

腾讯云可观测平台 业务日志



腾讯云

【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

文档目录

业务日志

- 概述

- 日志检索

- 日志加工

 - 解析规则

 - 设置全局索引

 - 管理全局索引

- 日志接入

 - Agent 接入日志（容器环境）

 - Skwalking 探针接入日志

- 时间格式说明

业务日志 概述

最近更新时间：2024-01-05 15:08:12

应用性能监控（APM）业务日志功能目前为公测阶段，诚邀免费体验！公测结束后将会根据上报量计费，届时我们将会至少提前两周通过邮件、短信、站内信等渠道通知您。

基本概念

应用性能监控新增日志采集与分析功能，实现基于 APM 探针的一站式业务指标、链路、日志采集功能。日志拥有完整记录事故现场的能力，是定位业务异常、代码故障的重要手段。但管理多个采集 Agent、无法有效关联指标、链路等监控数据，阻碍了管理与排障效率的提升，应用性能监控现支持统一探针采集，降低探针管理复杂度的同时，自动将日志与所属应用、Trace ID、Span ID 的信息相关联，帮助您更快、更精准地从异常请求下钻至异常日志，提高故障排查效率。

说明

日志分析当前仅支持广州地域，如需支持更多地域，请 [提交工单](#) 联系我们。

核心特点

- 一站式指标、链路、日志采集。
- Span 级链路与日志关联。
- 简化管理，快速接入。
- 为排障定制，高性价比。

日志检索

最近更新时间：2024-01-31 15:25:41

操作场景

当您的应用出现异常时，您可以通过日志检索功能，分析业务相关日志，实现高效、精准的故障定位异常。

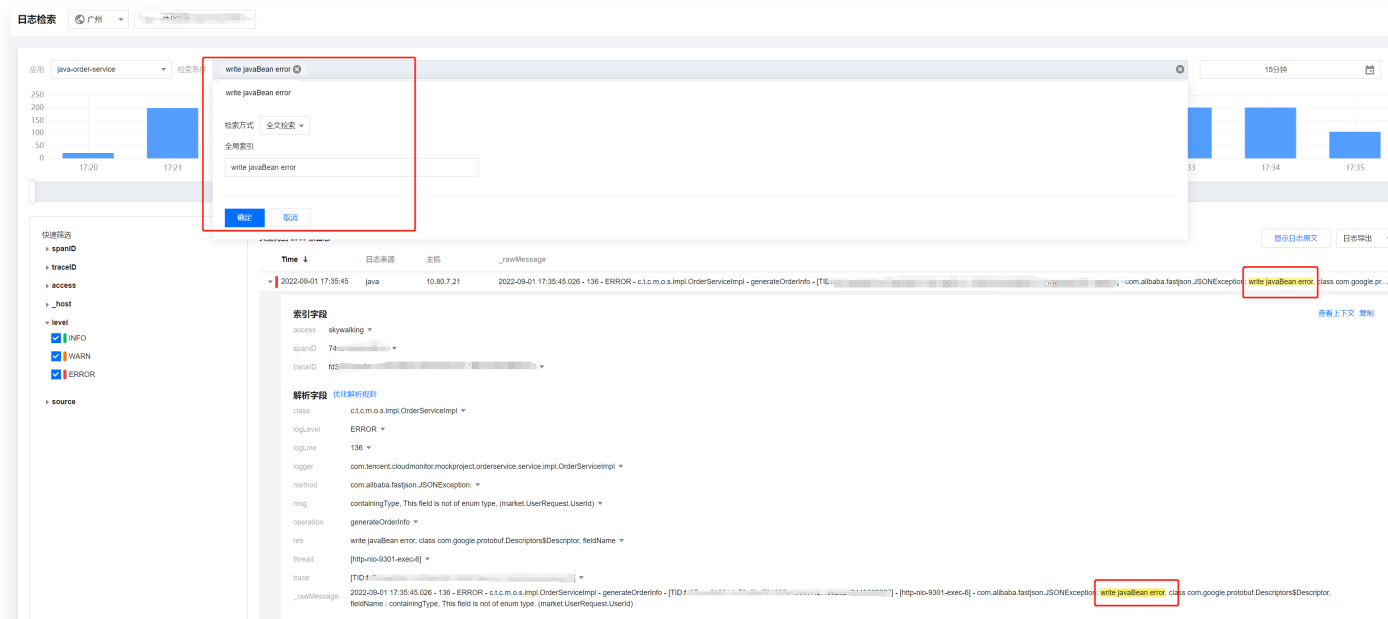
类别	描述	配置方式
全文索引	全文索引将原始日志整体切分为多个分词进行索引构建，检索时直接通过关键词进行检索（即全文检索）。 例如输入 <code>error</code> 表示检索包含 <code>error</code> 关键词的日志。	直接在日志检索时，输入关键字即可。
KV（键值）索引	键值索引将原始日志按字段（即 <code>key:value</code> ）分别切分为多个分词进行索引构建，检索时基于键值方式进行检索（即键值检索）。 例如输入 <code>level=error</code> 表示检索 <code>level</code> 字段中包含 <code>error</code> 的日志。	系统默认配置，或在 控制台 > 日志加工 > 全局索引 中配置。

操作步骤

全文检索

1. 登录 [腾讯云可观测平台](#) > [业务日志](#)。
2. 在左侧菜单栏中单击[日志检索](#)。
3. 在日志检索页面中选择对应的应用，单击[检索条件](#)。
4. 在弹框中选择全文检索并输入搜索关键字。

