请求的总时间，单位为秒。
request_length	请求的长度，包括请求行、请求头和请求正文。
status	请求状态。
body_bytes_sent	发送给客户端的字节数，不包括响应头的大小。
http_referer	URL跳转来源。
http_user_agent	客户端浏览器等信息。

- 日志样例

```
192.168.1.2 - - [10/Jul/2020:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1.11.4 Red Hat modified"
```

3.5.5. 使用分隔符模式采集日志

日志服务提供Logtail分隔符模式快速采集日志。采集到日志后，您可以进行多维度分析、加工、投递等操作。本文介绍如何通过日志服务控制台创建分隔符模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**分隔符-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

b. 安装完成后，单击**确认安装完毕**。

c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。

5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

> **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 创建Logtail配置，单击**下一步**。

参数	描述
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器的创建和销毁，并根据Tag采集指定容器的日志。关于容器文本日志采集，请参见通过DaemonSet-控制台方式采集容器文本日志。

参数	描述
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： ○ 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** ○ 目录通配符只支持星号（*）和半角问号（?）。 ○ 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 ○ 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
模式	默认为**分隔符模式**，可修改为其它模式。
日志样例	请务必使用实际场景的日志。例如： `127.0.0.1|#|-|#|13/Apr/2020:09:44:41 +0800|#|GET /1 HTTP/1.1|#|0.000|#|74|#|404|#|3650|#|-|#|curl/7.29.0` **说明** 分隔符模式只适用于采集单行日志。如果您要采集多行日志，建议您使用极简多行模式或完整正则模式。
分隔符	请根据您的日志格式选择正确的分隔符，例如竖线（|）。更多信息，请参见附录：分隔符简介及日志样例。 **说明** 指定分隔符为**不可见字符**时，您需要查找不可见字符在ASCII码中对应的十六进制数，输入的格式为 `0x`*不可见字符在ASCII码中对应的十六进制数* 。例如ASCII码中排行为1的不可见字符填写为**0x01**。
引用符	当日志字段内容中包含分隔符时，需要指定引用符进行包裹，被引用符包裹的内容会被日志服务解析为一个完整字段。请根据您的日志格式选择正确的引用符。 **说明** 指定引用符为**不可见字符**时，您需要查找不可见字符在ASCII码中对应的十六进制数，输入的格式为 `0x`*不可见字符在ASCII码中对应的十六进制数* 。例如ASCII码中排行为1的不可见字符填写为**0x01**。
日志抽取内容	日志服务会根据您输入的日志样例及选择的分隔符提取日志内容，并将其定义为Value，您需要分别为Value指定对应的Key。

参数	描述
是否接受部分字段	如果日志中分割出的字段数少于配置的Key数量，是否上传已解析的字段。开启表示上传，关闭表示丢弃本条日志。 例如日志为11|22|33|44|55，分隔符为竖线（|），日志内容将被解析为11、22、33、44和55，为其分别设置Key为A、B、C、D和E。 ○ 打开**是否接受部分字段**开关，则采集日志11|22|33|55时，55会作为Key D的Value被上传到日志服务。 ○ 关闭**是否接受部分字段**开关，则采集日志11|22|33|55时，该条日志会因字段与Key不匹配而被丢弃。
使用系统时间	配置日志时间，具体说明如下： ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要配置**指定时间字段Key名称**为日志中的时间字段，并根据时间字段的值配置**时间转换格式**。时间格式详情请参见时间格式。 例如日志中的时间信息为 **"time": "05/May/2016:13:30:28"**，则您可以配置**指定时间字段Key名称**为**time**，**时间转换格式**为**%d/%b/%Y:%H:%M:%S**。 **注意** ○ 当使用DaemonSet方式采集容器日志时，由于Logtail容器的时区为UTC，如果业务容器的时区为非UTC（例如设置了**容器与节点使用相同时区**），则必须在本Logtail配置的**高级选项**中，设置**时区属性**为**自定义时区**，并选择时区为业务容器的时区，否则日志时间将错误偏移。 ○ 默认情况下，日志服务中的日志时间戳精确到秒。如果原始日志中的时间字段具备更高的时间精度（毫秒、微秒或纳秒），并希望在日志服务中保留该时间精度，可在Logtail采集配置的扩展配置中添加enable_precise_timestamp参数完成设置。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。

参数	描述
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

附录：分隔符简介及日志样例

分隔符日志以换行符为边界，每一行都是一条日志。分隔符日志使用分隔符将一条日志分割成多个字段，支持单字符模式和多字符模式。如果字段内部包含分隔符，可以使用引用符对字段进行包裹。

- 单字符模式

  单字符模式的日志示例如下所示。

```
05/May/2016:13:30:28,10.10.*.*,"POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=****************&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1",200,18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10.*.*,"POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=****************&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1",401,23472,aliyun-sdk-java
```

  在单字符模式中，您需要指定分隔符，也可以同时指定引用符。

  ○ 分隔符：通过单字符的分隔符分割日志，例如：制表符（\t）、竖线（|）、空格、半角逗号（,）、半角分号（;）和不可见字符等单字符。分隔符不支持为双引号（"）。

  双引号（"）可以作为引用符，在字段边界出现，也可以作为字段内容出现。如果双引号（"）作为字段内容出现，需要进行转义，即在日志中处理为""。日志服务解析字段时会自动还原，将""还原为"。例如：分隔符为半角逗号（,），引用符为双引号（"），且日志字段内部包含双引号（"）和半角逗号（,），需要将包含半角逗号（,）的日志字段用引用符包裹，同时将日志字段中的双引号（"）转义为""。处理后的日志格式为：1999,Chevy,"Venture ""Extended Edition, Very Large""","",5000.00，该日志可以被解析为5个字段：1999、Chevy、Venture "Extended Edition, Very Large"、空字段和5000.00。

  ○ 引用符：日志字段内容中包含分隔符时，需要指定引用符进行包裹，被引用符包裹的内容会被日志服务解析为一个完整字段。

  引用符可以设置为制表符（\t）、竖线（|）、空格、半角逗号（,）、半角分号（;）和不可见字符等单字符。

  例如：分隔符为半角逗号（,），引用符为双引号（"），日志为1997,Ford,E350,"ac, abs, moon",3000.00，该日志可以被解析为5个字段：1997、Ford、E350、ac, abs, moon、3000.00。

- 多字符模式

  多字符模式的日志示例如下所示。

```
05/May/2016:13:30:28&&10.200.**.**&&POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=****************&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1&&200&&18204&&aliyun-sdk-java
05/May/2016:13:31:23&&10.200.**.**&&POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=****************&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=****************************** HTTP/1.1&&401&&23472&&aliyun-sdk-java
```

  多字符模式中，分隔符包括2~3个字符（例如：‖、&&&、^_^）。日志解析根据分隔符进行匹配，您无需使用引用符对日志字段进行包裹。

> **说明** 确保日志字段内容中不会出现分隔符的完整匹配，否则会导致字段误分割。

  例如：分隔符为&&，日志为1997&&Ford&&E350&&ac&abs&moon&&3000.00会被解析为5个字段：1997、Ford、E350、ac&abs&moon、3000.00。

3.5.6. 使用JSON模式采集日志

日志服务提供Logtail JSON模式快速采集JSON日志。采集到日志后，您可以进行多维度分析、加工、投递等操作。本文介绍如何通过日志服务控制台创建JSON模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。

背景信息

JSON日志建构于两种结构，包括Object类型（键值对的集合）和Array类型（值的有序列表）。

Logtail JSON模式支持解析Object类型的JSON日志，自动提取Object首层的键作为字段名称，Object首层的值作为字段值。但不支持解析Array类型的JSON日志。您可以使用完整正则模式或者极简模式采集Array类型的JSON日志。具体操作，请参见使用极简模式采集日志或使用完整正则模式采集日志。

JSON日志示例如下所示：

```
{"url": "POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "18204"}, "time": "05/Jan/2020:13:30:28"}
{"url": "POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.210", "user-agent": "aliyun-sdk-java", "request": {"status": "200", "latency": "10204"}, "time": "05/Jan/2020:13:30:29"}
```

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**JSON-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

> **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 创建Logtail配置，单击**下一步**。

参数	描述
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** ◦ 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 ◦ 目录通配符只支持星号（*）和半角问号（?）。
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** ◦ 目录通配符只支持星号（*）和半角问号（?）。 ◦ 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 ◦ 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集，请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**JSON模式**，可修改为其它模式。

参数	描述
使用系统时间	配置日志时间，具体说明如下： ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要配置**指定时间字段Key名称**为日志中的时间字段，并根据时间字段的值配置**时间转换格式**。时间格式详情请参见时间格式。 例如日志中的时间信息为 **"time": "05/May/2016:13:30:28"**，则您可以配置**指定时间字段Key名称**为**time**，**时间转换格式**为**%d/%b/%Y:%H:%M:%S**。 注意 ○ 当使用DaemonSet方式采集容器日志时，由于Logtail容器的时区为UTC，如果业务容器的时区为非UTC（例如设置了**容器与节点使用相同时区**），则必须在本Logtail配置的**高级选项**中，设置**时区属性**为**自定义时区**，并选择时区为业务容器的时区，否则日志时间将错误偏移。 ○ 默认情况下，日志服务中的日志时间戳精确到秒。如果原始日志中的时间字段具备更高的时间精度（毫秒、微秒或纳秒），并希望在日志服务中保留该时间精度，可在Logtail采集配置的扩展配置中添加enable_precise_timestamp参数完成设置。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 说明 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。

参数	描述
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

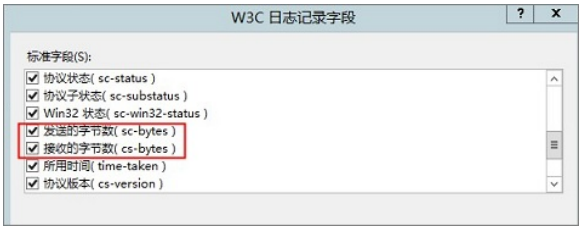
3.5.7. 使用IIS模式采集日志

IIS日志是服务器中的重要日志，日志服务支持通过IIS模式快速采集IIS日志并进行多维度分析。本文介绍如何通过日志服务控制台创建IIS模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。
- 已在服务器上选择合适的日志格式（IIS、NCSA和W3C）生成日志。

  日志服务推荐使用W3C格式记录IIS日志。当您选择W3C格式时，需要先配置W3C日志记录字段，即选中**发送的字节数（sc-bytes）**和**接收的字节数（cs-bytes）**，其他字段保持默认配置。



操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**IIS-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。

4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**Logtail配置**页签中，创建Logtail配置。

配置项	详情
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。

配置项	详情
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： ○ 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 说明 ○ 目录通配符只支持星号（*）和半角问号（?）。 ○ 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 ○ 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**IIS配置模式**，可修改为其它模式，其他模式的配置请参见概述。
日志格式	选择您的IIS服务器日志采用的日志格式，具体说明如下： ○ IIS：Microsoft IIS日志文件格式。 ○ NCSA：NCSA公用日志文件格式。 ○ W3C：W3C扩展日志文件格式。
IIS配置字段	配置IIS配置字段，具体说明如下： ○ 日志格式为IIS或NCSA时，日志服务已默认设置了IIS配置字段。 ○ 日志格式为W3C日志时，设置为IIS配置文件中logFile logExtFileFlags参数中的内容，例如： `logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerName, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie, Referer, ProtocolVersion, Host, HttpSubStatus"` ■ IIS5配置文件默认路径：C:\WINNT\system32\inetsrv\MetaBase.bin ■ IIS6配置文件默认路径：C:\WINDOWS\system32\inetsrv\MetaBase.xml ■ IIS7配置文件默认路径：C:\Windows\System32\inetsrv\config\applicationHost.config
IIS键名称	日志服务根据**IIS配置字段**中的配置，自动提取IIS键。
丢弃解析失败日志	○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

> **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

附录：日志样例及字段说明

IIS日志样例如下所示：

```
#Software: Microsoft Internet Information Services 7.5
#Version: 1.0
#Date: 2020-09-08 09:30:26
#Fields: date time s-sitename s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status sc-bytes cs-bytes time-taken
2009-11-26 06:14:21 W3SVC692644773 125.67.67.* GET /index.html - 80 - 10.10.10.10 Baiduspider+(+http://www.example.com)200 0 64 185173 296 0
```

- 字段前缀说明

前缀	说明
s-	服务器操作
c-	客户端操作
cs-	客户端到服务器的操作
sc-	服务器到客户端的操作

- 各个字段说明

字段	说明
date	客户端发送请求的日期。
time	客户端发送请求的时间。
s-sitename	客户端所访问的站点的Internet服务和实例的号码。
s-computername	生成日志的服务器名称。
s-ip	生成日志的服务器的IP地址。
cs-method	请求方法，例如：GET、POST。
cs-uri-stem	URI资源，表示请求访问的地址。
cs-uri-query	URI查询，表示查询HTTP请求中半角问号（?）后的信息。
s-port	服务器端口号。
cs-username	通过验证的域或用户名。 ◦ 如果是通过身份验证的用户，格式为 域\用户名 。 ◦ 如果是匿名用户，显示短划线（-）。
c-ip	访问服务器的客户端真实IP地址。
cs-version	协议版本，例如：HTTP 1.0、HTTP 1.1。
cs(User-Agent)	客户端使用的浏览器。
Cookie	发送或接受的Cookie内容，如果没有Cookie，则显示短划线（-）。
referer	表示用户访问的前一个站点。
cs-host	主机信息。
sc-status	HTTP协议返回状态。
sc-substatus	HTTP子协议的状态。

字段	说明
sc-win32-status	使用Windows术语表示的操作状态。
sc-bytes	服务器发送的字节数。
cs-bytes	服务器接收的字节数。
time-taken	请求所花费的时间，单位：毫秒。

3.5.8. 使用Apache模式采集日志

Apache日志是运维网站的重要信息，日志服务支持通过Apache模式快速采集Apache日志并进行多维度分析。本文介绍如何通过日志服务控制台创建Apache模式的Logtail配置采集日志。

前提条件

- 已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。
- 安装Logtail的服务器需具备访问远端服务器80端口和443端口的能力。
- 已在Apache日志配置文件中指定日志的打印格式、日志文件路径和名称。更多信息，请参见附录：日志格式和样例。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**Apache-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

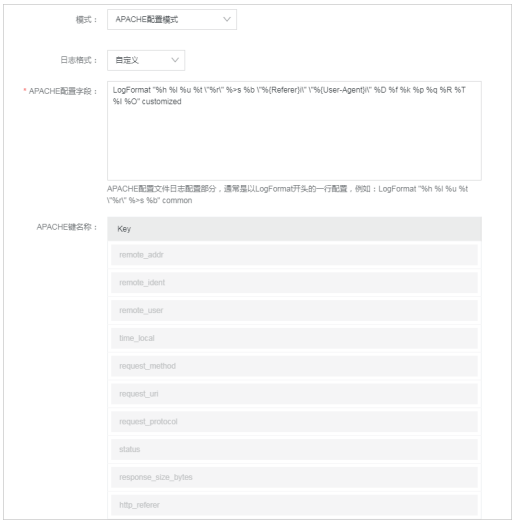
     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

     日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为**FAIL**，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**Logtail配置**页签中，创建Logtail配置。



配置项	详情
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。 日志的目录和文件名支持完整名称和通配符两种模式，文件名规则请参见Wildcard matching。日志文件查找模式为多层目录匹配，即指定目录（包含所有层级的目录）下所有符合条件的文件都会被查找到。例如： • */apsara/nuwa/**/*.log*表示*/apsara/nuwa*目录（包含该目录的递归子目录）中后缀名为*.log*的文件。 • */var/logs/app_*/*.log*表示*/var/logs*目录下所有符合*app_**模式的目录（包含该目录的递归子目录）中包含*.log*的文件。 **说明** • 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。 • 目录通配符只支持星号（*）和半角问号（?）。
设置采集黑名单	打开**设置采集黑名单**开关后，可进行黑名单配置，即可在采集时忽略指定的目录或文件。支持完整匹配和通配符模式匹配目录和文件名。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** • 目录通配符只支持星号（*）和半角问号（?）。 • 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1**目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/***。如果配置为*/home/admin/app1**，则黑名单不会生效。 • 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。

配置项	详情
是否为Docker文件	如果是Docker文件，可打开**是否为Docker文件**开关，直接配置内部路径与容器Tag。Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	默认为**APACHE配置模式**，可修改为其它模式。
日志格式	根据您的Apache日志配置文件中定义的格式进行选择，包括common、combined和自定义。
APACHE配置字段	Apache配置文件中的日志配置部分，通常以LogFormat开头。更多信息，请参见附录：日志格式和样例。 ○ 当配置**日志格式**为**common**或**combined**时，此处会自动填充对应格式的配置字段，请确认是否和Apache配置文件中定义的格式一致。 ○ 当配置**日志格式**为**自定义**时，请根据实际情况填写，例如**LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %l %O" customized**。
APACHE键名称	日志服务会根据**APACHE配置字段**中的内容自动读取Apache键。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。

参数	描述
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

单击**下一步**即表示完成Logtail配置，日志服务开始采集日志。

- Logtail配置生效时间最长需要3分钟，请耐心等待。
- 如果遇到Logtail采集报错，请参见如何查看Logtail采集错误信息。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

附录：日志格式和样例

当您要采集Apache日志时，您需要在Apache日志配置文件中指定日志的打印格式、日志文件路径和名称。例如**CustomLog "/var/log/apache2/access_log" combined**，表示日志打印时使用combined格式，日志文件路径为*/var/log/apache2/access_log*。具体的日志格式和日志样例如下所示：

- Apache日志格式
  - combined格式

    ```
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
    ```

  - common格式

    ```
    LogFormat "%h %l %u %t \"%r\" %>s %b"
    ```

  - 自定义格式

    ```
    LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %D %f %k %p %q %R %T %I %O" customized
    ```

相关字段说明如下所示，更多信息，请参见mod_log_config。

字段格式	字段名称	说明
%a	client_addr	客户端IP地址。
%A	local_addr	本地IP地址。
%b	response_size_bytes	响应字节大小，空值时显示为短划线（-）。
%B	response_bytes	响应字节大小，空值时为0。
%D	request_time_msec	请求时间，单位为微秒。
%f	filename	文件名。
%h	remote_addr	远程的主机名。
%H	request_protocol_supple	请求协议。
%I	bytes_received	服务器接收的字节数，需要启用mod_logio模块。
%k	keep_alive	在此连接上处理的请求数。

字段格式	字段名称	说明
%l	remote_ident	远程主机提供的识别信息。
%m	request_method_supple	请求方法。
%O	bytes_sent	服务器发送的字节数，需要启用mod_logio模块。
%p	remote_port	服务器端口号。
%P	child_process	子进程ID。
%q	request_query	查询字符串，如果不存在则为空字符串。
%r	request	请求内容，包括方法名、地址和HTTP协议。
%R	response_handler	服务端的处理程序类型。
%s	status	响应的HTTP状态，初始状态。
%>s	status	响应的HTTP状态，最终状态。
%t	time_local	服务器时间。
%T	request_time_sec	请求时间，单位为秒。
%u	remote_user	客户端用户名。
%U	request_uri_supple	请求的URI路径，不带查询字符串。
%v	server_name	服务器名称。
%V	server_name_canonical	根据UseCanonicalName指令设定的服务器名称。
“%{User-Agent}i”	http_user_agent	客户端信息。
“%{Rererer}i”	http_referer	来源页。

- Apache日志样例

```
192.168.1.2 - - [02/Feb/2020:17:44:13 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://localhost/x1.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

3.5.9. 导入历史日志文件

Logtail默认只采集增量的日志文件，如果您需要采集历史日志文件，可使用Logtail自带的导入历史文件功能。

前提条件

- 已在服务器上安装0.16.15（Linux系统）或1.0.0.1（Windows系统）及以上版本的Logtail，详情请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。
- 已创建Logtail配置并应用到机器组，详情请参见文本日志概述。

  如果该Logtail配置只用来导入历史文件，可以设置一个不存在的采集路径。

背景信息

Logtail基于监听文件的修改事件进行日志采集，还支持从本地文件中加载事件，以驱动日志采集。采集历史日志文件就是基于本地事件加载实现的功能。

您需要在Logtail的安装目录下执行导入历史文件的操作，该目录在不同操作系统中位于不同位置。

- Linux系统：*/usr/local/ilogtail*
- Windows系统：
  - 32位：*C:\Program Files\Alibaba\Logtail*
  - 64位：*C:\Program Files (x86)\Alibaba\Logtail*

> **说明**
> - 导入本地事件最长延迟为1分钟。
> - 由于加载本地事件属于特殊行为，Logtail会向服务器发送 `LOAD_LOCAL_EVENT_ALARM` 消息。
> - 如果您导入的文件量较大，建议修改Logtail启动参数，建议将CPU调整至2.0及以上，内存调整至512MB及以上，详情请参见设置Logtail启动参数。

操作步骤

1. 获取Logtail配置的唯一标识。

   您可以在Logtail安装目录下的*user_log_config.json*文件中获取Logtail配置的唯一标识。

   此处以Linux系统为例，查看Logtail配置的唯一标识。

   ```
   grep "##" /usr/local/ilogtail/user_log_config.json | awk '{print $1}'
                       "##1.0##log-config-test$multi"
                       "##1.0##log-config-test$ecs-test"
                       "##1.0##log-config-test$metric_system_test"
                       "##1.0##log-config-test$redis-status"
   ```

2. 添加本地事件。

   i. 在Logtail安装目录下，创建*local_event.json*文件。

   ii. 在*local_event.json*文件中添加本地事件，类型为标准JSON，格式如下所示。

   ```
   [
     {
       "config" : "${your_config_unique_id}",
       "dir" : "${your_log_dir}",
       "name" : "${your_log_file_name}"
     },
     {
       ...
     }
     ...
   ]
   ```

   > **说明** 为了防止Logtail加载无效的JSON，建议您先将本地事件配置保存在临时文件中，编辑完成后拷贝到*local_event.json*文件中。

参数	说明
config	填写步骤1中获取的Logtail配置唯一标识，例如：##1.0##log-config-test$ecs-test。
dir	历史日志文件所在目录，例如：*/data/logs*。 **说明** 文件夹不能以 `/` 结尾。
name	历史日志文件名，支持通配符，例如：access.log.2018-08-08、access.log*。

   本文以Linux系统为例，介绍配置示例。

   ```
   $ cat /usr/local/ilogtail/local_event.json
   [
     {
       "config": "##1.0##log-config-test$ecs-test",
       "dir": "/data/log",
       "name": "access.log*"
     },
     {
       "config": "##1.0##log-config-test$tmp-test",
       "dir": "/tmp",
       "name": "access.log.2017-08-09"
     }
   ]
   ```

常见问题

- 检查Logtail是否加载Logtail配置。

  通常情况下，保存*local_event.json*文件后，Logtail会在1分钟内将文件内容加载到内存中，并将*local_event.json*文件中的内容清空。

  您可以通过以下方式检查Logtail已是否加载Logtail采集配置。

  i. *local_event.json*文件中的内容被清空，则说明Logtail已读取到事件信息。

  ii. 检查Logtail安装目录中的*ilogtail.LOG*文件中是否包含process local event参数。如果*local_event.json*文件被清空但未查询到process local event参数，可能是因为*local_event.json*文件内容不合法而被过滤。

- 已加载Logtail采集配置但未采集到数据，是什么原因？
  - Logtail采集配置不合法。
  - *local_event.json*文件配置不合法。
  - 日志文件不在Logtail采集配置已设定的路径下。
  - 该日志文件已被Logtail采集过。

3.5.10. 时间格式

您在使用Logtail采集日志时，需要根据原始日志的时间字符串配置时间格式，Logtail会提取原始日志中的时间字符串并解析为Unix时间戳。本文介绍常见的时间格式及示例。

常见日志时间格式

Logtail支持的常见日志时间格式如下表所示。

> 说明
> - 默认情况下，日志服务中的日志时间戳精确到秒，所以时间格式只需配置到秒，无需配置毫秒、微秒等信息。
>
>   如果原始日志中的时间字段具备更高的时间精度（毫秒、微秒或纳秒），并希望在日志服务中保留该时间精度，可在Logtail采集配置的扩展配置中添加enable_precise_timestamp参数完成设置。更多信息，请参见高级选项、advanced参数说明。
> - 只需为时间字符串中的时间部分配置时间格式，其他内容（例如时区）无需配置。
> - 在Linux服务器中，Logtail支持strftime函数提供的所有时间格式。即能被strftime函数格式化的日志时间字符串都能被Logtail解析并使用。

时间格式	说明	示例
%a	星期的缩写。	Fri
%A	星期的全称。	Friday
%b	月份的缩写。	Jan
%B	月份的全称。	January
%d	每月第几天，十进制，范围为01~31。	07, 31
%h	月份的缩写，等同于%b。	Jan
%H	小时，24小时制。	22
%I	小时，12小时制。	11
%m	月份，十进制，范围为01~12。	08
%M	分钟，十进制，范围为00~59。	59
%n	换行符。	换行符
%p	AM或PM。	AM、PM
%r	12小时制的时间组合，等同于%I:%M:%S %p。	11:59:59 AM
%R	小时和分钟组合，等同于%H:%M。	23:59
%S	秒数，十进制，范围为00~59。	59
%t	Tab符号，制表符。	无

时间格式	说明	示例
%y	年份，十进制，不带世纪，范围为00~99。	04、98
%Y	年份，十进制。	2004、1998
%C	世纪，十进制，范围为00~99。	16
%e	每月第几天，十进制，范围为1~31。 如果是个位数字，前面需要加空格。	7、31
%j	一年中的天数，十进制，范围为001~366。	365
%u	星期几，十进制，范围为1~7，1表示周一。	2
%U	每年的第几周，星期天是一周的开始，范围为00~53。	23
%V	每年的第几周，星期一是一周的开始，范围为01~53。 如果一月份刚开始的一周>=4天，则认为是第1周，否则认为下一个星期是第1周。	24
%w	星期几，十进制，范围为0~6，0代表周日。	5
%W	每年的第几周，星期一是一周的开始，范围为00~53。	23
%c	标准的日期和时间。	Tue Nov 20 14:12:58 2020
%x	标准的日期，不带时间。	Tue Nov 20 2020
%X	标准的时间，不带日期。	11:59:59
%s	Unix时间戳。	1476187251

示例

常见的时间标准、示例及对应的时间表达式如下所示。

示例	时间表达式	时间标准
2017-12-11 15:05:07	%Y-%m-%d %H:%M:%S	自定义
[2017-12-11 15:05:07.012]	[%Y-%m-%d %H:%M:%S]	自定义
02 Jan 06 15:04 MST	%d %b %y %H:%M	RFC822
02 Jan 06 15:04 -0700	%d %b %y %H:%M	RFC822Z
Monday, 02-Jan-06 15:04:05 MST	%A, %d-%b-%y %H:%M:%S	RFC850
Mon, 02 Jan 2006 15:04:05 MST	%A, %d %b %Y %H:%M:%S	RFC1123
2006-01-02T15:04:05Z07:00	%Y-%m-%dT%H:%M:%S	RFC3339
2006-01-02T15:04:05.999999999Z07:00	%Y-%m-%dT%H:%M:%S	RFC3339Nano
1637843406	%s	自定义
1637843406123	%s	自定义（日志服务以秒级精度处理）

3.5.11. 日志主题

日志可以通过日志主题（Topic）来划分，您可以在日志服务控制台上创建Logtail配置时设置日志主题，也可以在使用API或SDK上传数据时设置日志主题。

- 空-不生成Topic

- 机器组Topic属性
- 文件路径正则
- 静态主题生成

空-不生成Topic

设置**Topic生成方式**为**空-不生成topic**，则表示不生成日志主题。

机器组Topic属性

该方式用于区分不同服务器产生的日志。如果您不同服务器上的日志的保存路径或文件名相同，您可以通过日志主题进行区分。

您可以将服务器添加到不同的机器组中，并为机器组设置不同的Topic属性，然后在创建Logtail配置时，将**Topic生成方式**设置为**机器组Topic属性**。Logtail上报数据时会将服务器所在机器组的Topic属性作为日志主题的名称上传至日志服务，在查询日志时需要指定日志主题（即机器组的Topic属性）为查询条件。

文件路径正则

该方式用于区分不同用户或实例产生的日志数据。如果不同的用户或者实例将日志保存在不同目录中，但是下级目录和日志文件名相同，日志服务在采集日志时无法明确区分日志是由哪个用户或实例产生的。

提取文件路径中的单个值

您可以在创建Logtail配置时将**Topic生成方式**设置为**文件路径正则**，并且输入文件路径的正则表达式，并使用捕获组捕获待提取的内容（此处的正则表达式需要完整匹配文件路径，捕获组有且仅有一个）。Logtail上报数据时会将用户名或实例名作为日志主题的名称上传至日志服务，在查询日志时需要指定日志主题（即用户名或实例名）为查询条件。

> **说明** 文件路径的正则表达式中，需要对正斜线（/）进行转义。

例如不同用户将日志记录在不同目录下，但是日志文件名称相同，目录路径如下所示。

```
/logs
  | - /userA/serviceA
    | - service.log
  | - /userB/serviceA
    | - service.log
  | - /userC/serviceA
    | - service.log
```

如果在Logtail采集配置中仅配置文件路径为*/logs*，文件名称为*service.log*，将三个*service.log*文件中的内容采集至同一个Logstore中，则无法区分日志具体由哪个用户产生。您可以通过如下方式提取文件路径中的值，生成不同的日志主题。

- 正则表达式

  ```
  \/(.*)\/serviceA\/.*
  ```

- 提取结果

  ```
  __topic__: userA
  __topic__: userB
  __topic__: userC
  ```

提取文件路径中的多个值

如果单个日志主题不足以区分日志的来源，则您可以提取文件路径中的多个值作为区分日志的信息，即您可以配置多个正则捕获组提取多个值作为日志的Tag。其中捕获组包括命名捕获组（?P<name>）或非命名捕获组。如果全是命名捕获组，则生成的tag字段为__tag__:{name}；如果全是非命名捕获组，则生成的tag字段为__tag__:__topic_{i}__，其中{i}为捕获组的序号。

> **说明** 当正则表达式中存在多个捕获组时，不会生成__topic__字段。

例如文件路径为*/logs/userA/serviceA/service.log*，您可以通过如下方式提取文件路径中的多个值。

- 示例1：使用非命名捕获组进行正则提取。
  - 正则表达式

    ```
    \/logs\/(.*?)\/(.*?)\/service.log
    ```

  - 提取结果

    ```
    __tag__:__topic_1__: userA
    __tag__:__topic_2__: serviceA
    ```

- 示例2：使用命名捕获组进行正则提取。

- 正则表达式

```
\/logs\/(?P<user>.*?)\/(?P<service>.*?)\/service.log
```

- 提取结果

```
__tag__:user: userA
__tag__:service: serviceA
```

静态主题生成

将**Topic生成方式**设置为**文件路径正则**，在**自定义正则**中输入 `customized:// + 自定义主题名` ，表示使用自定义的静态日志主题。

> **说明** Logtail 0.16.21（Linux系统）及以上版本支持该设置。

3.6. 采集容器日志

3.6.1. 概述

日志服务支持通过DaemonSet方式和Sidecar方式采集Kubernetes集群的容器日志，本文简要介绍两种方式的采集流程及区别。

采集方式介绍

DaemonSet方式运维简单、资源占用少、支持采集容器的标准输出和文本文件、配置方式灵活。但DaemonSet方式下，Logtail采集该节点内所有容器的日志，存在一定的性能瓶颈，且各个容器之间的隔离性较弱。Sidecar方式为每个需要采集日志的容器创建一个Sidecar容器，多租户隔离性好、性能好。

采集配置介绍

日志服务支持通过CRD方式和控制台方式创建采集配置，两者之间的区别如下所示。

对比项	CRD方式	控制台方式
操作复杂度	低	一般
功能项	支持除控制台方式外的高级配置	一般
上手难度	一般	低
网络连接	连接Kubernetes集群	连接互联网
与容器应用集成	支持	不支持
鉴权方式	Kubernetes鉴权	云账号鉴权

采集流程

DaemonSet方式采集流程如下所示。

1. 安装Logtail组件。
2. 创建采集配置。

   日志服务支持采集配置通过CRD和控制台两种方式创建采集配置，采集Kubernetes集群中的容器日志。

   - 通过DaemonSet-CRD方式采集容器日志。
   - 通过DaemonSet-控制台方式采集容器文本日志。
   - 通过DaemonSet-控制台方式采集容器标准输出

   > **说明** CRD配置可自动创建Project、Logstore、索引、机器组、Logtail配置等资源，且和Kubernetes集成性较好，推荐使用该方式。控制台配置操作更加简单，适合初次接触容器日志采集的用户。

Sidecar方式采集流程如下所示。

1. 安装Logtail组件。
2. 安装Sidecar及创建采集配置。

   日志服务支持通过CRD和控制台两种方式创建采集配置，采集Kubernetes集群中的容器日志。

   - 通过Sidecar-CRD方式采集容器文本日志。
   - 通过Sidecar-控制台方式采集容器文本日志

3.6.2. 安装Logtail组件

本文介绍如何在Kubernetes集群上安装Logtail组件。

背景信息

采集Kubernetes集群中的容器日志时，需先安装Logtail组件。

在安装Logtail组件过程中，系统自动完成以下操作：

1. 创建alibaba-log-configuration ConfigMap，该ConfigMap中包含日志服务配置信息，例如Project等。
2. （可选）创建AliyunLogConfig CRD资源。
3. （可选）部署alibaba-log-controller Deployment，用于监听AliyunLogConfig CRD资源的变更、创建Logtail采集配置。
4. 部署logtail-ds Daemonset，用于采集节点的日志。

阿里云Kubernetes集群

您可以为已有的Kubernetes集群安装Logtail组件，也可以在创建Kubernetes集群时选中**使用日志服务**，安装Logtail组件。

为已有的Kubernetes集群安装Logtail组件

> **注意** 此操作仅适用于专有版Kubernetes和托管版Kubernetes。

1. 登录容器服务管理控制台。
2. 在左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击目标集群。
4. 在左侧导航栏中，选择**运维管理 > 组件管理**。
5. 在**日志与监控**页签中，找到**logtail-ds**，然后单击**安装**。

   安装完成后，在该Project下自动创建名为`k8s-group-${your_k8s_cluster_id}`的机器组和名为`config-operation-log`的Logstore。

   > **注意** 请勿删除名为`config-operation-log`的Logstore。

创建Kubernetes集群时安装Logtail组件

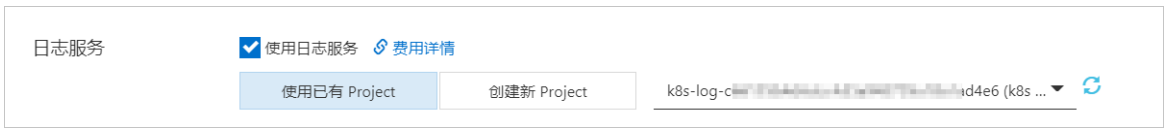
1. 登录容器服务管理控制台。
2. 在左侧导航栏中，单击**集群**。
3. 在**集群列表**页面中，单击**创建集群**。
4. 在**组件配置**配置项页中，选中**使用日志服务**。

   > **说明** 本操作仅介绍开启日志服务的关键步骤。关于创建集群的具体操作，请参见创建Kubernetes托管版集群。

   当选中**使用日志服务**后，会出现创建项目（Project）的提示。关于日志服务管理日志的组织结构，请参见项目（Project）。有以下两种创建Project方式。

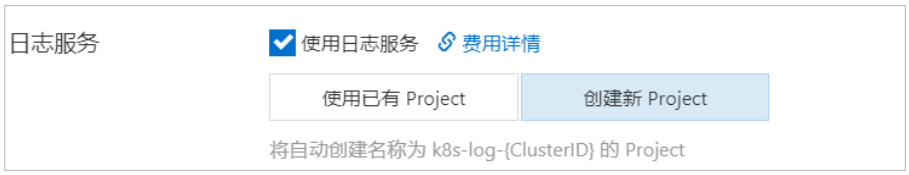
   - **使用已有Project**

     您可以选择一个已有的Project来管理采集到的容器日志。

   - **创建新Project**

     日志服务自动创建一个名为`k8s-log-{ClusterID}`的Project来管理采集到的容器日志。其中`ClusterID`为您新建的Kubernetes集群的唯一标识。

   安装完成后，在该Project下自动创建名为`k8s-group-${your_k8s_cluster_id}`的机器组和名为`config-operation-log`的Logstore。

> 注意 请勿删除名为 `config-operation-log` 的Logstore。

自建Kubernetes集群

1. 登录日志服务控制台。
2. 创建一个以 `k8s-log-custom-` 开头的Project。

   例如k8s-log-custom-sd89ehdq。具体操作，请参见创建Project。
3. 登录您的Kubernetes集群。
4. 执行如下命令安装Logtail及其他依赖组件。

   > 注意
   > - 请确保用于执行脚本的机器中，已安装kubectl命令。
   > - 目前，alibaba-log-controller组件只支持Kubernetes 1.6及以上版本。
   > - 如果您不需要CRD功能，可以删除*alibaba-cloud-log/templates/alicloud-log-config.yam*文件，然后重新执行下述命令。如果提示 `./alicloud-log-k8s-custom-install.sh: line 111: /root/alibaba-cloud-log/templates/alicloud-log-crd.yaml: No such file or directory` 错误，可忽略。

   i. 下载安装脚本。

   ```
   wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/kubernetes/alicloud-log-k8s-custom-install.sh
   ```

   ii. 修改权限。

   ```
   chmod 744 ./alicloud-log-k8s-custom-install.sh
   ```

   iii. 安装Logtail及其他依赖组件。

   ```
   sh ./alicloud-log-k8s-custom-install.sh your-project-suffix region-id aliuid access-key-id access-key-secret
   ```

   命令中各参数说明如下所示，请根据实际情况替换。

参数	说明
your-project-suffix	配置为您在步骤中创建的Project的名称中的自定义部分。例如Project名称为 `k8s-log-custom-sd89ehdq` ，则此处填写 `sd89ehdq` 。
region-id	您的Project所在的地域ID。例如华东1（杭州）的地域ID为 `cn-hangzhou` 。更多信息，请参见开服地域。
aliuid	您的阿里云账号ID。如何获取，请参见步骤一：获取日志服务所在的阿里云账号ID。
access-key-id	您的阿里云账号的AccessKey ID。推荐使用RAM用户的AccessKey，并授予RAM用户AliyunLogFullAccess权限。相关操作，请参见创建RAM用户及授权。
access-key-secret	您的阿里云账号的AccessKey Secret。推荐使用RAM用户的AccessKey并授予RAM用户AliyunLogFullAccess权限。相关操作，请参见创建RAM用户及授权。

   安装完成后，在该Project下自动创建名为 `k8s-group-${your_k8s_cluster_id}` 的机器组和名为 `config-operation-log` 的Logstore。

   > 注意
   > - 请勿删除名为 `config-operation-log` 的Logstore。
   > - 在自建Kubernetes集群上安装时，默认为Logtail授予 `privileged` 权限，主要为避免删除其他Pod时可能出现的 `container text file busy` 错误。更多信息，请参见Bug 1468249、Bug 1441737和 issue 34538。

常见问题

- 如何查看镜像版本？

  您可以通过镜像仓库进行查看，地址为https://cr.console.aliyun.com/repository/cn-shanghai/log-service/logtail/images。
- 多个Kubernetes集群如何共用一个日志服务Project？
  - 阿里云Kubernetes集群

    如果您希望将多个Kubernetes集群中的容器日志采集到同一个日志服务Project中，您可以在创建Kubernetes集群时选择相同的Project。

- 自建Kubernetes集群

  如果您希望将多个Kubernetes集群中的容器日志采集到同一个日志服务Project中，您可以在安装其他集群日志服务组件时，将安装参数中的{your-project-suffix}与您第一次安装集群日志服务组件时设置为一样。

  > **说明** 此方式不支持跨地域的Kubernetes多集群共享。

- 如何查看Logtail日志？

  Logtail日志存储在Logtail容器中的*/usr/local/ilogtail/*目录中，文件名为*ilogtail.LOG*和*logtail_plugin.LOG*。

  Logtail容器中的标准输出并不具备参考意义，请忽略以下标准输出内容。

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- 如何查看Kubernetes集群中日志服务相关组件的状态？

  执行如下命令进行查看。

```
kubectl get deploy alibaba-log-controller -n kube-system
kubectl get ds logtail-ds -n kube-system
```

- alibaba-log-controller启动失败，该怎么处理？

  请确认您是否按照以下方式进行安装。

  - 在Kubernetes集群的Master节点中执行安装命令。
  - 安装命令参数中输入的是您的集群ID。

  如果由于以上问题安装失败，请使用`kubectl delete -f deploy`命令删除已生成的安装模板并重新执行安装命令。

- 如何查看Kubernetes集群中Logtail DaemonSet状态？

  执行`kubectl get ds -n kube-system`命令查看Logtail DaemonSet状态。

  > **说明** Logtail容器所在的命名空间，默认为kube-system。

- 如何查看Logtail的版本号、IP地址、启动时间以及状态等信息？
  - 查看Logtail状态等信息。

    ```
    kubectl get po -n kube-system | grep logtail
    ```

    返回结果如下：

    ```
    NAME            READY     STATUS    RESTARTS   AGE
    logtail-ds-gb92k   1/1       Running   0          2h
    logtail-ds-wm7lw   1/1       Running   0          4d
    ```

- 查看Logtail的版本号、IP地址等信息。

  ```
  kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/app_info.json
  ```

  返回结果如下：

  ```
  {
     "UUID" : "",
     "hostname" : "logtail-ds-gb92k",
     "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.2_1517810940",
     "ip" : "192.0.2.0",
     "logtail_version" : "0.16.2",
     "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
     "update_time" : "2021-02-05 06:09:01"
  }
  ```

- 如何查看Logtail的运行日志？

  Logtail运行日志保存在*/usr/local/ilogtail/*目录下，文件名为*ilogtail.LOG*，轮转文件会压缩存储为*ilogtail.LOG.x.gz*。

  示例如下：

  ```
  [root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-ds-gb92k -n kube-system tail /usr/local/ilogtail/ilogtail.LOG
  [2018-02-05 06:09:02.168693] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume:start
  [2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume:succe
  ss
  [2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:369] start add existed check p
  oint events, size:0
  [2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.cpp:511] add existed check point e
  vents, size:0 cache size:0 event size:0 success count:0
  ```

- 如何重启某个Pod中的Logtail?

  i. 停止Logtail。

  其中 `logtail-ds-gb92k -n` 表示容器名， `kube-system` 表示命名空间，请根据实际情况替换。

  ```
  kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild stop
  ```

  返回如下结果表示停止成功。

  ```
  kill process Name: ilogtail pid: 7
  kill process Name: ilogtail pid: 9
  stop success
  ```

  ii. 启动Logtail。

  其中 `logtail-ds-gb92k -n` 表示容器名， `kube-system` 表示命名空间，请根据实际情况替换。

  ```
  kubectl exec logtail-ds-gb92k -n kube-system /etc/init.d/ilogtaild start
  ```

  返回如下结果表示启动成功。

  ```
  ilogtail is running
  ```

- 如何采集控制平面组件日志？
  - 如果是阿里云托管版ACK集群，您可以在容器服务管理控制台中开启控制平面组件日志的采集。具体操作，请参见收集托管集群控制平面组件日志。
  - 如果是自建Kubernetes集群或者阿里云专有版ACK集群，您可以参见通过DaemonSet-控制台方式采集容器标准输出完成采集。

后续步骤

创建Logtail采集配置，采集容器日志。

- DaemonSet方式
  - 如果您需要通过CRD方式采集日志，请参见通过DaemonSet-CRD方式采集容器日志。
  - 如果您需要通过控制台方式采集容器标准输出，请参见通过DaemonSet-控制台方式采集容器标准输出。
  - 如果您需要通过控制台方式采集容器文本日志，请参见通过日志服务采集Kubernetes容器日志。
- Sidecar方式
  - 如果您需要通过CRD方式采集日志，请参见通过Sidecar-CRD方式采集容器文本日志。
  - 如果您需要通过控制台方式采集日志，请参见通过Sidecar-控制台方式采集容器文本日志。

3.6.3. 通过DaemonSet-控制台方式采集容器文本日志

本文介绍如何在控制台上创建Logtail配置，并以DaemonSet方式采集容器文件日志。

前提条件

已安装Logtail组件。具体操作，请参见安装Logtail组件。

功能特点

Logtail支持将容器产生的文本日志和容器相关的元数据信息一起上传到日志服务。Logtail具备以下功能特点。

- 只需配置容器内的日志文件路径，无需关心该路径到宿主机的映射。
- 支持通过容器Label白名单指定待采集的容器。
- 支持通过容器Label黑名单排除不要采集的容器。
- 支持通过环境变量白名单指定待采集的容器。
- 支持通过环境变量黑名单排除不要采集的容器。
- 支持采集多行日志（例如Java Stack日志）。
- 支持上报容器日志时自动关联Meta信息（例如容器名、镜像、Pod、Namespace、环境变量等）。
- 当容器运行于Kubernetes时，Logtail还具有以下功能。
  - 支持通过Kubernetes Namespace名称、Pod名称、容器名称指定待采集的容器。
  - 支持通过Kubernetes Label白名单指定待采集的容器。
  - 支持通过Kubernetes Label黑名单排除不要采集的容器。
  - 支持上报容器日志时自动关联Kubernetes Label信息。

限制说明

- 采集停止策略：当容器被停止后，Logtail监听到容器 `die` 的事件后会停止该容器日志的采集。如果此时采集出现延迟，则可能丢失停止前的部分日志。
- Docker存储驱动限制：目前只支持overlay、overlay2，其他存储驱动需将日志所在目录通过数据卷挂载为临时目录。

  如果日志目录是以PVC方式挂载到NAS，则不支持使用Daemonset方式采集日志，建议使用Sidecar方式采集。
- 不支持采集软链接：目前Logtail无法访问业务容器的软链接，请按真实路径配置采集目录。
- 如果业务容器的数据目录是通过数据卷（Volume）挂载的，则不支持采集它的父目录，需设置采集目录为完整的数据目录。

  例如*/var/log/service*目录是数据卷挂载的路径，则设置采集目录为*/var/log*将采集不到该目录下的日志，需设置采集目录为*/var/log/service*。
- Kubernetes默认将宿主机根目录挂载到Logtail容器的 `/logtail_host` 目录。如果您要采集宿主机文本日志，则配置日志文件路径时，需加上 `/logtail_host` 前缀。

  例如需要采集宿主机上 `/home/logs/app_log/` 目录下的日志，则设置日志路径为 `/logtail_host/home/logs/app_log/` 。
- Logtail支持Docker和Containerd两种容器引擎的数据采集，访问路径说明如下：
  - Docker：Logtail通过*/run/docker.sock*访问Docker，请确保该路径存在且具备访问权限。
  - Containerd：Logtail通过*/run/containerd/containerd.sock*访问Containerd，请确保该路径存在且具备访问权限。

创建Logtail采集配置

1. 登录日志服务控制台。
2. 在**接入数据**区域，单击**Kubernetes-文件**。
3. 选择目标Project和Logstore，单击**下一步**。

   选择您在安装Logtail组件时所使用的Project。Logstore为您自定义创建的Logstore。
4. 单击**使用现有机器组**。

   安装Logtail组件后，日志服务自动创建名为 `k8s-group-${your_k8s_cluster_id}` 的机器组，您可以直接使用该机器组。
5. 选中目标机器组（ `k8s-group-${your_k8s_cluster_id}` ），将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果机器组心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 设置Logtail采集配置，单击**下一步**。
   i. 设置基本信息（例如配置名称、日志路径、模式等）。更多信息，请参见采集文本日志。
   ii. 打开**是否为Docker文件**的开关。
   iii. （可选）设置容器过滤条件。

- Logtail 1.0.34以下版本，只支持通过环境变量、容器Label进行容器过滤。详细说明，如下表所示。

  Kubernetes中的命名空间名和容器名会映射到容器Label中，分别为 `io.kubernetes.pod.namespace` 和 `io.kubernetes.container.name` 。推荐使用这两个容器Label进行容器过滤。如果这两个容器Label未满足需求，请使用环境变量的黑白名单进行容器过滤。例如某Pod所属的命名空间为backend-prod，容器名为worker-server，如果您要采集包含该容器的日志，可以设置容器Label白名单为 `io.kubernetes.pod.namespace : backend-prod` 或 `io.kubernetes.container.name : worker-server` 。

  > 注意
  > - 容器Label为Docker inspect中的Label，不是Kubernetes中的Label。如何获取，请参见获取容器Label。
  > - 环境变量为容器启动中配置的环境变量信息。如何获取，请参见获取容器环境变量。
  > - 请勿设置相同的LabelKey，如果重名只生效一个。

参数名称	参数说明
Label白名单	用于指定待采集的容器。如果您要设置容器Label白名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则容器Label中包含LabelKey的容器都匹配。 ■ 如果LabelValue不为空，则容器Label中包含LabelKey=LabelValue的容器才匹配。 LabelValue默认为字符串匹配，即只有LabelValue和容器Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如：配置LabelKey为*io.kubernetes.container.name*，配置LabelValue为*^(nginx\|cube)$*，表示可匹配名为nginx、cube的容器。 多个白名单之间为或关系，即只要容器Label满足任一白名单即可被匹配。
Label黑名单	用于排除不采集的容器。如果您要设置容器Label黑名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则容器Label中包含LabelKey的容器都将被排除。 ■ 如果LabelValue不为空，则容器Label中包含LabelKey=LabelValue的容器才会被排除。 LabelValue默认为字符串匹配，即只有LabelValue和容器Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如：配置LabelKey为*io.kubernetes.container.name*，配置LabelValue为*^(nginx\|cube)$*，表示可匹配名为nginx、cube的容器。 多个黑名单之间为或关系，即只要容器Label满足任一黑名单对即可被排除。
环境变量白名单	用于指定待采集的容器。如果您要设置环境变量白名单，那么EnvKey必填，EnvValue可选填。 ■ 如果EnvValue为空，则容器环境变量中包含EnvKey的容器都匹配。 ■ 如果EnvValue不为空，则容器环境变量中包含EnvKey=EnvValue的容器才匹配。 EnvValue默认为字符串匹配，即只有EnvValue和环境变量的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配，例如：配置EnvKey为*NGINX_SERVICE_PORT*，配置EnvValue为*^(80\|6379)$*，表示可匹配服务端口为80、6379的容器。 多个白名单之间为或关系，即只要容器的环境变量满足任一键值对即可被匹配。
环境变量黑名单	用于排除不采集的容器。如果您要设置环境变量黑名单，那么EnvKey必填，EnvValue可选填。 ■ 如果EnvValue为空，则容器环境变量中包含EnvKey的容器的日志都将被排除。 ■ 如果EnvValue不为空，则容器环境变量中包含EnvKey=EnvValue的容器才会被排除。 EnvValue默认为字符串匹配，即只有EnvValue和环境变量的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配，例如：配置EnvKey为*NGINX_SERVICE_PORT*，配置EnvValue为*^(80\|6379)$*，表示可匹配服务端口为80、6379的容器。 多个黑名单之间为或关系，即只要容器的环境变量满足任一键值对即可被排除。

- Logtail 1.0.34及以上版本，推荐使用Kubernetes层级的信息（Pod名称、Namespace名称、容器名称、Label）进行容器过滤。

  打开**是否部署于K8s**开关，选择如下资源进行容器过滤。

  > **说明** 由于在Kubernetes管控类资源（例如Deployment ）运行时更改Label，不会重启具体的工作资源Pod，因此Pod无法感知此变更，可能导致匹配规则失效。设置K8s Label黑白名单时，请以Pod中的Kubernetes Label为准。

参数名称	参数说明
K8s Pod名称正则匹配	通过Pod名称指定待采集的容器，支持正则匹配。例如设置为 *^(nginx-log-demo.*)$*，表示匹配以nginx-log-demo开头的Pod下的所有容器。
K8s Namespace正则匹配	通过Namespace名称指定采集的容器，支持正则匹配。例如设置为 *^(default|nginx)$*，表示匹配nginx命名空间、default命名空间下的所有容器。
K8s容器名称正则匹配	通过容器名称指定待采集的容器（Kubernetes容器名称是定义在spec.containers中），支持正则匹配。例如设置为 *^(container-test)$*，表示匹配所有名为container-test的容器。
K8s Label白名单	通过Kubernetes Label白名单指定待采集的容器。如果您要设置Kubernetes Label白名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则Kubernetes Label中包含LabelKey的容器都匹配。 ■ 如果LabelValue不为空，则Kubernetes Label中包含LabelKey=LabelValue的容器才匹配。 LabelValue默认为字符串匹配，即只有LabelValue和Kubernetes Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置**LabelKey**为*app*，设置**LabelValue**为 *^(test1\|test2)$*，表示匹配Kubernetes Label中包含app:test1、app:test2的容器。 多个白名单之间为或关系，即只要Kubernetes Label满足任一白名单即可被匹配。
K8s Label黑名单	通过Kubernetes Label黑名单排除不采集的容器。如果您要设置Kubernetes Label黑名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则Kubernetes Label中包含LabelKey的容器都被排除。 ■ 如果LabelValue不为空，则Kubernetes Label中包含LabelKey=LabelValue的容器才会被排除。 LabelValue默认为字符串匹配，即只有LabelValue和Kubernetes Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置**LabelKey**为*app*，设置**LabelValue**为 *^(test1\|test2)$*，表示匹配Kubernetes Label中包含app:test1、app:test2的容器。 多个黑名单之间为或关系，即只要Kubernetes Label满足任一黑名单对即可被排除。

iv. （可选）设置日志标签。

如果您使用的是Logtail 1.0.34及以上版本，您可以将环境变量和Kubernetes Label添加到日志，作为日志标签。

参数名称	参数说明
环境变量日志标签	设置环境变量日志标签后，日志服务将在日志中新增环境变量相关字段。例如设置**EnvKey**为*VERSION*，设置**EnvValue**为*env_version*，当容器中包含环境变量 `VERSION=v1.0.0` 时，会将该信息添加到日志中，即添加字段__tag__:__env_version__: v1.0.0。
K8s Label日志标签	设置Kubernetes Label日志标签后，日志服务将在日志中新增Kubernetes Label相关字段。例如设置**LabelKey**为*app*，设置**LabelValue**为 `k8s_label_app` ，当Kubernetes中包含Label `app=serviceA` 时，会将该信息添加到日志中，即添加字段__tag__:__k8s_label_app__: serviceA。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

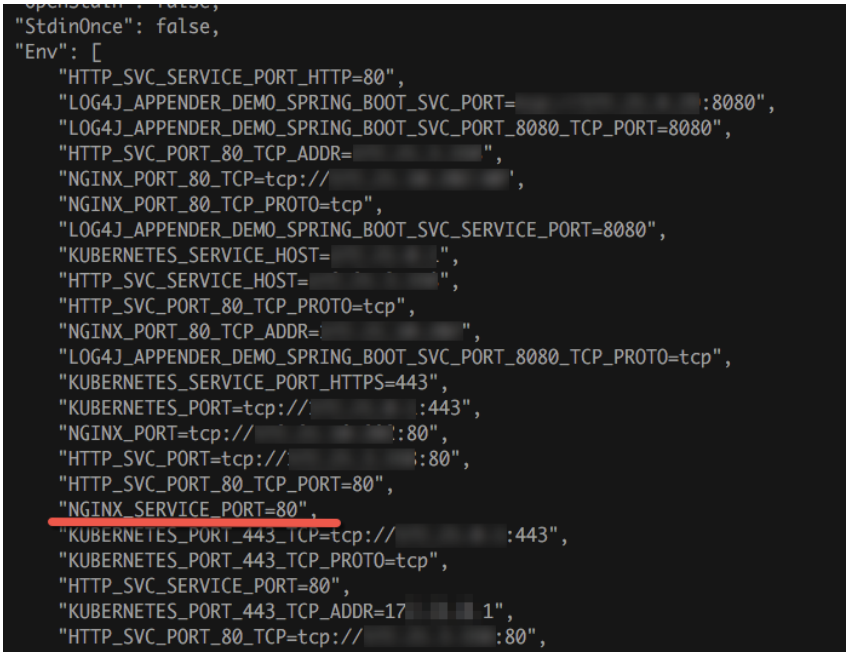
配置示例

示例1：通过环境变量黑白名单过滤容器

采集环境变量为 `NGINX_SERVICE_PORT=80` 且不为 `POD_NAMESPACE=kube-system` 的容器的文本日志，日志文件路径为 `/var/log/nginx/access.log` ，日志解析模式为**极简模式**。

1. 获取环境变量

   您可以登录容器所在的宿主机查看容器的环境变量。具体操作，请参见获取容器环境变量。



2. 创建Logtail采集配置

   Logtail采集配置示例如下图所示。极简模式的相关配置说明请参见使用极简模式采集日志。

示例2：通过容器Label黑白名单过滤容器

采集容器Label为 `io.kubernetes.container.name=nginx` 的容器的文本日志，日志文件路径为 `/var/log/nginx/access.log` ，日志解析方式为**极简模式**。

1. 获取容器Label

   您可以登录容器所在的宿主机查看容器的Label。具体操作，请参见获取容器Label。


```
"OnBuild": null,
"Labels": {
    "annotation.io.kubernetes.container.hash": "53073f5a",
    "annotation.io.kubernetes.container.restartCount": "0",
    "annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
    "annotation.io.kubernetes.container.terminationMessagePolicy": "File",
    "annotation.io.kubernetes.pod.terminationGracePeriod": "30",
    "io.kubernetes.container.logpath": "/var/log/pods/ad00a078-4182                    585/nginx_0.log",
    "io.kubernetes.container.name": "nginx",
    "io.kubernetes.docker.type": "container",
    "io.kubernetes.pod.name": "example-foo-86ccd54874-r4mfh",
    "io.kubernetes.pod.namespace": "default",
    "io.kubernetes.pod.uid": "ad0
    "io.kubernetes.sandbox.id": "                                                    1dfa6da112969",
    "maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"
},
"StopSignal": "SIGTERM"
```


2. 创建Logtail采集配置

   Logtail采集配置示例如下图所示。极简模式的相关配置说明请参见使用极简模式采集日志。

示例3：通过Kubernetes Namespace名称、Pod名称和容器名称过滤容器

采集default命名空间下以nginx-log-demo开头的Pod中的nginx-log-demo-0容器。

1. 获取Kubernetes层级的信息
   - 获取Pod信息

```
~/.kube » kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
nginx-log-demo-0-bxl79    1/1     Running   0          48d
nginx-log-demo-1-qmrqk    1/1     Running   0          48d
nginx-log-demo-2-7khv9    1/1     Running   0          48d
nginx-log-demo-3-j24xc    1/1     Running   0          48d
```

- 获取Namespace等信息

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/psp: ack.privileged
  creationTimestamp: "2022-01-06T18:42:43Z"
  generateName: nginx-log-demo-0-
  labels:
    controller-uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
    job-name: nginx-log-demo-0
  name: nginx-log-demo-0-bxl79
  namespace: default
  ownerReferences:
  - apiVersion: batch/v1
    blockOwnerDeletion: true
    controller: true
    kind: Job
    name: nginx-log-demo-0
    uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
  resourceVersion: "50566856"
  uid: ee10fb7d-d989-47b3-bc2a-e9ffbe767849
spec:
  containers:
  - args:
    - --log-type=nginx
    - --stdout=true
    - --total-count=100000000000
    - --log-file-size=1000000000
    - --log-file-count=2
    - --logs-per-sec=2778
    command:
    - /bin/mock_log
    image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
    imagePullPolicy: Always
    name: nginx-log-demo-0
    resources:
```

2. 创建Logtail采集配置

   Logtail采集配置示例如下图所示。极简模式的相关配置说明请参见使用极简模式采集日志。

示例4：通过Kubernetes Label过滤容器

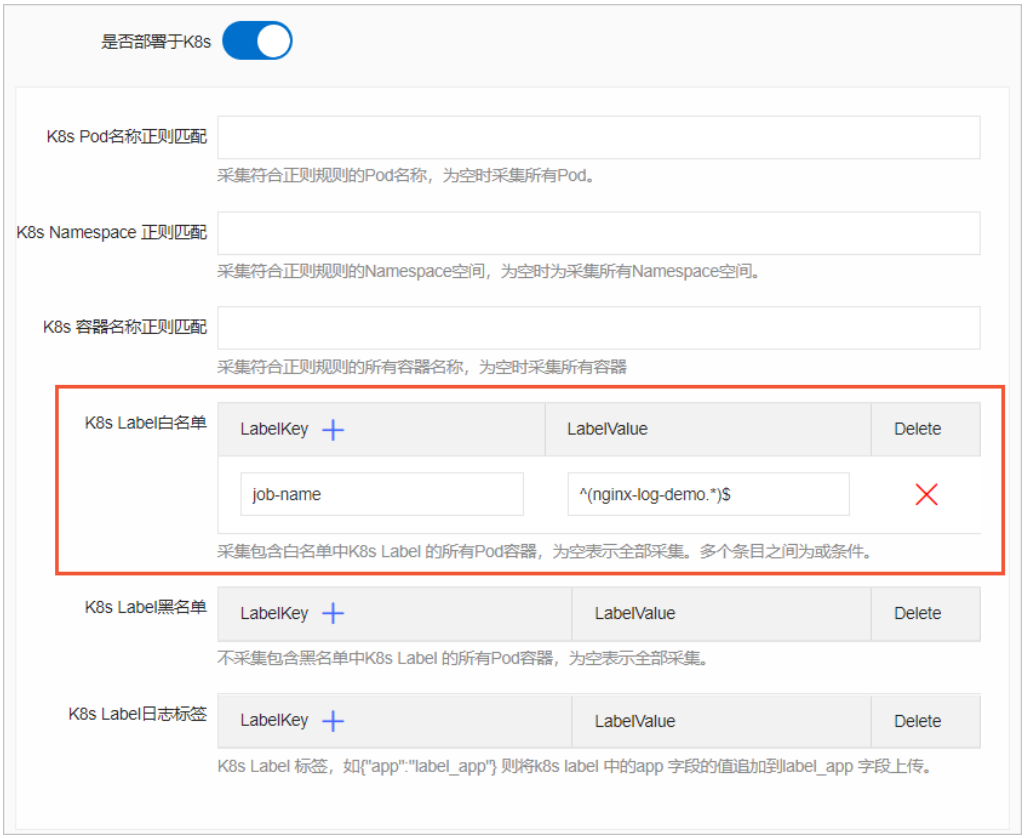
采集Kubernetes Label中Key为job-name，Value以nginx-log-demo开头的所有容器。

1. 获取Kubernetes Label

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/psp: ack.privileged
  creationTimestamp: "2022-01-06T18:42:43Z"
  generateName: nginx-log-demo-0-
  labels:
    controller-uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
    job-name: nginx-log-demo-0
  name: nginx-log-demo-0-bx179
  namespace: default
  ownerReferences:
  - apiVersion: batch/v1
    blockOwnerDeletion: true
    controller: true
    kind: Job
    name: nginx-log-demo-0
    uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
  resourceVersion: "50566856"
  uid: ee10fb7d-d989-47b3-bc2a-e9ffbe767849
```

2. 创建Logtail采集配置

Logtail采集配置示例如下图所示。极简模式的相关配置说明请参见使用极简模式采集日志。



默认字段

每条容器文本日志默认包含的字段如下表所示。

字段名称	说明
_image_name_	镜像名
_container_name_	容器名
_pod_name_	Pod名
namespace	Pod所在的命名空间
_pod_uid_	Pod的唯一标识
_container_ip_	Pod的IP地址

3.6.4. 通过DaemonSet-控制台方式采集容器标准输出

本文介绍如何通过控制台创建Logtail采集配置，并以DaemonSet采集方式采集容器标准输出。

前提条件

- 已安装Logtail组件。具体操作，请参见安装Logtail组件。
- 在您安装Logtail组件时所使用的Project中已创建Logstore。具体操作，请参见创建Logstore。

功能特点

Logtail支持采集容器内产生的标准输出，并附加容器的相关元数据信息一起上传到日志服务。Logtail具备以下功能特点。

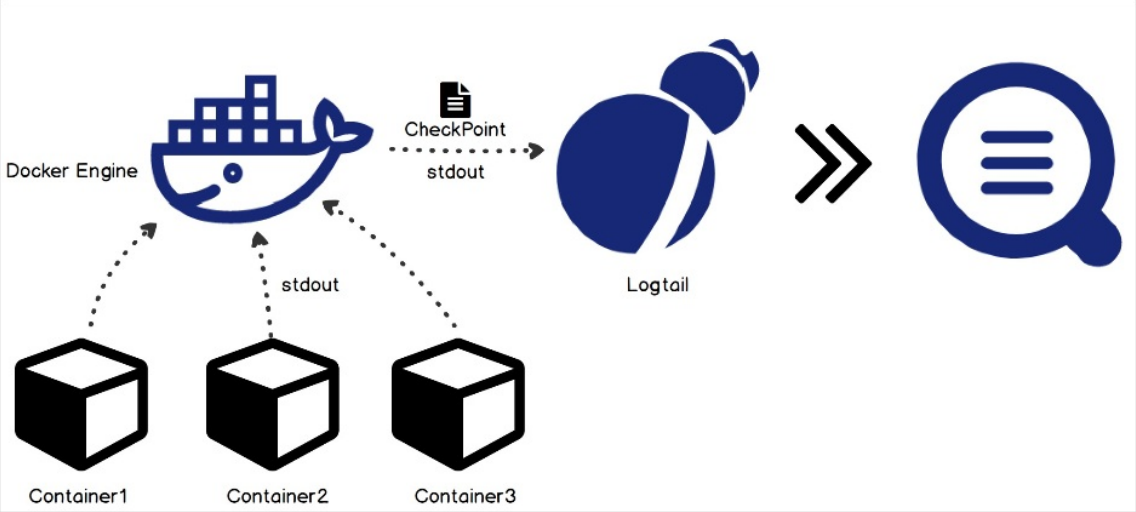
- 支持采集标准输出信息（stdout）和标准出错信息（stderr）。
- 支持通过容器Label白名单指定待采集的容器。
- 支持通过容器Label黑名单排除不要采集的容器。
- 支持通过环境变量白名单指定待采集的容器。

- 支持通过环境变量黑名单排除不要采集的容器。
- 支持采集多行日志（例如Java Stack日志等）。
- 支持上报容器标准输出时自动关联Meta信息（例如容器名、镜像、Pod、Namespace、环境变量等）。
- 当容器运行于Kubernetes时，Logtail还具有以下功能。
  - 支持通过Kubernetes Namespace名称、Pod名称、容器名称指定待采集的容器。
  - 支持通过Kubernetes Label白名单指定待采集的容器。
  - 支持通过Kubernetes Label黑名单排除不要采集的容器。
  - 支持上报容器标准输出时自动关联Kubernetes Label信息。

实现原理

Logtail与Docker的Domain Socket进行通信，查询该Docker上运行的所有容器，并根据容器中的Label和环境变量定位需要被采集的容器。Logtail通过 `docker logs` 命令获取指定容器日志。

Logtail在采集容器的标准输出时，会定期将采集的点位信息保存到checkpoint文件中。如果Logtail停止后再次启动，会从上一次保存的点位开始采集。



使用限制

- 此功能目前仅支持Linux操作系统，依赖Logtail 0.16.0及以上版本。版本查看与升级，请参见安装Logtail（Linux系统）。
- Logtail支持Docker和Containerd两种容器引擎的数据采集，访问路径说明如下：
  - Docker：Logtail通过*/run/docker.sock*访问Docker，请确保该路径存在且具备访问权限。
  - Containerd：Logtail通过*/run/containerd/containerd.sock*访问Containerd，请确保该路径存在且具备访问权限。
- 多行日志限制：为保证多行组成的一条日志不因为输出延迟而被分割成多条，多行日志情况下，采集的最后一条日志默认都会缓存一段时间。默认缓存时间为3秒，可通过 `BeginLineTimeoutMs` 参数修改，但此值不能低于1000（毫秒），否则容易出现误判。
- 采集停止策略：当容器被停止后，Logtail监听到容器 `die` 的事件后会停止采集该容器的标准输出。如果此时采集出现延迟，则可能丢失停止前的部分输出。
- Docker容器引擎限制：目前标准输出采集仅支持JSON类型的日志驱动。
- 上下文限制：默认一个Logtail采集配置在同一上下文中。如果需要每个容器的日志在不同上下文中，请单独为每个容器创建Logtail采集配置。
- 数据处理：采集到的数据默认保存在 `content` 字段中。Logtail对于采集到的容器标准输出，支持数据处理。更多信息，请参见使用Logtail插件处理数据。

创建采集配置

1. 登录日志服务控制台。
2. 在**接入数据**区域，单击**Kubernetes-标准输出**。
3. 选择目标Project和Logstore，单击**下一步**。

   选择您在安装Logtail组件时所使用的Project。Logstore为您自定义创建的Logstore。
4. 单击**使用现有机器组**。

   安装Logtail组件后，日志服务自动创建名为 `k8s-group-${your_k8s_cluster_id}` 的机器组，您可以直接使用该机器组。
5. 选中目标机器组（ `k8s-group-${your_k8s_cluster_id}` ），将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

> 注意 如果机器组心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 设置数据源，单击**下一步**。

   在**插件配置**中填写您的采集配置信息，示例如下所示。

```
{
    "inputs":[
        {
            "type":"service_docker_stdout",
            "detail":{
                "Stdout":true,
                "Stderr":true,
                "IncludeContainerLabel":{
                    "LabelKey":"LabelValue"
                },
                "ExcludeContainerLabel":{
                    "LabelKey":"LabelValue"
                },
                "IncludeK8sLabel":{
                    "LabelKey":"LabelValue"
                },
                "ExcludeK8sLabel":{
                    "LabelKey":"LabelValue"
                },
                "IncludeEnv":{
                    "EnvKey":"EnvValue"
                },
                "ExcludeEnv":{
                    "EnvKey":"EnvValue"
                },
                "ExternalK8sLabelTag":{
                    "EnvKey":"EnvValue"
                },
                "ExternalEnvTag":{
                    "EnvKey":"EnvValue"
                },
                "K8sNamespaceRegex":"^(default|kube-system)$",
                "K8sPodRegex":"^(deploy.*)$",
                "K8sContainerRegex":"^(container1|container2)$"
            }
        }
    ]
}
```

   重要参数说明如下：

   - 数据源类型

     固定为service_docker_stdout。

   - 容器过滤相关参数

     - Logtail 1.0.34以下版本，只支持通过环境变量、容器Label进行容器过滤。详细说明如下：

       Kubernetes中的命名空间名和容器名会映射到容器Label中，分别为 `io.kubernetes.pod.namespace` 和 `io.kubernetes.container.name` 。推荐使用这两个Label进行容器过滤。如果这两个Label未满足需求，请使用环境变量的黑白名单进行容器过滤。例如某Pod所属的命名空间为backend-prod，容器名为worker-server，如果您要采集包含该容器的日志，可以设置Label白名单为 `"io.kubernetes.pod.namespace" : "backend-prod"` 或 `"io.kubernetes.container.name" : "worker-server"` 。

       > 注意
       > - 容器Label为Docker inspect中的Label，不是Kubernetes中的Label。如何获取，请参见获取容器Label。
       > - 环境变量为容器启动中配置的环境变量信息。如何获取，请参见获取容器环境变量。
       > - 请勿设置相同的LabelKey，如果重名只生效一个。

参数	数据类型	是否必填	说明

参数	数据类型	是否必填	说明
IncludeLabel	map类型，其中LabelKey和LabelValue为string类型。	可选	容器Label白名单，用于指定待采集的容器。默认为空，表示采集所有容器的标准输出。如果您要设置容器Label白名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则容器Label中包含LabelKey的容器都匹配。 ■ 如果LabelValue不为空，则容器Label中包含LabelKey=LabelValue的容器才匹配。 LabelValue默认为字符串匹配，即只有LabelValue和容器Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置LabelKey为*io.kubernetes.container.name*，设置LabelValue为*^(nginx\|cube)$*，表示可匹配名为nginx、cube的容器。 多个白名单之间为或关系，即只要容器Label满足任一白名单即可匹配。
ExcludeLabel	map类型，其中LabelKey和LabelValue为string类型。	可选	容器Label黑名单，用于排除不采集的容器。默认为空，表示不排除任何容器。如果您要设置容器Label黑名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则容器Label中包含LabelKey的容器都将被排除。 ■ 如果LabelValue不为空，则容器Label中包含LabelKey=LabelValue的容器才会被排除。 LabelValue默认为字符串匹配，即只有LabelValue和容器Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置LabelKey为*io.kubernetes.container.name*，设置LabelValue为*^(nginx\|cube)$*，表示可匹配名为nginx、cube的容器。 多个黑名单之间为或关系，即只要容器Label满足任一黑名单对即可被排除。
IncludeEnv	map类型，其中EnvKey和EnvValue为string类型。	可选	环境变量白名单，用于指定待采集的容器。默认为空，表示采集所有容器的标准输出。如果您要设置环境变量白名单，那么EnvKey必填，EnvValue可选填。 ■ 如果EnvValue为空，则容器环境变量中包含EnvKey的容器都匹配。 ■ 如果EnvValue不为空，则容器环境变量中包含EnvKey=EnvValue的容器才匹配。 EnvValue默认为字符串匹配，即只有EnvValue和环境变量的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配，例如设置EnvKey为*NGINX_SERVICE_PORT*，设置EnvValue为*^(80\|6379)$*，表示可匹配服务端口为80、6379的容器。 多个白名单之间为或关系，即只要容器的环境变量满足任一键值对即可匹配。
ExcludeEnv	map类型，其中EnvKey和EnvValue为string类型。	可选	环境变量黑名单，用于排除不采集的容器。默认为空，表示不排除任何容器。如果您要设置环境变量黑名单，那么EnvKey必填，EnvValue可选填。 ■ 如果EnvValue为空，则容器环境变量中包含EnvKey的容器的日志都将被排除。 ■ 如果EnvValue不为空，则容器环境变量中包含EnvKey=EnvValue的容器才会被排除。 EnvValue默认为字符串匹配，即只有EnvValue和环境变量的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配，例如设置EnvKey为*NGINX_SERVICE_PORT*，设置EnvValue为*^(80\|6379)$*，表示可匹配服务端口为80、6379的容器。 多个黑名单之间为或关系，即只要容器的环境变量满足任一键值对即可被排除。

- Logtail 1.0.34及以上版本，推荐使用Kubernetes层级的信息（Pod名称、Namespace名称、容器名称、Label）进行容器过滤。

  > **说明** 由于在Kubernetes管控类资源（例如Deployment ）运行时更改Label，不会重启具体的工作资源Pod，因此Pod无法感知此变更，可能导致匹配规则失效。设置K8s Label黑白名单时，请以Pod中的Kubernetes Label为准。关于Kubernetes Label的更多信息，请参见Labels and Selectors。

参数	数据类型	是否必填	说明
IncludeK8sLabel	map类型，其中LabelKey和LabelValue为string类型。	可选	通过Kubernetes Label白名单指定待采集的容器。如果您要设置Kubernetes Label白名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则Kubernetes Label中包含LabelKey的容器都匹配。 ■ 如果LabelValue不为空，则Kubernetes Label中包含LabelKey=LabelValue的容器才匹配。 LabelValue默认为字符串匹配，即只有LabelValue和Kubernetes Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置**LabelKey**为*app*，设置**LabelValue**为*^(test1\|test2)$*，表示匹配Kubernetes Label中包含app:test1、app:test2的容器。 多个白名单之间为或关系，即只要Kubernetes Label满足任一白名单即可匹配。
ExcludeK8sLabel	map类型，其中LabelKey和LabelValue为string类型。	可选	通过Kubernetes Label黑名单排除不采集的容器。如果您要设置Kubernetes Label黑名单，那么LabelKey必填，LabelValue可选填。 ■ 如果LabelValue为空，则Kubernetes Label中包含LabelKey的容器都被排除。 ■ 如果LabelValue不为空，则Kubernetes Label中包含LabelKey=LabelValue的容器才会被排除。 LabelValue默认为字符串匹配，即只有LabelValue和Kubernetes Label的值完全相同才会匹配。如果该值以 `^` 开头并且以 `$` 结尾，则为正则匹配。例如设置**LabelKey**为*app*，设置**LabelValue**为*^(test1\|test2)$*，表示匹配Kubernetes Label中包含app:test1、app:test2的容器。 多个黑名单之间为或关系，即只要Kubernetes Label满足任一黑名单对即可被排除。
K8sNamespaceRegex	string	可选	通过Namespace名称指定采集的容器，支持正则匹配。例如设置为"K8sNamespaceRegex":"^(default|nginx)$"，表示匹配nginx命名空间、default命名空间下的所有容器。
K8sPodRegex	string	可选	通过Pod名称指定待采集的容器，支持正则匹配。例如设置为"K8sPodRegex":"^(nginx-log-demo.*)$",，表示匹配以nginx-log-demo开头的Pod下的所有容器。
K8sContainerRegex	string	可选	通过容器名称指定待采集的容器（Kubernetes容器名称是定义在spec.containers中），支持正则匹配。例如设置为"K8scontainerRegex":"^(container-test)$"，表示匹配所有名为container-test的容器。

- 日志标签相关参数

  如果您使用的是Logtail 1.0.34及以上版本，您可以将环境变量和Kubernetes Label添加到日志，作为日志标签。

参数	数据类型	是否必填	说明
ExternalEnvTag	map类型，其中EnvKey和EnvValue为string类型。	可选	设置环境变量日志标签后，日志服务将在日志中新增环境变量相关字段。例如设置**EnvKey**为*VERSION*，设置**EnvValue**为*env_version*，当容器中包含环境变量 `VERSION=v1.0.0` 时，会将该信息添加到日志中，即添加字段__tag__:__env_version__: v1.0.0。
ExternalK8sLabelTag	map类型，其中LabelKey和LabelValue为string类型。	可选	设置Kubernetes Label日志标签后，日志服务将在日志中新增Kubernetes Label相关字段。例如设置**LabelKey**为*app*，设置**LabelValue**为 `k8s_label_app` ，当Kubernetes中包含Label `app=serviceA` 时，会将该信息添加到日志中，即添加字段__tag__:__k8s_label_app__: serviceA。

- 其他参数

参数	数据类型	是否必填	说明
Stdout	boolean	可选	是否采集标准输出stdout。 该配置项为空，表示true，即默认采集标准输出stdout。
Stderr	boolean	可选	是否采集标准出错信息stderr。 该配置项为空，表示true，即默认采集标准出错信息stderr。
BeginLineRegex	string	可选	行首匹配的正则表达式。 该配置项为空，表示单行模式。 如果该表达式匹配某行的开头，则将该行作为一条新的日志，否则将此行拼接到上一条日志。
BeginLineTimeoutMs	int	可选	行首匹配的超时时间。 该配置项为空，表示使用默认的超时时间，即3000毫秒。 如果3000毫秒内没有出现新日志，则结束匹配，将最后一条日志上传到日志服务。
BeginLineCheckLength	int	可选	行首匹配的长度。 该配置项为空，表示使用默认的行首匹配的长度，即10×1024字节。 如果行首匹配的正则表达式在前N个字节即可体现，推荐设置此参数，提升行首匹配效率。
MaxLogSize	int	可选	日志最大长度。 该配置项为空，表示使用默认的最大长度，即512×1024字节。 如果日志长度超过该值，则不再继续查找行首，直接上传。
StartLogMaxOffset	int	可选	首次采集时回溯历史数据长度。建议取值在[131072,1048576]之间，单位：字节。 该配置项为空，表示使用默认的回溯历史数据长度，即131072字节（128 KB）。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

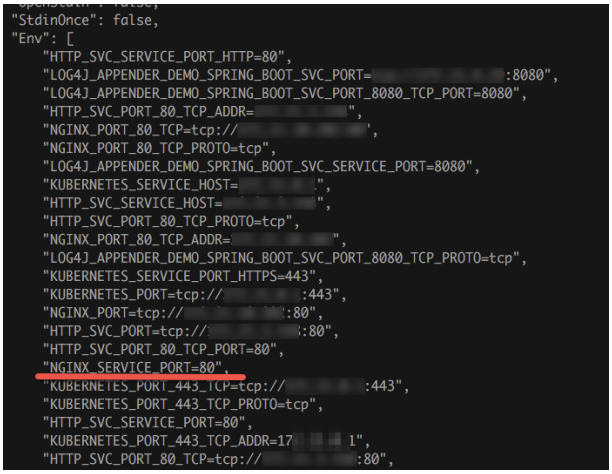
普通日志的Logtail采集配置示例

示例1：通过环境变量黑白名单过滤容器

采集环境变量为`NGINX_SERVICE_PORT=80`且不为`POD_NAMESPACE=kube-system`的容器的标准输出。

1. 获取环境变量。

   您可以登录容器所在的宿主机查看容器的环境变量。具体操作，请参见获取容器环境变量。



2. 创建Logtail采集配置。

   Logtail采集配置示例如下所示。

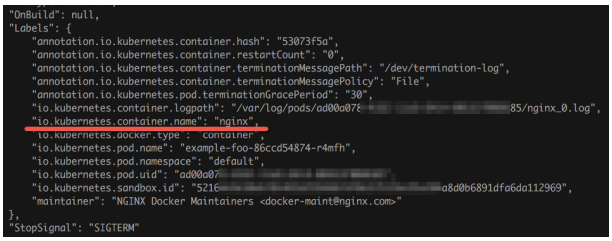
```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeEnv": {
                    "NGINX_SERVICE_PORT": "80"
                },
                "ExcludeEnv": {
                    "POD_NAMESPACE": "kube-system"
                }
            }
        }
    ]
}
```

示例2：通过容器Label黑白名单过滤容器

采集容器Label为 `io.kubernetes.container.name=nginx` 的容器的标准输出。

1. 获取容器Label。

   您可以登录容器所在的宿主机查看容器的Label。具体操作，请参见获取容器Label。



2. 创建Logtail采集配置。

   Logtail采集配置示例如下所示。