如下图：输入关键字“write javaBean error”后，检索包含出“write javaBean error”的所有日志。

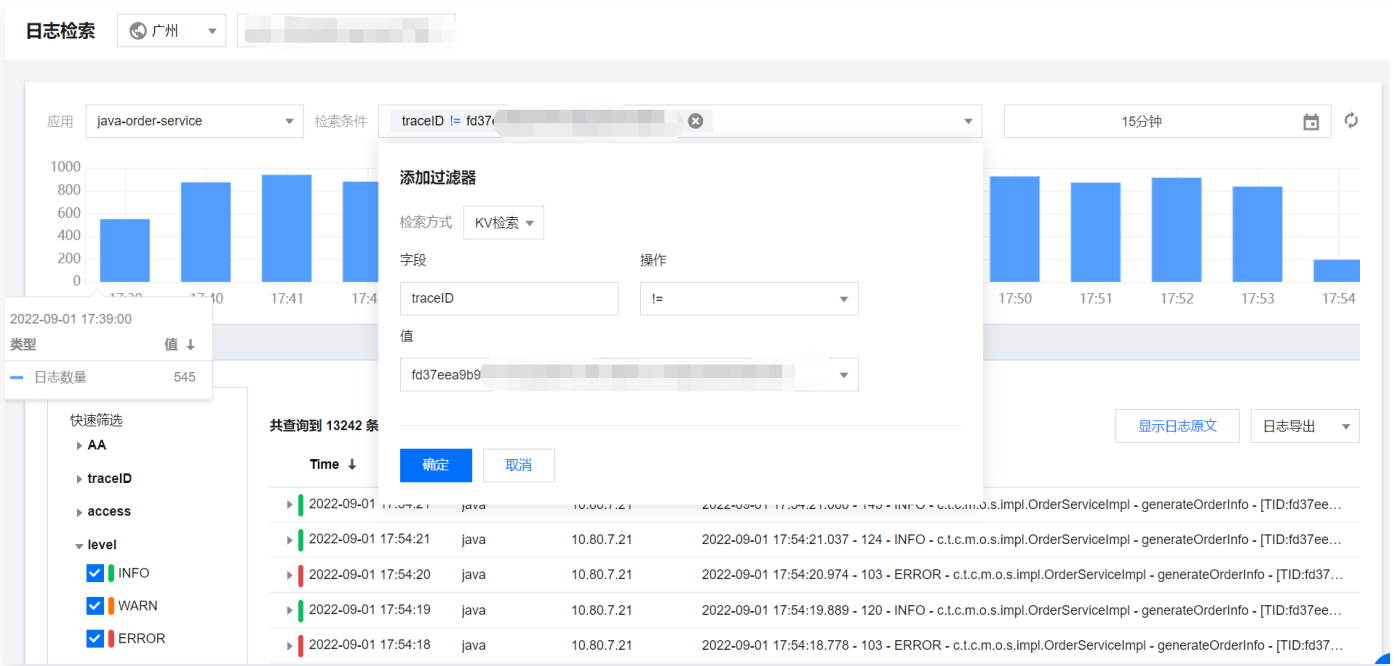


KV（键值）索引

系统默认为您配置下列索引，若需新建请前往[日志加工 > 全局索引](#)中添加。

索引名称	类型
access	String
level	String
service	String
source	String
spanID	String
traceID	String

1. 登录 [腾讯云可观测平台 > 业务日志](#)。
2. 在左侧菜单栏中单击[日志检索](#)。
3. 在日志检索页面中选择对应的应用，点击检索条件。
4. 在弹框中选择 **KV 检索**并输入选择搜索字段、操作和键值。



查看上下文

您还可以在日志详情中单击查看上下文检索，进行查看日志上下文信息，进一步排查应用性能问题。

```

2022-09-01 17:54:20.976 - 114 - ERROR - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - com.alibaba.fastjson.JSONException: write javaBean error, class com.google.protobuf.Descriptors$Descriptor, fieldName : containingType, This field is not of enum type. (market.MarketRequest.marketId)
2022-09-01 17:54:20.977 - 118 - WARN - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO MarketRequest{ "name": "DemoName for market-service", "marketId": "70" }
2022-09-01 17:54:20.960 - 88 - WARN - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO userInfo{"age":25,"dept":"1","id":1,"name":"user_1"}
2022-09-01 17:54:20.974 - 103 - ERROR - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - generateOrderInfo request delivery url:{"status":200,"msg":"OK","data":null}
2022-09-01 17:54:20.977 - 117 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO MarketRequest{ "name": "DemoName for market-service", "marketId": "70" }
2022-09-01 17:54:21.060 - 145 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO getUserInfo { "message": "Hello getUserInfoDemoName for user-service" }
2022-09-01 17:54:21.054 - 143 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO getAccountInfo { "message": "Hello DemoName for user-serviceThis is getAccountInfo" }
2022-09-01 17:54:21.037 - 135 - WARN - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - write javaBean error, class com.google.protobuf.Descriptors$Descriptor, fieldName : containingType, This field is not of enum type. (market.UserRequest.UserId)
2022-09-01 17:54:21.060 - 145 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO getUserInfo { "message": "Hello getUserInfoDemoName for user-service" }
2022-09-01 17:54:21.037 - 124 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO shopInfo { "message": "Hello DemoName for market-service" }
2022-09-01 17:54:21.049 - 141 - INFO - c.t.c.m.o.s.impl.OrderServiceImpl - generateOrderInfo - [TID:fd37... - [http-nio-9301-exec-2] - OrderInfoDTO getLoginInfo { "message": "Hello DemoName for user-service" }
    
```

快速筛选

将会展示所有索引键值，您也可以在快速筛选列表中，选择对应的键值，快速筛选您想要的日志。

快速筛选

- ▶ Action
- ▶ access
- ▼ level
 - INFO
 - ERROR
 - 未上报-
 - WARN
- ▶ source
- ▶ _host
- ▶ source_path

共查询到 93659 条日志

Time ↓	主机	日志来源	应用	_rawMessage
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.91782667+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.717267348+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.363761779+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.059067985+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.058984403+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961242918+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961159397+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961109939+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.960881383+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.897470588+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.877046988+08:00"...

生成指标

显示日志原文

日志导出 ▼

日志导出

您可以单击日志详情上方的日志导出 > 日志导出。

快速筛选

- ▶ Action
- ▶ access
- ▼ level
 - INFO
 - ERROR
 - 未上报-
 - WARN
- ▶ source
- ▶ _host
- ▶ source_path

共查询到 93659 条日志

Time ↓	主机	日志来源	应用	_rawMessage
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.91782667+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.717267348+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.363761779+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.059067985+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:48.058984403+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961242918+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961159397+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.961109939+08:00"...
▶ 2023-07-13 16:25:48	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.960881383+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.897470588+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.877046988+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.850946834+08:00"...
▶ 2023-07-13 16:25:47	[1258344699]-[30.167...	java	tapm-api-ap-guangzh...	{"Level":"INFO","Time":"2023-07-13T16:25:47.850900876+08:00"...

生成指标

显示日志原文

日志导出 ▼

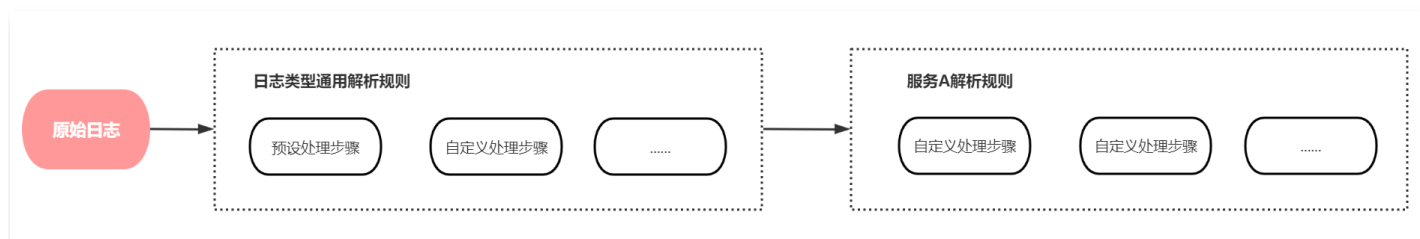
导出当前查询结果 ↓

导出记录

日志加工 解析规则

最近更新时间：2023-11-23 11:05:13

日志解析



如上图所示，应用性能监控支持配置预设解析规则对日志进行预处理。若预处理不满足您的需求，您还可以新增针对某个日志类型（例如：Java、Node.js），或者某个服务相关日志的自定义解析规则。

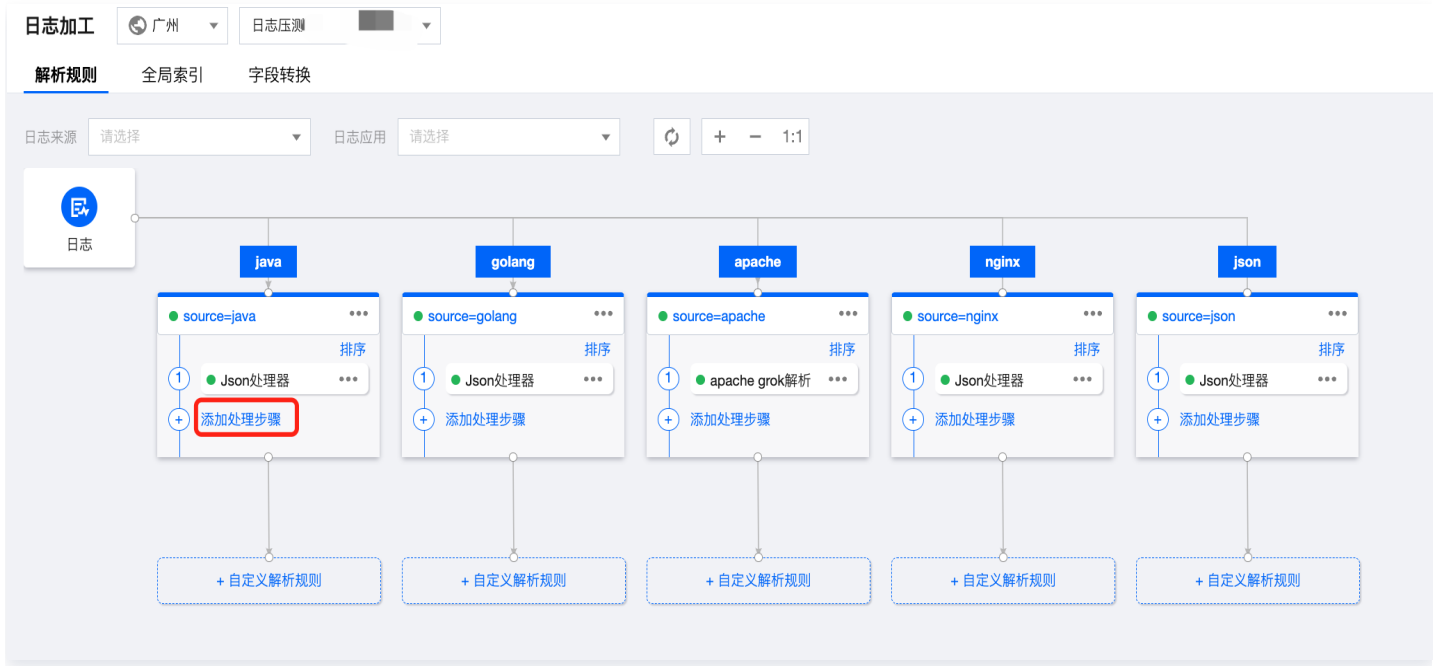
操作步骤

进入日志配置页面

1. 登录 [腾讯云可观测平台](#) > [业务日志](#)。
2. 在左侧菜单栏中单击**日志加工**。
3. 进入日志解析规则页面。

自定义特定日志类型处理规则

若您希望针对某类型的日志进行进一步加工处理，您可在对应类型日志的解析规则中，直接添加处理步骤。在解析规则页面，选择您的日志来源对应的解析规则，单击**添加处理步骤**。



新增某服务日志解析规则

若您希望针对某个应用的日志进行进一步加工处理，您需要先新建对应应用的解析规则，再在相关规则上添加处理步骤。

在解析规则页面，在对应日志来源的解析规则下一层，单击自定义解析规则。



在新建解析规则页面中，定义解析规则名称以及所选应用，以完成解析规则创建。

新建解析规则 ×

解析规则名称

日志来源 java ▼

应用 java-order-service ▼

添加处理步骤

添加完解析规则后，您可以单击**添加处理步骤**。配置说明如下。

- 基础信息：定义处理步骤名称。
- 输入原始日志（可选）。

如下图所示，系统为您自动拉取对应日志来源或应用下的最新一条日志，作为后续处理步骤配置参考样例。若所选样例不具有代表性，您可单击右上角**手动输入**，切换为手动输入模式，对样例进行调整或直接输入。

原始日志

_host [REDACTED]
 level INFO
 service java-user-service
 source java

自动获取
手动输入

```

_rawMessage 2022-09-05 10:22:36.720 - 31 - INFO - c.t.c.u.services.UserServiceImpl - getLoginInfo -
[TID:1[REDACTED]] - [grpc-default-executor-2392] -
userService getLoginInfo request:{
  "name": "DemoName for user-service",
  "UserId": "42"
}
    
```

前序步骤解析结果(自动生成)

```

{
  "_host" : "10.80.7.21"
  "logLevel" : "INFO"
  "method" : "userService"
  "thread" : "[grpc-default-executor-2392]"
  "operation" : "getLoginInfo"
  "timestamp" : "2022-09-05 10:22:36.720"
  "_rawMessage" :
  "2022-09-05 10:22:36.720 - 31 - INFO - c.t.c.u.services.UserServiceImpl - getLoginInfo -
  [TID:fd37eea9b9014c76a2bd501335e16e65.44.16623445566477869] - [grpc-default-executor-2392] -
  userService getLoginInfo request:{ \"name\": \"DemoName for user-service\", \"UserId\": \"42\" } \"
  \"source\" : \"java\"
  \"res\" : \"getLoginInfo request\"
  \"level\" : \"INFO\"
  \"msg\" : \"{ \"name\": \"DemoName for user-service\", \"UserId\": \"42\" } \"
  \"class\" : \"c.t.c.u.services.UserServiceImpl\"
  \"service\" : \"java-user-service\"
  \"logLine\" : \"31\"
  \"trace\" : \"[TID:fd37eea9b9014c76a2bd501335e16e65.44.16623445566477869]\"
}
    
```

- 定义处理规则

公测版本，仅支持如下四类处理类型，更多处理类型持续丰富中。

处理类型	能力说明	适用场景
GROK 解析	支持针对选定日志字段，通过配置 GROK 语法提取结构化字段	长文本字段内相关日志，进行切分及关键维度提取
JSON 解析	支持针对选定字段，按照 JSON 规则解析为结构化字段	JSON 格式的字符串，进行 K-V 提取
文本替换	支持针对选定字段，通过正则表达式、全等、部分匹配三种模式筛选并替换目标字符串	状态或业务码可读性转化，日志关键信息脱敏等
Geo IP 转换	支持针对选定的公网 IP 字段，解析初国家、省份、城市、ISP（运营商）	通过访问 IP，解析相关流向的地理来源