```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeLabel": {
                    "io.kubernetes.container.name": "nginx"
                }
            }
        }
    ]
}
```

示例3：通过Kubernetes Namespace名称、Pod名称和容器名称过滤容器

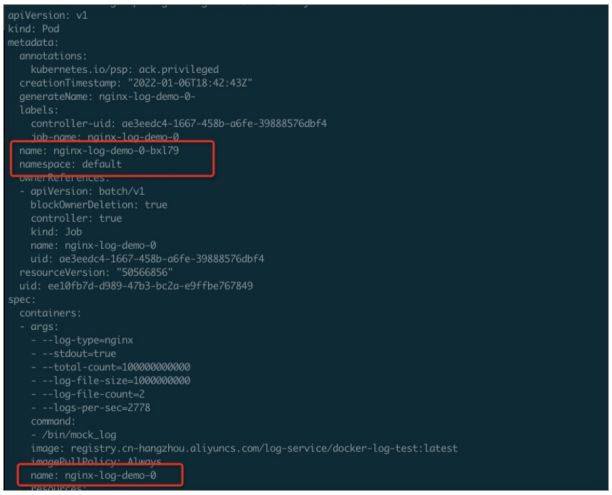
采集default命名空间下以nginx-log-demo开头的Pod中的nginx-log-demo-0容器的标准输出。

1. 获取Kubernetes层级的信息。

   i. 获取Pod信息。

```
~/.kube » kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-log-demo-0-bxl79  1/1     Running   0          48d
nginx-log-demo-1-qmrqk  1/1     Running   0          48d
nginx-log-demo-2-7khv9  1/1     Running   0          48d
nginx-log-demo-3-j24xc  1/1     Running   0          48d
```

ii. 获取Namespace等信息。



2. 创建Logtail采集配置。

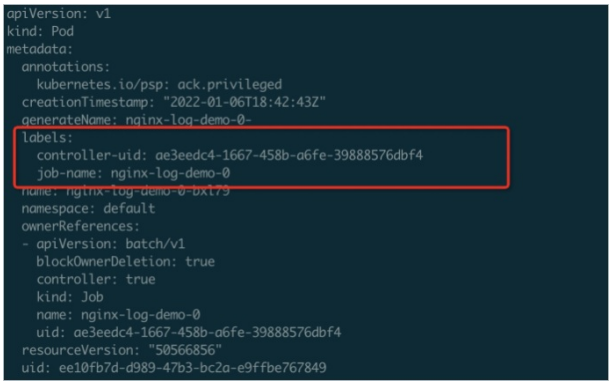
Logtail采集配置示例如下所示。

```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "K8sNamespaceRegex":"^(default)$",
                "K8sPodRegex":"^(nginx-log-demo.*)$",
                "K8sContainerRegex":"^(nginx-log-demo-0)$"
            }
        }
    ]
}
```

示例4：通过Kubernetes Label过滤容器

采集Kubernetes Label中Key为job-name，Value以nginx-log-demo开头的所有容器的标准输出。

1. 获取Kubernetes Label。



2. 创建Logtail采集配置。

Logtail采集配置示例如下所示。

```
{
    "inputs": [
        {
            "type": "service_docker_stdout",
            "detail": {
                "Stdout": true,
                "Stderr": true,
                "IncludeK8sLabel":{
                    "job-name":"^(nginx-log-demo.*)$"
                }
            }
        }
    ]
}
```

多行日志的Logtail采集配置示例

对于Java异常堆栈输出（多行日志），您可以参见如下配置。

- 日志示例

```
2021-02-03 14:18:41.968  INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service sta
rt
2021-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : java.lang.N
ullPointerException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:96)
...
2021-02-03 14:18:41.968  INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service sta
rt done
```

- Logtail采集配置

  采集容器Label为`app=monitor`的容器的Java异常堆栈输出，该输出内容以固定格式的日期开头。为提高匹配效率，此处只判断行首的10个字节。采集到日志服务后，日志服务使用正则表达将其解析成time、level、module、thread、message等字段。

  - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

    > **说明** 一个inputs中只允许配置一个类型的数据源。

  - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式。具体操作，请参见使用Logtail插件处理数据。

```
{
"inputs": [
  {
    "detail": {
      "BeginLineCheckLength": 10,
      "BeginLineRegex": "\\d+-\\d+-\\d+.*",
      "IncludeLabel": {
        "app": "monitor"
      }
    },
    "type": "service_docker_stdout"
  }
],
"processors": [
    {
        "type": "processor_regex",
        "detail": {
            "SourceKey": "content",
            "Regex": "(\\d+-\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\\w+)\\s+\\[([^]]+)]\\s+\\[([^]]+)]\\s+([\\s\\S]*)",
            "Keys": [
                "time",
                "level",
                "module",
                "thread",
                "message"
            ],
            "NoKeyError": true,
            "NoMatchError": true,
            "KeepSource": false
        }
    }
]
}
```

- 解析后的日志

  例如 `2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.controller.DemoController : service start done` 日志，经过解析后的日志内容如下所示。其中：

  - _time_为日志采集时间。您可以通过Logtail采集配置中的**时区属性**参数设置_time_的时区。
  - time为日志中存在的时间内容，是从日志中提取得到的。

```
__tag__:__hostname__:logtail-dfgef
_container_name_:monitor
_image_name_:example.com-hangzhou.aliyuncs.xxxxxxxxxxxxxx
_namespace_:default
_pod_name_:monitor-6f54bd5d74-rtzc7
_pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369
_source_:stdout
_time_:2018-02-02T14:18:41.979147844Z
time:2018-02-02 02:18:41.968
level:INFO
module:spring-cloud-monitor
thread:nio-8080-exec-4
class:c.g.s.web.controller.DemoController
message:service start done
```

日志字段

Kubernetes集群的每条日志默认上传的字段如下所示。

字段名称	说明
time	日志采集时间，例如 `2021-02-02T02:18:41.979147844Z` 。
source	日志源类型，stdout或stderr。
_image_name_	镜像名
_container_name_	容器名

字段名称	说明
_pod_name_	Pod名
namespace	Pod所在的命名空间
_pod_uid_	Pod的唯一标识

3.6.5. 通过DaemonSet-CRD方式采集容器日志

在容器中以DaemonSet模式安装Logtail后，可通过CRD方式创建Logtail采集配置采集容器日志。

前提条件

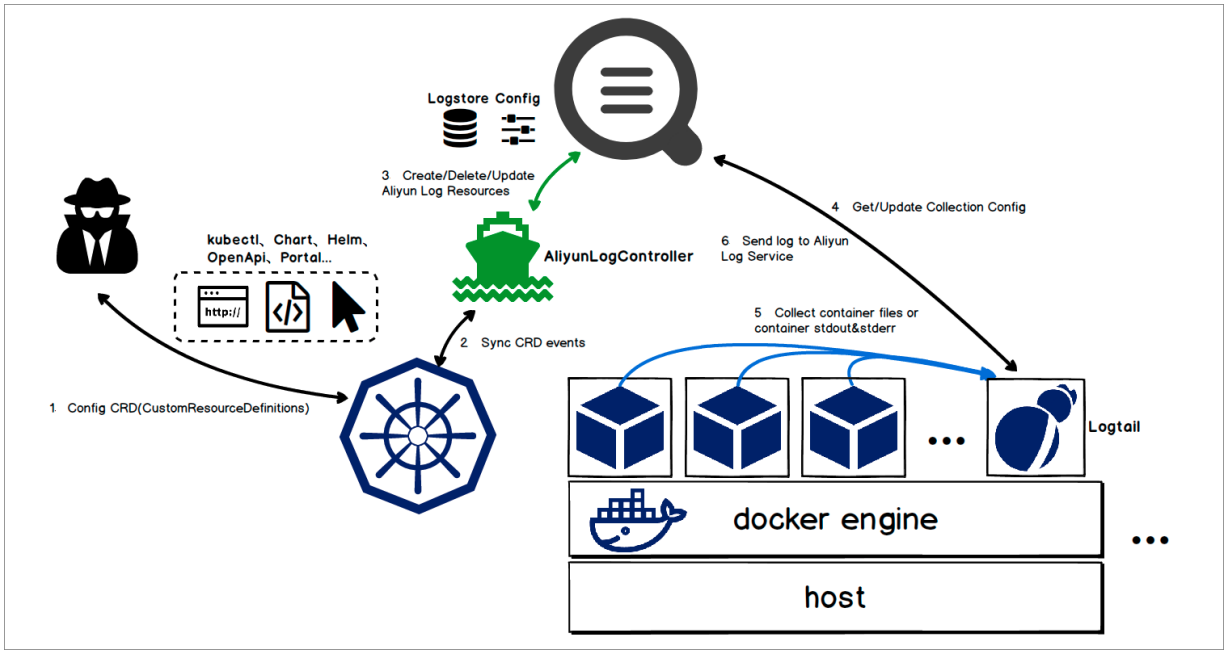
已安装Logtail组件。更多信息，请参见安装Logtail组件。

功能特点

Logtail支持将容器产生的文本日志和容器相关的元数据信息一起上传到日志服务。Logtail具备以下功能特点。

- 只需配置容器内的日志文件路径，无需关心该路径到宿主机的映射。
- 支持通过容器Label白名单指定待采集的容器。
- 支持通过容器Label黑名单排除不要采集的容器。
- 支持通过环境变量白名单指定待采集的容器。
- 支持通过环境变量黑名单排除不要采集的容器。
- 支持采集多行日志（例如Java Stack日志）。
- 支持上报容器日志时自动关联Meta信息（例如容器名、镜像、Pod、Namespace、环境变量等）。
- 当容器运行于Kubernetes时，Logtail还具有以下功能。
  - 支持通过Kubernetes Namespace名称、Pod名称、容器名称指定待采集的容器。
  - 支持通过Kubernetes Label白名单指定待采集的容器。
  - 支持通过Kubernetes Label黑名单排除不要采集的容器。
  - 支持上报容器日志时自动关联Kubernetes Label信息。

实现原理



CRD的内部工作流程如下：

1. 使用 `kubectl` 或其他工具应用AliyunLogConfig CRD。
2. alibaba-log-controller监听到CRD配置更新。
3. alibaba-log-controller根据CRD配置中的内容以及日志服务中Logtail采集配置的状态，自动向日志服务提交创建Logstore、创建Logtail采集配置以及应用机器组的请求。

4. Logtail定期请求Logtail采集配置所在服务器，获取新的或已更新的Logtail采集配置并进行热加载。
5. Logtail根据Logtail采集配置采集各个容器上的标准输出或文本日志。
6. Logtail将采集到的容器日志发送给日志服务。

使用限制

- 文本日志采集限制
  - 采集停止策略：当容器被停止后，Logtail监听到容器 `die` 的事件后会停止该容器日志的采集。如果此时采集出现延迟，则可能丢失停止前的部分日志。
  - 不支持采集软链接：目前Logtail无法访问业务容器的软链接，请按真实路径配置采集目录。
  - 如果业务容器的数据目录是通过数据卷（Volume）挂载的，则不支持采集它的父目录，需设置采集目录为完整的数据目录。

    例如*/var/log/service*目录是数据卷挂载的路径，则设置采集目录为*/var/log*将采集不到该目录下的日志，需设置采集目录为*/var/log/service*。
  - Kubernetes默认将宿主机根目录挂载到Logtail容器的 `/logtail_host` 目录。如果您要采集宿主机文本日志，则配置日志文件路径时，需加上 `/logtail_host` 前缀。

    例如需要采集宿主机上 `/home/logs/app_log/` 目录下的日志，则设置日志路径为 `/logtail_host/home/logs/app_log/` 。
  - Docker存储驱动限制：目前只支持overlay、overlay2，其他存储驱动需将日志所在目录通过数据卷挂载为临时目录。

    如果日志目录是以PVC方式挂载到NAS，则不支持使用Daemonset方式采集日志，建议使用Sidecar方式采集。
- 标准输出采集限制

  Docker容器引擎限制：目前标准输出采集仅支持JSON类型的日志驱动。
- 通用限制

  Logtail支持Docker和Containerd两种容器引擎的数据采集，访问路径说明如下：
  - Docker：Logtail通过*/run/docker.sock*访问Docker，请确保该路径存在且具备访问权限。
  - Containerd：Logtail通过*/run/containerd/containerd.sock*访问Containerd，请确保该路径存在且具备访问权限。

创建Logtail采集配置

您只需要定义AliyunLogConfig CRD即可创建Logtail采集配置。创建完成后，系统自动应用该Logtail采集配置。如果您要删除Logtail采集配置只需删除对应的CRD资源即可。

1. 登录Kubernetes集群。
2. 执行如下命令创建一个YAML文件。

   *cube.yaml*为文件名，请根据实际情况替换。

   ```
   vim cube.yaml
   ```
3. 在YAML文件输入如下脚本，并根据实际情况设置其中的参数。

   > **注意**
   > - 请确保configName字段值在日志服务Project中唯一存在。
   > - 如果多个CRD关联同一个Logtail采集配置，则删除或修改任意一个CRD均会影响到该Logtail采集配置，导致其他关联该Logtail采集配置的CRD状态与日志服务中Logtail采集配置的状态不一致。

   ```
   apiVersion: log.alibabacloud.com/v1alpha1      # 使用默认值，无需修改。
   kind: AliyunLogConfig                          # 使用默认值，无需修改。
   metadata:
     name: simple-stdout-example                  # 设置资源名，在当前Kubernetes集群内唯一。
   spec:
     project: k8s-my-project                      # [可选]设置Project名称。默认为安装Logtail组件时设置的Project。
     logstore: k8s-stdout                         # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
     shardCount: 2                                # [可选]设置Shard数量。默认值为2，取值范围1~10。
     lifeCycle: 90                                # [可选]设置Logstore中数据的存储时间，该参数值仅在新建Logstore时生效。默认值为90，取值范围为1~3650。其中，3650天为永久存储。
     logtailConfig:                               # 设置Logtail采集配置。
       inputType: plugin                          # 设置采集的数据源类型。file表示采集文本日志或plugin表示采集标准输出。
       configName: simple-stdout-example          # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
       inputDetail:                               # 设置Logtail采集配置的详细信息，具体配置请参见本文下方的示例。
         ...
   ```

参数	数据类型	是否必填	说明
project	string	否	Project名称。默认为安装Logtail组件时设置的Project。
logstore	string	是	Logstore名称。 如果您所指定的Logstore不存在，日志服务会自动创建。
shardCount	int	否	Shard数量。默认值为2，取值范围为1~10。
lifeCycle	int	否	Logstore中数据的存储时间。默认值为90，取值范围为1~3650。其中，3650天为永久存储。 注意 该参数值仅在新建Logstore时生效，即您只能在创建Logstore时，指定数据的存储时间。如果您在logstore参数中指定的Logstore已存在，则修改该参数值，不会生效。
machineGroups	array	否	机器组。默认为安装Logtail组件时，日志服务自动创建名为 `k8s-group-${your_k8s_cluster_id}` 的机器组。
logtailConfig	object	是	Logtail采集配置的详细定义，一般只需要定义其中的inputType参数、configName参数和inputDetail参数。详细参数说明，请参见Logtail配置。 logtailConfig的配置示例请参见Logtail采集配置示例（标准输出）和Logtail采集配置示例（文本日志）。

4. 执行如下命令使Logtail采集配置生效。

   *cube.yaml*为文件名，请根据实际情况替换。

   ```
   kubectl apply -f cube.yaml
   ```

   Logtail采集配置生效后，Logtail开始采集各个容器上的标准输出或文本日志，并发送到日志服务中。

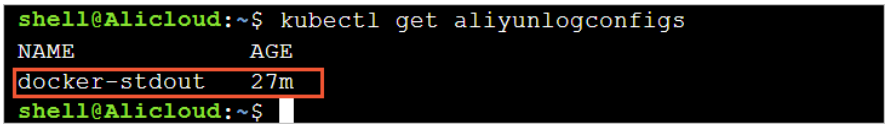
查看Logtail采集配置

您可以通过CRD方式或控制台方式查看Logtail采集配置，其中控制台方式请参见查看Logtail采集配置。

注意 如果您使用的是CRD方式，则您在控制台上对Logtail采集配置的修改不会同步到CRD中，但您在CRD中对Logtail采集配置的修改会同步到控制台。

查看当前Kubernetes集群中所有的Logtail采集配置

您可以执行 `kubectl get aliyunlogconfigs` 命令进行查看，返回结果下图所示。



查看Logtail采集配置的详细信息和状态

您可以执行 `kubectl get aliyunlogconfigs config_name -o yaml` 命令进行查看。其中，*config_name*为Logtail采集配置的名称，请根据实际情况替换。 返回结果如下图所示。

执行结果中的status字段和statusCode字段表示Logtail采集配置的状态。

- 如果statusCode字段的值为200，表示应用Logtail采集配置成功。
- 如果statusCode字段的值为非200，表示应用Logtail采集配置失败。

```
shell@Alicloud:~$ kubectl get aliyunlogconfigs docker-stdout -o yaml
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"log.alibabacloud.com/v1alpha1","kind":"AliyunLogConfig","metadata":{"annotations":{},"name":"docker-stdout","namespace":"default"},"spec":{"logstore":"
cube-stdout","logtailConfig":{"configName":"docker-stdout","inputDetail":{"plugin":{"inputs":[{"detail":{"Stderr":true,"Stdout":true},"type":"service_docker_stdout"}]}},"in
putType":"plugin"}}}
  creationTimestamp: "2021-10-29T08:40:33Z"
  generation: 2
  name: docker-stdout
  namespace: default
  resourceVersion: "1350968"
  uid: 35f9516b-[illegible]-69fabdc
spec:
  extenions: ""
  lifeCycle: null
  logstore: cube-stdout
  logtailConfig:
    configName: docker-stdout
    inputDetail:
      plugin:
        inputs:
        - detail:
            Stderr: true
            Stdout: true
          type: service_docker_stdout
    inputType: plugin
  machineGroups: null
  productCode: ""
  productLanguage: ""
  project: ""
  shardCount: null
status:
  status: OK
  statusCode: 200
```

Logtail采集配置示例（标准输出）

采集容器标准输出时，需将inputType设置为 `plugin` ，并将具体信息填写到 `inputDetail` 下的 `plugin` 字段。具体的参数及其说明请参见通过DaemonSet-控制台方式采集容器标准输出。

示例1：通过极简模式采集容器标准输出

通过极简模式采集除环境变量为 `COLLECT_STDOUT_FLAG=false` 之外的所有容器的标准输出（stdout和stderr）。其中，您可以登录容器所在的宿主机查看容器的环境变量。具体操作，请参见获取容器环境变量。CRD配置示例如下所示。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: simple-stdout-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-stdout
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集标准输出时，需设置为plugin。
    inputType: plugin
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: simple-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # 指定采集stdout和stderr。
              Stdout: true
              Stderr: true
              # 设置环境变量黑名单，采集除环境变量为COLLECT_STDOUT_FLAG=false之外的所有容器的标准输出。
              ExcludeEnv:
                COLLECT_STDOUT_FLAG: "false"
```

示例2：通过极简模式采集容器标准输出，并使用正则模式处理容器标准输出

您可以登录容器所在的宿主机查看容器的环境变量。具体操作，请参见获取容器环境变量。

通过极简模式采集容器中Grafana的访问日志，并使用正则模式将其解析为结构化数据。其中，Grafana容器的环境变量为 `GF_INSTALL_PLUGINS=grafana-piechart-....` ，您可以登录容器所在的宿主机进行查看。具体操作，请参见获取容器环境变量。

- CRD配置

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: regex-stdout-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-stdout-regex
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集标准输出时，需设置为plugin。
    inputType: plugin
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: regex-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # 指定只采集标准输出stdout，不采集标准错误stderr。
              Stdout: true
              Stderr: false
              # 设置环境变量白名单，只采集容器环境变量中EnvKey为GF_INSTALL_PLUGINS的容器的标准输出。
              IncludeEnv:
                GF_INSTALL_PLUGINS: ''
        processors:
          -
            # 指定正则模式解析所采集到的标准输出。
            type: processor_regex
            detail:
              # 设置原始字段名。采集到的容器标准输出默认保存在content字段中。
              SourceKey: content
              # 设置正则表达式，用于提取日志内容。
              Regex: 't=(\d+-\d+-\w+:\d+:\d+\+\d+) lvl=(\w+) msg="([^"]+)" logger=(\w+) userId=(\w+) orgId=(\w+) uname=(
\S*) method=(\w+) path=(\S+) status=(\d+) remote_addr=(\S+) time_ms=(\d+) size=(\d+) referer=(\S*).*'
              # 设置提取的字段列表。
              Keys: ['time', 'level', 'message', 'logger', 'userId', 'orgId', 'uname', 'method', 'path', 'status', 'remo
te_addr', 'time_ms', 'size', 'referer']
              # 保留原始字段。
              KeepSource: true
              # 出现无匹配的原始字段时会报错。
              NoKeyError: true
              # 正则表达式与原始字段的值不匹配时会报错。
              NoMatchError: true
```

- 原始日志

```
t=2018-03-09T07:14:03+0000 lvl=info msg="Request Completed" logger=context userId=0 orgId=0 uname= method=GET path=/ sta
tus=302 remote_addr=172.16.64.154 time_ms=0 size=29 referer=
```

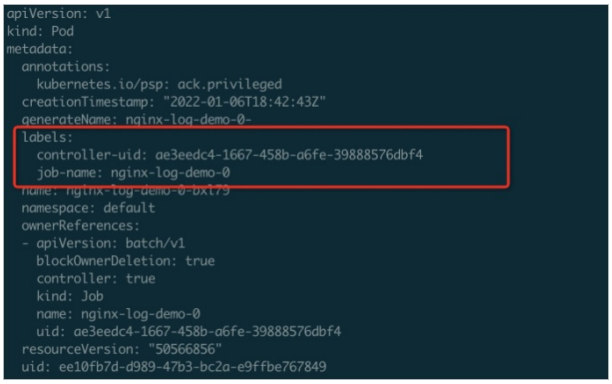
- 解析后的日志

示例3：通过Kubernetes Label过滤容器，采集对应容器的标准输出

采集Kubernetes Label中Key为job-name，Value以nginx-log-demo开头的所有容器的标准输出。

> **说明** Logtail 1.0.34及以上版本支持通过K8s Label过滤容器。

1. 获取Kubernetes Label。



```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    kubernetes.io/psp: ack.privileged
  creationTimestamp: "2022-01-06T18:42:43Z"
  generateName: nginx-log-demo-0-
  labels:
    controller-uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
    job-name: nginx-log-demo-0
  name: nginx-log-demo-0-bxl79
  namespace: default
  ownerReferences:
  - apiVersion: batch/v1
    blockOwnerDeletion: true
    controller: true
    kind: Job
    name: nginx-log-demo-0
    uid: ae3eedc4-1667-458b-a6fe-39888576dbf4
  resourceVersion: "50566856"
  uid: ee10fb7d-d989-47b3-bc2a-e9ffbe767849
```

2. 创建Logtail采集配置。

   配置示例如下所示。参数说明，请参见通过DaemonSet-控制台方式采集容器标准输出。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: simple-stdout-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-stdout
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集标准输出时，需设置为plugin。
    inputType: plugin
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: k8s-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # 指定采集stdout和stderr。
              Stdout: true
              Stderr: true
              # 设置K8s Label白名单，采集Kubernetes Label中Key为job-name，Value以nginx-log-demo开头的所有容器的标准输出。
              IncludeK8sLabel:
                job-name: "^(nginx-log-demo.*)$"
```

示例4：通过Kubernetes Namespace名称、Pod名称和容器名称过滤容器，采集对应容器的标准输出

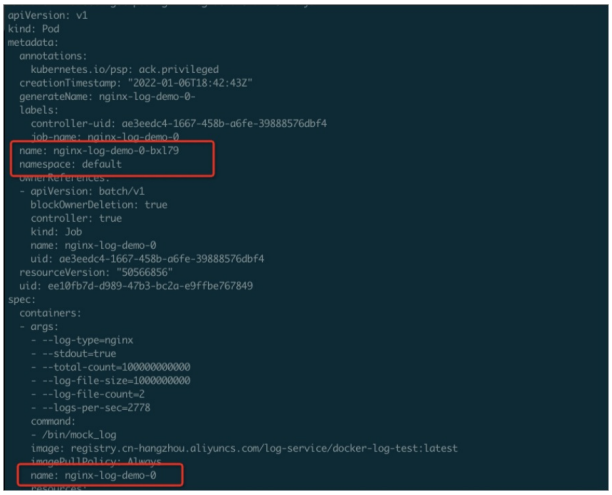
采集default命名空间下以nginx-log-demo开头的Pod中的nginx-log-demo-0容器的标准输出。

> **说明** Logtail 1.0.34及以上版本支持通过Kubernetes Namespace名称、Pod名称或容器名称过滤日志。

1. 获取Kubernetes层级的信息。
   i. 获取Pod信息。

```
~/.kube » kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-log-demo-0-bxl79  1/1     Running   0          48d
nginx-log-demo-1-qmrqk  1/1     Running   0          48d
nginx-log-demo-2-7khv9  1/1     Running   0          48d
nginx-log-demo-3-j24xc  1/1     Running   0          48d
```

ii. 获取Namespace等信息。



2. 创建Logtail采集配置。

配置示例如下所示。参数说明，请参见通过DaemonSet-控制台方式采集容器标准输出。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: simple-stdout-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-stdout
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集标准输出时，需设置为plugin。
    inputType: plugin
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: k8s-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # 指定采集stdout和stderr。
              Stdout: true
              Stderr: true
              K8sNamespaceRegex: ^(default)$
              K8sPodRegex: '^(nginx-log-demo.*)$'
              K8sContainerRegex: ^(nginx-log-demo-0)$
```

Logtail采集配置示例（文本日志）

采集容器文本日志时，需将 `inputType` 设置为 `file` ，并将具体信息填写到 `inputDetail` 内，具体字段及说明请参见通过DaemonSet-控制台方式采集容器文本日志。

示例1：通过极简模式采集容器文本日志

通过极简模式采集环境变量中包含EnvKey为 `ALIYUN_LOGTAIL_USER_DEFINED_ID` 的容器的文本日志，日志文件路径为*/data/logs/app_1/simple.LOG*。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: simple-file-example
spec:
   # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-file
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集文本日志时，需设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: simple-file-example
    inputDetail:
      # 指定通过极简模式采集文本日志。
      logType: common_reg_log
      # 设置日志文件所在路径。
      logPath: /data/logs/app_1
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: simple.LOG
      # 采集容器的文本日志时，需设置dockerFile为true。
      dockerFile: true
      # 设置环境变量白名单。只采集环境变量中包含EnvKey为ALIYUN_LOGTAIL_USER_DEFINED_ID的容器的文本日志。
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

示例2：通过完整正则模式采集容器文本日志

某Java程序日志为多行日志，日志中包含错误堆栈信息。您可以通过完整正则模式进行采集，并在Logtail采集配置中指定置行首正则表达式。

- 日志样例

```
[2018-05-11T20:10:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring sessions
java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F",...' for column 'data' at row 1
at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.translate(AbstractFallbackSQLExceptionTransla
tor.java:84)
at org.springframework.jdbc.support.AbstractFallbackSQLException
```

- CRD配置

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: regex-file-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-file
  logtailConfig:
    # 设置采集的数据源类型。采集文本日志时，需设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: regex-file-example
    inputDetail:
      # 指定通过完整正则模式采集文本日志。
      logType: common_reg_log
      # 设置日志文件的路径。
      logPath: /app/logs
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: error.LOG
      # 设置用于匹配日志行首的行首正则表达式。
      logBeginRegex: '\[\d+-\d+-\w+:\d+:\d+,\d+]\s\[\w+]\s.*'
      # 设置正则表达式，用于提取日志内容。
      regex: '\[([^]]+)]\s\[(\w+)]\s\[(\w+)]\s\[([^:]+):(\d+)]\s(.*)'
      # 设置提取的字段列表。
      key : ["time", "level", "method", "file", "line", "message"]
      # 使用完整正则模式采集日志时，默认从日志的time字段中提取时间。如果无需提取时间，可不设置该字段。如果您设置了timeFormat字段，则需配置adjustTimezone字段和logTimezone字段。
      timeFormat: '%Y-%m-%dT%H:%M:%S'
      # 由于Logtail默认工作在零时区，因此需通过如下配置，强制设置时区。
      adjustTimezone: true
      # 设置时区偏移量。日志时间为东八区，如果是其他时区，请调整该值。
      logTimezone: "GMT+08:00"
      # 解析失败时，上传原始日志。
      discardUnmatch: false
      # 采集容器的文本日志时，需设置dockerFile为true。
      dockerFile: true
      # 设置环境变量白名单。只采集环境变量中包含EnvKey为ALIYUN_LOGTAIL_USER_DEFINED_ID的容器的文本日志。
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

- 采集到的日志样例

示例3：通过分隔符模式采集容器文本日志

如果您要采集的容器文本日志中有明确的分隔符，您可以使用通过分隔符模式采集容器文本日志。分隔符日志以换行符为边界，每一行都是一条日志。分隔符日志使用分隔符将一条日志分割成多个字段。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: delimiter-file-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-file
  logtailConfig:
    # 设置采集的数据源类型。采集文本日志时，需设置为file。
    inputType: file
    configName: delimiter-file-example
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    inputDetail:
      # 指定通过分隔符模式采集日志。
      logType: delimiter_log
      # 设置日志文件的路径。
      logPath: /usr/local/ilogtail
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: delimiter_log.LOG
      # 设置分隔符。
      separator: '|&|'
      # 设置提取的字段列表。
      key : ['time', 'level', 'method', 'file', 'line', 'message']
      # 设置时间字段。
      timeKey: 'time'
      # 使用分隔符模式采集日志时，默认从日志的time字段中提取时间。如果无需提取时间，可不设置该字段。如果您设置了timeFormat字段，则需配置a
djustTimezone字段和logTimezone字段。
      timeFormat: '%Y-%m-%dT%H:%M:%S'
      # 由于Logtail默认工作在零时区，因此需通过如下配置，强制设置时区。
      adjustTimezone: true
      # 设置时区偏移量。日志时间为东八区，如果是其他时区，请调整该值。
      logTimezone: "GMT+08:00"
      # 解析失败时，上传原始日志。
      discardUnmatch: false
      # 采集容器的文本日志时，需设置dockerFile为true。
      dockerFile: true
      # 设置环境变量白名单。只采集环境变量中包含EnvKey为ALIYUN_LOGTAIL_USER_DEFINED_ID的容器的文本日志。
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ''
```

示例4：通过JSON模式采集容器文本日志

如果您要采集的容器文本日志为Object类型的JSON日志，则您可以使用JSON模式进行采集。

- 原始日志

```
{"url": "POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=U0Ujpek********&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A
30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bmkd6x7hAgQ7b1c%3D HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sd
k-java", "request": {"status": "200", "latency": "18204"}, "time": "05/Jan/2020:13:30:28"}
```

- CRD配置

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: json-file-example
spec:
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-file
  logtailConfig:
    # 设置采集的数据源类型。采集文本日志时，需设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: json-file-example
    inputDetail:
      # 指定通过JSON模式采集日志。
      logType: json_log
      # 设置日志文件的路径。
      logPath: /usr/local/ilogtail
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: json_log.LOG
      # 设置时间字段，如果无指定需求，则设置为timeKey: ''。
      timeKey: 'time'
      # 设置时间格式。如果无指定需求，则设置为timeFormat: ''。
      timeFormat: '%Y-%m-%dT%H:%M:%S'
      # 采集容器的文本日志时，需设置dockerFile为true。
      dockerFile: true
      # 设置环境变量白名单。只采集环境变量中包含EnvKey为ALIYUN_LOGTAIL_USER_DEFINED_ID的容器的文本日志。
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

示例5：通过Kubernetes信息过滤容器，采集对应容器的文本日志

采集指定容器中*/data/logs/app_1*目录下*simple.LOG*文件中的日志。过滤容器的条件如下所示：

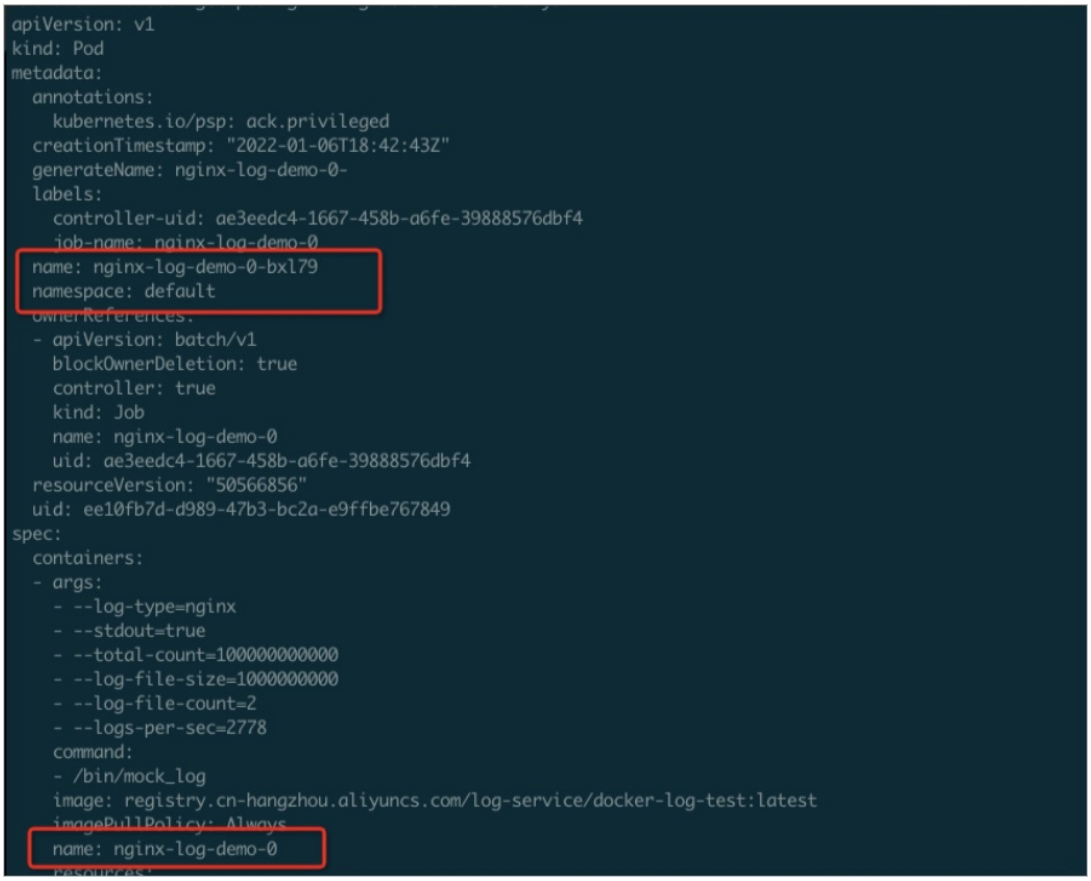
- default命名空间。
- 名称以nginx-log-demo开头的Pod。
- 名称为nginx-log-demo-0的容器。
- Kubernetes Label中Key为job-name，Value以nginx-log-demo为开头。

> **说明** Logtail 1.0.34及以上版本支持通过Kubernetes Namespace名称、Pod名称、容器名称或Kubernetes Label过滤日志。

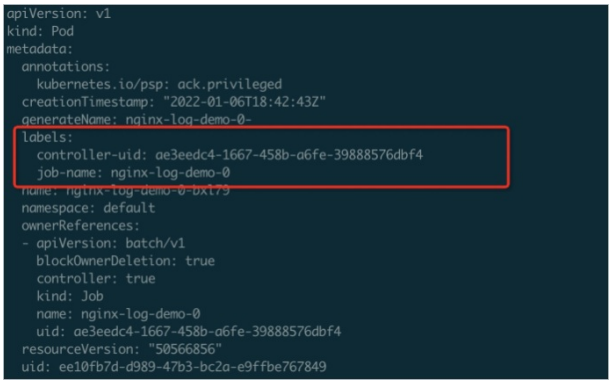
1. 获取Kubernetes层级的信息。
   i. 获取Pod信息。

```
~/.kube » kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-log-demo-0-bxl79  1/1     Running   0          48d
nginx-log-demo-1-qmrqk  1/1     Running   0          48d
nginx-log-demo-2-7khv9  1/1     Running   0          48d
nginx-log-demo-3-j24xc  1/1     Running   0          48d
```

ii. 获取Namespace等信息。



iii. 获取Kubernetes Label。



2. 创建Logtail采集配置

配置示例如下所示。更多信息，请参见通过DaemonSet-控制台方式采集容器标准输出。

> **说明**
> IncludeK8sLabel、ExcludeK8sLabel、K8sNamespaceRegex、K8sPodRegex、K8sContainerRegex、ExternalEnvTag、ExternalEnvTag等Kubernetes信息相关的参数，必须配置在advanced参数的k8s参数下。参数说明，请参见通过DaemonSet-控制台方式采集容器标准输出。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: simple-file-example
spec:
   # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: k8s-file
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型。采集文本日志时，需设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: simple-file-example
    inputDetail:
      # 指定通过极简模式采集文本日志。
      logType: common_reg_log
      # 设置日志文件所在路径。
      logPath: /data/logs/app_1
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: simple.LOG
      # 采集容器的文本日志时，需设置dockerFile为true。
      dockerFile: true
      #设置容器过滤条件。
      advanced:
        k8s:
          K8sNamespaceRegex: ^(default)$
          K8sPodRegex: '^(nginx-log-demo.*)$'
          K8sContainerRegex: ^(nginx-log-demo-0)$
          IncludeK8sLabel:
            job-name: "^(nginx-log-demo.*)$"
```

3.6.6. 通过Sidecar-CRD方式采集容器文本日志

本文介绍如何安装Sidecar及使用CRD方式创建Logtail配置，完成容器文本日志的采集。

前提条件

已安装Logtail组件。更多信息，请参见安装Logtail组件。

背景信息

通过Sidecar模式采集日志，依赖于Logtail容器和业务容器共享的日志目录。业务容器将日志写入到共享目录中，Logtail通过监控共享目录中日志文件的变化并采集日志。更多信息，请参见Sidecar日志采集介绍和Sidecar模式示例。

步骤一：安装Sidecar

1. 登录您的Kubernetes集群。
2. 创建一个YAML文件。

   *sidecar.yaml*为文件名，请根据实际情况替换。

   ```
   vim sidecar.yaml
   ```

3. 在YAML文件输入如下脚本，并根据实际情况设置其中的参数。

   > 注意 请确保配置文件中的*env*中的*TZ（Time Zone）*配置正确，否则原始日志与处理日志的时区不一致，可能会导致采集到的日志写入过去或未来的情况。例如如果是中国大陆，您可以设置时区为Asia/Shanghai。

   ```
   apiVersion: batch/v1
   kind: Job
   metadata:
     name: nginx-log-sidecar-demo
     namespace: default
   spec:
     template:
       metadata:
         name: nginx-log-sidecar-demo
       spec:
         restartPolicy: Never
         containers:
   ```

```
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-cou
nt=1000000000", "--logs-per-sec=100"]
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### logtail sidecar container
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        # when recevie sigterm, logtail will delay 10 seconds and then stop
        command:
        - sh
        - -c
        - /usr/local/ilogtail/run_logtail.sh 10
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        resources:
          limits:
            memory: 512Mi
          requests:
            cpu: 10m
            memory: 30Mi
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "${your_aliyun_user_id}"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "${your_machine_group_user_defined_id}"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
          ##### env tags config
          - name: "ALIYUN_LOG_ENV_TAGS"
            value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
          - name: "_pod_name_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: "_pod_ip_"
            valueFrom:
              fieldRef:
                fieldPath: status.podIP
          - name: "_namespace_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.namespace
          - name: "_node_name_"
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: "_node_ip_"
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### share this volume
      volumes:
      - name: nginx-log
        emptyDir: {}
```

i. 在配置脚本中找到如下内容，完成基础配置。

```
##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "${your_aliyun_user_id}"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "${your_machine_group_user_defined_id}"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
```

变量	说明
${your_aliyun_user_id}	配置为您的阿里云账号ID。更多信息，请参见步骤一：获取日志服务所在的阿里云账号ID。
${your_machine_group_user_defined_id}	配置机器组的自定义标识，请确保该标识在您的Project所在地域内唯一，例如nginx-log-sidecar。更多信息，请参见创建用户自定义标识机器组。
${your_region_config}	请根据日志服务Project所在地域和访问的网络类型填写。其中，地域信息请参见Logtail安装参数。 ■ 如果使用公网采集日志，格式为 `region-internet` ，例如：**华东 1（杭州）**为**cn-hangzhou-internet**。 ■ 如果使用阿里云内网采集日志，格式为 `region` 。例如：**华东 1（杭州）**为**cn-hangzhou**。

ii. 在配置脚本中找到如下内容，设置挂载路径。

> **说明** 建议使用emptyDir挂载方式。

```
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### share this volume
      volumes:
      - name: nginx-log
        emptyDir: {}
```

参数	说明
name	自定义设置卷的名称。 **注意** volumeMounts节点下的name参数与volumes节点下的name参数需设置为一致，即确保Logtail容器和业务容器挂载相同的卷上。
mountPath	设置挂载路径，即容器文本日志所在文件的路径。

iii. 在配置脚本中找到如下内容，设置延迟停止采集的时间。

通常情况下，延迟停止采集的时间为10秒，即Logtail容器在接收到外部停止信号后会等待10秒再退出，防止有部分数据没有采集完毕。

```
command:
- sh
- -c
- /usr/local/ilogtail/run_logtail.sh 10
```

4. 执行如下命令使*sidecar.yaml*文件配置生效。

*sidecar.yaml*为文件名，请根据实际情况替换。

```
kubectl apply -f sidecar.yaml
```

步骤二：创建Logtail采集配置

只需要定义AliyunLogConfig CRD即可创建Logtail采集配置。创建完成后，系统自动应用该Logtail采集配置。如果您要删除Logtail采集配置只需删除对应的CRD资源即可。

1. 登录Kubernetes集群。

2. 执行如下命令创建一个YAML文件。

   *cube.yaml*为文件名，请根据实际情况替换。

```
vim cube.yaml
```

3. 在YAML文件输入如下脚本，并根据实际情况设置其中的参数。

> **注意**
> - 请确保configName参数值在日志服务Project中唯一存在。
> - 如果多个CRD关联同一个Logtail采集配置，则删除或修改任意一个CRD均会影响到该Logtail采集配置，导致其他关联该Logtail采集配置的CRD状态与服务端不一致。
> - 通过Sidecar模式只能采集文本日志，需将dockerFile参数设置为false。

```
apiVersion: log.alibabacloud.com/v1alpha1       # 使用默认值，无需修改。
kind: AliyunLogConfig                           # 使用默认值，无需修改。
metadata:
  name: simple-stdout-example                   # 设置资源名，在当前Kubernetes集群内唯一。
spec:
  project: k8s-my-project                       # （可选）设置Project名称。默认值为安装Logtail组件时设置的Project。
  logstore: k8s-stdout                          # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  machineGroups:                                # 设置机器组的名称，需与您在安装Sidecar时设置的${your_machine_group_user_defined_id}的值一致。Sidecar与CRD通过此处配置的机器组建立关联。
  - nginx-log-sidecar
  shardCount: 2                                 # （可选）设置Shard数量。默认值为2，取值范围1~10。
  lifeCycle: 90                                 # （可选）设置Logstore中数据的存储时间。默认值为90，取值范围为1~3650。其中，3650天为永久存储。
  logtailConfig:                                # 设置Logtail采集配置。
    inputType: file                             # 设置采集的数据源类型，通过Sidecar-CRD方式只支持采集文本日志，即需要设置为file。
    configName: simple-stdout-example           # 设置Logtail配置名称，与资源名（metadata.name）保持一致。
    inputDetail:                                # 设置Logtail采集配置的详细信息，具体配置请参见本文下方的示例。
    ...
```

参数	数据类型	是否必填	说明
project	string	否	Project名称。默认值为安装Logtail组件时所设置的Project。
logstore	string	是	Logstore名称。 如果您所指定的Logstore不存在，日志服务会自动创建。
shardCount	int	否	Shard数量。默认值为2，取值范围为1~10。
lifeCycle	int	否	Logstore中数据的存储时间。默认值为90，取值范围为1~3650。其中，3650天为永久存储。

machineGroups	array	是	设置机器组名称，需与您在安装Sidecar时设置的*${your_machine_group_user_defined_id}*的值一致，例如nginx-log-sidecar。
			日志服务将根据您设置的名称自动创建机器组，建立Sidecar与CRD之间的关联。
			注意
			机器组自定义标识的配置格式如下所示，请严格按照此格式配置。
			`machineGroups:`
			`- nginx-log-sidecar`

logtailConfig	object	是	Logtail采集配置的详细定义，一般只需要定义其中的inputType参数、configName参数和inputDetail参数。详细参数说明，请参见Logtail配置。 相关示例，请参见配置示例（单目录）、配置示例（多目录）。

4. 执行如下命令使Logtail采集配置生效。

*cube.yaml*为文件名，请根据实际情况替换。

```
kubectl apply -f cube.yaml
```

创建Logtail采集配置后，您可以通过CRD方式或控制台方式查看Logtail采集配置。具体操作，请参见查看Logtail采集配置。

配置示例（单目录）

通过Sidecar-CRD方式采集本地IDC上自建Kubernetes集群中nginx-log-demo容器的文本日志（包括Nginx访问日志和Nginx错误日志）。基本信息如下：

- 日志服务所在地域为华东1（杭州），需要通过公网采集。
- 待挂载的卷的名称为nginx-log，挂载方式为emptyDir，将nginx-log卷分别挂载到nginx-log-demo容器和Logtail容器的*/var/log/nginx*目录下。
- Nginx访问日志所在文件的路径为*/var/log/nginx/access.log*，用于存储Nginx访问日志的目标Logstore为nginx-access。
- Nginx错误日志所在文件的路径为*/var/log/nginx/error.log*，用于存储Nginx错误日志的目标Logstore为nginx-error。
- Sidecar示例

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx-log-sidecar-demo
  namespace: default
spec:
  template:
    metadata:
      name: nginx-log-sidecar-demo
    spec:
      restartPolicy: Never
      containers:
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count
=1000000000", "--logs-per-sec=100"]
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### logtail sidecar container
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        # when recevie sigterm, logtail will delay 10 seconds and then stop
        command:
        - sh
        - -c
        - /usr/local/ilogtail/run_logtail.sh 10
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "1023****3423"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "nginx-log-sidecar"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/cn-hangzhou-internet/ilogtail_config.json"
          ##### env tags config
          - name: "ALIYUN_LOG_ENV_TAGS"
            value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
          - name: "_pod_name_"
            valueFrom:
              fieldRef:
```

```
            fieldRef:
              fieldPath: metadata.name
        - name: "_pod_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.podIP
        - name: "_namespace_"
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: "_node_name_"
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        - name: "_node_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.hostIP
      volumeMounts:
      - name: nginx-log
        mountPath: /var/log/nginx
    ##### share this volume
    volumes:
    - name: nginx-log
      emptyDir: {}
```

- CRD示例

  创建两个Logtail采集配置用于采集Nginx访问日志和Nginx错误日志。

  - 采集Nginx访问日志

    > **注意** Sidecar模式下，需将dockerFile参数设置为false。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 资源名称，在您的K8s集群中必须唯一。
  name: nginx-log-access-example
spec:
  # 设置Project名称。默认值为安装Logtail时所设置的Project。
  project: k8s-nginx-sidecar-demo
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: nginx-access
  # 设置机器组名称，需与您在安装Sidecar时设置的${your_machine_group_user_defined_id}的值一致。
  machineGroups:
  - nginx-log-sidecar
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型，通过Sidecar-CRD方式只支持采集文本日志，即需要设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: nginx-log-access-example
    inputDetail:
      # 指定通过完整正则模式采集容器文本日志。
      logType: common_reg_log
      # 设置日志文件所在路径。
      logPath: /var/log/nginx
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: access.log
      # sidecar模式下，需设置dockerFile为false。
      dockerFile: false
      # 设置用于匹配日志行首的行首正则表达式。如果为单行模式，设置成'.*'。
      logBeginRegex: '.*'
      # 设置正则表达式，用于提取日志内容。请根据实际情况设置。
      regex: '(\S+)\s(\S+)\s\S+\s\S+\s"(\S+)\s(\S+)\s+([^"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s"([^"]+)"\s.*'
      # 设置提取的字段列表。
      key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size",user-agent"]
```

- 采集Nginx错误日志

  > 注意 Sidecar模式下，需将dockerFile参数设置为false。

```
# config for error log
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: nginx-log-error-example
spec:
  # 设置Project名称。默认值为安装Logtail时所设置的Project。
  project: k8s-nginx-sidecar-demo
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: nginx-error
  # 设置机器组名称，需与您在安装Sidecar时设置的${your_machine_group_user_defined_id}的值一致。
  machineGroups:
  - nginx-log-sidecar
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型，通过Sidecar-CRD方式只支持采集文本日志，即需要设置为file。
    inputType: file
    # 设置Logtail配置名称，必须与资源名(metadata.name)相同。
    configName: nginx-log-error-example
    inputDetail:
      # 指定通过完整正则模式采集容器文本日志。
      logType: common_reg_log
      # 设置日志文件的路径。
      logPath: /var/log/nginx
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: error.log
      # sidecar模式下，需设置dockerFile为false。
      dockerFile: false
```

配置示例（多目录）

通过Sidecar-CRD方式采集本地IDC上自建Kubernetes集群中nginx-log-demo容器的文本日志（存储在不同目录下的Nginx访问日志）。基本信息如下：

- 日志服务所在地域为华东1（杭州），需要通过公网采集。
- 待挂载的卷的名称为nginx-log和nginx-logs，挂载方式为emptyDir。将nginx-log卷分别挂载到nginx-log-demo容器和Logtail容器的*/var/log/nginx*目录下。将nginx-logs卷分别挂载到nginx-log-demo容器和Logtail容器的*/var/log/nginxs*目录下。
- 一个日志文件的路径为*/var/log/nginx/access.log*，另一个日志文件的路径为*/var/log/nginxs/access.log*。
- 用于存储Nginx访问日志的目标Logstore为nginx-access。
- Sidecar示例

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx-log-sidecar-demo
  namespace: default
spec:
  template:
    metadata:
      name: nginx-log-sidecar-demo
    spec:
      restartPolicy: Never
      containers:
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-count
=1000000000", "--logs-per-sec=100"]
        lifecycle:
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
        - name: nginx-logs
          mountPath: /var/log/nginxs
      ##### logtail sidecar container
```

```
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        # when recevie sigterm, logtail will delay 10 seconds and then stop
        lifecycle:
        command:
        - sh
        - -c
        - /usr/local/ilogtail/run_logtail.sh 10
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        resources:
          limits:
            memory: 512Mi
          requests:
            cpu: 10m
            memory: 30Mi
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "1023****3423"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "nginx-log-sidecar"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/cn-hangzhou-internet/ilogtail_config.json"
          ##### env tags config
          - name: "ALIYUN_LOG_ENV_TAGS"
            value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
          - name: "_pod_name_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: "_pod_ip_"
            valueFrom:
              fieldRef:
                fieldPath: status.podIP
          - name: "_namespace_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.namespace
          - name: "_node_name_"
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: "_node_ip_"
            valueFrom:
              fieldRef:
                fieldPath: status.hostIP
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
        - name: nginx-logs
          mountPath: /var/log/nginxs
      ##### share this volume
      volumes:
      - name: nginx-log
        emptyDir: {}
      - name: nginx-logs
        emptyDir: {}
```

- CRD示例

  创建两个Logtail采集配置用于采集不同目录下的Nginx访问日志。

- 采集*/var/log/nginx/access.log*下的Nginx访问日志

  > **注意** Sidecar模式下，需将dockerFile参数设置为false。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: nginx-log-access-example
spec:
  # 设置Project名称。默认值为安装Logtail时所设置的Project。
  project: k8s-nginx-sidecar-demo
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: nginx-access
  # 设置机器组名称，需与您在安装Sidecar时设置的${your_machine_group_user_defined_id}的值一致。
  machineGroups:
  - nginx-log-sidecar
  # 设置Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型，通过Sidecar-CRD方式只支持采集文本日志，即需要设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: nginx-log-access-example
    inputDetail:
      # 指定通过完整正则模式采集文本日志。
      logType: common_reg_log
      # 设置日志文件的路径。
      logPath: /var/log/nginx
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: access.log
      # sidecar模式下，需设置dockerFile为false。
      dockerFile: false
      # 设置用于匹配日志行首的行首正则表达式。如果为单行模式，设置成 '.*'。
      logBeginRegex: '.*'
      # 设置正则表达式，用于提取日志内容。
      regex: '(\S+)\s(\S+)\s\S+\s\S+\s"(\S+)\s(\S+)\s+([^"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s"([^"]+)"\s.*'
      # 提取的字段列表。
      key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size",user-agent"]
```

- 采集*/var/log/nginxs/access.log*下的Nginx访问日志

  > **注意** Sidecar模式下，需将dockerFile参数设置为false。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # 设置资源名，在当前Kubernetes集群内唯一。
  name: nginxs-log-access-example
spec:
  # 设置Project名称。默认值为安装Logtail时所设置的Project。
  project: k8s-nginx-sidecar-demo
  # 设置Logstore名称。如果您所指定的Logstore不存在，日志服务会自动创建。
  logstore: nginxs-access
  # 设置机器组名称，需与您在安装Sidecar时设置的${your_machine_group_user_defined_id}的值一致。
  machineGroups:
  - nginx-log-sidecar
  # Logtail采集配置。
  logtailConfig:
    # 设置采集的数据源类型，通过Sidecar-CRD方式只支持采集文本日志，即需要设置为file。
    inputType: file
    # 设置Logtail采集配置的名称，必须与资源名(metadata.name)相同。
    configName: nginxs-log-access-example
    inputDetail:
      # 指定通过完整正则模式采集文本日志。
      logType: common_reg_log
      # 设置日志文件的路径。
      logPath: /var/log/nginxs
      # 设置日志文件的名称。支持通配符星号（*）和半角问号（?），例如log_*.log。
      filePattern: access.log
      # 在sidecar模式下，需设置dockerFile为false。
      dockerFile: false
      # 设置用于匹配日志行首的行首正则表达式。如果为单行模式，设置成 .*。
      logBeginRegex: '.*'
      # 设置正则表达式，用于提取日志内容。
      regex: '(\S+)\s(\S+)\s\S+\s\S+\s"(\S+)\s(\S+)\s+([^"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d+)\s(\S+)\s"([^"]+)"\s.*'
      # 提取的字段列表。
      key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "response-size",user-agent"]
# config for error log
```

3.6.7. 通过Sidecar-控制台方式采集容器文本日志

本文介绍如何安装Sidecar及使用控制台方式创建Logtail采集配置，完成容器文本日志的采集。

前提条件

已安装Logtail组件。更多信息，请参见安装Logtail组件。

背景信息

通过Sidecar模式采集日志，依赖于Logtail容器和业务容器共享的日志目录。业务容器将日志写入到共享目录中，Logtail通过监控共享目录中日志文件的变化并采集日志。更多信息，请参见Sidecar日志采集介绍和Sidecar模式示例。

步骤一：安装Sidecar

1. 登录您的Kubernetes集群。
2. 创建一个YAML文件。

   *sidecar.yaml*为文件名，请根据实际情况替换。

```
vim sidecar.yaml
```

3. 在YAML文件输入如下脚本，并根据实际情况设置其中的参数。

   > **注意** 请确保配置文件中的*env*中的*TZ（Time Zone）*配置正确，否则原始日志与处理日志的时区不一致，可能会导致采集到的日志写入过去或未来的情况。例如如果是中国大陆，您可以设置时区为Asia/Shanghai。

```
apiVersion: batch/v1
kind: Job
metadata:
```

```
  name: nginx-log-sidecar-demo
  namespace: default
spec:
  template:
    metadata:
      name: nginx-log-sidecar-demo
    spec:
      restartPolicy: Never
      containers:
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/access.log", "--total-cou
nt=1000000000", "--logs-per-sec=100"]
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### logtail sidecar container
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        # when recevie sigterm, logtail will delay 10 seconds and then stop
        command:
        - sh
        - -c
        - /usr/local/ilogtail/run_logtail.sh 10
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        resources:
          limits:
            memory: 512Mi
          requests:
            cpu: 10m
            memory: 30Mi
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "${your_aliyun_user_id}"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "${your_machine_group_user_defined_id}"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
          ##### env tags config
          - name: "ALIYUN_LOG_ENV_TAGS"
            value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
          - name: "_pod_name_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
          - name: "_pod_ip_"
            valueFrom:
              fieldRef:
                fieldPath: status.podIP
          - name: "_namespace_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.namespace
          - name: "_node_name_"
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: "_node_ip_"
            valueFrom:
```

```
            fieldRef:
              fieldPath: status.hostIP
      volumeMounts:
      - name: nginx-log
        mountPath: /var/log/nginx
    ##### share this volume
    volumes:
    - name: nginx-log
      emptyDir: {}
```

i. 在配置脚本中找到如下内容，完成基础配置。

```
##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "${your_aliyun_user_id}"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "${your_machine_group_user_defined_id}"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
```

变量	说明
${your_aliyun_user_id}	配置为您的阿里云账号ID。更多信息，请参见步骤一：获取日志服务所在的阿里云账号ID。
${your_machine_group_user_defined_id}	配置机器组的自定义标识，请确保该标识在您的Project所在地域内唯一，例如nginx-log-sidecar。更多信息，请参见创建用户自定义标识机器组。
${your_region_config}	请根据日志服务Project所在地域和访问的网络类型填写。其中，地域信息请参见Logtail安装参数。 ■ 如果使用公网采集日志，格式为 `region-internet` ，例如：**华东 1（杭州）**为**cn-hangzhou-internet**。 ■ 如果使用阿里云内网采集日志，格式为 `region` 。例如：**华东 1（杭州）**为**cn-hangzhou**。

ii. 在配置脚本中找到如下内容，设置挂载路径。

> **说明** 建议使用emptyDir挂载方式。

```
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### share this volume
      volumes:
      - name: nginx-log
        emptyDir: {}
```

参数	说明
name	自定义设置卷的名称。 **注意** volumeMounts节点下的name参数与volumes节点下的name参数需设置为一致，即确保Logtail容器和业务容器挂载相同的卷上。
mountPath	设置挂载路径，即容器文本日志所在文件的路径。

iii. 在配置脚本中找到如下内容，设置延迟停止采集的时间。

通常情况下，延迟停止采集的时间为10秒，即Logtail容器在接收到外部停止信号后会等待10秒再退出，防止有部分数据没有采集完毕。

```
command:
- sh
- -c
- /usr/local/ilogtail/run_logtail.sh 10
```

4. 执行如下命令使*sidecar.yaml*文件配置生效。

*sidecar.yaml*为文件名，请根据实际情况替换。

```
kubectl apply -f sidecar.yaml
```

步骤二：创建机器组

1. 登录日志服务控制台。
2. 在Project列表中，单击您在安装Logtail组件时所使用的Project。
3. 在左侧导航栏中，选择**资源 > 机器组**。
4. 在**机器组**列表中，选择**图标 > 创建**。
5. 在**创建机器组**面板中，配置如下信息，然后单击**确定**。

参数	说明
名称	机器组名称。 **注意** 创建后，不支持修改机器组名称，请谨慎填写。
机器组标识	选择**用户自定义标识**。
机器组Topic	机器组Topic用于区分不同服务器产生的日志数据。更多信息，请参见日志主题。
用户自定义标识	配置为您在安装Sidecar时配置的用户自定义标识，即需与您在安装Sidecar时设置的*${your_machine_group_user_defined_id}*的值一致，例如nginx-log-sidecar。

步骤三：创建Logtail采集配置

1. 登录日志服务控制台。
2. 在**接入数据**区域，单击**正则-文本日志**。

   本文以采集正则-文本文件为例，其他文本文件采集请参见采集文本日志。
3. 选择目标Project和Logstore，单击**下一步**。

   选择您在安装Logtail组件时所使用的Project，Logstore为您自定义创建的Logstore。
4. 单击**使用现有机器组**。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   该机器组为您在步骤二：创建机器组中创建的机器组。

   注意 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为**FAIL**，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 创建Logtail采集配置，单击**下一步**。

   目前支持通过极简模式、Nginx模式、分隔符模式、JSON模式、完整正则模式采集日志。具体操作，请参见采集文本日志。

   注意 sidecar模式下，请勿打开**是否为Docker文件**开关。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

3.6.8. 采集标准Docker容器日志

本文介绍如何部署Logtail容器及创建Logtail配置，采集标准Docker容器日志。

步骤一：部署Logtail容器

1. 拉取Logtail镜像。

   ```
   docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail
   ```

   其中，*registry.cn-hangzhou.aliyuncs.com*需根据实际情况替换，地域信息请参见Logtail安装参数。如果您的服务器处于阿里云VPC网络中，需将registry修改为 registry-vpc。

2. 启动Logtail容器。

   > **说明** 请在配置参数前执行以下任意一种配置，否则删除其他container时可能出现错误 `container text file busy` 。
   >
   > - Centos 7.4及以上版本（除Centos 8.0以外）设置fs.may_detach_mounts=1。更多信息，请参见Bug 1468249、Bug 1441737和issue 34538。
   > - 为Logtail授予 `privileged` 权限，启动参数中添加 `--privileged` 。更多信息，请参见docker run命令。

   根据实际情况替换模板中的3个参数 `${your_region_name}` 、 `${your_aliyun_user_id}` 和 `${your_machine_group_user_defined_id}` 。

   ```
   docker run -d -v /:/logtail_host:ro -v /var/run:/var/run --env ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/${your_region_name}/ilogtail_config.json --env ALIYUN_LOGTAIL_USER_ID=${your_aliyun_user_id} --env ALIYUN_LOGTAIL_USER_DEFINED_ID=${your_machine_group_user_defined_id} registry.cn-hangzhou.aliyuncs.com/log-service/logtail
   ```

参数	参数说明
`${your_region_name}`	请根据日志服务Project所在地及网络类型填写。其中，地域信息请参见Logtail安装参数。 • 如果为公网，格式为 `region-internet` ，例如：**华东 1（杭州）**为cn-hangzhou-internet。 • 如果为阿里云内网，格式为 `region` 。例如：**华东 1（杭州）**为cn-hangzhou。

参数	参数说明
`${your_aliyun_user_id}`	您的阿里云主账号ID。更多信息，请参见配置用户标识。
`${your_machine_group_user_defined_id}`	您机器组的自定义标识，请确保该标识在您的Project所在地域内唯一。更多信息，请参见创建用户自定义标识机器组。

说明

您可以自定义配置Logtail容器的启动参数，只需保证以下前提条件。

i. 启动时，必须配置3个环境变量 `ALIYUN_LOGTAIL_USER_DEFINED_ID` 、 `ALIYUN_LOGTAIL_USER_ID` 、 `ALIYUN_LOGTAIL_CONFIG` 。

ii. 必须宿主机将*/var/run*挂载到Logtail容器的*/var/run*目录。

iii. 将宿主机根目录挂载到Logtail容器的 `/logtail_host` 目录。

iv. 如果Logtail日志*/usr/local/ilogtail/ilogtail.LOG*中出现 `The parameter is invalid : uuid=none` 的错误日志，请在宿主机上创建一个*product_uuid*文件，在其中输入任意合法UUID（例如 `169E98C9-ABC0-4A92-B1D2-AA6239C0D261` ），并把该文件挂载到Logtail容器的*/sys/class/dmi/id/product_uuid*目录。

步骤二：创建采集配置

请根据您的需求在控制台上创建采集配置。

- 如果您需要采集Docker文件，操作步骤与采集Kubernetes文件类似。更多信息，请参见通过DaemonSet-控制台方式采集文本文件。
- 如果您需要采集Docker标准输出，操作步骤与采集Kubernetes标准输出类似。更多信息，请参见通过DaemonSet-控制台方式采集标准输出。
- 如果您需要采集宿主机文本文件。更多信息，请参见采集宿主机文本文件。

  默认将宿主机根目录挂载到Logtail容器的*/logtail_host*目录。配置路径时，您需要加上此前缀。例如需要采集宿主机上*/home/logs/app_log/*目录下的日志，配置页面中日志路径设置为*/logtail_host/home/logs/app_log/*。

其中，在创建机器组时，请在**用户自定义标识**中输入步骤一：部署Logtail容器时配置的 `ALIYUN_LOGTAIL_USER_DEFINED_ID` 。

默认字段

- Docker标准输出

  每条日志默认上传字段如下所示。

字段名	说明
time	数据上传时间，例如： `2018-02-02T02:18:41.979147844Z`
source	输入源类型，stdout或stderr
_image_name_	镜像名
_container_name_	容器名
_container_ip_	容器IP地址

- Docker文件

  默认每条日志上传的字段如下所示。

字段名	说明
_image_name_	镜像名
_container_name_	容器名
_container_ip_	容器IP地址

其他操作

- 查看Logtail运行状态。

  您可以执行 `docker exec ${logtail_container_id} /etc/init.d/ilogtaild status` 命令查看Logtail运行状态。

- 查看Logtail的版本号、IP地址和启动时间等信息。

  您可以执行 `docker exec ${logtail_container_id} cat /usr/local/ilogtail/app_info.json` 命令查看Logtail相关信息。

- 查看Logtail的运行日志。

  Logtail运行日志保存在*/usr/local/ilogtail/*目录下，文件名为*ilogtail.LOG*，轮转文件会压缩存储为*ilogtail.LOG.x.gz*。示例如下：

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 tail -n 5 /usr/local/ilogtail/ilogtail.LOG
[2018-02-06 08:13:35.721864]    [INFO]    [8]    [build/release64/sls/ilogtail/LogtailPlugin.cpp:104]    logtail plugin
Resume:start
[2018-02-06 08:13:35.722135]    [INFO]    [8]    [build/release64/sls/ilogtail/LogtailPlugin.cpp:106]    logtail plugin
Resume:success
[2018-02-06 08:13:35.722149]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:369]    start add exi
sted check point events, size:0
[2018-02-06 08:13:35.722155]    [INFO]    [8]    [build/release64/sls/ilogtail/EventDispatcher.cpp:511]    add existed c
heck point events, size:0    cache size:0    event size:0    success count:0
[2018-02-06 08:13:39.725417]    [INFO]    [8]    [build/release64/sls/ilogtail/ConfigManager.cpp:3776]    check containe
r path update flag:0    size:1
```

  容器stdout并不具备参考意义，请忽略以下stdout输出。

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: m
ust be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: m
ust be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: m
ust be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

- 重启Logtail。

  请参考以下示例重启Logtail。

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/ilogtaild start
ilogtail is running
```

3.6.9. 采集Kubernetes事件

本文档主要介绍如何使用eventer将Kubernetes中的事件采集到日志服务。

日志服务支持通过eventer方式和K8s事件中心采集Kubernetes中的事件。

- eventer方式

  Kubernetes事件采集相关源码请参见GitHub。

- K8s事件中心（推荐）

  日志服务还支持通过日志服务控制台的**K8s事件中心**采集Kubernetes中的事件并自动创建可视化报表和相关告警。更多信息，请参见创建并使用K8s事件中心。

创建采集配置

> **说明**
> - 如果您使用的是阿里云Kubernetes，请参见创建并使用K8s事件中心。
> - 如果您使用的是自建Kubernetes，需配置部署示例中的以下参数：endpoint、project、logStore、regionId、internal、accessKeyId和accessKeySecret。

部署示例如下所示。

```
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: kube-eventer
  name: kube-eventer
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app: kube-eventer
  template:
    metadata:
      labels:
        app: kube-eventer
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      dnsPolicy: ClusterFirstWithHostNet
      serviceAccount: kube-eventer
      containers:
        - image: registry.cn-hangzhou.aliyuncs.com/acs/kube-eventer:v1.2.5-cc7ec54-aliyun
          name: kube-eventer
          command:
            - "/kube-eventer"
            - "--source=kubernetes:https://kubernetes.default"
            ## .send to sls
            ## --sink=sls:https://{endpoint}?project={project}&logStore=k8s-event&regionId={region-id}&internal=false&acce
ssKeyId={accessKeyId}&accessKeySecret={accessKeySecret}
            - --sink=sls:https://cn-beijing.log.aliyuncs.com?project=k8s-xxxx&logStore=k8s-event&regionId=cn-beijing&inter
nal=false&accessKeyId=xxx&accessKeySecret=xxx
          env:
          # If TZ is assigned, set the TZ value as the time zone
          - name: TZ
            value: "Asia/Shanghai"
          volumeMounts:
            - name: localtime
              mountPath: /etc/localtime
              readOnly: true
            - name: zoneinfo
              mountPath: /usr/share/zoneinfo
              readOnly: true
          resources:
            requests:
              cpu: 10m
              memory: 50Mi
            limits:
              cpu: 500m
              memory: 250Mi
      volumes:
        - name: localtime
          hostPath:
            path: /etc/localtime
        - name: zoneinfo
          hostPath:
```

```
            path: /usr/share/zoneinfo
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: kube-eventer
rules:
  - apiGroups:
      - ""
    resources:
      - events
    verbs:
      - get
      - list
      - watch
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: kube-eventer
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: kube-eventer
subjects:
  - kind: ServiceAccount
    name: kube-eventer
    namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: kube-eventer
  namespace: kube-system
```

配置项	类型	是否必选	说明
endpoint	string	必选	日志服务的Endpoint。更多信息，请参见服务入口。
project	string	必选	日志服务的Project。
logStore	string	必选	日志服务的Logstore。
internal	string	自建Kubernetes：必选。	自建Kubernetes必须设置为false。
regionId	string	自建Kubernetes：必选。	日志服务所在地域ID。更多信息，请参见服务入口。
accessKeyId	string	自建Kubernetes：必选。	AccessKey ID，建议使用RAM用户的AccessKey信息。更多信息，请参见访问密钥。
accessKeySecret	string	自建Kubernetes：必选。	AccessKey Secret，建议使用RAM用户的AccessKey信息。更多信息，请参见访问密钥。

日志样例

采集到的日志样例如下所示。

```
hostname:  cn-hangzhou.i-***********"
level:  Normal
pod_id:  2a360760-****
pod_name:  logtail-ds-blkkr
event_id:  {
   "metadata":{
      "name":"logtail-ds-blkkr.157b7cc90de7e192",
      "namespace":"kube-system",
      "selfLink":"/api/v1/namespaces/kube-system/events/logtail-ds-blkkr.157b7cc90de7e192",
      "uid":"2aaf75ab-****",
      "resourceVersion":"6129169",
      "creationTimestamp":"2019-01-20T07:08:19Z"
   },
   "involvedObject":{
      "kind":"Pod",
      "namespace":"kube-system",
      "name":"logtail-ds-blkkr",
      "uid":"2a360760-****",
      "apiVersion":"v1",
      "resourceVersion":"6129161",
      "fieldPath":"spec.containers{logtail}"
   },
   "reason":"Started",
   "message":"Started container",
   "source":{
      "component":"kubelet",
      "host":"cn-hangzhou.i-***********"
   },
   "firstTimestamp":"2019-01-20T07:08:19Z",
   "lastTimestamp":"2019-01-20T07:08:19Z",
   "count":1,
   "type":"Normal",
   "eventTime":null,
   "reportingComponent":"",
   "reportingInstance":""
}
```

日志字段	类型	说明
hostname	string	事件发生所在的主机名。
level	string	日志等级，包括Normal、Warning。
pod_id	string	Pod的唯一标识，仅在该事件类型和Pod相关时才具有此字段。
pod_name	string	Pod名，仅在该事件类型和Pod相关时才具有此字段。
eventId	json	该字段为JSON类型的字符串，为事件的详细内容。

3.7. 使用Logtail插件采集数据

3.7.1. 概述

本文介绍通过Logtail插件采集数据的概念及相关配置流程。

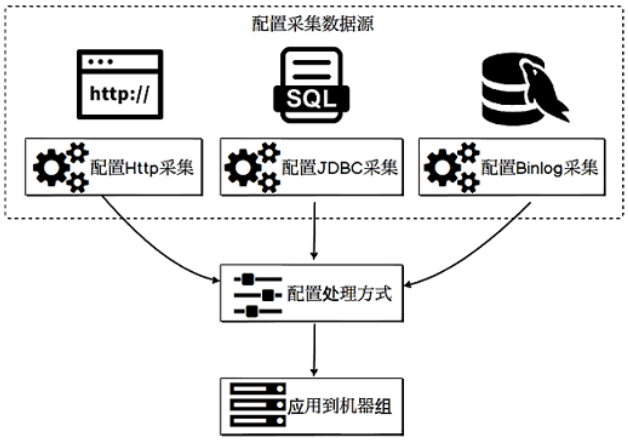
简介

> 注意 Logtail插件不支持内核版本低于2.6.32（不包含）的Linux系统。

Logtail具备灵活的插件扩展机制，支持定制采集插件实现各类数据的采集，例如：

- 通过Logtail插件采集HTTP数据，并将数据处理结果上传到日志服务，可以进行实时的服务可用性检测和持续的可用性监控。
- 通过Logtail插件采集MySQL查询结果，可以根据自增ID或时间等标志实现增量数据同步。
- 通过Logtail插件采集MySQL Binlog，可以增量订阅数据库的变更数据，并进行实时查询与分析。

配置流程



1. 配置采集方式。

   不同数据源具有不同的Logtail采集配置，请按照您的数据源类型设置对应的Logtail采集配置。

   - 采集MySQL Binlog
   - 采集MySQL查询结果
   - 采集HTTP数据
   - 采集容器标准输出
   - 采集Beats和Logstash数据源
   - 采集Syslog
   - 采集Windows事件日志
   - 采集Docker事件
   - 采集Systemd Journal日志

2. 配置处理方式。

   您可对一个数据源配置多个处理方式，Logtail会根据配置顺序逐一执行各个处理方式。各类数据源均支持所有类型的处理方式，详情请参见概述。

3. 应用到机器组。

   将采集配置和处理配置应用到指定的机器组，Logtail会自动拉取配置并开始采集。

3.7.2. 采集MySQL Binlog

本文介绍如何通过日志服务控制台创建Logtail采集配置来采集MySQL Binlog。

前提条件

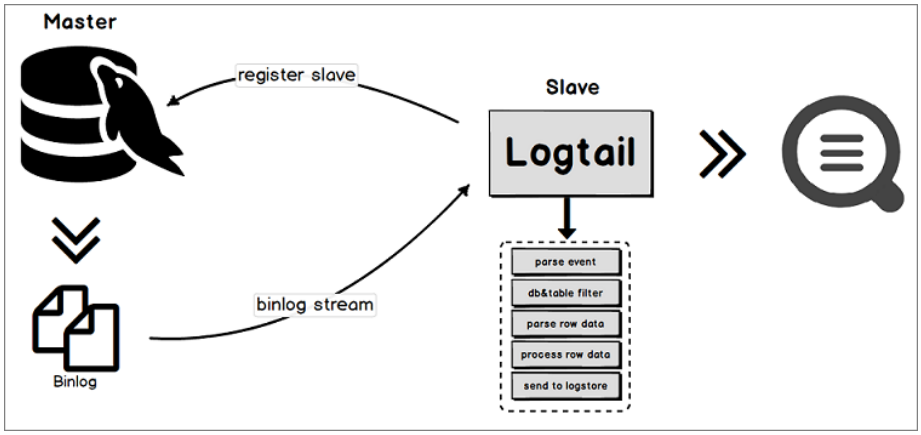
已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

> **说明** 目前支持Linux Logtail 0.16.0及以上版本，Window Logtail 1.0.0.8及以上版本。

原理

Logtail内部实现了MySQL Slave节点的交互协议，具体流程如下所示。

1. Logtail将自己伪装为MySQL Slave节点向MySQL master节点发送dump请求。
2. MySQL master节点收到dump请求后，会将自身的Binlog实时发送给Logtail。
3. Logtail对Binlog进行事件解析、过滤、数据解析等操作，并将解析好的数据上传到日志服务。



功能特点

- 通过Binlog增量采集数据库的更新操作数据，性能优越。支持RDS等MySQL协议的数据库。
- 支持多种数据库过滤方式。
- 支持设置Binlog位点。
- 支持通过Checkpoint机制同步保存状态。

使用限制

- Logtail 1.0.31及以上版本支持MySQL 8.0。
- MySQL必须开启Binlog，且Binlog必须为row模式（RDS默认已开启Binlog）。

```
# 查看是否开启Binlog
mysql> show variables like "log_bin";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_bin       | ON    |
+---------------+-------+
1 row in set (0.02 sec)
# 查看Binlog类型
mysql> show variables like "binlog_format";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| binlog_format | ROW   |
+---------------+-------+
1 row in set (0.03 sec)
```

- ServerID唯一，即需要同步的MySQL的Slave ID唯一。
- RDS限制
  - 无法直接在RDS服务器上安装Logtail，您需要将Logtail安装在能连通RDS实例的服务器上。
  - RDS备库不支持Binlog采集，您需要配置RDS主库进行采集。

应用场景

适用于数据量较大且性能要求较高的数据同步场景。

- 增量订阅数据库改动进行实时查询与分析。
- 数据库操作审计。
- 使用日志服务对数据库更新信息进行自定义查询分析、可视化、对接下游流计算、导入MaxCompute离线计算、导入OSS长期存储等操作。

注意事项

建议您适当放开对Logtail的资源限制以应对流量突增等情况，避免Logtail因为资源超限被强制重启，对您的数据造成不必要的风险。

您可以通过*/usr/local/ilogtail/ilogtail_config.json*文件修改相关参数。更多信息，请参见设置Logtail启动参数。

如下示例表示将CPU的资源限制放宽到双核，将内存资源的限制放宽到2048MB。

```
{
    ...
    "cpu_usage_limit":2,
    "mem_usage_limit":2048,
    ...
}
```

数据可靠性

建议您启用MySQL服务器的全局事务ID（GTID）功能，并将Logtail升级到0.16.15及以上版本以保证数据可靠性，避免因主备切换造成的数据重复采集。

- 数据漏采集：Logtail与MySQL服务器之间的网络长时间中断时，可能会产生数据漏采集情况。

  如果Logtail和MySQL master节点之间的网络发生中断，MySQL master节点仍会不断地产生新的Binlog数据并且回收旧的Binlog数据。当网络恢复，Logtail与MySQL master节点重连成功后，Logtail会使用自身的checkpoint向MySQL master节点请求更多的Binlog数据。但由于长时间的网络中断，它所需要的数据很可能已经被回收，这时会触发Logtail的异常恢复机制。在异常恢复机制中，Logtail会从MySQL master节点获取最近的Binlog位置，以它为起点继续采集，这样就会跳过checkpoint和最近的Binlog位置之间的数据，导致数据漏采集。

- 数据重复采集：当MySQL master节点和slave节点之间的Binlog序号不同步时，发生了主备切换事件，可能会产生数据重复采集情况。

  在MySQL主备同步的设置下，MySQL master节点会将产生的Binlog同步给MySQL slave节点，MySQL slave节点收到后存储到本地的Binlog文件中。当MySQL master节点和slave节点之间的Binlog序号不同步时，发生了主备切换事件，以Binlog文件名和文件大小偏移量作为checkpoint的机制将导致数据重复采集。

  例如，有一段数据在MySQL master节点上位于 `(binlog.100, 4)` 到 `(binlog.105, 4)` 之间，而在MySQL slave节点上位于 `(binlog.1000, 4)` 到 `(binlog.1005, 4)` 之间，并且Logtail已经从MySQL master节点获取了这部分数据，将本地checkpoint更新到了 `(binlog.105, 4)` 。如果此时发生了主备切换且无任何异常发生，Logtail将会继续使用本地checkpoint `(binlog.105, 4)` 去向新的MySQL master节点采集binlog。但是因为新的MySQL master上的 `(binlog.1000, 4)` 到 `(binlog.1005, 4)` 这部分数据的序号都大于Logtail所请求的序号，MySQL master将它们返回给Logtail，导致重复采集。

操作步骤

1.
2. 在**接入数据**区域，选择**MySQL BinLog-插件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。

   插件配置中已提供模板，包括inputs和processors，请根据您的需求替换配置参数。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。

   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

```
{
 "inputs": [
     {
         "type": "service_canal",
         "detail": {
             "Host": "************.mysql.rds.aliyuncs.com",
             "Port": 3306,
             "User" : "root",
             "ServerID" : 56321,
             "Password": "********",
             "IncludeTables": [
                 "user_info\\..*"
             ],
             "ExcludeTables": [
                 ".*\\.\\S+_inner"
             ],
             "TextToString" : true,
             "EnableDDL" : true
         }
     }
 ]
}
```

参数	类型	是否必须	说明
type	string	是	数据源类型，固定为service_canal。
Host	string	否	数据库主机。不配置时，默认为127.0.0.1。
Port	int	否	数据库端口。不配置时，默认为3306。

User	string	否	数据库用户名。不配置时，默认为root。
			需保证配置的用户具有数据库读权限以及MySQL REPLICATION权限，示例如下。
			`CREATE USER canal IDENTIFIED BY 'canal';`
			`GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%';`
			`-- GRANT ALL PRIVILEGES ON *.* TO 'canal'@'%' ;`
			`FLUSH PRIVILEGES;`