根据您的具体需求，您可以选择对应的解析类型并完成相应配置。系统会自动根据样例日志，为您模拟按照您输入规则解析后的结构化日志，方便您进行校验。

定义解析规则

类型 ▼
JSON 解析

字段 ▼
msg

错误处理 ▼
跳过当前算子逻辑

高级选项

解析结果

```

{
  "thread" : "[grpc-default-executor-2392]"
  "class" : "c.t.c.u.services.UserServiceImpl"
  "timestamp" : "2022-09-05 10:22:36.720"
  "name" : "DemoName for user-service"
  "source" : "java"
  "msg" : "{ \"name\": \"DemoName for user-service\", \"UserId\": \"42\" } "
  "method" : "userService"
  "_rawMessage" :
  "2022-09-05 10:22:36.720 - 31 - INFO - c.t.c.u.services.UserServiceImpl - getLoginInfo -
  [TID: ] - [grpc-default-executor-2392] -
  userService getLoginInfo request:{ \"name\": \"DemoName for user-service\", \"UserId\": \"42\" } "
  "_host" : " "
  "res" : "getLoginInfo request"
  "operation" : "getLoginInfo"
  "UserId" : "42"
  "service" : "java-user-service"
  "level" : "INFO"
  "logLevel" : "INFO"
  "logLine" : "31"
  "trace" : "[TID: ]|"
}

```

设置全局索引

最近更新时间：2023-11-23 11:05:13

操作场景

应用性能监控将为您默认配置 Access、Level、Trace ID、SpanID、Service、Host 等索引，提高优化解析效率。同时，对于日志文本中解析出的 K-V 类型字段，我们提供了原生 K-V 检索能力，无需索引即可使用。若上述检索效率不能满足您的需求，或您有通用的业务字段，例如：订单号、客户 ID 等字段需要频繁检索。您可按照如下方式新增全局索引，在日志检索时可快速通过该字段检索相关日志。

说明

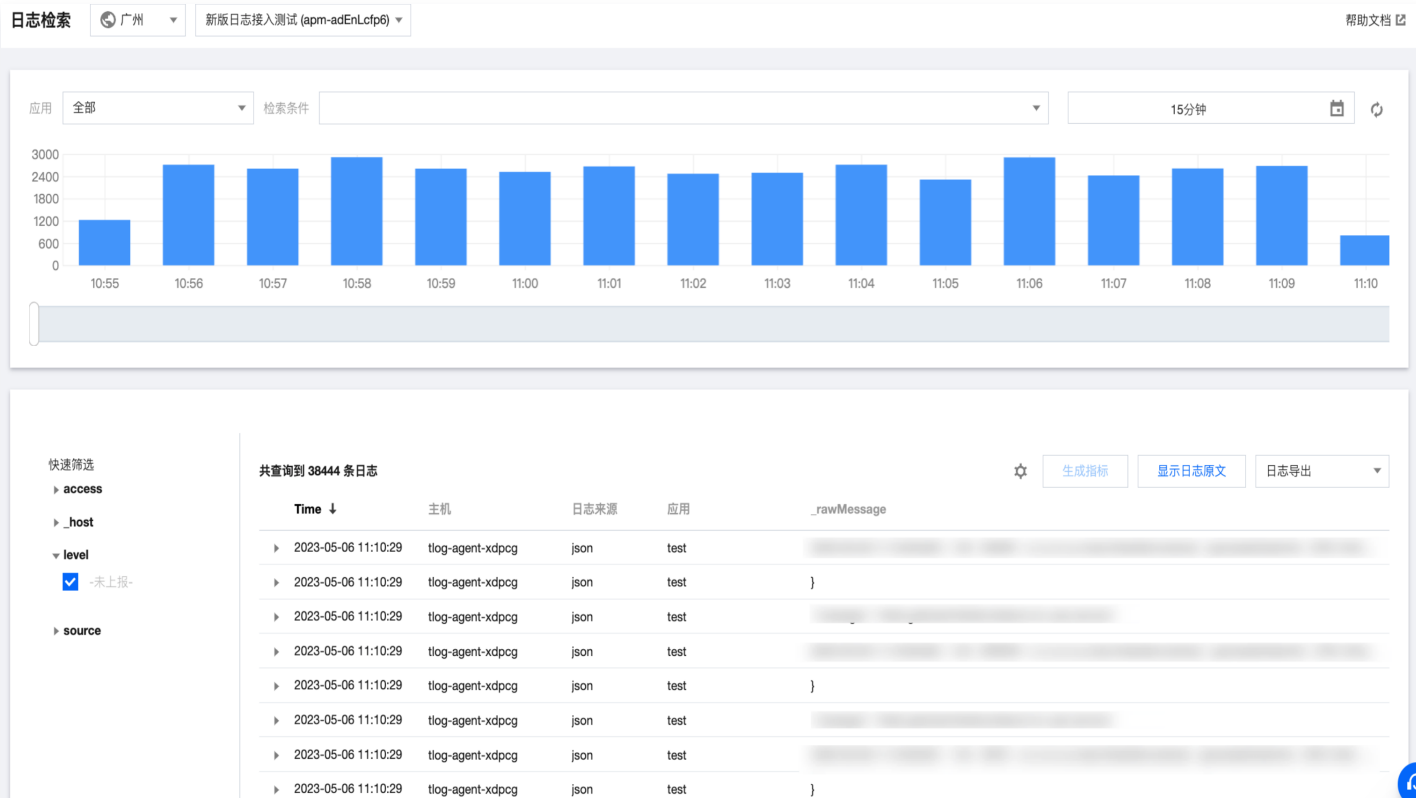
- 系统将会为您默认配置 access、level、service、source、spanID、traceID 键值索引，默认配置的键值索引不支持删除、编辑。
- 不支持创建与默认配置索引重名的索引 Key。
- 索引只对创建后新写入的日志数据生效。