Password	string	否	数据库密码。不配置时，默认为空。 如果安全需求较高，建议将访问用户名和密码配置为xxx，待配置同步至本地机器后，在本地文件*/usr/local/ilogtail/user_log_config.json*找到对应配置进行修改，详情请参见修改本地配置。 **说明** 如果您在控制台上修改了此参数，同步至本地后会覆盖当前本地的配置。
ServerID	int	否	Logtail伪装成的Mysql Slave的ID。不配置时，默认为125。 **说明** ServerID对于MySQL数据库必须唯一，否则会采集失败。
IncludeTables	string数组	是	包含的表名称（包括db，例如：test_db.test_table），可配置为正则表达式。如果某个表不符合IncludeTables中的任一条件则该表不会被采集。如果您希望采集所有表，请将此参数设置为.*\\..*。 **说明** 如果需要完全匹配，请在前面加上^，后面加上$，例如：^test_db\\.test_table$。

参数	类型	是否必须	说明
ExcludeTables	string 数组	否	忽略的表名称（包括db，例如：test_db.test_table），可配置为正则表达式。如果某个表符合ExcludeTables中的任一条件则该表不会被采集。不设置时默认收集所有表。 **说明** 如果需要完全匹配，请在前面加上^，后面加上$，例如：^test_db\\.test_table$。

StartBinName	string	否	首次采集的Binlog文件名。不设置时，默认从当前时间点开始采集。
			如果想从指定位置开始采集，可以查看当前的Binlog文件以及文件大小偏移量，并将StartBinName、StartBinlogPos设置成对应的值，示例如下。
			`# StartBinName设置成 "mysql-bin.000063", StartBinlogPos设置成 0`
			`mysql> show binary logs;`
			`+------------------+-----------+`
			`| Log_name | File_size |`
			`+------------------+-----------+`
			`| mysql-bin.000063 | 241 |`
			`| mysql-bin.000064 | 241 |`
			`| mysql-bin.000065 | 241 |`
			`| mysql-bin.000066 | 10778 |`
			`+------------------+-----------+`
			`4 rows in set (0.02 sec)`
			说明
			当指定StartBinName时，第一次采集会产生较大流量。

StartBinlogPos	int	否	首次采集的Binlog文件的偏移量，不设置时，默认为0。
EnableGTID	bool	否	是否附加全局事务ID。不设置时，默认为true。设置为false时，表示上传的数据不附加全局事务ID。
EnableInsert	bool	否	是否采集insert事件的数据。不设置时，默认为true。设置为false时，表示将不采集insert事件数据。
EnableUpdate	bool	否	是否采集update事件的数据。不设置时，默认为true。设置为false时，表示不采集update事件数据。
EnableDelete	bool	否	是否采集delete事件的数据。不设置时，默认为true。设置为false时，表示不采集delete事件数据。
EnableDDL	bool	否	是否采集DDL（data definition language）事件数据。不设置时， 默认为false，表示不收集DDL事件数据。 **说明** 该选项不支持IncludeTables和ExcludeTables过滤。
Charset	string	否	编码方式。不设置时，默认为utf-8。
TextToString	bool	否	是否将text类型的数据转换成字符串。不设置时，默认为false，表示不转换。
PackValues	bool	否	是否将事件数据打包成JSON格式。默认为false，表示不打包。如果设置为true，Logtail会将事件数据以JSON格式集中打包到data和old_data两个字段中，其中old_data仅在row_update事件中有意义。 示例：假设数据表有三列数据c1，c2，c3，设置为false，row_insert事件数据中会有c1，c2，c3三个字段，而设置为true时，c1，c2，c3会被统一打包为data字段，值为 `{"c1":"...", "c2": "...", "c3": "..."}` 。 **说明** 该参数仅支持Logtail 0.16.19及以上版本。

参数	类型	是否必须	说明
EnableEventMeta	bool	否	是否采集事件的元数据，默认为false，表示不采集。 Binlog事件的元数据包括event_time、event_log_position、event_size和event_server_id。 **说明** 该参数仅支持Logtail 0.16.21及以上版本。

下发Logtail采集配置到服务器后，如果您的数据库存在相关的修改操作，Logtail会立即上报数据采集到日志服务。

说明 Logtail默认采集Binlog的增量数据。

修改本地配置

如果您没有在**插件配置**中输入真实的Host、User、Password等信息，可以在插件配置下发到本地后进行手动修改。

1. 登录Logtail所在服务器。
2. 打开*/usr/local/ilogtail/user_log_config.json*文件，找到service_canal关键字，修改Host、User、Password等字段。
3. 执行以下命令重启Logtail。

```
sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
```

后续步骤

Logtail采集Binlog到日志服务后，您可以在日志服务控制台上查看日志。例如：对 `user_info` 数据库下的 `SpecialAlarm` 表分别执行 `INSERT` 、 `UPDATE` 、 `DELETE` 操作，数据库表结构、数据库操作及日志样例如下所示。

- 表结构

```
CREATE TABLE `SpecialAlarm` (
`id` int(11) unsigned NOT NULL AUTO_INCREMENT,
`time` datetime NOT NULL,
`alarmtype` varchar(64) NOT NULL,
`ip` varchar(16) NOT NULL,
`count` int(11) unsigned NOT NULL,
PRIMARY KEY (`id`),
KEY `time` (`time`) USING BTREE,
KEY `alarmtype` (`alarmtype`) USING BTREE
) ENGINE=MyISAM AUTO_INCREMENT=1;
```

- 数据库操作

  执行INSERT、DELETE和UPDATE三种操作。

```
insert into specialalarm (`time`, `alarmType`, `ip`, `count`) values(now(), "NO_ALARM", "10.10.**.***", 55);
delete from specialalarm where id = 4829235  ;
update specialalarm set ip = "10.11.***.**" where id = "4829234";
```

  为 `zc.specialalarm` 创建一个索引。

```
ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC);
```

- 日志样例

  在查询分析页面，查看每种操作对应的日志，日志样例如下所示。

- INSERT语句

```
__source__:  10.30.**.**
__tag__:__hostname__:  iZbp145dd9fccu*****
__topic__:
_db_:  zc
_event_:  row_insert
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:536
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  113
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829235
ip:  10.10.***.***
time:  2017-11-01 12:31:41
```

- DELETE语句

```
__source__:  10.30.**.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_delete
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:537
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  114
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829235
ip:  10.10.**.***
time:  2017-11-01 12:31:41
```

- UPDATE语句

```
__source__:  10.30.**.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_update
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:538
_host_:  *********.mysql.rds.aliyuncs.com
_id_:  115
_old_alarmtype:  NO_ALARM
_old_count:  55
_old_id:  4829234
_old_ip:  10.10.22.133
_old_time:  2017-10-31 12:04:54
_table_:  specialalarm
alarmtype:  NO_ALARM
count:  55
id:  4829234
ip:  10.11.***.***
time:  2017-10-31 12:04:54
```

- DDL（data definition language）语句

```
__source__:  10.30.**.**
__tag__:__hostname__:  iZbp145dd9fccu****
__topic__:
_db_:  zc
_event_:  row_update
_gtid_:  7d2ea78d-b631-11e7-8afb-00163e0eef52:539
_host_:  *********.mysql.rds.aliyuncs.com
ErrorCode:  0
ExecutionTime:  0
Query:  ALTER TABLE `zc`.`specialalarm`
ADD INDEX `time_index` (`time` ASC)
StatusVars:
```

字段	说明
host	数据库host名称。
_db_	数据库名称。
table	表的名称。
event	事件类型。
_id_	本次采集的自增ID，从0开始，每次采集一个binlog事件后加1。
gtid	全局事务ID。
filename	Binlog文件名。
offset	Binlog文件大小偏移量，该值只会在每次commit后更新。

3.7.3. 采集MySQL查询结果

本文介绍如何通过日志服务控制台创建Logtail采集配置来采集MySQL查询结果。

前提条件

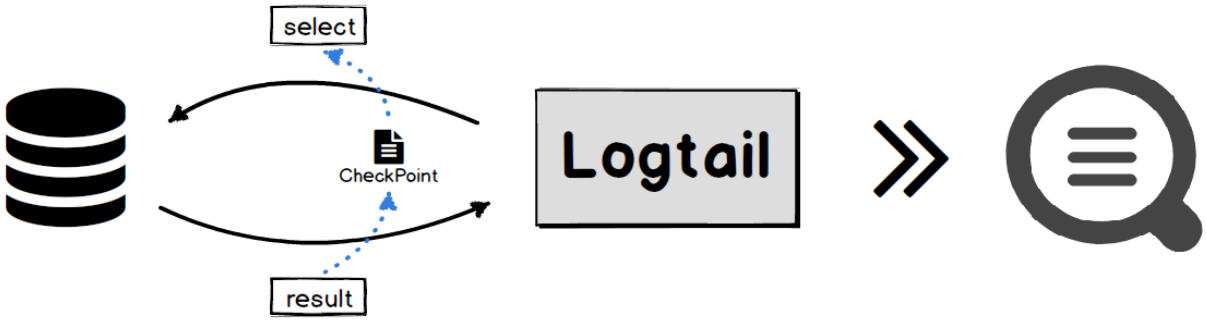
已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

> **说明** 目前支持Linux Logtail 0.16.0及以上版本，Window Logtail 1.0.0.8及以上版本。

原理

Logtail根据Logtail采集配置定期执行指定的SELECT语句，将返回结果作为数据上传到日志服务。

Logtail获取到执行结果时，会将结果中配置的CheckPoint字段保存在本地，当下次执行SELECT语句时，会将上一次保存的CheckPoint带入到SELECT语句中，以此实现增量数据采集。



功能

- 支持MySQL类型的数据库。
- 支持分页设置。
- 支持时区设置。
- 支持超时设置。
- 支持保存CheckPoint状态。
- 支持SSL。
- 支持限制每次最大采集数量。

应用场景

- 根据数据中的自增ID或时间等标志采集增量数据。
- 根据筛选条件自定义同步。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**MySQL查询结果-插件**。

3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。
   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

```
{
  "inputs": [
    {
      "type": "service_mysql",
      "detail": {
        "Address": "************.mysql.rds.aliyuncs.com",
        "User": "****",
        "Password": "*******",
        "DataBase": "****",
        "Limit": true,
        "PageSize": 100,
        "StateMent": "select * from db.VersionOs where time > ?",
        "CheckPoint": true,
        "CheckPointColumn": "time",
        "CheckPointStart": "2018-01-01 00:00:00",
        "CheckPointSavePerPage": true,
        "CheckPointColumnType": "time",
        "IntervalMs": 60000
      }
    }
  ]
}
```

参数	类型	是否必选	说明
type	string	是	数据源类型，固定为service_mysql。
Address	string	否	MySQL地址。不配置时，默认为127.0.0.1:3306。
User	string	否	数据库用户名。不配置时，默认为root。

参数	类型	是否必选	说明
Password	string	否	数据库密码。不配置时，默认为空。 如果安全需求较高，建议将SQL访问用户名和密码配置为xxx，待配置同步至本地机器后，在本地文件*/usr/local/ilogtail/user_log_config.json*找到对应配置进行修改。更多信息，请参见修改本地配置。 **说明** 如果您在控制台上修改了此参数，同步至本地后会覆盖当前本地的配置。
DataBase	string	否	数据库名称。
DialTimeOutMs	int	否	数据库连接超时时间，单位：ms。不配置时，默认为5000ms。
ReadTimeOutMs	int	否	数据库读取超时时间，单位：ms。不配置时，默认为5000ms。
StateMent	string	否	SQL语句。 设置CheckPoint为true时，StateMent中SQL语句的where条件中必须包含CheckPointColumn，并将该列的值配置为?。例如：CheckPointColumn配置为id，则StateMent配置为`SELECT * from ... where id > ?`。
Limit	boolean	否	是否使用Limit分页。不配置时，默认为false，表示不使用Limit分页。 建议使用Limit进行分页。设置Limit为true后，进行SQL查询时，会自动在StateMent中追加LIMIT语句。
PageSize	int	否	分页大小，Limit为true时必须配置。
MaxSyncSize	int	否	每次同步最大记录数。不配置时，默认为0，表示无限制。
CheckPoint	boolean	否	是否使用checkpoint。不配置时，默认为false，表示不使用checkpoint。
CheckPointColumn	string	否	checkpoint列名称。 CheckPoint为true时必须配置。 **注意** 该列的值必须递增，否则可能会出现数据漏采集问题（每次查询结果中的最大值将作为下次查询的输入）。
CheckPointColumnType	string	否	checkpoint列类型，支持int和time两种类型。int类型的内部存储为int64，time类型支持MySQL的date、datetime、time。 CheckPoint为true时必须配置。
CheckPointStart	string	否	checkpoint初始值。 CheckPoint为true时必须配置。
CheckPointSavePerPage	boolean	否	设置为true，则每次分页时保存一次checkpoint；设置为false，则每次同步完后保存checkpoint。
IntervalMs	int	是	同步间隔，单位：ms。

修改本地配置

如果您没有在**插件配置**中输入真实的Address、User、Password等信息，可以在插件配置下发到本地后进行手动修改。

1. 登录Logtail所在服务器。
2. 打开*/usr/local/ilogtail/user_log_config.json*文件，找到service_mysql关键字，修改Address、User、Password等字段。
3. 执行以下命令重启Logtail。

```
sudo /etc/init.d/ilogtaild stop; sudo /etc/init.d/ilogtaild start
```

后续步骤

Logtail采集MySQL查询结果到日志服务后，您可以在日志服务控制台上进行查看。数据库表结构和Logtail采集到的日志样例如下所示。

- 表结构

```
CREATE TABLE `VersionOs` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT COMMENT 'id',
  `time` datetime NOT NULL,
  `version` varchar(10) NOT NULL DEFAULT '',
  `os` varchar(10) NOT NULL,
  `count` int(11) unsigned NOT NULL,
  PRIMARY KEY (`id`),
  KEY `timeindex` (`time`)
)
```

- 日志样例

```
"count":  "4"
"id:  "721097"
"os:  "Windows"
"time:  "2017-08-25 13:00:00"
"version":  "1.3.0"
```

3.7.4. 采集HTTP数据

Logtail插件会根据您的采集配置定期请求指定的URL，将请求返回的body内容作为数据源上传到日志服务。本文介绍如何通过日志服务控制台创建Logtail采集配置来采集HTTP数据。

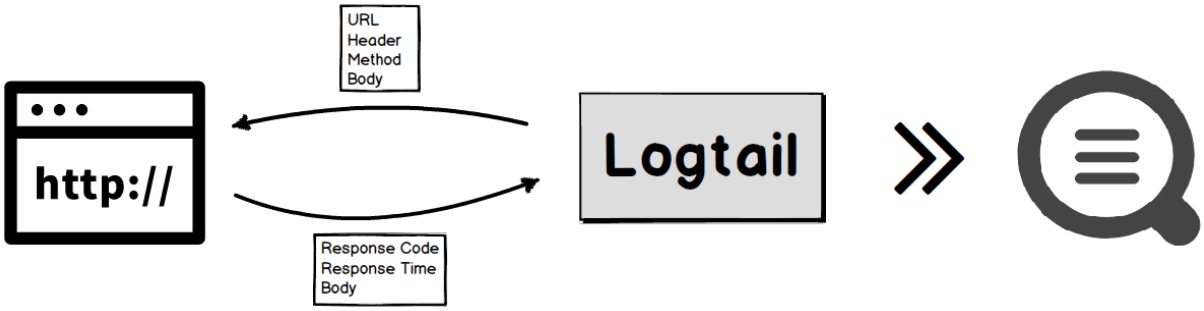
前提条件

已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

> **说明** 目前支持Linux Logtail 0.16.0及以上版本，Window Logtail 1.0.0.8及以上版本。

原理

Logtail根据您在采集配置中设置的HTTP请求的URL、Method、Header、Body等信息，定期对指定URL发起请求，将请求返回的状态码、body内容以及响应时间做为数据源上传到日志服务。



功能

- 支持配置多个URL。
- 支持配置HTTP方法。
- 支持配置HTTP请求的间隔。
- 支持自定义请求头。
- 支持HTTPS。
- 支持检测body是否匹配固定模式。

应用场景

- 监控应用状态（以HTTP方式提供监控接口），例如：
  - Nginx
  - Docker（HTTP方式）
  - Elastic Search
  - Haproxy

- 其他以HTTP方式提供监控接口的服务
- 检测服务可用性。

  定期请求服务，通过状态码以及请求延迟做服务的可用性监控。
- 定期拉取数据，例如微博评论、粉丝数等。

使用限制

- URL必须以 `http` 或 `https` 开头。
- 不支持自定义证书。
- 不支持交互式通信方式。

操作步骤

例如：每隔1000ms请求一次nginx status模块，URL为 `http://127.0.0.1/ngx_status` ，使用正则表达式提取返回body中的状态信息，操作步骤如下所示。

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**自定义数据插件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。

     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

     日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。
   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

```
{
 "inputs": [
     {
         "type": "metric_http",
         "detail": {
             "IntervalMs": 1000,
             "Addresses": [
                 "http://127.0.0.1/ngx_status"
             ],
             "IncludeBody": true
         }
     }
 ],
 "processors" : [
     {
         "type": "processor_regex",
         "detail" : {
             "SourceKey": "content",
             "Regex": "Active connections: (\\d+)\\s+server accepts handled requests\\s+(\\d+)\\s+(\\d+)\\s+(\\d+)\\s+
Reading: (\\d+) Writing: (\\d+) Waiting: (\\d+).*",
             "Keys": [
                 "connection",
                 "accepts",
                 "handled",
                 "requests",
                 "reading",
                 "writing",
                 "waiting"
             ],
             "FullMatch": true,
             "NoKeyError": true,
             "NoMatchError": true,
             "KeepSource": false
         }
     }
 ]
}
```

参数	类型	是否必选	参数说明
type	string	是	数据源类型，固定为metric_http。
Addresses	string 数组	是	URL列表。 **说明** 必须以 `http` 或 `https` 开头。
IntervalMs	int	是	每次请求的间隔，单位：ms。
Method	string	否	请求的方法名。必须大写，默认为 `GET` 。
Body	string	否	HTTP Body字段内容，默认为空。
Headers	key：string， value：string map	否	HTTP Header的内容，默认为空。
PerAddressSleepMs	int	否	Addresses列表中，每个URL请求的间隔时间，单位：ms，默认值：100 ms。
ResponseTimeoutMs	int	否	请求超时的时间，单位：ms，默认值：5000 ms。
IncludeBody	bool	否	是否采集请求的body，默认值：false。如果为true，则将请求body内容存放在名为content的key中。
FollowRedirects	bool	否	是否自动处理重定向，默认值：false。
InsecureSkipVerify	bool	否	是否跳过HTTPS安全检查，默认值：false。

参数	类型	是否必选	参数说明
ResponseStringMatch	string	否	对返回的body内容进行正则表达式检查，检查结果存放在名为_response_match_的key中，如果匹配，value为yes；如果不匹配，value为no。

执行结果

采集完成后，您可以在日志服务控制台查看数据，除通过正则表达式解析过的数据外，还包括HTTP请求附加的method、address、time、code、result信息。

```
"Index" : "7"
"connection" : "1"
"accepts" : "6079"
"handled" : "6079"
"requests" : "11596"
"reading" : "0"
"writing" : "1"
"waiting" : "0"
"_method_" : "GET"
"_address_" : "http://127.0.0.1/ngx_status"
"_response_time_ms_" : "1.320"
"_http_response_code_" : "200"
"_result_" : "success"
```

每次请求，默认上传以下字段。

字段	说明
address	请求地址。
method	请求方法。
_response_time_ms_	响应延迟时间，单位：ms。
_http_response_code_	状态码。
result	请求的结果，取值为success、invalid_body、match_regex_invalid、mismatch、timeout。
_response_match_	返回的body内容是否匹配ResponseStringMatch字段。如果不存在`ResponseStringMatch`字段，值为null，如果指定了ResponseStringMatch字段，值为yes或no。

3.7.5. 采集Syslog

本文介绍如何通过日志服务控制台创建Logtail采集配置来采集Syslog。

前提条件

已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

> **说明** 目前支持Linux Logtail 0.16.13及以上版本，Window Logtail 1.0.0.8及以上版本。

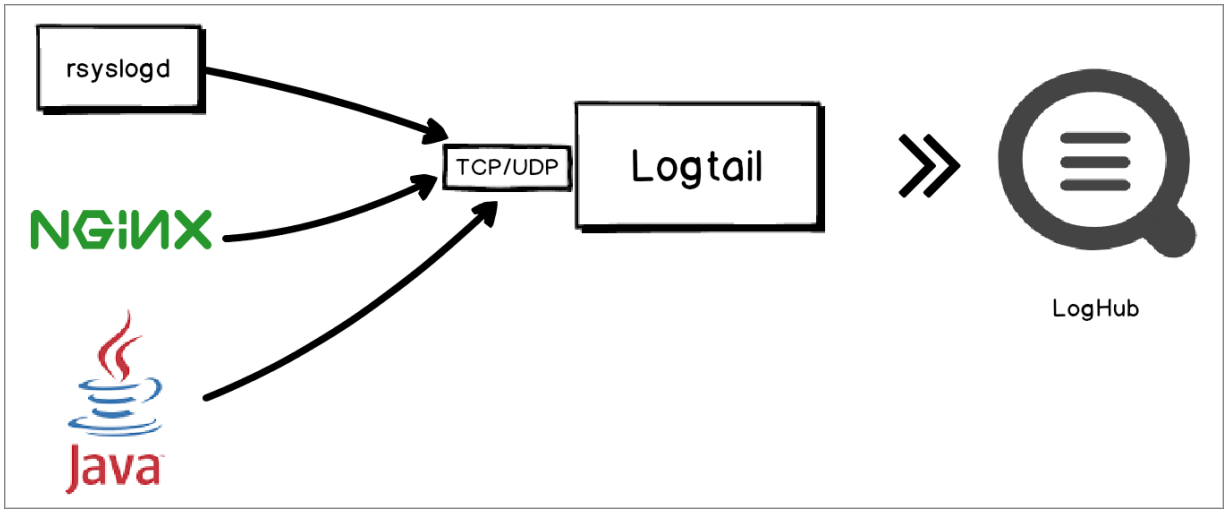
简介

在Linux服务器中，您可以通过rsyslog等syslog agent将本地的syslog数据转发到指定服务器IP地址和端口上。为指定服务器添加Logtail采集配置后，Logtail插件会以TCP协议或UDP协议接收转发过来的syslog数据，并根据指定的syslog协议进行解析，提取日志中的facility、tag（program）、severity、content等字段。syslog协议支持RFC3164和RFC5424。

您可以同时配置多个Logtail插件，例如同时使用TCP和UDP监听127.0.0.1:9000。

实现原理

通过Logtail插件对指定的地址和端口进行监听后，Logtail开始采集数据，包括通过rsyslog采集的系统日志、 Nginx转发的访问日志或错误日志，以及通过syslog客户端转发的日志。



配置Logtail插件采集syslog

1. 为rsyslog添加一条转发规则。
   i. 在syslog所在的服务器上修改rsyslog的配置文件*/etc/rsyslog.conf*，在配置文件的最后添加一行转发规则。
   添加转发规则后，rsyslog会将syslog转发至指定IP地址和端口上。
   - 如果通过当前服务器采集本机syslog，配置转发地址为127.0.0.1，端口为任意非知名的空闲端口。
   - 如果通过其他服务器采集本机syslog，配置转发地址为其他服务器的公网IP，端口为任意非知名的空闲端口。

   例如以下配置表示将所有的日志都通过TCP转发至127.0.0.1:9000，配置文件详细说明请参见RSyslog Documentation。

   ```
   *.* @@127.0.0.1:9000
   ```

   ii. 执行以下命令重启rsyslog，使日志转发规则生效。

   ```
   sudo service rsyslog restart
   ```

2. 登录日志服务控制台。
3. 在**接入数据**区域，选择**自定义数据插件**。
4. 选择目标Project和Logstore，单击**下一步**。
5. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。
     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。
     日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
6. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

7. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

> **说明** 一个inputs中只允许配置一个类型的数据源。

- processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

同时监听UDP和TCP的示例配置如下：

```
{
    "inputs": [
        {
            "type": "service_syslog",
            "detail": {
                "Address": "tcp://127.0.0.1:9000",
                "ParseProtocol": "rfc3164"
            }
        },
        {
            "type": "service_syslog",
            "detail": {
                "Address": "udp://127.0.0.1:9001",
                "ParseProtocol": "rfc3164"
            }
        }
    ]
}
```

配置项	类型	是否必选	说明
type	string	是	数据源类型，固定为service_syslog。
Address	string	否	指定Logtail插件监听的协议、地址和端口，Logtail插件会根据Logtail采集配置进行监听并获取日志数据。格式为[tcp/udp]://[*ip*]:[*port*]。不配置时，默认为tcp://127.0.0.1:9999。 **说明** - Logtail插件配置中设置的监听协议、地址和端口号必须与rsyslog配置文件设置的转发规则相同。 - 如果安装Logtail的服务器有多个IP地址可接收日志，可以将地址配置为0.0.0.0，表示监听服务器的所有IP地址。
ParseProtocol	string	否	指定解析日志所使用的协议，默认为空，表示不解析。其中： - rfc3164：指定使用RFC3164协议解析日志。 - rfc5424：指定使用RFC5424协议解析日志。 - auto：指定插件根据日志内容自动选择合适的解析协议。
IgnoreParseFailure	boolean	否	指定解析失败后的操作，不配置时，默认为true，表示放弃解析，直接填充所返回的content字段。配置为false，表示解析失败时丢弃日志。

配置Logtail插件采集Nginx日志

Nginx支持直接把访问日志以syslog协议转发到指定IP地址和端口。如果您希望把服务器上包括Nginx访问日志在内的所有数据都以syslog形式集中投递到日志服务，可以根据需求创建Logtail采集配置进行采集。

1. 为Nginx添加一条转发规则。
   i. 在Nginx服务器的*nginx.conf*文件中增加转发规则，详情请参见Nginx官网说明。
   例如，在配置文件中增加如下内容。

```
http {
    ...
    # Add this line.
    access_log syslog:server=127.0.0.1:9000,facility=local7,tag=nginx,severity=info combined;
    ...
}
```

   ii. 执行以下命令重启Nginx服务，使配置生效。

```
sudo service nginx restart
```

2. 创建Logtail采集配置。具体操作，请参见配置Logtail插件采集syslog。

后续步骤

Logtail采集Syslog到日志服务后，您可以在日志服务控制台查看日志。

字段	说明
hostname	主机名，如果日志中未提供则获取当前主机名。
program	协议中的tag字段。
priority	协议中的priority字段。
facility	协议中的facility字段。
severity	协议中的severity字段。
unixtimestamp	日志对应的时间戳。
content	日志内容，如果解析失败的话，此字段包含未解析日志的所有内容。
_ip_	当前主机的IP地址。
_client_ip_	传输日志的客户端IP地址。

3.7.6. 采集Beats和Logstash数据源

本文介绍如何通过日志服务控制台创建Logtail采集配置来采集Beats和Logstash数据源。

前提条件

- 已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

  > **说明** 目前仅支持Linux Logtail 0.16.9及以上版本，Window Logtail 1.0.0.8及以上版本。

- 已通过Logstash或Beats系列软件采集到数据。
  - 如果要通过Logstash采集数据，请参见Logstash-Lumberjack-Output。
  - 如果要通过Beats系列软件采集数据，请参见Beats-Lumberjack-Output。

    本文以使用PacketBeat软件采集本地网络数据包，并使用Logtail Lumberjack插件上传到日志服务为例，进行说明。配置PacketBeat输出方式为Logstash，示例如下所示。

    ```
    output.logstash:
      hosts: ["127.0.0.1:5044"]
    ```

背景信息

基于Logstash、Beats系列软件对Lumberjack协议的支持，Logtail可以通过Lumberjack协议将Beats系列软件（MetricBeat、PacketBeat、Winlogbeat、Auditbeat、Filebeat、Heartbeat等）、Logstash采集的数据上传到日志服务。

> **说明**
> - 同一Logtail可配置多个Lumberjack插件，但多个插件不能监听同一端口。
> - Lumberjack插件支持SSL，上传Logstash采集的数据需要使用该功能。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**自定义数据插件**。

3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。
   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

   由于Beats和Logstash输出的都是JSON格式数据，因此需要使用 `processor_anchor` 将JSON展开。

```
{
  "inputs": [
    {
      "detail": {
        "BindAddress": "0.0.0.0:5044"
      },
      "type": "service_lumberjack"
    }
  ],
  "processors": [
    {
      "detail": {
        "Anchors": [
          {
            "ExpondJson": true,
            "FieldType": "json",
            "Start": "",
            "Stop": ""
          }
        ],
        "SourceKey": "content"
      },
      "type": "processor_anchor"
    }
  ]
}
```

参数	类型	是否必选	说明
type	string	是	数据源类型，固定为service_lumberjack。
BindAddress	string	否	Lumberjack协议绑定的地址。不配置时，默认为127.0.0.1:5044，支持自定义。如果Lumberjack协议需要被局域网内其他主机访问，请配置为0.0.0.0:5044。

参数	类型	是否必选	说明
V1	bool	否	是否使用Lumberjack V1版本协议。不配置时，默认为false。目前Logstash支持Lumberjack V1。
V2	bool	否	是否使用Lumberjack V2版本协议。不配置时，默认为true。目前Beats系列软件支持Lumberjack V2。
SSLCA	string	否	证书授权机构（Certificate Authority）颁发的签名证书路径，默认为空。如果是自签名证书可不设置此选项。
SSLCert	string	否	证书的路径，默认为空。
SSLKey	string	否	证书对应私钥的路径，默认为空。
InsecureSkipVerify	bool	否	是否跳过SSL安全检查。不配置时，默认为false，执行SSL安全检查。

后续步骤

Logtail采集数据到日志服务后，您可以在日志服务控制台上进行查看。

```
_@metadata_beat:  packetbeat
_@metadata_type:  doc
_@metadata_version:  6.2.4
_@timestamp:  2018-06-05T03:58:42.470Z
__source__:  **.**.**.**
__tag__:__hostname__:  *******
__topic__:
_beat_hostname:  bdbe0b8d53a4
_beat_name:  bdbe0b8d53a4
_beat_version:  6.2.4
_bytes_in:  56
_bytes_out:  56
_client_ip:  192.168.5.2
_icmp_request_code:  0
_icmp_request_message:  EchoRequest(0)
_icmp_request_type:  8
_icmp_response_code:  0
_icmp_response_message:  EchoReply(0)
_icmp_response_type:  0
_icmp_version:  4
_ip:  127.0.0.1
_path:  127.0.0.1
_responsetime:  0
_status:  OK
_type:  icmp
```

3.7.7. 采集Systemd Journal日志

Logtail支持从原始的二进制文件中采集Linux系统的Systemd Journal日志。本文介绍如何通过日志服务控制台创建Logtail采集配置来采集Systemd Journal日志。

前提条件

已在服务器上安装Logtail，详情请参见安装Logtail（Linux系统）。

> **说明** 目前仅支持Linux Logtail 0.16.18及以上版本。

简介

Systemd是专用于Linux操作系统的系统与服务管理器。当作为启动进程（PID=1）运行时，它将作为初始化系统运行，启动并维护各种用户空间的服务。Systemd统一管理所有Unit的日志（包括内核和应用日志），配置文件一般为*/etc/systemd/journald.conf*。

> **说明** 运行的操作系统需支持Journal日志格式。

功能

- 支持设置初始采集位置，后续采集会自动保存checkpoint，应用重启时不影响进程。
- 支持过滤指定的Unit。
- 支持采集内核日志。
- 支持自动解析日志等级。
- 支持以容器方式采集宿主机上的Journal日志，适用于Docker、Kubernetes场景。

应用场景

- 监听内核事件，出现异常时自动告警。
- 采集所有系统日志，用于长期存储，减少磁盘空间占用。
- 采集软件（Unit）的输出日志，用于分析或告警。
- 采集所有Journal日志，可以从所有日志中快速检索关键词或日志，相比Journalctl的查询效率大幅提升。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**自定义数据插件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 在**创建机器组**页签中，创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。
   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

```
{
  "inputs": [
    {
      "detail": {
        "JournalPaths": [
          "/var/log/journal"
        ],
        "Kernel": true,
        "ParsePriority": true,
        "ParseSyslogFacility": true
      },
      "type": "service_journal"
    }
  ]
}
```

配置项	类型	是否必选	说明
type	string	是	数据源类型，固定为service_journal。

配置项	类型	是否必选	说明
JournalPaths	string数组	是	Journal日志路径，建议配置为Journal日志所在目录，例如*/var/log/journal*。
SeekPosition	string	否	首次采集方式，可以配置为head或tail。不配置时，默认为tail。 ○ head表示采集所有数据。 ○ tail表示只采集Logtail采集配置被应用后的新数据。
Kernel	bool	否	是否采集内核日志，默认为true，表示采集内核日志。
Units	string数组	否	指定采集的Unit列表，默认为空，表示全部采集。
ParseSyslogFacility	bool	否	是否解析syslog日志的facility字段。不配置时，默认为false，表示不解析。

ParsePriority	bool	否	是否解析Priority字段。不配置时，默认为false，表示不解析。
			设置为true时，ParsePriority映射关系如下所示。
			`"0": "emergency"`
			`"1": "alert"`
			`"2": "critical"`
			`"3": "error"`
			`"4": "warning"`
			`"5": "notice"`
			`"6": "informational"`
			`"7": "debug"`

UseJournalEventTime	bool	否	是否使用Journal日志中的字段作为日志时间。不配置时，默认为false，表示使用采集时间作为日志时间。实时日志采集一般相差3秒以内。

示例

- 示例1

  从默认的 `/var/log/journal` 目录采集Journal日志，采集配置为：

```
{
  "inputs": [
    {
      "detail": {
        "JournalPaths": [
          "/var/log/journal"
        ]
      },
      "type": "service_journal"
    }
  ]
}
```

  日志样例：

```
MESSAGE:  rejected connection from "192.168.0.250:43936" (error "EOF", ServerName "")
PACKAGE:  embed
PRIORITY:  6
SYSLOG_IDENTIFIER:  etcd
_BOOT_ID:  fe919cd1268f4721bd87b5c18afe59c3
_CAP_EFFECTIVE:  0
_CMDLINE:  /usr/bin/etcd --election-timeout=3000 --heartbeat-interval=500 --snapshot-count=50000 --data-dir=data.etcd --
name 192.168.0.251-name-3 --client-cert-auth --trusted-ca-file=/var/lib/etcd/cert/ca.pem --cert-file=/var/lib/etcd/cert/
etcd-server.pem --key-file=/var/lib/etcd/cert/etcd-server-key.pem --peer-client-cert-auth --peer-trusted-ca-file=/var/li
b/etcd/cert/peer-ca.pem --peer-cert-file=/var/lib/etcd/cert/192.168.0.251-name-3.pem --peer-key-file=/var/lib/etcd/cert/
192.168.0.251-name-3-key.pem --initial-advertise-peer-urls https://192.168.0.251:2380 --listen-peer-urls https://192.168
.0.251:2380 --advertise-client-urls https://192.168.0.251:2379 --listen-client-urls https://192.168.0.251:2379 --initial
-cluster 192.168.0.249-name-1=https://192.168.0.249:2380,192.168.0.250-name-2=https://192.168.0.250:2380,192.168.0.251-n
ame-3=https://192.168.0.251:2380 --initial-cluster-state new --initial-cluster-token abac64c8-baab-4ae6-8412-4253d3cfb0c
f
_COMM:  etcd
_EXE:  /opt/etcd-v3.3.8/etcd
_GID:  995
_HOSTNAME:  iZbp1f7y2ikfe4l8nx95amZ
_MACHINE_ID:  f0f31005fb5a436d88e3c6cbf54e25aa
_PID:  10926
_SOURCE_REALTIME_TIMESTAMP:  1546854068863857
_SYSTEMD_CGROUP:  /system.slice/etcd.service
_SYSTEMD_SLICE:  system.slice
_SYSTEMD_UNIT:  etcd.service
_TRANSPORT:  journal
_UID:  997
__source__:  172.16.1.4
__tag__:__hostname__:  logtail-ds-8kqb9
__topic__:
_monotonic_timestamp_:  1467135144311
_realtime_timestamp_:  1546854068864309
```

- 示例2

  Kubernetes场景下，使用DaemonSet模式采集宿主机的系统日志，由于日志中有很多并不重要的字段，使用处理插件只挑选较为重要的日志字段。采集配置为：

```
{
  "inputs": [
    {
      "detail": {
        "JournalPaths": [
          "/logtail_host/var/log/journal"
        ],
        "ParsePriority": true,
        "ParseSyslogFacility": true
      },
      "type": "service_journal"
    }
  ],
  "processors": [
    {
      "detail": {
        "Exclude": {
          "UNIT": "^libcontainer.*test"
        }
      },
      "type": "processor_filter_regex"
    },
    {
      "detail": {
        "Include": [
          "MESSAGE",
          "PRIORITY",
          "_EXE",
          "_PID",
          "_SYSTEMD_UNIT",
          "_realtime_timestamp_",
          "_HOSTNAME",
          "UNIT",
          "SYSLOG_FACILITY",
          "SYSLOG_IDENTIFIER"
        ]
      },
      "type": "processor_pick_key"
    }
  ]
}
```

日志样例：

```
MESSAGE:  rejected connection from "192.168.0.251:48914" (error "EOF", ServerName "")
PRIORITY:  informational
SYSLOG_IDENTIFIER:  etcd
_EXE:  /opt/etcd-v3.3.8/etcd
_HOSTNAME:  iZbp1i0czq3zgvxlx7u8ueZ
_PID:  10590
_SYSTEMD_UNIT:  etcd.service
__source__:  172.16.0.141
__tag__:__hostname__:  logtail-ds-dp48x
__topic__:
_realtime_timestamp_:  1547975837008708
```

3.7.8. 采集Docker事件

Docker事件信息记录了容器、镜像、插件、网络、存储等所有交互事件。本文介绍如何通过日志服务控制台创建Logtail采集配置来采集Docker事件。

前提条件

已在服务器上安装Logtail。更多信息，请参见安装Logtail（Linux系统）。

> **说明** 目前仅支持Linux Logtail 0.16.18及以上版本。

限制说明

- Logtail可运行在容器模式或宿主机上，需具备访问Docker的权限（可以访问到 `/var/run/docker.sock` ）。

Logtail采集Kubernetes日志请参见采集Kubernetes日志，采集标准容器日志请参见采集标准Docker容器日志。

- Logtail在重启或停止期间，无法采集容器事件。

应用场景

- 监控所有容器的启停事件，当核心容器停止后立即告警。
- 采集所有容器事件，用于审计、安全分析、问题排查。
- 监控所有镜像的拉取事件，如果拉取非合法路径的镜像时立即告警。

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**自定义数据插件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 在**创建机器组**页签中，创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。
   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

```
{
  "inputs": [
    {
      "detail": {},
      "type": "service_docker_event"
    }
  ]
}
```

配置项	类型	是否必须	说明
type	string	是	数据源类型，固定为service_docker_event。
EventQueueSize	int	否	事件缓冲队列大小。不配置时，默认为10，无特殊需求请保持默认设置。

后续步骤

Logtail采集Docker事件到日志服务后，您可以在日志服务控制台查看日志，日志样例如下所示。

- 样例1：镜像拉取事件

```
__source__:  10.10.10.10
__tag__:__hostname__:  logtail-ds-77brr
__topic__:
_action_:  pull
_id_:  registry.cn-hangzhou.aliyuncs.com/ringtail/eventer:v1.6.1.3
_time_nano_:  1547910184047414271
_type_:  image
name:  registry.cn-hangzhou.aliyuncs.com/ringtail/eventer
```

- 样例2：Kubernetes中容器的销毁事件

```
__source__:  10.10.10.10
__tag__:__hostname__:  logtail-ds-xnvz2
__topic__:
_action_:  destroy
_id_:  af61340b0ac19e6f5f32be672d81a33fc4d3d247bf7dbd4d3b2c030b8bec4a03
_time_nano_:  1547968139380572119
_type_:  container
annotation.kubernetes.io/config.seen:  2019-01-20T15:03:03.114145184+08:00
annotation.kubernetes.io/config.source:  api
annotation.scheduler.alpha.kubernetes.io/critical-pod:
controller-revision-hash:  2630731929
image:  registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0
io.kubernetes.container.name:  POD
io.kubernetes.docker.type:  podsandbox
io.kubernetes.pod.name:  logtail-ds-44jbg
io.kubernetes.pod.namespace:  kube-system
io.kubernetes.pod.uid:  6ddcf598-1c81-11e9-9ddf-00163e0c7cbe
k8s-app:  logtail-ds
kubernetes.io/cluster-service:  true
name:  k8s_POD_logtail-ds-44jbg_kube-system_6ddcf598-1c81-11e9-9ddf-00163e0c7cbe_0
pod-template-generation:  9
version:  v1.0
```

Docker事件的日志字段如下，详细信息请参见Docker官方文档。

字段	说明
type	资源类型，例如container、image。
action	操作类型，例如destroy、status。
_id_	事件唯一标识。
_time_nano_	事件的时间戳。

3.7.9. 采集Windows事件日志

Windows Logtail支持通过插件采集Windows事件日志。本文介绍如何通过日志服务控制台创建Logtail采集配置来采集Windows事件日志。

前提条件

已在服务器上安装Logtail。更多信息，请参见安装Logtail（Windows系统）。

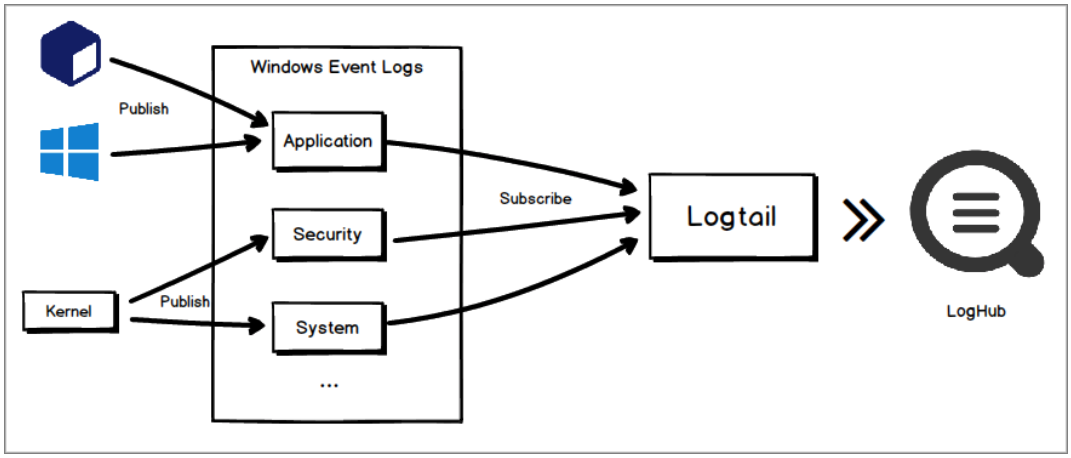
> **说明** 目前仅支持Windows Logtail 1.0.0.0及以上版本。

原理

对于事件日志，Windows提供了Windows Event Log和Event Logging两套API，前者是后者的升级，仅在Windows Vista及以上的版本中提供。Logtail插件会根据所运行的系统，自动选择API（优先选择Windows Event Log）来获取Windows事件日志。

Windows事件日志采用发布订阅的模式，应用程序或者内核将事件日志发布到指定的通道（例如Application、Security、System），Logtail通过对应的Logtail插件调用Windows API，实现对这些通道的订阅，从而不断地获取相关的事件日志并发送到日志服务。

Logtail支持同时采集多个通道事件，例如同时采集应用程序和系统日志。



查看通道信息

您可以在Windows服务器的**事件查看器**中查看通道信息。

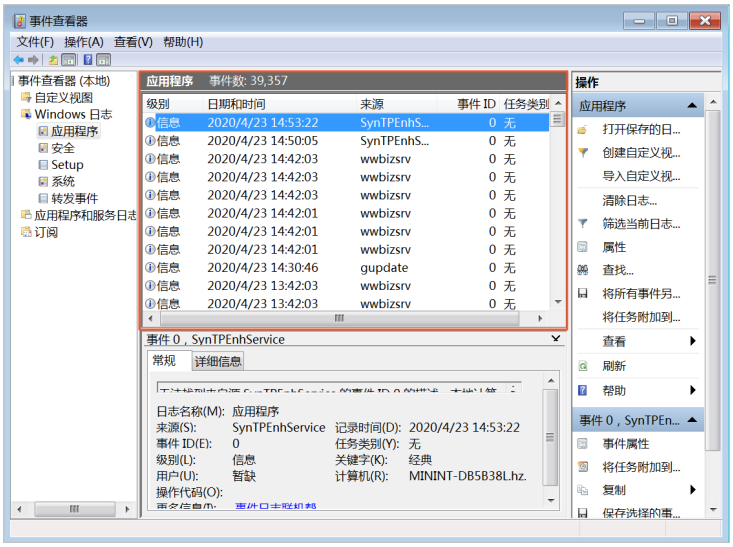
1. 单击**开始**。
2. 搜索并打开**事件查看器**。
3. 在左侧导航栏中展开**Windows 日志**。
4. 查看通道的全名。

   在**Windows日志**下，选择目标通道，右键单击**属性**，查看通道全名。**Windows日志**下常用通道全名如下所示。

   - 应用程序：Application
   - 安全：Security
   - Setup：Setup
   - 系统：System
5. 查看通道相关信息。

   在**Windows日志**下，单击目标通道，在页面中间区域查看事件的级别、日期和时间、来源和事件ID等信息。

   在采集配置中，可根据这些信息进行过滤。



采集步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**自定义数据插件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 在**创建机器组**页签中，创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。

a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

更多信息，请参见安装Logtail（ECS实例）。

> **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

b. 安装完成后，单击**确认安装完毕**。

c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。

5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

> **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 在**数据源设置**页签中，设置**配置名称**和**插件配置**，然后单击**下一步**。
   - inputs为Logtail采集配置，必选项，请根据您的数据源配置。

     > **说明** 一个inputs中只允许配置一个类型的数据源。

   - processors为Logtail处理配置，可选项。您可以配置一种或多种处理方式，详情请参见概述。

   此处以同时采集**应用程序**和**系统**两个通道为例，其中IgnoreOlder设置为3天以避免采集过多的事件日志。

```
{
    "inputs": [
        {
            "type": "service_wineventlog",
            "detail": {
                "Name": "Application",
                "IgnoreOlder": 259200
            }
        },
        {
            "type": "service_wineventlog",
            "detail": {
                "Name": "System",
                "IgnoreOlder": 259200
            }
        }
    ]
}
```

参数	类型	是否必选	说明
type	string	是	数据源类型，固定为service_wineventlog。
Name	string	是	待采集事件日志所属的通道名称，只能指定一个。不配置时，默认为Application，表示采集**应用程序**通道中的事件日志。您可以在Windows系统中查看通道全名，详情请参见步骤 4。
IgnoreOlder	uint	否	根据事件时间过滤日志，此配置是相对于采集开始时间的偏移量，单位为秒，早于此设置的日志会被忽略。 例如： ◦ 设置为3600，表示相对于采集开始时间一小时前的日志都会被忽略。 ◦ 设置为14400，表示相对于采集开始时间四小时前的日志都会被忽略。 默认为空，表示不根据事件时间进行过滤，采集服务器上所有的历史事件日志。 **说明** 该选项仅在首次配置采集时生效，Logtail会记录事件采集的Checkpoint，保证不会重复采集事件日志。

参数	类型	是否必选	说明
Level	string	否	根据事件等级过滤日志，默认值为information, warning, error, critical，表示采集除了verbose等级外的其他所有日志。 可选等级包括：information、warning、error、critical、verbose。您可以使用英文逗号（,）指定多个等级。 **说明** 该参数仅支持Windows Event Log API，即只能在Windows Vista 及以上的操作系统上使用。
EventID	string	否	根据事件ID过滤日志，可以指定正向过滤（单个或范围）或者反向过滤（不支持范围设置）。默认为空，表示采集所有事件。例如： - 1-200表示只采集事件ID在1-200范围内的事件日志。 - 20表示只采集事件ID为20的事件日志。 - -100表示采集除了事件ID为100以外的所有事件日志。 - 1-200,-100表示采集1-200范围内除了100以外的事件日志。 您可以使用英文逗号（,）指定多个值。 **说明** 该参数仅支持Windows Event Log API，即只能在Windows Vista 及以上的操作系统上使用。
Provider	string 数组	否	根据事件来源过滤日志。例如设置为["App1", "App2"] 表示只采集来源名字为App1和App2的事件日志，其他事件日志都会被忽略。 默认为空，表示采集所有来源的事件。 **说明** 该参数仅支持Windows Event Log API，即只能在Windows Vista 及以上的操作系统上使用。
IgnoreZeroValue	boolean	否	并非每条事件日志都拥有所有的字段，您可以使用此参数过滤空字段，空字段的定义根据类型而定，例如整数类型使用0表示空字段。 默认为false，表示不过滤空字段。

后续步骤

采集Windows至日志服务后，您可以在日志服务控制台上查看日志。

字段名	字段类型	说明
activity_id	string	表示当前事件所属活动的全局事务ID，同一个活动的事件具有相同的全局事务ID。
computer_name	string	产生当前事件的节点名。
event_data	JSON object	和当前事件相关的数据。
event_id	int	当前事件的ID。
kernel_time	int	当前事件消耗的内核时间，一般为0。
keywords	JSON array	当前事件关联的关键字，用于事件分类。
level	string	当前事件的等级。
log_name	string	获取当前事件的通道名，即logtail采集配置中Name参数。
message	string	当前事件关联的消息。
message_error	string	在解析当前事件关联消息时发生的错误信息。
opcode	string	当前事件关联的操作码。
process_id	int	当前事件的进程ID。
processor_id	int	当前事件对应的处理器ID，一般为0。
processor_time	int	当前事件消耗的处理器时间，一般为0。
provider_guid	string	当前事件来源的全局事务ID。
record_number	int	当前事件关联的记录编号。事件的记录编号会随着每条事件的写入递增，当超过2^{32}（Event Logging）或2^{64}（Windows Event Log）后会重新从0开始。
related_activity_id	string	当前事件所属活动关联的其他活动的全局事务ID。
session_id	int	当前事件的会话ID，一般为0。
source_name	string	当前事件的来源，即logtail采集配置中Provider参数。
task	string	当前事件关联的任务。
thread_id	int	当前事件的线程ID。
type	string	获取当前事件使用的API。
user_data	JSON object	当前事件关联的用户数据。
user_domain	string	当前事件关联的用户的域。
user_identifier	string	当前事件关联的用户Windows安全标识（Security Identifier）。
user_name	string	当前事件关联的用户名。
user_time	int	当前事件消耗的用户态时间，一般为0。
user_type	string	当前事件关联的用户的类型。
version	int	当前事件的版本号。
xml	string	当前事件最原始的信息，XML格式。

3.8. 使用Logtail插件处理数据

3.8.1. 概述

当您的业务日志太复杂或不固定，固定解析模式（Nginx模式、完整正则模式、JSON模式等）无法满足日志解析需求时，您可以使用Logtail插件解析日志。您可添加一个或多个Logtail插件处理配置，Logtail会根据处理配置顺序逐一执行。

使用限制

- 性能限制

  使用Logtail插件进行数据处理时，Logtail会消耗更多的资源（以CPU为主），请根据实际情况调整Logtail的参数配置。更多信息，请参见设置Logtail启动参数。当原始数据量的生成速度超过5 MB/s时，不建议您使用过于复杂的插件组合来处理数据，您可以使用Logtail插件进行简单处理，再通过数据加工完成进一步处理。

- 文本日志限制

  在固定解析模式基础上，日志服务支持通过插件处理采集到的文本日志，但存在以下限制：

  - 启用插件处理后，部分文本模式的高级功能将失效，包括过滤器配置、上传原始日志、机器时区、丢弃解析失败日志、接受部分字段（分隔符模式）等，但其中部分功能可通过相关插件实现。
  - 插件对文本日志的处理采用行模式，即文件级别的元数据（例如__tag__:__path__、__topic__等）会被存放到每条日志中，启用插件处理后会影响和Tag相关的功能，包括：
    - 无法使用上下文查询、LiveTail功能，这些功能依赖于__tag__:__path__等字段。
    - __topic__字段会被重命名为__log_topic__。
    - __tag__:__path__等字段不再具备原生字段索引，需单独创建索引。

配置说明

处理配置的Key为processors，Value为JSON Object数组，数组内每个Object代表一个处理配置。

一个处理配置包含两个字段：type和detail，其中type为该处理配置所使用的Logtail插件名称，detail为详细配置，示例如下：

```
"processors" : [
    {
        "type": "processor_split_char",
        "detail": {
            "SourceKey": "content",
            "SplitSep": "|",
            "SplitKeys": [
                "method",
                "type",
                "ip",
                "time",
                "req_id",
                "size",
                "detail"
            ]
        }
    },
    {
        "type": "processor_anchor",
        "detail": {
            "SourceKey": "detail",
            "Anchors": [
                {
                    "Start": "appKey=",
                    "Stop": ",env=",
                    "FieldName": "appKey",
                    "FieldType": "string"
                }
            ]
        }
    }
]
```

您可以使用Logtail插件完成如下操作。

Logtail插件	说明
processor_regex	使用processor_regex插件（正则模式）提取字段。更多信息，请参见正则模式。
processor_anchor	使用processor_anchor插件（标定模式）提取字段。更多信息，请参见标定模式。
processor_split_char	使用processor_split_char插件（单字符分隔符模式）提取字段。更多信息，请参见单字符分隔符模式。
processor_split_string	使用processor_split_string插件（多字符分隔符模式）提取字段。更多信息，请参见多字符分隔符模式。
processor_split_key_value	使用processor_split_key_value插件（键值对模式）提取字段。更多信息，请参见键值对模式。

Logtail插件	说明
processor_add_fields	使用processor_add_fields插件添加字段。更多信息，请参见添加字段。
processor_drop	使用processor_drop插件丢弃字段。更多信息，请参见丢弃字段。
processor_rename	使用processor_rename插件重命名字段。更多信息，请参见重命名字段。
processor_packjson	使用processor_packjson插件将一个或多个字段打包为一个JSON Object格式的字段。更多信息，请参见打包字段。
processor_json	使用processor_json插件对字段值进行JSON展开。更多信息，请参见展开JSON字段。
processor_filter_regex	使用processor_filter_regex插件过滤日志。更多信息，请参见过滤日志。
processor_gotime	使用processor_gotime插件（Go语言时间格式）解析原始日志中的时间字段，并可将解析结果设置为日志时间。更多信息，请参见Go语言时间格式。
processor_strptime	使用processor_strptime插件（strptime时间格式）解析原始日志中的时间字段，并可将解析结果设置为日志时间。更多信息，请参见strptime时间格式。
processor_geoip	使用processor_geoip插件将数据中的IP地址转换为地理位置（国家、省份、城市、经纬度）。更多信息，请参见转换IP地址。

3.8.2. 提取字段

您可以通过正则模式、标定模式、单字符分隔符模式、多字符分隔符模式、键值对模式提取日志字段。本文介绍各个模式的参数说明和配置示例。

正则模式

通过正则表达式提取目标字段。

- 参数说明

  配置type为processor_regex，detail说明如下表所示。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
Regex	String	是	正则表达式，使用`()`标注待提取的字段。
Keys	String数组	是	提取的字段名，例如["ip", "time", "method"]。
NoKeyError	Boolean	否	无匹配的原始字段时是否报错。如果未添加该参数，则默认使用false，表示不报错。
NoMatchError	Boolean	否	正则表达式与原始字段的值不匹配时是否报错。如果未添加该参数，则默认使用false，表示不报错。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用false，表示不保留。
FullMatch	Boolean	否	如果未添加该参数，则默认使用true，表示只有字段完全匹配Regex参数中的正则表达式时才被提取。配置为false，表示部分字段匹配也会进行提取。

- 配置示例

  使用正则模式提取content字段的值，并设置字段名为ip、time、method、url、request_time、request_length、status、length、ref_url和browser。配置示例如下：

  - 原始数据

```
"content" : "10.200.**.** - - [10/Aug/2017:14:57:51 +0800] \"POST /PutData?Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature=<yourSignature> HTTP/1.1\" 0.024 18204 200 37 \"-\" \"aliyun-sdk-java"
```

- Logtail插件处理配置

```
{
    "type" : "processor_regex",
    "detail" : {"SourceKey" : "content",
         "Regex" : "([\\d\\.]+) \\S+ \\S+ \\[(\\S+) \\S+\\] \"(\\w+) ([^\\\"]*)\" ([\\d\\.]+) (\\d+) (\\d+) (\\d+|-) \
"([^\\\"]*)\" \"([^\\\"]*)\" (\\d+)",
         "Keys"    : ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "
browser"],
         "NoKeyError" : true,
         "NoMatchError" : true,
         "KeepSource" : false
    }
}
```

- 处理结果

```
"ip" : "10.200.**.**"
"time" : "10/Aug/2017:14:57:51"
"method" : "POST"
"url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A3
0%20GMT&Topic=raw&Signature=<yourSignature>"
"request_time" : "0.024"
"request_length" : "18204"
"status" : "200"
"length" : "27"
"ref_url" : "-"
"browser" : "aliyun-sdk-java"
```

标定模式

通过标定起始和结束关键字进行字段提取。如果是JSON类型的字段，您可以进行JSON展开。

- 参数说明

  配置type为processor_anchor，detail说明如下表所示。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
Anchors	Anchor数组	是	标定项列表。
NoAnchorError	Boolean	否	查找不到关键字时是否报错。如果未添加该参数，则默认使用false，不报错。
NoKeyError	Boolean	否	无匹配的原始字段时是否报错。如果未添加该参数，则默认使用false，不报错。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用false，表示不保留。

  其中，Anchors参数详细说明如下表所示。

参数	类型	是否必选	说明
Start	String	是	起始关键字。如果为空，表示匹配字符串开头。
Stop	String	是	结束关键字，如果为空，表示匹配字符串结尾。
FieldName	String	是	提取的字段名。
FieldType	String	是	字段的类型，取值为string或json。
ExpondJson	Boolean	否	是否进行JSON展开。如果未添加该参数，则默认使用false，表示不展开。 仅当FieldType为json时生效。
ExpondConnecter	String	否	JSON展开的连接符。如果未添加该参数，则默认使用下划线（_）。

参数	类型	是否必选	说明
MaxExpondDepth	Int	否	JSON展开最大深度。如果未添加该参数，则默认为0，表示无限制。

- 配置示例

  使用标定模式提取content字段的值，并设置字段名为time、val_key1、val_key2、val_key3、value_key4_inner1、value_key4_inner2。配置示例如下：

  - 原始数据

    ```
    "content" : "time:2017.09.12 20:55:36\tjson:{\"key1\" : \"xx\", \"key2\": false, \"key3\":123.456, \"key4\" : { \"inner1\" : 1, \"inner2\" : false}}"
    ```

  - Logtail插件处理配置

    ```
    {
       "type" : "processor_anchor",
       "detail" : {"SourceKey" : "content",
          "Anchors" : [
              {
                  "Start" : "time",
                  "Stop" : "\t",
                  "FieldName" : "time",
                  "FieldType" : "string",
                  "ExpondJson" : false
              },
              {
                  "Start" : "json:",
                  "Stop" : "",
                  "FieldName" : "val",
                  "FieldType" : "json",
                  "ExpondJson" : true
              }
          ]
      }
    }
    ```

  - 处理结果

    ```
    "time" : "2017.09.12 20:55:36"
    "val_key1" : "xx"
    "val_key2" : "false"
    "val_key3" : "123.456"
    "value_key4_inner1" : "1"
    "value_key4_inner2" : "false"
    ```

单字符分隔符模式

通过单字符的分隔符提取字段。该方式支持使用引用符对分隔符进行包裹。

- 参数说明

  配置type为processor_split_char，detail说明如下表所示。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
SplitSep	String	是	分隔符。必须为单字符，可设置为不可见字符，例如\u0001。
SplitKeys	String数组	是	分割日志后设置的字段名，例如["ip", "time", "method"]。
QuoteFlag	Boolean	否	是否使用引用符。如果未添加该参数，则默认使用false，表示不使用。
Quote	String	否	引用符。必须为单字符，可以为不可见字符，例如\u0001。 仅当QuoteFlag配置为true时有效。

参数	类型	是否必选	说明
NoKeyError	Boolean	否	无匹配的原始字段时是否报错。如果未添加该参数，则默认使用false，表示不报错。
NoMatchError	Boolean	否	分隔符不匹配时是否报错。如果未添加该参数，则默认使用false，表示不报错。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用false，表示不保留。

- 示例

  使用竖线（|）分隔符提取content字段的值，并设置字段名为ip、time、method、url、request_time、request_length、status、length、ref_url和browser。配置示例如下：

  - 原始日志

    ```
    "content" : "10.**.**.**|10/Aug/2017:14:57:51 +0800|POST|PutData?
    Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
    Signature=<yourSignature>|0.024|18204|200|37|-|
    aliyun-sdk-java"
    ```

  - Logtail处理插件配置

    ```
    {
       "type" : "processor_split_char",
       "detail" : {"SourceKey" : "content",
          "SplitSep" : "|",
          "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status", "length", "ref_url", "
    browser"]
      }
    }
    ```

  - 处理结果

    ```
    "ip" : "10.**.**.**"
    "time" : "10/Aug/2017:14:57:51 +0800"
    "method" : "POST"
    "url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A3
    0%20GMT&Topic=raw&Signature=<yourSignature>"
    "request_time" : "0.024"
    "request_length" : "18204"
    "status" : "200"
    "length" : "27"
    "ref_url" : "-"
    "browser" : "aliyun-sdk-java"
    ```

多字符分隔符模式

通过多字符的分隔符提取字段。该方式不支持指定引用符对分隔符进行包裹。

- 参数说明

  配置type为processor_split_string，detail说明如下表所示。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
SplitSep	String	是	分隔符。您可设置不可见字符，例如\u0001\u0002。
SplitKeys	String数组	是	分割后的字段名，例如["key1","key2"]。
PreserveOthers	Boolean	否	如果待分割的字段长度大于SplitKeys参数中的字段长度时是否保留超出部分。如果未添加该参数，则默认使用false，表示不保留。
ExpandOthers	Boolean	否	是否解析超出部分。如果未添加该参数，则默认使用false，表示不继续解析。
ExpandKeyPrefix	String	否	超出部分的命名前缀。例如配置expand_，则Key为expand_1、expand_2。

参数	类型	是否必选	说明
NoKeyError	Boolean	否	无匹配的原始字段时是否报错。如果未添加该参数，则默认使用false，表示不报错。
NoMatchError	Boolean	否	分隔符不匹配时是否报错。如果未添加该参数，则默认使用false，表示不报错。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用false，表示不保留。

- 示例

  使用分隔符|#|提取content字段的值，并设置字段名为ip、time、method、url、request_time、request_length和status、expand_1、expand_2和expand_3。配置示例如下：

  - 原始日志

    ```
    "content" : "10.**.**.**|#|10/Aug/2017:14:57:51 +0800|#|POST|#|PutData?
    Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&
    Signature=<yourSignature>|#|0.024|#|18204|#|200|#|27|#|-|#|
    aliyun-sdk-java"
    ```

  - Logtail插件处理配置

    ```
    {
       "type" : "processor_split_string",
       "detail" : {"SourceKey" : "content",
          "SplitSep" : "|#|",
          "SplitKeys" : ["ip", "time", "method", "url", "request_time", "request_length", "status"],
          "PreserveOthers" : true,
          "ExpandOthers" : true,
          "ExpandKeyPrefix" : "expand_"
      }
    }
    ```

  - 处理结果

    ```
    "ip" : "10.**.**.**"
    "time" : "10/Aug/2017:14:57:51 +0800"
    "method" : "POST"
    "url" : "/PutData?Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A3
    0%20GMT&Topic=raw&Signature=<yourSignature>"
    "request_time" : "0.024"
    "request_length" : "18204"
    "status" : "200"
    "expand_1" : "27"
    "expand_2" : "-"
    "expand_3" : "aliyun-sdk-java"
    ```

键值对模式

通过切分键值对的方式提取字段。

- 参数说明

  配置type为processor_split_key_value，detail说明如下表所示。

  > **说明** Logtail 0.16.26及以上版本支持此插件。

参数	类型	是否必选	说明
SourceKey	string	是	原始字段名。
Delimiter	string	否	键值对之间的分隔符。如果未添加该参数，则默认使用制表符 `\t` 。
Separator	string	否	单个键值对中键与值之间的分隔符。如果未添加该参数，则默认使用冒号（:）。
KeepSource	Boolean	否	提取完毕后是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。

参数	类型	是否必选	说明
ErrIfSourceKeyNotFound	Boolean	否	无匹配的原始字段时是否告警。如果未添加该参数，则默认使用true，表示告警。
DiscardWhenSeparatorNotFound	Boolean	否	无匹配的原始字段时是否丢弃该键值对。如果未添加该参数，则默认使用false，表示不丢弃。
ErrIfSeparatorNotFound	Boolean	否	当指定的分隔符（Separator）不存在时是否告警。如果未添加该参数，则默认使用true，表示告警。

- 配置示例

  按照键值对方式分割content字段的值。其中，键值对间分隔符为制表符`/t`，键值对中的分隔符为冒号（:）。配置示例如下：

  - 原始数据

    ```
    "content": "class:main\tuserid:123456\tmethod:get\tmessage:\"wrong user\""
    ```

  - Logtail插件处理配置

    ```
    {
      "processors":[
        {
          "type":"processor_split_key_value",
          "detail": {
            "SourceKey": "content",
            "Delimiter": "\t",
            "Separator": ":",
            "KeepSource": true
          }
        }
      ]
    }
    ```

  - 处理结果

    ```
    "content": "class:main\tuserid:123456\tmethod:get\tmessage:\"wrong user\""
    "class": "main"
    "userid": "123456"
    "method": "get"
    "message": "\"wrong user\""
    ```

3.8.3. 添加字段

您可以使用processor_add_fields插件添加日志字段。本文介绍processor_add_fields插件的参数说明和配置示例。

参数说明

配置type为processor_add_fields，detail说明如下表所示。

> **说明** Logtail 0.16.28及以上版本支持该插件。

参数	类型	是否必选	说明
Fields	Map	否	待添加字段和字段值。键值对格式，支持添加多个。
IgnoreIfExist	Boolean	否	当Key相同时是否忽略。如果未添加该参数，则默认使用false，表示不忽略。

配置示例

添加aaa2字段和aaa3字段，配置示例如下：

- 原始日志

  ```
  "aaa1":"value1"
  ```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_add_fields",
      "detail": {
        "Fields": {
          "aaa2": "value2",
          "aaa3": "value3"
        }
      }
    }
  ]
}
```

- 处理结果

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

3.8.4. 丢弃字段

您可以使用processor_drop插件丢弃日志字段。本文介绍processor_drop插件的参数说明和配置示例。

参数说明

配置type为processor_drop，detail说明如下表所示。

> **说明** Logtail 0.16.28及以上版本支持该插件。

参数	类型	是否必选	说明
DropKeys	String数组	否	指定待丢弃的字段，支持配置多个。

配置示例

丢弃日志中的aaa1字段和aaa2字段，配置示例如下：

- 原始日志

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_drop",
      "detail": {
        "DropKeys": ["aaa1","aaa2"]
      }
    }
  ]
}
```

- 处理结果

```
"aaa3":"value3"
```

3.8.5. 重命名字段

您可以使用processor_rename插件重命名字段。本文介绍processor_rename插件的参数说明和配置示例。

参数说明

配置type为processor_rename，detail说明如下表所示。

> **说明** Logtail 0.16.28及以上版本支持此插件。

参数	类型	是否必选	说明
NoKeyError	Boolean	是	无匹配字段时是否报错。如果未添加该参数，则默认使用false，表示不报错。
SourceKeys	String数组	是	待重命名的原始字段。
DestKeys	String数组	是	重命名后的字段。

配置示例

将aaa1字段重命名为bbb1，将aaa2字段重命名bbb2，配置示例如下：

- 原始日志

```
"aaa1":"value1"
"aaa2":"value2"
"aaa3":"value3"
```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_rename",
      "detail": {
        "SourceKeys": ["aaa1","aaa2"],
        "DestKeys": ["bbb1","bbb2"],
        "NoKeyError": true
      }
    }
  ]
}
```

- 处理结果

```
"bbb1":"value1"
"bbb2":"value2"
"aaa3":"value3"
```

3.8.6. 打包字段

您可以使用processor_packjson插件将一个或多个字段打包为JSON Object格式的字段。本文介绍processor_packjson插件的参数说明和配置示例。

参数说明

配置type为processor_packjson，detail说明如下表所示。

> **说明** Logtail 0.16.28及以上版本支持此插件。

参数	类型	是否必选	说明
SourceKeys	String数组	是	待打包的原始字段。
DestKey	String	否	打包后的JSON字段。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。
AlarmIfIncomplete	Boolean	否	不存在任何原始字段时是否告警。如果未添加该参数，则默认使用true，表示告警。

配置示例

将指定的a字段和b字段打包成JSON字段d_key，配置示例如下：

- 原始日志

```
"a":"1"
"b":"2"
```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_packjson",
      "detail": {
        "SourceKeys": ["a","b"],
        "DestKey":"d_key",
        "KeepSource":true,
        "AlarmIfEmpty":true
      }
    }
  ]
}
```

- 处理结果

```
"a":"1"
"b":"2"
"d_key":"{\"a\":\"1\",\"b\":\"2\"}"
```

3.8.7. 展开JSON字段

您可以使用processor_json插件展开JSON字段。本文介绍processor_json插件的参数说明和配置示例。

参数说明

配置type为processor_json，detail说明如下表所示。

> **说明** Logtail 0.16.28及以上版本支持该插件。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
NoKeyError	Boolean	否	无匹配字段时是否报错。如果未添加该参数，则默认使用true，表示报错。
ExpandDepth	Int	否	JSON展开的深度。如果未添加该参数，则默认为0，表示不限制。1表示当前层级，以此类推。
ExpandConnector	String	否	JSON展开时的连接符，可以为空。如果未添加该参数，则默认使用下划线（_）。
Prefix	String	否	JSON展开时对字段附加的前缀。如果未添加该参数，则默认为空。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。
UseSourceKeyAsPrefix	Boolean	否	是否将原始字段名作为所有JSON展开字段名的前缀。如果未添加该参数，则默认使用false，表示否。
KeepSourceIfParseError	Boolean	否	解析失败时，是否保留原始日志。如果未添加该参数，则默认使用true，表示保留原始日志。

配置示例

对s_key字段进行JSON展开，并使用j和原始字段名s_key作为JSON展开后字段名的前缀。配置示例如下：

- 原始日志

```
"s_key":"{\"k1\":{\"k2\":{\"k3\":{\"k4\":{\"k51\":\"51\",\"k52\":\"52\"},\"k41\":\"41\"}}})"
```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_json",
      "detail": {
        "SourceKey": "s_key",
        "NoKeyError":true,
        "ExpandDepth":0,
        "ExpandConnector":"-",
        "Prefix":"j",
        "KeepSource": false,
        "UseSourceKeyAsPrefix": true
      }
    }
  ]
}
```

- 处理结果

```
"js_key-k1-k2-k3-k4-k51":"51"
"js_key-k1-k2-k3-k4-k52":"52"
"js_key-k1-k2-k3-k41":"41"
```

3.8.8. 过滤日志

您可以使用processor_filter_regex插件过滤日志。本文介绍processor_filter_regex插件的参数说明和配置示例。

参数说明

配置type为processor_filter_regex，detail说明如下表所示。

> **说明** 一条日志只有完全匹配Include中的正则表达式，且不匹配Exclude中的正则表达式时才会被采集，否则直接丢弃。

参数	类型	是否必选	说明
Include	JSON Object	否	Key为日志字段，Value为该字段值匹配的正则表达式。Key之间为与关系。如果日志中所有字段的值符合对应的正则表达式，则采集该日志。
Exclude	JSON Object	否	Key为日志字段，Value为该字段值匹配的正则表达式。Key之间为或关系。如果日志中任意一个字段的值符合对应的正则表达式，则不采集该日志。

配置示例

只采集ip以10开头、method为POST且browser不为aliyun.*的日志，配置示例如下：

- 原始日志
  - 日志1

    ```
    "ip" : "10.**.**.**"
    "method" : "POST"
    "browser" : "aliyun-sdk-java"
    ```

  - 日志2

    ```
    "ip" : "10.**.**.**"
    "method" : "POST"
    "browser" : "chrome"
    ```

  - 日志3

    ```
    "ip" : "192.168.*.*"
    "method" : "POST"
    "browser" : "ali-sls-ilogtail"
    ```

- Logtail插件处理配置

```
{
   "type" : "processor_filter_regex",
    "detail" : {
         "Include" : {
            "ip" : "10\\..*",
            "method" : "POST"
         },
         "Exclude" : {
            "browser" : "aliyun.*"
         }
    }
}
```

- 处理结果

日志	是否采集	原因
日志1	不采集	browser字段的值匹配Exclude中的正则表达式。
日志2	采集	符合条件。
日志3	不采集	ip字段的值不匹配Include中的正则表达式。

3.8.9. 提取日志时间

您可以使用processor_gotime插件或processor_strptime插件解析原始日志中的时间字段。本文介绍两种插件的参数说明和配置示例。

Go语言时间格式

processor_gotime插件使用Go语言时间格式解析原始日志中的时间字段，并支持将解析结果设置为日志服务中的日志时间。

- 参数说明

  配置type为processor_gotime，detail说明如下表所示。

  > **说明** Logtail 0.16.28及以上版本支持该插件。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
SourceFormat	String	是	原始时间的格式。
SourceLocation	Int	是	原始时间的时区。参数值为空时，表示Logtail所在主机或容器的时区。
DestKey	String	是	解析后的目标字段。
DestFormat	String	是	解析后的时间格式。
DestLocation	Int	否	解析后的时区。参数值为空时，表示本机时区。
SetTime	Boolean	否	是否将解析后的时间设置为日志时间。如果未添加该参数，则默认使用true，表示设置。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。
NoKeyError	Boolean	否	无匹配的原始字段时是否报错。如果未添加该参数，则默认使用true，表示报错。
AlarmIfFail	Boolean	否	提取失败是否告警。如果未添加该参数，则默认使用true，表示告警。

- 示例

  原始时间（s_key字段）的格式为`2006-01-02 15:04:05（东八区）`，现将原始时间解析为`2006/01/02 15:04:05（东九区）`格式，添加到d_key字段中，并设置解析结果为日志服务中的日志时间。

- 原始日志

```
"s_key":"2019-07-05 19:28:01"
```

- 配置详情

```
{
  "processors":[
    {
      "type":"processor_gotime",
      "detail": {
        "SourceKey": "s_key",
        "SourceFormat":"2006-01-02 15:04:05",
        "SourceLocation":8,
        "DestKey":"d_key",
        "DestFormat":"2006/01/02 15:04:05",
        "DestLocation":9,
        "SetTime": true,
        "KeepSource": true,
        "NoKeyError": true,
        "AlarmIfFail": true
      }
    }
  ]
}
```

- 处理结果

```
"s_key":"2019-07-05 19:28:01"
"d_key":"2019/07/05 20:28:01"
```

strptime时间格式

processor_strptime插件使用Linux strptime时间格式解析日志中的时间字段，并支持将解析结果设置为日志时间。

- 参数说明

  配置type为processor_strptime，detail说明如下表所示。

  > **说明** Logtail 0.16.28及以上版本支持该插件。

参数	类型	是否必选	说明
SourceKey	String	是	原始字段名。
Format	String	是	原始时间的格式。
AdjustUTCOffset	Boolean	否	是否调整时区。如果未添加该参数，则默认使用false，表示不调整。
UTCOffset	Int	否	用于调整的时区偏移秒数。例如28800表示东八区。
AlarmIfFail	Boolean	否	提取失败时是否告警。如果未添加该参数，则默认使用true，表示进行告警。
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。
EnablePreciseTimestamp	Boolean	否	是否提取高精度时间。如果未添加该参数，则默认使用false，表示不提取高精度时间。 默认情况下，设置为true后，该插件会将SourceKey参数对应的字段值解析为毫秒级别的时间戳，并存入PreciseTimestampKey参数对应的字段。 **说明** - 设置该参数为true前，请确保SourceKey参数对应的字段值（原始时间）支持相应的时间精度（ms、us或ns）。 - 仅Logtail 1.0.32及以上版本支持。

参数	类型	是否必选	说明
PreciseTimestampKey	String	否	保存高精度时间戳的字段。如果未添加该参数，则默认为precise_timestamp字段。
PreciseTimestampUnit	String	否	高精度时间戳的单位。如果未添加该参数，则默认为ms。取值包括ms（毫秒）、us（微秒）、ns（纳秒）。

- 示例

  将 `%Y/%m/%d %H:%M:%S` 格式的原始时间（log_time字段的值）解析为对应的日志时间，时区使用机器所在时区。

  - 示例1：假设时区为东八区。
    - 原始日志

      ```
      "log_time":"2016/01/02 12:59:59"
      ```

    - Logtail插件处理配置

      ```
      {
        "processors":[
          {
            "type":"processor_strptime",
            "detail": {
              "SourceKey": "log_time",
              "Format": "%Y/%m/%d %H:%M:%S"
            }
          }
        ]
      }
      ```

    - 处理结果

      ```
      "log_time":"2016/01/02 12:59:59"
      Log.Time = 1451710799
      ```

  - 示例2：假设时区为东七区。
    - 原始日志

      ```
      "log_time":"2016/01/02 12:59:59"
      ```

    - Logtail插件处理配置

      ```
      {
        "processors":[
          {
            "type":"processor_strptime",
            "detail": {
              "SourceKey": "log_time",
              "Format": "%Y/%m/%d %H:%M:%S",
              "AdjustUTCOffset": true,
              "UTCOffset": 25200
            }
          }
        ]
      }
      ```

    - 处理结果

      ```
      "log_time":"2016/01/02 12:59:59"
      Log.Time = 1451714399
      ```

- 示例3：假设时区为东七区。
  - 原始日志

    ```
    "log_time":"2016/01/02 12:59:59.123"
    ```

  - Logtail插件处理配置

    ```
    {
      "processors":[
        {
          "type":"processor_strptime",
          "detail": {
            "SourceKey": "log_time",
            "Format": "%Y/%m/%d %H:%M:%S.%f",
            "EnablePreciseTimestamp": true
          }
        }
      ]
    }
    ```

  - 处理结果

    ```
    "log_time":"2016/01/02 12:59:59.123"
    "precise_timestamp": 1451714399123
    Log.Time = 1451714399
    ```

- 常见的时间表达式

  > **说明** processor_strptime插件支持%f格式解析，表示秒的小数部分，最高精度为纳秒。

示例	时间表达式
2016/01/02 12:59:59	%Y/%m/%d %H:%M:%S
2016/01/02 12:59:59.1	%Y/%m/%d %H:%M:%S.%f
2016/01/02 12:59:59.987654321 +0700 (UTC)	%Y/%m/%d %H:%M:%S.%f %z (%Z)
2016/Jan/02 12:59:59,123456	%Y/%b/%d %H:%M:%S,%f
2019-07-15T04:16:47:123Z	%Y-%m-%dT%H:%M:%S:%f

3.8.10. 转换IP地址

您可以使用processor_geoip插件将日志中的IP地址转换为地理位置（国家、省份、城市、经纬度）。本文介绍processor_geoip插件的参数说明和配置示例。

参数说明

配置type为processor_geoip，detail说明如下表所示。

> **说明**
> - Logtail安装包中没有GeoIP数据库，您需要手动下载GeoIP数据库到Logtail所在服务器并配置，建议下载精确到City粒度的数据库。更多信息，请参见MaxMind GeoLite2。
> - 请确保数据库格式为MMDB类型。

参数	类型	是否必选	说明
SourceKey	String	是	待进行IP地址转换的原始字段名。
DBPath	String	是	GeoIP数据库的全路径。例如*/user/data/GeoLite2-City_20180102/GeoLite2-City.mmdb*。
NoKeyError	Boolean	否	无匹配的原始字段名时是否报错。如果未添加该参数，则默认使用false，表示不报错。
NoMatchError	Boolean	否	IP地址无效或在数据库中未匹配到该IP地址时是否报错。如果未添加该参数，则默认使用false，表示不报错。

参数	类型	是否必选	说明
KeepSource	Boolean	否	是否保留原始字段。如果未添加该参数，则默认使用true，表示保留。
Language	String	否	语言属性。如果未添加该参数，则默认使用zh-CN。请确保您的GeoIP数据库中包含相应的语言。

配置示例

将日志中的IP地址转换成对应的地理位置，示例如下：

- 原始日志

```
"source_ip" : "**.**.**.**"
```

- Logtail插件处理配置

```
{
   "type": "processor_geoip",
    "detail": {
         "SourceKey": "ip",
         "NoKeyError": true,
         "NoMatchError": true,
         "KeepSource": true,
         "DBPath" : "/user/local/data/GeoLite2-City_20180102/GeoLite2-City.mmdb"
    }
}
```

- 处理结果