操作步骤

步骤1：创建全局索引

方法一：在日志详情中设置全局索引

若您日志检索页面进行检索时，感到部分字段检索速度不达预期，您可以单击字段旁的下拉框，选择**设为全局索引**。



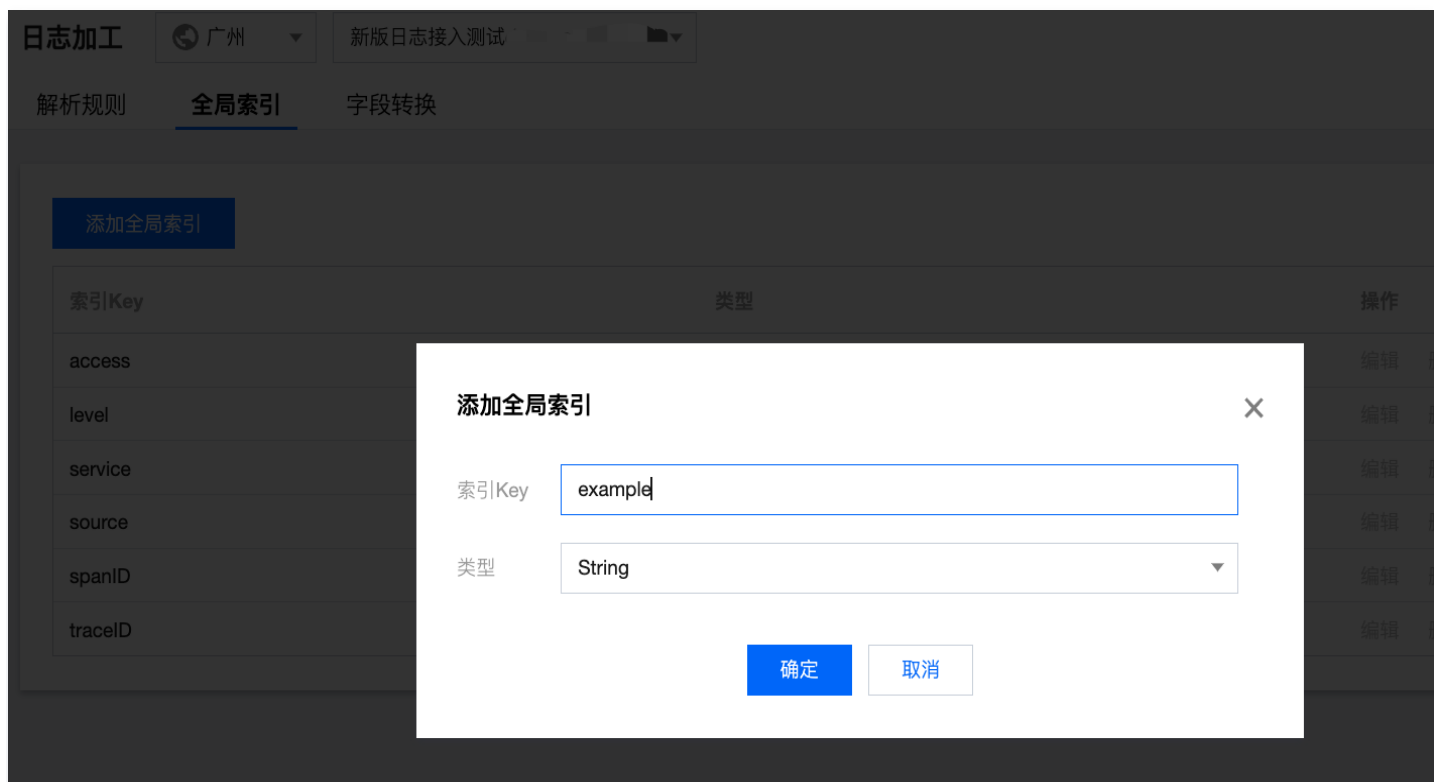
在对话框中，确认索引列名，并选择字段类型。根据所选字段类型不同，K-V 检索时支持的筛选方式也会略有不同。



方法二：日志加工页面手动添加

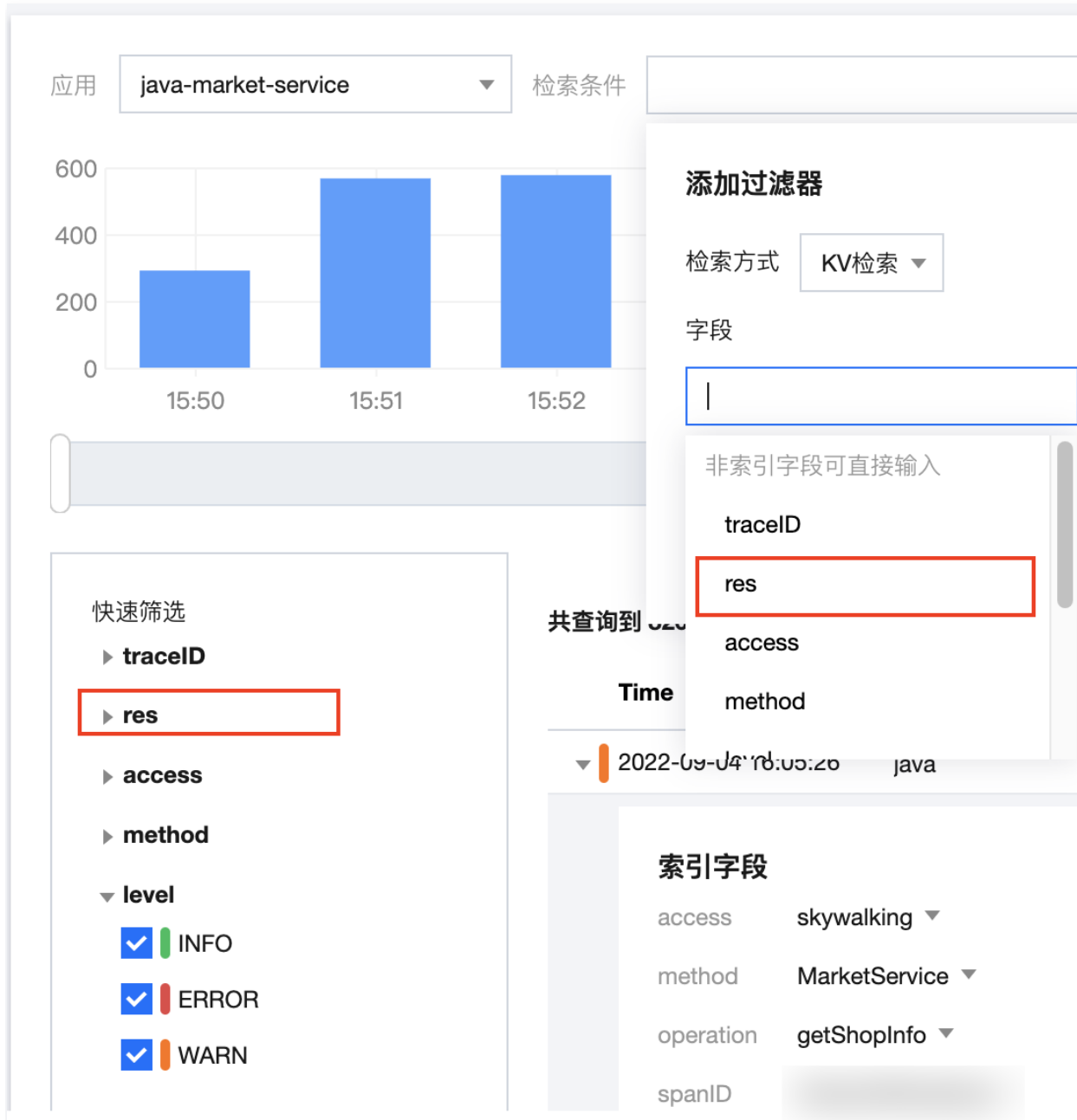
1. 登录 [腾讯云可观测平台](#)。
2. 在左侧菜单栏中单击业务日志 > 日志加工。
3. 在日志加工页面选择全局索引 > 添加全局索引。

- 在弹框中填写键值并选择数据类型，根据所选字段类型不同，K-V 检索时支持的筛选方式也会略有不同。
- 完成后单击**确定**即可。



步骤2：使用全局索引

索引创建完毕后，您可在日志检索页面左侧的快速筛选框，以及检索条件中的 KV 检索快速选择中使用。



The screenshot displays the logging search interface for the application 'java-market-service'. It includes a bar chart showing search results for times 15:50, 15:51, and 15:52. A '快速筛选' (Quick Filter) sidebar on the left lists filters like 'tracelID', 'res', 'access', 'method', and 'level'. A '添加过滤器' (Add Filter) dropdown menu is open, showing search methods ('KV检索') and a list of fields including 'tracelID', 'res', 'access', and 'method'. Below the dropdown, an '索引字段' (Index Fields) section shows mappings like 'access' to 'skywalking', 'method' to 'MarketService', and 'operation' to 'getShopInfo'.