```
"source_ip_city_" : "**.**.**.**"
"source_ip_province_" : "浙江省"
"source_ip_city_" : "杭州"
"source_ip_province_code_" : "ZJ"
"source_ip_country_code_" : "CN"
"source_ip_longitude_" : "120.********"
"source_ip_latitude_" : "30.********"
```

3.8.11. 追加字段

您可以使用processor_appender插件为指定的字段（可以为不存在的字段）追加特定的值，支持在字段值中添加模板变量。该插件通常与input_prometheus、input_system_v2等时序监控相关的插件结合使用，用于给拉取到的Prometheus数据追加特定的值。

> 注意 Logtail 0.16.66及以上版本支持processor_appender插件。

参数说明

配置type为*processor_appender*，detail说明如下表所示。

插件说明

参数	类型	是否必选	参数说明
Key	string	是	字段名称。
Value	string	是	添加的字段值。日志服务支持在该字段值中添加模板变量。更多信息，请参见模板变量。
SortLabels	boolean	否	如果您要添加__labels__字段，即配置Key为*__labels__*，则需要设置SortLabels为*true*，用于对Labels进行重新排序，避免因为Labels不遵循字母序而导致查询异常。该值默认为false。

模板变量

模板变量	说明	配置示例	结果示例
{{__ip__}}	替换为Logtail所在服务器的IP地址。	"Value": "{{__ip__}}"	"Value": "192.0.2.1"

模板变量	说明	配置示例	结果示例
{{__host__}}	替换为Logtail所在服务器的主机名。	"Value": "{{__host__}}"	"Value": "logtail-ds-xdfaf"
{{$xxxx}}	通过环境变量引用，需以美元符号（$）开头。替换为环境变量的取值。	"Value": "{{$WORKING_GROUP}}"	"Value": "prod"

示例

例如Logtail所在服务器的IP地址为192.0.2.1，主机名为david，存在环境变量WORKING_GROUP的值为prod。如果您需要为__labels__字段添加以上数据，可参见如下配置：

- 原始数据

```
"__labels__":"a#$#b"
```

- Logtail插件处理配置

```
{
  "processors":[
    {
      "type":"processor_appender",
      "detail": {
        "Key": "__labels__",
        "Value": "|host#$#{{__host__}}|ip#$#{{__ip__}}|group#$#{{$WORKING_GROUP}}",
        "SortLabels": true
      }
    }
  ]
}
```

- 处理结果

```
"__labels__":"a#$#b|group#$#prod|host#$#david|ip#$#192.0.2.1"
```

3.9. 使用内置的Logtail告警监控规则

日志服务已内置告警监控规则，您只需开启对应的告警实例即可实时监控Logtail，并可通过钉钉等渠道接收到告警通知。本文介绍使用Logtail内置告警监控规则的操作步骤。

前提条件

已为目标Project开启重要日志功能。具体操作，请参见开通服务日志。

背景信息

当您为目标Project开启重要日志功能后，日志服务会自动在您所选择的Project下创建一个名为**internal-diagnostic_log**的Logstore，用于记录Logtail心跳日志。日志服务基于该日志预设了Logtail告警监控规则，用于实时监控Logtail。

步骤一：配置行动策略

Logtail内置告警监控规则默认绑定SLS Logtail内置行动策略，因此您在开启告警实例前，需先在该行动策略中设置对应的通知渠道。

1. 登录日志服务控制台。
2. 在Project列表中，找到目标Project。

   该Project为您在开启重要日志时，所选择的Project。
3. 在左侧导航栏中，单击**告警**。
4. 在**告警中心**页面，选择**告警管理 > 行动策略**。
5. 在行动策略列表中，找到目标行动策略（sls.app.logtail.builtin），单击**修改**。

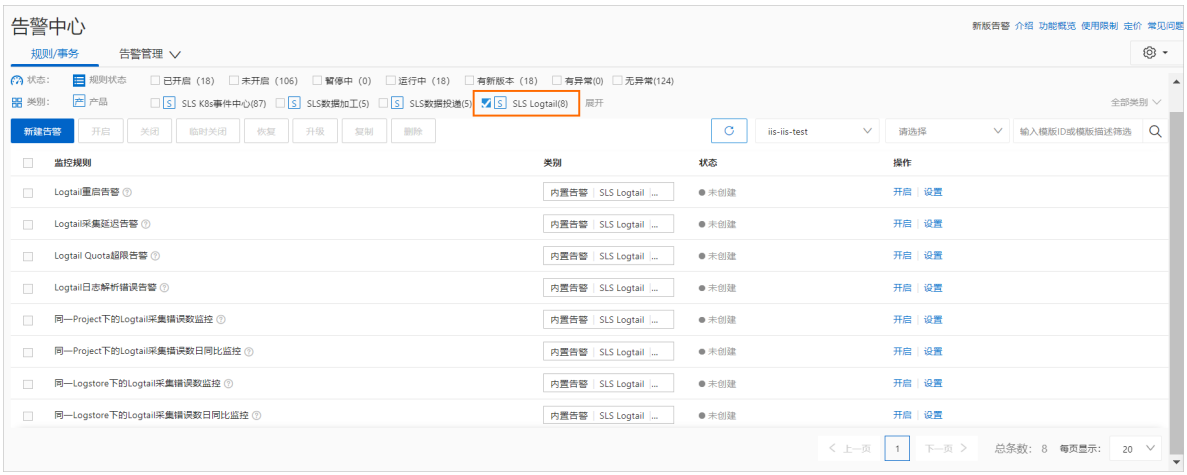
   您也可以创建新的行动策略用于告警通知。具体操作，请参见创建行动策略。
6. 在**编辑行动策略**对话框中，将**请求地址**修改为钉钉群机器人的Webhook地址。其他选项，保持默认配置。然后单击**确认**。

   如何获取钉钉群机器人的Webhook地址，请参见钉钉-自定义。您也可以根据业务需求，使用其他告警渠道。具体操作，请参见通知渠道说明。

步骤二：开启告警实例

日志服务已内置多种告警监控规则，您只需根据业务需求，开启对应的告警实例即可。

1. 在**告警中心**页面，单击**规则/事务**。
2. 在**规则/事务**页签中，单击**SLS Logtail**。



3. 在告警监控规则列表中，单击目标告警监控规则对应的**开启**。

   每个告警监控规则已预设参数，您可以直接单击**开启**。如果您要修改参数设置，可单击**设置**，进行修改。关于参数说明的更多信息，请参见Logtail告警监控规则。

Logtail告警监控规则

日志服务已内置如下告警监控规则，用于监控Logtail。

- Logtail重启告警
- Logtail采集延迟告警
- Logtail Quota超限告警
- Logtail日志解析错误告警
- 同一Project下的Logtail采集错误数监控
- 同一Project下的Logtail采集错误数日同比监控
- 同一Logstore下的Logtail采集错误数监控
- 同一Logstore下的Logtail采集错误数日同比监控

- Logtail重启告警

项目	说明
作用	监控Logtail的重启行为。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当同一个客户端出现Logtail重启次数超过设定的阈值时，触发告警。
参数配置	◦ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ◦ **重启次数阈值（严重）**：过去5分钟内，当同一个客户端出现Logtail重启次数大于该阈值时，则触发严重级别的告警。默认值为3。 ◦ **重启次数阈值（高）**：过去5分钟内，当同一个客户端出现Logtail重启次数大于该阈值时，则触发高级别的告警。默认值为1。 ◦ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ◦ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ◦ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- Logtail采集延迟告警

项目	说明
作用	监控Logtail的采集延迟。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。

项目	说明
触发条件	过去5分钟内，当有Logstore出现Logtail采集延迟时，触发告警。
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- Logtail Quota超限告警

项目	说明
作用	监控Logtail Quota超限情况。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当有Logstore出现因Quota超限导致Logtail发送数据失败时，触发告警。
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- Logtail日志解析错误告警

项目	说明
作用	监控Logtail日志解析错误的异常。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当有Logstore出现Logtail日志解析错误时，触发告警。
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- 同一Project下的Logtail采集错误数监控

项目	说明
作用	监控Logtail采集错误的数量。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当同一个Project出现Logtail采集错误数量超过设定的阈值时，触发告警。

项目	说明
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **采集错误数量阈值**：过去5分钟内，当同一个Project出现Logtail采集错误数量大于该阈值时，触发告警。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- 同一Project下的Logtail采集错误数日同比监控

项目	说明
作用	监控Logtail采集错误数量的日同比变化情况。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当同一个Project出现Logtail采集错误数量同比昨日增长率超过设定的阈值时，触发告警。
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **增长率阈值**：过去5分钟内，当同一个Project出现Logtail采集错误数量同比昨日增长率大于该阈值时，触发告警。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- 同一Logstore下的Logtail采集错误数监控

项目	说明
作用	监控Logtail采集错误的数量。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。
触发条件	过去5分钟内，当同一个Logstore出现Logtail采集错误数量超过设定的阈值时，触发告警。
参数配置	○ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ○ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ○ **采集错误数量阈值**：过去5分钟内，当同一个Logstore出现Logtail采集错误数量大于该阈值时，触发告警。 ○ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ○ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ○ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

- 同一Logstore下的Logtail采集错误数日同比监控

项目	说明
作用	监控Logtail采集错误数量的日同比变化情况。
检测频率&检测时间范围	每5分钟检测一次过去5分钟内的数据。

项目	说明
触发条件	过去5分钟内，当同一个Logstore出现Logtail采集错误数量同比昨日增长率超过设定的阈值时，触发告警。
参数配置	◦ **行动策略**：当前告警监控规则所绑定的行动策略，日志服务将通过该行动策略给指定用户发送告警通知。默认值为SLS Logtail内置行动策略（sls.app.logtail.builtin），您也可以修改或新建行动策略。具体操作，请参见创建行动策略。 ◦ **严重度**：告警的严重度，包括严重、高、中、低和报告。默认值为中。 ◦ **增长率阈值**：过去5分钟内，当同一个Logstore出现Logtail采集错误数量同比昨日增长率大于该阈值时，触发告警。 ◦ **静默期**：告警静默期，即设置重复通知的间隔。重复的告警在静默期内不会被重复通知。例如1d（1天）、2h（2小时）、3m（3分钟）。 ◦ **恢复通知**：监控对象恢复正常时，日志服务将以告警通知形式发送一条恢复通知。 ◦ **连续触发阈值**：连续多少次执行检查评估都满足触发条件时，才会触发告警。

3.10. Logtail限制说明

本文简介Logtail采集数据时在文件采集、Checkpoint管理、配置、资源、性能、错误处理等方面的限制。

文件采集限制

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

Checkpoint管理

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

配置限制

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

资源、性能限制

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

错误处理限制

项目	能力与限制
网络错误处理	在出现网络异常时会主动重试并自动调整重试间隔。
资源配额超限处理	如果数据发送速率超出Logstore最大配额，Logtail会阻塞采集并自动重试。
超时最大尝试时间	如果数据持续发送失败超过6小时，则丢弃该数据。
状态自检	支持异常情况下自动重启，例如程序异常退出及使用资源超限等。

其他限制

项目	能力与限制
日志采集延迟	正常情况下从日志写入磁盘到Logtail采集的日志延迟不超过1秒（阻塞状态下除外）。
日志上传策略	Logtail会将同一文件的日志自动聚合上传，聚合条件为日志超过2000条、日志总大小超过2 M或者日志采集时间超过3秒，任一条件满足则触发上传行为。

3.11. Logtail发布历史

本文档为您介绍日志服务Logtail的发布历史。

1.1.0

- 新功能

  netping插件支持httping和DNS解析耗时。

1.0.34

- 新功能

  新增Skywalking Logging API。

- 优化

  支持快速释放已停止的containerd容器文件句柄。

- 问题修复

  修复containerd容器的Kubernetes Label无法匹配问题。

1.0.32

- 新功能

  采集文本日志时，支持通过扩展配置（"enable_precise_timestamp": true）或processor_strptime插件解析高精度时间。

- 优化
  - 优化Kubernetes场景下rootfs探测机制。
  - 优化Kubernetes场景下容器运行时的识别机制。

- 问题修复

  修复netping插件在Windows系统中的异常问题。

1.0.31

- 新功能
  - Logtail采集配置支持环境变量替换。
  - 新增netping插件，用于采集指定的IP地址与目标IP地址之间的网络ping数据。
  - gotime插件支持将提取的日志时间转换为timestamp格式。
  - 采集syslog日志时，新增_client_ip_字段，表示传输日志的客户端IP地址。

- 优化

  优化容器标准输出流采集内存。

1.0.30

- 新功能
  - Prometheus插件支持通过多个Logtail采集配置采集同一台机器上的Prometheus数据。
  - 容器服务Kubernetes的Windows节点支持add-on token鉴权。

- 问题修复
  - 修复进程采集插件在Linux系统中发生threadNum与fdNum指标错误问题。
  - 修复SkyWalking插件出现ConfigurationDiscoveryService not implement错误问题。

1.0.29

- 问题修复

  修复采集容器标准输出时，通过正则匹配Label失效的问题（该问题发生在Logtail 1.0.27、Logtail 1.0.28版本中）。

1.0.28

- 新功能
  - 支持采集SNMP协议数据。
  - SkyWalking V3版本插件支持过滤instance属性。

- 问题修复

  修复SkyWalking V2版本插件的Span ID不正确问题。

1.0.27

- 新功能
  - 新增processor_regex插件。
  - 将Kafka发送渠道中的Partition策略修改为Hash策略。

- 优化

  优化主机指标的采集功能，支持采集IO Counter指标。

- 问题修复
  - 修复service_http_server插件不释放UNIX链接问题。
  - 修复Logtail同时运行多份metric_meta_kubernetes插件采集配置时冲突问题。

1.0.26

- 新功能
  - 支持采集进程指标。
  - 采集主机指标时，新增文件句柄以及TCP协议的采集。
  - 支持采集Kubernetes集群的Meta信息。
  - 支持采集主机的Meta信息。
  - 新增gRPC输出插件。
  - 采集容器日志时，支持Kubernetes集群语义识别。
  - 支持采集SkyWalking V2版本的Trace数据。
  - 支持在Windows i386平台运行input_canal插件。
- 优化

  优化容器环境下主机指标采集的准确性。

1.0.25

- 问题修复

  修复导入历史数据时潜在的崩溃问题。
- 优化

  加强在文件系统readdir API返回不精确元数据时的逻辑处理。

1.0.24

- 问题修复

  修复Logtail刚启动时发送的数据未携带自定义标识符的问题。

1.0.22

- 问题修复

  修复在全球加速模式下的网络中断时，Logtail可能停止上报状态数据（非用户数据）到日志服务的问题。

1.0.21

Logtail 1.0.21版本是首个全地域发布的Logtail 1.0版本，具备Logtail 0.16.64版本的所有功能，新增以下功能：

- 新功能
  - 新增配置项exactly_once_concurrency，实现了Logtail可以在本地磁盘记录细粒度的Checkpoint信息（文件级别）。更多信息，请参见Logtail配置。
  - 新增配置项enable_log_time_auto_adjust，实现了日志时间可自适应服务器本地时间。更多信息，请参见设置Logtail启动参数。
  - 新增配置项enable_log_position_meta，用于在日志中添加该日志所属原始文件的元数据信息。更多信息，请参见Logtail配置。
  - 新增配置项specified_year，用于使用当前时间中的年份或指定年份补全日志时间。更多信息，请参见Logtail配置。

0.16.68

- 问题修复
  - 修复采集容器标准输出时，未正确处理P（partial）标签导致的解析失败问题。
  - 修复在service_skywalking_agent_v3插件跨应用情况下，Links中的SpanID和ParentSpanID不正确的问题。

0.16.64

- 优化

  上调请求容器引擎时的超时时间，将3秒调整为30秒。新增环境变量DOCKER_CLIENT_REQUEST_TIMEOUT，用于设置请求容器引擎的超时时间。
- 问题修复
  - 修复service_skywalking插件的父Span ID发生错误的缺陷。
  - 修复根据环境变量创建的采集配置的逻辑在容器引擎异常时可能退出的缺陷。

0.16.62

> **说明** 如果您使用的是Logtail 0.16.58、0.16.60版本，建议您升级到Logtail 0.16.62版本。

- 问题修复

  修复在数据乱序场景下小概率发生的数据发送失败问题。

0.16.60

- 新功能

  支持采集containerd环境的容器数据。

0.16.56

- 优化

  调整服务日志中net_err_stat指标的覆盖范围，仅覆盖网络引起的发送错误。

0.16.54

- 新功能

  在服务日志中新增net_err_stat指标，记录最近1分钟、5分钟、15分钟内发生的发送错误的数量。

0.16.52

如果和容器（标准输出、容器文件）相关的采集配置较多，建议升级Logtail到0.16.52及以上版本，可有效地降低CPU开销。

- 优化

  优化容器数据采集场景的CPU开销。

0.16.50

- 新功能

  支持运行时按需安装service_telegraf插件（仅限ECS用户）。

0.16.48

- 优化

  优化service_telegraf插件，支持单机多个配置。

0.16.46

> **说明** 如果您在杭州、上海、北京地域，升级Logtail至0.16.46及以上版本，可避免Logtail在遇到网络抖动时切换Endpoint。

- 优化

  严格限制允许Logtail使用的网络类型。

0.16.44

- 新功能

  新增service_telegraf插件，支持采集指标数据。

0.16.42

- 新功能

  黑名单过滤支持多级匹配，例如*/path/**/log*。

- 优化

  优化本地IP地址获取策略。在原先策略失效时，获取列表中的第一个IP地址。

0.16.40

- 新功能
  - 新增主机状态数据插件metric_system_v2。
  - 新增环境变量ALIYUN_LOGTAIL_MAX_DOCKER_CONFIG_UPDATE_TIMES对应的参数max_docker_config_update_times，适用于在K8s环境中频繁创建Job短时任务的场景。
- 优化

  优化容器采集场景中采集配置较多时的性能（CPU 开销）。

- 问题修复

修复processor_split_log_string插件偶尔产生空行的问题。

0.16.38

- 新功能
  - 完整正则模式支持自定义时间字段名。
  - 在processor_json、processor_regex、processor_split_char插件中，新增KeepSourceIfParseError参数，支持解析失败时保留原始数据。更多信息，请参见使用Logtail插件处理数据。

0.16.36

- 新功能

  新增加密插件processor_encrypt。

0.16.34

- 新功能

  新增HTTP Probe，支持K8s健康检查。

- 问题修复
  - 修复某些环境中，由libcurl导致的core。
  - 修复在CentOS 8系统中安装Logtail，缺少libidn库的问题。

0.16.32

- 新功能

  在processor_json插件中，新增IgnoreFirstConnector参数。更多信息，请参见展开JSON字段。

0.16.30

> 注意 此版本长时间运行时有潜在的打开文件失败风险，建议升级至最新版本。

- 新功能

  在采集Docker标准输出及文件时，新增K8s级别的过滤功能。

- 优化

  优化网络条件较差时同地域Logstore之间的并发竞争。

- 问题修复

  修复由于文件打开逻辑错误小概率发生的checkpoint丢失问题。

0.16.28

- 新功能

  新增参数，用于配置首次采集的Tail大小。

- 优化

  优化容器元信息获取逻辑，降低异常容器对整体的影响。

- 问题修复
  - 修复docker_stdout在复杂环境下的内存泄漏问题。
  - 修复JSON模式下对毫秒时间戳不完整支持的问题。

0.16.26

- 新功能

  支持采集containerd的日志。

- 问题修复
  - 修复极低概率下发生的轮转文件丢失checkpoint的问题。
  - 修复本地采集配置文件*/etc/ilogtail/user_config.d*在*/usr/local/ilogtail/user_log_config.json*文件不存在时未被加载的问题。

0.16.24

- 新功能
  - 支持通过环境变量配置working_ip和working_hostname。
  - 新增force_quit_read_timeout参数，支持设置强制退出的超时时间（持续阻塞无法读取事件）。
  - 支持向插件传递path、topic等tag。

- 新增aggregator_shardhash插件，支持在插件内设置shardhash。
- 新增处理插件processor_gotime、processor_rename、processor_add_fields、processor_json、processor_packjson。更多信息，请参见使用Logtail插件处理数据。
- 更新LogtailInsight，新增进度查看功能（需要设置mark_offset_global_flag或customized_fields.mark_offset）。

- 优化
  - 优化Journal长时间运行内存偏高的情况，尽可能及早释放。
  - 优化在本地无配置的情况下首次获取配置的时间间隔。
- 问题修复
  - 修复多个Logtail配置的情况下可能产生的重复采集问题。
  - 修复毫秒、微秒时间戳不支持JSON int64的问题。

0.16.23

- 新功能
  - 支持毫秒、微秒时间戳（%s）。
  - 支持加载多个本地Logtail配置（/etc/ilogtail/config.d/）。
  - 支持加载多个本地用户配置（/etc/ilogtail/user_config.d/）。
  - 新增处理插件processor_split_key_value、processor_strptime。更多信息，请参见键值对模式、提取日志时间。
  - 新增oas_connect_timeout、oas_request_timeout参数，支持网络慢的场景。
- 优化

  取消混合配置（file+plugin）中对inputs的限制。

0.16.21

- 新功能
  - 支持自定义静态主题设置。
  - 支持黑名单过滤。
  - 在service_canal插件中新增EnableEventMeta参数，支持采集MySQL Binlog对应的header信息。
- 优化

  优化插件系统停止机制。
- 问题修复

  修复GBK日志潜在的内存泄漏。

0.16.18

- 新功能
  - 支持采集Docker事件。更多信息，请参见采集Docker事件。
  - 支持采集Systemd Journal日志。更多信息，请参见采集Systemd Journal日志。
  - 新增处理插件processor_pick_key 、processor_drop_last_key。
- 优化
  - 优化容器日志以及插件采集内存占用。
  - 优化采集容器标准输出（stdout）多行日志的性能。

0.16.16

- 新功能
  - 支持自动创建K8s审计日志相关的资源。
  - 支持通过环境变量配置启动参数，例如CPU、内存、发送并发等。
  - 支持通过环境变量配置自定义tag上传。
  - sidecar模式支持自动创建配置。更多信息，请参见通过Sidecar-CRD方式采集容器文本日志。
- 优化

  自动保存aliuid文件到本地文件。
- 问题修复
  - 修复采集容器文件出现极低概率的crash的问题。
  - 修复通过环境变量创建出的配置在K8s中存在的 `IncludeLabel` 不生效问题。

0.16.15

- 新功能
  - 采集MySQL Binlog时，支持GTID模式。在采集MySQL Binlog时自动开启该模式。
  - 历史数据导入文件名支持指定通配符。
  - K8s支持自动创建索引配置。
- 优化
  - 当分行失败时，支持检查`discardUnMatch`并上报分行失败的日志。
  - 遇到unknown send error时自动重试，防止极低情况下数据丢失（例如发送的数据包中途被篡改）。

0.16.14

- 新功能
  - 导入历史数据支持通配符模式。
  - 增加启动配置项`default_tail_limit_kb`，用于配置首次采集文件跳转大小（默认1024KB）。
  - 增加采集配置项`batch_send_seconds`，用于配置数据包发送的时间。
  - 增加采集配置项`batch_send_bytes`，用于配置数据包的大小。
- 优化

  采集容器标准输出（stdout）时，支持自动合并被Docker Engine拆分的日志。

0.16.13

新功能

- 支持通过环境变量配置日志采集。
- 支持采集MySQL Binlog中的meta数据，即新增日志字段`_filename_`和`_offset_`。
- 安装脚本支持VPC下自动选择参数。
- 支持全球加速安装模式。更多信息，请参见步骤二：配置Logtail采集加速。

0.16.11

- 优化

  采集MySQL Binlog时，支持采集filename和offset信息。
- 问题修复

  修复使用多行模式采集容器标准输出（stdout）时有一定概率出现异常的问题。

0.16.10

- 优化

  升级容器标准输出（stdout）采集方式。

0.16.9

- 问题修复
  - 修复极低概率下出现的socket fd泄漏问题。
  - 增加容器文件采集配置更新频率限制。

0.16.8

- 新功能
  - 新增Logtail Lumberjack插件，用于采集Logstash、Beats数据源。更多信息，请参见采集Beats和Logstash数据源。
  - 增加inotify黑名单功能。
- 问题修复
  - 修复旧安装包参数不统一的问题。
  - 修复在部分系统下安装Logtail时无法正确获取OS版本的问题。

0.16.6

- 新功能
  - 支持采集主机监控数据。
  - 支持采集Redis监控数据。
  - 支持采集MySQL Binlog中的DDL（data definition language）。

- 支持采集容器标准输出（stdout）和容器文件时，通过docker ENV（environment）过滤。

- 问题修复
  - 兼容MySQL table无主键情况下的数据采集。
  - 兼容容器采集模式下因容器引擎订阅通道不稳定造成事件丢失的问题。

0.16.5

- 新功能

  采集容器标准输出（stdout）时，新增多行采集模式。更多信息，请参见多行日志的Logtail采集配置示例。

0.16.4

- 新功能
  - 支持Docker&Kubernetes部署方案。
  - 支持采集容器标准输出（stdout）和容器文件。更多信息，请参见通过DaemonSet-控制台方式采集容器标准输出、通过DaemonSet-控制台方式采集容器文本日志。

0.16.2

- 新功能

  新增processor_geoip插件。更多信息，请参见转换IP地址。

0.16.0

- 新功能
  - 支持采集MySQL Binlog、MySQL查询结果、HTTP数据。更多信息，请参见使用Logtail插件采集数据。
  - 支持组合解析配置：正则模式、标定模式、分隔符模式、过滤器。

3.12. Logtail常见问题

本文介绍Logtail相关的常见问题。

分类		文档链接
基本概念说明		• Logtail基本问题 • 日志采集Agent对比
心跳问题		Logtail机器组无心跳排查思路
采集问题	通用场景	• 查询本地采集状态 • 如何查看Logtail采集错误信息 • Logtail采集日志失败的排查思路 • 日志服务采集数据常见的错误类型 • 如何实现文件中的日志被采集多份
	容器场景	如何排查容器日志采集异常
部署与管控问题	通用场景	• Logtail服务的app_info.json文件中IP地址为空 • ECS经典网络切换为VPC后，如何更新Logtail配置 • 如何采集企业内网服务器日志 • 如何使用Logtail自动诊断工具
	Windows场景	Windows实例安装Logtail服务日志提示异常信息
	容器场景	如何采集K8s集群的容器日志
日志格式解析问题		• 如何调试正则表达式 • 如何优化正则表达式的性能 • 如何通过完整正则模式采集多种格式日志

4.云产品日志采集
4.1. 云产品日志概述

日志服务支持采集弹性计算、存储服务、安全、数据库等多种阿里云云产品的日志数据，包括云产品的操作信息、运行状况、业务动态等信息。

> **说明** 日志服务支持通过日志审计服务自动跨账号采集日志的云产品覆盖基础（ActionTrail、容器服务Kubernetes版）、存储（OSS、NAS）、网络（SLB、ALB、API网关、VPC）、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。更多信息，请参见日志审计服务。

日志服务支持的具体产品及其日志采集和后续的操作信息如下列表所示：

- 弹性计算

云产品	日志	Project和Logstore	仪表盘
云服务器ECS	全量日志	自定义 通过Logtail采集。具体操作，请参见Logtail采集。	自定义
容器服务Kubernetes版	○ Kubernetes审计日志 ○ Kubernetes事件中心 ○ Ingress访问日志	自定义 通过Logtail采集。具体操作，请参见采集容器日志。	自定义
函数计算	执行日志	○ Project：aliyun-fc-*地域ID*-bb47c1e4-cdd4-5318-978e-d748952652c8 ○ Logstore：function-log	自定义

- 存储服务

云产品	日志	Project和Logstore	仪表盘
对象存储OSS	访问日志	○ Project：oss-log-*阿里云账号ID-地域ID* ○ Logstore：oss-log-store	○ 访问中心 ○ 审计中心 ○ 运维中心 ○ 性能中心
文件存储NAS	访问日志	○ Project：nas-*阿里云账号ID-地域ID* ○ Logstore：nas-nfs	○ nas-nfs-nas_summary_dashboard_cn ○ nas-nfs-nas_audit_dashboard_cn ○ nas-nfs-nas_detail_dashboard_cn

- 安全

云产品	日志	Project和Logstore	仪表盘
DDoS高防	全量日志	○ 中国内地的DDoS实例 ■ Project：ddos-pro-project-*阿里云账号ID*-cn-hangzhou ■ Logstore：ddos-pro-logstore ○ 非中国内地的DDoS实例 ■ Project：ddos-pro-*阿里云账号ID*-ap-southeast-1 ■ Logstore：ddos-pro-logstore	○ DDoS运营中心 ○ DDos访问中心

云产品	日志	Project和Logstore	仪表盘
DDoS高防（新BGP&国际）	全量日志	◦ DDoS高防（新BGP） ▪ Project：ddoscoo-project-*阿里云账号ID*-cn-hangzhou ▪ Logstore：ddoscoo-logstore ◦ DDoS高防（国际） ▪ Project：ddosdip-project-*阿里云账号ID*-ap-southeast-1 ▪ Logstore：ddoscoo-logstore	◦ DDoS运营中心 ◦ DDos访问中心
云安全中心	◦ 安全日志 ▪ 漏洞日志 ▪ 基线日志 ▪ 安全告警日志 ◦ 网络日志 ▪ DNS解析日志 ▪ 本地DNS日志 ▪ 网络会话日志 ▪ Web访问日志 ◦ 主机日志 ▪ 进程启动日志 ▪ 网络连接日志 ▪ 登录流水日志 ▪ 暴力破解日志 ▪ 进程快照日志 ▪ 账号快照日志 ▪ 端口快照日志	◦ Project：sas-log-*阿里云账号ID-地域ID* ◦ Logstore：sas-log	◦ 网络日志 ▪ DNS访问中心 ▪ 网络会话中心 ▪ Web访问中心 ◦ 主机日志 ▪ 登录中心 ▪ 进程中心 ▪ 网络连接中心 ◦ 安全日志 ▪ 基线中心 ▪ 漏洞中心 ▪ 告警中心
Web应用防火墙WAF	◦ 访问日志 ◦ 攻击防护日志	◦ 中国内地的WAF实例 ▪ Project：waf-project-*阿里云账号ID*-cn-hangzhou ▪ Logstore：waf-logstore ◦ 非中国内地的WAF实例 ▪ Project：waf-project-*阿里云账号ID*-ap-southeast-1 ▪ Logstore：waf-logstore	◦ 运营中心 ◦ 访问中心 ◦ 安全中心
云防火墙	互联网流量日志	◦ Project：cloudfirewall-project-*阿里云账号ID*-cn-hangzhou ◦ Logstore：cloudfirewall-logstore	报表

- 网络

云产品	日志	Project和Logstore	仪表盘
负载均衡SLB	7层访问日志	自定义	◦ slb-user-log-slb_layer7_operation_center_cn ◦ slb-user-log-slb_layer7_access_center_cn

云产品	日志	Project和Logstore	仪表盘
专有网络VPC	流日志	自定义	◦ Logstore Name-vpc_flow_log_traffic_cn ◦ Logstore Name-vpc_flow_log_rejection_cn ◦ Logstore Name-vpc_flow_log_overview_cn
弹性公网EIP	互联网流量日志	自定义	eip_monitoring
API网关	访问日志	自定义	Logstore Name_apigateway访问日志

- 数据库

云产品	日志	Project和Logstore	仪表盘
云原生分布式数据库PolarDB-X 1.0	SQL审计日志	◦ Project：drds-audit-*地域ID-阿里云账号ID* ◦ Logstore：drds-audit-log	◦ 运营中心（简化版） ◦ 性能中心 ◦ 安全中心
云数据库RDS	SQL 审计日志	自定义	◦ RDS审计运营中心 ◦ RDS审计性能中心 ◦ RDS审计安全中心
云数据库Redis	◦ 审计日志 ◦ 慢日志 ◦ 运行日志	◦ Project：nosql-*阿里云账号ID-地域ID* ◦ Logstore： ▪ redis_audit_log ▪ redis_slow_run_log	◦ Redis审计中心 ◦ Redis慢日志中心
云数据库MongoDB	◦ 审计日志 ◦ 慢日志 ◦ 运行日志	◦ Project：nosql-*阿里云账号ID-地域ID* ◦ Logstore： ▪ mongo_audit_log ▪ mongo_slow_run_log	Mongo审计日志中心

- 管理与监控

云产品	日志	Project和Logstore	仪表盘
操作审计	ActionTrail访问日志	◦ Project：自定义Project ◦ Logstore：actiontrail_跟踪名称	actiontrail_跟踪名称_audit_center_cn
	平台操作日志	自定义	innertrail_跟踪名称_audit_center_cn

- 物联网

云产品	日志	Project和Logstore	仪表盘
物联网平台	云端运行日志	◦ Project：iot-log-*阿里云账号ID-地域ID* ◦ Logstore：iot_logs	IoT运营中心

4.2. 云产品日志通用操作

您在各个云产品控制台开通日志功能后，即可执行日志服务相关操作。

常用操作

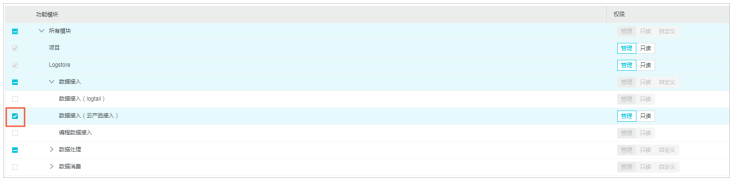
采集到云产品日志后，您可以执行如下操作。

操作	说明
查询和分析日志	查询和分析日志。具体操作，请参见查询和分析日志。
查询和分析时序数据	查询和分析时序数据。具体操作，请参见查询和分析时序数据。
查看原始日志	在查询和分析页面，查看原始日志。
快速分析	快速分析日志字段。具体操作，请参见快速分析。
统计图表	一切通过查询分析语句得到的结果都能以统计图表形式展示。具体操作，请参见统计图表。
设置告警	为查询和分析结果设置告警。具体操作，请参见告警。
创建仪表盘	新建一个仪表盘并将统计图表添加到仪表盘中。具体操作，请参见添加统计图表到仪表盘。
下载数据	日志服务支持将数据下载到本地。具体操作，请参见下载日志。
数据加工	您可以将采集到的数据进行规整、富化、分发、汇总等加工处理。具体操作，请参见数据加工。
数据消费	您可以将采集到的数据进行消费。具体操作，请参见数据消费。
数据投递	您可以将采集到的数据投递到OSS、Maxcompute、EMR等云产品中进行存储或计算分析。具体操作，请参见数据投递。

RAM用户授权

如果您要使用RAM用户操作云产品日志，您需要使用阿里云主账号为RAM用户授权。

您可以使用权限助手配置RAM用户权限，详情请参见配置权限助手。



4.3. 函数计算执行日志

4.3.1. 使用前须知

阿里云函数计算服务联合日志服务推出函数执行日志查询分析功能，将函数执行日志存储到日志服务的Logstore中，您可以进行代码调试、故障分析、数据分析等操作。本文介绍函数计算服务日志分析功能所涉及的资产详情、费用说明、使用限制等信息。

资产详情

- 专属Project和Logstore

  开通日志分析功能后，系统默认创建一个名为aliyun-fc-地域ID-bb47c1e4-cdd4-5318-978e-d748952652c8的Project，以及一个名为function-log的专属Logstore。

- 仪表盘

  无对应的专属仪表盘，支持自定义仪表盘。

费用说明

- 目前，函数计算服务不针对日志管理功能收取额外费用。
- 函数计算服务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，详情请参见日志服务产品定价。

使用限制

专属日志库不支持写入其他数据，在查询分析、告警、消费等功能上无特殊限制。

4.3.2. 开通日志功能

您在查询和分析函数计算执行日志前，需先开通日志功能。本文介绍如何在函数计算控制台上开通日志功能。

前提条件

已开通函数计算。更多信息，请参见开通服务。

操作步骤

> **说明** 如果您使用RAM用户进行日志功能开通，则需先为RAM用户授权。具体操作，请参见RAM用户授权。

1. 登录函数计算FC控制台。
2. 在左侧导航栏，单击**服务及函数**。
3. 在顶部菜单栏中，选择目标地域。
4. 在**服务列表**页面，单击**创建服务**。
5. 在**创建服务**面板中，配置相关参数，单击**确定**。

   重要参数说明如下表所示。更多信息，请参见创建服务。

> **注意** 首次启用日志功能时，单击**确定**后，系统将提示您未创建函数计算FC默认角色AliyunFcDefaultRole。请在提示框中，单击**立即授权**，然后根据页面提示，完成授权。

参数	说明
名称	自定义服务名称。
日志功能	是否启用阿里云日志服务。 ○ **启用**：启用后，日志服务将在对应的地域生成名为aliyun-fc-地域ID-****的Project和名为function-log的Logstore，并为该Logstore开启索引。 ○ **禁止**：不会将函数计算执行日志推送到日志服务。

后续步骤

日志服务采集到日志后，您可以在日志服务控制台执行查询分析、下载、投递、加工日志，创建告警等操作。更多信息，请参见云产品日志通用操作。

4.3.3. 日志字段详情

本文介绍函数计算执行日志的字段详情。

参数	说明
__topic__	日志主题，此处与服务名一致
functionName	函数名
message	记录日志信息
qualifiter	服务版本的别名
serviceName	服务名
versionId	服务版本号

4.4. OSS访问日志

4.4.1. 使用前须知

阿里云对象存储（OSS）联合日志服务推出OSS访问日志实时查询功能，帮助您完成OSS的操作审计、访问统计、异常事件回溯和问题定位等工作。本文介绍OSS日志实时查询功能相关的资产详情、费用说明以及使用限制。

资产详情

- 专属Project和Logstore

开通实时日志查询功能后，系统默认在对应的地域创建一个名为oss-log-阿里云账号ID-地域ID的Project，以及一个名为oss-log-store的专属Logstore。

- 专属仪表盘

  默认生成4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
访问中心	展示总体运营状况信息，包括PV、UV、流量以及外网访问地图分布等。
审计中心	展示文件操作的统计信息，包括读、写、删除文件等操作。
运维中心	展示OSS的运维信息，包括请求数量、操作失败的分布统计等。
性能中心	展示OSS性能的统计信息，包括外网下载/上传性能分布、不同网络与文件大小的传输性能、文件下载差异列表等。

费用说明

- 目前，OSS日志实时查询分析功能免费提供最近7天的OSS日志实时查询分析，限额900GB/天的日志写入额度（如果一条日志为1KB，约为9亿条），超出部分，由日志服务单独收费。如果您设置的日志存储时间大于7天，则超出7天的部分，由日志服务单独收费。日志服务计费说明请参见计费项。
- OSS日志实时查询功能免费提供每月16个Shard的额度，超出部分由日志服务单独收费。
- 日志服务对专属Logstore的读取流量、外网流量、数据加工、数据投递等按照标准收费，费用说明请参见按量付费。

使用限制

- 专属Logstore不支持写入其他数据、修改索引等操作，但在查询、统计、告警等功能上无特殊限制。
- 不支持删除专属Logstore。
- 阿里云账号欠费时，OSS日志实时查询功能暂停使用。

4.4.2. 开通日志实时查询功能

您在执行日志查询分析前，需先开通日志实时查询功能。本文介绍如何在OSS控制台上开通日志实时查询功能。

前提条件

- 已创建OSS Bucket，详情请参见创建存储空间。

  请确保日志服务Project和OSS Bucket在同一阿里云账号且在同一地域。

- 已授权日志服务使用AliyunLogImportOSSRole角色访问OSS。

  单击云资源访问授权，根据提示完成授权。

  > **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OSS日志无法正常推送到日志服务。

操作步骤

1. 登录OSS管理控制台。
2. 在**Bucket 列表**中，单击目标Bucket名称。
3. 单击**日志管理 > 实时查询**。
4. 单击**立即开通**。

   开通后，日志服务立即开始采集日志，并默认为您创建专属Project、Logstore以及配置索引。

后续步骤

日志服务采集到OSS日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作，详情请参见云产品日志通用操作。

4.4.3. 日志字段详情

本文介绍OSS日志类型及相关的日志字段详情。

OSS日志类型

日志类型	说明
访问日志	记录相关OSS Bucket的所有访问日志，实时采集。
批量删除日志	记录批量删除日志时具体的删除信息，实时采集。 **说明** 当您调用DeleteObjects接口时，访问日志中会有一条请求记录。
每小时计量日志	记录特定OSS Bucket每个小时累计的一些统计计量，延迟时间为几小时，用于辅助分析。

OSS自带日志和OSS访问记录日志

OSS访问日志实时查询分析功能由日志服务提供，包括OSS访问数据、批量删除数据和每小时计量数据的日志记录、采集、存储和分析等系列功能。OSS自带日志是OSS产品自带的访问数据的日志记录和存储功能，同样记录OSS存储空间的访问信息。

日志服务提供的日志包含OSS访问记录日志的全部信息，但是日志字段与OSS自带日志字段不同，如下表所示。更多信息，请参见访问日志。

OSS自带日志字段	日志服务-OSS日志字段
Remote IP	client_ip
Time	time
Request-URI	request-uri
HTTP Status	http_status
SentBytes	response_body_length
RequestTime (ms)	response_time
Referer	referer
User-Agent	user-agent
HostName	host
Request ID	request_id
LoggingFlag	logging_flag
Requester Aliyun ID	requester_id
Operation	operation
Bucket	bucket
Key	object
ObjectSize	object_size
Server Cost Time (ms)	server_cost_time
Error Code	error_code
Request Length	request_length
UserID	owner_id
Delta DataSize	delta_data_size
Sync Request	sync_request

访问日志

字段名称	含义
__topic__	日志主题，固定为oss_access_log。
acc_access_region	如果是传输加速请求，该字段为请求接入点所在地域名，否则为短划线（-）。
access_id	请求者的AccessKey ID。
bucket	OSS Bucket名称。
bucket_location	OSS Bucket所在的数据中心，一般格式为oss-<region ID>。
bucket_storage_type	OSS Object存储类型。 • standard：标准存储类型。 • archive：归档存储类型。 • infrequent_access：低频访问存储类型。
client_ip	发起请求的IP地址，即客户端IP地址、其网络防火墙或Proxy IP地址。
content_length_in	请求头中Content-Length的值，单位：字节。
content_length_out	响应头中Content-Length的值，单位：字节。
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（-）。
error_code	OSS返回的错误码。更多信息，请参见错误响应。
host	请求访问域名，例如：bucket123.oss-cn-beijing.aliyuncs.com。
http_method	HTTP请求方法。
http_status	HTTP请求返回的状态。
http_type	HTTP请求类型，包括HTTP和HTTPS。
logging_flag	是否开启定期导出日志到OSS Bucket的功能，true表示开启。
object	请求的OSS Object，格式为URL编码，查询时可以使用`select url_decode(object)`解码。
object_size	OSS Object的大小，单位：字节。
operation	访问类型。更多信息，请参见访问类型。
owner_id	OSS Bucket拥有者的阿里云账号ID。
referer	请求的HTTP Referer。
request_id	请求ID。
request_length	HTTP请求的大小，包括header，单位：字节。
request_uri	HTTP请求的URI，包括query-string，格式为URL编码，查询时可以使用`select url_decode(request_uri)`解码。
requester_id	请求者的ID，如果是匿名访问，则显示为短划线（-）。
response_body_length	HTTP响应中的Body大小，不包括header。
response_time	HTTP响应时间，单位：毫秒。
server_cost_time	OSS服务器处理本次请求所花的时间，单位：毫秒。
sign_type	签名类型。 • NotSign：未签名。 • NormalSign：一般方式签名。 • UriSign：通过URL签名。 • AdminSign：管理员账号。

字段名称	含义
sync_request	同步请求类型 。 • 短划线（-）：一般请求。 • cdn：CDN回源。 • sync-public：跨区域复制。 • lifecycle：设置生命周期规则。
time	OSS收到请求的时间，例如27/Feb/2018:13:58:45。如果需要时间戳可以使用__time__字段。
user-agent	HTTP的User-Agent头，例如curl/7.15.5。
vpc_addr	OSS所在VPC的VIP地址。 该地址为整数类型（例如343819108），您可以使用 `int_to_ip(cast(vpc_addr as bigint))` ，将其转换为IP地址形式。
vpc_id	OSS所在VPC的ID。
restore_priority	解冻优先级。
extend_information	扩展字段，默认为短划线（-）。 如果是通过RAM角色发起的请求，则日志会记录相关的RAM角色信息，拼接规则为 `requesterParentId,roleName,roleSessionName,roleOwnerId` ，以半角逗号（,）分隔，可能会继续拼接新字段。

批量删除日志

当您调用DeleteObjects接口时，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字段值为短划线（-）。查看具体的删除文件的列表，需要查看批量删除日志。批量删除日志的字段及说明如下，可以通过request_id字段关联。

字段名称	说明
__topic__	日志主题，固定为oss_batch_delete_log。
client_ip	发起请求的IP地址，例如客户端IP地址、其网络防火墙或Proxy的IP地址。
user_agent	HTTP的User-Agent头，例如curl/7.15.5。
bucket	OSS Bucket名称。
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：字节。
response_body_length	HTTP响应Body的大小，不包括header。
object	请求的OSS Object，格式为URL编码，查询时可以使用 `select url_decode(object)` 解码。
object_size	OSS Object的大小，对应请求对象的大小，单位：字节。
operation	访问类型。更多信息，请参见访问类型 。
bucket_location	OSS Bucket所在的数据中心，格式为oss-<region ID>。
http_method	HTTP请求方法，例如POST。
referer	请求的HTTP Referer。
request_id	请求ID。
http_status	HTTP请求返回的状态。
sync_request	同步请求类型。 • 短划线（-）：一般请求。 • cdn：CDN回源。 • sync-public：跨区域复制。

字段名称	说明
request_uri	请求的URI，包括query-string，格式为URL编码，查询时可以使用 `select url_decode(request_uri)` 解码。
host	请求访问域名，例如bucket123.oss-cn-beijing.aliyuncs.com。
logging_flag	是否开启定期导出日志到OSS Bucket的功能，true表示开启。
server_cost_time	OSS服务器处理本次请求的时间，单位：毫秒。
owner_id	OSS Bucket拥有者的阿里云账号ID。
requester_id	请求者的ID，如果匿名访问则为短划线（-）。
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OSS Bucket每个小时累计的计量信息，供辅助分析时参考使用。

字段名称	说明
__topic__	日志主题，固定为oss_metering_log。
owner_id	OSS Bucket拥有者的阿里云账号ID。
bucket	OSS Bucket名称。
cdn_in	CDN流入量，单位：字节。
cdn_out	CDN流出量，单位：字节。
get_request	GET请求次数。
intranet_in	内网流入量，单位：字节。
intranet_out	内网流出量，单位：字节。
network_in	外网流入量，单位：字节。
network_out	外网流出量，单位：字节。
put_request	PUT请求次数。
storage_type	OSS Bucket存储类型 。 • standard：标准存储类型。 • archive：归档存储类型。 • infrequent_access：低频访问存储类型。
storage	OSS Bucket存储量，单位：字节。
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：字节。
process_img	处理图像。
sync_in	同步流入量，单位：字节。
sync_out	同步流出量，单位：字节。
start_time	计量开始时间戳。
end_time	计量截止时间戳。
region	OSS Bucket所在地域。
bucket_location	OSS Bucket所在的数据中心，一般格式为oss-<region ID>。

访问类型

访问类型如下表所示。更多信息，请参见API概览。

操作	描述
AbortMultiPartUpload	断点上传-中止。
AppendObject	追加上传文件。
CompleteUploadPart	完成断点上传。
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Lifecycle配置。
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	查看跨地域复制。
GetBucketReplicationProgress	查看跨地域复制进度。
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	查看文件信息。
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。
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。

操作	描述
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。
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。

4.5. NAS访问日志

4.5.1. 使用前须知

阿里云文件存储（NAS）联合日志服务推出日志分析功能，提供NAS访问日志的实时采集、查询、分析、加工、消费等一站式服务。本文介绍NAS访问日志功能相关的资产详情、费用说明及使用限制等。

资产详情

- 专属Project和Logstore

  开通日志分析功能后，系统默认在不同的地域各创建一个名为nas-阿里云账号ID-地域ID的Project，以及一个名为nas-nfs的专属Logstore。

  > **说明** 请勿删除NAS日志相关的日志服务Project和Logstore，否则将无法正常采集日志到日志服务。

- 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，详情请参见创建仪表盘。

仪表盘	说明
nas-nfs-nas_summary_dashboard_cn	展示NAS总体运营情况，包括最近访问的Volume个数、写入总流量、读取总量、最近访问的客户端个数等信息。
nas-nfs-nas_audit_dashboard_cn	展示NAS文件系统操作统计信息，包括创建操作数、删除文件数、读取文件数等信息。
nas-nfs-nas_detail_dashboard_cn	展示NAS文件系统明细信息，包括最近访问的文件数量、操作趋势等信息。

费用说明

- 目前，NAS不针对日志分析功能收取额外费用。
- NAS将日志转储到日志服务后，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，详情请参见日志服务产品定价。

使用限制

- 专属Logstore不支持写入其他数据，但在查询、统计、告警等功能上无特殊限制。
- 目前，仅支持NFS协议的文件系统。

4.5.2. 开通日志分析功能

本文介绍如何在NAS控制台上开通日志分析功能，将日志采集到日志服务中。

前提条件

- 已创建NFS文件系统并完成挂载，详情请参见Linux系统挂载NFS文件系统。
- 已授权NAS使用AliyunNASLogArchiveRole角色访问日志服务。

  单击云资源访问授权，根据提示完成授权。

  > **说明**
  > - 该操作仅在首次配置时需要，需要由阿里云主账号进行授权。
  > - 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NAS日志无法正常推送到日志服务。

操作步骤

1. 登录NAS控制台。
2. 在左侧导航栏，单击**监控审计 > 日志分析**。
3. 在**日志分析**页面，单击**新建日志转储**。
4. 在**新建日志转储**页面，配置**文件系统类型**和**文件系统ID/名称**，并单击**确定**。

后续步骤

日志服务采集到NAS访问日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作，详情请参见云产品日志通用操作。

4.5.3. 日志字段详情

本文介绍NAS访问日志的字段详情。

字段名称	说明
__topic__	日志主题，固定为nas_audit_log
ArgIno	文件系统inode号
AuthRc	授权返回码
NFSProtocolRc	NFS协议返回码

字段名称	说明
OpList	NFSv4 Procedures编号
Proc	NFSv3 Procedures编号
RWSize	读写大小，单位为字节
RequestId	请求ID
ResIno	lookup的资源inode号
SourceIp	客户端IP地址
User	阿里云账号ID
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，单位为微秒

4.6. DDoS高防（旧版）日志

4.6.1. 使用前须知

阿里云日志服务和DDoS高防联合推出全量日志功能，提供网站访问日志、CC攻击日志的实时采集、查询、分析、加工、消费等一站式服务，帮您排查网站访问异常、追踪CC攻击者来源、分析网站运营情况等。本文介绍全量日志功能相关的资产详情、费用说明及使用限制等。

> **说明** 目前DDoS高防已升级为DDoS新BGP高防，DDoS高防（旧版）控制台仅用于已拥有DDoS高防实例的用户维护业务，无法创建新的DDoS高防实例。

功能优势

- 配置简单：轻松配置即可实时采集高防日志，添加新网站后自动为其开启日志采集。
- 实时分析：依托日志服务，提供实时日志分析，并提供开箱即用的报表中心，对CC攻击状况以及客户访问细节了如指掌。
- 实时告警：支持基于特定指标定制准实时的监测与告警，确保关键业务异常时可及时响应。
- 生态体系：支持对接其他生态如流计算、云存储、可视化方案，进一步挖掘数据价值。

资产详情

- 专属Project和Logstore
  - 如果您购买的是中国内地的DDoS实例，则开通全量日志功能后，系统默认创建一个名为ddos-pro-project-阿里云账号ID-cn-hangzhou的Project，以及一个名为ddos-pro-logstore的专属Logstore。
  - 如果您购买的是非中国内地地域的DDoS实例，则开通全量日志功能后，系统默认创建一个名为ddos-pro-阿里云账号ID-ap-southeast-1，以及一个名为ddos-pro-logstore的专属Logstore。

    > **说明** 专属Logstore不支持写入其他数据。在查询、统计、告警、消费等功能上无特殊限制。

- 专属仪表盘

  默认生成2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，详情请参见创建仪表盘。

仪表盘	说明
DDoS运营中心	展示被DDoS高防保护的网站的总体运营状况，包括有效请求率、有效流量、请求与拦截、攻击概况等数据。
DDos访问中心	展示被DDoS高防保护的网站总体访问状况，包括PV、UV、流入流量、网络in带宽峰值、网络out带宽峰值、访问趋势、来源分布等数据。

费用说明

- DDoS高防提供如下免费额度，超过部分按照标准方式收费。
  - 每天100 GB的数据写入流量
  - 每天4个Shard租用费用
  - 三天的免费日志存储时间
- DDoS将日志推送到日志服务后，在日志服务进行查询、分析、告警监控、可视化等操作免费，进行流量读取、数据加工、数据投递、告警短信语音通知等按照日志服务标准方式收费。更多信息，请参见日志服务产品定价。

4.6.2. 开通全量日志功能

本文介绍如何在DDoS高防控制台上开通全量日志分析功能，将DDoS高防全量日志采集到日志服务中。

前提条件

- 已有可用的DDoS高防实例，并添加域名。

  目前DDoS高防已升级为DDoS新BGP高防，DDoS高防（旧版）控制台仅用于已拥有DDoS高防实例的用户维护业务，无法创建新的DDoS高防实例。
- 已开通日志服务产品。

操作步骤

1. 登录DDoS高防管理控制台。
2. 在左侧导航栏中，选择**日志 > 全量日志**。
3. 在**全量日志**页面，根据页面提示，授权DDos高防服务使用AliyunDDoSCOOLogArchiveRole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。

4. 选择您需要开启DDoS高防日志采集功能的网站，并打开**状态**开关。

后续步骤

日志服务采集到DDoS高防全量日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作，详情请参见云产品日志通用操作。

4.6.3. 日志字段详情

本文档介绍DDos高防日志的字段。

字段	说明
__topic__	日志主题（Topic），固定为ddos_access_log。
body_bytes_sent	请求发送Body的大小，单位：字节
cache_status	Cache状态
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。 • 1表示阻断。 • 其他内容表示通过。
content_type	内容类型
host	源网站

字段	说明
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IP地址。
https	该请求是否为HTTPS请求。 • true表示该请求是HTTPS请求。 • false表示该请求是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（-）。
querystring	请求字符串
real_client_ip	访问客户的真实IP地址。如果获取不到，则显示为短划线（-）。
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，例如200
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_browser_version	浏览器版本
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，多个地址用逗号（,）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
user_id	阿里云账号ID

4.7. DDoS高防（新BGP&国际）日志

4.7.1. 使用前须知

阿里云DDoS高防（新BGP）、DDoS高防（国际）联合日志服务推出全量日志分析功能，提供网站访问日志、CC攻击日志的实时采集、查询、分析、加工、消费等一站式服务，帮助您排查网站访问异常、追踪CC攻击者来源、分析网站运营情况等。本文介绍全量日志分析功能相关的资产详情、费用说明及使用限制等。

根据要接入DDoS高防进行防护的业务服务器的部署地域的不同，DDoS高防提供新BGP（Anti-DDoS Pro）和国际（Anti-DDoS Premium）两种解决方案。您可以在DDoS高防控制台在顶部导航栏中选择DDoS高防（新BGP）、DDoS高防（国际）服务。DDoS高防（新BGP）适用于业务服务器部署在中国内地地域的场景，DDoS高防（国际）服务适用于业务服务部署在中国内地以外地域的场景。

资产详情

- 专属Project和Logstore
  - DDoS高防（新BGP）

    开通全量日志分析功能后，系统默认创建一个名为ddoscoo-project-阿里云账号ID-cn-hangzhou的Project，以及一个名为ddoscoo-logstore的专属Logstore。
  - DDoS高防（国际）

    开通全量日志分析功能后，系统默认创建一个名为ddosdip-project-阿里云账号ID-ap-southeast-1的Project，以及一个名为ddosdip-logstore的专属Logstore。
- 专属仪表盘

  默认生成2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
DDoS运营中心	展示被DDoS高防（新BGP）、DDoS高防（国际）保护的网站的总体运营状况，包括有效请求率、有效流量、请求与拦截、攻击概况等数据。
DDos访问中心	展示被DDoS高防（新BGP）、DDoS高防（国际）保护的网站总体访问状况，包括PV、UV、流入流量、网络in带宽峰值、网络out带宽峰值、访问趋势、来源分布等数据。

费用说明

由DDoS高防（新BGP）、DDoS高防（国际）售卖全量日志分析功能，收取相关费用。如果您要执行数据加工、投递、从外网接入点流式读取数据操作，由日志服务收取加工计算费用、数据投递费用和外网读取流量费用。更多信息，请参见计费项。

使用限制

- 专属Logstore不支持写入其他数据，但在查询、统计、告警、消费等功能上无特殊限制。
- 阿里云日志服务产品须处于可用状态（不欠费），否则全量日志分析功能暂停使用。
- 不支持删除专属Logstore，不支持在日志服务控制台上修改专属Logstore的存储时间，您可以在DDoS控制台上修改存储时长（30~180天）。
- 请确保日志存储空间充足，当日志存储空间被占满后，无法写入新的日志。

  > **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。
- 请确保全量日志分析功能在有效期内，当全量日志分析功能到期七天后仍未续费延长有效期，DDoS高防服务端将自动删除所有日志。

功能优势

- 配置简单：轻松配置即可实时采集高防日志，添加新网站后自动为其开启日志采集。
- 实时分析：依托日志服务，提供实时日志分析，并提供开箱即用的报表中心，对CC攻击状况以及客户访问细节了如指掌。
- 实时告警：支持基于特定指标定制准实时的监测与告警，确保关键业务异常时可及时响应。
- 生态体系：支持对接其他生态如流计算、云存储、可视化方案，进一步挖掘数据价值。

4.7.2. 开通全量日志分析功能

本文介绍如何在DDoS高防（新BGP、国际）控制台上开通全量日志分析功能，将DDoS高防（新BGP、国际）全量日志采集到日志服务中。

前提条件

- 已创建新BGP或国际高防实例。具体操作，请参见购买DDoS高防实例。
- 已为新BGP或国际高防实例接入域名。具体操作，请参见添加网站。

操作步骤

> **说明** 如果您使用RAM用户开通全量日志功能，则需先为RAM用户授权。详情请参见RAM用户授权。

1. 登录DDoS高防控制台。
2. 在页面左上角，选择服务地域为**中国内地**。

   此处以DDoS高防（新BGP）服务为例。如果您要使用DDoS高防（国际）服务，请选择服务地域为**非中国内地**。
3. 在左侧导航栏，选择**调查分析 > 全量日志分析**。
4. 购买日志存储容量和时长。

   如果您已购买日志存储容量和时长，请跳过此步骤。

   i. 单击**立即购买**。

   ii. 在全量日志购买页面，配置如下参数。

参数	说明
适用产品	选择**新BGP高防**。 此处以DDoS高防（新BGP）服务为例。如果您要使用DDoS高防（国际）服务，请选择**DDoS高防（国际）**。
日志存储量	日志的最大存储空间，单位：TB。当您选购的日志存储空间占满后，将无法存储新的日志。 一般情况下，每条请求日志大约为2 KB，如果您的业务的平均请求量为500 QPS，则一天的日志存储所需要的存储空间为：500×60×60×24×2=86,400,000 KB（即82 GB）。系统默认日志存储周期为180天，如果您需要存储最近180天的日志，则需要选择的日志存储量为14,832 GB（约14.5 TB）。
购买时长	全量日志分析功能的有效期，到期后，将停止存储新的日志。 **警告** 如果全量日志分析功能到期七天后仍未续费延长有效期，DDoS高防服务端将自动删除所有日志。

   iii. 单击**立即购买**，完成支付。
5. 在**全量日志分析**页面，根据页面提示，授权新BGP高防服务使用AliyunDDoSCOOLogArchiveRole角色访问日志服务。

   > **说明**
   > - 该操作仅在首次配置时需要，且需要由阿里云账号进行授权。
   > - 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。
6. 在**全量日志分析**页面，选择网站域名，开启其状态开关，为网站域名启用全量日志分析功能。

   > **说明** 建议您在全量日志分析功能使用期间，定期关注全量日志存储空间的使用情况和服务有效期。当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储影响日志存储的连续性。

后续步骤

日志服务采集到DDoS高防（新BGP、国际）全量日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作，详情请参见云产品日志通用操作。

4.7.3. 管理日志存储空间

本文介绍如何在DDoS高防（新BGP）控制台或DDoS高防（国际）控制台上管理和操作日志存储空间。

操作步骤

1. 登录DDoS高防控制台。
2. 在页面左上角，选择服务地域为**中国内地**。

   此处以DDoS高防（新BGP）为例。如果您要使用DDoS高防（国际），请选择服务地域为**非中国内地**。
3. 在左侧导航栏，单击**调查分析 > 全量日志分析**。
4. 在**全量日志分析**页面，管理日志存储空间。

   具体操作如下所列。
   - 查看日志存储空间。

> **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储影响日志存储的连续性。

- 延长使用时长。

  单击**续费**，选择续费时长。

- 升级存储空间。

  单击**升级**，升级存储空间。

- 清空日志。

  单击**清空**，清空日志。

> **警告** 日志清空后将无法复原，请务必谨慎使用清空功能。

4.7.4. 日志字段详情

本文档介绍DDoS高防（新BGP）、DDoS高防（国际）访问日志的字段。

字段	说明
__topic__	日志主题。 • DDoS高防（新BGP）：固定为ddoscoo_access_log • DDoS高防（国际）：固定为ddosdip_access_log
body_bytes_sent	请求发送Body的大小，单位为字节。
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IP地址。
https	是否为HTTPS请求，其中： • true：该请求是HTTPS请求。 • false：该请求是HTTP请求。
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	访问客户的真实IP地址。如果获取不到，则显示为短划线（-）。
isp_line	线路信息，例如BGP、电信、联通等。
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，单位为字节。
request_method	请求的HTTP方法。
request_time_msec	请求时间，单位为毫秒。

字段	说明
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。
cc_action	CC防护策略行为，例如none、challenge、pass、close、captcha、wait、login等。
cc_blocks	是否被CC防护策略阻断，其中： • 1：表示阻断。 • 其他内容表示通过。 日志中可能不存在该字段，而是以last_result字段记录请求是否被CC防护策略阻断。
last_result	表示是否被CC防护策略阻断，其中： • ok：表示通过。 • failed：表示不通过，包括校验未通过和阻断。 日志中可能不存在该字段，而是以cc_blocks字段记录请求是否被CC防护策略阻断。
cc_phase	CC防护策略，例如seccookie、server_ip_blacklist、static_whitelist、 server_header_blacklist、server_cookie_blacklist、server_args_blacklist、qps_overmax等。
ua_browser	浏览器。 日志中可能不存在该字段。
ua_browser_family	浏览器系列。 日志中可能不存在该字段。
ua_browser_type	浏览器类型。 日志中可能不存在该字段。
ua_browser_version	浏览器版本。 日志中可能不存在该字段。
ua_device_type	客户端设备类型。 日志中可能不存在该字段。
ua_os	客户端操作系统。 日志中可能不存在该字段。
ua_os_family	客户端操作系统系列。 日志中可能不存在该字段。
upstream_addr	回源地址列表，格式为IP:Port，多个地址用英文逗号（,）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。
user_id	阿里云账号ID。
querystring	请求字符串。

4.8. DDoS原生防护日志

4.8.1. 使用前须知

日志服务联合阿里云DDoS原生防护推出防护日志分析功能。开启防护分析后，您可以通过防护日志查询和分析DDoS原生防护实例上的清洗、黑洞和代播牵引事件信息。帮您排查网站访问异常、分析网站运营情况等。本文介绍DDoS原生防护日志分析功能相关的资产详情、费用说明及使用限制等。

资产详情

- 专属Project和Logstore

  开通原生防护日志分析功能后，系统默认创建一个名为ddosbgp-project-阿里云账号ID-cn-hangzhou的Project，以及一个名为ddosbgp-logstore的专属Logstore。

- 专属仪表盘

  默认生成2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，详情请参见创建仪表盘。

仪表盘	说明
DDoS原生防护事件报表	展示DDoS原生防护对保护网站的黑洞和代播牵引事件的统计报表。
DDoS原生清洗分析报表	展示DDoS原生防护对被保护网站的攻击流量进行清洗的报表，包括入流量监控、入流量分布、入流量协议类型分布等数据。

费用说明

- 由DDoS原生防护售卖日志分析功能，根据日志存储时长和日志存储容量收取费用。DDoS原生防护分析功能公测中，提供防护流量全量日志分析和报表功能，公测期间免费。
- DDoS原生防护将日志推送到日志服务后，在日志服务进行查询、分析、告警监控、可视化等操作免费，进行流量读取、数据加工、数据投递、告警短信语音通知等按照日志服务标准方式收费。更多信息，请参见日志服务产品定价。

使用限制

- 专属Logstore不支持写入其他数据，但在查询、统计、告警、消费等功能上无特殊限制。
- 不支持删除专属Logstore。
- 不支持在日志服务控制台上修改专属Logstore的存储时间，您可以在DDoS防护产品控制台上修改存储时长（30~180天）。
- 请确保日志存储空间充足，当日志存储空间被占满后，无法写入新的日志。

  > **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。

4.8.2. 开通原生防护日志

本文介绍在DDoS原生防护控制台上开通原生防护日志分析功能的操作方法。借助日志服务的查询分析功能，可以可视化分析DDoS原生防护日志，有助于您防御DDoS攻击。

前提条件

已创建DDoS原生防护实例。更多信息，请参见购买DDoS原生防护企业版实例。

操作步骤

1. 登录流量安全产品控制台。
2. 在左侧导航栏，选择**网络安全 > DDoS原生防护 > 防护分析**。
3. 在顶部菜单栏左上角处，选择实例所在资源组和地域。
4. 如果是首次使用防护分析，参照界面提示向导完成RAM授权。
5. 升级DDoS原生防护实例。
   i. 在**防护分析**页面，从**选择实例**列表中，选择目标DDoS原生防护实例。
   ii. 单击**立即升级**。
   iii. 在**变配**页面，单击**开启防护分析（公测）**。
   iv. 选中**服务协议**，并单击**立即购买**。

v. 完成支付。

> **说明** 在公测期间支持免费开通DDoS原生防护分析功能。

6. 返回**防护分析**页面，单击**立即开启**。

当前实例的防护分析状态更新为开启后，原生防护将自动采集当前实例的防护日志（存储在阿里云日志服务中），并向您提供查询分析和报表功能。您可以通过设置状态开关，开启、关闭当前实例的防护分析功能。

> **说明** 建议您在原生防护日志分析功能使用期间，定期关注原生防护日志存储空间的使用情况和服务有效期。当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储影响日志存储的连续性。

后续步骤

日志服务采集到DDoS原生防护日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作。更多信息，请参见云产品日志通用操作。

4.8.3. 日志字段详情

本文档介绍DDoS原生防护访问日志的字段。

字段	说明
__topic__	日志主题，固定为ddosbgp_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
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

4.9. 云安全中心日志

4.9.1. 使用前须知

阿里云日志服务与云安全中心产品全方位对接，提供风险威胁数据的实时采集、查询与分析、加工、消费等一站式服务，帮助您全面了解、有效处理服务器的安全隐患，实现对云上资产的集中安全管理。 本文介绍云安全中心日志分析功能相关的资产详情及使用限制。

资产详情

- 专属Project和Logstore

  开通日志分析功能后，系统默认创建一个名为sas-log-阿里云账户ID-区域名的Project，以及一个名为sas-log的专属Logstore。

  > **说明** 如果您误删了专属Logstore，系统提示sas-log Logstore不存在，并且您当前Logstore的所有日志数据丢失。这种情况下，您需提交工单重置处理。重置后您需重新开通日志分析服务，已丢失的日志数据无法恢复。

- 专属仪表盘

  日志分析功能覆盖3大类14种日志，默认生成9个仪表盘。

日志类型	仪表盘	说明
网络日志	DNS访问中心	展示服务器上DNS查询的全局视图，包括外网查询成功率、本地以及外网DNS查询的分布、趋势等。
	网络会话中心	展示资产相关网络会话的全局视图，包括连接趋势与分布、链接目标以及接入的趋势与分布等。
	Web访问中心	展示主机对外HTTP以及基于主机的Web服务被访问的全局视图，包括请求成功率、访问趋势与有效率、被访问域名的分布以及其他相关分布等。
主机日志	登录中心	展示主机登录信息的全局视图，包括登录源和目标地址地理分布、趋势、登录端口和类型分布等。
	进程中心	展示主机进程启动相关的全局视图，包括进程启动的趋势与分布、进程类型以及特定bash或Java程序的启动分布等。
	网络连接中心	展示主机网络连接变化的全局视图，包括连接的趋势与分布、连接目标以及接入的趋势与分布等。
安全日志	基线中心	展示基线检查相关的全局视图，包括检查问题分布、新增/处理的趋势、状态等。
	漏洞中心	展示漏洞相关的全局视图，包括漏洞分布，新增、验证、修复状态等。
	告警中心	展示安全告警相关的全局视图，包括新增、处理的趋势与状态等信息。

费用说明

- 日志服务对专属日志库的读写流量、索引流量、存储、shard数、读写次数不收取任何费用，但对外网流量、数据加工的计算、投递的计算等会按照标准收费。更多信息，请参见日志服务产品定价。
- 日志分析功能需要额外付费，由云安全中心收取，费用说明请参见计费模式。

使用限制

- 专属日志库不支持写入其他数据。
- 根据《网络安全法》日志至少存储六个月的要求，推荐每台服务器配置30GB的日志存储容量。

4.9.2. 开通日志分析功能

您在执行日志分析前，需先开通日志分析功能。本文介绍如何在云安全中心控制台上开通日志分析功能。

前提条件

已开通日志服务、云安全中心服务。

操作步骤

1. 登录云安全中心控制台。
2. 在左侧导航栏，单击**调查响应 > 日志分析**。
3. 在**开通日志服务**配置向导中，单击**立即开通**。

4. 在购买页面，选择版本并设置日志存储容量，其他参数说明请参见购买云安全中心。

   设置**日志存储容量**后，即表示开通日志分析功能。

   > 注意
   > - 云安全中心基础版不支持开通日志分析功能。
   > - 云安全中心企业版支持查看网络日志、安全日志和主机日志，高级版、基础杀毒版仅支持查看安全日志和主机日志。

5. 根据页面提示，完成支付。

   开通后，日志服务立即开始采集日志，并默认为您创建专属Project、Logstore以及配置索引。

后续步骤

日志服务采集到日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作，详情请参见云产品日志通用操作。

4.9.3. 日志字段详情

本文介绍云安全中心支持的14种日志的字段详情。

网络日志

- DNS日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-log-dns
additional	additional字段，多个值之间用竖线（|）分隔
additional_num	additional数量
answer	DNS回答信息，多个值之间用竖线（|）分隔
answer_num	DNS回答信息数量
authority	authority字段，多个值之间用竖线（|）分隔
authority_num	authority数量
client_subnet	客户端子网
dst_ip	目标IP地址
dst_port	目标端口
in_out	数据的传输方向 ○ in表示流入 ○ out表示流出
qid	查询ID
qname	查询域名
qtype	查询类型
query_datetime	查询时间戳，单位：毫秒
rcode	返回代码
region	地域ID ○ 1：北京 ○ 2：青岛 ○ 3：杭州 ○ 4：上海 ○ 5：深圳 ○ 6：其它

字段名称	说明
response_datetime	返回时间，例如：2018-09-25 09:59:16
src_ip	源IP地址
src_port	源端口

- 本地DNS日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为local-dns
answer_rdata	DNS回答信息，多个值之间用竖线（|）分隔
answer_ttl	DNS回答的时间周期，多个值之间用竖线（|）分隔
answer_type	DNS回答的类型，多个值之间用竖线（|）分隔
anwser_name	DNS回答的名称，多个值之间用竖线（|）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	查询ID
instance_id	实例ID
internet_ip	互联网IP地址
ip_ttl	IP周期
query_name	查询域名
query_type	查询类型
src_ip	源IP地址
src_port	源端口
time	查询的时间戳，单位：秒
time_usecond	响应耗时，单位：微秒
tunnel_id	通道ID

- 网络会话日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-log-session
asset_type	关联的资产类型，例如：ECS
dst_ip	目标IP地址
dst_port	目标端口
proto	协议类型，例如：tcp、udp
session_time	会话时间，例如：2018-09-25 09:59:49

字段名称	说明
src_ip	源IP地址
src_port	源端口

- Web访问日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-log-http
content_length	内容长度
dst_ip	目标IP地址
dst_port	目标端口
host	访问的主机名
jump_location	重定向地址
method	HTTP请求方法，例如：GET
referer	客户端向服务器发送请求时的HTTP referer，告知服务器访问来源的HTTP链接
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
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

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-vul-log
name	漏洞名称
alias_name	漏洞别名
op	操作信息 ○ new：新增 ○ verify：验证 ○ fix：修复
status	状态信息，详情请参见安全日志状态码。
tag	漏洞标签，例如：oval、system、cms

字段名称	说明
type	漏洞类型，例如： ○ sys：windows漏洞 ○ cve：Linux漏洞 ○ cms：Web CMS漏洞 ○ EMG： 紧急漏洞
uuid	客户端号

- 基线日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-hc-log
level	日志级别，例如：ow、mediam、high
op	操作信息，例如： ○ new：新增 ○ verify：验证 ○ fix：修复
risk_name	风险名称
status	状态信息，详情请参见安全日志状态码。
sub_type_alias	子类型别名，中文格式
sub_type_name	子类型名称
type_name	类型名称
type_alias	类型别名，中文格式
uuid	客户端号

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
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

type_name	sub_type_name
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
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

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为sas-security-log
data_source	数据源 ○ 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
level	告警级别，例如：suspicious、serious、remind
name	名称
op	操作信息，例如： ○ new：新增 ○ dealing：处理
status	状态信息，详情请参见安全日志状态码。
uuid	客户端号

主机日志

- 进程启动日志

字段名称	说明
__time__	日志时间
__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	启动进程的完整命令行
username	用户名
uid	用户ID
pid	进程ID
filename	进程文件名
filepath	进程文件所在的完整路径
groupname	用户组
ppid	父进程ID
pfilename	父进程文件名
pfilepath	父进程文件所在的完整路径
containerhostname	容器主机名
containerpid	容器PID
containerimageid	镜像ID
containerimagename	镜像名称
containername	容器名称
containerid	容器ID

字段名称	说明
cwd	进程运行目录

- 进程快照日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为aegis-snapshot-process
uuid	客户端号
ip	客户端主机的IP地址
cmdline	启动进程的完整命令行
pid	进程ID
name	进程文件名
path	进程文件所在的完整路径
md5	进程文件MD5，超过1MB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
user	用户名
uid	用户ID

- 登录日志

> **说明** 1分钟内的重复登录会被合并为1条日志，warn_count字段表示次数。

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为aegis-log-login
uuid	客户端号
ip	客户端主机的IP地址
warn_ip	登录来源IP地址
warn_port	登录端口
warn_type	登录类型，例如： ○ SSHLOGIN：SSH登录 ○ RDPLOGIN：远程桌面登录 ○ IPCLOGIN：IPC登录
warn_user	登录用户名
warn_count	登录次数，例如：值为3，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
__time__	日志时间
__topic__	日志主题，固定为aegis-log-crack

字段名	说明
uuid	客户端号
ip	客户端机器IP地址
warn_ip	登录来源IP地址
warn_port	登录端口
warn_type	登录类型，例如： ○ SSHLOGIN：SSH登录 ○ RDPLOGIN：远程桌面登录 ○ IPCLOGIN：IPC登录
warn_user	登录用户名
warn_count	失败登录次数

- 网络连接日志

> **说明** 主机上每隔10秒到1分钟采集一次变化的网络连接日志，而一个网络连接的状态从建立到结束的过程中部分状态会被采集到。

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为aegis-log-network
uuid	客户端号
ip	客户端机器IP地址
src_ip	源IP地址
src_port	源端口
dst_ip	目标IP地址
dst_port	目标端口
proc_name	进程名
proc_path	进程路径
proto	协议，例如：udp、raw
status	连接状态，详情请参见网络连接状态。

网络连接状态

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

状态值	描述
9	fin_wait2
10	time_wait
11	delete_tcb

- 端口快照日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为aegis-snapshot-port
uuid	客户端号
ip	客户端机器IP地址
proto	协议，例如：tcp、udp、raw
src_ip	监听的IP地址
src_port	监听的端口
pid	进程ID
proc_name	进程名

- 账户快照日志

字段名称	说明
__time__	日志时间
__topic__	日志主题，固定为aegis-snapshot-host
uuid	客户端号
ip	客户端机器IP地址
user	用户
perm	是否拥有root权限 ◦ 0：没有 ◦ 1：有
home_dir	home目录
groups	用户属于的组
last_chg	密码最后修改日期
shell	Shell命令
domain	Windows域
tty	登录的终端
warn_time	密码到期提醒日期
account_expire	账号超期日期
passwd_expire	密码超期日期
login_ip	最后一次登录的远程IP地址
last_logon	最后一次登录的日期和时间

字段名称	说明
status	用户状态 ○ 0：禁用 ○ 1：正常

4.10. WAF日志

4.10.1. 使用前须知

阿里云Web应用防火墙（WAF）联合日志服务推出WAF日志服务功能，提供网站域名访问日志、攻击防护日志的实时采集、查询、分析、加工、消费等一站式服务，满足等保合规要求和您的网站业务防护和运营需求。本文介绍WAF日志服务功能相关的资产详情、费用说明及使用限制等。

资产详情

- 专属Project和Logstore
  - 如果您购买的是WAF中国内地实例，则开通WAF日志服务功能后，系统默认创建一个名为waf-project-阿里云账号ID-cn-hangzhou，以及一个名为waf-logstore的专属Logstore。
  - 如果您购买的是WAF海外地区实例，则开通WAF日志服务功能后，系统默认创建一个名为waf-project-阿里云账号ID-ap-southeast-1，以及一个名为waf-logstore的专属Logstore。
- 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，详情请参见创建仪表盘。

仪表盘	说明
运营中心	展示网站业务的有效率、攻击情况等运营指标，网络In带宽峰值、Out带宽峰值、请求数等流量指标，运营趋势及攻击概况等信息。
访问中心	展示网站业务的PV、UV等基本访问指标，访问趋势、访问来源分布等信息。
安全中心	展示网站业务遭受攻击的基本指标、攻击类型、攻击趋势、来源分布等信息。

费用说明

由WAF售卖日志服务功能，根据日志存储时长和日志存储容量收取费用。如果您要执行数据加工、投递、从外网接入点流式读取数据等操作，由日志服务收取加工计算费用、数据投递费用和外网读取流量费用，详情请参见日志服务产品定价。

使用限制

- 仅通过包年包月方式开通的高级版、企业版或旗舰版WAF，支持WAF日志服务功能。
- 阿里云日志服务产品需处于可用状态（不欠费），否则WAF日志服务功能暂停使用。
- 专属Logstore不支持写入其他数据，但在查询、统计、告警、消费等功能上无特殊限制。
- 不支持删除专属Logstore及修改专属Logstore的存储时间。
- 请确保WAF日志存储空间充足，当日志存储空间被占满后，无法写入新的WAF日志。

> **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。

功能优势

- 等保合规：存储六个月以上的网站访问日志，助力网站符合等保合规要求。
- 配置简单：轻松配置即可实时采集网站域名访问日志和攻击防护日志。同时，支持自定义日志存储的时长和容量，自由选择日志采集的网站。
- 实时分析：依托日志服务，提供实时日志分析，并提供开箱即用的报表中心，让您对网站业务的各种Web攻击状况以及客户访问细节了如指掌。
- 实时告警：支持基于特定指标定制近实时的监测与告警，确保关键业务异常时可及时响应。
- 生态体系：支持对接其他生态如流计算、云存储、可视化方案，进一步挖掘数据价值。

应用场景

- 追踪Web攻击日志，溯源安全威胁。
- 实时查看Web请求活动，洞察状态与趋势。
- 快速了解安全运营效率，及时反馈处理。
- 输出安全网络日志到自建的数据与计算中心。

4.10.2. 开通WAF日志服务

本文介绍如何在Web应用防火墙控制台上开通WAF日志服务功能，将日志采集到日志服务中。

前提条件

- 已通过包年包月方式开通高级版、企业版或旗舰版WAF，详情请参见开通WAF。

  > 说明
  > - 在开通页面，需选择**日志服务**为**开启**，并选择合适的存储时长和存储容量。
  > - 如果您开通的是基础进阶版WAF，可升级为高级版、企业版或旗舰版WAF后再开通WAF日志服务，详情请参见升级实例。

- 已接入网站，详情请参见接入WAF。

操作步骤

1. 登录Web应用防火墙控制台。
2. 在左侧导航栏，选择**安全运营 > 日志服务**。
3. 根据页面提示，授权WAF使用AliyunWAFAccessingLogRole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WAF日志无法正常推送到日志服务。

4. 在**日志服务**页面，单击**立即开启**。
5. 在**日志服务**页面，选择已接入WAF防护的网站域名，单击域名右侧的**状态**开关，为该网站域名开启WAF日志服务。

后续步骤

日志服务采集到WAF日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作，详情请参见云产品日志通用操作。

4.10.3. 管理日志存储空间

本文介绍如何在Web应用防火墙管理控制台上，管理日志存储空间。

操作步骤

1. 登录Web应用防火墙控制台。
2. 在左侧导航栏，选择**安全运营 > 日志服务**。
3. 在**日志服务**页面，管理日志存储空间，具体操作如下所示。
   - 查看日志存储空间。

     > 说明 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况存在两个小时的延迟。当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储影响日志存储的连续性。

   - 升级存储空间：单击**升级**，升级存储容量。
   - 清空存储空间：单击**清空**，清空存储空间。

     开通WAF日志服务后，您总共有4次清空日志存储空间的机会。

     > 警告 日志清空后将无法复原，请务必谨慎使用清空功能。

4.10.4. 日志字段详情

本文介绍网站域名的访问日志和攻防日志的字段详情。

字段	说明
__topic__	日志主题，固定为waf_access_log。
account_action	客户端请求命中的账户安全规则对应的防护动作。取值仅有*block*，表示拦截。更多信息，请参见WAF防护动作（action）说明。
account_rule_id	客户端请求命中的账户安全规则的ID。
account_test	客户端请求命中的账户安全规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
acl_action	客户端请求命中的IP地址黑名单、自定义防护策略（ACL访问控制）规则对应的防护动作。取值为block、captcha_strict、captcha、js、captcha_strict_pass、captcha_pass、js_pass。更多信息，请参见WAF防护动作（action）说明。
acl_rule_id	客户端请求命中的IP地址黑名单、自定义防护策略（ACL访问控制）规则的ID。
acl_rule_type	客户端请求命中的IP地址黑名单、自定义防护策略（ACL访问控制）规则的类型。取值： • custom：自定义防护策略（ACL访问控制）规则。 • blacklist：IP地址黑名单规则。
acl_test	客户端请求命中的IP地址黑名单、自定义防护策略（ACL访问控制）规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
algorithm_rule_id	客户端请求命中的典型爬虫行为识别规则的ID。
antiscan_action	客户端请求命中的扫描防护规则对应的防护动作。取值仅有*block*，表示拦截。更多信息，请参见WAF防护动作（action）说明。
antiscan_rule_id	客户端请求命中的扫描防护规则的ID。
antiscan_rule_type	客户端请求命中的扫描防护规则的类型。取值： • highfreq：高频Web攻击封禁规则。 • dirscan：目录遍历防护规则。 • scantools：扫描工具封禁规则。 • collaborative：协同防御规则。
antiscan_test	客户端请求命中的扫描防护规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
block_action	触发了拦截动作的WAF防护类型。详细说明如下： 注意 由于WAF功能升级，该字段已失效。新增final_plugin字段用于替代该字段。如果您在业务中使用了block_action，请尽快将其替换成final_plugin。 • tmd：CC攻击防护。 • waf：Web应用攻击防护。 • acl：精准访问控制。 • deeplearning：深度学习引擎。 • antiscan：扫描防护。 • antifraud：数据风控。 • antibot：防爬封禁。
body_bytes_sent	客户端请求体的字节数。
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间使用英文逗号（,）分隔。

字段	说明
cc_action	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）规则对应的防护动作。取值为block、captcha、js、captcha_pass和js_pass。更多信息，请参见WAF防护动作（action）说明。
cc_blocks	是否被CC防护功能拦截。取值： • 1表示拦截。 • 其他值均表示通过。
cc_rule_id	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：自定义防护策略（CC攻击防护）规则。 • system：CC安全防护规则。
cc_test	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）规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
content_type	被请求的内容类型。
deeplearning_action	客户端请求命中的深度学习引擎规则对应的防护动作。取值仅有*block*，表示拦截。更多信息，请参见WAF防护动作（action）说明。
deeplearning_rule_id	客户端请求命中的深度学习引擎规则的ID。
deeplearning_rule_type	客户端请求命中的深度学习引擎规则的类型。取值： • xss：跨站脚本防护规则。 • code_exec：代码执行防护规则。 • webshell：webshell防护规则。 • sqli：SQL注入防护规则。 • lfilei：本地文件包含防护规则。 • rfilei：远程文件包含防护规则。 • crlf：CRLF注入防护规则。 • other：其他防护规则。
deeplearning_test	客户端请求命中的深度学习引擎规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
dlp_rule_id	客户端请求命中的防敏感信息泄露规则的ID。
dlp_test	客户端请求命中的防敏感信息泄露规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
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。
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。
final_action	WAF对客户端请求最终执行的防护动作。取值为block、captcha_strict、captcha和js。更多信息，请参见WAF防护动作（action）说明。 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验证后触发放行的情况），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*）> 严格滑块验证（*captcha_strict*）> 普通滑块验证（*captcha*）> JS验证（*js*）。

字段	说明
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。取值： • waf：规则防护引擎。 • deeplearning：深度学习引擎。 • dlp：防敏感信息泄露。 • account：账户安全。 • normalized：主动防御。 • acl：IP地址黑名单、自定义防护策略（ACL访问控制）。 • cc：CC安全防护、自定义防护策略（CC攻击防护）。 • antiscan：扫描防护。 • scene：场景化配置。 • antifraud：数据风控。 • intelligence：爬虫威胁情报。 • algorithm：典型爬虫行为识别。 • wxbb：App防护。 您可以在Web应用防火墙控制台的**防护配置 > 网站防护**页面，配置以上防护模块。关于不同防护模块的介绍，请参见网站防护配置概述。 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验证后触发放行的情况），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。
host	客户端请求头部的Host字段，表示被访问的域名（基于您的业务设置，也可能是IP地址）。
http_cookie	客户端请求头部的Cookie字段，表示访问来源客户端的Cookie信息。
http_referer	客户端请求头部的Referer字段，表示请求的来源URL信息。 如果请求无来源URL信息，则该字段显示短划线（-）。
http_user_agent	客户端请求头部的User-Agent字段，包含请求来源的客户端浏览器标识、操作系统标识等信息。
http_x_forwarded_for	客户端请求头部的X-Forwarded-For（XFF）字段，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，取值： • 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（-）。
normalized_action	客户端请求命中的主动防御规则对应的防护动作。取值为block和continue。更多信息，请参见WAF防护动作（action）说明。
normalized_rule_id	客户端请求命中的主动防御规则的ID。
normalized_rule_type	客户端请求命中的主动防御规则的类型。取值： • User-Agent：User-Agent基线规则（即请求头的User-Agent字段不在基线范围。其他规则类型的含义与此类似）。 • Referer：Referer基线规则。 • URL：URL基线规则。 • Cookie：Cookie基线规则。 • Body：Body基线规则。
normalized_test	客户端请求命中的主动防御规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
querystring	客户端请求中的查询字符串，具体指被请求URL中问号（?）后面的部分。

字段	说明
real_client_ip	WAF对客户端请求进行分析后，判定发起该请求的真实客户端IP地址，便于您在业务中直接使用。 WAF无法判定真实客户端IP地址时（例如，由于用户通过代理服务器访问、请求头中IP字段有误等），该字段显示短划线（-）。
region	WAF实例的地域ID。取值： • cn：中国内地。 • int：海外地区。
remote_addr	与WAF建立连接的IP地址。 如果WAF与客户端直接连接，该字段等同于客户端IP；如果WAF前面还有其他七层代理（例如，CDN），该字段表示上一级代理的IP地址。
remote_port	与WAF建立连接的端口。 如果WAF与客户端直接连接，该字段等同于客户端端口；如果WAF前面还有其他七层代理（例如，CDN），该字段表示上一级代理的端口。
request_length	客户端请求的字节数，包含请求行、请求头和请求体。单位：字节。
request_method	客户端请求的请求方法。
request_path	被请求的相对路径，具体指被请求URL中域名后面且问号（?）前面的部分（不包含查询字符串）。
request_time_msec	WAF处理客户端请求所用的时间。单位：毫秒。
request_traceid	WAF为客户端请求生成的唯一标识。
scene_action	客户端请求命中的场景化配置规则对应的防护动作。取值为block、captcha、js、captcha_pass和js_pass。更多信息，请参见WAF防护动作（action）说明。
scene_id	客户端请求命中的场景化配置规则对应的场景ID。
scene_rule_id	客户端请求命中的场景化配置规则的ID。
scene_rule_type	客户端请求命中的场景化配置规则的类型。取值： • bot_aialgo：AI智能防护规则。 • js：简单脚本过滤规则。 • intelligence：爬虫威胁情报库匹配、IDC黑名单封禁规则。 • sdk：App（已集成SDK）签名异常、设备特征异常规则。 • cc：IP限速、自定义会话限速规则。
scene_test	客户端请求命中的场景化配置规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
server_port	被请求的目的端口。
server_protocol	源站服务器响应WAF回源请求的协议及版本号。
ssl_cipher	客户端请求使用的加密套件。
ssl_protocol	客户端请求使用的SSL/TLS协议和版本。
status	WAF返回给客户端的HTTP响应状态信息。
time	客户端请求的发起时间。
ua_browser	发起请求的浏览器的名称。
ua_browser_family	发起请求的浏览器所属系列。
ua_browser_type	发起请求的浏览器的类型。
ua_browser_version	发起请求的浏览器的版本。

字段	说明
ua_device_type	发起请求的客户端的设备类型。
ua_os	发起请求的客户端的操作系统类型。
ua_os_family	发起请求的客户端所属的操作系统系列。
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	当前WAF实例所属的阿里云账号ID。
waf_action	客户端请求命中的规则防护引擎规则对应的防护动作。取值仅有*block*，表示拦截。更多信息，请参见WAF防护动作（action）说明。
waf_test	客户端请求命中的规则防护引擎规则对应的防护模式。取值： • true：观察模式，即仅记录日志，不触发拦截等防护动作。 • false：防护模式，WAF对命中防护规则的请求执行拦截等防护动作。
waf_rule_id	客户端请求命中的规则防护引擎规则的ID。
waf_rule_type	客户端请求命中的规则防护引擎规则的类型。取值： • xss：跨站脚本防护规则。 • code_exec：代码执行防护规则。 • webshell：webshell防护规则。 • sqli：SQL注入防护规则。 • lfilei：本地文件包含防护规则。 • rfilei：远程文件包含防护规则。 • crlf：CRLF注入防护规则。 • other：其他防护规则。

4.11. 云防火墙日志

4.11.1. 使用前须知

阿里云云防火墙联合日志服务推出日志分析功能，提供互联网流量日志的实时采集、查询、分析、加工、消费等一站式服务，满足等保合规要求。本文介绍云防火墙日志分析功能相关的资产详情、费用说明及使用限制等。

资产详情

- 专属Project和Logstore

  开通云防火墙日志分析功能后，日志服务默认为您创建一个名为cloudfirewall-project-阿里云账号ID-ap-southeast-1的Project，以及一个名为cloudfirewall-logstore的专属Logstore。

  > **说明** 如果您使用的是金融云，开通云防火墙日志分析功能后，日志服务默认为您创建一个名为cloudfirewall-project-阿里云账号ID-cn-hangzhou-finance，以及一个名为cloudfirewall-logstore的专属Logstore。

- 专属仪表盘

  默认生成1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，详情请参见创建仪表盘。

仪表盘	说明
报表	展示云防火墙的基本指标、流量来源流出分布、系统稳定性等。

费用说明

由云防火墙售卖日志分析功能，根据日志存储时长和日志存储容量收取费用。如果您要执行数据加工、投递、从外网接入点流式读取数据操作，由日志服务收取加工计算费用、数据投递费用和外网读取流量费用，详情请参见日志服务产品定价。

使用限制

- 专属Logstore不支持写入其他数据，但在查询、统计、告警、消费等功能上无特殊限制。
- 不支持删除专属Logstore及修改专属Logstore的存储时间。
- 阿里云日志服务产品需处于可用状态（无欠费），否则云防火墙日志分析功能暂停使用。
- 请确保日志存储空间充足，当日志存储空间被占满后，无法写入新的日志。

> **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。

功能优势

- 等保合规：存储六个月的网站访问日志，助力网站符合等保合规要求。
- 配置简单：轻松配置即可实时采集互联网流量日志。
- 实时分析：依托日志服务，提供实时日志分析能力、开箱即用的报表中心，让您对经过云防火墙的互联网流量以及用户访问细节了如指掌。
- 实时告警：支持基于特定指标定制近实时的监测与告警，确保关键业务异常时可及时响应。
- 生态体系：支持对接其他生态如流计算、云存储、可视化方案，进一步挖掘数据价值。

4.11.2. 开通日志分析功能

本文介绍如何在云防火墙控制台上开通日志分析功能，从而将互联网流量日志采集到日志服务中。

操作步骤

1. 登录云防火墙管理控制台。
2. 在左侧导航栏，单击**日志分析 > 日志分析**。
3. 单击**立即开通**。
4. 购买云防火墙，并根据页面提示完成支付。

   其中，**日志分析**需选择为**是**，并根据业务需求选择合适的日志存储容量。其他参数配置请参见购买云防火墙服务。
5. 在**日志分析**页面，选择**互联网流量日志**，开启其状态开关。

> **注意** 建议您在日志分析功能使用期间，定期关注日志存储空间的使用情况，当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储。

后续步骤

在日志服务采集到互联网流量日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作。更多信息，请参见云产品日志通用操作。

4.11.3. 管理日志存储空间

本文介绍如何在云防火墙控制台上管理日志存储空间，包括查看、升级及清空存储空间。

操作步骤

1. 登录云防火墙管理控制台。
2. 在左侧导航栏，单击**日志分析 > 日志分析**。
3. 在**日志分析**页面，管理日志存储空间，具体操作如下所示。
   - 查看日志存储空间。

     > **说明** 控制台中显示的日志存储空间用量并非实时更新，与实际使用情况间存在两个小时的延迟。当日志存储空间使用量超过70%时，请及时升级日志存储量规格，避免新产生的日志无法存储。
   - 升级存储空间：单击**升级容量**，升级存储空间。
   - 清空存储空间：单击 **清空**，清空存储空间。

清空存储空间有一定的次数限制。

> 警告 清空存储空间后将无法复原日志数据，请务必谨慎使用清空功能。

4.11.4. 日志字段详情

本文介绍互联网流量日志的字段详情。

字段名称	说明
__topic__	日志主题，固定为cloudfirewall_access_log
log_type	日志类型，目前固定为internet_log，表示互联网流量日志。
aliuid	阿里云账号ID
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
domain	域名
dst_ip	目的IP地址
dst_port	目的端口
end_time	会话结束时间，Unix时间戳，单位：秒
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，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，单位：字节
out_packet_count	出方向报文数
out_pps	出方向流量大小，单位：pps
region_id	访问流量所属地域，例如cn-beijing
rule_result	命中规则结果，包括： • 通过：pass • 告警：alert • 丢弃：drop
src_ip	源IP地址
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，分钟取整数，Unix时间戳，单位：秒

字段名称	说明
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
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）

4.12. 负载均衡7层访问日志

4.12.1. 使用前须知

日志服务联合负载均衡（SLB）推出访问日志功能，您可以通过负载均衡的访问日志了解客户端用户行为、客户端用户的地域分布，排查问题等。本文介绍负载均衡7层访问日志功能相关的资产详情、费用说明、使用限制等信息。

资产说明

- 自定义日志服务Project、Logstore
  - 该Logstore默认开启索引，并配置部分字段的索引。

    您可以修改索引，修改索引后只对新数据生效，您还可以对历史数据重建索引。具体操作，请参见重建索引。
  - 该Logstore默认永久保存日志，您也可以修改日志存储时间。具体操作，请参见管理Logstore。

> **注意** 请勿删除负载均衡7层访问日志相关的日志服务Project和Logstore，否则将无法正常采集日志到日志服务。

- 专属仪表盘

  默认生成2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。您可以自定义创建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
slb_layer7_operation_center_cn	展示总体运营情况，包括PV、UV、请求成功率、请求报文流量、返回客户端流量等内容。
slb_layer7_access_center_cn	展示访问细节信息，包括：客户端PV分布、请求方法PV趋势、状态码PV趋势、top客户端、请求报文流量拓扑等内容。

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。

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，请参见计费项。

功能优势

- 简单：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。日志服务可以1秒钟分析一亿日志，且相较于开源方案有明显成本优势。
- 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。

4.12.2. 开通访问日志功能

本文介绍如何在负载均衡控制台上开通访问日志功能，将负载均衡7层访问日志采集到日志服务中。

前提条件

- 已创建负载均衡实例。更多信息，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。更多信息，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。更多信息，请参见创建Project和Logstore。

操作步骤

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。更多信息，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。

后续步骤

日志服务采集到负载均衡7层访问日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作。更多信息，请参见云产品日志通用操作。

4.12.3. 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 referer header的内容。

字段	说明
http_user_agent	Proxy收到的请求报文中HTTP user-agent header的内容。
http_x_forwarded_for	Proxy收到的请求报文中HTTP x-forwarded-for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scheme，http或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。
slbid	负载均衡实例ID。
ssl_cipher	建立SSL连接使用的密码。
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	VIP地址。
write_response_time	Proxy收到后端的请求后，把这个请求发送给客户端的时间。单位：毫秒。

4.13. 负载均衡4层秒级监控指标

4.13.1. 使用前须知

日志服务与负载均衡联合推出负载均衡4层秒级监控功能。您可以通过秒级监控指标查看负载均衡相关的秒级流量、CPS、错误率等信息，进行更精细的服务监控和问题定位。

资产说明

- 自定义日志服务Project和MetricStore

  > **说明** 请勿删除负载均衡4层秒级监控指标相关的日志服务Project和MetricStore，否则将无法正常推送秒级监控指标到日志服务。

- 专属仪表盘

  默认生成2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。您可以自定义创建仪表盘用于查询和分析结果的展示。更多信息，请参见创建仪表盘。

仪表盘	说明
SLB四层秒级监控	展示目标SLB实例、端口、出流量、入流量、连接数等指标的变化趋势信息。

仪表盘	说明
SLB四层秒级监控分析	展示目标SLB实例、端口、出流量、入流量、连接数等指标的统计信息。

费用说明

- 目前，负载均衡不针对秒级监控功能收取额外费用。
- 负载均衡将秒级监控指标推送到日志服务后，日志服务根据存储空间、读取流量、请求次数、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TCP或UDP协议监听的负载均衡实例才支持秒级监控功能。
- 日志服务Project与负载均衡实例需处于同一地域。

4.13.2. 开通秒级监控

本文介绍如何在负载均衡控制台上开通秒级监控功能，将负载均衡4层秒级监控指标推送到日志服务中。

前提条件

- 已创建负载均衡实例。具体操作，请参见创建实例。
- 已为负载均衡实例配置TCP监听或UDP监听。具体操作，请参见添加TCP监听、添加UDP监听。
- 在负载均衡实例所在地域，已创建日志服务Project和MetricStore。具体操作，请参见创建Project和创建MetricStore。

操作步骤

1. 登录负载均衡控制台。
2. 在左侧导航栏，选择**传统型负载均衡 CLB（原SLB） > 实例管理**。
3. 在实例列表中，单击目标实例。
4. 单击**高精度秒级监控**。
5. 开启当前地域的高精度秒级监控功能。

   如果您所在地域已开启高精度秒级监控功能，请跳过此步骤。

   i. 单击**开启当前地域的高精度秒级监控**。

   ii. 在**开通负载均衡高精度秒级监控**对话框中，选择您已创建的Project和MetricStore，选中**我已知晓上述信息**，然后单击**确定**。

   > 注意
   > - 在执行此操作时，系统自动创建一个名为AliyunServiceRoleForSlbLogDelivery的RAM角色，用于授权负载均衡使用该角色将秒级监控指标推送到日志服务。
   > - 请勿取消授权或删除RAM角色，否则将导致秒级监控指标无法正常推送到日志服务。

6. 为目标TCP监听或UDP监听开通秒级监控功能。

   i. 单击**设置**。

   ii. 在**设置**页签中，打开目标TCP监听或UDP监听对应的**秒级监控**开关。

后续步骤

日志服务采集到负载均衡4层秒级监控数据后，您可以在日志服务控制台上执行查询与分析、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。

4.13.3. 秒级监控指标详情

本文介绍负载均衡4层秒级监控指标的详情。

本文涉及的指标遵循时序数据格式。更多信息，请参见时序数据（Metric）。您可以使用PromQL或SQL语句进行查询和分析。更多信息，请参见时序数据查询与分析简介。

指标说明

指标	说明
actConnsPS	每秒活跃连接数
connsPS	每秒新建连接数

指标	说明
dropConnPS	每秒丢弃连接数
failConnPS	每秒失败连接数
inActConnPS	每秒非活跃连接数
inBitsPS	每秒入比特数，单位：bit/s。
inDropBitsPS	每秒入丢弃比特数，单位：bit/s。
inDropPktsPS	每秒入丢弃包数
inPktsPS	每秒入包数
aclDropBitsPS	每秒ACL丢弃比特数，单位：bit/s。
aclDropPktsPS	每秒ACL丢弃包数
maxConnsPs	每秒并发连接数
outBitsPS	每秒出比特数，单位：bit/s。
outDropBitsPS	每秒出丢弃比特数，单位：bit/s。
outDropPktsPS	每秒出丢弃包数

标签说明

标签	说明
lbId	负载均衡实例ID
listenerId	监听实例ID
protocol	协议
vip	监听的IP地址
vport	监听的端口

4.14. VPC流日志

4.14.1. 使用前须知

阿里云专有网络（VPC）联合日志服务提供流日志功能，用于记录VPC网络中弹性网卡流量、VPC流量及交换机流量等，帮助您检查访问控制规则、监控网络流量和排查网络故障。本文介绍流日志功能相关的资产详情、费用说明以及使用限制。

流日志功能捕获的流量信息以日志方式写入日志服务中。每条日志会捕获特定捕获窗口中的特定五元组网络流，捕获窗口大约为10分钟，该段时间内流日志功能先聚合数据，再发布日志。如果您选择为VPC或交换机创建流日志，则会捕获VPC和交换机中所有弹性网卡的流量，包括开启流日志功能后新建的弹性网卡。

资产说明

- 自定义日志服务Project、Logstore

  > **说明**
  > - 请勿删除VPC流日志相关的日志服务Project和Logstore，否则将无法正常采集日志到日志服务。
  > - 开通VPC流日志功能后，VPC流日志相关的Logstore的数据保存时间被强制修改为7天。

- 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，详情请参见创建仪表盘。

仪表盘	说明
Logstore Name-vpc_flow_log_traffic_cn	展示总体运营信息，包括源地址字节数热力图、Top 10 ACCEPT字节数的目标端口、各协议的每分钟字节数等内容。
Logstore Name-vpc_flow_log_rejection_cn	展示安全组和网络ACL拒绝记录的流量信息，包括REJECT总字节数、REJECT字节数占比、REJECT总包数、REJECT包数占比等内容。
Logstore Name-vpc_flow_log_overview_cn	展示总体概览信息，包括Action总次数、ACCEPT总字节数、REJECT总字节数、ACCEPT总包数等内容。

费用说明

目前，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+日志服务的服务费。

- 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。

- 日志服务的服务费

  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，详情请参见日志服务产品定价。

使用限制

- 地域限制
  - 日志服务Project和当前资源实例需处于同一地域。
  - 流日志功能正在公测中。您可以提交工单申请。

    流日志功能支持的地域如下表所示。

区域	支持流日志的地域
亚太	华北1（青岛）、华北2（北京）、华北3（张家口）、华北5（呼和浩特）、华北6（乌兰察布）、华东1（杭州）、华东2（上海）、华南1（深圳）、华南2（河源）、华南3（广州）、西南1（成都）、中国（香港）、日本（东京）、新加坡、澳大利亚（悉尼）、马来西亚（吉隆坡）、印度尼西亚（雅加达）
欧洲与美洲	美国（硅谷）、美国（弗吉尼亚）、德国（法兰克福）、英国（伦敦）
中东与印度	印度（孟买）、阿联酋（迪拜）

- 资源限制

资源	默认限制	提升配额
每个地域支持创建的流日志实例的数量	10个	提交工单。
不支持创建流日志的VPC	VPC中含有以下实例规格族中的任一实例： ecs.c1、ecs.c2、ecs.c4、ecs.ce4、ecs.cm4、ecs.d1、ecs.e3、ecs.e4、ecs.ga1、ecs.gn4、ecs.gn5、ecs.i1、ecs.m1、ecs.m2、ecs.mn4、ecs.n1、ecs.n2、ecs.n4、ecs.s1、ecs.s2、ecs.s3、ecs.se1、ecs.sn1、ecs.sn2、ecs.t1、ecs.xn4	升级不支持VPC高级功能的ECS实例的规格或释放不支持VPC高级功能的ECS实例。 ○ 升级操作，请参见包年包月实例升配规格和按量付费实例变配规格。

资源	默认限制	○ 释放操作，请参见释放实例。 提升配额
不支持创建流日志的交换机	交换机所属的VPC中含有以下实例规格族中的任一实例： ecs.c1、ecs.c2、ecs.c4、ecs.ce4、ecs.cm4、ecs.d1、ecs.e3、ecs.e4、ecs.ga1、ecs.gn4、ecs.gn5、ecs.i1、ecs.m1、ecs.m2、ecs.mn4、ecs.n1、ecs.n2、ecs.n4、ecs.s1、ecs.s2、ecs.s3、ecs.se1、ecs.sn1、ecs.sn2、ecs.t1、ecs.xn4	说明 如果您的VPC、交换机所属VPC、弹性网卡所属VPC中含有实例规格族限制中的任一实例，且您已经创建了流日志，为了保证正常使用流日志功能，请升级实例规格。
不支持创建流日志的弹性网卡	弹性网卡所属的VPC中含有以下实例规格族中的任一实例： ecs.c1、ecs.c2、ecs.c4、ecs.ce4、ecs.cm4、ecs.d1、ecs.e3、ecs.e4、ecs.ga1、ecs.gn4、ecs.gn5、ecs.i1、ecs.m1、ecs.m2、ecs.mn4、ecs.n1、ecs.n2、ecs.n4、ecs.s1、ecs.s2、ecs.s3、ecs.se1、ecs.sn1、ecs.sn2、ecs.t1、ecs.xn4	

4.14.2. 开通流日志功能

本文介绍如何在VPC控制台上开通流日志功能，将流日志采集到日志服务中。

前提条件

- 已创建资源实例。具体操作，请参见创建弹性网卡、创建和管理专有网络和创建和管理交换机。
- 在资源实例所在地域，已创建日志服务Project和Logstore。具体操作，请参见创建Project和创建Logstore。

操作步骤

> 说明 如果您使用RAM用户进行流日志功能开通，则需先为RAM用户授权。具体操作，请参见RAM用户授权。

1. 登录专有网络管理控制台。
2. 在左侧导航栏，选择**运维与监控 > 流日志**。
3. 首次使用流日志功能时，单击**立即授权**。根据页面提示，完成授权。

   授权成功后才能保证VPC流日志被写入到日志服务中。

   > 注意 请勿取消授权或删除RAM角色，否则将导致VPC流日志无法正常推送到日志服务。

4. 在顶部菜单栏中，选择实例所在地域。

   支持流日志功能的地域，请参见功能发布及地域支持情况。
5. 在**流日志**页面，单击**创建流日志**。
6. 在**创建流日志**页面，配置相关参数，单击**确定**。

参数	说明
流日志名称	输入流日志名称。 名称长度为2~128个字符，以英文字母或中文开始，可包含数字，短划线（-）和下划线（_）。

参数	说明
资源类型	选择流量的资源类型，然后选择相应的资源。支持选择以下资源类型： ○ **专有网络**：捕获指定的VPC内所有弹性网卡的流量信息。如果VPC内有属于不支持捕获流日志的ECS实例规格族的ECS实例，则不能捕获该ECS实例弹性网卡的流量信息。 ○ **交换机**：捕获指定的交换机内所有弹性网卡的流量信息。如果交换机内有属于不支持捕获流日志的ECS实例规格族的ECS实例，则不能捕获该ECS实例弹性网卡的流量信息。 ○ **弹性网卡**：捕获指定的弹性网卡的流量信息。如果该弹性网卡绑定的ECS实例属于不支持捕获流日志的ECS实例规格族，则不能捕获该弹性网卡的流量信息。 不支持捕获流日志的ECS实例规格族： ecs.c1、ecs.c2、ecs.c4、ecs.ce4、ecs.cm4、ecs.d1、ecs.e3、ecs.e4、ecs.ga1、ecs.gn4、ecs.gn5、ecs.i1、ecs.m1、ecs.m2、ecs.mn4、ecs.n1、ecs.n2、ecs.n4、ecs.s1、ecs.s2、ecs.s3、ecs.se1、ecs.sn1、ecs.sn2、ecs.t1、ecs.xn4 如需捕获流日志，请升级ECS实例规格。 具体操作，请参见包年包月实例升配规格和按量付费实例变配规格。
资源实例	选择待捕获流量的资源实例。
流量类型	选择流量的类型。 ○ **全部流量**：捕获指定资源的全部流量。 ○ **被访问控制允许的流量**：捕获指定资源被安全组规则允许的流量。 ○ **被访问控制拒绝的流量**：捕获指定资源被安全组规则拒绝的流量。
项目（Project）	选择用于管理VPC流日志相关资源（Logstore、仪表盘等）的日志服务Project。 ○ **选择现有Project**：从已有的Project中进行选择。 ○ **新建Project**：新建一个Project。更多信息，请参见创建Project。
日志库（Logstore）	选择用于存储VPC流日志的Logstore。 ○ **选择现有 Logstore**：从已有的Logstore中进行选择。 ○ **新建 Logstore**：新建一个Logstore。更多信息，请参见创建Logstore。
开启流日志分析报表功能	开启该功能后，日志服务为该Logstore开启索引并建立仪表盘。 开启索引后，您才能查询和分析VPC流日志。
描述	输入流日志的描述。 描述信息长度为2~256个字符，不能以 `http://` 和 `https://` 开头。

执行结果

后续步骤

日志服务采集到VPC流日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作。更多信息，请参见云产品日志通用操作。

4.14.3. 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	账号ID
eni-id	弹性网卡ID

字段	说明
srcaddr	源地址
srcport	源端口
dstaddr	目的地址
dstport	目的端口
protocol	流量的IANA协议编号，详情请参见 Internet 协议编号。
direction	流量方向 • 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 • 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 • ACCEPT：安全组允许记录的流量。 • REJECT：安全组未允许记录的流量。

4.15. 弹性公网IP日志

4.15.1. 使用前须知

阿里云弹性公网（EIP）联合日志服务推出高精度秒级监控功能，以日志形式将高精度网络带宽监控数据推送到日志服务，帮助您实时监控互联网业务流量变化，及时调整弹性公网IP的带宽峰值。本文介绍弹性公网IP日志相关的资产详情、费用说明以及使用限制。

资产说明

- 自定义日志服务Project、Logstore

  > **说明**
  > - 请勿删除弹性公网IP相关的日志服务Project和Logstore，否则将无法正常采集日志到日志服务。
  > - 开启弹性公网IP秒级监控功能后，弹性公网IP相关的Logstore的数据保存时间被强制修改为7天。

- 专属仪表盘

  默认生成1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，详情请参见创建仪表盘。

仪表盘	说明
eip_monitoring	实时监测互联网业务流量变化，包括出入方向秒级带宽峰值、出入方向秒级包速率、出入方向秒级丢包速率、出入方向新建TCP连接速率等信息。

费用说明

- 目前，弹性公网IP不针对日志功能收取额外费用。
- 弹性公网IP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，详情请参见日志服务产品定价。

使用限制

- 弹性公网IP实例和日志服务Project需处于同一地域。
- 每个阿里云账号可开启10个弹性公网IP的高精度监控。

  如果您需要开启更多实例的高精度监控，请提交工单申请。

4.15.2. 开启秒级监控

本文介绍如何在VPC控制台上开启弹性公网IP的秒级监控功能，以日志形式将监控数据采集到日志服务中。

前提条件

- 已购买弹性公网IP，详情请参见申请EIP。
- 已在弹性公网IP实例所在地域，创建日志服务Project和Logstore，详情请参见步骤二：创建Project和Logstore。

操作步骤

1. 登录专有网络管理控制台。
2. 在页面上方，选择弹性公网IP所属地域。
3. 在左侧导航栏，单击**公网访问 > 公网质量工具箱**。
4. 在**公网质量工具箱**页面，单击**高精度秒级监控工具**。
5. 在**高精度秒级监控**页面，根据提示，完成授权。

   授权VPC使用AliyunVPCLogArchiveRole角色访问日志服务资源。

   > **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。更多信息，请参见RAM用户授权。
   > - 请勿取消授权或删除RAM角色，否则将导致弹性公网IP的秒级监控数据无法正常推送到日志服务。

6. 在**高精度秒级监控**页面，找到目标弹性公网IP，单击**开启秒级监控**。
7. 在**日志设置**页面，选择logProject和logStore，单击**确定**。

后续步骤

日志服务采集到弹性公网IP日志后，您可以执行查询分析、下载、投递、加工日、创建告警等操作。具体操作，请参见云产品日志通用操作。

4.15.3. 日志字段详情

本文介绍弹性公网IP日志的字段详情。

字段	含义
__topic__	日志主题，固定为eip
type	标明秒级峰值数据属性信息
tid	发送的ID标识
time	发送时间
gw_ip	网关IP地址
eip	弹性公网IP地址
in_Bps	入方向字节速率，单位：Bytes/s
out_Bps	出方向字节速率，单位：Bytes/s
in_pps	入方向数据包速率，单位：pps
out_pps	出方向数据包速率，单位：pps
In_syn_speed	入方向新建TCP的连接速率，单位：pps
out_syn_speed	出方向新建TCP连接时的速率，单位：pps

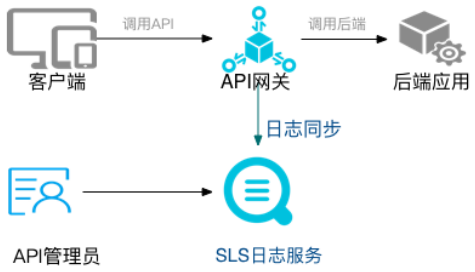
字段	含义
In_syn_ack_speed	入方向确认TCP连接时的速率，单位：pps
out_syn_ack_speed	出方向确认TCP连接时的速率，单位：pps
In_fin_speed	入方向关闭TCP连接时的速率，单位：pps
out_fin_speed	出方向关闭TCP连接时的速率，单位：pps
In_rst_speed	入方向重置TCP连接时的速率，单位：pps
out_rst_speed	出方向重置TCP连接时的速率，单位：pps
out_ratelimit_drop_speed	出方向限速丢包速率，单位：pps
in_ratelimit_drop_speed	入方向限速丢包速率，单位：pps
out_drop_speed	出方向丢包速率，单位：pps
in_drop_speed	入方向丢包速率，单位：pps
timestamp	时间戳，单位：ms

4.16. API网关访问日志

4.16.1. 使用前须知

阿里云API网关联合日志服务推出日志管理功能，提供API访问日志的实时采集、查询、分析、加工、消费等一站式服务。本文介绍API网关日志管理功能相关的资产详情、费用说明以及使用限制。

阿里云API网关提供API托管服务，在微服务聚合、前后端分离、系统集成上为您提供诸多便利。每一次API请求对应生成一条访问日志，内容包括调用者IP地址、请求URL、响应延迟、返回状态码、请求字节数和响应字节数等重要信息，便于您了解Web服务的运行状况。



资产详情

- 自定义日志服务Project和Logstore

  > **说明** 请勿删除API网关日志管理功能对应的日志服务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1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
*Logstore Name*_apigateway访问日志	展示API网关的全局统计信息，包括请求量大小、成功率、错误率、延时情况、调用API的App数量、错误情况统计、TOP分组、TOP API、Top延迟等。

费用说明

- API网关不针对日志管理功能收取额外费用。
- API网关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

- API网关实例和日志服务Project需处于同一地域。
- 一个地域仅支持创建一个日志配置，该地域所有的API网关访问日志均通过该配置推送到对应的日志服务Logstore中。

4.16.2. 开通日志管理功能

本文介绍如何在API网关控制台上开通日志管理功能，将API网关访问日志推送到日志服务中。

前提条件

已创建Project和Logstore。具体操作，请参见创建Project和创建Logstore。

操作步骤

> **说明** 如果您使用RAM用户进行日志管理功能开通，则需先为RAM用户授权。更多信息，请参见RAM用户授权。

1. 登录API网关控制台。
2. 在左侧导航栏中，选择**开放API > 日志管理**。
3. 在顶部菜单栏中，选择地域。
4. 在**日志管理**页面，单击**创建日志配置**。
5. 在**创建日志配置**面板中，选择您已创建的Project和Logstore，并单击**确定**。
6. 在**系统提示**对话框中，单击**去SLS控制台配置**。
7. 在日志服务控制台，开启索引。更多信息，请参见配置索引。
   设置索引后，您才可以进行日志查询和分析等操作。

后续步骤

日志服务采集到API网关访问日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作。更多信息，请参见云产品日志通用操作。

4.16.3. 日志字段详情

本文介绍API网关访问日志的字段详情。

字段名称	说明
apiGroupUid	API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	调用的HTTP方法
path	请求的路径
domain	调用的域名
statusCode	HTTP的状态码
errorMessage	错误信息
appId	调用者应用ID
appName	调用者应用名称
clientIp	调用者客户端的IP地址
exception	后端返回的具体错误信息
providerAliUid	API提供者的帐户ID

字段名称	说明
region	地域，例如：cn-hangzhou
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	返回数据大小，单位：字节
serviceLatency	后端延迟时间，单位：毫秒

4.17. ActionTrail访问日志

4.17.1. 使用前须知

阿里云操作审计（ActionTrail）联合日志服务推出操作记录投递功能，将操作记录以日志形式实时采集到日志服务中。日志服务提供实时查询、可视化分析、告警、投递、加工等功能，帮助你实时掌握重要云资产的操作情况。本文介绍ActionTrail操作记录投递功能相关的资产详情、费用说明等信息。

资产说明

- 自定义Project

  您在开启ActionTrail日志功能时，需自定义一个Project。

- 专属Logstore

  开启ActionTrail日志功能后，日志服务默认为您创建一个名为actiontrail_*跟踪名称*的Logstore。

  - 该Logstore默认开启索引，并配置部分字段的索引。

    您可以修改索引，修改索引后只对新数据生效，您还可以对历史数据重建索引。具体操作，请参见重建索引。

  - 该Logstore默认永久保存日志，您也可以修改日志存储时间。具体操作，请参见管理Logstore。

- 专属仪表盘

  默认生成1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。您可以自定义创建仪表盘用于查询结果展示。具体操作，请参见创建仪表盘。

仪表盘	说明
actiontrail_*跟踪名称*_audit_center_cn	展示云资源操作的实时动态，包括PV、UV、来源服务数、事件来源分布、PV/UV趋势等内容。

费用说明

- ActionTrail日志管理功能由ActionTrail侧收取费用。更多信息，请参见计费说明。
- ActionTrail将日志投递到日志服务后，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，计费说明请参见计费项。

使用限制

- 跟踪功能是指将审计事件投递到OSS Bucket或日志服务Logstore中，目前您在所有地域最多只能创建5个跟踪。
- 专属Logstore不支持写入其他数据，但在查询、分析、告警等功能上无特殊限制。

应用场景

- 实时监控名下所有的云资源操作，排查与分析异常操作。您还可以通过日志记录溯源意外删除、高危操作等。
- 通过日志分析结果追踪重要资源操作的分布与来源，并优化对应的产品策略。
- 实时查询分析ActionTrail操作日志，查看所有资源操作的分布、时间趋势等信息，帮助运维人员实时监控云资源运行状况。
- 根据运营需求、数据需求定制多样化的查询分析语句、快速查询等，还可以根据资源使用状况、用户登录情况等定制实时数据大盘。

4.17.2. 开通日志功能

本文介绍如何在ActionTrail控制台上开通日志功能，将ActionTrail操作记录投递到日志服务中。

前提条件

已授权ActionTrail使用AliyunActionTrailDefaultRole角色投递日志到日志服务中。

单击云资源访问授权，根据提示完成授权。

> 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。

操作步骤

1. 登录操作审计控制台。
2. 在左侧导航栏中，单击**操作审计 > 创建跟踪**。
3. 在**创建跟踪**页面，配置相关参数。

   具体参数配置请参见创建单账号跟踪。

   i. 配置跟踪基本属性信息，然后单击**下一步**。

   ii. 配置事件投递信息，然后单击**下一步**。

   此处将事件投递至日志服务，具体参数说明如下表所示。

参数	说明
将事件投递到日志服务SLS	选中**将事件投递到日志服务SLS**。 根据需求，选择**创建新的日志项目**或**选择已有的日志项目**。
日志库所属地域	选择日志库所在地域。
日志项目名称	选择已有的日志项目或新建一个日志项目。

   iii. 预览及确认配置信息，单击**提交**。

后续步骤

日志服务采集到ActionTrail访问日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作，详情请参见云产品日志通用操作。

4.17.3. 日志字段详情

本文介绍ActionTrail操作日志的字段详情。

字段名称	说明
__topic__	日志主题，固定为actiontrail_audit_event。
event	事件主体，JSON格式，事件主体的内容随事件变化。
event.eventId	事件ID，事件唯一标识
event.eventName	事件名称
event.eventSource	事件来源
event.eventType	事件类型
event.eventVersion	事件格式版本，固定为1。
event.acsRegion	事件所在地域
event.requestId	操作云资源的请求ID
event.apiVersion	API版本
event.errorMessage	云资源处理API请求发生错误时，记录的错误消息。
event.serviceName	事件相关的云资源名称，例如Vpc
event.sourceIpAddress	事件发生的源IP地址

字段名称	说明
event.userAgent	发送API请求的客户端代理标识
event.requestParameters.HostId	API请求输入参数中的主机ID
event.requestParameters.Name	API请求输入参数中的名称
event.requestParameters.Region	API请求输入参数中的地域
event.userIdentity.accessKeyId	发起API请求的阿里云账号的AccessKey ID
event.userIdentity.accountId	发起API请求的阿里云主账号ID
event.userIdentity.principalId	请求者ID，例如type为ram-user，则此处为阿里云主账号ID。
event.userIdentity.type	身份类型 • root-account：阿里云主账号 • ram-user：RAM 用户 • assumed-role：RAM角色 • system：阿里云服务
event.userIdentity.userName	身份名称，例如type为ram-user，则此处为RAM用户名。

4.18. 平台操作日志

4.18.1. 使用前须知

阿里云操作审计（ActionTrail）联合日志服务推出平台操作日志（Inner-ActionTrail）功能，提供平台操作日志的实时采集、查询、分析、加工、消费等一站式服务，满足您平台操作日志相关的分析与审计需求。本文介绍平台操作日志相关的资产详情、费用说明及使用限制等。

> **说明** 目前，平台操作日志功能支持采集对象存储OSS的平台操作日志、云服务器ECS的平台操作日志、云数据库RDS的平台操作日志、容器服务Kubernetes版ACK的平台操作日志和E-MapReduce的平台操作日志。

资产说明

- 自定义Project和Logstore
  - 该Logstore默认开启索引，并配置部分字段的索引。
  - 该Logstore默认永久保存日志，您也可以修改日志存储时间，详情请参见管理Logstore。

  > **说明** 请勿删除平台操作日志相关的日志服务Project和Logstore，否则将无法正常采集日志到日志服务。
- 专属仪表盘

  默认生成1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。您可以自定义创建仪表盘用于查询结果展示，详情请参见创建仪表盘。

仪表盘	说明
innertrail_*跟踪名称*_audit_center_cn	展示云资源操作的实时动态，包括PV、UV、来源服务数、事件来源分布、PV/UV趋势等内容。

费用说明

- 目前，ActionTrail不针对平台操作日志功能收取额外费用。
- ActionTrail将平台操作日志投递到日志服务后，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，计费说明请参见日志服务产品定价。

使用限制

- 平台操作日志功能（Inner-ActionTrail）需要您提工单或联系您的销售经理获得使用权限。
- 阿里云日志服务产品需处于可用状态（无欠费）。
- 所有平台操作日志只能投递到1个Logstore中。

- 专属Logstore不支持写入其他数据，但在查询、统计、告警、消费等功能上无特殊限制。
- 不支持修改Logstore的数据存储时长。

功能优势

- 等保合规：存储六个月及以上的平台操作日志，助力产品符合等保合规要求。
- 配置简单：轻松配置即可实时采集平台操作日志。
- 实时分析：依托日志服务产品，提供近实时日志分析能力、开箱即用的报表中心，让您对平台操作日志的分布及细节了如指掌。
- 实时告警：支持基于特定指标定制近实时的监测与告警，确保关键业务异常时可及时响应。
- 生态体系：支持对接其他生态系统（例如流计算、云存储、可视化），进一步挖掘数据价值。

应用场景

- 追踪平台操作日志，查看资产变化原因。
- 近实时查看平台操作日志，审计与评估。
- 输出日志到自建的数据与计算中心。

4.18.2. 开通平台操作日志功能

本文介绍如何通过日志服务控制台采集平台操作日志。

前提条件

- 已提工单或联系您的销售经理获得平台操作日志功能（Inner-ActionTrail）使用权限。
- 已授权ActionTrail使用AliyunActionTrailDefaultRole角色投递日志到日志服务中。

  单击云资源访问授权，根据提示完成授权。

  > 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。

- 已创建Project和Logstore，详情请参见创建Project和Logstore。

操作步骤

1.
2. 在**接入数据**区域，选择**平台操作日志（Inner-ActionTrail）**。

   您也可以登录操作审计控制台，在**平台操作审计 > 跟踪列表**页面中接入平台操作日志。

   > 说明
   > - 如果您是在操作审计控制台开通平台操作日志功能，则默认生成一个名为innertrail_跟踪名称的专属Logstore。
   > - 在日志服务控制台上开通平台操作日志功能后，不会同步到操作审计控制台。如果您已在日志服务控制台中开通，又在操作审计控制台中创建跟踪，则操作审计侧的操作会覆盖日志服务侧的操作。
   > - 如果您在日志服务控制台的**接入数据**区域找不到**平台操作日志（Inner-ActionTrail）**或在操作审计控制台中找不到**平台操作审计 > 跟踪列表**页面，请提工单或联系您的销售经理申请平台操作日志功能（Inner-ActionTrail）使用权限。

3. 选择目标Project和Logstore，单击**下一步**。
4. 在**数据源配置**页签中，单击**下一步**。

说明

- 所有平台操作日志只能投递到1个Logstore中。
- 如果您需要关闭平台操作日志功能，可以在**数据源配置**页签中进行关闭。
- 如果您是在操作审计控制台上开通平台操作日志功能，请在操作审计控制台的**平台操作审计 > 跟踪列表**中删除跟踪，即可关闭日志投递。
- 关闭日志投递后，新产生的Inner-ActionTrail日志不再投递到Logstore中，已投递的日志在Logstore存储时间到期后自动删除。

5. 在**查询分析配置**页签中，单击**下一步**。

   日志服务默认为平台操作日志对应的Logstore开启并配置索引。

后续步骤

日志服务采集到平台操作日志后，您可以执行查询分析、下载、投递、加工日志，创建告警等操作，详情请参见云产品日志通用操作。

4.18.3. 平台操作事件结构定义

本文为您介绍一个平台操作事件包含的字段及其含义，并为您提供相关的示例。

平台操作事件的字段

字段名称	字段说明
EventID	事件ID。事件的唯一标识。
EventVersion	事件定义的版本。
EventProduct	被操作的云服务名称。例如：OSS。
EventName	云服务使用的API所对应的事件名称。例如：Set Bucket Quota Limit。
EventDescription	操作原因。例如：工单ID、内部运维或变更单ID、安全扫描操作ID等。
EventType	操作类型。取值： • CUSTOMER_INITIATED_SUPPORT 阿里云内部人员针对用户授权的技术支持操作。例如：基于工单问题处理等操作事件。 • ALIYUN_INITIATED_SERVICE 阿里云内部人员或系统基于运维需求所发起的操作。例如：因集群硬件设施超过保障期发起的跨集群Bucket迁移。 • ALIYUN_INITIATED_PENALTY 阿里云内部人员或系统基于法律法规要求对用户公开数据进行处置的操作事件。

字段名称	字段说明
EmployeeID	操作人员在阿里云的全球唯一工号加密后的密文。 当事件为系统程序自动操作时，该字段为空；当事件为平台技术人员的手动操作时，该字段非空。必要时您可以通过提交工单向阿里云平台提供该字段以查询某事件的具体操作人。
EventMethod	操作形态。例如：正常的读写操作、使用内部接口的读写操作或者其他操作（例如：备份恢复）等。
ResourceType	事件归属的资源类型。例如：ACS::ACK::Cluster。
ResourceID	资源ID。例如：OSS Bucket ID。
ResourceRegionID	发生事件的资源归属的地域ID。
ResourceOwnerID	发生事件的资源归属的阿里云账号ID。
EventAdditionalDetail	事件的补充信息。
EventTime	操作行为发生的时间（UTC格式）。例如：2021-03-22T05:23:37Z。
EventLevel	操作对应的客户告知程度。取值： • NOTICE：仅在事件中记录。 • WARNING：在事件中记录并主动向客户发送告警。
EventLocation	阿里云技术人员执行操作所处的国家地理位置。例如：CN（中国内地）。

示例

```
{
    "EmployeeID": "64tSfLheCbLra9ClKaUF86J4DkP84p3n6H6sc4BS****",
    "EventAdditionalDetail": "{\"filter\":\"user_id:153915067560****\",\"groupbys\":\"ts,storage_type\",\"max\":\"100000\"
,\"endts\":\"1616947199\",\"orderby\":\"ts\"}",
    "EventDescription": "requestID: 61167C65-B80D-4876-A573-D61DD4238AA2",
    "EventID": "4facb9c7-d970-4f53-af5b-4ee08f51****",
    "EventLevel": "NOTICE",
    "EventLocation": "CN",
    "EventMethod": "Regular Read",
    "EventName": "DescribeK8sResourceGroup",
    "EventProduct": "ACK",
    "EventTime": "2021-03-29T09:44:51Z",
    "EventType": "ALIYUN_INITIATED_SERVICE",
    "EventVersion": "1.0.0",
    "ResourceID": "cd63fb222a3be44a89df72686b343****",
    "ResourceOwnerID": "129242164613****",
    "ResourceRegionID": "cn-hangzhou",
    "ResourceType": "ACS::ACK::Cluster"
}
```

4.19. PolarDB-X 1.0 SQL审计日志

4.19.1. 使用前须知

云原生分布式关系型数据库PolarDB-X 1.0联合日志服务推出SQL审计与分析功能，将SQL审计日志投递到日志服务中，实现日志的实时查询、可视化分析、告警、投递、加工等操作。本文介绍PolarDB-X 1.0 SQL审计日志相关的资产详情、费用说明、使用限制等信息。

资产详情

- 专属日志服务Project和Logstore

  开通实时日志查询功能后，系统默认在对应的地域创建一个名为drds-audit-地域名-阿里云账户ID的Project，以及一个名为drds-audit-log的专属Logstore。

- 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，详情请参见创建仪表盘。

仪表盘	说明
运营中心（简化版）	展示DRDS数据库实例的SQL执行指标、分布、趋势等。
性能中心	展示数据库的性能指标、快慢分布、慢SQL的具体分布与来源等。
安全中心	展示数据库的安全指标、大批量删除、危险SQL执行情况等。

费用说明

- 目前，云原生分布式关系型数据库不针对SQL审计与分析功能收取额外费用。
- 将日志投递到日志服务后，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，详情请参见日志服务产品定价。

限制说明

- 专属Logstore不支持写入其他数据、修改索引等操作，但在查询、统计、告警等功能上无特殊限制。
- 目前仅支持中国（香港）、新加坡地域。

  日志服务的日志审计服务也支持接入PolarDB-X 1.0 SQL审计日志，详情请参见日志审计服务。

4.19.2. 开启SQL审计与分析功能

本文介绍如何在云原生分布式数据库控制台上开启SQL审计与分析功能，将PolarDB-X 1.0 SQL审计日志推送到日志服务。

前提条件

- 已开通阿里云日志服务。
- 已购买PolarDB-X 1.0 实例。
- 已创建数据库。

操作步骤

1. 登录分布式关系型数据库控制台。
2. 在页面左上角，选择实例所在地域。
3. 在**实例列表名称**页面，单击目标实例名称。
4. 在左侧导航栏中，选择**诊断与优化 > SQL审计与分析**。
5. 根据页面提示，授权PolarDB-X 1.0使用AliyunDRDSDefaultRole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。更多信息，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PolarDB-X 1.0 SQL审计日志无法正常推送到日志服务。

6. 开启SQL审计与分析功能。
   i. 选择目标数据库，打开数据库状态栏或日志状态栏中的功能开关。

ii. （可选）导入历史数据。

默认情况下，只推送开启SQL审计与分析功能之后产生的日志到日志服务。如果您要分析未开启SQL审计与分析功能前的日志，可在**需要您配置日志存储时间**对话框中，打开**是否导入历史数据**的功能开关并选择**回溯开始时间**和**回溯结束时间**，导入历史数据。最大支持导入七天的历史数据。

您可以在控制台右上角的**任务进度**中查看日志导入的进度，导入过程需一定的时间，请耐心等待。

iii. 单击**启用**。

执行结果

日志服务采集到PolarDB-X 1.0 SQL审计日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作。更多信息，请参见云产品日志通用操作。

4.19.3. 日志字段详情

本文介绍PolarDB-X 1.0 SQL审计日志的字段详情。

字段	说明
__topic__	日志主题，格式为drds_audit_log_*实例ID_数据库名*，例如：drds_audit_log_drdsxyzabcd_demo_drds_db。
affect_rows	执行SQL返回的行数。 • 增删改时表示影响的行数。 • 查询语句时表示返回的行数。 5.3.4-15378085及之后的实例版本支持该字段。
client_ip	访问PolarDB-X 1.0实例的客户端IP地址。
client_port	访问PolarDB-X 1.0实例的客户端端口。
db_name	PolarDB-X 1.0数据库名。
fail	SQL执行是否出错。 • 0：成功。 • 1：失败。 5.3.4-15378085及之后的实例版本支持该字段。
hint	SQL执行的HINT。
instance_id	PolarDB-X 1.0实例ID。
response_time	响应时间，单位：ms。 5.3.4-15378085及之后的实例版本支持该字段。
sql	执行的SQL语句。
sql_code	模板SQL的HASH值。
sql_time	SQL开始执行的时间，格式为yyyy-MM-dd HH:mm:ss:SSS。

字段	说明
sql_type	SQL类型，包括Select、Insert、Update、Delete、Set、Alter、Create、Drop、Truncate、Replace和Other。
sql_type_detail	SQL解析器的名称。
table_name	数据库表名，多表之间用半角逗号（,）分隔。
trace_id	SQL执行的TRACE ID。如果是事务， 使用跟踪ID、短划线（-）和数字表示，例如drdsabcdxyz-1、drdsabcdxyz-2。
user	执行SQL的用户名。

4.20. RDS SQL审计日志

4.20.1. 使用前须知

阿里云关系型数据库RDS联合日志服务推出RDS SQL审计日志投递功能，将RDS SQL审计日志投递到日志服务中。日志服务提供实时查询、可视化分析、告警、投递、加工等功能。本文介绍RDS SQL审计日志相关的资产详情、费用说明、使用限制等信息。

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、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

- 开启SQL洞察（MySQL）功能后，RDS开始按量计费（每小时费用 = 每小时扣费时的审计日志总量 x 单价）。

  > **说明** 三节点企业版（原金融版）实例的SQL洞察功能免费。

- 投递日志到日志服务后，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。

- 开启SQL洞察（MySQL）功能的RDS实例才能投递SQL审计日志到日志服务。
- 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。

CloudLens for RDS

目前，日志服务还推出了CloudLens for RDS，用于接入RDS SQL日志，支持自动采集、易用性提升，推荐您使用CloudLens for RDS方式。更多信息，请参见CloudLens for RDS。

CloudLens for RDS方式和接入数据-RDS审计方式中的采集配置是互通的，两者主要区别如下：

属性	接入数据-RDS审计	CloudLens for RDS
指定RDS实例粒度	支持	支持
灵活指定存储目标库	支持	支持
自动采集	不支持	支持
手动采集	支持	支持
查看采集状态视图	不支持	支持

4.20.2. 采集RDS SQL审计日志

本文介绍如何通过日志服务控制台采集RDS SQL审计日志。

前提条件

- 已创建RDS实例，且已开启付费版的SQL洞察（RDS MySQL实例）功能。具体操作，请参见创建RDS MySQL实例、SQL洞察。
- 在RDS实例所在地域，已创建日志服务Project和Logstore。具体操作，请参见创建Project和Logstore。

操作步骤

1.
2. 在**接入数据**区域，选择**RDS 审计-云产品**。
3. 在**选择日志空间**页签中，选择您已创建的目标Project和Logstore，单击**下一步**。
4. 在**数据源配置**页签中，完成RAM授权及开通投递，单击**下一步**。

> **说明**
> - 如果您还未授权日志服务分发日志，请单击**RAM授权**右侧的**授权**，根据页面提示完成授权。
> - 如果页面中未显示目标RDS实例或开启投递失败，可能是因为您的RDS实例不符合条件。您可以参见本文中的前提条件检查您的RDS实例。

5. 在**查询分析配置**页签中，单击**下一步**。

日志服务默认为RDS SQL审计日志对应的Logstore开启并配置索引。如果您要修改索引，请参见配置索引。

后续步骤

日志服务采集到RDS SQL审计日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作。具体操作，请参见云产品日志通用操作。

4.20.3. 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。
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。

4.21. Redis日志

4.21.1. 使用前须知

日志服务和数据库Redis联合推出日志审计功能，将日志服务的部分功能融合到Redis的审计日志、慢日志、运行日志中，实现日志实时查询、可视化分析、告警、投递、加工等功能。本文介绍Redis日志相关的资源说明、费用说明及使用限制。

资源说明

- 专属Project和Logstore

  开通日志审计功能后，系统默认在对应的地域创建一个名为nosql-阿里云账号ID-地域ID的Project，以及名为redis_audit_log和redis_slow_run_log的专属Logstore。

  - redis_audit_log Logstore用于存放Redis审计日志。
  - redis_slow_run_log Logstore用于存放Redis慢日志和运行日志。

- 专属仪表盘

  > **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。您可以自定义创建仪表盘用于查询结果展示。更多信息，请参见创建仪表盘。

  - redis_audit_log Logstore下默认生成一个仪表盘。

仪表盘	说明
Redis审计中心	展示Redis审计日志信息，包括访问用户数、访问客户端数、审计日志条数、平均RT、平均QPS等信息。

- redis_slow_run_log Logstore下默认生成一个仪表盘。

仪表盘	说明
Redis慢日志中心	展示Redis慢日志信息，包括访问用户数、访问客户端数、审计日志条数、平均RT、平均QPS等信息。

费用说明

- 云数据库Redis根据审计日志的存储空间和保存时长按量收取费用，不同地域的收费标准有所区别，更多信息，请参见收费项与价格。

  慢日志和运行日志无写入、存储和查询等费用。
- 推送Redis日志到日志服务后，如果您要执行数据加工、投递、从外网接入点流式读取数据操作，由日志服务收取加工计算费用、数据投递费用和外网读取流量费用。更多信息，请参见日志服务产品定价。

使用限制

- 专属Logstore不支持写入其他数据、修改及删除索引、修改属性等操作，但在查询、统计、告警等功能上无特殊限制。
- 不支持在日志服务侧（控制台、API、SDK等）修改专属Logstore的数据存储时间。您可以在Redis控制台上修改。
- 阿里云日志服务产品需处于可用状态（不欠费），否则全量日志分析功能暂停使用。
- 目前只支持版本号为Redis 4.0及以上，且实例的小版本为最新。版本类型为社区版或企业版（性能增强系列）的Redis实例。

4.21.2. 开通日志审计功能

本文介绍如何在Redis管理控制台上开通日志审计功能，将Redis审计日志推送到日志服务中。

前提条件

已创建满足如下条件的Redis实例。

- Redis实例为社区版或Tair性能增强型。
- Redis实例的引擎版本为4.0或以上，且实例的小版本为最新。关于如何升级版本，请参见升级大版本和升级小版本。

操作步骤

> **说明** 如果您使用RAM用户进行日志功能开通，则需先为RAM用户授权。具体操作，请参见RAM用户授权。

1. 登录Redis管理控制台。
2. 在左侧导航栏中，单击**实例列表**。
3. 在顶部菜单栏中，选择目标Redis实例所在资源组和地域。
4. 在**实例列表**页面，单击目标Redis实例。
5. 在左侧导航栏中，选择**日志管理 > 审计日志**。
6. 选择**正式版**，设置**日志保留时长**，然后单击**费用估算并开通**。
7. 在**Redis审计日志费用预估**对话框，设置**实例平均每秒写入次数**和**实例每次写入平均数据体积**，然后单击**开通**。

后续步骤

日志服务采集到Redis日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作，详情请参见云产品日志通用操作。

4.21.3. 日志字段详情

Redis日志包括审计日志、慢日志和运行日志，本文介绍各类日志的字段详情。

审计日志

审计日志存储在名为redis_audit_log的Logstore中，具体字段说明如下表所示。

字段	说明
__topic__	日志主题 • redis_audit_log：db节点的审计日志 • redis_proxy_audit_log：proxy节点的审计日志
account	数据库账号名称
command	执行的Redis命令
db	数据库名称

字段	说明
extend_information	附加信息
instanceid	Redis实例ID
ip	IP地址
is_cautious	是否为危险操作 • 0：否 • 1：是
latency	延迟
time	时间戳，例如1597048424
type	日志类型

慢日志

慢日志存储在名为redis_slow_run_log的Logstore中，具体字段说明如下表所示。

字段	说明
__topic__	日志主题 • redis_audit_log：db节点的慢日志 • redis_proxy_audit_log：proxy节点的慢日志
account	数据库账号名称
command	执行的Redis命令
db	数据库名称
extend_information	附加信息
instanceid	Redis实例ID
ip	IP地址
is_cautious	是否为危险操作 • 0：否 • 1：是
latency	延迟
time	时间戳，例如1597048424
type	日志类型

运行日志

运行日志存储在名为redis_slow_run_log的Logstore中，具体字段说明如下表所示。

字段	说明
__topic__	日志主题，固定为redis_run_log
extend_information	附加信息
instanceid	Redis实例ID
node_type	运行日志类型 • proxy：proxy节点的运行日志 • db：db节点的运行日志
runlog	运行日志内容

字段	说明
time	时间戳，例如1597048424

4.22. MongoDB日志

4.22.1. 使用前须知

阿里云日志服务联合数据库MongoDB推出日志审计功能，将日志服务的部分功能融合到自身的审计日志、慢日志、运行日志中，实现日志实时查询、可视化分析、告警、投递、加工等功能。本文介绍MongoDB日志相关的资源说明、费用说明及使用限制。

资源说明

- 专属Project和Logstore

  开通日志审计功能后，系统默认在对应的地域创建一个名为nosql-阿里云账号ID-地域ID的Project，以及名为mongo_audit_log和mongo_slow_run_log的专属Logstore。

  - mongo_audit_log Logstore用于存放MongoDB审计日志。
  - mongo_slow_run_log Logstore用于存放MongoDB慢日志和运行日志。

- 专属仪表盘

  > **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。

  mongo_audit_log Logstore下默认生成一个仪表盘。

仪表盘	说明
Mongo审计日志中心	展示MongoDB审计日志信息，包括访问用户数、访问客户端数、平均访问时长、审计日志条数等信息。

费用说明

- 云数据库MongoDB日志审计功能包括免费试用版和正式版，目前只提供免费试用版，不收取费用。
- 推送日志到日志服务后，如果您要执行查询与分析、告警等操作，由日志服务收取索引、告警通知等费用。更多信息，请参见日志服务产品定价。

  > **说明** 目前审计日志推送到日志服务后，不支持拉取、消费、投递和加工操作。

使用限制

- 专属Logstore不支持写入其他数据、修改索引等操作，但在查询、统计、告警等功能上无特殊限制。
- 不支持删除专属Project和Logstore。
- 专属Logstore的数据存储时间为1天，不支持修改。
- 阿里云日志服务产品需处于可用状态（不欠费），否则全量日志分析功能暂停使用。
- 目前只支持版本类型为三节点及以上副本集实例或分片集群实例。

4.22.2. 开通日志审计功能

本文介绍如何在MongoDB控制台上开通日志审计功能，将日志推送到日志服务中。

前提条件

已创建实例类型为三节点及以上副本集实例或分片集群实例。具体操作，请参见创建副本集实例、创建分片集群实例。

操作步骤

1. 登录MongoDB管理控制台。
2. 根据实例类型，在左侧导航栏，单击**副本集实例列表**或**分片集群实例列表**。
3. 在顶部菜单栏中，选择实例所在的资源组和地域。
4. 在实例列表中，单击目标实例。
5. 在左侧导航栏中，选择**数据安全性 > 审计日志**。
6. 首次使用日志审计功能时，单击**开通授权**，完成授权。

完成授权后，系统将生成AliyunServiceRoleForMongoDB角色。云数据库MongoDB将通过该角色，访问当前阿里云账号下日志服务的相关资源。更多信息，请参见MongoDB服务关联角色。

> **注意** 请勿取消授权或删除RAM角色，否则将导致MongoDB审计日志无法正常推送到日志服务。

7. 在**欢迎使用新版审计日志**页面中，设置日志保留时长，然后单击**开通服务**。
8. 在**开通服务**对话框中，单击**确定**。

后续步骤

日志服务采集到MongoDB日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作。更多信息，请参见云产品日志通用操作。

4.22.3. 日志字段详情

MongoDB日志包括审计日志、慢日志和运行日志，本文介绍各类日志的字段详情。

审计日志

审计日志存储在名为mongo_audit_log的Logstore中，具体字段说明如下表所示。

> **说明** 审计日志和慢日志的日志字段相同，通过audit_type字段区分。慢日志中的audit_type字段值固定为*slowop*，如果为其他值，则为审计日志。

字段	说明
__topic__	日志主题，固定为mongo_audit_log
audit_type	日志类型，例如Command
coll	数据集合
db	数据库名称
docs_examined	文档扫描行数
instanceid	MongoDB实例ID
keys_examined	索引扫描行数
latency	消耗时间
optype	操作类别 • query：查询 • find：查询 • insert：插入 • update：更新 • delete：删除 • remove：删除 • getMore：读取 • command：协议命令
return_num	返回记录数
thread_id	线程ID
time	时间戳
user	登录MongoDB数据库的用户名
user_ip	连接MongoDB客户端的IP地址

慢日志

慢日志存储在名为mongo_slow_run_log的Logstore中，具体字段说明如下表所示。

字段	说明
__topic__	日志主题，固定为mongo_run_log

字段	说明
audit_type	日志类型，固定为slowop
coll	数据集合
db	数据库名称
docs_examined	文档扫描行数
instanceid	MongoDB实例ID
keys_examined	索引扫描行数
latency	消耗时间
optype	操作类别 • query：查询 • find：查询 • insert：插入 • update：更新 • delete：删除 • remove：删除 • getMore：读取 • command：协议命令
return_num	返回记录数
thread_id	线程ID
time	时间戳
user	登录MongoDB数据库的用户名
user_ip	连接MongoDB客户端的IP地址

运行日志

运行日志存储在名为mongo_slow_run_log的Logstore中，具体字段说明如下表所示。

字段	说明
__topic__	日志主题，固定为mongo_run_log
category	日志类别，例如NETWORK（网络链接日志）
connection	日志连接信息
content	日志内容
instanceid	MongoDB实例ID
ip	IP地址
level	日志级别
port	端口号
time	日志生成时间

4.23. IoT日志

4.23.1. 使用前须知

阿里云日志服务和物联网平台联合推出日志功能，将云端运行日志推送到日志服务中。日志服务提供实时查询、可视化分析、告警、投递、加工等功能。本文介绍IoT云端运行日志相关的资产详情、费用说明、使用限制等信息。

资源说明

- 专属Project和Logstore

  开通日志功能后，系统默认在对应的地域创建一个名为iot-log-阿里云账号ID-地域ID的Project，以及名为iot_logs的专属Logstore。

- 专属仪表盘

  默认生成1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，详情请参见创建仪表盘。

仪表盘	说明
IoT运营中心	展示设备运行情况和异常问题，包括设备上下线次数、设备上线IP按区域分布、数据解析脚本错误码分布、物模型校验错误码分布、服务端订阅流转消息量、云产品流转消息量、云端API调用错误量分布等信息。

费用说明

- 目前，物联网平台不针对日志转储功能收取额外费用。
- 将日志投递到日志服务后，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，详情请参见日志服务产品定价。

使用限制

- 专属Logstore不支持写入其他数据、修改索引等操作，在查询分析、告警、消费等功能上无特殊限制。
- 不支持删除专属Project和Logstore。
- 一个地域仅支持创建一个日志配置，该地域所有的产品的云端运行日志均通过该配置推送到对应的日志服务Logstore中。
- 目前仅支持华东2（上海）、华南1（深圳）、新加坡、日本（东京）、德国（法兰克福）、美国（硅谷）和美国（弗吉尼亚）地域。

4.23.2. 开通日志转储功能

本文介绍如何在物联网平台控制台上开启日志转储功能，将云端运行日志推送到日志服务。

前提条件

- 已开通日志服务。
- 已创建产品并接入设备。具体操作，请参见设备接入。

操作步骤

1. 登录物联网平台控制台。

   > **说明** 目前仅支持阿里云账号使用日志转储功能。

2. 在**实例概览**页面，单击目标实例。
3. 在左侧导航栏，选择**监控运维 > 日志服务**。
4. 选择目标产品。
5. 在**云端运行日志转储**页签下，单击**立即开启**。
6. 在**日志配置**对话框中，单击**确定**。

   开通日志转储功能后，系统自动创建一个服务关联角色AliyunServiceRoleForIoTLogExport。物联网平台使用此角色来访问您在日志服务云产品中的资源。

   > **说明**
   > - 如果AliyunServiceRoleForIoTLogExport角色已存在，不会重复创建。
   > - 请勿取消授权或删除RAM角色，否则将导致云端运行日志无法正常推送到日志服务。

后续步骤

- 将云端运行日志转储到日志服务后，您可以执行查询分析、下载、投递、加工、创建告警等操作，详情请参见云产品日志通用操作。
- 如果您不再需要转储日志到日志服务，可单击**停止转储**。

  停止转储后，新产生的日志不再导出到日志服务Logstore中。已导出的日志达到设置的日志存储时间后被自动清空。

4.23.3. 日志字段详情

本文介绍IoT日志字段详情。

字段名	说明
__topic__	日志主题，固定为iot_log
bizCode	业务类型
code	结果状态码 • 200表示成功。 • 其他结果码表示失败，详情请参见云端运行日志。
content	日志内容
deviceName	设备名称
instanceId	实例ID
messageId	消息ID

operation	操作
	• 固件升级相关操作
	  ○ OTAFirmwarePush：固件推送，包括发起时推送、确认时推送、上线时推送。
	  ○ OTAVersionReport：设备上报固件版本。
	  ○ OTAProgressReport：设备上报升级进度。
	• 数据解析相关操作
	  ○ RawDataToProtocol：原始数据转换为Alink协议数据。
	  ○ ProtocolToRawData：Alink协议数据转换为原始数据。
	• 物模型数据上报相关操作
	  ○ *check*：检查物模型。
	  ○ 消息体中的method，请参见什么是物模型。
	• 设备行为相关操作
	  ○ online：设备上线。
	  ○ offline：设备下线。

productKey	产品Key 所有产品共用Logstore，不同产品通过ProductKey字段区分。
reason	错误原因
requestId	请求ID
status	请求结果 • true表示成功。 • false表示失败。
utcTime	收集时间，UTC格式
traceId	追踪ID
clientId	设备端上报的clientId
params	传入参数
resultData	结果

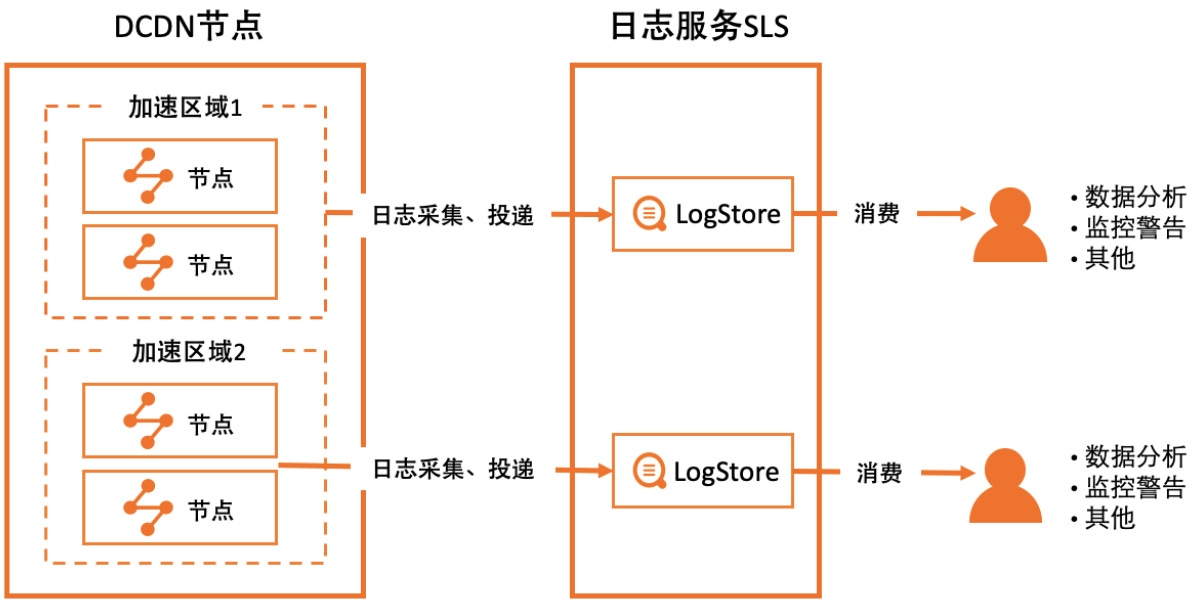
4.24. DCDN实时日志

4.24.1. 使用前须知

阿里云日志服务与全站加速DCDN联合推出日志功能，支持近实时采集DCDN实时日志。日志服务提供实时查询、可视化分析、告警、投递、加工等一站式服务。本文介绍DCDN实时日志相关的功能介绍、资产详情、费用说明及使用限制。

功能介绍

您的业务在使用DCDN服务时，会产生大量的网络日志数据。通过实时日志功能，您可以实时采集节点产生的日志，并投递到日志服务进行存储和消费，以便快速监控和定位业务问题。



资产详情

- 专属Project和Logstore

  开启日志投递功能后，日志服务自动在对应的地域创建一个名为dcdn-edge-rtlog-地域-随机ID的Project，以及一个名为dcdn-edge-rtlog的Logstore。

  > **说明** 请勿删除DCDN实时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

  > **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，详情请参见创建仪表盘。

仪表盘	说明
DCDN访问中心	实时监测DCDN业务流量变化，包括PV总数、UV总数、错误请求百分比、Top10 URI、Top10 IP地址、错误码趋势、错误域名 Top10等信息。

费用说明

- 针对实时日志功能，DCDN收取实时日志采集费用。更多信息，请参见增值服务计费-实时日志条数。
- DCDN将日志投递到日志服务后，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

- 同一个域名，可以绑定多个海外采集地域，暂不支持同时绑定相同的采集地域或同时绑定中国内地+海外地域。例如example.com不能绑定**美国和中国内地**或都绑定**中国内地**，但可以同时绑定**美国和印度**。
- 目前，一个日志服务投递区域只能绑定一个Project。

功能优势

- 时效性高：开通服务后，日志数据自动投递到日志服务，可查看实时日志分析结果，日志时效性在3分钟以内。
- 字段丰富：实时日志提供更加丰富的日志字段。更多信息，请参见日志字段详情。
- 业务合规：实时日志采集投递具备合规性。

- 强大数据分析能力：DCDN与SLS进行产品联动，DCDN投递到SLS中的日志会默认生成可视化分析模版，帮助您更好的分析与监控您的业务，示例场景：
  - 精细化的URL分析：分析某个域名的访问趋势、人群画像（地理位置、访问来源等）、可用性等。
  - 业务可靠性分析：访问的错误码信息、错误域名信息、错误URL信息、错误IP信息等。
  - 网络质量分析：响应时延、缓存命中率等。
  - 人群画像分析：地理位置分布、终端信息分析、访问来源信息等。

4.24.2. 开启实时日志投递

通过实时日志投递功能，您可以实时采集DCDN加速域名产生的日志，并投递到日志服务进行分析，以便快速监控和定位业务问题。本文主要介绍如何开启实时日志投递功能。

前提条件

- 已添加加速域名。具体操作，请参见添加加速域名。
- 已开通日志服务。

操作步骤

1. 登录全站加速控制台。
2. 在左侧导航栏，选择**数据中心 > 日志管理 > 实时日志**。
3. 开通实时日志投递功能。

   您在首次使用实时日志投递功能时，DCDN控制台将提示您开通实时日志投递功能。请根据页面提示完成操作。
4. 在**实时日志**页面，完成如下配置，开启实时日志投递。
   i. 在**实时日志**页面，单击**创建投递项目**。
   ii. 在**创建实时日志项目**对话框中，配置项目信息，然后单击**下一步**。

参数	说明
项目标识	用于标识实时日志项目，需保证唯一性。
日志类型	实时日志支持投递的日志类型，包括： ■ 用户访问日志。 ■ 用户回源日志。
采集字段	每个实时日志中包含的字段列表，字段名称和说明请参见日志字段详情。
采样率	接收实时日志投递请求的百分比，取值：0～100%。 **说明** 投递的日志条数约为全量日志数量×采样率。如果需要接收所有日志投递请求，请将采样率设为100%。

iii. 在**创建实时日志项目**对话框中，选择采集地域和投递地域，然后单击**下一步**。

> **注意** 创建实时日志项目后，不可修改**采集区域**和**SLS投递区域**。您可以删除已经创建的实时日志项目，然后重新创建。

参数	说明
采集区域	采集地域。
SLS投递区域	日志服务Project所在地域。日志服务将在该地域创建Project，用于管理实时日志。
授权	首次创建时，需配置授权，授予DCDN使用服务关联角色AliyunServiceRoleForDCDNRealTimeLogDelivery访问日志服务SLS的资源。更多信息，请参见实时日志服务关联角色。

DCDN实时日志被采集后将投递到日志服务的指定地域进行存储，对应表如下：

（DCDN）采集区域	SLS投递区域
中国内地	■ 华东1（杭州） ■ 华东2（上海） ■ 华北1（青岛） ■ 华北2（北京） ■ 华北3（张家口） ■ 华南1（深圳）
欧洲	德国（法兰克福）
美国	美国（硅谷）
印度	印度（孟买）
其他	新加坡

iv. 在**创建实时日志项目**对话框中，选择投递域名，然后单击**下一步**。

后续步骤

日志服务采集到DCDN实时日志后，您可以执行查询分析、下载、投递、加工、创建告警等操作。更多信息，请参见云产品日志通用操作。

4.24.3. 日志字段详情

本文介绍DCDN实时日志的字段详情。

字段名称	说明
unixtime	请求时间。
domain	请求域名。
method	请求方法。
scheme	请求协议。
uri	请求资源。
uri_param	请求参数。
client_ip	最终用户的真实IP地址，可以是公网IP地址或局域网IP地址。
proxy_ip	代理IP地址。
remote_ip	与DCDN节点直接连接的公网IP地址。
remote_port	和DCDN节点直接连接公网端口。
refer_protocol	HTTP Referer中的协议信息。
refer_domain	HTTP Referer中的域名信息。

字段名称	说明
refer_uri	HTTP Referer中的URI信息。
refer_param	HTTP Referer中的参数信息。
request_size	请求大小（包体+头部）。
request_time	请求响应时间，单位：毫秒。
response_size	请求返回大小，单位：字节。
return_code	请求响应码。
sent_http_content_range	应答头里表示的Range信息（由源站创建），例如bytes：0~99/200。
server_addr	服务的DCDN节点IP地址。
server_port	服务的DCDN节点服务端口。
body_bytes_sent	实际发送的Body大小，单位：字节。
content_type	请求的资源类型。
hit_info	是否命中信息（直播、动态加速除外），取值： • *HIT*：命中。 • *MISS*：未命中。
http_range	用户请求中Header头Range字段取值，例如bytes：0~100。
user_agent	用户代理信息。
user_info	用户信息。
uuid	请求唯一标识。
via_info	Via头信息。
xforwordfor	请求头中的X-Forwarded-For字段。

5.数据导入

5.1. 导入OSS数据

您可以将日志文件上传到OSS中，并通过数据导入方式将OSS数据导入到日志服务，实现日志数据的查询分析、加工等操作。目前日志服务只支持导入5 GB以内的OSS文件，压缩文件大小按照压缩后的大小计算。

前提条件

- 已上传日志文件到OSS Bucket中。具体操作，请参见上传文件。
- 已创建Project和Logstore。具体操作，请参见创建Project和创建Logstore。
- 已经完成云资源访问授权，即已授权日志服务使用AliyunLogImportOSSRole角色访问您的OSS资源。

  如果您使用的是RAM用户，还需授予RAM用户PassRole权限，授权策略如下所示。具体操作，请参见创建自定义权限策略和为RAM用户授权。