索引类型说明

索引类型	说明
String	字符串，由数字、字母、下划线组成的一串字符
Int64	有符号64位整数
Int32	有符号32位整数
Int16	有符号16位整数
Int8	有符号8位整数

Float64	64位浮点数
Float32	32位浮点数
UInt64	无符号64位整数
UInt32	无符号32位整数
UInt16	无符号16位整数
UInt8	无符号8位整数
DateTime	日期与时间，例如：2022-09-04 16:00:00

管理全局索引

最近更新时间：2024-01-05 15:08:12

操作场景

本文将为您介绍如何编辑或删除全局索引。

- 预设索引字段用于维持基本检索功能，不支持编辑和删除。
- 索引编辑后，由于后台需要进行数据作业，根据数据量，会于5-30分钟后生效。
- 索引删除，删除索引会导致对应字段删除，请慎重操作。我们会为您保留原始信息，所以您仍可通过全文模糊搜索定位到对应字段。

操作步骤

1. 登录 [腾讯云可观测平台-业务日志](#)。
2. 在左侧菜单栏中单击**日志加工**。
3. 在日志加工页面切换为全局索引菜单。
4. 在全局索引页面可进行自定义的全局索引的删除和编辑。（系统预设的索引字段不支持删除和编辑）

The screenshot shows the 'Global Index' management interface. At the top, there are tabs for '日志加工' (Log Processing) and '全局索引' (Global Index). Below the tabs, there is a table of indices. The table has three columns: '索引Key' (Index Key), '类型' (Type), and '操作' (Action). The '操作' column contains '编辑' (Edit) and '删除' (Delete) buttons for each index. A red box highlights the '编辑' and '删除' buttons for the 'AA' and 'levelpp' indices.

索引Key	类型	操作
access	String	编辑 删除
level	String	编辑 删除
service	String	编辑 删除
source	String	编辑 删除
spanID	String	编辑 删除
traceID	String	编辑 删除
AA	String	编辑 删除
levelpp	String	编辑 删除

日志接入

Agent 接入日志（容器环境）

最近更新时间：2024-01-05 15:08:12

本文将为您介绍如何在容器环境下通过 sidecar 的方式安装 Agent 并完成日志接入。

前提条件

开通应用性能监控服务，并完成业务系统创建。参见 [快速入门](#)。

操作步骤

步骤1：日志上报配置

进入 [腾讯云可观测平台 > 日志概览](#) 页面，单击接入日志。

接入流程：

- 选择您的上报地域、业务模块、采集集群。
- 填写业务名称。

地域 *

业务模块 * [新建业务](#)

应用名称 *
最长50个字符，只能包含小写字母、数字及分隔符（“-”、“_”），且不能以分隔符开头或结尾

采集集群 *

采集 Agent 已安装

步骤2：日志采集配置

- 选择负载均衡类型、命名空间、工作负载、容器等信息。
- 采集路径配置：输入您日志存储的文件路径，支持输入多个路径。
- 高级配置（可选）：通过高级配置，可以额外配置采集模式、编码方式、日志采集排除文件、多行日志采集、保留和排除行等。

采集配置

配置1

负载类型 * 指定工作负载 指定 Pod Label

命名空间 * cert-manager

工作负载 * Deployment cert-manager

容器 * cert-manager

采集路径 * 11 / *

[+ 添加路径](#)高级配置 [使用默认配置](#)

确定

取消

[添加配置](#)

步骤3: Sidecar 方式安装探针

1. 进入 [容器服务控制台](#) > 集群。
2. 单击对应的容器集群名称或 ID，
3. 在左侧菜单栏中单击工作负载。
4. 在工作负载列表中单击更新 Pod 配置。
5. 在数据卷信息中单击添加数据卷。

基本信息

所在地域: 华南地区(广州)
集群ID: [模糊]
所在命名空间: dev
资源名称: tapm-api (deployment)

数据卷 (选填)

数据卷名称: [模糊]	数据卷类型: 使用ConfigMap	[模糊]	✎ ✕
数据卷名称: [模糊]	数据卷类型: 使用ConfigMap	[模糊]	✎ ✕
数据卷名称: [模糊]	数据卷类型: 使用ConfigMap	[模糊]	✎ ✕
数据卷名称: [模糊]	数据卷类型: 使用ConfigMap	[模糊]	✎ ✕
数据卷名称: [模糊]	数据卷类型: 使用主机路径	主机路径设置①	✎ ✕

添加数据卷

为容器提供存储，目前支持临时路径、主机路径、云硬盘数据卷。文件存储NFS、配置文件、PVC，还需挂载到容器的指定路径中。[使用指引](#)