```
{
   "Statement": [
    {
      "Effect": "Allow",
      "Action": "ram:PassRole",
      "Resource": "acs:ram:*:*:role/aliyunlogimportossrole"
    }
  ],
  "Version": "1"
}
```

创建数据导入配置

> **说明** OSS Normal类型的文件支持文件粒度的标记，该类型文件的内容被修改后将全部重新导入一次。OSS Appendable类型的文件支持行级粒度标记，可追加内容的增量读取。

1. 登录日志服务控制台。
2. 在**接入数据**区域的**数据导入**页签中，选择**OSS-数据导入**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 设置导入配置。

i. 在**数据源设置**页签中，配置如下参数。

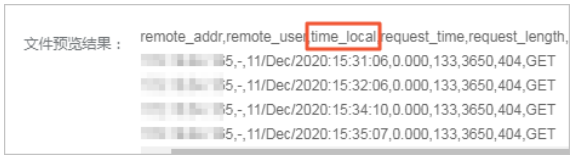
参数	说明
配置名称	设置配置的名称。
OSS Region	待导入的OSS文件所在Bucket的地域。 如果OSS Bucket和日志服务Project处于同一地域，可节省公网流量且传输速度快。
Bucket	待导入的OSS文件所在的Bucket。
文件夹前缀	待导入的OSS文件所在文件夹的前缀，用于准确定位待导入的文件夹。例如待导入的文件都在*csv/*目录下，则可以指定前缀为*csv/*。 如果不配置文件夹前缀，则遍历整个OSS Bucket。 **说明** 不支持导入冷归档的OSS文件。如果是冷归档文件，可以通过OSS相关接口解冻后，再导入。更多信息，请参见冷归档存储（Cold Archive）。
正则过滤	用于过滤文件的正则表达式，只有文件名（包含文件路径）匹配该正则表达式的文件才会被导入。默认为空，表示不过滤。 例如OSS文件为*testdata/csv/bill.csv*，您可以配置正则表示式为**(testdata/csv/)(.*)**。
数据格式	文件的解析格式，如下所示。 ■ CSV：分隔符分割的文本文件，支持指定文件中的首行为字段名称或手动指定字段名称。除字段名称外的每一行都会被解析为日志字段的值。 ■ 单行JSON：逐行读取OSS文件，将每一行看做一个JSON对象进行解析。解析后，JSON对象中的各个字段对应为日志中的各个字段。 ■ Parquet：Parquet格式，无需任何配置，自动解析成日志格式。不支持预览。 ■ 单行文本日志：将OSS文件中的每一行解析为一条日志。 ■ 跨行文本日志：多行模式，支持指定首行或者尾行的正则表达式解析日志。
压缩格式	待导入的OSS文件的压缩格式，日志服务根据对应格式进行解压并读取数据。
编码格式	待导入的OSS文件的编码格式。
解冻归档文件	如果待导入的OSS文件为归档存储，则需要解冻后才能读取。开启此功能，则归档文件会自动解冻。

ii. 单击**预览**，查看文件预览结果。

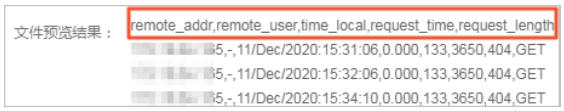
iii. 确认无误后，单击**下个配置**。

iv. 在**数据格式配置**页签中，完成如下配置。

- 日志时间相关参数说明

参数	说明
使用系统时间	配置日志时间，具体说明如下： ■ 打开**使用系统时间**开关，则解析后的日志时间显示为文件导入时的系统时间。 ■ 关闭**使用系统时间**开关，则需要手动配置时间字段和时间格式。 **说明** 推荐使用系统时间。因为日志时间可作为普通字段建立索引，用于日志查询，所以在导入历史数据时，如果数据时间早于当前时间减去Logstore数据保存时间，例如保存7天，那么时间为7天前的日志，无法在控制台上查询。
提取时间正则	如果您选择的数据格式为单行文本日志、跨行文本日志，则在关闭**使用系统时间**开关后，您需要使用正则表示式提取日志时间。 例如日志文件中的日志样例为**127.0.0.1 - - [10/Sep/2018:12:36:49 0800] "GET /index.html HTTP/1.1"**，则您可以配置**提取时间正则**为**[0-9]{0,2}\/[0-9a-zA-Z]+\/[0-9:,]+**。
时间字段	如果您选择的数据格式为CSV、单行JSON、Parquet，则在关闭**使用系统时间**开关后，您需要配置一个时间字段。 例如CSV格式文件如下所示，则您可以配置**时间字段**为**time_local**。 
时间格式	如果关闭**使用系统时间**开关，需要指定一个Java SimpleDateFormat语法的时间格式，用于解析时间字段。 时间格式的语法详情请参见Class SimpleDateFormat。常见的时间格式请参见时间格式。 **说明** Java SimpleDateFormat不支持Unix时间戳，如果您要使用Unix时间戳，时间格式指定为epoch。
时区	如果关闭**使用系统时间**开关，需要指定一个时区，用于解析日志时间的时区。如果提取的日志时间中已有时区信息，则此参数无效。

- 其他参数说明
  - CSV特有参数

参数	说明
分隔符	配置日志的分隔符，默认值为半角逗号（,）。
Quote	当日志字段内包含分隔符时，需要使用引用符包裹，默认值为双引号（"）。
转义符	配置日志的转义符，默认值为反斜线（\）。
跨行日志最大行数	当一条日志跨多行时，需要指定最大行数，默认值为1。
首行作为字段名称	打开**首行作为字段名称**开关后，将使用CSV文件中的首行作为字段名称。例如提取下图中的首行为日志字段的名称。 
自定义字段列表	关闭**首行作为字段名称**开关后，请根据需求自定义日志字段名称，多个字段名称之间用半角逗号（,）隔开。
跳过行数	指定跳过的日志行数。例如配置为1，则表示从CSV文件中的第2行开始采集日志。

  - 跨行文本日志特有参数

参数	说明
正则匹配位置	配置正则表示式匹配的位置，具体说明如下： ■ 选择**首行正则**，则使用正则表示式去匹配一条日志的行首，未匹配部分为该条日志的一部分，直到达到最大行数。 ■ 选择**尾行正则**，则使用正则表示式去匹配一条日志的行尾，未匹配部分为下一条日志的一部分，直到达到最大行数。
正则表达式	根据日志内容，配置正确的正则表达式。更多信息，请参见如何调试正则表达式。
最大行数	一条日志最大的行数。

v. 设置数据格式完成后，单击**测试**。

vi. 测试成功后，单击**下个配置**。

vii. 在**调度间隔**页签中，配置如下参数。

参数	说明
导入间隔	导入OSS文件到日志服务的时间间隔。
立即执行	开启**立即执行**，则立即执行一次导入操作。

viii. 配置完成后，单击**下一步**。

5. 预览数据及设置索引，然后单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

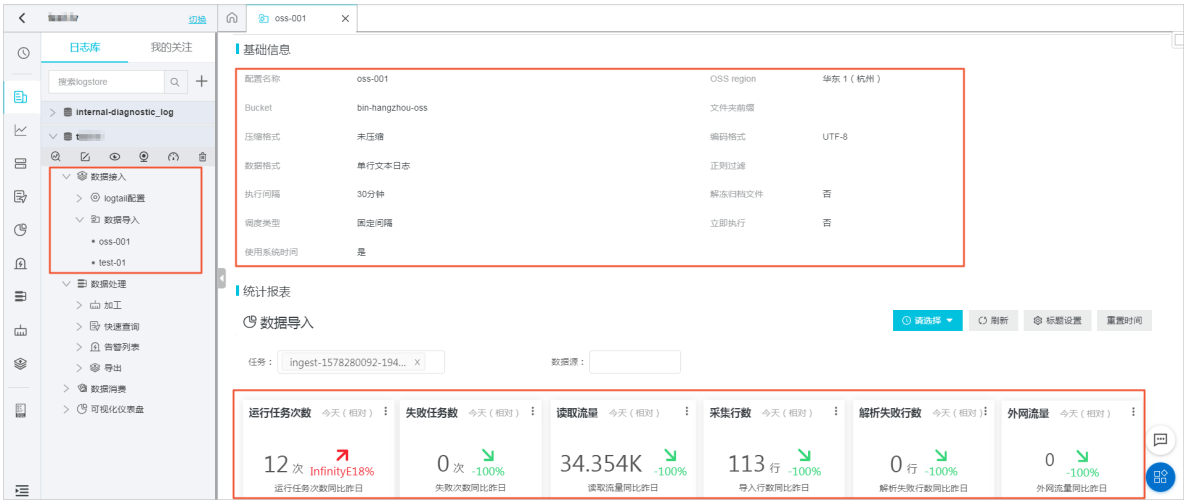
   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

6. 单击**下一步**，完成导入配置。

查看导入配置

创建导入配置成功后，您可以在控制台中查看已创建的导入配置及生成的统计报表。

1. 在**Project**列表区域，单击目标Project。
2. 在**日志存储 > 日志库**中，选择目标日志库下的**数据接入 > 数据导入**，单击配置名称。
3. 在**导入配置概览**页面，查看导入配置的基本信息和统计报表。



相关操作

在配置的**导入配置概览**页面，您还可以进行如下操作。

- 修改配置

  单击**修改配置**，修改导入配置的相关信息，详情请参见设置导入配置。

- 删除配置

  单击**删除配置**，删除该导入配置。

  > **警告** 删除后不可恢复，请谨慎操作。

常见问题

问题	可能原因	解决方法
创建导入配置时无法选择OSS Bucket。	未配置AliyunLogImportOSSRole角色授权。	参见文本中的前提条件完成授权。
无法导入数据。	文件大小超出5 GB。	缩小单个OSS文件的大小。
导入数据后，无法查询分析数据。	未配置索引或者索引未生效。	建议您提前为Logstore建立索引，避免导入数据后索引未生效问题。具体操作，请参见配置索引。如果问题已经发生，您可以尝试重建索引。具体操作，请参见重建索引。
无法导入归档文件失败。	未开启解冻归档功能。	• 方法1：修改导入配置，打开**解冻归档文件**开关。 • 方法2：重建1个导入配置，打开**解冻归档文件**开关。
配置**正则过滤**参数后，未采集到目标数据。	• 正则表示式配置错误。 • 文件数量太多，在超时时间内未遍历到匹配的文件。	重新配置**正则过滤**参数。如果仍未采集到目标数据，可能是因为文件数量太多，请设置较精准的目录缩小遍历的文件数目。
导入日志成功，但在日志服务控制台上查不到数据。	日志时间超出Logstore数据保存时间，过期数据已被删除。	检查查询时间范围和Logstore数据保存时间。
使用提取的日志时间查询数据，却在该时间未查询到数据。	时间格式配置错误。	检查时间格式为Java SimpleDateFormat标准格式。更多信息，请参见Class SimpleDateFormat。
多行文本日志解析错误。	首行正则表达式或尾行正则表达式配置错误。	检查首行正则或尾行正则的正确性。
导入速度突然变慢。	• 没有足量未导入的数据。 • OSS文件数量太多，遍历OSS文件占用处理时间。	1. 确认是否有足量未导入的数据。 2. 确认是否是OSS文件数量太多，遍历OSS文件占用处理时间。如果是该原因，您可以使用**文件夹前缀**、**正则过滤**减少单个导入任务所匹配的文件数量或者在OSS中迁移已导入的文件到其他目录或Bucket。

5.2. 导入MaxCompute数据

您可以将日志文件保存到MaxCompute中，并通过数据导入方式将MaxCompute数据导入到日志服务，实现日志数据的查询分析、数据加工等操作。

前提条件

- 已开通MaxCompute并上传日志文件数据。具体操作，请参见导入数据。
- 已创建Project和Logstore。具体操作，请参见创建Project和创建Logstore。

创建数据导入配置

1.
2. 在**接入数据**区域的**数据导入**页签中，单击**MaxCompute-数据导入**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 设置导入配置。
   i. 在**数据源**设置中，配置如下参数。

参数	说明
配置名称	设置配置的名称。
MaxCompute项目	待导入数据所在的MaxCompute项目名称。
Table	待导入数据所在的表名称。
分区描述	MaxCompute的表分区描述。更多信息，请参见分区。 如果存在多级分区，使用半角逗号（,）分隔。
AccessKey ID	用于访问MaxCompute的AccessKey ID。 如何获取，请参见访问密钥。 **说明** 请确保您的AccessKey具有访问对应MaxCompute项目的权限。
AccessKey Secret	用于访问MaxCompute的AccessKey Secret。
Endpoint	用于访问MaxCompute的Endpoint。更多信息，请参见Endpoint。
Tunnel Endpoint	用于访问MaxCompute的Tunnel Endpoint。更多信息，请参见Endpoint。

   ii. 单击**预览**，查看文件预览结果。
   iii. 确认无误后，单击**下个配置**。

iv. 在**数据格式配置**页签中，配置如下参数。

参数	说明
使用系统时间	■ 开启**使用系统时间**，则解析后的日志时间显示为导入时的系统时间。 ■ 关闭**使用系统时间**，则需要手动配置时间字段和时间格式。 **说明** 推荐开启**使用系统时间**，日志时间可作为普通字段建立索引，用于日志查询。在导入历史数据时，如果数据时间早于当前时间减去Logstore数据保存时间，例如保存7天，那么时间为7天前的日志无法在控制台上查询。
时间字段	如果关闭**使用系统时间**，则需要指定一个用于提取日志时间的字段。
时间格式	如果关闭**使用系统时间**，需要指定一个Java SimpleDateFormat语法的时间格式，用于解析时间字段或者使用正则表达式提取到的字符串。时间格式的语法详情请参见Class SimpleDateFormat。常见的时间格式请参见时间格式。 **说明** Java SimpleDateFormat不支持Unix时间戳，如果您要使用Unix时间戳，时间格式指定为epoch。
时区	如果关闭**使用系统时间**，需要指定一个时区，用于解析日志时间的时区。如果日志格式中已经有时区信息，则此参数无效。

v. （可选）设置数据格式完成后，单击**测试**。

vi. 测试成功后，单击**下个配置**。

vii. （可选）在**调度间隔**页签中，配置如下参数。

参数	说明
导入间隔	MaxCompute数据导入日志服务的时间间隔。
立即执行	开启**立即执行**，则立即执行一次导入操作。

viii. 配置完成后，单击**下一步**。

5. 预览数据及设置索引，然后单击**下一步**。

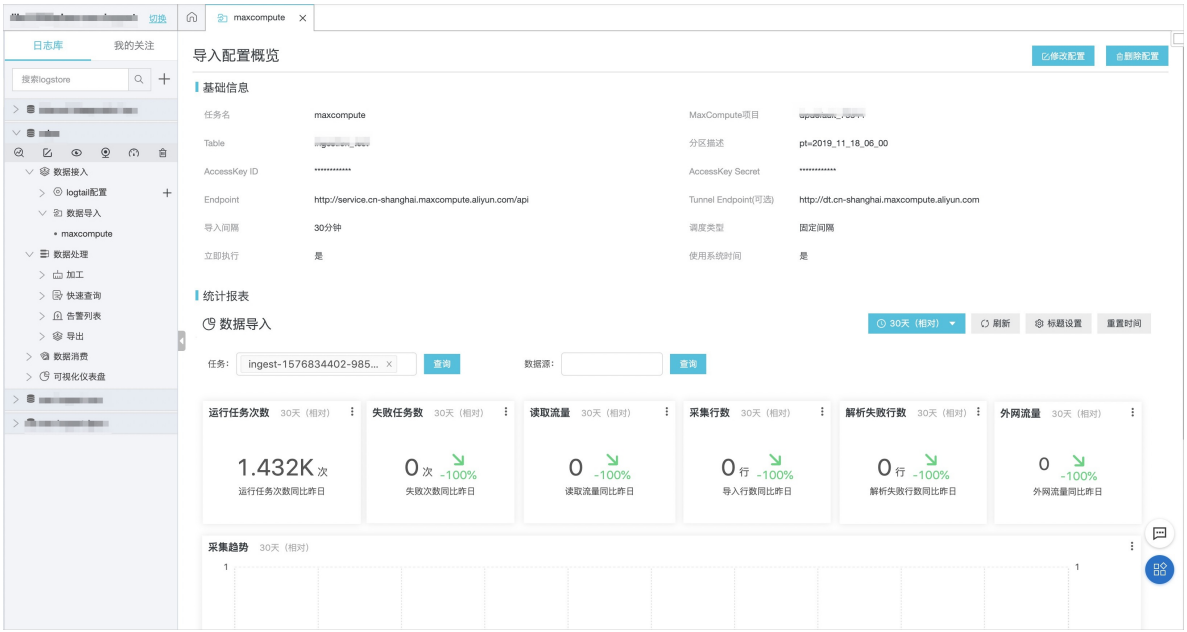
   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   说明 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

查看导入配置

创建导入配置成功后，您可以在控制台中查看已创建的导入配置及生成的统计报表。

1. 单击目标Project。
2. 选择目标日志库下的**数据接入 > 数据导入**，单击配置名称。
3. 在**导入配置概览**页面，查看导入配置的基本信息和统计报表。



相关操作

在**导入配置概览**页面，您还可以进行如下操作。

- 修改配置

  单击**修改配置**，修改导入配置的相关配置，具体配置请参见设置导入配置。

- 删除配置

  单击**删除配置**，删除该导入配置。

> **说明** 删除后不可恢复，请谨慎操作。

5.3. 时间格式

在创建导入任务时，需设置对应时间字段的格式。本文介绍时间格式的语法和示例。

时间格式语法

字符	说明	示例
G	纪元标记	AD
y	年份	2001
M	月份	July、07
d	日期	10
h	小时，取值：1~12（AM、PM）	12
H	一天中的小时，取值：0~23	22
m	分钟	30
s	秒	55
S	毫秒	234
E	星期	Tuesday
D	一年中的第几天	360
F	一个月中的周几	2
w	一年中的第几周	40

字符	说明	示例
W	一个月中的第几周	1
a	AM、PM	PM
k	一天中的小时，取值：1~24	24
k	小时，取值：0~11（AM、PM）	10
z	时区	Eastern Standard Time
'	文字定界符	Delimiter
''	单引号	'

时间格式样例

日期格式	解析语法	解析后的值（单位：秒）
2020-05-02 17:30:30	yyyy-MM-dd HH:mm:ss	1588411830
2020-05-02 17:30:30:123	yyyy-MM-dd HH:mm:ss:SSS	1588411830
2020-05-02 17:30	yyyy-MM-dd HH:mm	1588411800
2020-05-02 17	yyyy-MM-dd HH	1588410000
20-05-02 17:30:30	yy-MM-dd HH:mm:ss	1588411830
2020-05-02T17:30:30V	yyyy-MM-dd'T'HH:mm:ss'V'	1588411830
Sat May 02 17:30:30 CST 2020	EEE MMM dd HH:mm:ss zzz yyyy	1588411830

6.其他采集方式

6.1. 使用Web Tracking采集日志

日志服务支持通过Web Tracking采集HTML、H5、iOS和Android平台的日志，并支持自定义维度和指标。本文介绍如何使用Web Tracking采集日志。

背景信息

您可以通过Web Tracking采集各种浏览器、iOS App或Android App的用户信息，例如：

- 用户使用的浏览器、操作系统、分辨率等信息。
- 用户浏览行为记录（例如：用户在网站上的单击行为、购买行为等信息）。
- 用户在App中的停留时间、是否活跃等信息。

注意事项

- 使用Web Tracking则表示该Logstore打开互联网匿名写入权限，没有经过有效鉴权，可能产生脏数据。
- GET请求不支持上传16 KB以上的Body内容。
- POST请求每次写入的日志数量上限为10 MB。更多信息，请参见PutLogs。

步骤一：开通Web Tracking

您可以通过控制台方式或者SDK方式开通Logstore的Web Tracking功能。

- 控制台方式
  i. 登录日志服务控制台。
  ii. 在Project列表区域，单击目标Project。
  iii. 在**日志存储 > 日志库**页签中，选择目标Logstore右侧的 > **修改**。
  iv. 在**Logstore属性**页面，单击右上方的**修改**。
  v. 打开**WebTracking**开关，并单击**保存**。
- SDK方式

  通过日志服务Java SDK方式开通Web Tracking。 更多信息，请参见Java SDK快速入门。

```
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.LogStore;
import com.aliyun.openservices.log.exception.LogException;
public class WebTracking {
  static private String accessId = "your accesskey id";
  static private String accessKey = "your accesskey";
  static private String project = "your project";
  static private String host = "log service data address";
  static private String logStore = "your logstore";
  static private Client client = new Client(host, accessId, accessKey);
  public static void main(String[] args) {
      try {
          //在已经创建的Logstore上开通Web Tracking功能。
          LogStore logSt = client.GetLogStore(project, logStore).GetLogStore();
          client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), true));
          //关闭Web Tracking功能。
          //client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.GetShardCount(), false));
          //新建支持Web Tracking功能的Logstore。
          //client.UpdateLogStore(project, new LogStore(logStore, 1, 1, true));
      }
      catch (LogException e){
          e.printStackTrace();
      }
  }
}
```

步骤二：采集日志

开通Web Tracking后，您可以通过以下方法上传日志到Logstore中。

- 通过浏览器JavaScript SDK上传日志。具体操作，请参见浏览器JavaScript SDK。
- 通过小程序JavaScript SDK上传日志。具体操作，请参见小程序JavaScript SDK。

- 通过HTTP GET请求上传日志。参见如下命令上传日志，请根据实际值替换参数。

```
curl --request GET 'http://${project}.${host}/logstores/${logstore}/track?APIVersion=0.6.0&key1=val1&key2=val2'
```

参数	是否必填	说明
${project}	是	Project名称。
${host}	是	日志服务所在地域的Endpoint。更多信息，请参见服务入口。
${logstore}	是	Logstore名称。
APIVersion=0.6.0	是	保留字段。
__topic__=yourtopic	否	指定日志主题。
key1=val1&key2=val2	是	您要上传到日志服务的键值对（Key-Value），可以有多个，请确保长度小于16 KB。

- 通过HTML img标签上传日志。

```
<img src='http://${project}.${host}/logstores/${logstore}/track.gif?APIVersion=0.6.0&key1=val1&key2=val2'/>
<img src='http://${project}.${host}/logstores/${logstore}/track_ua.gif?APIVersion=0.6.0&key1=val1&key2=val2'/>
```

track_ua.gif除了上传自定义的参数外，还会将HTTP头中的UserAgent、referer也作为日志中的字段。

> **说明** 如果您需要采集HTTPS页面的referer，那么上述Web Tracking的链接也必须为HTTPS。

- 通过HTTP POST请求上传日志。

  如果请求的数据量比较大，可以使用POST方法上传数据。更多信息，请参见PutWebtracking。

6.2. 使用Kafka协议上传日志

您可以使用各类Kafka Producer SDK或采集工具来采集日志，并通过Kafka协议上传到日志服务。本文介绍通过Kafka协议将日志上传到日志服务的操作步骤。

相关限制

- 支持的Kafka协议版本为：0.8.0到2.1.1V2。
- 为保证日志传输安全性，必须使用SASL_SSL连接协议。
- 如果您的Logstore有多个Shard，请使用负载均衡模式上传日志。
- 目前只支持将Kafka Producer SDK或采集Agent采集到的日志使用Kafka协议上传日志到日志服务。

配置方式

使用Kafka协议上传日志时，您需要配置以下参数。

参数	说明
连接类型	为保证日志传输安全性，连接协议必须为SASL_SSL。
hosts	初始连接的集群地址，格式为 *project名称*.Endpoint，请根据Project所在的Endpoint进行配置。更多信息，请参见服务入口。 • 阿里云内网：端口号为10011，例如test-project-1.cn-hangzhou-intranet.log.aliyuncs.com:10011。 • 公网：端口号为10012，例如test-project-1.cn-hangzhou.log.aliyuncs.com:10012。
topic	配置为日志服务Logstore名称。
username	配置为日志服务Project名称。
password	配置为阿里云AK，格式为${access-key-id}#${access-key-secret}。请根据实际情况，将${access-key-id}替换为您的AccessKey ID，将${access-key-secret}替换为您的AccessKey Secret。建议使用RAM用户的AK。更多信息，请参见授权。
证书文件	日志服务的域名均具备可信任证书，您只需使用服务器自带的根证书即可，例如：*/etc/ssl/certs/ca-bundle.crt*。

示例一：通过Beats系列软件采集日志到日志服务

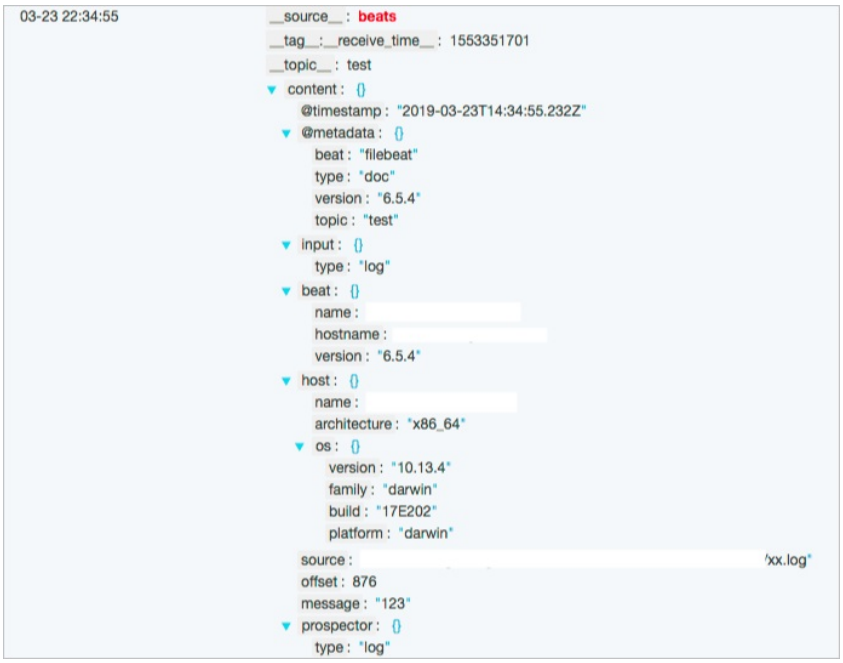
Beats系列软件（MetricBeat、PacketBeat、Winlogbeat、Auditbeat、Filebeat、Heartbeat等）采集到日志后，支持通过Kafka协议将日志上传到日志服务。更多信息，请参见Beats-Kafka-Output。

- 配置示例

```
output.kafka:
  # initial brokers for reading cluster metadata
  hosts: ["test-project-1.cn-hangzhou.log.aliyuncs.com:10012"]
  username: "yourusername"
  password: "yourpassword"
  ssl.certificate_authorities:
  # message topic selection + partitioning
  topic: 'test-logstore-1'
  partition.round_robin:
    reachable_only: false
  required_acks: 1
  compression: gzip
  max_message_bytes: 1000000
```

- 日志样例

  Beats系列软件默认输出的日志为JSON类型，您可以给content字段创建JSON类型的索引。更多信息，请参见JSON类型。



示例二：通过Collectd采集日志到日志服务

Collectd是一个守护（daemon）进程，用于定期采集系统和应用程序的性能指标，并支持通过Kafka协议上传到日志服务。更多信息，请参见Write Kafka Plugin。

将Collectd采集到日志上传到日志服务时，还需安装Kafka插件以及相关依赖。例如：在linux Centos中，可以使用yum安装Kafka插件，命令为 `sudo yum install collectd-write_kafka` ，安装RPM请参见Collectd-write_kafka。

- 配置示例

  示例中将日志输出格式（Format）设置为JSON，除此之外，还支持Command、Graphite类型。更多信息，请参见Collectd配置文档。

```
<Plugin write_kafka>
  Property "metadata.broker.list" "test-project-1.cn-hangzhou.log.aliyuncs.com:10012"
  Property "security.protocol" "sasl_ssl"
  Property "sasl.mechanism" "PLAIN"
  Property "sasl.username" "yourusername"
  Property "sasl.password" "yourpassword"
  Property "broker.address.family" "v4"
  <Topic "test-logstore-1">
    Format JSON
    Key "content"
  </Topic>
</Plugin>
```

- 日志样例

  使用JSON模式输出日志后，您可以给content字段创建JSON类型的索引。更多信息，请参见JSON类型。

示例三：使用Telegraf采集日志到日志服务

Telegraf是由Go语言编写的代理程序，内存占用小，用于收集、处理、汇总数据指标。Telegraf具有丰富的插件及具备集成功能，可从其运行的系统中获取各种指标、从第三方API中获取指标以及通过statsd和Kafka消费者服务监听指标。

将Telegraf采集到的日志通过kafka协议上传到日志服务前，您需要先修改配置文件。

- 配置示例

  示例中将日志输出格式（Format）设置为JSON，除此之外还支持Graphite、Carbon2等类型。更多信息，请参见Telegraf输出格式。

> **说明** Telegraf必须配置一个合法的*tls_ca*路径，使用服务器自带的根证书的路径即可。Linux环境中，根证书CA路径一般为*/etc/ssl/certs/ca-bundle.crt*。

```
[[outputs.kafka]]
  ## URLs of kafka brokers
  brokers = ["test-project-1.cn-hangzhou.log.aliyuncs.com:10012"]
  ## Kafka topic for producer messages
  topic = "test-logstore-1"
  routing_key = "content"
  ## CompressionCodec represents the various compression codecs recognized by
  ## Kafka in messages.
  ## 0 : No compression
  ## 1 : Gzip compression
  ## 2 : Snappy compression
  ## 3 : LZ4 compression
  compression_codec = 1
  ## Optional TLS Config tls_ca = "/etc/ssl/certs/ca-bundle.crt"
  # tls_cert = "/etc/telegraf/cert.pem" # tls_key = "/etc/telegraf/key.pem"
  ## Use TLS but skip chain & host verification
  # insecure_skip_verify = false
  ## Optional SASL Config
  sasl_username = "yourusername"
  sasl_password = "yourpassword"
  ## Data format to output.
  ## https://github.com/influxdata/telegraf/blob/master/docs/DATA_FORMATS_OUTPUT.md
  data_format = "json"
```

- 日志样例

  使用JSON模式输出日志后，您可以给content字段创建JSON类型的索引。更多信息，请参见JSON类型。

示例四：使用Fluentd采集日志到日志服务

Fluentd是一个开源的日志收集器，是云端原生计算基金会（CNCF）的成员项目之一，遵循Apache 2 License协议。

Fluentd支持众多输入、处理、输出插件，支持通过Kafka插件将日志上传到日志服务，您只需安装并配置Kafka插件即可。更多信息，请参见fluent-plugin-kafka。

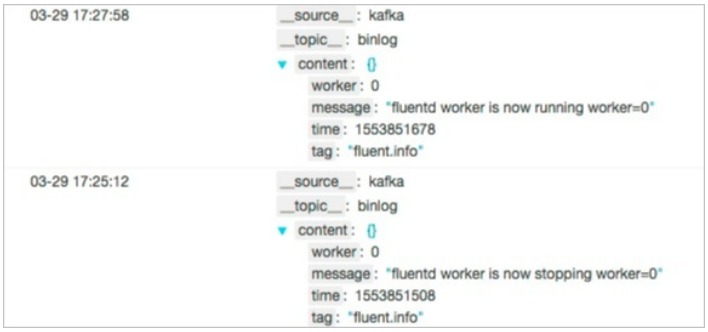
- 配置示例

  示例中将日志输出格式（Format）设置为JSON，除此之外还支持数十种Format类型。更多信息，请参见Fluentd Formatter。

```
<match **>
  @type kafka
  # Brokers: you can choose either brokers or zookeeper.
  brokers      test-project-1.cn-hangzhou.log.aliyuncs.com:10012
  default_topic test-logstore-1
  default_message_key content
  output_data_type json
  output_include_tag true
  output_include_time true
  sasl_over_ssl true
  username yourusername  //请根据真实值，替换yourusername。
  password yourpassword   //请根据真实值，替换yourpassword。
  ssl_ca_certs_from_system true
  # ruby-kafka producer options
  max_send_retries 10000
  required_acks 1
  compression_codec gzip
</match>
```

- 日志样例

  使用JSON模式输出日志后，您可以给content字段创建JSON类型的索引。更多信息，请参见JSON类型。



示例五：使用Logstash采集日志到日志服务

Logstash是一个具备实时处理能力、开源的日志采集引擎，可以动态采集不同来源的日志。

Logstash内置Kafka输出插件，您可以配置Logstash实现日志通过kafka协议上传到日志服务。由于日志服务使用SASL_SSL连接协议，因此还需要配置SSL证书和jaas文件。

- 配置示例
  i. 创建jaas文件，并保存到任意路径（例如*/etc/kafka/kafka_client_jaas.conf*）。

  将如下内容添加到jaas文件中。

  ```
  KafkaClient {
    org.apache.kafka.common.security.plain.PlainLoginModule required
    username="yourusername"
    password="yourpassword";
  };
  ```

  ii. 配置SSL信任证书，保存到任意路径（例如：*/etc/kafka/client-root.truststore.jks*）。

  下载根证书，保存到任意路径（例如：*/etc/kafka/root.pem*），然后通过keytool命令生成.jks格式的文件（首次生成时，需要配置密码）。

  ```
  keytool -keystore client-root.truststore.jks -alias root -import -file /etc/kafka/root.pem
  ```

  iii. 配置Logstash。

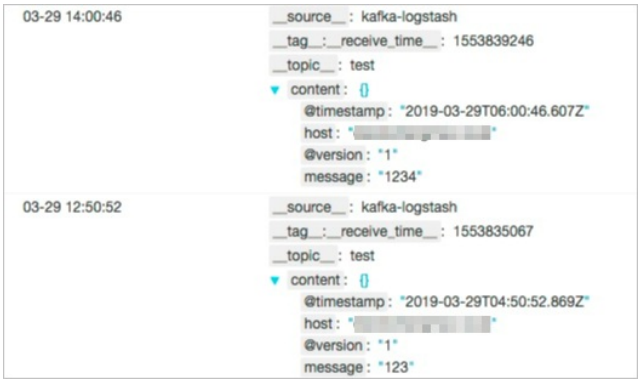
  示例中将日志输出格式（Format）设置为JSON，除此之外还支持数十种Format类型。更多信息，请参见Logstash Codec。

  > **说明** 本示例为连通性测试的配置，您的生产环境中建议删除stdout的输出配置。

  ```
  input { stdin { } }
  output {
    stdout { codec => rubydebug }
    kafka {
      topic_id => "test-logstore-1"
      bootstrap_servers => "test-project-1.cn-hangzhou.log.aliyuncs.com:10012"
      security_protocol => "SASL_SSL"
      ssl_truststore_location => "/etc/client-root.truststore.jks"
      ssl_truststore_password => "123456"
      jaas_path => "/etc/kafka_client_jaas.conf"
      sasl_mechanism => "PLAIN"
      codec => "json"
      client_id => "kafka-logstash"
    }
  }
  ```

- 日志样例

  使用JSON模式输出日志后，您可以给content字段创建JSON类型的索引。更多信息，请参见JSON类型。



错误信息

使用Kafka协议上传日志失败时，会按照Kafka的错误信息返回对应的错误信息，如下表所示，Kafka协议错误信息详情请参见error list。

错误信息	说明	推荐解决方式
NetworkException	出现网络错误时返回该错误信息。	一般等待1秒后重试即可。

错误信息	说明	推荐解决方式
TopicAuthorizationException	鉴权失败时返回该错误信息。	一般是您提供的AccessKey错误或没有写入对应Project、Logstore的权限。请填写正确的且具备写入权限的AccessKey。
UnknownTopicOrPartitionException	出现该错误可能有两种原因： • 不存在对应的Project或Logstore。 • Project所在地域与填入的Endpoint不一致。	请确保已创建对应的Project和Logstore。如果已创建还是提示该错误，请检查Project所在地域是否和填入的Endpoint一致。
KafkaStorageException	服务端出现异常时返回该错误信息。	一般等待1秒后重试即可。

6.3. 使用Syslog协议上传日志

您可以使用Rsyslog、Syslog-ng采集日志并通过Syslog协议上传到日志服务。本文介绍通过Syslog协议将日志上传到日志服务的操作步骤。

相关限制

- Syslog协议必须为标准的RFC5424协议。
- 每条日志最大支持64KB。
- 为保证数据传输安全性，数据传输必须使用基于TCP的TLS1.2（Transport-level security）。

配置方式

> **注意** 通常线下的VPN、路由器等设备不支持TLS协议或配置RFC5424格式，建议使用Logtail的Syslog插件采集这些设备的数据。具体操作，请参见采集Syslog。

使用Syslog协议上传日志时，需配置日志上传地址，格式为 `Project名称.日志服务Endpoint:Syslog协议端口` ，例如**test-project-1.cn-hangzhou-intranet.log.aliyuncs.com:10009**。请根据您的日志服务Project所在地域选择Endpoint。更多信息，请参见服务入口。Syslog的端口为10009。同时您需要在STRUCTURED-DATA字段中配置日志服务Project、Logstore，阿里云账号AccessKey等信息。

参数	说明	示例
STRUCTURED-DATA	固定为Logservice。	Logservice
Project	日志服务Project名称，请提前在日志服务中创建Project。	test-project-1
Logstore	日志服务Logstore名称，请提前在日志服务中创建Logstore。	test-logstore-1
access-key-id	AccessKey ID。建议使用RAM用户的AccessKey。更多信息，请参见授权。	<yourAccessKeyId>
access-key-secret	AccessKey Secret。建议使用RAM用户的AccessKey。更多信息，请参见授权。	<yourAccessKeySecret>

示例1：使用Rsyslog采集日志

Linux服务器默认自带Rsyslog。您可以使用Rsyslog采集系统日志，然后通过syslog协议上传到日志服务。不同版本的Rsyslog的配置文件略有不同，您可执行**man rsyslogd**命令查看Rsyslog版本。

> **说明** 请确保Rsyslog已安装了gnutls模块。如果未安装，请执行**sudo apt-get install rsyslog-gnutls**或**sudo yum install rsyslog-gnutls**命令进行安装。

1. 打开Rsyslog配置文件。

   通常Rsyslog配置文件路径为*/etc/rsyslog.conf*。
2. 根据实际情况，配置如下信息，并添加到Rsyslog配置文件的末尾。
   - Rsyslog V8及以上

     其中 `$DefaultNetstreamDriverCAFile` 配置为系统根证书所在路径。

```
# Setup disk assisted queues
$WorkDirectory /var/spool/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1     # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g       # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on     # save messages to disk on shutdown
$ActionQueueType LinkedList       # run asynchronously
$ActionResumeRetryCount -1        # infinite retries if host is down
$ActionSendTCPRebindInterval 100  # close and re-open the connection to the remote host every 100 of messages sent.
#RsyslogGnuTLS set to default ca path
$DefaultNetstreamDriverCAFile /etc/ssl/certs/ca-bundle.crt
template(name="LogServiceFormat" type="string"
string="<%pri%>1 %timestamp:::date-rfc3339% %HOSTNAME% %app-name% %procid% %msgid% [logservice project=\"test-projec
t-1\" logstore=\"test-logstore-1\" access-key-id=\"<yourAccessKeyId>\" access-key-secret=\"<yourAccessKeySecret>\"]
%msg%\n"
)
# Send messages to Loggly over TCP using the template.
action(type="omfwd" protocol="tcp" target="test-project-1.cn-hangzhou.log.aliyuncs.com" port="10009" template="LogSe
rviceFormat" StreamDriver="gtls" StreamDriverMode="1" StreamDriverAuthMode="x509/name" StreamDriverPermittedPeers="*
.cn-hangzhou.log.aliyuncs.com")
```

- Rsyslog V7及以下

  其中 `$DefaultNetstreamDriverCAFile` 配置为系统根证书所在路径。

```
# Setup disk assisted queues
$WorkDirectory /var/spool/rsyslog       # where to place spool files
$ActionQueueFileName fwdRule1           # unique name prefix for spool files $ActionQueueMaxDiskSpace 1g
# 1gb space limit (use as much as possible) $ActionQueueSaveOnShutdown on           # save messages to disk on shutd
own
$ActionQueueType LinkedList             # run asynchronously
$ActionResumeRetryCount -1              # infinite retries if host is down $ActionSendTCPRebindInterval 100        #
close and re-open the connection to the remote host every 100 of messages sent.
# RsyslogGnuTLS set to default ca path
$DefaultNetstreamDriverCAFile /etc/ssl/certs/ca-bundle.crt
$ActionSendStreamDriver gtls
$ActionSendStreamDriverMode 1
$ActionSendStreamDriverAuthMode x509/name
$ActionSendStreamDriverPermittedPeer test-project-1.cn-hangzhou.log.aliyuncs.com
template(name="LogServiceFormat" type="string" string="<%pri%>1 %timestamp:::date-rfc3339% %HOSTNAME% %app-name% %pr
ocid% %msgid% [logservice project=\"test-project-1\" logstore=\"test-logstore-1\" access-key-id=\"<yourAccessKeyId>\
" access-key-secret=\"<yourAccessKeySecret>\"] %msg%\n")
*.* action(type="omfwd" protocol="tcp" target="test-project-1.cn-hangzhou.log.aliyuncs.com" port="10009" template="L
ogServiceFormat")
```

3. 重启Rsyslog。

   执行**sudo service rsyslog restart**、**sudo /etc/init.d/syslog-ng restart**或**systemctl restart rsyslog**命令重启Rsyslog。

4. 使用**logger**命令生成测试日志。

   例如执行**logger hello world!**命令生成日志。

示例2：使用Syslog-ng采集日志

syslog-ng是基于syslog协议的Unix和类Unix系统的开源软件。您可以执行**sudo yum install syslog-ng**或**sudo apt-get install syslog-ng**命令安装Syslog-ng。

> **说明** Linux服务器上默认安装Rsyslog，但是Rsyslog和Syslog-ng无法同时工作，如果您要使用Syslog-ng请先卸载Rsyslog。

1. 打开Syslog-ng配置文件。

   通常Syslog-ng配置文件地址为*/etc/syslog-ng/syslog-ng.conf*。

2. 根据实际情况，配置如下信息，并添加到Syslog-ng配置文件的末尾。

```
### Syslog-ng Logging Config for LogService ###
template LogServiceFormat {
    template("<${PRI}>1 ${ISODATE} ${HOST:--} ${PROGRAM:--} ${PID:--} ${MSGID:--} [logservice project=\"test-project-1
\" logstore=\"test-logstore-1\" access-key-id=\"<yourAccessKeyId>\" access-key-secret=\"<yourAccessKeySecret>\"] $MSG\
n"); template_escape(no);
};
destination d_logservice{
     tcp("test-project-1.cn-hangzhou.log.aliyuncs.com" port(10009)
     tls(peer-verify(required-untrusted))
     template(LogServiceFormat));
};
log {
     source(s_sys); # default use s_sys
     destination(d_logservice);
};
### END Syslog-ng Logging Config for LogService ###
```

3. 重启Syslog-ng。

   执行**sudo /etc/init.d/syslog-ng restart**、**sudo service syslog-ng restart**或**sudo systemctl restart syslog-ng**命令重启Syslog-ng。

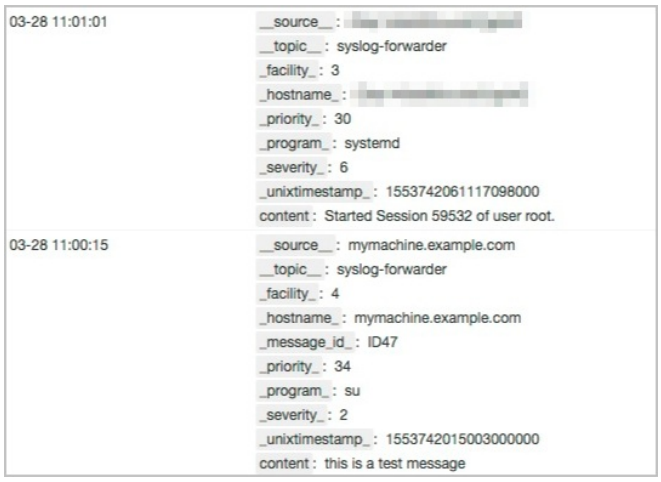
4. 使用**logger**命令生成测试日志。

   例如执行**logger hello world!**命令生成日志。

日志样例

上传日志到日志服务后，您可以在日志服务控制台查看日志。关于日志字段详情，请参见RFC5424协议。

> **说明** 为避免泄露AccessKey信息，日志服务默认将上报的Logservice字段删除。



字段名	说明
__source__	原始日志中的hostname字段。
__topic__	固定为syslog-forwarder。
__facility__	facility（设备、模块）信息。
__program__	进程名。
__serverity__	日志严重性。
__priority__	日志优先级。
__unixtimestamp__	原始日志中的时间戳（单位：纳秒）。
content	原始日志中的msg字段。

常见问题与排查

- 测试日志上传

您可以使用Netcat测试日志上传，以检查网络是否连通以及AccessKey是否具有上传权限。

i. 登录要测试日志上传的服务器。

ii. 执行以下命令安装Netcat。

```
sudo yum install nmap-ncat
```

iii. 执行以下命令连接日志服务。

```
ncat --ssl <yourProject>.<yourEndpoint> 10009
```

示例命令：

```
ncat --ssl test-project-1.cn-hangzhou.log.aliyuncs.com 10009
```

iv. Netcat不会自动判断网络连接是否中断，您需要在执行**ncat**命令后的30秒内，输入要发送的日志，然后按回车键。

```
<34>1 2019-03-28T03:00:15.003Z mymachine.example.com su - ID47 [logservice project="<yourProject>" logstore="<yourLo
gstore>" access-key-id="<yourAccessKeyID>" access-key-secret="<yourAccessKeySecret>"] this is a test message
```

示例命令：

```
<34>1 2019-03-28T03:00:15.003Z mymachine.example.com su - ID47 [logservice project="trace-doc-test" logstore="doc-te
st-001-logs" access-key-id="LTAI4***" access-key-secret="HfJEw***"] this is a test message
```

v. 在日志服务控制台预览日志，验证日志是否上传成功。

具体操作，请参见日志预览。

时间/来源	内容
2019-03-28 11:00:15 mymachine.example.com	content:this is a test message _hostname_:mymachine.example.com _severity_:2 _facility_:4 _message_id_:ID47 _unixtimestamp_:1553742015003000000 _program_:su _priority_:34

- 诊断采集错误

  如果手动上传日志失败，您可通过诊断采集错误查看具体报错信息。更多信息，请参见如何查看Logtail采集错误信息。

- 查看Rsyslog报错日志

  Rsyslog日志默认保存在*/var/log/message*中，您可通过**vim**命令查看。

  - 报错信息（1）

    ```
    dlopen: /usr/lib64/rsyslog/lmnsd_gtls.so: cannot open shared object file: No such file or directory
    ```

    该错误是因为没有安装gnutls模块，请执行**sudo apt-get install rsyslog-gnutls**或**sudo yum install rsyslog-gnutls**安装gnutls并重启Rsyslog。

  - 报错信息（2）

    ```
    unexpected GnuTLS error -53 - this could be caused by a broken connection. GnuTLS reports:Error in the push function
    ```

    该错误是因为TCP连接长期闲置被强制关闭，Rsyslog会进行自动重连，您无需关注。

- 查看Syslog-ng报错日志

  Syslog-ng日志默认保存在Journal日志中，您可执行**systemctl status syslog-ng.service**和**journalctl -xe**命令查看日志。

  如果出现如下报错，请检查配置文件格式是否合法或配置是否存在冲突，例如不支持配置多个`internal()`。

  ```
  Job for syslog-ng.service failed because the control process exited with error code. See "systemctl status syslog-ng.ser
  vice" and "journalctl -xe" for details
  ```

6.4. Logstash

6.4.1. 安装Logstash

本文介绍Logstash的安装步骤。

背景信息

Logstash是一款开源的数据采集软件，您可以通过logstash-output-logservice插件，将Logstash采集到的日志上传到日志服务。logstash-output-logservice插件的GitHub项目地址为Logstash插件。

操作步骤

1. 安装Java。
   i. 下载安装包。
   请进入Java官网下载JDK并双击进行安装。更多信息，请参见Java官网。
   ii. 设置环境变量。
   在**控制面板 > 系统 > 高级系统设置 > 高级 > 环境变量**中，设置环境变量。
   - PATH：*C:\Program Files\Java\jdk1.8.0_73\bin*
   - CLASSPATH：*C:\Program Files\Java\jdk1.8.0_73\lib;C:\Program Files\Java\jdk1.8.0_73\lib\tools.jar*
   - JAVA_HOME：*C:\Program Files\Java\jdk1.8.0_73*

   其中，请根据实际的Java版本替换*jdk1.8.0_73*。
   iii. 验证Java安装结果。
   执行**java -version**命令，显示如下类似结果表示安装Java完成。

```
PS C:\Users\Administrator> java -version
java version "1.8.0_73"
Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
Java HotSpot(TM) 64-Bit Server VM (build 25.73-b02, mixed mode)
PS C:\Users\Administrator> javac -version
javac 1.8.0_73
```

2. 安装Logstash。
   i. 下载安装包。

   > **说明** 建议下载Logstash 5.0及以上版本。

   ii. 解压安装包到指定目录。
3. 安装Logstash插件。
   i. 根据服务器所处网络环境选择安装模式。
   - 在线安装

     该插件托管于RubyGems。更多信息，请参见logstash-output-logservice 。

     进入Logstash安装目录，执行如下命令安装Logstash插件。

```
PS C:\logstash-6.4.3> .\bin\logstash-plugin install logstash-output-logservice
```

   - 离线安装
     a. 下载离线安装包。
     b. 进入Logstash安装目录，执行如下命令安装Logstash插件。

```
bin/logstash-plugin install file:///root/logstash-offline-plugins.zip
```

   ii. 验证Logstash插件安装结果。
   执行如下命令，如果返回的已安装的插件列表中有logstash-output-logservice，则表示安装成功。

```
PS C:\logstash-6.4.3> .\bin\logstash-plugin list
```

6.4.2. 创建Logstash采集配置和处理配置

本文为您介绍如何创建Logstash采集配置和处理配置。

插件介绍

- logstash-input-file插件

  logstash-input-file插件以tail方式采集日志，详情请参见logstash-input-file。
- logstash-output-logservice插件

  logstash-output-logservice插件将采集到的日志进行处理并上传到日志服务。

操作步骤

1. 创建一个配置文件到*C:\logstash-2.2.2-win\conf*\目录。

   请根据实际情况，替换logstash-2.2.2-win。您可以为每种日志创建一个配置文件，格式为**.conf*。

2. 创建采集配置和处理配置。

请根据实际情况，配置如下信息（input为采集配置，详情配置请参见Logstash官方文档，output为处理配置），并添加到配置文件中。

> **说明**
> - 配置文件格式必须以UTF-8无BOM格式编码，可以通过文本编辑器修改文件编码格式。
> - path参数中配置的文件路径，请使用UNIX模式的分隔符，例如*C:/test/multiline/*.log*，否则无法支持模糊匹配。
> - type参数在同一个配置文件内保持一致。如果服务器上存在多个Logstash配置文件，请保证各配置文件中的type参数唯一。

```
input {
  file {
    type => "iis_log_1"
    path => ["C:/inetpub/logs/LogFiles/W3SVC1/*.log"]
    start_position => "beginning"
  }
}
filter {
  if [type] == "iis_log_1" {
  #ignore log comments
  if [message] =~ "^#" {
    drop {}
  }
  grok {
    # check that fields match your IIS log settings
    match => ["message", "%{TIMESTAMP_ISO8601:log_timestamp} %{IPORHOST:site} %{WORD:method} %{URIPATH:page} %{NOTSPAC
E:querystring} %{NUMBER:port} %{NOTSPACE:username} %{IPORHOST:clienthost} %{NOTSPACE:useragent} %{NUMBER:response} %{N
UMBER:subresponse} %{NUMBER:scstatus} %{NUMBER:time_taken}"]
  }
    date {
    match => [ "log_timestamp", "YYYY-MM-dd HH:mm:ss" ]
      timezone => "Etc/UTC"
  }
  useragent {
    source=> "useragent"
    prefix=> "browser"
  }
  mutate {
    remove_field => [ "log_timestamp"]
  }
  }
}
output {
  if [type] == "iis_log_1" {
  logservice {
        codec => "json"
        endpoint => "***"
        project => "***"
        logstore => "***"
        topic => ""
        source => ""
        access_key_id => "***"
        access_key_secret => "***"
        max_send_retry => 10
        max_buffer_items => 4000
        max_buffer_bytes => 2097152
        max_buffer_seconds => 3
    }
    }
}
```

处理配置参数说明

参数	是否必填	说明
endpoint	是	日志服务的服务入口。
project	是	日志服务Project。
logstore	是	日志服务Logstore。

参数	是否必填	说明
topic	是	日志主题。
source	是	日志来源。如果为空，则自动取本机IP地址。
access_key_id	是	阿里云账号的AccessKey ID。
access_key_secret	是	阿里云账号的AccessKey Secret。
max_send_retry	是	数据包发送到日志服务发生异常时的最大重试次数，重试不成功的数据包会被丢弃，重试间隔为200毫秒。
max_buffer_items	否	每个数据包所缓存的日志条数。 如果未设置，表示使用默认值（4000条）。
max_buffer_bytes	否	每个数据包所缓存的日志大小，最大值为10485760。单位：Bytes。 如果未设置，表示使用默认值（2097152 Bytes）。
max_buffer_seconds	否	最大缓存时间。单位：秒。 如果未设置，表示使用默认值（3秒）。

3. 重启Logstash。

   具体操作，请参见启动服务。

后续步骤

在PowerShell中启动logstash.bat，logstash进程会在前台工作，一般用于配置测试和采集调试。建议调试通过后，把Logstash设置为Windows服务。可以保持后台运行以及开机自启动。更多信息，请参见设置Logstash为Windows服务。

6.4.3. 设置Logstash为Windows服务

本文介绍如何使用NSSM软件将Logstash设置为Windows服务。

背景信息

在PowerShell中启动logstash.bat后，Logstash进程会在前台工作，一般用于配置测试和采集调试。建议调试完成后把Logstash设置为Windows服务，实现后台运行以及开机自启动。您可以使用NSSM软件将Logstash设置为Windows服务，NSSM详细介绍请参见NSSM官方文档。

通过NSSM软件，您还可以使用命令行启动、停止、修改和删除服务。

添加服务

一般用于首次部署时执行，如果已添加过服务，请跳过该步骤。

您可以执行以下命令添加服务。

- 32位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe install logstash "C:\logstash-2.2.2-win\bin\logstash.bat" "agent -f C:\logstash-2.2.2-win\conf"
```

- 64位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe install logstash "C:\logstash-2.2.2-win\bin\logstash.bat" "agent -f C:\logstash-2.2.2-win\conf"
```

启动服务

您可以执行以下命令启动服务。

- 32位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe start logstash
```

- 64位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe start logstash
```

停止服务

您可以执行以下命令停止服务。

- 32位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe stop logstash
```

- 64位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe stop logstash
```

修改服务

您可以执行以下命令修改服务。

- 32位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe edit logstash
```

- 64位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe edit logstash
```

删除服务

您可以执行以下命令删除服务。

- 32位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe remove logstash
```

- 64位系统

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe remove logstash
```

6.4.4. 进阶功能

Logstash 提供了 大量插件，可以满足个性化需求，例如：

- grok：通过正则表达式将日志内容解析成多个字段。
- json_lines、json：提供结构化解析 JSON 类型日志功能。
- date：提供日志内容中有关日期、时间字段的解析、转换功能。
- multiline：可自定义更为复杂的多行日志类型。
- kv：提供结构化解析 Key-Value 类型日志格式功能。

6.4.5. Logstash 错误处理

配置Logstash采集日志数据后，如果在日志采集过程中遇到错误，可以按照报错类型选择对应处理方式。

通过Logstash收集日志数据时，如遇到以下收集错误，请按照对应建议进行处理。

- **错误描述**：在日志服务控制台查看到数据乱码。

  解决方法：Logstash默认支持UTF8格式文件编码，请确认输入的文件编码是否正确。

- **错误描述**：日志服务控制台提示错误。

  解决方法：日志服务控制台提示 `io/console not supported; tty will not be manipulated` 错误，不影响产品功能，请忽略。

其它错误类型建议通过Google或Logstash论坛，查询帮助信息。

6.5. SDK采集

为了能让您更高效地使用日志服务，日志服务提供了多个语言版本（.NET、Java、Python、PHP、C等）的SDK（Software Development Kit），您可以根据业务需求选择语言版本使用。

使用前须知

不同语言的日志服务SDK具体实现细节会有所不同，但是它们都是日志服务API在不同语言上的封装，实现的功能也基本一致。具体包括如下几个方面：

- 实现对日志服务API接口的统一封装，让您不需要关心具体的API请求构建和响应解析。而且各个不同语言的接口使用也非常接近，方便您在不同语言间切换。更多信息，请参见接口规范。
- 实现日志服务API的数字签名逻辑，让您不需要关心API的签名逻辑细节，降低使用日志服务API的难度。更多信息，请参见请求签名。
- 实现日志服务日志的ProtoBuffer格式封装，让您在写入日志时不需要关心ProtoBuffer格式的具体细节。更多信息，请参见ProtoBuffer格式。

- 实现日志服务API中定义的压缩方法，让您不用关心压缩实现的细节。部分语言的SDK支持启用压缩模式写入日志（默认为使用压缩方式）。
- 提供统一的错误处理机制，让您可以使用语言所熟悉的方式处理请求异常。更多信息，请参见错误处理机制。
- 目前所有语言实现的SDK仅提供同步请求方式。

SDK列表

下表列举了日志服务不同语言的SDK的参考文档和GitHub源码。

SDK语言	参考文档	GitHub源码
Java	Java SDK概述	Log Service Java SDK、Log Service SDK for Java 0.6.0 API
.NET Core	.NET Core SDK概述	Log Service .NET Core SDK
.NET	.NET SDK概述	Log Service .NET SDK
PHP	PHP SDK概述	Log Service PHP SDK
Python	Python SDK概述	Log Service Python SDK、User Guide
Node.js	Node.js SDK概述	Log Service Node.js SDK
C	C SDK	Log Service C SDK
GO	Go SDK概述	Log Service Go SDK
iOS	iOS SDK概述	Log Service iOS SDK、Objective-C SDK
Android	Android SDK概述	Log Service Android SDK
C++	C++ SDK概述	Log Service C++ SDK
JavaScript SDK	• 浏览器JavaScript SDK • 小程序JavaScript SDK	无

7.采集常见日志
7.1. 采集Log4j日志

本文介绍如何通过Loghub Log4j Appender或Logtail来采集Log4j日志。

背景信息

Log4j是Apache的一个开放源代码项目，通过Log4j，可以控制日志的优先级、输出目的地和输出格式。日志的优先级从高到低为ERROR、WARN、 INFO、DEBUG，日志的输出目的地指定了将日志打印到控制台还是文件中，输出格式控制了输出的日志内容格式。 本文档以Log4j的默认配置为例进行说明，如下所示。

```
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss:SSS zzz} [%t] %-5level %logger{36} - %msg%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="com.foo.Bar" level="trace">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

日志输出样例如下所示。

```
2013-12-25 19:57:06,954 [10.10.10.10] WARN impl.PermanentTairDaoImpl - Fail to Read Permanent Tair,key:e:470217319319741_1
,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]
```

通过Loghub Log4j Appender采集Log4j日志

通过Loghub Log4j Appender采集Log4j日志的操作步骤请参见Log4j Appender。

通过Logtail采集Log4j日志

1.
2. 在**接入数据**区域，选择**正则-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 在**Logtail配置**页签中，创建Logtail配置。

参数	说明
配置名称	Logtail配置的名称，设置后不可修改。 您也可以单击**导入其他配置**，导入其他Project中已创建的Logtail配置。
日志路径	指定日志的目录和文件名。
设置采集黑名单	开启该功能后，可设置**黑名单配置**。黑名单配置可在采集时忽略指定的目录或文件，目录和文件名支持完整匹配，也支持通配符模式匹配。例如： ○ 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。
是否为Docker文件	如果是Docker文件，可以直接配置内部路径与容器Tag，Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	配置为**完整正则模式**。
单行模式	关闭**单行模式**。
日志样例	输入如下Log4j日志样例。 `2013-12-25 19:57:06,954 [10.10.10.10] WARN impl.PermanentTairDaoImpl - Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]`
行首正则表示式	填写日志样例后，单击**自动生成**，生成行首正则表达式。本示例中，以日期时间表示一行的开头，行首正则表达式为\d+-\d+-\d+\s.*。
提取字段	开启**提取字段**，通过正则表达式将日志内容提取为Key-Value对。
正则表达式	本示例中，配置为(\d+-\d+-\d+\s\d+:\d+:\d+,\d+)\s\[([^\]]*)\]\s(\S+)\s+(\S+)\s-\s(.*)，在实际场景中，根据以下方式配置正则表达式。 ○ 自动生成正则表达式 在**日志样例**框中，选中需要提取的字段，单击**生成正则**，自动生成正则表达式。 ○ 手动输入正则表达式。 单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否可以解析、提取日志样例，详情请参见如何调试正则表达式。
日志抽取内容	通过正则表达式将日志内容提取为Value后，您需要为每个Value设置对应的Key。
使用系统时间	本示例中关闭了**使用系统时间**，并配置时间格式为%Y-%m-%d %H:%M:%S，在实际场景中，根据以下方式配置。 ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要在日志抽取内容中指定time字段，并根据time字段的值配置时间转换格式。时间格式详情请参见时间格式。
丢弃解析失败日志	○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

Logtail配置完成后，日志服务开始采集Log4j日志。

7. 预览数据及设置索引，单击**下一步**。

日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

说明 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

7.2. 采集Python日志

本文介绍如何通过Logtail采集Python日志。

背景信息

Python的logging模块提供通用的日志系统，可供第三方模块或者应用使用。logging模块定义了不同的日志级别和记录日志的方式。logging模块包括logger、handler、filter、formatter四个组件。

在formatter中定义日志输出格式，采用%(key)s形式。

```
import logging
import logging.handlers
LOG_FILE = 'tst.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes = 1024*1024, backupCount = 5) # 实例化handler
fmt = '%(asctime)s - %(filename)s:%(lineno)s - %(levelno)s %(levelname)s %(pathname)s %(module)s %(funcName)s %(created)f
%(thread)d %(threadName)s %(process)d %(name)s - %(message)s' //定义日志格式
formatter = logging.Formatter(fmt)   # 实例化formatter。
handler.setFormatter(formatter)      # 为handler添加formatter。
logger = logging.getLogger('tst')    # 获取名为tst的logger。
logger.addHandler(handler)           # 为logger添加handler。
logger.setLevel(logging.DEBUG)
logger.info('first info message')
logger.debug('first debug message')
```

具体字段说明如下所示。

字段	说明
%(name)s	生成日志的Logger名称。
%(levelno)s	数字形式的日志级别。
%(levelname)s	文本形式的日志级别，包括DEBUG、INFO、WARNING、ERROR和CRITICAL。
%(pathname)s	该日志所在源文件的路径。
%(filename)s	文件名。
%(module)s	该日志所在的模块名。
%(funcName)s	日志输出函数的函数名。
%(lineno)d	日志输出函数的语句所在的代码行。
%(created)f	日志创建时间，UNIX标准时间格式，表示从1970-1-1 00:00:00 UTC计算起的秒数。
%(relativeCreated)d	日志创建时间与logging模块被加载时间的时间差，单位为毫秒。
%(asctime)s	日志创建时间。默认格式是2003-07-08 16:49:45,896，半角逗号（,）后面的数字为毫秒数。
%(msecs)d	日志创建时间，毫秒级别。
%(thread)d	线程ID。
%(threadName)s	线程名称。
%(process)d	进程ID。
%(message)s	日志信息。

输出的日志样例如下所示。

```
2015-03-04 23:21:59,682 - log_test.py:16 - tst - first info message
2015-03-04 23:21:59,682 - log_test.py:17 - tst - first debug message
```

操作步骤

1.
2. 在**接入数据**区域，选择**正则-文本日志**。

3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

        日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**Logtail配置**页签中，创建Logtail配置。

参数	说明
配置名称	Logtail配置的名称，在其所属Project内必须唯一。创建Logtail配置成功后，无法修改其名称。 您也可以单击**导入其他配置**，导入其他已创建的Logtail配置。
日志路径	指定日志的目录和文件名。
设置采集黑名单	开启该功能后，可设置**黑名单配置**。黑名单配置可在采集时忽略指定的目录或文件，目录和文件名支持完整匹配，也支持通配符模式匹配。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。 **说明** • 目录通配符只支持星号（*）和半角问号（?）。 • 如果您在配置**日志路径**时使用了通配符，但又需要过滤掉其中部分路径时，需在黑名单中填写对应的完整路径来保证过滤生效。 例如您配置**日志路径**为*/home/admin/app*/log/*.log*，但要过滤*/home/admin/app1*目录下的所有子目录，则需配置黑名单，即选择**按目录路径**，配置路径为*/home/admin/app1*/**。如果配置为*/home/admin/app1**，则黑名单不会生效。 • 匹配黑名单过程存在计算开销，建议黑名单条目数在10条内。
是否为Docker文件	如果是Docker文件，可以直接配置内部路径与容器Tag，Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	配置为**完整正则模式**。
单行模式	开启**单行模式**，即每行为一条日志。

参数	说明

日志样例	输入如下日志样例。
	`2016-02-19 11:06:52,514 - test.py:19 - 10 DEBUG test.py test <module>`
	`1455851212.514271 139865996687072 MainThread 20193 tst - first debug message`

提取字段	开启**提取字段**，通过正则表达式将日志内容提取为Key-Value对。
正则表达式	本示例中，配置为(\d+-\d+-\d+\s\S+)\s-\s([^:]+):(\d+)\s+-\s+(\d+)\s+(\w+)\s+(\S+)\s+(\w+)\s+(\S+)\s+(\S+)\s+(\d+)\s+(\w+)\s+(\d+)\s+(\w+)\s+-\s+(.*)，在实际场景中，根据以下方式配置正则表达式。 ○ 自动生成正则表达式 在**日志样例**框中，选中需要提取的字段，单击**生成正则**，自动生成正则表达式。 ○ 手动输入正则表达式。 单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否可以解析、提取日志样例，详情请参见如何调试正则表达式。
日志抽取内容	开启**提取字段**后，需要设置。 通过正则表达式将日志内容提取为Value后，您需要为每个Value设置对应的Key。
使用系统时间	开启**提取字段**后，需要设置。 ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要在日志抽取内容中指定time字段，并根据time字段的值配置时间转换格式。时间格式详情请参见时间格式。
丢弃解析失败日志	○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。

参数	描述
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

Logtail配置完成后，日志服务开始采集Python日志。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

7.3. 采集Node.js日志

本文介绍如何使用Logtail采集Node.js日志。

背景信息

log4js是一个Node.js日志管理工具，您可以通过log4js把Node.js日志打印到文件中，并自定义日志格式，便于日志采集和整理。

```
var log4js = require('log4js');
log4js.configure({
  appenders: [
    {
      type: 'file', //文件输出
      filename: 'logs/access.log',
      maxLogSize: 1024,
      backups:3,
      category: 'normal'
    }
  ]
});
var logger = log4js.getLogger('normal');
logger.setLevel('INFO');
logger.info("this is a info msg");
logger.error("this is a err msg");
```

通过log4js将日志打印到文件中，输出的日志样例如下所示。log4js分为6个输出级别，从低到高分别为trace、debug、info、warn、error、fatal。

```
[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
[2016-02-24 17:42:38.951] [ERROR] normal - this is a err msg
```

操作步骤

1.
2. 在**接入数据**区域，选择**正则-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

     更多信息，请参见安装Logtail（ECS实例）。

     > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。
     c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

     日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 在**Logtail配置**页签中，创建Logtail配置。

参数	说明
配置名称	Logtail配置的名称，设置后不可修改。 您也可以单击**导入其他配置**，导入其他Project中已创建的Logtail配置。
日志路径	指定日志的目录和文件名。
设置采集黑名单	开启该功能后，可设置**黑名单配置**。黑名单配置可在采集时忽略指定的目录或文件，目录和文件名支持完整匹配，也支持通配符模式匹配。例如： • 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 • 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 • 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 • 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 • 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。
是否为Docker文件	如果是Docker文件，可以直接配置内部路径与容器Tag，Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	配置为**完整正则模式**。
单行模式	开启**单行模式**，即每行为一条日志。
日志样例	输入如下日志样例。 `[2016-01-31 12:02:25.844] [INFO] access - 10.10.10.10 - - "GET /user/projects/ali_sls_log?ignoreError=true HTTP/1.1" 304 - "http://aliyun.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/10.10.10.10 Safari/537.36"`

参数	说明
提取字段	开启**提取字段**，通过正则表达式将日志内容提取为Key-Value对。
正则表达式	本示例中，配置为\[([^]]+)]\s\[(\w+)]\s(\w+)\s-\s(\S+)\s-\s-\s"([^"]+)"\s(\d+)[^"]+("[^"]+)"\s"([^"]+).*，在实际场景中，根据以下方式配置正则表达式。 ○ 自动生成正则表达式 在**日志样例**框中，选中需要提取的字段，单击**生成正则**，自动生成正则表达式。 ○ 手动输入正则表达式。 单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否可以解析、提取日志样例，详情请参见如何调试正则表达式。
日志抽取内容	通过正则表达式将日志内容提取为Value后，您需要为每个Value设置对应的Key。
使用系统时间	开启**提取字段**后，需要设置。 具体说明如下： ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要在日志抽取内容中指定time字段，并根据time字段的值配置时间转换格式。时间格式详情请参见时间格式。
丢弃解析失败日志	是否丢弃解析失败的日志，具体说明如下： ○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。

参数	描述
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

Logtail配置完成后，日志服务开始采集Node.js日志。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

7.4. 采集WordPress日志

本文介绍如何使用Logtail采集WordPress日志。

背景信息

WordPress是使用PHP语言和MySQL数据库开发的博客平台，并逐步演化成一款内容管理系统软件。WordPress日志样例如下所示。

```
192.0.2.0 - - [07/Jan/2020:21:06:39 +0800] "GET /wp-admin/js/password-strength-meter.min.js?ver=4.4 HTTP/1.0" 200 776 "http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.0.2.1 Safari/537.36"
```

操作步骤

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**正则-文本日志**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 创建机器组。
   - 如果您已有可用的机器组，请单击**使用现有机器组**。
   - 如果您还没有可用的机器组，请执行以下操作（以ECS为例）。
     a. 在**ECS机器**页签中，通过手动选择实例方式选择目标ECS实例，单击**立即执行**。

        更多信息，请参见安装Logtail（ECS实例）。

        > **说明** 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。手动安装Logtail后，您还需要在该服务器上手动配置用户标识。具体操作，请参见配置用户标识。

     b. 安装完成后，单击**确认安装完毕**。

c. 在**创建机器组**页面，输入**名称**，单击**下一步**。

日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。

5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

> **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。

6. 在**Logtail配置**页签中，创建Logtail配置。

参数	说明
配置名称	Logtail配置的名称，设置后不可修改。 您也可以单击**导入其他配置**，导入其他Project中已创建的Logtail配置。
日志路径	指定日志的目录和文件名。
设置采集黑名单	开启该功能后，可设置**黑名单配置**。黑名单配置可在采集时忽略指定的目录或文件，目录和文件名支持完整匹配，也支持通配符模式匹配。例如： ○ 选择**按目录路径**，配置路径为*/home/admin/dir1*，则表示在采集时忽略*/home/admin/dir1*目录下的所有内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/dir**，则表示在采集时忽略*/home/admin/*目录下所有以dir开头的子目录下的内容。 ○ 选择**按目录路径**，配置路径为*/home/admin/*/dir*，则表示在采集时忽略*/home/admin/*目录下二级目录名为dir的子目录下的所有内容。 例如*/home/admin/a/dir*目录下的内容被忽略，*/home/admin/a/b/dir*目录下的内容被采集。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*.log*，则表示在采集时忽略*/home/admin/*目录下所有以private开头，以.log结尾的文件。 ○ 选择**按文件路径**，配置路径为*/home/admin/private*/*_inner.log*，则表示在采集时忽略*/home/admin/*目录下以private开头的目录内，以_inner.log结尾的文件。 例如*/home/admin/private/app_inner.log*文件被忽略，*/home/admin/private/app.log*文件被采集。
是否为Docker文件	如果是Docker文件，可以直接配置内部路径与容器Tag，Logtail会自动监测容器创建和销毁，并根据Tag进行过滤采集指定容器的日志。关于容器文本日志采集请参见通过DaemonSet-控制台方式采集容器文本日志。
模式	配置为**完整正则模式**。
单行模式	关闭**单行模式**。
日志样例	输入如下WordPress日志样例。 `192.0.2.0 - - [07/Jan/2020:21:06:39 +0800] "GET /wp-admin/js/password-strength-meter.min.js?ver=4.4 HTTP/1.0" 200 776 "http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/192.0.2.1 Safari/537.36"`
行首正则表达式	填写日志样例后，单击**自动生成**，生成行首正则表达式。本示例中，以IP地址表示一行的开头，行首正则表达式为\d+\.\d+\.\d+\.\d+\s-\s.*。
提取字段	开启**提取字段**，通过正则表达式将日志内容提取为Key-Value对。
正则表达式	本示例中，配置为(\S+) - - \[([^\]]*)] "(\S+) ([^"]+)" (\S+) (\S+) "([^"]+)" "([^"]+)"，在实际场景中，根据以下方式配置正则表达式。 ○ 自动生成正则表达式 在**日志样例**框中，选中需要提取的字段，单击**生成正则**，自动生成正则表达式。 ○ 手动输入正则表达式。 单击**手动输入正则表达式**，手动配置。配置完成后，单击**验证**即可验证您输入的正则表达式是否可以解析、提取日志样例，详情请参见如何调试正则表达式。
日志抽取内容	开启**提取字段**后，需要设置。 通过正则表达式将日志内容提取为Value后，您需要为每个Value设置对应的Key。

参数	说明
使用系统时间	本示例中关闭了**使用系统时间**，并配置时间格式为%d/%b/%Y:%H:%M:%S，在实际场景中，根据以下方式配置。 ○ 打开**使用系统时间**开关，则日志时间为采集日志时，Logtail所在主机或容器的系统时间。 ○ 关闭**使用系统时间**开关，则您需要在日志抽取内容中指定time字段，并根据time字段的值配置时间转换格式。时间格式详情请参见时间格式。
丢弃解析失败日志	○ 打开**丢弃解析失败日志**开关，解析失败的日志不上传到日志服务。 ○ 关闭**丢弃解析失败日志**开关，日志解析失败时，原始日志将作为__raw_log__字段的值上传到日志服务。
最大监控目录深度	设置日志目录被监控的最大深度。最大深度范围：0~1000，0代表只监控本层目录。

请根据您的需求选择高级配置。如果没有特殊需求，建议保持默认配置。

参数	描述
启用插件处理	打开**启用插件处理**开关后，您可以设置Logtail插件处理日志。更多信息，请参见概述。 **说明** 打开**启用插件处理**开关后，上传原始日志、时区属性、丢弃解析失败日志、过滤器配置、接受部分字段（分隔符模式）等功能不可用。
上传原始日志	打开**上传原始日志**开关后，原始日志将作为__raw__字段的值与解析过的日志一起上传到日志服务。
Topic生成方式	设置Topic生成方式。更多信息，请参见日志主题。 ○ **空-不生成Topic**：默认选项，表示设置Topic为空字符串，在查询日志时不需要输入Topic即可查询。 ○ **机器组Topic属性**：设置为机器组Topic属性，用于明确区分不同服务器产生的日志数据。 ○ **文件路径正则**：设置为文件路径正则，则需要设置**自定义正则**，用正则表达式从路径里提取一部分内容作为Topic。用于区分不同用户或实例产生的日志数据。
日志文件编码	设置日志文件编码格式，取值为utf8、gbk。
时区属性	采集日志时，日志时间的时区属性。 ○ 机器时区：默认为Logtail所在主机或容器的时区。 ○ 自定义时区：手动选择时区。
超时属性	如果一个日志文件在指定时间内没有任何更新，则认为该文件已超时。 ○ 永不超时：持续监控所有日志文件，永不超时。 ○ 30分钟超时：如果日志文件在30分钟内没有更新，则认为已超时，并不再监控该文件。 选择**30分钟超时**时，还需设置**最大超时目录深度**，范围为1~3。
过滤器配置	只采集完全符合过滤器条件的日志。例如： ○ 满足条件即采集，例如设置**Key**为**level**，**Regex**为**WARNING|ERROR**，表示只采集level为WARNING或ERROR类型的日志。 ○ 过滤不符合条件的日志。更多信息，请参见Regular-Expressions.info。 ■ 设置**Key**为**level**，**Regex**为**^(?!.*(INFO|DEBUG)).***，表示不采集level中包含INFO或DEBUG类型的日志。 ■ 设置**Key**为**level**，**Regex**为**^(?!(INFO|DEBUG)$).***，表示不采集level等于INFO或DEBUG类型的日志。 ■ 设置**Key**为**url**，**Regex**为**.*^(?!.*(healthcheck)).***，表示不采集URL中带有healthcheck的日志。例如Key为url，Value为/inner/healthcheck/jiankong.html的日志将不会被采集。 更多信息，请参见regex-exclude-word、regex-exclude-pattern。
首次采集大小	通过**首次采集大小**，可以确认首次采集的新文件的内容位置。日志服务默认**首次采集大小**为1024 KB，即： ○ 首次采集时，如果文件小于1024 KB，则从文件内容起始位置开始采集。 ○ 首次采集时，如果文件大于1024 KB，则从距离文件末尾1024 KB的位置开始采集。 您可以通过此处修改**首次采集大小**，取值范围为0~10485760，单位为KB。

参数	描述
扩展配置	Logtail的扩展配置。更多信息，请参见advanced参数说明。 例如您想要通过当前Logtail配置去采集其他Logtail配置已匹配的文件，并指定聚合发送周期，可添加如下配置。 `{ "force_multiconfig": true, "batch_send_interval": 3 }`

Logtail配置完成后，日志服务开始采集WordPress日志。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

7.5. 采集Unity3D日志

本文介绍如何通过Web Tracking采集Unity3D日志。

背景信息

Unity3D是由Unity Technologies开发的，一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎。

日志服务支持使用Web Tracking采集Unity3D日志，Web Tracking详情请参见使用Web Tracking采集日志。本文以采集Unity Debug.Log为例，介绍Unity日志的采集。

操作步骤

1. 开通Web Tracking，详情请参见使用Web Tracking采集日志。
2. 注册Unity3D LogHandler。

   在Unity editor中创建C#文件*LogOutputHandler.cs*，输入如下脚本，并根据实际情况修改相关变量，分别为：

   - project：日志服务项目名称。
   - logstore：日志库名称。
   - serviceAddr：日志服务项目的地址，详情请参见服务入口。

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour
{
    //Register the HandleLog function on scene start to fire on debug.log events
    public void OnEnable()
    {
        Application.logMessageReceived += HandleLog;
    }
    //Remove callback when object goes out of scope
    public void OnDisable()
    {
        Application.logMessageReceived -= HandleLog;
    }
    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "http address of your log service project";
    //Capture debug.log output, send logs to Loggly
    public void HandleLog(string logString, string stackTrace, LogType type)
    {
        string parameters = "";
        parameters += "Level=" + WWW.EscapeURL(type.ToString());
        parameters += "&";
        parameters += "Message=" + WWW.EscapeURL(logString);
        parameters += "&";
        parameters += "Stack_Trace=" + WWW.EscapeURL(stackTrace);
        parameters += "&";
        //Add any User, Game, or Device MetaData that would be useful to finding issues later
        parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/logstores/" + logstore + "/track?APIVersion=0.6.0&" +
parameters;
        StartCoroutine(SendData(url));
    }
    public IEnumerator SendData(string url)
    {
        WWW sendLog = new WWW(url);
        yield return sendLog;
    }
}
```

提供上述脚本可以异步发送日志到阿里云日志服务中，您还可以添加更多想要采集的字段。

3. 产生Unity3D日志。

   在工程中创建*LogglyTest.cs*文件，并输入如下脚本。

```
using UnityEngine;
using System.Collections;
public class LogglyTest : MonoBehaviour {
    void Start () {
        Debug.Log ("Hello world");
    }
}
```

4. 查看日志。

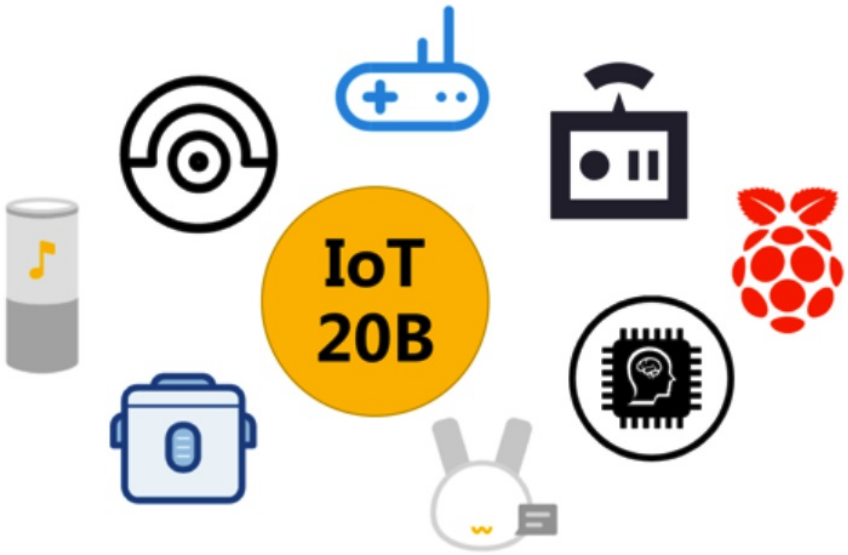
   运行Unity程序后，即可在日志服务控制台看到已采集的日志。

8.最佳实践

8.1. 采集-IoT/嵌入式日志

IoT（Internet of Things）正在高速增长，越来越多设备开始逐步走进日常生活，例如智能路由器、各种电视棒、天猫精灵、扫地机器人等，让我们体验到智能领域的便利。传统软件领域的嵌入式开发模式在IoT设备领域的应用遇到了很多挑战，IoT设备数目多、分布广，难以调试且硬件受限，传统的设备日志解决方案无法完美满足需求。

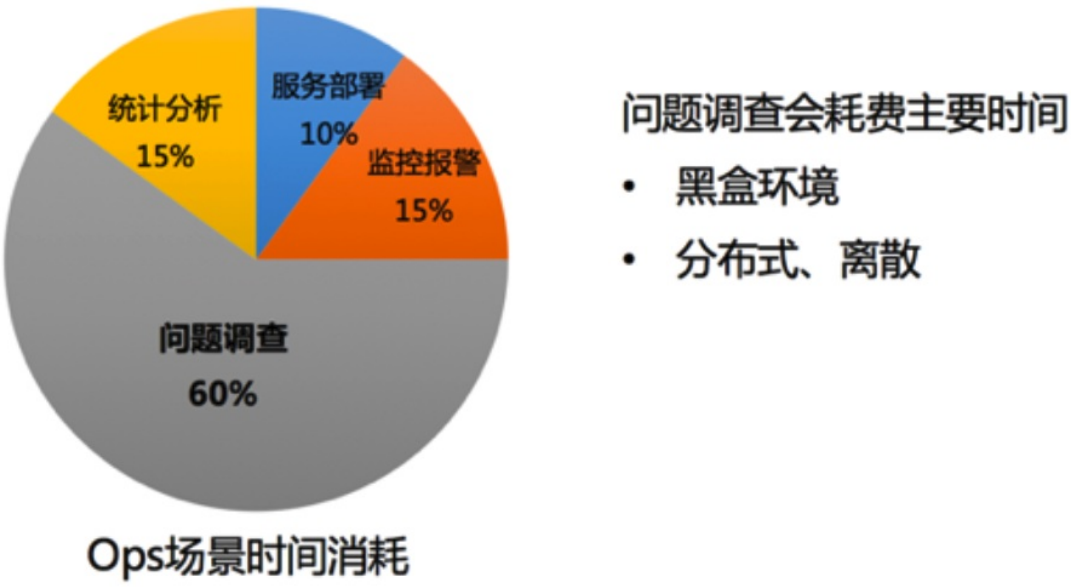
日志服务团队结合IoT设备的特点，为IoT设备量身定制一套日志数据采集方案：C Producer。



嵌入式开发需求

作为IoT/嵌入式工程师，除了需要深厚的开发功底外，面对海量的设备，如何有能力管理、监控、诊断黑盒设备至关重要。嵌入式开发需求主要有以下几点：

- 数据采集：如何实时采集分散在全球各地的百万/千万级设备上的数据？
- 调试：如何使用一套方案既满足线上数据采集又满足开发时的实时调试？
- 线上诊断：某个线上设备出现错误，如何快速定位设备，查看引起该设备出错的上下文是什么？
- 监控：当前有多少个设备在线？工作状态分布如何？地理位置分布如何？出错设备如何实时告警？
- 数据实时分析：设备产生数据如何与实时计算、大数据仓库对接，构建用户画像？



IoT领域面临的主要挑战

思考以上问题的解决方案，我们发现在传统软件领域那一套手段面临IoT领域基本全部失效，主要挑战来自于IoT设备这些特点：

- 设备数目多：在传统运维领域管理1万台服务器属于一家大公司了，但10万在线对于IoT设备而言只是一个小门槛。
- 分布广：硬件一旦部署后，往往会部署在全国、甚至全球各地。
- 黑盒：难以登录并调试，大部分情况属于不可知状态。
- 资源受限：出于成本考虑，IoT设备硬件较为受限（例如总共只有32MB内存），传统PC领域手段往往失效。

C Producer

日志服务量身定制的日志数据采集解决方案。

日志服务客户端Logtail在X86服务器上有百万级部署，可以参见文章：Logtail技术分享：，。除此之外，日志服务提供多样化的采集方案：

- 移动端SDK：Android/IOS平台数据采集，一天已有千万级DAU。
- Web Tracking（JS）：类似百度统计，Analytics轻量级采集方式，无需签名。

日志服务团队结合IoT设备的特点，为IoT设备量身定制一套日志数据采集方案：C Producer。

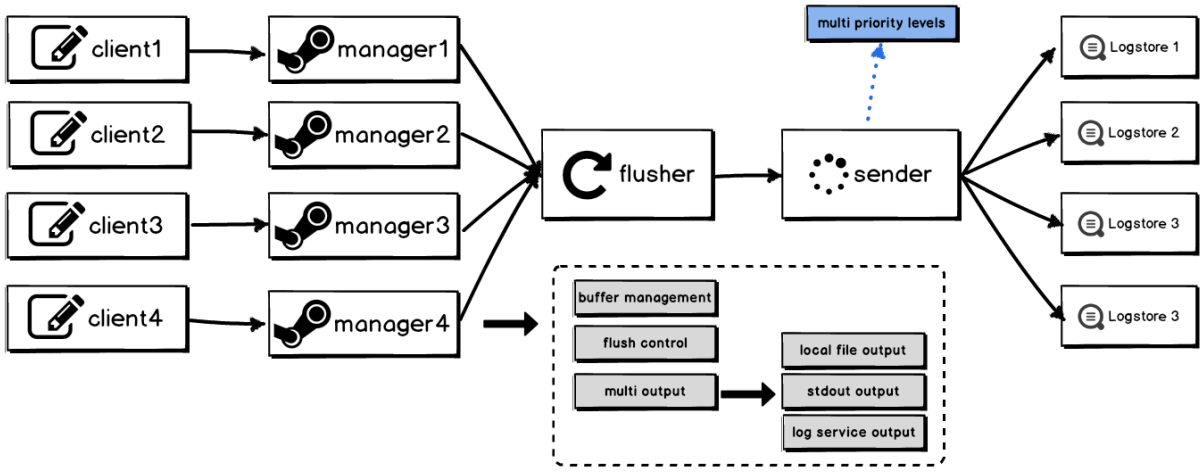
C Producer特点

C Producer Library继承Logtail稳定、高性能、低资源消耗等特点，可以定位是一个轻量级Logtail，虽没有Logtail实时配置管理机制，但具备除此之外70%功能，包括：

- 提供多租户概念：可以对多种日志（例如Metric，DebugLog，ErrorLog）进行优先级分级处理，同时配置多个客户端，每个客户端可独立配置采集优先级、目标Project和Logstore等。
- 支持上下文查询：同一个客户端产生的日志在同一上下文中，支持查看某条日志前后相关日志。
- 并发发送，断点续传：支持缓存上限可设置，超过上限后日志写入失败。

此外，C Producer还具备以下IoT设备专享功能，例如：

- 本地调试：支持将日志内容输出到本地，并支持轮转、日志数、轮转大小设置。
- 细粒度资源控制：支持针对不同类型数据/日志设置不同的缓存上限、聚合方式。
- 日志压缩缓存：支持将未发送成功的数据压缩缓存，减少设备内存占用。



功能优势

C Producer作为IoT设备的量身定制方案，在以下方面具备明显优势：

- 客户端高并发写入：可配置的发送线程池，支持每秒数十万条日志写入，详情参见性能测试。
- 低资源消耗：每秒20万日志写入只消耗70%CPU；同时在低性能硬件（例如树莓派）上，每秒产生100条日志对资源基本无影响。
- 客户端日志不落盘：既数据产生后直接通过网络发往服务端。
- 客户端计算与I/O逻辑分离：日志异步输出，不阻塞工作线程。
- 支持多优先级：不同客户端可配置不同的优先级，保证高优先级日志最先发送。
- 本地调试：支持设置本地调试，便于您在网络不通的情况下本地测试应用程序。

在以上场景中，C Producer Library简化您程序开发的步骤。您无需关心日志采集细节实现、也不用担心日志采集会影响您的业务正常运行，大大降低数据采集门槛。

C Producer方案与其他嵌入式采集方案对比如下：

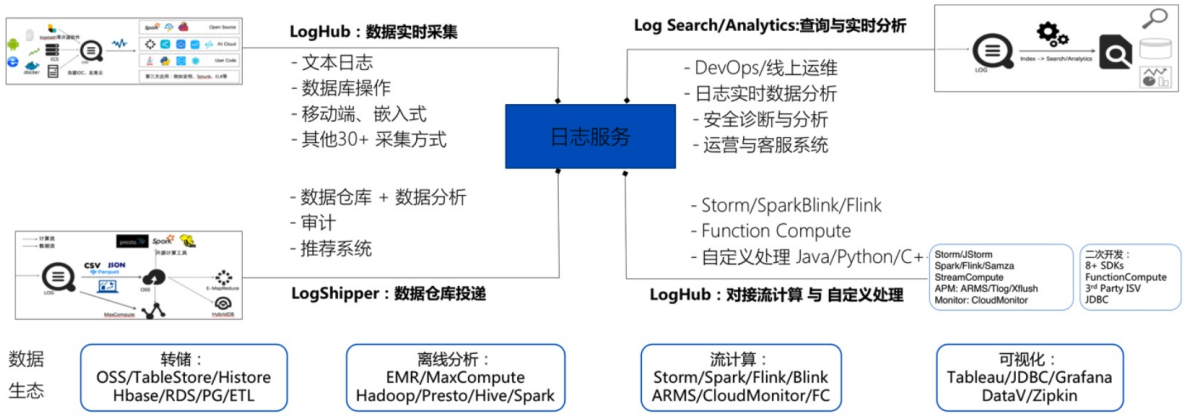
类别		C Producer	其他方案
	平台	移动端+嵌入式	移动端为主
	上下文	支持	不支持
	多日志	支持	不支持（一种日志）

编程 类别		C Producer	其他方案
	自定义格式	支持	不支持（提供若干个有限字段）
	优先级	支持	不支持
	环境参数	可配置	可配置
稳定性	并发度	高	一般
	压缩算法	LZ4（效率与性能平衡）+GZIP	优化
	低资源消耗	优化	一般
传输	断电续传	支持	默认不支持，需要二次开发
	接入点	8（中国）+8（全球）	杭州
调试	本地日志	支持	手动支持
	参数配置	支持	不支持
实时性	服务端可见	1秒（99.9%），3秒（Max）	1-2小时
自定义处理		15+对接方式	定制化实时+离线方案

C Producer+日志服务解决方案

C Producer结合阿里云日志服务产品配合使用，即可完成IoT设备日志全套解决方案。

- 规模大
  - 支持亿级别客户端实时写入。
  - 支持PB/Day数据量。
- 速度快
  - 采集快：写入零延迟，写入即可消费。
  - 查询快：一秒内，复杂查询（5个条件）可处理10亿级数据。
  - 分析快：一秒内，复杂分析（5个维度聚合+GroupBy）可聚合亿级别数据。
- 对接广
  - 与阿里云各类产品无缝打通。
  - 各种开源格式存储、计算、可视化系统完美兼容。

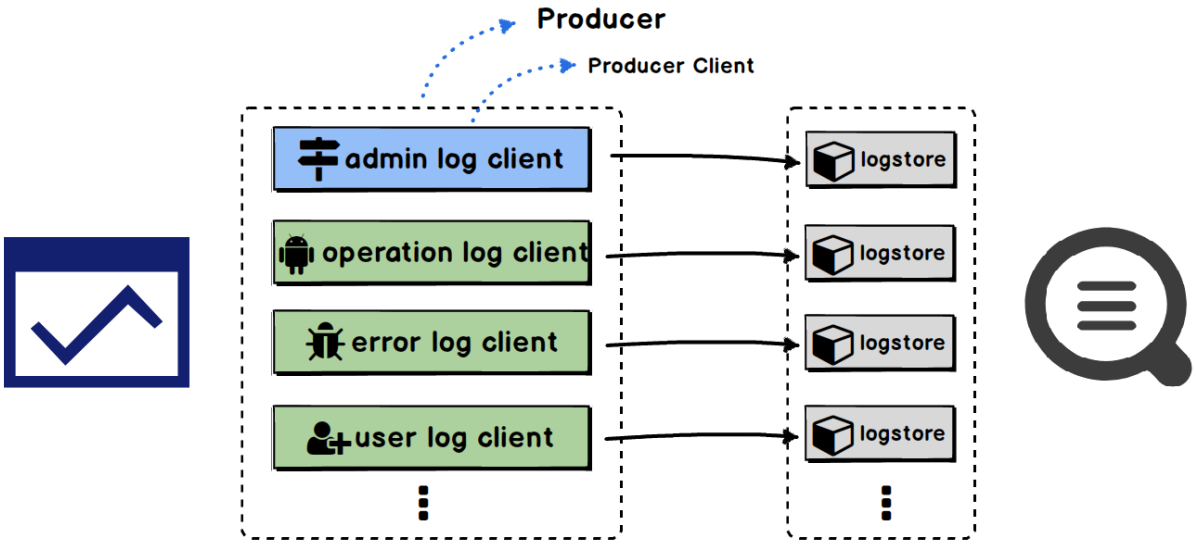


下载与使用

下载地址：Github

一个应用可创建多个Producer，每个Producer可包含多个Client，每个Client可单独配置目的地址、日志级别、是否本地调试、缓存大小、自定义标识、topic等信息。

详细安装方式及操作步骤，请参见README。



性能测试

环境配置

- 高性能场景：传统X86服务器。
- 低性能场景：树莓派（低功耗环境）。

配置如下：

高性能场景	低性能场景
• CPU : Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50GHz • MEM : 64GB • OS : Linux version 2.6.32-220.23.2.ali1113.el5.x86_64 • GCC : 4.1.2 • c-producer：动态库 162K、静态库140K (测试使用静态库，编译后的binary 157KB，所有都是strip后)	型号：树莓派3B CPU：Broadcom BCM2837 1.2GHz A53 64位(使用主机USB供电，被降频到600MHz) 内存：1GB DDR2 OS: Linux 4.9.41-v7+ #1023 SMP armv71 GNU/Linux GCC：6.3.0 (Raspbian 6.3.0-18+rpi1) c-producer：动态库 179K、静态库162K (测试使用静态库，编译后的binary 287KB，所有都是strip后)

C Producer配置

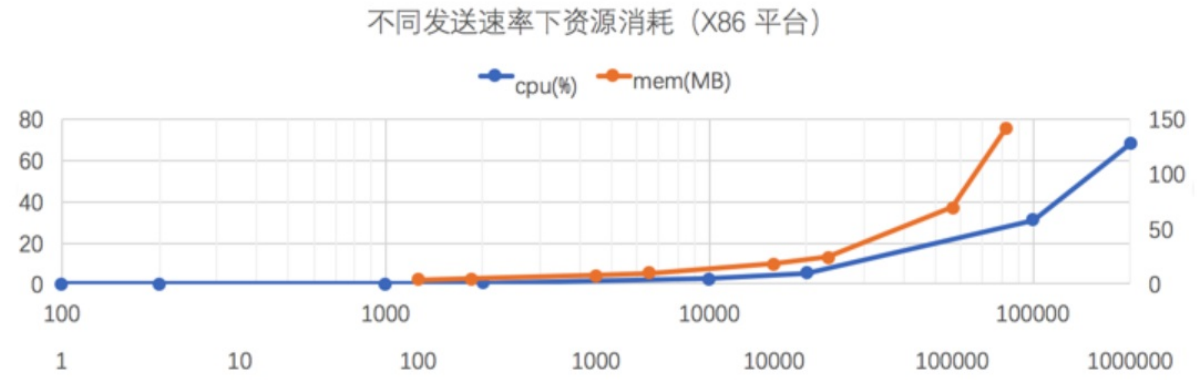
- ARM（树莓派）
  - 缓存：10 MB
  - 聚合时间：3秒（聚合时间、聚合数据包大小、聚合日志数任一满足即打包发送）
  - 聚合数据包大小：1 MB
  - 聚合日志数：1000
  - 发送线程：1
  - 自定义tag：5
- X86
  - 缓存：10MB
  - 聚合时间：3秒（聚合时间、聚合数据包大小、聚合日志数任一满足即打包发送）
  - 聚合数据包大小：3 MB
  - 聚合日志数：4096
  - 发送线程：4
  - 自定义tag：5

日志样例（9个键值对，数据量约为350字节）

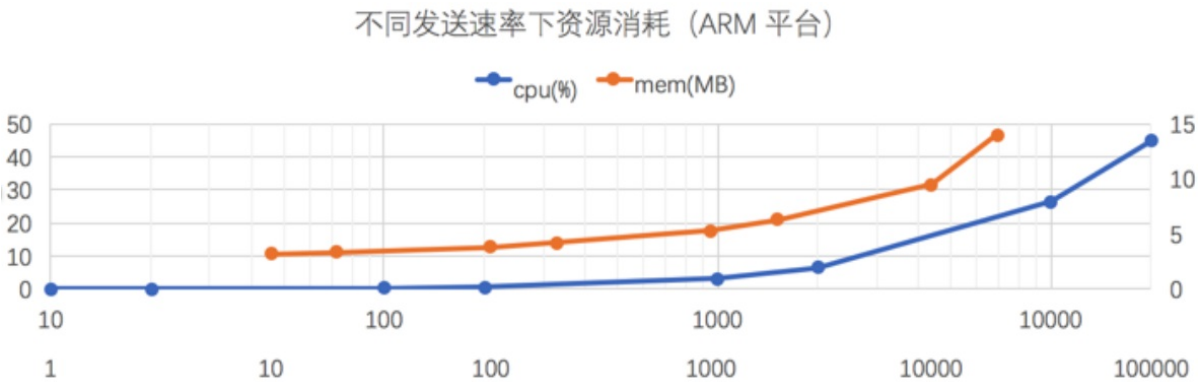
```
__source__: 192.0.2.1
__tag__:1: 2
__tag__:5: 6
__tag__:a: b
__tag__:c: d
__tag__:tag_key: tag_value
__topic__: topic_test
_file_: /disk1/workspace/tools/aliyun-log-c-sdk/sample/log_producer_sample.c
_function_: log_producer_post_logs
_level_: LOG_PRODUCER_LEVEL_WARN
_line_: 248
_thread_: 40978304
LogHub: Real-time log collection and consumption
Search/Analytics: Query and real-time analysis
Interconnection: Grafana and JDBC/SQL92
Visualized: dashboard and report functions
```

测试结果

- X86平台结果
  - C Producer可以轻松到达90 MB/s的发送速度，每秒上传日志20万，占用CPU只有70%，内存140 MB。
  - 服务器在200条/s，发送数据对于CPU基本无影响（降低到0.01%以内）。
  - 客户线程发送一条数据（输出一条日志）的平均耗时为1.2 us。



- 树莓派平台结果
  - 在树莓派的测试中，由于CPU的频率只有600 MHz，性能差不多是服务器的1/10左右，每秒可发送最多2万条日志。
  - 树莓派在20条/s的时候，发送数据对于CPU基本无影响（降低到0.01%以内）。
  - 客户线程发送一条数据（输出一条日志）的平均耗时为：12 us左右（树莓派通过USB连接到PC共享网络）。



8.2. 采集-通过WebTracking采集日志

本文档为您介绍如何通过WebTracking采集日志数据到日志服务中，并对采集到的日志数据进行查询和分析。

背景信息

当发送重要邮件时为了确认对方已读，都会在邮件中设置**读取回执**标签，可以在对方已读时收到提醒信息。读取回执这种模式用途很广，例如。

- 发送传单时，确保对方已读。
- 推广网页时，多少用户做了点击。
- 移动App运营活动页面，分析用户访问情况。

对这类个性化的采集与统计，针对网站与站长的传统方案都无法胜任，主要难点在于。

- 个性化需求难满足：用户产生行为并非移动端场景，其中会包括一些运营个性化需求字段，例如：来源、渠道、环境、行为等参数。
- 开发难度大/成本高：为完成一次数据采集、分析需求，首先需要购买云主机、公网IP、开发数据接收服务器、消息中间件等，并且通过互备保障服务高可用。接下来需要开发服务端并进行测试。
- 使用不易：数据达到服务端后，还需要工程师先清洗结果并导入数据库，生成运营需要的数据。
- 无法弹性：无法预估用户的使用量，因此需要预留很大的资源池。

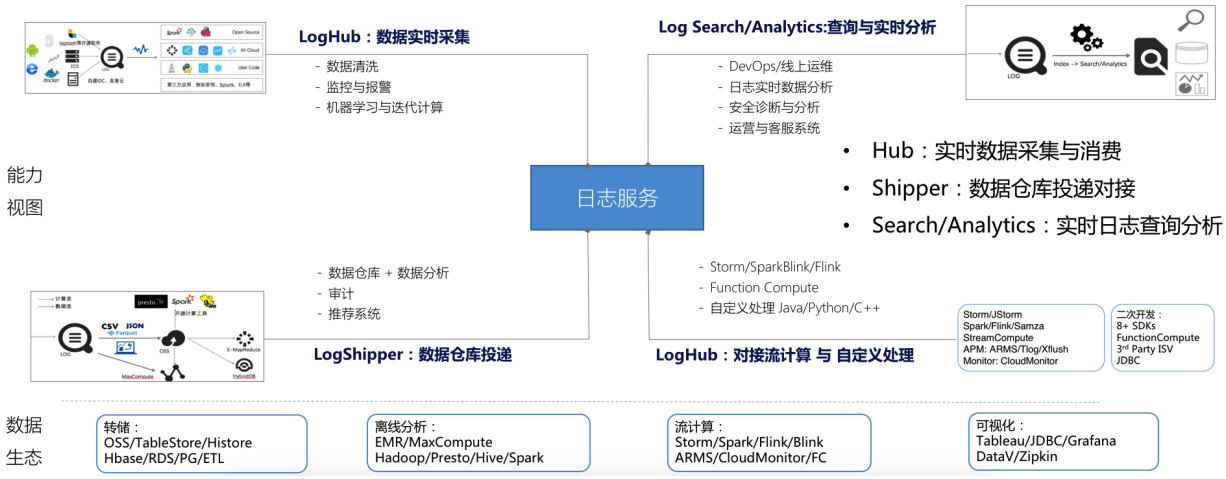
从以上几点看，当一个面向内容投放的运营需求来了后，如何能以快捷的手段满足这类用户行为采集、分析需求是一个很大的挑战。

日志服务提供Web Tracking、JS、Tracking Pixel SDK用于解决以上轻量级埋点采集场景，可以在1分钟时间内完成埋点和数据上报工作。

功能特点

这里引入采集+分析方案基于阿里云日志服务，该服务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。服务功能包括。

- LogHub：实时采集与消费。与Blink、Flink、Spark Streaming、Storm、Kepler等打通。
- 数据投递：LogShipper。与MaxCompute、E-MapReduce、OSS、Function Compute等打通。
- 查询与实时分析：LogSearch/Analytics。与DataV、Grafana、Zipkin、Tableau等打通。



采集端优势

日志服务提供30+种数据采集方式，针对服务器、移动端、嵌入式设备及各种开发语言都提供完整的解决方案。

- Logtail：针对X86服务器设计Agent。
- Android/iOS：针对移动端SDK。
- Producer Library：面向受限CPU/内存、智能设备。

本文档中介绍的轻量级采集方案（Web Tracking）只需一个HTTP Get请求即可将数据传输至日志服务Logstore端，适应各种无需任何验证的静态网页、广告投放、宣传资料和移动端数据采集。相比其他日志采集方案，特点如下。

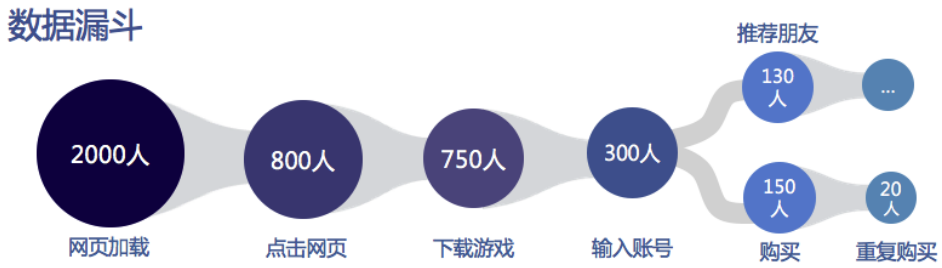
Web Tracking接入流程

Web Tracking（也叫Tracking Pixel）术语来自于HTML语法中的图片标签，即您可以在页面上嵌入一个0 Pixel图片，该图片默认对用户不可见，当访问该页面显示加载图片时，会顺带发起一个Get请求到服务端，这个时候就会把参数传给服务端。具体操作，请参见使用Web Tracking采集日志。

应用场景

当您有一个新内容时（例如新功能、新活动、新游戏、新文章），作为运营人员总是迫不及待地希望能尽快传达到用户，因为这是获取用户的第一步、也是最重要的一步。

以游戏发行为例，市场很大一笔费用进行游戏推广，例如投放了1W次广告。广告成功加载的有2000人次，约占20%。其中点击的有800人次，最终下载并注册账号试玩的往往少之又少。



可见，能够准确、实时地获得内容推广有效性对于业务非常重要。为了达到整体推广目标，运营人员往往会挑选各个渠道来进行推广。

- 用户站内信（Mail）、官网博客（Blog）、首页文案（Banner等）。
- 短信、用户Email、传单等。
- 新浪微博、钉钉用户群、微信公众账号、知乎论坛、今日头条等新媒体。

操作步骤

1. 开启Web Tracking功能。

   在日志服务中创建一个Logstore（例如叫：myclick），并开启WebTracking功能。

2. 生成Web Tracking标签。

i. 为需要宣传的文档（article=1001）面对每个宣传渠道增加一个标识，并生成Web Tracking标签（以Img标签为例）。

- 站内信渠道（mailDec）。

```
<img src="http://example.cn-hangzhou.log.aliyuncs.com/logstores/myclick/track_ua.gif?APIVersion=0.6.0&from=mailDec&article=1001" alt="" title="">
```

- 官网渠道（aliyunDoc）。

```
<img src="http://example.cn-hangzhou.log.aliyuncs.com/logstores/myclick/track_ua.gif?APIVersion=0.6.0&from=aliyundoc&article=1001" alt="" title="">
```

- 用户邮箱渠道（email）。

```
<img src="http://example.cn-hangzhou.log.aliyuncs.com/logstores/myclick/track_ua.gif?APIVersion=0.6.0&from=email&article=1001" alt="" title="">
```

其他更多渠道可以在from参数后加上，也可以在URL中加入更多需要采集的参数。

ii. 将img标签放置在宣传内容中，并进行发布。

3. 分析日志。

在完成埋点采集后，您可以使用日志服务LogSearch/Analytics功能对海量日志数据进行实时查询与分析。在结果分析可视化上，除自带Dashboard外，还支持DataV、Grafana、Tableau等对接方式。

以下是截止目前采集日志数据，您可以在搜索框中输入关键词进行查询。

也可以在查询后输入SQL进行秒级的实时分析并可视化。

i. 设计查询语句。

以下是对用户点击、阅读日志进行实时分析的语句，更多字段和分析场景可以参见分析语法。

- 当前投放总流量与阅读数。

```
* | select count(1) as c
```

- 每个小时阅读量的曲线。

```
* | select count(1) as c, date_trunc('hour',from_unixtime(__time__)) as time group by time order by time desc limit 100000
```

- 每种渠道阅读量的比例。

```
* | select count(1) as c, f group by f desc
```

- 阅读量来自哪些设备。

```
* | select count_if(ua like '%Mac%')  as mac, count_if(ua like '%Windows%')  as win, count_if(ua like '%iPhone%')  as ios, count_if(ua like '%Android%')  as android
```

- 阅读量来自哪些省市。

```
* | select ip_to_province(__source__) as province, count(1) as c group by province order by c desc limit 100
```

ii. 将这些实时数据配置到一个实时刷新Dashboard中。

> **说明** 当您看完本文时，会有一个不可见的Img标签记录本次访问，您可以在本页面源代码中查看该标签。

8.3. 采集-搭建移动端日志直传服务

在移动互联的时代，直接通过手机应用上传数据越来越普遍，对于日志场景，我们希望将手机应用的日志直接上传到日志服务中，而不需要通过应用服务端中转，这样用户就能专注在自己的业务逻辑开发。

背景信息

普通模式下，日志写入日志服务需要开启主账号的访问密钥，用于鉴权以及防篡改。移动应用如果通过此模式接入日志服务，需要将您的AK信息保存在移动端，存在AK泄漏的数据安全风险。一旦发生AK泄漏，需要升级移动应用，并替换AK，代价太大。另一种移动端日志接入日志服务的方式为通过用户服务器中转，但是该种模式下，如果移动应用数量较大，用户的应用服务器需要承载所有的移动端数据，对服务器的规模有较高的要求。

为了避免以上问题，日志服务为您提供能更安全、更便捷的移动应用日志数据采集方案，即通过RAM搭建一个基于移动服务的移动应用数据直传服务。相较于直接使用AK访问日志服务，用户不需要在移动应用端保存AK，不存在AK泄露的风险。使用临时Token访问，更加安全，Token有生命周期，并且可以针对Token定制更加复杂的权限控制，例如限制IP段的访问权限等。成本低，用户不需要准备很多服务器，移动应用直联云平台，只有控制流走用户自己的应用服务器。

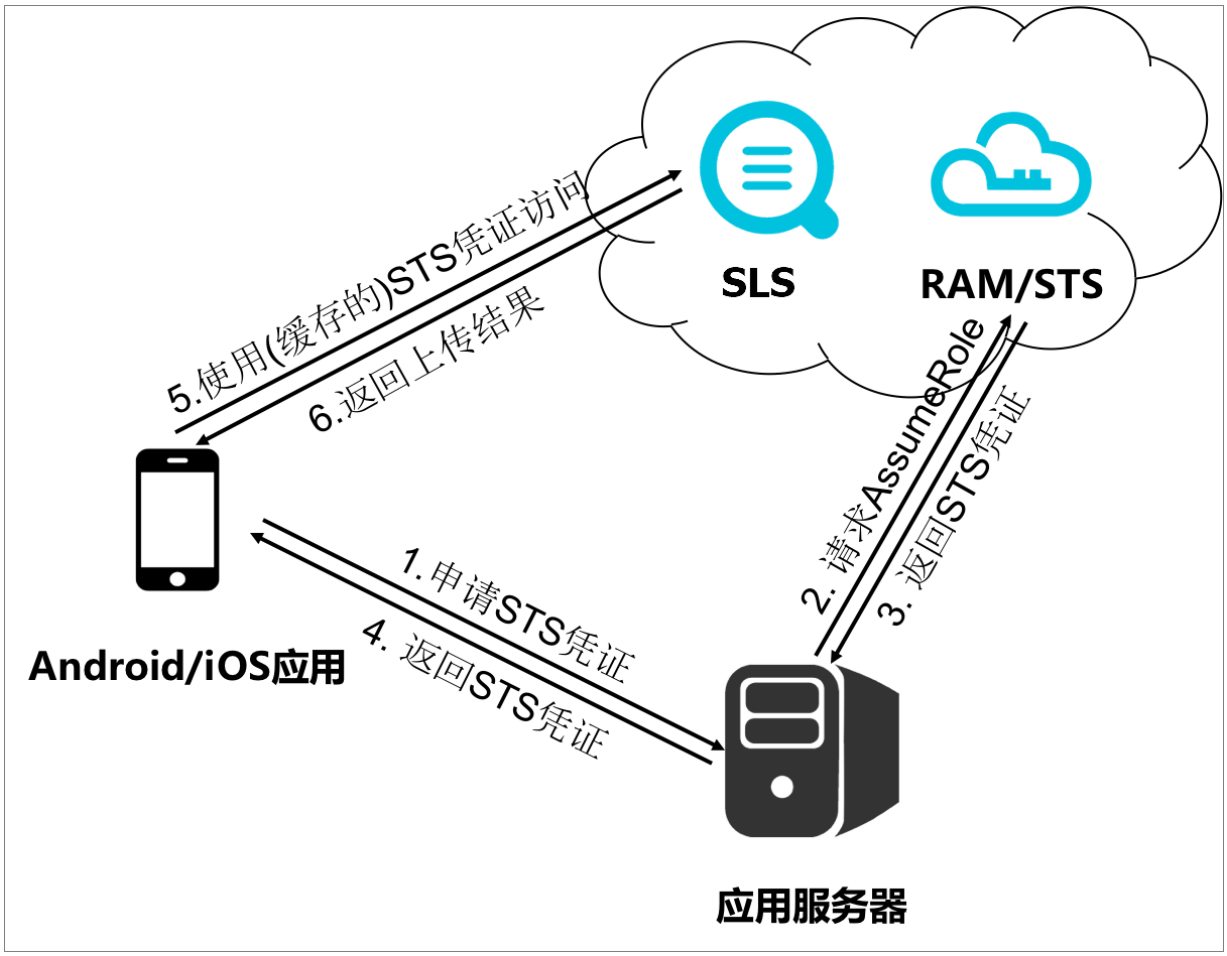
您可以创建日志服务的RAM用户角色，并配置移动应用作为RAM子用户扮演该角色，从而做到在30分钟内搭建一个基于日志服务的移动应用日志直传服务。所谓直传就是移动应用通过直连方式访问日志服务，只有控制流走用户自己的服务器。

优势

通过RAM搭建一个基于日志服务的移动应用数据直传服务，具有以下优势：

- 访问方式更加安全。临时、灵活的赋权鉴权。
- 成本低，用户不需要准备很多服务器。移动应用直联云平台，只有控制流走用户自己的应用服务器。
- 高并发，支持海量用户。日志服务有海量的上传和下载带宽。
- 弹性。日志服务有无限扩容的存储空间。

架构图如下所示



架构说明

节点	说明
Android/iOS 移动应用	最终用户手机上的应用，日志的来源。
LOG	即阿里云日志服务，负责存储应用上传的日志数据。
RAM/STS	阿里云访问控制云产品，提供用户身份管理和资源访问控制服务。负责生成临时上传凭证。
用户应用服务器	即提供该Android/iOS应用的开发者开发的手机应用后台服务，管理应用上传和下载的Token，以及用户在应用上传数据的元数据信息。

配置流程

1. 应用向用户的应用服务器申请一个临时访问凭证。

   Android/iOS应用不能直接存储AccessKeyID/AccessKeySecret，这样会存在泄密的风险。所以应用必须向用户的应用服务器申请一个临时上传凭证（下文将此临时上传凭证称为Token）。这个Token是有时效性的，如果这个Token的过期时间是30分钟（这个时间可以由应用服务器指定），那么在这30分钟里面，该Android/iOS应用可以使用这个Token访问日志服务， 30分钟后再重新获取。

2. 用户的应用服务器检测上述请求的合法性，然后返回Token给应用。
3. 手机拿到这个Token后就可以访问日志服务了。

本文档主要介绍应用服务器如何向RAM服务申请这个Token，和Android/iOS应用如何获取Token。

操作步骤

1. 授权用户角色操作日志服务。

   创建日志服务的RAM用户角色，并配置移动应用作为RAM子用户扮演该角色，详细步骤请参见创建可信实体为阿里云账号的RAM角色及授权。

   配置完成后，您将获取以下参数。

   - RAM子用户的accessKeyId、accessKey。
   - 角色的资源路径RoleArn。

2. 搭建一个应用服务器。

   为了方便开发，本教程提供了多个语言的版本示例程序供您下载，下载地址见文章最底部。

   每个语言包下载下来后，都会有一个配置文件config.json如下所示：

   ```
   {
       "AccessKeyID" : "",
       "AccessKeySecret" : "",
       "RoleArn" : "",
       "TokenExpireTime" : "900",
       "PolicyFile": "policy/write_policy.txt"
   }
   ```

   > **说明**
   >
   > i. AccessKeyID：填写您的访问密钥ID。
   >
   > ii. AccessKeySecret：填写您的访问密钥Secret。
   >
   > iii. RoleArn：填写用户角色的RoleArn。
   >
   > iv. TokenExpireTime：指Android/iOS应用取到这个Token的失效时间。注意，最少是900s，默认值可以不修改。
   >
   > v. PolicyFile：填写的是该Token所要拥有的权限列表的文件， 默认值可以不修改。

   本文档提供了两种最常用Token权限文件，位于policy目录下面。

   - write_policy.txt：指定了该Token拥有该账号下Project的写入权限。
   - readonly_policy.txt：指定了该Token拥有该账号下Project的读取权限。

   您也可以根据自己的需求设计自己的policy文件。

   返回的数据格式：

```
//正确返回
{
    "StatusCode":200,
    "AccessKeyId":"STS.3p***dgagdasdg",
    "AccessKeySecret":"rpnwO9***tGdrddgsR2YrTtI",
   "SecurityToken":"CAES+wMIARKAAZhjH0EUOIhJMQBMjRywXq7MQ/cjLYg80Aho1ek0Jm63XMhr9Oc5s'∂'∂3qaPer8p1YaX1NTDiCFZWFkvlHf1p
QhuxfKBc+mRR9KAbHUefqH+rdjZqjTF7p2m1wJXP8S6k+G2MpHrUe6TYBkJ43GhhTVFMuM3BZajY3VjZWOXBIODRIR1FKZjIiEjMzMzE0MjY0NzM5MTE4N
jkxMSoLY2xpZGSSDgSDGAGESGTETqOio6c2RrLWRlbW8vKgoUYWNzOm9zczoqOio6c2RrLWRlbW9KEDExNDg5MzAxMDcyNDY4MThSBTI2ODQyWg9Bc3N1b
WVkUm9sZVVzZXJgAGoSMzMzMTQyNjQ3MzkxMTg2OTExcglzZGstZGVtbzI=",
   "Expiration":"2017-11-12T07:49:09Z",
}
//错误返回
{
    "StatusCode":500,
    "ErrorCode":"InvalidAccessKeyId.NotFound",
    "ErrorMessage":"Specified access key is not found."
}
```

正确返回说明：（下面五个变量将构成一个Token）

状态码	说明
StatusCode	表示获取Token的状态，获取成功时，返回值是200。
AccessKeyId	表示Android/iOS应用初始化LogClient获取的 AccessKeyId
AccessKeySecret	表示Android/iOS应用初始化LogClient获取AccessKeySecret。
SecurityToken	表示Android/iOS应用初始化的Token。
Expiration	表示该Token失效的时间。主要在Android SDK会自动判断是否失效，自动获取Token。

错误返回说明：

错误码	说明
StatusCode	表示获取Token的状态，获取失败时，返回值是500。
ErrorCode	表示错误原因
ErrorMessage	表示错误的具体信息描述。

代码示例的运行方法：

对于Java版本（依赖于Java 1.7+），将包下载解压后，新建一个Java工程，将依赖和代码以及配置拷贝到工程里面，运行main函数即可，程序默认会监听7080端口，等待HTTP请求，其他语言类似。

3. 移动端构造HTTP请求，从应用服务器获取Token。

   HTTP请求及回应格式如下。

```
Request URL: GET https://localhost:7080/
Response:
{
"StatusCode":"200",
"AccessKeyId":"STS.XXXXXXXXXXXXXXX",
"AccessKeySecret":"",
"SecurityToken":"",
"Expiration":"2017-11-20T08:23:15Z"
}
```

> **说明** 本文档中所有示例仅为演示服务器搭建流程，用户可以在此基础上进行自定义开发。

源码下载

应用服务器代码示例：PHP、Java、Ruby、Node.js。

8.4. 采集Zabbix数据

Zabbix作为常用的开源监控系统，提供了丰富的告警规则用于系统监控。日志服务支持将Zabbix中的监控数据采集到Logstore中。本文介绍将Zabbix数据采集到日志服务的操作步骤。

前提条件

- 已下载及安装Zabbix。具体操作，请参见下载与安装Zabbix。

  本教程中，将Zabbix安装在阿里云ECS上为例。
- 已创建Project和Logstore。具体操作，请参见创建Project和创建Logstore。

步骤一：配置数据存储路径

Zabbix会将监控数据保存在其所在的机器上，您可以根据如下步骤设置监控数据的存储路径。

1. 登录Zabbix所在服务器。
2. 打开*zabbix_server.conf*文件。

   ```
   vim /etc/zabbix/zabbix_server.conf
   ```
3. 在*zabbix_server.conf*文件中，设置数据存储路径。

   ```
   ExportDir=/tmp/
   ```
4. 重启Zabbix服务，使配置生效。

   ```
   systemctl restart zabbix-server
   ```

   配置生效后，Zabbix会在*/tmp*目录下生产文件（文件名后缀为*.ndjson*），用于保存监控数据。

步骤二：创建Logtail采集配置

1. 登录日志服务控制台。
2. 在**接入数据**区域，选择**JSON-文本日志**。
3. 选择您已创建的Project和Logstore，单击**下一步**。
4. 创建机器组。
   i. 在**ECS机器**页签中，选中Zabbix所在的ECS实例，单击**立即执行**。

      更多信息，请参见安装Logtail（ECS实例）。

      如果Zabbix是安装在自建集群或其他云厂商服务器上，需要手动安装Logtail。更多信息，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）。

   ii. 安装完成后，单击**确认安装完毕**。

   iii. 在**创建机器组**页面，输入**名称**，单击**下一步**。

      日志服务支持创建IP地址机器组和用户自定义标识机器组，详细参数说明请参见创建IP地址机器组和创建用户自定义标识机器组。
5. 选中目标机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。

   > **注意** 如果创建机器组后立刻应用，可能因为连接未生效，导致心跳为FAIL，您可单击**自动重试**。如果还未解决，请参见Logtail机器组无心跳进行排查。
6. 创建Logtail配置，单击**下一步**。

   Zabbix监控数据为JSON类型，所以推荐使用JSON模式进行数据采集。其中**日志路径**需设置为您在步骤一：配置数据存储路径中设置的数据存储路径，其他参数详情请参见使用JSON模式采集日志。

7. 预览数据及设置索引，单击**下一步**。

   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

   > **说明** 如果您要查询分析日志，那么全文索引和字段索引属性必须至少启用一种。同时启用时，以字段索引为准。

8.5. 跨阿里云账号采集日志

本文介绍跨阿里云账号采集服务器日志的操作步骤。

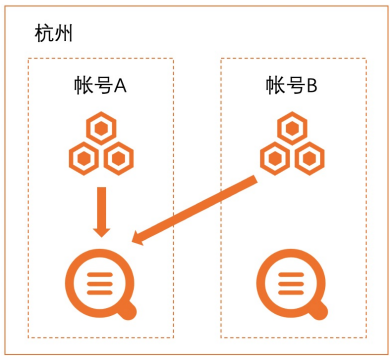
背景信息

您要通过Logtail采集服务器日志时，在服务器上安装Logtail后，还需配置日志服务所在阿里云账号ID为用户标识，表示该账号有权限通过Logtail采集该服务器日志。否则在机器组中会显示服务器心跳失败，导致Logtail无法采集日志到日志服务。

例如某电商公司拥有两个电商应用，部署在阿里云杭州地域的ECS集群上，并使用杭州地域的日志服务进行日志管理。

- 应用A部署在阿里云账号A（12****456）下的ECS集群（Linux系统）上，并使用该账号下的日志服务进行日志管理。
- 应用B部署在阿里云账号B（17****397）下的ECS集群（Linux系统）上，并使用该账号下的日志服务进行日志管理。

现公司业务调整，计划将两个应用的日志集中采集到阿里云账号A（12****456）下的日志服务中，即将两个应用的日志分别采集到同一个日志服务Project下的不同Logstore中。因此您需要新增一个Logtail采集配置、机器组和Logstore，用于采集和存储应用B相关的日志。应用A相关的日志采集保持不变（使用原有的Logtail采集配置、机器组和Logstore）。



步骤一：创建用户标识文件

1. 登录阿里云账号B下的ECS服务器。

   > **注意** 您需要在ECS集群B的每台ECS服务器中创建用户标识文件。

2. 执行如下命令创建用户标识文件。

   您需要配置阿里云账号A为用户标识，即创建阿里云账号A的同名文件。更多信息，请参见配置用户标识。

   ```
   touch /etc/ilogtail/users/12****456
   ```

步骤二：创建用户自定义标识机器组

1. 在ECS服务器上创建机器组的自定义用户标识文件。

   > **注意** 您需要在ECS集群B的每台ECS服务器中创建机器组的用户自定义标识文件。

   i. 登录阿里云账号B下的ECS服务器。

   ii. 在指定目录下创建*/etc/ilogtail/user_defined_id*文件并添加用户自定义标识。

   例如配置用户自定义标识为 `application_b` ，则在文件中输入 `application_b` ，并保存。文件路径说明，请参见创建用户自定义标识机器组。

2. 在日志服务控制台上创建机器组。

   i. 使用阿里云账号A登录日志服务控制台。

   ii. 在Project列表区域，单击目标Project。

   iii. 在左侧导航栏中，选择**资源 > 机器组**。

   iv. 选择**机器组**右侧的 > **创建机器组**。

   v. 在**创建机器组**对话框中，配置如下参数，然后单击**确定**。

   其中**用户自定义标识**需设置为您在步骤中设置的用户自定义标识。其他参数说明，请参见创建用户自定义标识机器组。

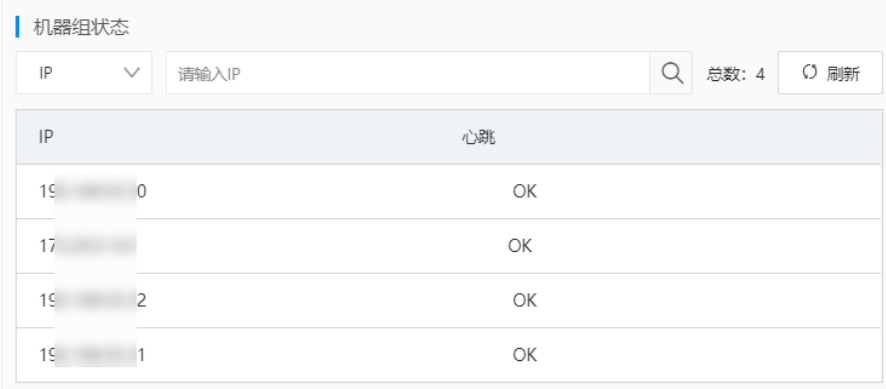
   

3. 检查机器组中的服务器心跳都为OK。

   i. 在机器组列表中，单击目标机器组。

ii. 在**机器组配置**页面，查看使用了相同用户自定义标识的ECS服务器及其心跳状态。

心跳为**OK**表示ECS服务器与日志服务的连接正常。如果显示**FAIL**请参见Logtail机器组无心跳。

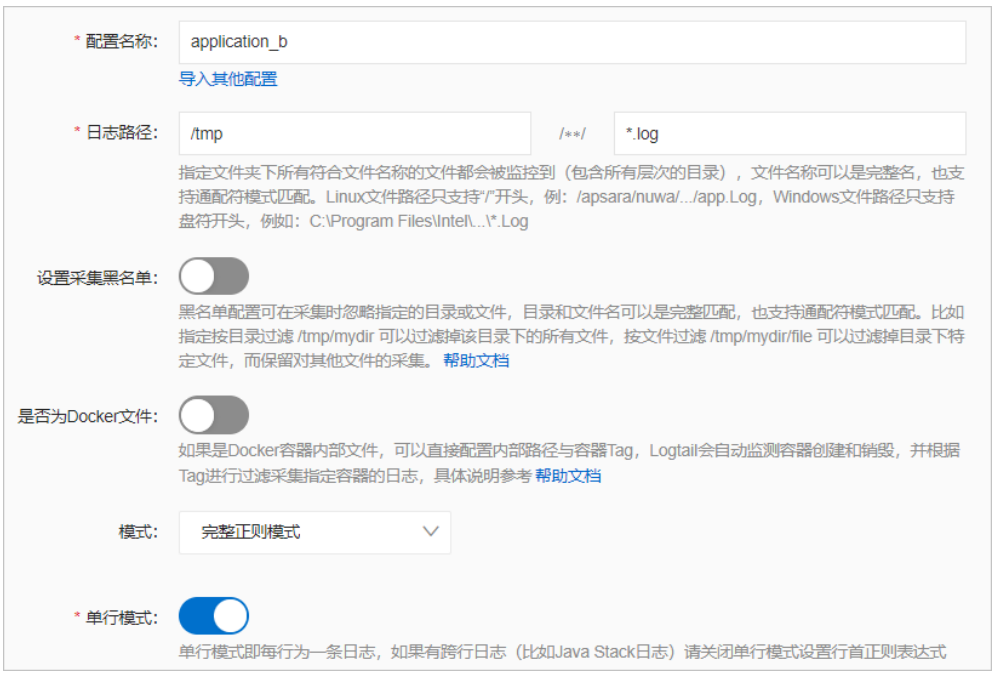


步骤三：采集日志

1. 使用阿里云账号A登录日志服务控制台。
2. 在**接入数据**区域，选择**正则-文本日志**。
3. 在**选择日志空间**向导中，选择目标Project和Logstore，单击**下一步**。
4. 在**创建机器组**向导中，单击**使用现有机器组**。
5. 在**机器组配置**向导中，选中您在步骤二中创建的机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。
6. 创建Logtail采集配置，单击**下一步**。

   具体参数说明，请参见使用完整正则模式采集日志。

> 注意
> - 默认一个文件只能匹配一个Logtail采集配置。此时账号B下的采集未停止，账号A下的Logtail采集配置无法生效，因此您需要使用如下方式使账号A下的Logtail采集配置生效。
>   - 停止账号B下的采集，即使用账号B登录日志服务控制台，从目标机器组中移除Logtail采集配置。具体操作，请参见应用Logtail采集配置。
>   - 在账号A下添加强制采集配置。更多信息，请参见如何实现文件中的日志被采集多份。
> - 此处创建Logtail采集配置成功后，请删除阿里云账号B下的原有Logtail采集配置，避免重复采集日志。如何删除，请参见删除Logtail采集配置。



7. 预览数据及设置索引，单击**下一步**。

日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

相关操作

如果您需要将阿里云账号B下的历史数据迁移到当前的Logstore中，可以在原Logstore中创建数据加工任务，将数据复制到当前Logstore中。具体操作，请参见复制Logstore数据。

> 注意
>
> 跨账号加工数据时，需使用自定义角色或密钥方式进行授权，此处以自定义角色为例。
>
> - 第一个**角色ARN**用于授予数据加工任务使用该角色来读取源Logstore中的数据。角色权限配置说明请参见授予RAM角色源Logstore读权限。
> - 第二个**角色ARN**用于授予数据加工任务使用该角色将数据加工结果写入目标Logstore。角色权限配置说明请参见授予RAM角色目标Logstore写权限（跨账号）。

8.6. 跨阿里云账号采集容器日志

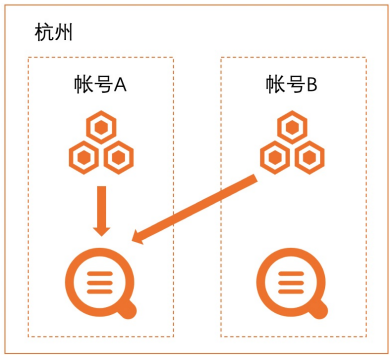
本文介绍跨阿里云账号采集阿里云Kubernetes中的容器日志的操作步骤。

背景信息

例如某电商公司拥有两个电商应用，部署在阿里云杭州地域的Kubernetes集群中，并使用杭州地域的日志服务进行日志管理。

- 应用A部署在阿里云账号A（12****456）下的Kubernetes集群中，并使用该账号下的日志服务进行日志管理。
- 应用B部署在阿里云账号B（17****397）下的Kubernetes集群中，并使用该账号下的日志服务进行日志管理。

现公司业务调整，计划将两个应用的日志集中采集到阿里云账号A（12****456）下的日志服务中，即将两个应用的日志分别采集到同一个日志服务Project下的不同Logstore中。因此您需要新增一个Logtail采集配置、机器组和Logstore，用于采集和存储应用B相关的日志。应用A相关的日志采集保持不变（使用原有的Logtail采集配置、机器组和Logstore）。



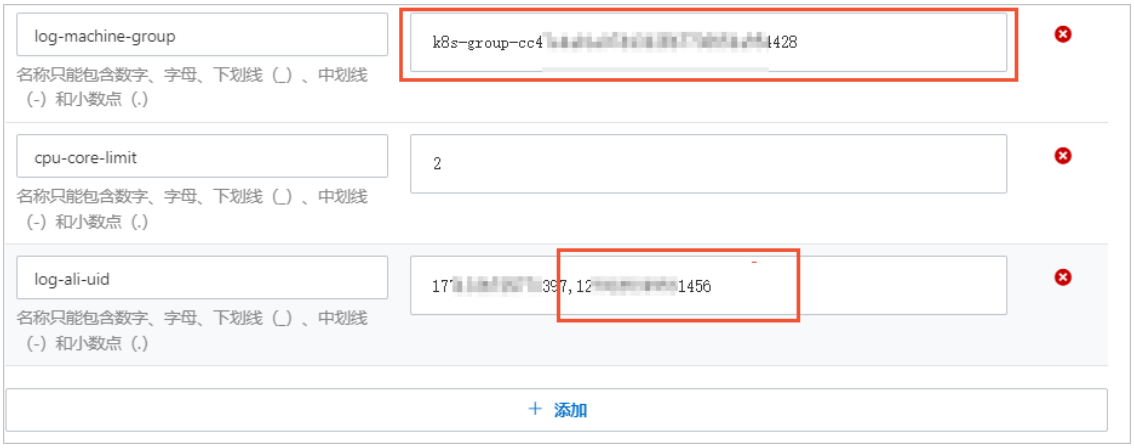
步骤一：设置阿里云账号为用户标识

1. 使用阿里云账号B登录容器服务管理控制台。
2. 设置阿里云账号A为用户标识。
   i. 在左侧导航栏中，单击**集群**。
   ii. 在**集群列表**页面中，单击目标集群。
   iii. 在左侧导航栏中，选择**配置管理 > 配置项**。
   iv. 选择**命名空间**为**kube-system**，然后在配置项列表中单击**alibaba-log-configuration**对应的**编辑**。

v. 在**编辑**面板中，完成如下操作，然后单击**确定**。

在**log-ali-uid**配置项中增加阿里云账号A的ID，然后记录**log-machine-group**配置项的值（例如k8s-group-cc47****54428），在创建机器组时需设置**用户自定义标识**为该值。

多个账号之间使用半角逗号（,）相隔，例如 `17****397,12****456` 。



3. 重启logtail-ds，使配置生效。

i. 在左侧导航栏中，选择**工作负载 > 进程守护集**。

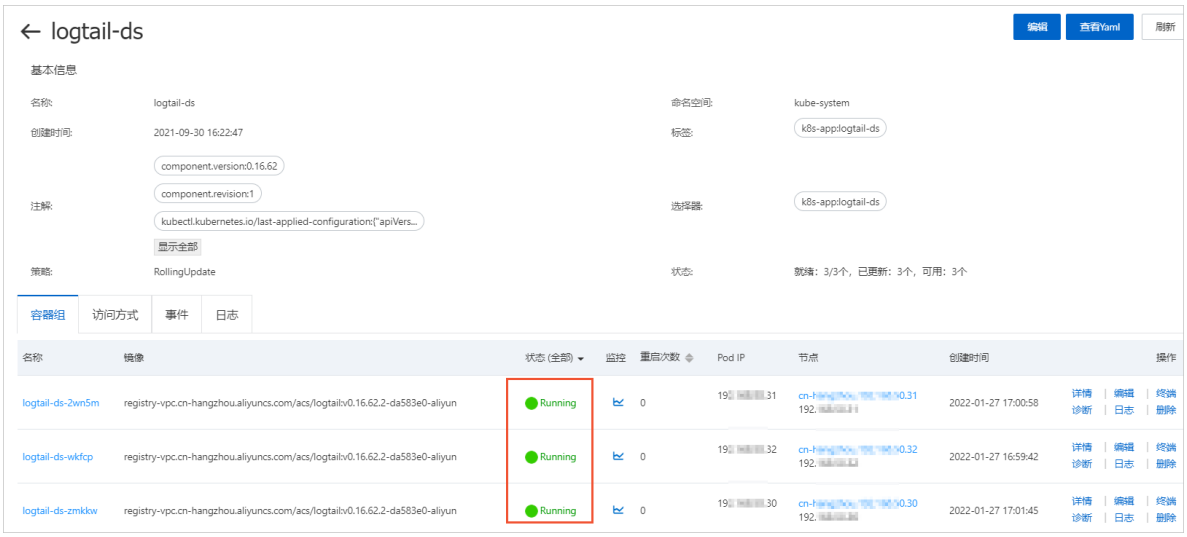
ii. 在守护进程集列表中，单击**logtail-ds**对应的**编辑**。

iii. 在**环境变量**区域，单击**新增**。

iv. 新增一个任意内容的自定义变量（例如random_id：43915743165147190534 9）。

v. 单击**更新**。

在**logtail-ds**详情页面，确认各个容器组的状态为**Running**且创建时间为您更新配置后的时间。



步骤二：创建机器组

1. 使用阿里云账号A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，选择**资源 > 机器组**。
4. 选择**机器组**右侧的 > **创建机器组**。
5. 在**创建机器组**对话框中，配置如下参数，然后单击**确定**。

其中**用户自定义标识**需设置为您在步骤一：设置阿里云账号为用户标识中获取的机器组标识（例如k8s-group-cc47****54428）。其他参数说明，请参见创建用户自定义标识机器组。

6. 检查机器组中的服务器心跳都为OK。
   i. 在机器组列表中，单击目标机器组。
   ii. 在**机器组配置**页面，查看容器节点（ECS）的心跳状态。
   心跳为**OK**表示容器节点与日志服务的连接正常。如果显示**FAIL**请参见Logtail机器组无心跳。

步骤三：创建Logtail采集配置

1. 使用阿里云账号A登录日志服务控制台。
2. 在**数据接入**区域，单击**Kubernetes-文件**。
3. 选择目标Project和Logstore，单击**下一步**。
4. 单击**使用现有机器组**。
5. 选中您在步骤二：创建机器组中所创建的机器组，将该机器组从**源机器组**移动到**应用机器组**，单击**下一步**。
6. 设置Logtail采集配置，单击**下一步**。
   具体参数说明，请参见通过DaemonSet-控制台方式采集容器文本日志。

   > 注意
   > - 默认一个文件只能匹配一个Logtail采集配置。此时账号B下的采集未停止，账号A下的Logtail采集配置无法生效，因此您需要使用如下方式使账号A下的Logtail采集配置生效。
   >   - 停止账号B下的采集，即使用账号B登录日志服务控制台，从目标机器组中移除Logtail采集配置。具体操作，请参见应用Logtail采集配置。
   >   - 在账号A下添加强制采集配置。更多信息，请参见如何实现文件中的日志被采集多份。
   > - 此处创建Logtail采集配置成功后，请删除阿里云账号B下的原有Logtail采集配置，避免重复采集日志。如何删除，请参见删除Logtail采集配置。

7. 预览数据及设置索引，单击**下一步**。
   日志服务默认开启全文索引。您也可以根据采集到的日志，手动或者自动设置字段索引。更多信息，请参见配置索引。

相关操作

如果您需要将阿里云账号B下的历史数据迁移到当前的Logstore中，可以在原Logstore中创建数据加工任务，将数据复制到当前Logstore中。具体操作，请参见复制Logstore数据。

注意

跨账号加工数据时，需使用自定义角色或密钥方式进行授权，此处以自定义角色为例。

- 第一个**角色ARN**用于授予数据加工任务使用该角色来读取源Logstore中的数据。角色权限配置说明请参见授予RAM角色源Logstore读权限。
- 第二个**角色ARN**用于授予数据加工任务使用该角色将数据加工结果写入目标Logstore。角色权限配置说明请参见授予RAM角色目标Logstore写权限（跨账号）。

9.常见问题

9.1. 数据采集常见问题

本文列举日志服务数据采集常见问题。

- Logtail基本问题
- Logtail采集日志失败的排查思路
- 如何使用Logtail自动诊断工具
- Logtail机器组无心跳排查思路
- ECS经典网络切换为VPC后，如何更新Logtail配置
- Logtail服务的app_info.json文件中IP地址为空
- 如何采集企业内网服务器日志
- 如何排查容器日志采集异常
- 查询本地采集状态
- 如何查看Logtail采集错误信息
- 日志服务采集数据常见的错误类型
- 如何调试正则表达式
- 如何优化正则表达式的性能
- 如何通过完整正则模式采集多种格式日志
- 如何采集K8s集群的容器日志
- SLB访问日志采集不到
- 日志采集Agent对比
- 日志服务采集功能与Kafka对比
- Windows实例安装Logtail服务日志提示异常信息

9.2. 日志管理

日志服务如何存储、管理用户的日志？

日志库（Logstore）是日志服务中的日志存储和查询的基本单元，通常用于存储一类日志数据。目前，支持在控制台或者通过API完成对日志库的增删改查操作。日志库创建完成后，用户通过API或SDK向指定日志库写入日志数据。如果用户希望收集阿里云ECS服务器的数据，日志服务提供的Logtail日志收集服务同样非常方便地收集到日志数据。

删除日志库，日志数据是否丢失？

删除日志库会导致日志数据丢失，请谨慎操作。

日志服务日志保存多长时间？可否修改这个保存时限？

日志服务有三项功能都与日志保存时间有关，分别如下：

- LogHub（日志中枢）/LogSearch（日志索引与查询）：根据需求自行设置。
- LogShipper（日志投递）：日志投递至OSS、MaxCompute后，生命周期在以上产品中设置。

希望把日志最终存储到OSS，怎样节省在日志服务上的花费？

日志服务的索引分析提供强大功能的同时会产生一定费用，如果您的需求是将日志保存到OSS上，且没有自定义日志查询、分析等需求，可以通过以下方式削减账单费用。

注意事项

- 索引默认关闭，如您并未开启索引和分析功能，请修改Logstore数据保存时间减少数据存储费用。
- 修改关闭索引分析功能，会使得日志关键词查询、日志统计分析、Dashboard、告警等功能不可用，请谨慎操作。
- 修改Logstore数据保存时间。

  修改Logstore数据保存时间为1天。日志服务收取一定的Logstore数据存储费用，您可以选择缩减存储时间以降低消费。
- 关闭索引功能。
  i. 开启OSS投递功能，将Logstore数据准实时投递到OSS保存。
  ii. 在**Logstore列表**页，单击**查询**。

iii. 删除索引以关闭索引分析功能。

执行以上步骤后，日志服务仅收取您很低的使用LogHub功能费用，了解更多请参考按量付费。

9.3. Logtail基本问题

什么是Logtail?

Logtail是日志服务提供的一种便于日志接入的日志采集客户端。通过在您的机器上安装Logtail来监听指定的日志文件并自动把新写入到文件的日志上传到您所指定的日志库。

Logtail是否可以采集静态不变的日志文件？

Logtail基于文件系统的修改事件来监听文件的变化，并将实时产生的日志发送到日志服务。如果日志文件没有发生任何修改行为，将不会被Logtail采集。

Logtail支持哪些平台？

- Linux

  支持如下版本的Linux x86-64（64位）服务器

  - Alibaba Cloud Linux 2.1903
  - RedHat Enterprise 6、7、8
  - CentOS Linux 6、7、8
  - Debian GNU/Linux 8、9、10
  - Ubuntu 14.04、16.04、18.04、20.04
  - SUSE Linux Enterprise Server 11、12、15
  - OpenSUSE 15.1、15.2、42.3
  - 其他基于glibc 2.5及以上版本的Linux操作系统

- Windows

  Microsoft Windows Server 2008和Microsoft Windows 7支持X86和X86_64，其他版本仅支持X86_64。

  - Microsoft Windows Server 2008
  - Microsoft Windows Server 2012
  - Microsoft Windows Server 2016
  - Microsoft Windows Server 2019
  - Microsoft Windows 7
  - Microsoft Windows 10
  - Microsoft Windows Server Version 1909
  - Microsoft Windows Server Version 2004

如何安装、升级Logtail客户端？

- 安装：请参见安装Logtail（ECS实例）、安装Logtail（Linux系统）或安装Logtail（Windows系统）。
- 升级：请参见升级Logtail（Linux）或升级Logtail（Windows）。

> **说明** 正在使用中的Logtail，只能通过手动升级。

如何配置Logtail采集日志

日志服务支持通过Logtail采集文本日志和容器日志，还支持通过Logtail插件采集日志，具体操作请参见如下链接：

- 采集文本日志
- 采集容器日志
- 使用Logtail插件采集日志

Logtail如何工作？

Logtail采集原理包括监听文件、读取文件、处理日志、过滤日志、聚合日志和发送数据等过程。更多信息，请参见Logtail采集原理。

Logtail是否支持日志文件轮转？

Logtail支持日志文件轮转。例如：app.LOG文件通过日志文件轮转生成app.LOG.1、app.LOG.2等，Logtail会自动检测到日志文件轮转过程，并保证这个过程中不会丢失日志。

Logtail如何处理网络异常？

网络异常、写入Quota满时，Logtail会停止读取正在采集的日志，保持文件打开，并在稍后进行重试。

Logtail日志采集延时如何？

Logtail基于事件进行日志采集，一般会在3秒内将日志发往日志服务。

如何采集历史日志？

如果日志的时间与Logtail处理该日志的系统时间相差5分钟以上，即被认为是历史日志。Logtail默认只采集增量的日志文件，如果您需要采集历史日志文件，可使用Logtail自带的导入历史日志功能。更多信息，请参见导入历史日志文件。

修改Logtail配置后多久生效？

您在控制台上修改Logtail配置后，Logtail在3分钟之内加载新配置并生效。

如何排查Logtail采集日志问题？

Logtail采集问题排查思路如下所示。更多信息，请参见Logtail采集日志失败的排查思路。

1. 确认Logtail心跳状态为OK。
2. 确认日志文件中的日志在实时生成。
3. 确认Logtail配置中的正则表达式与日志内容相匹配。

9.4. 如何采集企业内网服务器日志？

本文以NGINX为例说明如何将企业内网服务器日志采集到日志服务。

前提条件

已创建Project和Logstore。更多信息，请参见创建Project和创建Logstore。

背景信息

如果您的多台服务器部署在企业内网中且没有公网访问权限，但您希望将这些服务器的日志采集到日志服务进行查询与分析，您可以将一台具有公网访问权限的企业内网服务器配置为网关机，通过该网关机将企业内网服务器日志采集至日志服务。

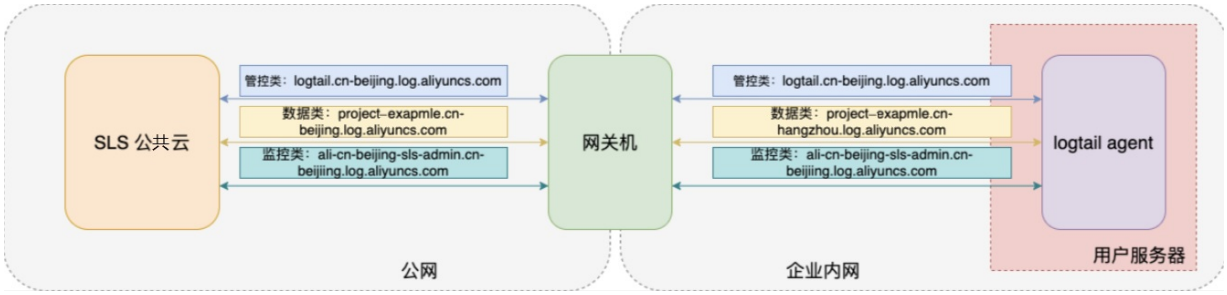
您可以选择任意一款反向代理服务器（例如NGINX）来配置网关机。NGINX是一款开源的高性能HTTP和反向代理服务器。更多信息，请参见NGINX。

工作原理

您可以通过网关机采集企业内网服务器日志到日志服务，其中涉及的Logtail和日志服务交互的域名有如下三类。

- 一类以 `logtail` 开头，格式为 `logtail.${region}.log.aliyuncs.com` ，例如 `logtail.cn-beijing.log.aliyuncs.com` 。该类域名用于负责管控类的请求交互。
- 一类以Project名称开头，格式为 `${project_name}.${region}.log.aliyuncs.com` ，例如 `project-example.cn-beijing.log.aliyuncs.com` 。该类域名用于数据类的请求交互。
- 一类以 `ali-${region}-sls-admin` 开头，格式为 `ali-${region}-sls-admin.${region}.log.aliyuncs.com` ，例如 `ali-cn-beijing-sls-admin.cn-beijing.log.aliyuncs.com` 。该类域名用于监控数据的上报。

上述域名中的 `${region}` 为目标Project所在地域， `${project_name}` 为目标Project名称。



步骤一：开通匿名写

提交工单，申请开通匿名写。

步骤二：配置网关机

使用NGINX将一台具有公网访问权限的企业内网服务器配置为网关机的操作步骤如下：

1. 登录待配置为网关机的服务器。
2. 安装NGINX。

   具体操作，请参见安装NGINX。
3. 在*nginx.conf*文件中添加以下配置。

   此处以默认的HTTP访问方式为例。其中，*${DNS服务器地址}*，请根据实际值替换。

```
server {
     listen       80;
     server_name  *.log.aliyuncs.com;
     location / {
         resolver ${DNS服务器地址};
         proxy_set_header  Host $host;
         proxy_set_header X-Forwarded-For $remote_addr;
         proxy_pass http://$host:80$request_uri;
         break;
     }
 }
```

步骤三：绑定企业内网服务器与网关机

配置网关机后，需要绑定企业内网服务器与网关机。

1. 登录某台企业内网服务器。
2. 安装Logtail。
   - Linux系统，请参见安装Logtail（Linux系统）。
   - Windows系统，请参见安装Logtail（Windows系统）。
3. 配置域名解析。

   这里以使用DNSmasq、Linux系统为例。

   i. 在*/etc/resolv.conf*文件中添加如下脚本，用于设置本机为DNS服务器。

```
nameserver 127.0.0.1
```

   ii. 在*/etc/dnsmasq.conf*文件中添加如下脚本，用于绑定网关机。

   请根据实际值，替换*${网关机IP地址}*。

```
address=/.log.aliyuncs.com/${网关机IP地址}
```

4. 重复执行步骤1~，绑定其他企业内网服务器与网关机。

步骤四：验证网络

1. 登录某台企业内网服务器。
2. 执行如下命令。

   下述命令中的*${region}*为目标Project所在地域，*${project_name}*为目标Project名称，请根据实际情况替换。

```
curl http://logtail.${region}.log.aliyuncs.com
```

```
curl http://${project_name}.${region}.log.aliyuncs.com
```

```
curl http://ali-${region}-sls-admin.${region}.log.aliyuncs.com
```

系统返回如下类似信息，表示网络正常。

```
{"Error":{"Code":"OLSInvalidMethod","Message":"The script name is invalid : /","RequestId":"62591BC7C08B7BD4AA99FCD4"}
}
```

3. 重复执行步骤~，验证其他企业内网服务器的网络。

常见问题

当采集过程中遇到问题时，您可以提交工单，联系技术支持人员获取帮助。

9.5. 如何排查容器日志采集异常

当您使用Logtail采集容器（标准容器、Kubernetes）日志时，如果采集状态异常，可以根据本文进行问题排查、运行状态检查等运维操作。

问题排查

- 排查机器组心跳是否异常
- 排查容器日志采集是否异常

排查机器组心跳是否异常

您可以通过检查机器组心跳的状态来判断容器中的Logtail是否已正确安装。

1. 查看机器组心跳状态。
   i. 登录日志服务控制台。
   ii. 在Project列表区域，单击目标Project。
   iii. 在左侧导航栏中，选择**资源 > 机器组**。
   iv. 在机器组列表中，单击目标机器组。
   v. 在**机器组配置**页面，查看机器组状态并记录心跳状态为**OK**的节点数。
2. 检查容器集群中Worker节点数。
   i. 登录容器集群。
   ii. 执行如下命令，查看集群中Worker节点数。

   ```
   kubectl get node | grep -v master
   ```

   系统将返回如下类似结果。

   ```
   NAME                                STATUS    ROLES     AGE       VERSION
   cn-hangzhou.i-bp17enxc2us3624wexh2  Ready     <none>    238d      v1.10.4
   cn-hangzhou.i-bp1ad2b02jtqd1shi2ut  Ready     <none>    220d      v1.10.4
   ```

3. 对比心跳状态为**OK**的节点数是否和容器集群中Worker节点数一致。根据对比结果选择排查方式。
   - 机器组中所有节点的心跳状态均为**Failed**。
     - 如果您要采集标准Docker容器日志，请参见参数说明，检查*${your_region_name}*、*${your_aliyun_user_id}*和*${your_machine_group_user_defined_id}*是否填写正确。
     - 如果您使用的是阿里云Kubernetes集群，请提交工单，进行排查。
     - 如果您使用的是自建Kubernetes集群，请参见参数说明，检查*{your-project-suffix}*、*{regionId}*、*{aliuid}*、*{access-key-id}*和*{access-key-secret}*是否已正确填写。

       如果填写错误，请执行 `helm del --purge alibaba-log-controller` 命令，删除安装包，然后重新安装。
   - 机器组心跳状态为OK的节点数量少于集群中的Worker节点数量。
     - 判断是否已使用YAML文件手动部署DaemonSet。
       a. 执行如下命令。如果存在返回结果，则表示您之前已使用YAML文件手动部署DaemonSet。

       ```
       kubectl get po -n kube-system -l k8s-app=logtail
       ```

       b. 下载最新版本DaemonSet模板。
       c. 根据实际值，配置*${your_region_name}*、*${your_aliyun_user_id}*、*${your_machine_group_name}*等参数。
       d. 执行如下命令，更新文件。

       ```
       kubectl apply -f ./logtail-daemonset.yaml
       ```

     - 其他情况，请提交工单。

排查容器日志采集是否异常

如果您在日志服务控制台的**预览**或**查询**页面未查到日志，则说明日志服务未采集到您的容器日志。请确认容器状态后，执行如下检查。

> **注意** 如果您要采集容器内的文本日志，需注意：
> - 如果在日志服务下发Logtail采集配置后，日志文件没有修改事件，则Logtail不采集日志。
> - 只支持采集容器默认存储或挂载到本地的文件中的日志，暂不支持其他存储方式。

1. 查看机器组心跳是否存在异常。具体操作，请参见查看机器组心跳是否存在异常。
2. 检查Logtail采集配置是否正确。

   检查Logtail采集配置中的IncludeLabel、ExcludeLabel、IncludeEnv、ExcludeEnv等配置是否符合您的采集需求。

   > **说明**
   > - 其中此处的Label为容器Label，即Docker inspect中的Label，不是Kubernetes中的Label。
   > - 您可以将IncludeLabel、ExcludeLabel、IncludeEnv和ExcludeEnv配置临时去除，查看是否可以正常采集到日志。如果可以，则说明是上述参数的配置存在问题。

其他运维操作

- 登录Logtail容器
- 查看Logtail的运行日志
- Logtail的容器标准输出说明
- 查看Kubernetes集群中日志相关组件状态
- 查看Logtail的版本号信息、IP地址、启动时间
- 误删CRD创建的Logstore后，如何处理

登录Logtail容器

- 普通Docker
  i. 在宿主机上执行如下命令，查询Logtail容器。

  ```
  docker ps | grep logtail
  ```

  系统将返回如下类似结果。

  ```
  223****6e        registry.cn-hangzhou.aliyuncs.com/log-service/logtail                              "/usr/local/ilogt
  a..."   8 days ago          Up 8 days                                 logtail-iba
  ```

  ii. 执行如下命令，登录Logtail容器。

  ```
  docker exec -it 223****6e  bash
  ```

  其中，`223****6e`为容器ID，请根据实际值替换。
- Kubernetes
  i. 执行如下命令，查询Logtail的Pod。

  ```
  kubectl get po -n kube-system | grep logtail
  ```

  系统将返回如下类似结果。

  ```
  logtail-ds-****d                                        1/1       Running    0          8d
  logtail-ds-****8                                        1/1       Running    0          8d
  ```

  ii. 执行如下命令，登录Pod。

  ```
  $kubectl exec -it -n kube-system logtail-ds-****d bash
  ```

  其中，`logtail-ds-****d`为Pod ID，请根据实际值替换。

查看Logtail的运行日志

Logtail日志存储在Logtail容器中的*/usr/local/ilogtail/*目录中，文件名为*ilogtail.LOG*和*logtail_plugin.LOG*。

1. 登录Logtail容器。具体操作，登录Logtail容器。
2. 打开*/usr/local/ilogtail/*目录。

   ```
   cd /usr/local/ilogtail
   ```

3. 查看*ilogtail.LOG*和*logtail_plugin.LOG*文件。

```
cat ilogtail.LOG
cat logtail_plugin.LOG
```

Logtail容器的标准输出（stdout）说明

Logtail容器中的标准输出并不具备参考意义，请忽略以下标准输出内容。

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: mus
t be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: mus
t be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/merged: mus
t be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

查看Kubernetes集群中日志服务相关组件的状态

执行如下命令进行查看。

```
kubectl get deploy alibaba-log-controller -n kube-system
kubectl get ds logtail-ds -n kube-system
```

查看Logtail的版本号、IP地址、启动时间

1. 登录Logtail容器。具体操作，请参见登录Logtail容器。
2. 执行如下命令，查看Logtail的版本号、IP地址、启动时间。

   相关信息存储在Logtail容器的*/usr/local/ilogtail/app_info.json*文件中。

   ```
   kubectl exec logtail-ds-****k -n kube-system cat /usr/local/ilogtail/app_info.json
   ```

   系统将返回如下类似结果。

   ```
   {
      "UUID" : "",
      "hostname" : "logtail-****k",
      "instance_id" : "0EB****_172.20.4.2_1517810940",
      "ip" : "172.20.4.2",
      "logtail_version" : "0.16.2",
      "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64",
      "update_time" : "2018-02-05 06:09:01"
   }
   ```

误删由CRD创建的Logstore后，如何处理

如果您删除了由CRD自动创建出的Logstore，则已采集的数据无法恢复，并且针对此Logstore的CRD配置会失效，您可以选择以下方案避免日志采集异常。

- 在CRD配置中使用其他Logstore，避免使用手动误删的Logstore。
- 重启*alibaba-log-controller* Pod。

  您可通过如下命令查找该Pod。

  ```
  kubectl get po -n kube-system | grep alibaba-log-controller
  ```

9.6. 如何获取容器的Label和环境变量

日志服务支持通过容器Label和环境变量指定待采集的容器。此处的Label是指Docker inspect中的容器Label，环境变量是指在容器启动中配置的环境变量，因此您需要登录容器所在的宿主机进行获取。

获取容器Label

1. 登录容器所在的宿主机（例如ECS）。

2. 执行如下命令获取容器ID。

   *orders*为容器组名称，请根据实际情况替换。

   ```
   docker ps | grep orders
   ```

   返回结果中的2ba4ebdaf503表示容器ID。

   ```
   [root@iZbp14up92567375kqxjeqZ ~]# docker ps | grep orders
   2ba4ebdaf503        43e27feaa78a                                                  "/usr/local/bin/java…"   2 months ago        Up 2 months
                       k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
   0778af9ae173        registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0     "/pause"                 2 months ago        Up 2 months
                       k8s_POD_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
   ```

3. 执行如下命令获取容器Label。

   *2ba4ebdaf503*为容器ID，请根据实际情况替换。

   ```
   docker inspect 2ba4ebdaf503
   ```

   返回结果中的Labels字段表示容器Label。

   ```
           "OnBuild": null,
           "Labels": {
               "annotation.com.aliyun.ack.hashVersion": "1.16.6",
               "annotation.io.kubernetes.container.hash": "eabe30b0",
               "annotation.io.kubernetes.container.ports": "[{\"containerPort\":80,\"protocol\":\"TCP\"}]",
               "annotation.io.kubernetes.container.restartCount": "0",
               "annotation.io.kubernetes.container.terminationMessagePath": "/dev/termination-log",
               "annotation.io.kubernetes.container.terminationMessagePolicy": "File",
               "annotation.io.kubernetes.pod.terminationGracePeriod": "30",
               "io.kubernetes.container.logpath": "/var/log/pods/victor-center_orders-7895d5f946-s6xxj_2348cd71-[illegible]fc03a9c571/orders/0.log",
               "io.kubernetes.container.name": "orders",
               "io.kubernetes.docker.type": "container",
               "io.kubernetes.pod.name": "orders-7895d5f946-s6xxj",
               "io.kubernetes.pod.namespace": "victor-center",
               "io.kubernetes.pod.uid": "2348cd71-[illegible]9c571",
               "io.kubernetes.sandbox.id": "0778af[illegible]228983eafcc0e9a274c3eef83e785f298568",
               "msd_java_build_commit": "05998875b[illegible]13e15b2e7",
               "msd_java_build_date": "2017-11-21T12:52:16+0000",
               "msd_java_build_version": "0.0.2-SNAPSHOT"
           }
       },
       "NetworkSettings": {
           "Bridge": "",
           "SandboxID": "",
           "HairpinMode": false,
   ```

获取容器环境变量

1. 登录容器所在的宿主机（例如ECS）。
2. 执行如下命令获取容器ID。

   *orders*为容器组名称，请根据实际情况替换。

   ```
   docker ps | grep orders
   ```

   返回结果中的2ba4ebdaf503表示容器ID。

   ```
   [root@iZbp14up92567375kqxjeqZ ~]# docker ps | grep orders
   2ba4ebdaf503        43e27feaa78a                                                  "/usr/local/bin/java…"   2 months ago        Up 2 months
                       k8s_orders_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
   0778af9ae173        registry-vpc.cn-hangzhou.aliyuncs.com/acs/pause-amd64:3.0     "/pause"                 2 months ago        Up 2 months
                       k8s_POD_orders-7895d5f946-s6xxj_victor-center_2348cd71-6a91-4b5f-af26-73fc03a9c571_0
   ```

3. 执行如下命令获取容器的环境变量。

   *2ba4ebdaf503*为容器ID，请根据实际情况替换。

   ```
   docker exec 2ba4ebdaf503 env
   ```