实例内容器

更新Pod配置 **取消**

6. 找到目标服务所在 Pod 配置挂载点，共享日志目录。

新增数据卷

数据卷类型: 使用主机路径

数据卷名称: log

主机路径: ←

检查类型: DirectoryOrCreate

如果给定路径中不存在任何内容，则将根据需要创建一个空目录，且权限设置为0755

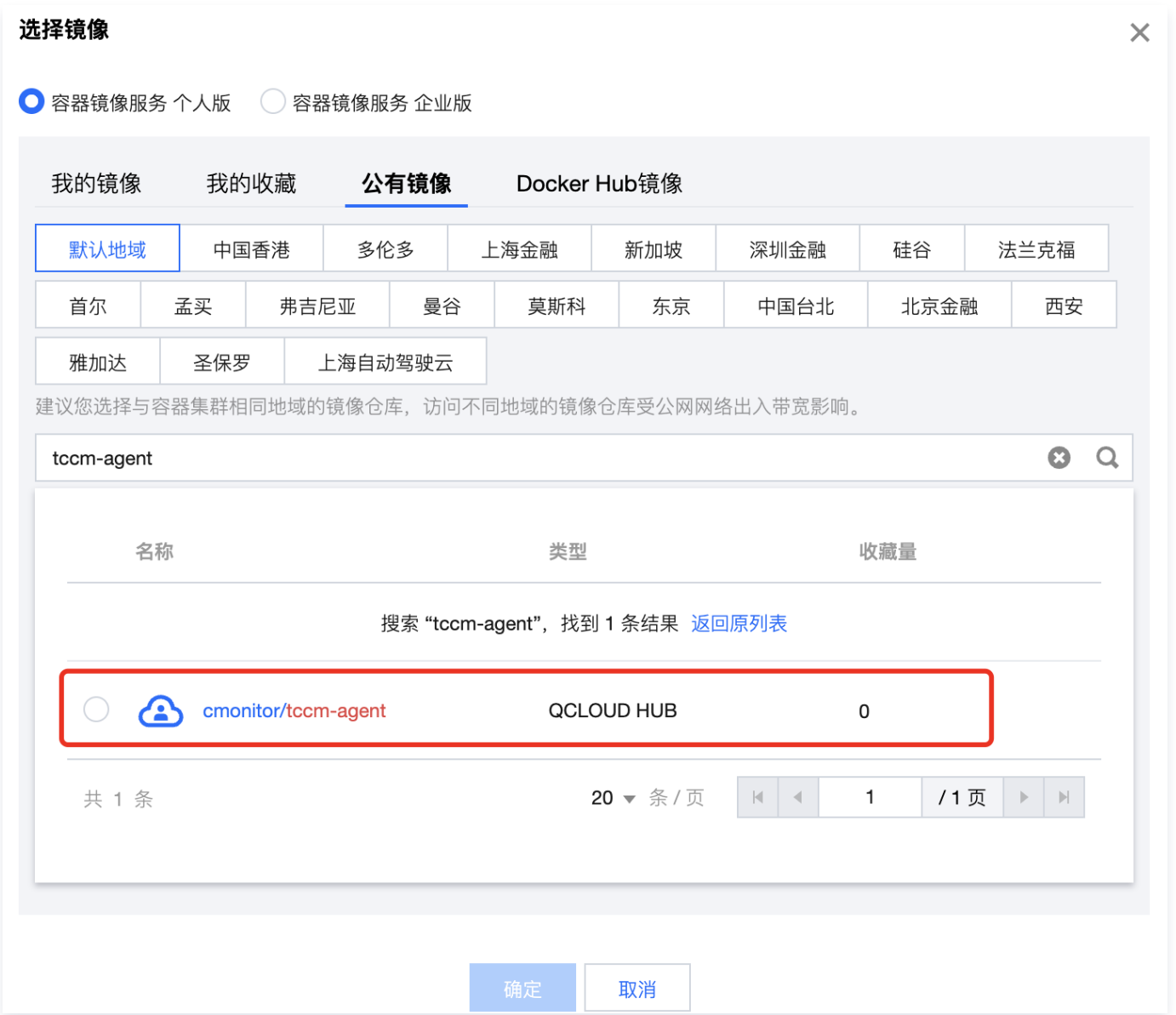
确认 **取消**

7. 找到业务所在服务 Pod 添加该挂载点，指向业务日志目录。



8. Sidecar 方式部署 Agent。

搜索 Agent 镜像 `ccr.ccs.tencentyun.com/cmonitor/tccm-agent`，选择版本1.2.12并部署。



9. 为 Agent 所在 sidecar 添加日志挂载点。

名称

镜像 [选择镜像](#)

镜像版本 (Tag) [选择镜像版本](#)

镜像拉取策略 Always IfNotPresent Never

总是从远程拉取该镜像

环境变量

名称	值	操作
...	...	✕
...	...	✕
...	...	✕
...	...	✕

[新增变量](#)

变量名为空时，在变量名称中粘贴一行或多行key=value或key: value的键值对可以实现快速批量输入

挂载点 ✕

[添加挂载点](#)

CPU/内存限制 CPU限制 内存限制

步骤4: 配置上报信息

在更新 Pod 配置页面，为 Agent sidecar 中配置如下环境变量。

变量名	变量值
TCCM_TOKEN	腾讯云账号 APPID
TCCM_REGION	业务所属地域
TCCM_LOG_TOKEN	在应用性能监控平台获取到的 Token

名称

镜像 [选择镜像](#)

镜像版本 (Tag) [选择镜像版本](#)

镜像拉取策略 Always IfNotPresent Never

默认使用本地镜像，若本地无该镜像则远程拉取该镜像

环境变量

自定义	PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin	×
自定义	TCCM_TOKE N	2! 042	×
自定义	TCCM_REGIO N	ap-guangzhou	×
自定义	TCCM_LOG_ TOKEN	QqL. JqzBe	×

[新增变量](#)

变量名为空时，在变量名称中粘贴一行或多行key=value或key: value的键值对可以实现快速批量输入

挂载点 挂载子路径 ×

[添加挂载点](#)

CPU/内存限制 CPU限制 内存限制

步骤5: 验证是否上报成功

上述步骤配置完成后，等待1~2分钟，若在 [日志检索](#) 所看到上报的日志，则说明上报成功。

Skwalking 探针接入日志

最近更新时间：2024-01-05 15:08:12

支持 Skywalking 探针（目前只支持 Java 的 Log4j/ Logback/ Log4j2）进行日志接入、接入成功后您将可以日志检索及分析。本文将为您介绍如何接入日志服务。

前提条件

已接入应用，若未接入可参见 [接入指南](#)。

操作步骤

步骤1：日志上报配置

1. 进入 [腾讯云可观测平台-日志概览](#) 页面，单击日志接入。
2. 在接入流程中，进行下列配置：
 - 选择您的上报地域和业务系统。公测期上报地域仅支持广州。
 - 根据您的日志内容，选择适合的日志类型。并选择 skywalking 作为接入方式。

步骤2：选择任意一种方式接入日志

说明

Logback、Log4j、Log4j2：都是 Apache 的开源项目，常用的日志框架，通过使用这些日志框架，我们可以控制日志信息输送的目的地，还可以通过一个配置文件来灵活地进行配置，而不需要修改应用的代码。

- **Logback** 是一个开源日志组件，是 Slf4j 的原生实现框架，相比 Log4j，Logback 执行速度更快，占用内存更少，是 spring boot 默认集成的日志框。
- **Log4j & Log4j2** **Log4j** 和 **Log4j2** 也都是 Apache 的开源日志框架，Log4j2 是 Log4j 1.x 和 Logback 的改进版，采用无锁异步等，使日志吞吐量、性能比 log4j 1.x 提高 10 倍，并解决了一些老版本的 bug，且配置更加简单灵活。

方式一：Logback

1. 在 pom 文件中添加相关依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-logback-1.x</artifactId>
<version>8.5.0</version>
</dependency>
```

2. 在 resources 中新建 logback-spring.xml 文件。

```
<configuration>
<include resource="org/springframework/boot/logging/logback/defaults.xml">
</include>

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">
  <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
    <layout
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.TracedPatternLogbackL
ayout">
      <Pattern>-%clr(%d{${LOG_DATEFORMAT_PATTERN:-yyyy-MM-dd
HH:mm:ss.SSS}}){faint} %clr(${LOG_LEVEL_PATTERN:-%5p}) %clr(${PID:- }
{magenta})%clr(---){faint} %clr([%15.15t]){faint} %clr(%-40.40logger{39}){cyan}
%clr(:){faint} %m%n${LOG_EXCEPTION_CONVERSION_WORD:-%wEx}</Pattern>
    </layout>
  </encoder>
</appender>
<appender name="grpc-log"
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.log.GRPCLogClientAppe
nder">
  <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
    <layout
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.mdc.TracedMDCPattern
LogbackLayout">
      <Pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} [%X{tid}] [%thread] %-5level
%logger{36} %msg%n</Pattern>
    </layout>
  </encoder>
</appender>
</configuration>
```

3. 在 agent.config 中添加 以下内容。