```
JAVA_OPTS=-Xms64m -Xmx128m -XX:PermSize=32m -XX:MaxPermSize=64m -XX:+UseG1GC -Djava.security.egd=file:/dev/urandom
FRONT_END_SERVICE_HOST=172.2[illegible].225
PAYMENT_PORT_80_TCP=tcp://172.[illegible].5:80
CATALOGUE_DB_SERVICE_HOST=172.[illegible].96
MONGO_SERVICE_PORT_MONGO=27017
ANTICHEATING_PORT_80_TCP_PROTO=tcp
CATALOGUE_SERVICE_PORT_CATALOGUE=80
FRONT_END_PORT_8079_TCP_PROTO=tcp
FRONT_END_PORT_8079_TCP_ADDR=172.[illegible]25
USER_PORT_80_TCP_PROTO=tcp
MONGO_PORT_27017_TCP_ADDR=172.[illegible].199
PAYMENT_PORT=tcp://172.[illegible]5:80
CARTS_PORT_80_TCP=tcp://17[illegible]5.30:80
INTEGRAL_PORT=tcp://172.21.[illegible]3:80
CATALOGUE_PORT_80_TCP=tcp://172.[illegible]3:80
USER_PORT_80_TCP=tcp://172.21.[illegible]101:80
KUBERNETES_PORT=tcp://172.21.[illegible]443
CARTS_PORT=tcp://172.[illegible].30:80
TEST_PORT_27017_TCP_ADDR=172.[illegible].60
INTEGRAL_PORT_80_TCP_ADDR=17[illegible].12.53
CATALOGUE_SERVICE_HOST=17[illegible].21.[illegible].133
SESSION_DB_PORT=tcp://172.[illegible]49:6379
INTEGRAL_SERVICE_PORT_INTEGRAL=80
RABBITMQ_PORT_5672_TCP_PORT=5672
RABBITMQ_PORT_5672_TCP_ADDR=172.[illegible].52
ORDERS_SERVICE_PORT=80
TEST_PORT_27017_TCP_PORT=27017
CATALOGUE_DB_PORT_3306_TCP=tcp://172.[illegible]6:3306
INTEGRAL_DB_SERVICE_PORT=3306
INTEGRAL_SERVICE_PORT=80
CARTS_SERVICE_HOST=172.[illegible].30
CARTS_SERVICE_PORT=80
CATALOGUE_PORT_80_TCP_PROTO=tcp
SESSION_DB_PORT_6379_TCP_PORT=6379
```

9.7. 查询本地采集状态

Logtail具备自身健康度以及日志采集进度查询的功能，便于您对于日志采集问题进行自检，同时您可基于该功能定制日志采集的状态监控。

1. 使用指南
   i. all命令
   ii. active命令
   iii. logstore命令
   iv. logfile命令
   v. history命令
2. 命令返回值
3. 功能使用场景示例
   i. 监控Logtail运行状态
   ii. 监控日志采集进度
   iii. 判断日志文件是否采集完毕
   iv. 日志采集问题排查

使用指南

确认已安装支持状态查询功能的Logtail客户端之后，在客户端输入对应命令即可查询本地采集状态。如何安装Logtail，请参见安装Logtail（Linux系统）。

在客户端输入命令`/etc/init.d/ilogtaild -h`，确认当前客户端是否支持本地采集状态查询功能。若输出`logtail insight, version`关键字则表示已安装支持此功能的Logtail。

```
/etc/init.d/ilogtaild -h
Usage: ./ilogtaild { start | stop (graceful, flush data and save checkpoints) | force-stop | status | -h for help}$
logtail insight, version : 0.1.0
commond list :
       status all [index]
             get logtail running status
       status active [--logstore | --logfile] index [project] [logstore]
             list all active logstore | logfile. if use --logfile, please add project and logstore. default --logstore
       status logstore [--format=line | json] index project logstore
             get logstore status with line or json style. default --format=line
       status logfile [--format=line | json] index project logstore fileFullPath
             get log file status with line or json style. default --format=line
       status history beginIndex endIndex  project logstore [fileFullPath]
             query logstore | logfile history status.
index :   from 1 to 60. in all, it means last $(index) minutes; in active/logstore/logfile/history, it means last $(index)
*10 minutes
```

Logtail目前支持的查询命令、命令功能、可查询的时间区间以及结果统计的时间窗口如下：

命令	功能	可查询时间区间	统计窗口
all	查询Logtail的运行状态	最近60分钟	1分钟
active	查询当前活跃（有数据采集）的Logstore或日志文件	最近600分钟	10分钟
logstore	查询Logstore的采集状态	最近600分钟	10分钟
logfile	查询日志文件的采集状态	最近600分钟	10分钟
history	查询Logstore或日志文件一段时间内的采集状态	最近600分钟	10分钟

说明

- 命令中的 `index` 参数代表查询的时间窗口索引值，有效值为1~60，从当前时间开始计算。若统计窗口是1分钟，则查询距当前 `(index, index-1]` 分钟内的窗口；若统计窗口是10分钟，则查询距当前 `(10*index, 10*(index-1)]` 分钟内的统计窗口。
- 所有查询命令均属于status子命令，因此主命令为status。

all命令

命令格式

```
/etc/init.d/ilogtaild status all [ index ]
```

说明 all命令用来查看Logtail的运行状态。index为可选参数，不输入时默认代表1。

示例

```
/etc/init.d/ilogtaild status all 1
ok
/etc/init.d/ilogtaild status all 10
busy
```

输出信息描述

项目	描述	紧急度	解决方法
ok	当前状态正常。	无	无需处理。
busy	当前采集速度较高，Logtail状态正常。	无	无需处理。
many_log_files	当前正在采集的日志数较多。	低	检查配置中是否包含无需采集的文件。

项目	描述	紧急度	解决方法
process_block	当前日志解析出现阻塞。	低	检查日志产生速度是否过高，若一直出现，按需修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
send_block	当前发送出现阻塞。	较高	检查日志产生速度是否过高以及网络状态是否正常，若一直出现，按需修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
send_error	日志数据上传失败。	高	上传失败。更多信息，请参见如何查看Logtail采集错误信息。

active命令

命令格式

```
/etc/init.d/ilogtaild status active [--logstore] index
 /etc/init.d/ilogtaild status active --logfile index project-name logstore-name
```

> **说明**
> - 命令 `active [--logstore] index` 用来查看当前活跃的Logstore， `--logstore` 参数可以省略，命令含义不变。
> - 命令 `active --logfile index project-name logstore-name` 用来查看某Project中Logstore下的所有活跃日志文件。
> - active命令用来逐级查看活跃的日志文件。建议您先定位当前活跃的Logstore，再定向查询该Logstore下的活跃日志文件。

示例

```
/etc/init.d/ilogtaild status active 1
sls-zc-test : release-test
sls-zc-test : release-test-ant-rpc-3
sls-zc-test : release-test-same-regex-3
/etc/init.d/ilogtaild status active --logfile 1 sls-zc-test release-test
/disk2/test/normal/access.log
```

输出信息描述

- 执行命令 `active --logstore index` ，则以 `project-name : logstore-name` 形式输出当前所有活跃Logstore；若执行命令 `active --logfile index project-name logstore-name` ，则输出活跃日志文件的完整路径。
- 若Logstore或日志文件在查询窗口期内没有日志采集活动，则不会出现在active命令的输出信息中。

logstore命令

命令格式

```
/etc/init.d/ilogtaild status logstore [--format={line|json}] index project-name logstore-name
```

> **说明**
> - 使用logstore命令查询Logstore的采集状态时，只返回采集文本文件时的采集状态。不支持查询采集容器标准输出的采集状态。
> - logstore命令指定以line或JSON形式输出指定Project和Logstore的采集状态。
> - 如果不配置 `--format=` 参数，则默认选择 `--format=line` ，按照line形式输出回显信息。 `--format` 参数需位于 `logstore` 之后。
> - 若无该Logstore或该Logstore在当前查询窗口期没有日志采集活动，则line形式输出为空，JSON下为 `null` 。

示例

```
/etc/init.d/ilogtaild status logstore 1 sls-zc-test release-test-same
time_begin_readable : 17-08-29 10:56:11
time_end_readable : 17-08-29 11:06:11
time_begin : 1503975371
time_end : 1503975971
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503975970
read_count : 687
avg_delay_bytes : 0
max_unsend_time : 0
min_unsend_time : 0
max_send_success_time : 1503975968
send_queue_size : 0
send_network_error_count : 0
send_network_quota_count : 0
send_network_discard_count : 0
send_success_count : 302
send_block_flag : false
sender_valid_flag : true
/etc/init.d/ilogtaild status logstore --format=json 1 sls-zc-test release-test-same
{
   "avg_delay_bytes" : 0,
   "config" : "##1.0##sls-zc-test$same",
   "last_read_time" : 1503975970,
   "logstore" : "release-test-same",
   "max_send_success_time" : 1503975968,
   "max_unsend_time" : 0,
   "min_unsend_time" : 0,
   "parse_fail_lines" : 0,
   "parse_success_lines" : 230615,
   "project" : "sls-zc-test",
   "read_bytes" : 65033430,
   "read_count" : 687,
   "send_block_flag" : false,
   "send_network_discard_count" : 0,
   "send_network_error_count" : 0,
   "send_network_quota_count" : 0,
   "send_queue_size" : 0,
   "send_success_count" : 302,
   "sender_valid_flag" : true,
   "status" : "ok",
   "time_begin" : 1503975371,
   "time_begin_readable" : "17-08-29 10:56:11",
   "time_end" : 1503975971,
   "time_end_readable" : "17-08-29 11:06:11"
}
```

输出信息描述

关键字	含义	单位
status	该Logstore整体状态。具体状态、含义以及修改方式见下表。	无
time_begin_readable	可读的开始时间。	无
time_end_readable	可读的截止时间。	无
time_begin	统计开始时间。	Unix时间戳，秒
time_end	统计结束时间。	Unix时间戳，秒
project	Project名。	无

关键字	含义	单位
logstore	Logstore名。	无
config	采集配置名（由 `##1.0##` + project + `$` + config组成的全局唯一配置名）。	无
read_bytes	窗口内读取日志数。	字节
parse_success_lines	窗口内日志解析成功的行数。	行
parse_fail_lines	窗口日志解析失败的行数。	行
last_read_time	窗口内最近的读取时间。	Unix时间戳，秒
read_count	窗口内日志读取次数。	次数
avg_delay_bytes	窗口内平均每次读取时当前偏移量与文件大小差值的平均值。	字节
max_unsend_time	窗口结束时发送队列中未发送数据包的最大时间，队列空时为0。	Unix时间戳，秒
min_unsend_time	窗口结束时发送队列中未发送数据包的最小时间，队列空时为0。	Unix时间戳，秒
max_send_success_time	窗口内发送成功数据的最大时间。	Unix时间戳，秒
send_queue_size	窗口结束时当前发送队列中未发送数据包数。	个
send_network_error_count	窗口内因网络错误导致发送失败的数据包个数。	个
send_network_quota_count	窗口内因quota超限导致发送失败的数据包个数。	个
send_network_discard_count	窗口内因数据异常或无权限导致丢弃数据包的个数。	个
send_success_count	窗口内发送成功的数据包个数。	个
send_block_flag	窗口结束时发送队列是否阻塞。	无
sender_valid_flag	窗口结束时该Logstore的发送标志位是否有效，true代表正常，false代表可能因为网络错误或quota错误而被禁用。	无

Logstore状态列表

状态	含义	处理方式
ok	状态正常	无需处理。
process_block	日志解析阻塞	检查日志产生速度是否过高，若一直出现，按需修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
parse_fail	日志解析失败	检查日志格式与日志采集配置是否一致。
send_block	当前发送出现阻塞	检查日志产生速度是否过高以及网络状态是否正常，若一直出现，按需修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
sender_invalid	日志数据发送异常	检查网络状态。更多信息，请参见如何查看Logtail采集错误信息。

logfile命令

命令格式

```
/etc/init.d/ilogtaild status logfile [--format={line|json}] index project-name logstore-name fileFullPath
```

> **说明**
>
> - logfile命令指定以line或JSON形式输出指定日志文件的采集状态。
> - 如果不配置 `--format=` 参数，则默认选择 `--format=line` ，按照line形式输出回显信息。
> - 若无该logfile或该logfile在当前查询窗口期没有日志采集活动，则line形式输出为空，JSON下为 `null` 。
> - `--format` 参数需位于 `logfile` 之后。
> - `filefullpath` 必须是全路径名。

示例

```
/etc/init.d/ilogtaild status logfile 1 sls-zc-test release-test-same /disk2/test/normal/access.log
time_begin_readable : 17-08-29 11:16:11
time_end_readable : 17-08-29 11:26:11
time_begin : 1503976571
time_end : 1503977171
project : sls-zc-test
logstore : release-test-same
status : ok
config : ##1.0##sls-zc-test$same
file_path : /disk2/test/normal/access.log
file_dev : 64800
file_inode : 22544456
file_size_bytes : 17154060
read_offset_bytes : 17154060
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503977170
read_count : 667
avg_delay_bytes : 0
/etc/init.d/ilogtaild status logfile --format=json 1 sls-zc-test release-test-same /disk2/test/normal/access.log
{
   "avg_delay_bytes" : 0,
   "config" : "##1.0##sls-zc-test$same",
   "file_dev" : 64800,
   "file_inode" : 22544456,
   "file_path" : "/disk2/test/normal/access.log",
   "file_size_bytes" : 17154060,
   "last_read_time" : 1503977170,
   "logstore" : "release-test-same",
   "parse_fail_lines" : 0,
   "parse_success_lines" : 230615,
   "project" : "sls-zc-test",
   "read_bytes" : 65033430,
   "read_count" : 667,
   "read_offset_bytes" : 17154060,
   "status" : "ok",
   "time_begin" : 1503976571,
   "time_begin_readable" : "17-08-29 11:16:11",
   "time_end" : 1503977171,
   "time_end_readable" : "17-08-29 11:26:11"
}
```

输出信息描述

关键字	含义	单位
status	该日志文件当前窗口期的采集状态，参见logstore命令的status。	无
time_begin_readable	可读的开始时间。	无
time_end_readable	可读的截止时间。	无
time_begin	统计开始时间。	Unix时间戳，秒
time_end	统计结束时间。	Unix时间戳，秒

关键字	含义	单位
project	Project名。	无
logstore	Logstore名。	无
file_path	该日志文件路径。	无
file_dev	该日志文件的device id。	无
file_inode	该日志文件的inode。	无
file_size_bytes	窗口内最近一次扫描到的该文件大小。	字节
read_offset_bytes	当前该文件解析偏移量。	字节
config	采集配置名（由 `##1.0##` + project + `$` + config组成的全局唯一配置名）。	无
read_bytes	窗口内读取日志数。	字节
parse_success_lines	窗口内日志解析成功的行数。	行
parse_fail_lines	窗口内日志解析失败的行数。	行
last_read_time	窗口内最近的读取时间。	Unix时间戳，秒
read_count	窗口内日志读取次数。	次数
avg_delay_bytes	窗口内平均每次读取时当前偏移量与文件大小差值的平均值。	字节

history命令

命令格式

```
/etc/init.d/ilogtaild status history beginIndex endIndex project-name logstore-name [fileFullPath]
```

> **说明**
>
> - history命令用来查询Logstore或日志文件一段时间内的采集状态。
> - `beginIndex`、`endIndex`分别为代码查询窗口索引的起始值和终止值，需确保`beginIndex <= endIndex`。
> - 若参数中不输入`fileFullPath`，则代码查询Logstore的采集信息；否则查询日志文件的采集信息。

示例

```
/etc/init.d/ilogtaild status history 1 3 sls-zc-test release-test-same /disk2/test/normal/access.log
        begin_time          status      read  parse_success  parse_fail      last_read_time  read_count  avg_delay    devi
ce     inode  file_size  read_offset
 17-08-29 11:26:11              ok   62.12MB         231000           0   17-08-29 11:36:11         671         0B     648
00  22544459    18.22MB      18.22MB
 17-08-29 11:16:11              ok   62.02MB         230615           0   17-08-29 11:26:10         667         0B     648
00  22544456    16.36MB      16.36MB
 17-08-29 11:06:11              ok   62.12MB         231000           0   17-08-29 11:16:11         687         0B     648
00  22544452    14.46MB      14.46MB
```

```
$/etc/init.d/ilogtaild status history 2 5 sls-zc-test release-test-same
        begin_time          status      read  parse_success  parse_fail      last_read_time  read_count  avg_delay  send_q
ueue  network_error  quota_error  discard_error  send_success  send_block  send_valid          max_unsend          min_uns
end    max_send_success
 17-08-29 11:16:11              ok   62.02MB         230615           0   17-08-29 11:26:10         667         0B
0              0            0              0           300       false        true   70-01-01 08:00:00   70-01-01 08:00:00
17-08-29 11:26:08
 17-08-29 11:06:11              ok   62.12MB         231000           0   17-08-29 11:16:11         687         0B
0              0            0              0           303       false        true   70-01-01 08:00:00   70-01-01 08:00:00
17-08-29 11:16:10
 17-08-29 10:56:11              ok   62.02MB         230615           0   17-08-29 11:06:10         687         0B
0              0            0              0           302       false        true   70-01-01 08:00:00   70-01-01 08:00:00
17-08-29 11:06:08
 17-08-29 10:46:11              ok   62.12MB         231000           0   17-08-29 10:56:11         692         0B
0              0            0              0           302       false        true   70-01-01 08:00:00   70-01-01 08:00:00
17-08-29 10:56:10
```

输出信息描述

- 该命令以列表形式输出Logstore或日志文件的历史采集信息，每个窗口期一行。
- 输出字段含义请参见 `logstore` 和 `logfile` 命令。

命令返回值

正常返回值

所有命令输入有效情况下返回值为0（包括无法查询到Logstore或日志文件），例如：

```
/etc/init.d/ilogtaild status logfile --format=json 1 error-project error-logstore /no/this/file
null
echo $?
0
/etc/init.d/ilogtaild status all
ok
echo $?
0
```

异常返回值

返回值非0时，说明发生异常，请参考以下信息。

返回值	类型	输出	问题排查
10	无效命令或缺少参数	`invalid param, use -h for help.`	输入 `-h` 查看帮助。
1	查询超过1-60的时间窗口	`invalid query interval`	输出 `-h` 查看帮助。
1	无法查询到指定时间窗口	`query fail, error: $(error)` 。更多信息，请参见errno释义。	可能原因是logtail启动时间小于查询时间跨度，其他情况请提交工单处理。
1	查询窗口时间不匹配	`no match time interval, please check logtail status`	检查Logtail是否在运行，其他情况请提交工单处理。
1	查询窗口内没有数据	`invalid profile, maybe logtail restart`	检查Logtail是否在运行，其他情况请提交工单处理。

示例

```
/etc/init.d/ilogtaild status nothiscmd
invalid param, use -h for help.
echo $?
10
/etc/init.d/ilogtaild status/all 99
invalid query interval
echo $?
1
```

功能使用场景示例

通过Logtail健康度查询可以获取Logtail当前整体状态；通过采集进度查询可以获取采集过程中的相关指标信息。用户可根据获取的这些信息实现自定义的日志采集监控。

监控Logtail运行状态

通过 `all` 命令实现Logtail运行状态监控。

实现方式：每隔一分钟定期查询Logtail当前状态，若连续5分钟状态处于 `process_block` 、 `send_block` 、 `send_error` 则触发报警。

具体报警持续时间以及监控的状态范围可根据具体场景中日志采集重要程度调整。

监控日志采集进度

通过 `logstore` 命令实现具体日志库采集进度监控。

实现方式：定期每隔10分钟调用 `logstore` 命令获取该logstore的状态信息，若 `avg_delay_bytes` 超过1MB（1024*1024）或 `status` 不为 `ok` 则触发报警。

具体 `avg_delay_bytes` 报警阈值可根据日志采集流量调整。

判断日志文件是否采集完毕

通过 `logfile` 命令判断日志文件是否采集完毕。

实现方式：日志文件已经停止写入后，定期每隔10分钟调用 `logfile` 命令获取该文件的状态信息，若该文件 `read_offset_bytes` 与 `file_size_bytes` 一致，则该日志文件已经采集完毕。

日志采集问题排查

若发现某台服务器日志采集进度延迟，可用 `history` 命令查询该服务器上相关的采集信息。

1. `send_block_flag` 为true，则说明日志采集进度延迟block在网络部分。
   - 若 `send_network_quota_count` 大于0时，需要对Logstore的Shard进行分裂。更多信息，请参见管理Shard。
   - 若 `send_network_error_count` 大于0时，需要检查网络连通性。
   - 若无相关network error，则需要调整Logtail的发送并发以及流量限制。更多信息，请参见发送并发以及流量限制。
2. 发送部分相关参数正常但 `avg_delay_bytes` 较高。
   - 可根据 `read_bytes` 计算出日志平均解析速度，判断日志产生流量是否异常。
   - 可适当调整logtail的资源使用限制。更多信息，请参见设置Logtail启动参数。
3. `parse_fail_lines` 大于0。

   检查日志采集解析配置是否能够匹配所有日志。

9.8. Logtail采集日志失败的排查思路

使用Logtail采集日志后，如果预览页面为空或查询页面无数据，您可以根据本文步骤进行排查。

操作步骤

1. 确认机器组心跳是否正常。

   在日志服务控制台上查看机器组心跳状态。具体操作，请参见查看机器组状态。
   - 如果心跳为**FAIL**，请参见Logtail机器组无心跳排查思路进行排查。
   - 如果心跳为**OK**，请进行下一步。
2. 确认是否已创建Logtail采集配置。
   - 如果未创建Logtail采集配置，请参见创建Logtail采集配置进行创建。
   - 如果已创建Logtail采集配置，请进行下一步。

   > 注意 请务必确保Logtail采集配置中设置的日志路径与目标服务器上的日志文件匹配。
3. 确认Logtail采集配置是否已应用到机器组。

   在**机器组配置**页面，查看是否已将Logtail采集配置应用到机器组。更多信息，请参见管理机器组。
   - 如果未应用到机器组，请参见应用Logtail采集配置完成操作。
   - 如果已应用到机器组，请进行下一步。
4. 查看采集错误。

   完成上述排查后，您需要确认日志文件是否实时产生新日志。Logtail只采集增量日志，如果日志文件没有更新则不会被读取，如果日志文

件有更新但未在日志服务中查询到，您可以通过以下方式诊断。

- 查看采集错误

  更多信息，请参见如何查看Logtail采集错误信息。

- 查看Logtail日志

  Logtail客户端会记录关键信息以及所有WARNING、ERROR日志。如果您要了解完整的错误，可以在以下路径中查看客户端日志。

  - Linux：*/usr/local/ilogtail/ilogtail.LOG*、*/usr/local/ilogtail/logtail_plugin.LOG*（HTTP、MySQL Binlog、MySQL查询结果等输入源的日志记录）
  - Windows x64：*C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log*
  - Windows x32：*C:\Program Files\Alibaba\Logtail\logtail_*.log*

- 确认是否存在用量超限

  如果有大日志量或者大文件量的采集需求，可能需要修改Logtail的启动参数，以达到更高的日志采集吞吐量。更多信息，请参见设置Logtail启动参数。

在完成上述检查后，如果问题仍未解决，请提交工单，并在工单中提供排查过程中发现的关键信息。

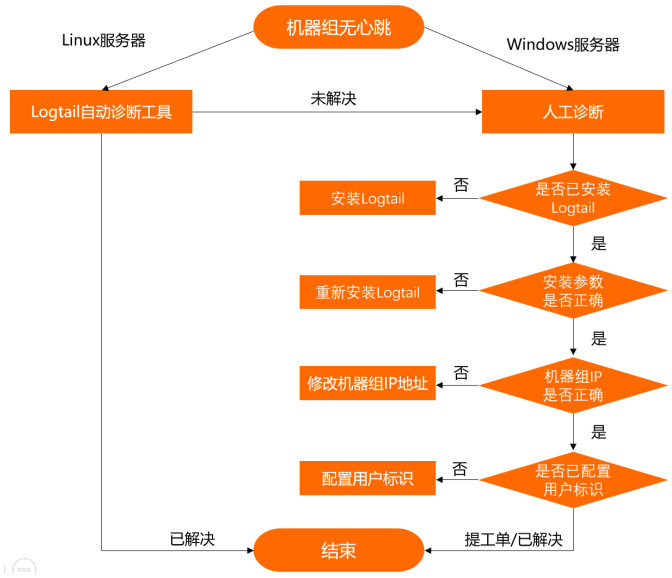
9.9. Logtail机器组无心跳排查思路

通过Logtail采集日志时，如果Logtail机器组心跳状态异常，您可使用Logtail自动诊断工具或人工诊断方式排查问题。本文主要介绍Logtail机器组无心跳的排查思路。

排查流程

使用Logtail采集日志时，在服务器上安装Logtail后，Logtail会定时向服务端发送心跳包。如果机器组无心跳，说明客户端和服务端连接失败。日志服务提供自动诊断和人工诊断两种方式，您可以根据需求选择。

- 自动诊断：日志服务提供针对Linux服务器的Logtail自动诊断工具。更多信息，请参见Logtail自动诊断工具。
- 人工诊断：Logtail自动诊断工具未检查出问题或服务器为Windows系统时，请参见本文进行人工诊断。



步骤一：检查是否已安装Logtail

通过查看Logtail状态来确定是否在目标服务器上安装Logtail。

- Linux服务器

  执行如下命令，查看Logtail状态。

  ```
  sudo /etc/init.d/ilogtaild status
  ```

  如果系统返回如下信息，表示已安装Logtail。

  ```
  ilogtail is running
  ```

- Windows服务器

  i. 打开运行窗口，输入 `services.msc` ，打开服务窗口。

  ii. 查看LogtailDaemon服务和LogtailWorker服务的运行状态。

如果正在运行，表示已安装Logtail。

根据上述方法确认是否已安装Logtail。

- 如果未安装Logtail，请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）进行安装。

  安装时，请务必按照您日志服务Project所属地域以及网络类型进行安装。关于网络类型的更多信息，请参见选择网络。
- 如果已安装Logtail，请执行下一步检查。

步骤二：检查Logtail安装参数是否正确

安装Logtail时，需要为Logtail客户端指定正确的服务端访问入口，即根据日志服务Project所在地域选择Logtail安装参数，然后根据网络类型选择不同的安装方式。如果安装参数或安装脚本错误，可能导致Logtail机器无心跳。关于不同地域的服务入口，请参见服务入口。

Logtail配置文件*ilogtail_config.json*中记录了Logtail安装参数及所选的安装方式，该文件路径说明如下：

- Linux服务器：*/usr/local/ilogtail/ilogtail_config.json*
- Windows x64服务器：*C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json*
- Windows x86服务器：*C:\Program Files\Alibaba\Logtail\ilogtail_config.json*

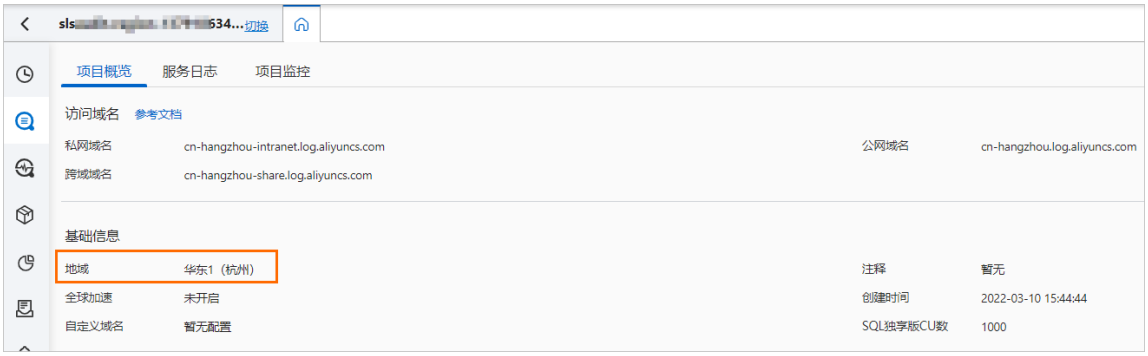
1. 检查*ilogtail_config.json*文件中客户端连接的服务入口所属地域与您Project所在地域是否一致。
   i. 在服务器上执行如下命令，查看Logtail客户端连接的服务入口的所属地域。

   ```
   cat /usr/local/ilogtail/ilogtail_config.json
   ```

   系统显示如下信息，表示Logtail被安装在华东1（杭州）地域的ECS实例中。

   ```
   [root@ ~]# cat /usr/local/ilogtail/ilogtail_config.json
   {
       "config_server_address" : "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
       "data_server_list" :
       [
           {
               "cluster" : "cn-hangzhou",
               "endpoint" : "cn-hangzhou-intranet.log.aliyuncs.com"
           }
       ],
       "cpu_usage_limit" : 0.4,
       "mem_usage_limit" : 384,
       "max_bytes_per_sec" : 20971520,
       "bytes_per_sec" : 1048576,
       "buffer_file_num" : 25,
       "buffer_file_size" : 20971520,
       "buffer_map_num" : 5,
       "streamlog_open" : false,
       "streamlog_pool_size_in_mb" : 50,
       "streamlog_rcv_size_each_call" : 1024,
       "streamlog_formats":[],
       "streamlog_tcp_port" : 11111
   }
   ```

   ii. 在日志服务控制台上查看目标Project的所属地域。

2. 检查*ilogtail_config.json*文件中配置的域名，确认是否根据服务器所属网络环境选择了正确的Logtail安装方式。

   例如*ilogtail_config.json*中记录Logtail配置的域名为 `cn-hangzhou-intranet.log.aliyuncs.com` 。

   - Linux服务器

     执行如下命令，检查网络连通性。

     ```
     curl logtail.cn-hangzhou-intranet.log.aliyuncs.com
     ```

     如果系统返回如下类似信息，表示网络连接成功。

```
{"Error":{"Code":"OLSInvalidMethod","Message":"The script name is invalid : /","RequestId":"5DD39230BE9910FC6CF17609
"}}
```

- Windows服务器

  执行如下命令，检查网络连通性。

  ```
  telnet logtail.cn-hangzhou-intranet.log.aliyuncs.com 80
  ```

  如果系统返回如下类似信息，表示网络连接成功。

  ```
  Trying 100*0*7*5...
  Connected to logtail.cn-hangzhou-intranet.log.aliyuncs.com.
  Escape character is '^]'.
  ```

- 如果检查失败，说明安装时选择了错误的参数，所以会显示执行了错误的安装命令。请参见安装Logtail（Linux系统）或安装Logtail（Windows系统）选择正确的安装参数。
- 如果Logtail已正确安装，请执行下一步检查。

步骤三：检查机器组IP地址是否正确

机器组中配置的IP地址必须和Logtail获取到的服务器IP地址一致，否则机器组无心跳或无法采集日志。Logtail获取服务器IP地址的方式如下：

- 如果没有设置主机名绑定，则获取服务器中第一块网卡的IP地址。
- 如果在*/etc/hosts*文件中设置了主机名绑定，则会获取绑定的主机名对应的IP地址。

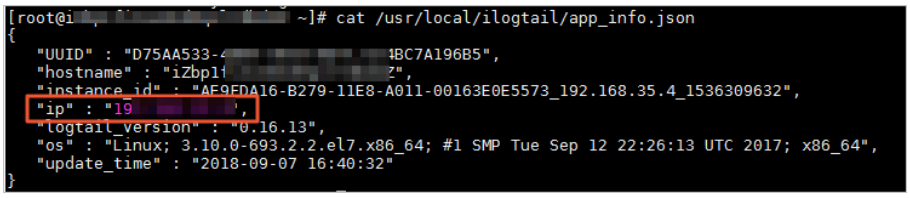
> **说明** 您可以通过hostname查看主机名。

1. 查看Logtail获取的IP地址。

   *app_info.json*文件中的ip字段中记录了Logtail获取的IP地址，该文件路径说明如下：

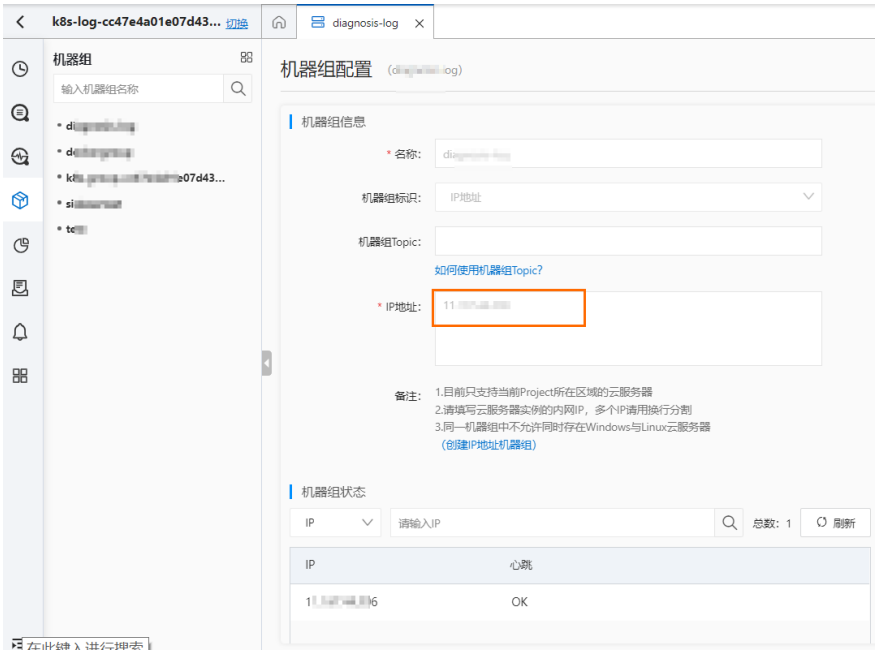
   - Linux服务器：*/usr/local/ilogtail/app_info.json*
   - Windows x64服务器：*C:\Program Files (x86)\Alibaba\Logtail\app_info.json*
   - Windows x86服务器：*C:\Program Files\Alibaba\Logtail\app_info.json*

   > **注意**
   > - 如果 `app_info.json` 文件中ip字段为空，Logtail无法工作。此时您需为服务器设置IP地址并重启Logtail。
   > - `app_info.json` 文件仅做记录，修改该文件并不会改变Logtail获取的IP地址。

2. 查看机器组中配置的IP地址。

   具体操作，请参见管理机器组。



- 如果机器组中配置的IP地址与Logtail客户端获取的IP地址不一致，则需要修改。
  - 如果机器组中配置的IP地址错误，请修改机器组中的IP地址，然后等待1分钟再查看机器组心跳状态。
  - 如果修改了服务器上的网络配置（例如修改了*/etc/hosts*文件），请重启Logtail以获取新的IP地址，并根据`app_info.json`文件中的`ip`字段修改机器组内配置的IP地址。您可以根据如下方法重启Logtail。
    - Linux服务器：

      ```
      sudo /etc/init.d/ilogtaild stop
      sudo /etc/init.d/ilogtaild start
      ```

    - Windows服务器
      a. 打开运行窗口，输入`services.msc`，打开服务窗口。
      b. 重启LogtailWorker服务。
- 如果机器组中配置的IP地址与Logtail客户端获取的IP地址一致，请执行下一步检查。

步骤四：检查是否已配置用户标识

如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，您要通过Logtail采集该服务器日志，需在服务器上安装Logtail后，手动配置日志服务所在阿里云账号ID为用户标识，表示该账号有权限通过Logtail采集该服务器日志。更多信息，请参见配置用户标识。

检查*/etc/ilogtail/users*目录下是否有与阿里云账号ID同名的文件。

- 如果有，表示已配置用户标识。
- 如果没有，请配置用户标识。具体操作，请参见配置用户标识。

> 注意 必须是阿里云账号ID。如何获取，请参见获取日志服务所在的阿里云账号ID。

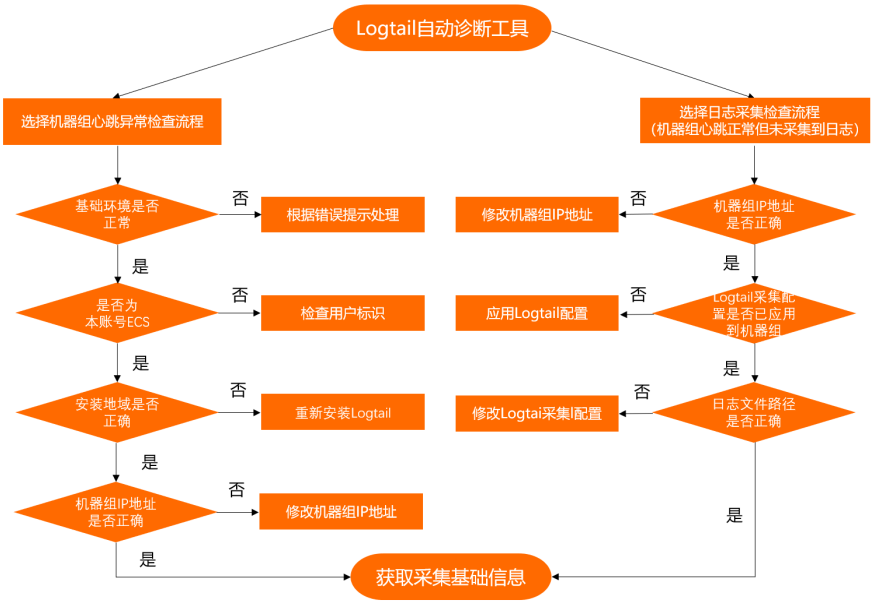
如果您的问题仍未解决，请提交工单，然后在工单中请提供您的Project、Logstore、机器组、app_info.json、ilogtail_config.json以及自助诊断工具的输出内容。

9.10. 如何使用Logtail自动诊断工具

当您使用Logtail采集日志发生异常时，可通过Logtail自助诊断工具查看Logtail客户端是否存在异常，并根据工具提示快速定位并解决问题。

> 注意 Logtail自动诊断工具仅支持Linux系统的服务器。

诊断流程



下载及运行诊断工具

1. 登录Linux服务器。
2. 下载诊断工具脚本。

   ```
   wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/checkingtool.sh -O checkingtool.sh
   ```

   如果执行以上命令无法正常下载，请执行以下命令，通过备用地址进行下载。

   ```
   wget http://logtail-corp.oss-cn-hangzhou-zmf.aliyuncs.com/linux64/checkingtool.sh -O checkingtool.sh
   ```

3. 安装curl工具。

   Logtail自动诊断工具需通过curl工具进行网络连通性检查，请确保目标服务器已安装curl工具。

4. 运行诊断工具。

   ```
   chmod 744 ./checkingtool.sh
   ./checkingtool.sh
   sh checkingtool.sh
   ```

   系统将返回如下类似信息。

   ```
   [Info]:     Logtail checking tool version : 0.3.0
   [Input]:  please choose which item you want to check :
               1. MachineGroup heartbeat fail.
               2. MachineGroup heartbeat is ok, but log files have not been collected.
   Item :
   ```

5. 根据提示输入 `1` 或 `2` ，脚本会根据您的选择执行不同检查流程。
   - `1` 表示执行机器组心跳异常检查。机器组心跳失败时，请选择此项。
   - `2` 表示执行日志采集检查。机器组心跳成功，但日志文件没有被采集时，请选择此项。

机器组心跳异常检查

选择机器组心跳异常检查后，Logtail自动诊断工具将执行下述一系列的检查。

1. 检查基础环境是否正常。
   - 是否安装Logtail。
   - 是否运行Logtail。
   - SSL状态是否正常。
   - 与日志服务之间是否有网络联通。

```
[Info]:     Logtail checking tool version : 0.3.0
[Input]:  please choose which item you want to check :
               1. MachineGroup heartbeat fail.
               2. MachineGroup heartbeat is ok, but log files have not been collected.
       Item : 1
[Info]:     Check logtail install files
[Info]:     Install file: ilogtail_config.json exists.                          [ OK ]
[Info]:     Install file: /etc/init.d/ilogtaild exists.                         [ OK ]
[Info]:     Install file: ilogtail exists.                                      [ OK ]
[Info]:     Bin file: /usr/local/ilogtail/ilogtail_0.14.2 exists.               [ OK ]
[Info]:     Logtail version :                                                   [ OK ]
[Info]:     Check logtail running status
[Info]:     Logtail is runnings.                                                [ OK ]
[Info]:     Check network status
[Info]:     Logtail is using ip: 11.XX.XX.187
[Info]:     Logtail is using UUID: 0DF18E97-0F2D-486F-B77F-XXXXXXXXXXXX
[Info]:     Check SSL status
[Info]:     SSL status OK.                                                      [ OK ]
[Info]:     Check logtail config server
[Info]:     config server address: http://config.sls.aliyun-inc.com
[Info]:     Logtail config server OK                                            [ OK ]
```

如果出现 `Error` 信息，请根据提示信息进行处理。

2. 检查您的服务器是否为当前阿里云账号下的ECS。

```
[Input]: Is your server non-Alibaba Cloud ECS or not belong to the same account with the current Project of Log Servic
e ? (y/N)
```

- 如果您的服务器是当前阿里云账号下的ECS，则输入 `N` 。
- 如果您的服务器是与日志服务属于不同账号的ECS、其他云厂商的服务器和自建IDC时，则输入 `y` 。

  输入 `y` 后，诊断工具将输出本地配置的用户标识信息。请确认其中是否包含了您的阿里云账号ID。如果未包含，请配置用户标识。具体操作，请参见配置用户标识。

  ```
  [Input]:  Is your server non-Alibaba Cloud ECS or not belong to the same account with the current Project of Log Ser
  vice ? (y/N)y
  [Info]:     Check aliyun user id(s)
  [Info]:     aliyun user id : 126XXXXXXXXX79 .                                 [ OK ]
  [Info]:     aliyun user id : 165XXXXXXXXX50 .                                 [ OK ]
  [Info]:     aliyun user id : 189XXXXXXXXX57 .                                 [ OK ]
  [Input]:  Is your project owner account ID is the above IDs ? (y/N)
  ```

3. 检查您Project所在地域是否和安装Logtail时所选地域一致。

```
[Input]: please make sure your project is in this region : { cn-hangzhou } (y/N) :
```

如果不一致，请重新安装Logtail。具体操作，请参见安装Logtail（Linux系统）。

4. 检查您机器组中配置的IP地址或用户自定义标识是否与提示信息中的一致。

```
[Input]:  please make sure your machine group's ip is same with : { 11.XX.XX.187 } or your machine group's userdefined
-id is in : { XX-XXXXX } (y/N) :
```

如果不一致，请修改机器组中的IP地址或用户自定义标识。更多信息，请参见修改机器组。

检查日志采集检查

选择日志采集检查后，Logtail自动诊断工具将执行下述一系列的检查。

1. 检查您机器组中配置的IP地址是否与提示信息中的一致。

```
[Input]:  please make sure your machine group's ip is same with : { 11.XX.XX.187 } (y/N) :
```

如果不一致，请修改机器组中的IP地址。具体操作，请参见修改机器组。

2. 检查您的Logtail采集配置是否已应用到目标机器组中。

```
[Input]: please make sure you have applied collection config to the machine group (y/N) :Y
```

如果未应用，请将您的Logtail采集配置应用到目标机器组中。具体操作，请参见应用Logtail采集配置。

3. 检查Logtail采集配置中的日志文件是否正确。

   检查时，请输入您需要检查的日志文件全路径。如果未找到匹配项，请确认配置的路径是否可以匹配目标文件。

如果配置错误，请修改Logtail采集配置，然后等待1分钟后再次执行此脚本重新检查。如何修改Logtail采集配置，请参见修改Logtail采集配置。

```
[Input]:  please input your log file's full path (eg. /var/log/nginx/access.log) :/disk2/logs/access.log
[Info]:     Check specific log file
[Info]:     Check if specific log file [ /disk2/logs/access.log ] is included by user config.
[Warning]:  Specific log file doesnt exist.                                    [ Warning ]
[Info]:     Matched config found:                                              [ OK ]
[Info]:     [Project] -> sls-zc-xxxxxx
[Info]:     [Logstore] -> release-xxxxxxx
[Info]:     [LogPath] -> /disk2/logs
[Info]:     [FilePattern] -> *.log
```

检查通过但采集依然异常

如果所有的检查全部通过，但仍然采集异常，请在脚本最后的选择中输入 `y` 并回车确认。系统将返回如下信息。

```
[Input]: please make sure all the check items above have passed. If the problem persists, please copy all the outputs and submit a ticket in the ticket system. : (y/N)y
```

请提交工单，并在工单中提供检查脚本输出的信息。

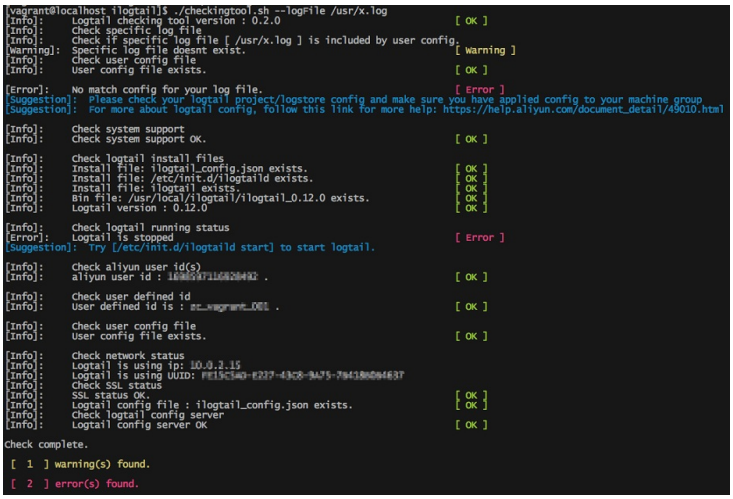
运行快速检查

运行快速检查时，您无需确认。快速检查可用于二次封装自定义检查脚本。

> **说明** 运行快速检查时，会输出您在Logtail客户端中配置的用户标识（阿里云账号ID）和机器组的自定义用户标识。如果不存在，也不会告警。如果您设置了这些信息，请查看提示信息中的内容与您设置的是否一致。如果不一致，请按照以下方法重新设置。
>
> - 配置用户标识
> - 创建用户自定义标识机器组

您可以运行 `./checkingtool.sh --logFile [LogFileFullPath]` 命令进行快速检查。 如果提示异常，请根据提示信息进行处理。

> **说明** 如果指定的日志文件检查通过且Logtail运行环境正常，建议在日志服务控制台中查看该配置项的异常日志。更多信息，请参见如何查看Logtail采集错误信息。



Logtail采集异常的常见问题

运行Logtail自动诊断工具后，可以诊断Logtail采集异常的原因，您可以根据具体原因查找对应的解决方案。常见Logtail采集问题原因及解决方案如下。

常见问题	解决方法
安装文件丢失	重装Logtail。
Logtail未运行	执行 `/etc/init.d/ilogtaild start` 命令，启动Logtail。

常见问题	解决方法
多个Logtail进程	1. 执行 `/etc/init.d/ilogtaild stop` 命令，关闭Logtail。 2. 执行 `/etc/init.d/ilogtaild start` 命令，启动Logtail。
443端口被禁用	在防火墙中，开启443端口。
无法找到配置服务器	确认是否已正确安装Linux Logtail。 如果安装错误，请重新执行安装命令。更多信息，请参见安装Logtail（Linux系统）。
不存在用户配置	确认是否已执行以下操作： 1. 已创建Logtail采集配置。 2. 机器组中包含该服务器。 3. 已经将Logtail采集配置应用到机器组。
没有匹配指定日志文件	确认是否正确创建Logtail采集配置。
指定日志文件匹配多次	匹配多次时，Logtail会随机选择一个Logtail采集配置。建议去重。

诊断工具常用参数

常用参数	说明
`--help`	查看帮助文档。
`--logFile [LogFileFullPath]`	检测Logtail是否采集 `LogFileFullPath` 路径中的日志，同时检查基本的Logtail运行环境（安装文件完整性、运行状态、阿里云账号ID、网络连通性等）。
`--logFileOnly [LogFileFullPath]`	只检测Logtail是否采集 `LogFileFullPath` 路径中的日志。
`--envOnly`	只检测Logtail运行环境。

9.11. ECS经典网络切换为VPC后，如何更新Logtail配置

您将ECS网络类型从经典网络切换为VPC后，您需要重启Logtail并更新Logtail机器组配置，才能继续采集该ECS日志。

操作步骤

1. 以管理员身份重启Logtail。
   - Linux系统

     ```
     sudo /etc/init.d/ilogtaild stop
     sudo /etc/init.d/ilogtaild start
     ```

   - Windows系统
     a. 打开运行窗口，输入 `services.msc`，打开服务窗口。
     b. 重启LogtailWorker服务。
2. 更新机器组配置。
   - IP地址机器组

     如果是IP地址机器组，则需要将机器组内的IP地址更换为重启Logtail后获取的IP地址，即*app_info.json*文件中ip字段的值。*app_info.json*文件路径：
     - Linux：*/usr/local/ilogtail/app_info.json*
     - Windows x64：*C:\Program Files (x86)\Alibaba\Logtail\app_info.json*
     - Windows x32：*C:\Program Files\Alibaba\Logtail\app_info.json*
   - 用户自定义标识机器组

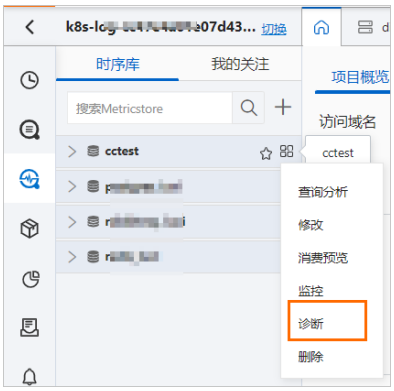
     如果是用户自定义标识机器组，则无需手动更新机器组配置。

9.12. 如何查看Logtail采集错误信息

您在使用Logtail采集日志时，可能遇到正则解析失败、文件路径不正确、流量超过Shard服务能力等错误。日志服务控制台提供诊断功能，支持诊断Logtail采集错误。

操作步骤

1. 登录日志服务控制台。
2. 在Project列表区域，单击目标Project。
3. 在**日志库**列表中，单击目标Logstore右侧的图标，然后单击**诊断**。



4. 查看日志采集错误。

   日志采集错误面板中将展示该Logstore所对应的所有Logtail采集错误列表。您可以单击目标错误代码，查看错误详情。更多信息，请参见日志服务采集数据常见的错误类型。

5. 查询目标服务器的采集错误。

   在**日志采集错误**面板中，输入服务器IP地址，可查看目标服务器的所有采集错误。

   处理问题完毕后，您可以查看是否仍有报错。历史报错在过期前仍显示，您可以忽略这部分报错，仅确认在问题处理完毕的时间点之后是否有新的错误。其中Logtail上报错误信息的时间间隔为10分钟。

> **说明** 如果需要查看所有解析失败而被丢弃的完整日志，请登录服务器查看*/usr/local/ilogtail/ilogtail.LOG*文件。

9.13. 日志服务采集数据常见的错误类型

本文介绍日志服务采集数据常见的错误类型及对应的解决方法。

如果您遇到其他问题，请提交工单处理。

错误类型	错误说明	解决方法
LOG_GROUP_WAIT_TOO_LONG_ALARM	数据包从产生到发送的过程中等待的时间较长。	检查发送是否正常，或者是否存在数据量超过默认配置、配额不足或者网络存在问题。
LOGFILE_PERMINSSION_ALARM	Logtail无权限读取指定文件。	检查服务器Logtail的启动账号，建议以root方式启动。
SPLIT_LOG_FAIL_ALARM	行首正则与日志行首匹配失败，无法对日志做分行。	检查行首正则正确性。 如果是单行日志可以配置为`.*`。

错误类型	错误说明	解决方法
MULTI_CONFIG_MATCH_ALARM	默认情况下，一个文件只能匹配一个Logtail配置。当多个Logtail配置匹配同一个文件时，只会生效一个。 说明 Docker标准输出可以被多个Logtail配置采集。	• 删除多余的logtail配置。 • 修改相关配置，实现一个文件可匹配多个Logtail配置。更多信息，请参见如何实现文件中的日志被采集多份。
REGEX_MATCH_ALARM	完整正则模式下，日志内容和正则表达式不匹配。	复制错误信息中的日志样例，并生成新的正则表达式。
PARSE_LOG_FAIL_ALARM	JSON、分隔符等模式下，由于日志格式不符合定义而解析失败。	单击错误信息，查看失败的详细报错。
CATEGORY_CONFIG_ALARM	Logtail采集配置不合法。	常见的错误为正则表达式提取文件路径作为Topic失败，其它错误请提交工单。
LOGTAIL_CRASH_ALARM	Logtail因超过服务器资源使用上限而崩溃。	修改CPU、内存使用上限。更多信息，请参见设置Logtail启动参数。
REGISTER_INOTIFY_FAIL_ALARM	在Linux系统中注册日志监听失败，可能由于没有文件夹权限或文件夹被删除。	检查Logtail是否有权限访问该文件夹，或者该文件夹是否被删除。
DISCARD_DATA_ALARM	配置Logtail使用的CPU资源不够或网络发送流控。	修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
SEND_DATA_FAIL_ALARM	• 阿里云账号未创建AccessKey。 • Logtail客户端所在机器与日志服务无法连通或者网络链路质量较差。 • 日志服务端写入配额不足。	• 使用阿里云账号创建AccessKey。 • 检查本地配置文件*/usr/local/ilogtail/ilogtail_config.json*，执行**curl <服务器地址>**，查看是否有内容返回。 • 为Logstore增加Shard数量，以支持更大数据量的写入。
REGISTER_INOTIFY_FAIL_ALARM	Logtail为日志目录注册的inotify watcher失败。	检查目录是否存在以及目录权限设置。
SEND_QUOTA_EXCEED_ALARM	日志写入流量超出限制。	在控制台上增加Shard数量。更多信息，请参见分裂Shard。
READ_LOG_DELAY_ALARM	日志采集进度落后于日志产生进度，一般是由于配置Logtail使用的CPU资源不够或是网络发送流控导致。	修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
DROP_LOG_ALARM	日志采集进度落后于日志产生进度，且未处理的日志轮转超过20个，一般是由于配置Logtail使用的CPU资源不够或是网络发送流控导致。	修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
LOGDIR_PERMINSSION_ALARM	没有日志监控目录读取权限。	检查日志监控目录是否存在。如果存在，请检查目录权限设置。
ENCODING_CONVERT_ALARM	编码转换失败。	检查日志编码格式配置是否与日志编码格式一致。
OUTDATED_LOG_ALARM	过期的日志，日志时间落后超过12小时。可能原因： • 日志解析进度落后超过12小时。 • 用户自定义时间字段配置错误。 • 日志记录程序时间输出异常。	• 查看是否存在READ_LOG_DELAY_ALARM。 如果存在，按照READ_LOG_DELAY_ALARM处理方式解决；如果不存在，请检查时间字段配置。 • 检查时间字段配置。如果时间字段配置正确，请检查日志记录程序时间输出是否正常。
STAT_LIMIT_ALARM	日志采集配置目录中的文件数超限。	检查采集的目标目录下是否有较多的文件和子目录，合理设置监控的根目录和目录最大监控深度。 您也可以修改mem_usage_limit参数。更多信息，请参见设置Logtail启动参数。
DROP_DATA_ALARM	进程退出时日志落盘到本地超时，此时会丢弃未落盘完成的日志。	该报错通常为采集严重阻塞导致。您可以修改CPU使用上限或网络发送并发限制。更多信息，请参见设置Logtail启动参数。
INPUT_COLLECT_ALARM	输入源采集异常。	根据错误提示处理。
HTTP_LOAD_ADDRESS_ALARM	HTTP数据采集配置中，设置的Addresses不合法。	检查Addresses合法性。

错误类型	错误说明	解决方法
HTTP_COLLECT_ALARM	HTTP数据采集异常。	根据错误提示排查，一般由于超时导致。
FILTER_INIT_ALARM	过滤器初始化异常。	一般由于过滤器的正则表达式非法导致，请根据提示修复。
INPUT_CANAL_ALARM	MySQL Binlog运行异常。	根据错误提示排查。 在配置更新时，canal服务可能重启，服务重启的错误可以忽略。
CANAL_INVALID_ALARM	MySQL Binlog内部状态异常。	此错误一般由于运行时表的schema信息变更导致meta不一致。请确认报错期间是否修改过表的schema。其他情况，请提交工单。
MYSQL_INIT_ALARM	MySQL初始化异常。	根据错误提示处理。
MYSQL_CHECKPOING_ALARM	MySQL Checkpoint格式异常。	确认是否修改该配置中的Checkpoint相关配置。其他情况，请提交工单。
MYSQL_TIMEOUT_ALARM	MySQL查询超时。	确认MySQL服务器和网络是否异常。
MYSQL_PARSE_ALARM	MySQL查询结果解析失败。	确认MySQL配置的Checkpoint格式是否匹配对应字段的格式。
AGGREGATOR_ADD_ALARM	向队列中添加数据失败。	由于数据发送过快。如果真实数据量很大，则可忽略。
ANCHOR_FIND_ALARM	processor_anchor插件错误、配置错误或存在不符合配置的日志。	单击错误查看详细报错，报错根据内容分为如下类型。请根据详细报错中的信息，检查相应的配置是否存在问题。 • `anchor cannot find key`：配置中指定了SourceKey但日志中不存在对应的字段。 • `anchor no start`：无法从SourceKey的值中找到Start对应的内容。 • `anchor no stop`：无法从SourceKey的值中找到Stop对应的内容。
ANCHOR_JSON_ALARM	processor_anchor插件错误，对已配置的Start和Stop所确定的内容执行JSON展开时发生错误。	单击错误查看详细报错，检查所处理的内容以及相关的配置，确定是否有配置错误或不合法日志。
CANAL_RUNTIME_ALARM	Binlog插件运行时错误。	单击错误查看详细报错，根据错误信息进行进一步地排查。一般情况下，该错误与所连接的MySQL master相关。
CHECKPOINT_INVALID_ALARM	Checkpoint解析失败。	单击错误查看详细报错，根据其中的检查点键、检查点内容（前1024个字节）以及具体的错误信息进行进一步排查。
DIR_EXCEED_LIMIT_ALARM	Logtail同时监听的目录数超出限制。	检查当前Logstore的采集配置以及该Logtail上应用的其他配置是否会包含较多的目录数，合理设置监控的根目录和目录最大监控深度。
DOCKER_FILE_MAPPING_ALARM	执行Logtail命令添加Docker文件映射失败。	单击错误查看详细报错，根据其中的命令以及具体的错误信息进行进一步排查。
DOCKER_FILE_MATCH_ALARM	无法在Docker容器中查找到指定文件。	单击错误查看详细报错，根据其中的容器信息以及查找的文件路径进行进一步排查。
DOCKER_REGEX_COMPILE_ALARM	service_docker_stdout插件错误，根据配置中的BeginLineRegex编译失败。	单击错误查看详细报错，检查其中的正则表达式是否正确。
DOCKER_STDOUT_INIT_ALARM	service_docker_stdout插件初始化失败。	单击错误查看详细报错，报错根据内容分为如下类型。 • `host...version...error`：检查配置中指定的Docker Engine是否可访问。 • `load checkpoint error`：加载检查点失败，如无影响可忽略此错误。 • `container...`：指定容器存在非法Label值，目前仅允许配置stdout和stderr。请结合详细错误进行检查。

错误类型	错误说明	解决方法
DOCKER_STDOUT_START_ALARM	service_docker_stdout插件采集时，stdout大小超过限制。	一般由于首次采集时stdout已存在，可忽略。
DOCKER_STDOUT_STAT_ALARM	service_docker_stdout插件无法检测到stdout。	一般由于容器退出时无法访问到stdout，可忽略。
FILE_READER_EXCEED_ALARM	Logtail同时打开的文件对象数量超过限制。	一般由于当前处于采集状态的文件数过多，请检查采集配置是否合理。
GEOIP_ALARM	processor_geoip插件错误。	单击错误查看详细报错，报错根据内容分为如下类型。 • `invalid ip...`：获取IP地址失败，请检查配置中的SourceKey是否正确或是否存在不合法日志。 • `parse ip...`：根据IP地址解析城市失败，请查看详细错误信息进行排查。 • `cannot find key...`：无法从日志中查看到指定的SourceKey，请检查配置是否正确或是否存在不合法日志。
HTTP_INIT_ALARM	metric_http插件错误，配置中指定的ResponseStringMatch正则表达式编译错误。	单击错误查看详细报错，检查其中的正则表达式是否正确。
HTTP_PARSE_ALARM	metric_http插件错误，获取HTTP响应失败。	单击错误查看详细报错，根据其中的具体错误信息对配置内容或所请求的HTTP服务器进行检查。
INIT_CHECKPOINT_ALARM	Binlog插件错误，加载检查点失败，插件将忽略检查点并从头开始处理。	单击错误查看详细报错，根据其中的具体错误信息来确定是否可忽略此错误。
LOAD_LOCAL_EVENT_ALARM	Logtail执行了本地事件处理。	此警告一般不会出现，如果非人为操作引起此警告，才需要进行错误排查。请单击错误查看详细报错，根据其中的文件名、配置名、project、logstore等信息进行进一步地排查。
LOG_REGEX_FIND_ALARM	processor_split_log_regex以及processor_split_log_string插件错误，无法从日志中获取到配置中指定的SplitKey。	单击错误查看详细报错，检查是否存在配置错误的情况。
LUMBER_CONNECTION_ALARM	service_lumberjack插件错误，停止插件时关闭服务器错误。	单击错误查看详细报错，根据其中的具体错误信息进行进一步排查，此错误一般可忽略。
LUMBER_LISTEN_ALARM	service_lumberjack插件错误，初始化进行监听时发生错误。	单击错误查看详细报错，报错根据内容分为如下类型。 • `init tls error...`：请结合具体的错误信息检查TLS相关的配置是否正确 • `listen init error...`：请结合具体的错误信息检查地址相关的配置是否正确。
LZ4_COMPRESS_FAIL_ALARM	Logtail执行LZ4压缩发生错误。	单击错误查看详细报错，根据其中的log lines、project、category、region等值来进行进一步排查。
MYSQL_CHECKPOINT_ALARM	MySQL插件错误，检查点相关错误。	单击错误查看详细报错，报错根据内容分为如下类型。 • `init checkpoint error...`：初始化检查点失败，请根据错误信息检查配置指定的检查点列以及所获取的值是否正确。 • `not matched checkpoint...`：检查点信息不匹配，请根据错误信息检查是否是由于配置更新等人为原因导致的错误，如果是则可忽略。
NGINX_STATUS_COLLECT_ALARM	nginx_status插件错误，获取状态发生错误。	单击错误查看详细报错，根据其中的URL以及具体的错误信息来进行进一步排查。
NGINX_STATUS_INIT_ALARM	nginx_status插件错误，初始化解析配置中指定的URL失败。	单击错误查看详细报错，根据其中的URL检查地址是否正确配置。
OPEN_FILE_LIMIT_ALARM	Logtail已打开文件数量超过限制，无法打开新的文件。	单击错误查看详细报错，根据其中的日志文件路径、Project、Logstore等信息进行进一步排查。
OPEN_LOGFILE_FAIL_ALARM	Logtail打开文件出错。	单击错误查看详细报错，根据其中的日志文件路径、Project、Logstore等信息进行进一步排查。

错误类型	错误说明	解决方法
PARSE_DOCKER_LINE_ALARM	service_docker_stdout插件错误，解析日志失败。	单击错误查看详细报错，报错根据内容分为如下类型。 • `parse docker line error: empty line`：日志为空。 • `parse json docker line error...`：以JSON格式解析日志失败，请根据错误信息以及日志的前512个字节进行排查。 • `parse cri docker line error...`：以CRI格式解析日志失败，请根据错误信息以及日志的前512个字节进行排查。
PLUGIN_ALARM	插件初始化及相关调用发生错误。	单击错误查看详细报错，报错根据内容分为如下类型，请根据具体的错误信息进行进一步排查。 • `init plugin error...`：初始化插件失败。 • `hold on error...`：暂停插件运行失败。 • `resume error...`：恢复插件运行失败。 • `start service error...`：启动 service input 类型的插件失败。 • `stop service error...`：停止 service input 类型的插件失败。
PROCESSOR_INIT_ALARM	processor_regex插件错误，编译配置中指定的Regex正则表达式失败。	单击错误查看详细报错，检查其中的正则表达式是否正确。
PROCESS_TOO_SLOW_ALARM	Logtail日志解析速度过慢。	1. 单击错误查看详细报错，根据其中的日志数量、缓冲区大小、解析时间来确定是否正常。 2. 如果不正常，检查Logtail所在节点是否有其他进程占用了过多的CPU资源或是存在效率较低的正则表达式等不合理的解析配置。
REDIS_PARSE_ADDRESS_ALARM	redis插件错误，配置中提供的ServerUrls存在解析失败的情况。	单击错误查看详细报错，对其中报错的URL进行检查。
REGEX_FIND_ALARM	processor_regex插件错误，无法从日志中找到配置中SourceKey指定的字段。	单击错误查看详细报错，检查是否存在SourceKey配置错误或日志不合法的情况。
REGEX_UNMATCHED_ALARM	processor_regex插件错误，匹配失败。	单击错误查看详细报错，报错根据内容分为如下类型，请根据具体的错误信息进行排查。 • `unmatch this log content...`：日志无法匹配配置中的正则表达式 • `match result count less...`：匹配的结果数量少于配置中指定的 Keys 数量。
SAME_CONFIG_ALARM	同一个Logstore下存在同名的配置，后发现的配置会被抛弃。	单击错误查看详细报错，根据其中的配置路径等信息排查是否存在配置错误的情况。
SPLIT_FIND_ALARM	split_char以及split_string插件错误，无法从日志中找到配置中SourceKey指定的字段。	单击错误查看详细报错，检查是否存在SourceKey配置错误或日志不合法的情况。
SPLIT_LOG_ALARM	processor_split_char以及processor_split_string插件错误，解析得到的字段数量与SplitKeys中指定的不相同。	单击错误查看详细报错，检查是否存在SourceKey配置错误或日志不合法的情况。
STAT_FILE_ALARM	通过LogFileReader对象进行文件采集时发生错误。	单击错误查看详细报错，根据其中的文件路径、错误信息进行进一步排查。
SERVICE_SYSLOG_INIT_ALARM	service_syslog插件错误，初始化失败。	单击错误查看详细报错，检查配置中的Address是否正确。

错误类型	错误说明	解决方法
SERVICE_SYSLOG_STREAM_ALARM	service_syslog插件错误，通过TCP采集时发生错误。	单击错误查看详细报错，报错根据内容分为如下类型，请根据详细报错中的具体错误信息进行排查。 • `accept error...`：执行Accept时发生错误，插件将等待一段时间后重试。 • `setKeepAlive error...`：设置 Keep Alive失败，插件将跳过此错误并继续运行。 • `connection i/o timeout...`：通过TCP读取时超时，插件将重设超时并继续读取。 • `scan error...`：TCP 读取错误，插件将等待一段时间后重试。
SERVICE_SYSLOG_PACKET_ALARM	service_syslog插件错误，通过UDP采集时发生错误。	单击错误查看详细报错，报错根据内容分为如下类型，请根据详细报错中的具体错误信息进行排查。 • `connection i/o timeout...`：通过UDP读取时超时，插件将重设超时并继续读取。 • `read from error...`：UDP读取错误，插件将等待一段时间后重试。
PARSE_TIME_FAIL_ALARM	解析日志时间失败。	您可以通过以下两种方法定位及解决问题： • 正则表达式提取的时间字段是否正确。 • 指定的时间字段内容是否匹配配置中的时间表达式。

9.14. 如何优化正则表达式的性能

通过优化正则表达式的性能，可以达到优化采集性能的目的。

关于如何优化正则表达式，为您提供以下建议：

- 使用更为精确的字符。

  不要盲目地使用 `.*` 来匹配字段，这个表达式包含了很大的搜索空间，很容易就发生误匹配、导致匹配性能下降。比如您要提取的字段只由字母组成，那么使用 `[A-Za-z]` 即可。

- 使用正确的量词。

  不盲目地使用 +,*。比如您使用 `\d` 来匹配 IP 地址，那么相比 `\d+`，`\d{1,3}` 可能会具有更高的性价比。

- 多次调试。

  调试类似于排查错误，您同样可以在网站**Regex101**上对您的正则表达式所花费的时间进行调试，一旦发现大量的回溯，可以及时优化。

9.15. 如何通过完整正则模式采集多种格式日志

完整正则模式要求日志必须采用统一的格式，但有些时候日志中可能会包含多种格式，您可以采用Schema-On-Write和Schema-On-Read两种模式处理。

以 Java 日志为例，作为一个程序日志，它一般既包含正常信息，也会包含异常栈等错误信息。

- WARNING类型的多行日志
- INFO类型的简单文本日志
- DEBUG类型的键值日志

```
[2018-10-01T10:30:31,000] [WARNING] java.lang.Exception: another exception happened
    at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
    at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
    at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
[2018-10-01T10:30:32,000] [INFO] info something
[2018-10-01T10:30:33,000] [DEBUG] key:value key2:value2
```

采集方案如下：

- Schema-On-Write：为同一份日志应用多个Logtail配置，每个Logtail配置具有不同的正则配置，从而能够正确地实现字段提取。

  > **说明** 默认情况下，一个文件只能匹配一个Logtail配置。如果文件中的日志需要被采集多份，请参见如何实现文件中的日志被采集多份。

- Schema-On-Read：使用它们共同的正则表达式来采集。

例如采用多行日志采集，将时间和日志等级作为行首正则，剩余部分为message。如果希望进一步分析message，可以为该字段建立索引，然后利用日志服务的正则提取等查询分析功能，从message字段提取需要的内容，基于该内容进行分析。

> **说明** 此方案仅推荐应用于同时分析的日志数量较小的场景下（例如千万级）。

9.16. SLB访问日志采集不到

如果发现无法正常采集SLB访问日志，请按照以下步骤排查。

1. 确认是否为SLB实例开通了访问日志

每个SLB实例需要单独设置，开通后的产生的访问日志将实时写入您的日志服务Logstore。

请登录SLB控制台， 在左侧导航栏中单击**日志管理 > 访问日志**，查看**访问日志（7层）**列表中。

- 确认列表中是否存在指定SLB实例。
- 确认SLB实例对应的**SLS日志存储**一列中记录的日志保存位置。

  此处显示的是日志保存的日志服务Project和Logstore，请在正确的位置查看是否存在SLB日志。

2. 确认RAM授权是否正确

开通访问日志功能时，系统会指引您完成RAM角色授权，成功授权后才能开通访问日志功能。如果RAM角色没有正常创建、或创建后被删除，都会导致日志采集后无法投递到日志服务Logstore。

排查方式

请登录RAM控制台，在**角色管理**页面查找是否存在AliyunLogArchiveRole。

- **如果AliyunLogArchiveRole不存在**，请使用主账号登录后并单击快速授权链接，完成授权所需要的RAM角色创建。
- **如果AliyunLogArchiveRole存在**，请单击角色名称，查看角色授权策略是否正确。

  以下是默认的授权策略，如果您的策略与默认策略不相同，可能之前修改过授权策略，请将授权策略改为默认的授权策略。

```
{
  "Version": "1",
  "Statement": [
    {
      "Action": [
        "log:PostLogStoreLogs"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

3. 确认是否产生日志事件

如果在日志服务控制台没有查看到SLB访问日志，可能是由于没有日志产生。例如：

- 当前实例未配置七层监听。

  目前只支持SLB七层监听的实例访问日志，暂不支持四层实例日志采集。常见的七层监听协议有HTTP/HTTPS等，详细说明请查看监听概述。
- 不采集开通功能之前的历史日志。

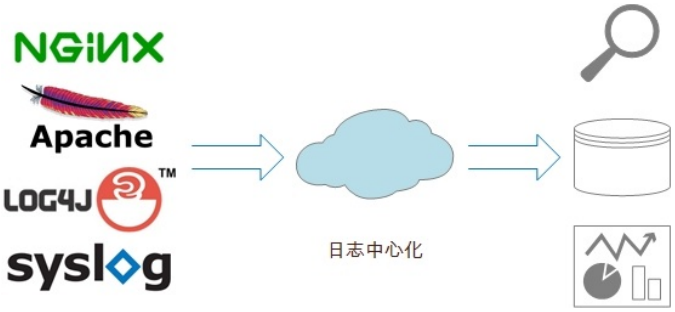
  开通SLB访问日志功能之后，从当前时间开始采集日志。
- 指定实例没有访问请求。

  必须对实例上的监听进行访问请求，才会产生访问日志。

9.17. 日志采集Agent对比

日志采集场景下客户端测评

DT时代，数以亿万计的服务器、移动终端、网络设备每天产生海量的日志。中心化的日志处理方案有效地解决了在完整生命周期内对日志的消费需求，而日志从设备采集上云是始于足下的第一步。



三款日志采集工具

- Logstash
  - 开源界ELK stack中的”L”，社区活跃，生态圈提供大量插件支持。
  - Logstash基于JRuby实现，可以跨平台运行在JVM上。
  - 模块化设计，有很强的扩展性和互操作性。
- Fluentd
  - 开源社区中流行的日志采集工具，td-agent已正式商用，由Treasure Data公司维护，是本文选用的评测版本。
  - Fluentd基于CRuby实现，并对性能表现的一些关键组件用C语言重新实现，整体性能不错。
  - Fluentd设计简洁，pipeline内数据传递可靠性高。
  - 相较于Logstash，其插件支持相对少一些。
- Logtail
  - 阿里云日志服务的生产者，经过多年阿里集团大数据场景考验。
  - 采用C++语言实现，在稳定性、资源控制、管理等方面表现较好，性能良好。
  - 相比于Logstash、Fluentd的社区支持，Logtail功能较为单一，专注日志采集功能。

功能对比

功能项	Logstash	Fluentd	Logtail
日志读取	轮询	轮询	事件触发
文件轮转	支持	支持	支持
Failover处理 （本地checkpoint）	支持	支持	支持
通用日志解析	支持grok（基于正则表达式）解析	支持正则表达式解析	支持正则表达式解析
特定日志类型	支持delimiter、key-value、json等主流格式	支持delimiter、key-value、json等主流格式	支持delimiter、key-value、json等主流格式
数据发送压缩	插件支持	插件支持	LZ4
数据过滤	支持	支持	支持
数据buffer发送	插件支持	插件支持	支持
发送异常处理	插件支持	插件支持	支持
运行环境	JRuby实现，依赖JVM环境	CRuby、C实现，依赖Ruby环境	C++实现，无特殊要求
线程支持	支持多线程	多线程受GIL限制	支持多线程
热升级	不支持	不支持	支持
中心化配置管理	不支持	不支持	支持
运行状态自检	不支持	不支持	支持cpu/内存阈值保护

日志文件采集场景-性能对比

以Nginx的access log为样例，如下一条日志365字节，结构化成14个字段：

在下面的测试中，将模拟不同的压力将该日志重复写入文件，每条日志的time字段取当前系统时间，其它13个字段相同。相比于实际场景，模拟场景在日志解析上并无差异，有一点区别是：较高的数据压缩率会减少网络写出流量。

Logstash

logstash-2.0.0版本，通过grok解析日志并写出到kafka（内置插件，开启gzip压缩）。

日志解析配置：

```
grok {
    patterns_dir=>"/home/admin/workspace/survey/logstash/patterns"
    match=>{ "message"=>"%{IPORHOST:ip} %{USERNAME:rt} - \[%{HTTPDATE:time}\] \"%{WORD:method} %{DATA:url}\" %{NUMBER:stat
us} %{NUMBER:size} \"%{DATA:ref}\" \"%{DATA:agent}\" \"%{DATA:cookie_unb}\" \"%{DATA:cookie_cookie2}\" \"%{DATA:monitor_tr
aceid}\" %{WORD:cell} %{WORD:ups} %{BASE10NUM:remote_port}" }
    remove_field=>["message"]
}
```

测试结果：

写入TPS	写入流量（KB/s）	CPU使用率（%）	内存使用（MB）
500	178.22	22.4	427
1000	356.45	46.6	431
5000	1782.23	221.1	440
10000	3564.45	483.7	450

Fluentd

td-agent-2.2.1版本，通过正则表达式解析日志并写入kafka（第三方插件fluent-plugin-kafka，开启gzip压缩）。

日志解析配置：

```
<source>
  type tail
  format /^(?<ip>\S+)\s(?<rt>\d+)\s-\s\[(?<time>[^\]]*)\]\s"(?<url>[^\"]+)"\s(?<status>\d+)\s(?<size>\d+)\s"(?<ref>[^\"]+)
"\s"(?<agent>[^\"]+)"\s"(?<cookie_unb>\d+)"\s"(?<cookie_cookie2>\w+)"\s"(?
<monitor_traceid>\w+)"\s(?<cell>\w+)\s(?<ups>\w+)\s(?<remote_port>\d+).*$/
  time_format %d/%b/%Y:%H:%M:%S %z
  path /home/admin/workspace/temp/mock_log/access.log
  pos_file /home/admin/workspace/temp/mock_log/nginx_access.pos
  tag nginx.access
</source>
```

测试结果：

写入TPS	写入流量（KB/s）	CPU使用率（%）	内存使用（MB）
500	178.22	13.5	61
1000	356.45	23.4	61
5000	1782.23	94.3	103

> **说明** 受GIL限制，Fluentd单进程最多使用1个cpu核，可以使用插件multiprocess以多进程的形式支持更大的日志吞吐。

Logtail

logtail 0.9.4版本，设置正则表达式进行日志结构化，数据LZ4压缩后以HTTP协议写到阿里云日志服务，设置batch_size为4000条。

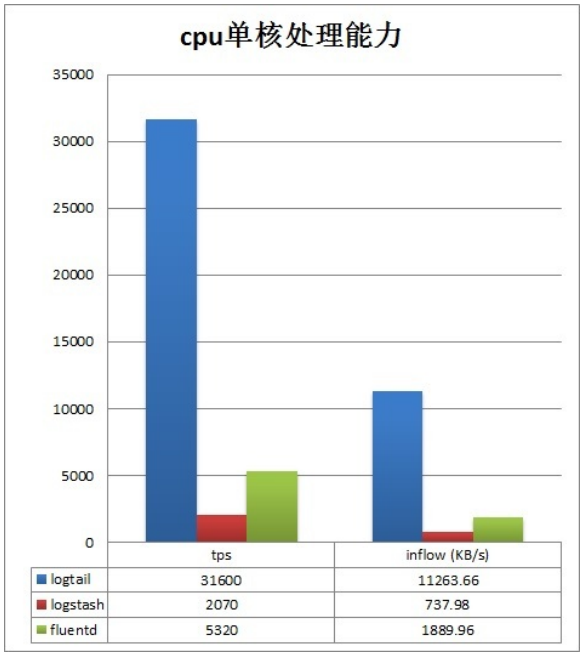
日志解析配置：

```
logRegex : (\S+)\s(\d+)\s-\s\[([^]]+)]\s"([^"]+)"\s(\d+)\s(\d+)\s"([^"]+)"\s"([^"]+)"\s(\d+)"\s"(\w+)"\s"(\w+)"\s(\w+)\s(\w+)\s(\d+).*
keys : ip,rt,time,url,status,size,ref,agent,cookie_unb,cookie_cookie2,monitor_traceid,cell,ups,remote_port
timeformat : %d/%b/%Y:%H:%M:%S
```

测试结果：

写入TPS	写入流量（KB/s）	CPU使用率（%）	内存使用（MB）
500	178.22	1.7	13
1000	356.45	3	15
5000	1782.23	15.3	23
10000	3564.45	31.6	25

单核处理能力对



总结

可以看到三款日志工具各有特点：

- Logstash支持所有主流日志类型，插件支持最丰富，可以灵活DIY，但性能较差，JVM容易导致内存使用量高。
- Fluentd支持所有主流日志类型，插件支持较多，性能表现较好。
- Logtail占用机器CPU/内存资源最少，性能吞吐量较好，针对常用日志场景支持全面，但缺少插件等机制，灵活性和可扩展性不如以上两个客户端。

9.18. 日志服务采集功能与Kafka对比

本文展示了日志服务LogHub与Kafka的功能对比。

日志服务是基于飞天Pangu构建的针对日志平台化服务。日志服务提供各种类型日志的实时采集、存储、分发及实时查询能力。通过标准化的RESTful API提供服务。日志服务LogHub提供公共的日志采集、分发通道，您如果不想搭建、运维Kafka集群，可以使用日志服务LogHub功能。

Kafka是分布式消息系统，由于其高吞吐和水平扩展，被广泛使用于消息的发布和订阅。以开源软件的方式提供，您可以根据需要搭建Kafka集群。

概念	Kafka	LogHub
存储对象	topic	logstore
水平分区	partition	shard
数据消费位置	offset	cursor

概念	Kafka	LogHub

LogHub与Kafka功能比较

功能	Kafka	LogHub
使用依赖	自建或共享Kafka集群	阿里云日志服务
通信协议	TCP网络	Http（RESTful API），且为80端口
访问控制	无	基于云账号的签名认证与访问控制
动态扩容	暂无	支持动态Shard个数弹性伸缩（Merge/Split）
多租户QoS	暂无	基于Shard的标准化流控
数据拷贝数	用户自定义	默认3份拷贝
failover/replication	调用工具完成	自动
扩容/升级	调用工具完成，影响服务	自动
写入模式	round robin/key hash	暂只支持round robin/key hash
当前消费位置	保存在Kafka集群的zookeeper	服务端维护
保存时间	配置确定	根据需求动态调整

9.19. 如何实现文件中的日志被采集多份

本文介绍文件中的日志被采集多份的解决方案。

默认情况下，一个文件只能匹配一个Logtail配置。当多个Logtail配置匹配同一个文件时，只会生效1个。因为在客户端上对文件中的日志采集多份需要消耗多倍的CPU、内存、磁盘IO和网络IO开销，将对同机部署的其他服务性能造成额外影响，并非优化的日志采集方案。

一份日志需要被存储到不同Logstore时，您可以使用日志服务数据加工功能。通过数据加工对日志进行复制，可避免对同机的其他服务性能造成影响。具体操作，请参见复制Logstore数据。

如果您的需求是必须在客户端上对文件中的日志采集多份，则可以使用如下两种方案。

- 创建目录软链接

  为文件所在的目录创建软链接。例如*/home/log/nginx/log/log.log*文件中的日志需要被采集两份，则执行如下命令创建该文件所在目录的软链接。在一个Logtail配置中使用原路径，在另一个Logtail配置中使用软链接路径。

  ```
  ln -s /home/log/nginx/log /home/log/nginx/link_log
  ```

- 添加强制采集配置

  在Logtail采集配置的**高级选项 > 扩展配置**中添加 `{"force_multiconfig": true}` ，实现目标文件可被多个Logtail配置匹配，从而实现目标文件中的日志被采集多份。