```
plugin.toolkit.log.grpc.reporter.server_host=${SW_GRPC_LOG_SERVER_HOST:pl.ap-
guangzhou.apm.tencentcs.com}
plugin.toolkit.log.grpc.reporter.server_port=${SW_GRPC_LOG_SERVER_PORT:}

plugin.toolkit.log.grpc.reporter.max_message_size=${SW_GRPC_LOG_MAX_MESSA
GE_SIZE:10485760}

plugin.toolkit.log.grpc.reporter.upstream_timeout=${SW_GRPC_LOG_GRPC_UPSTRE
AM_TIMEOUT:30}
```

方式二: Log4j2

1. 在 pom 文件中添加相关依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-log4j-2.x</artifactId>
<version>{project.release.version}</version>
</dependency>
```

2. 在 log4j2.xml 中增加如下内容支持日志中打印 Traceld。

```
<Appenders>
<Console name="Console" target="SYSTEM_OUT">
  <PatternLayout pattern="%d [%traceld] %-5p %c{1}:%L - %m%n">
  </PatternLayout>
</Console>
</Appenders>
```

3. 在 log4j2.xml 中添加 GRPCLogClientAppender。

```
<GRPCLogClientAppender name="grpc-log">
<PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} -
%msg%n"></PatternLayout>
</GRPCLogClientAppender>
```

4. 在 agent.config 中添加 以下内容:

```
plugin.toolkit.log.grpc.reporter.server_host=${SW_GRPC_LOG_SERVER_HOST:pl.ap-
guangzhou.apm.tencentcs.com}
plugin.toolkit.log.grpc.reporter.server_port=${SW_GRPC_LOG_SERVER_PORT:}

plugin.toolkit.log.grpc.reporter.max_message_size=${SW_GRPC_LOG_MAX_MESSA
GE_SIZE:10485760}

plugin.toolkit.log.grpc.reporter.upstream_timeout=${SW_GRPC_LOG_GRPC_UPSTRE
AM_TIMEOUT:30}
```

方式三: Log4j

1. 在 pom 文件中添加相关依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-log4j-1.x</artifactId>
<version>{project.release.version}</version>
</dependency>
```

2. 在 log4j.properties 中添加 GRPCLogClientAppender。

```
<configuration>
<include resource="org/springframework/boot/logging/logback/defaults.xml">
</include>

<appender name="console" class="ch.qos.logback.core.ConsoleAppender">
  <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
    <layout
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.TracedPatternLogbackL
ayout">
      <Pattern>-%clr(%d${LOG_DATEFORMAT_PATTERN:-yyyy-MM-dd
HH:mm:ss.SSS}){faint} %clr(${LOG_LEVEL_PATTERN:-%5p}) %clr(${PID:- }
{magenta})%clr(---){faint} %clr([%15.15t]){faint} %clr(%-40.40logger{39}){cyan}
%clr(:){faint} %m%n${LOG_EXCEPTION_CONVERSION_WORD:-%wEx}</Pattern>
    </layout>
  </encoder>
</appender>

<appender name="grpc-log"
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.log.GRPCLogClientAppe
nder">
  <encoder class="ch.qos.logback.core.encoder.LayoutWrappingEncoder">
    <layout
class="org.apache.skywalking.apm.toolkit.log.logback.v1.x.mdc.TracedMDCPattern
LogbackLayout">
      <Pattern>%d{yyyy-MM-dd HH:mm:ss.SSS} [%X{tid}] [%thread] %-5level
%logger{36} %msg%n</Pattern>
    </layout>
  </encoder>
</appender>
</configuration>
```

3. 在 agent.config 中添加 以下内容。

```
plugin.toolkit.log.grpc.reporter.server_host=${SW_GRPC_LOG_SERVER_HOST:pl.ap-  
guangzhou.apm.tencentcs.com}  
plugin.toolkit.log.grpc.reporter.server_port=${SW_GRPC_LOG_SERVER_PORT:}  
plugin.toolkit.log.grpc.reporter.max_message_size=${SW_GRPC_LOG_MAX_MESSA  
GE_SIZE:10485760}  
plugin.toolkit.log.grpc.reporter.upstream_timeout=${SW_GRPC_LOG_GRPC_UPSTR  
EAM_TIMEOUT:30}
```

步骤3: 验证是否上报成功

上述步骤配置完成后，等待1~2分钟，若在 [日志检索](#) 所看到上报的日志，则说明上报成功。

时间格式说明

最近更新时间：2024-01-05 15:08:12

业务日志要求每条日志必须具有时间属性，以便系统按时间维度对数据进行管理。您可以参见下文配置时间解析格式。

解析格式说明

参数格式	说明	例子
G	代号	AD
y	年，即当前真正所属的年份	1996 ; 96
Y	周年，就是这天所在的周所属的年份	2009 ; 09
M	月份（01 - 12），01表示一月份	July ; Jul ; 07
w	一年中的一周，1-52	27
W	一个月中的一周，1-4	2
D	一年中的某一天，1-366	189
d	一个月中的第几天，1-31	10
F	一个月中的星期几，（1 = 星期一，...，7 = 星期日）	2
E	星期中的日期名称	Tuesday ; Tue
u	星期几（1 = 星期一，...，7 = 星期日）	1
a	上午（AM）/下午标记（PM）	PM
H	一天中的小时数（0-23）	0
k	一天中的小时（1-24）	24
K	上午/下午的小时（0-11）	0
h	上午/下午的时间（1-12）	12

m	小时中的分钟，0-60	30
s	分秒	55
S	毫秒	978
z	时区	Pacific Standard Time ; CST ; GMT+08:00
Z	时区	-0800
X	时区	-08 ; -0800 ; -08:00

配置示例

下列以中国标准时间（CST，+0800）2022-07-04 12:08:56为例：

时间表示示例	时间提取格式
2022.07.04 AD at 12:08:56 CST	"yyyy.MM.dd G 'at' HH:mm:ss z"
Wed, Jul 4, '01	"EEE, MMM d, 'yy"
12:08PM	"h:mm a"
0:08 PM, CST	"K:mm a, z"
02022.July.04 AD 12:08 PM	"yyyyy.MMMMMM.dd GGG hh:mm aaa"
Wed, 4 Jul 2022 12:08:56 +0800	"EEE, d MMM yyyy HH:mm:ss Z"
220704120856+0800	"yyMMddHHmmssZ"
2022-07-04T12:08:56.235+0800	"yyyy-MM-dd'T'HH:mm:ss.SSSZ"
2022-07-04T12:08:56.235+08:00	"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"
2022-W27-3	"YYYY-'W'ww-u"