日志服务

常见问题

腾讯云

文档目录

常见问题
健康监测问题解释
日志上传相关

最近更新时间：2024-05-29 11:46:11

如何应对提示参数错误?

该错误说明通过 API 或 SDK 上传日志时，请求输入参数 填写错误，请确保参数填写正确。

哪些原因可能会导致鉴权失败?

该错误说明上传日志时出现鉴权错误。 该类问题的出现通常可能由以下几种原因导致：

错误原因	解决方法
密钥不存在	请在控制台检查密钥是否已被删除或者禁用。 如状态正常，请检查密钥是否填写正确，注意前后不得有空格。您可单击 此处 查看您的密钥信息。
签名错误	签名计算错误，请对照调用方式中的 签名方法 检查签名计算过程。
签名过期	请对照调用方式中的签名方法文档重新计算签名。
请求未授权	无上传日志权限。请前往 CAM 控制台，为您的账号添加 CLS 上传日志的权限。
密钥非法	密钥格式不正确。您可单击 此处 查看您的密钥信息。
其他	若以上原因均已排除，且错误持续存在， 请联系 在线客服 提交工单反馈。

如何应对上传日志大小超限?

单条日志上传请求中的日志大小存在超限， 请根据以下规格限制调整日志上传大小：

限制项	说明
上传日志	一个 pb 包里 logGroup 不能超过10个。
	一个 pb 包里 logGroup 包含日志条数最多为10000条，至少包含1条。
	log 中单个 value 最大为1MB。
	一个 pb 包里 logGroup 部分所有 Value 大小最大总和为5MB。
	单次上传请求的包体压缩前不能超过6MB。

为什么会出现触发流控或频控?

该错误说明存在以下规格超限的情况：

限制项	说明
写频控	单个日志主题分区写请求限制：500QPS。
写流控	单个日志主题分区写流量限制：5MB/s。

建议减少日志上传的频次与流量。 若不希望改变日志上传的频次或流量，建议为日志主题 开启自动分裂 。

如何应对上传请求错误?

该错误说明存在其他上传错误，请联系 在线客服 提交工单反馈。

采集配置相关

最近更新时间：2024-10-18 10:20:11

非结构化提取模式和结构化提取模式的区别?

非结构化提取模式指的是采集配置中，提取模式配置为单行全文或多行全文。 这类提取模式不会对日志进行格式化解析， 而是全文采集。 在这种提取模式下， 日志采集上来只能进行最简单的全文检索， 日志条数统计，日志条数告警。
CLS 日志服务支持更强大的基于日志内容中的特定字段进行检索分析，可视化与监控告警。 该能力依赖对所采集日志进行格式化解析， 即在采集配置中选择结构化提取模式。若您的日志输出格式统一， 建议配置结构化提取模式（免费）， 以最大提升您的日志分析体验。

如何配置结构化提取模式?

1. 登录 日志服务， 在左侧导航栏中，单击**日志主题**，进入日志主题管理页面。
2. 找到目标日志主题， 单击**日志主题名称**进入日志主题配置页面。
3. 选择**采集配置**页签，在**LogListener 采集配置**找到提取模式为单行或多行全文的采集配置，单击查看进入采集配置详情页面。
4. 在采集配置详情页中， 单击**修改配置**进入采集配置编辑页。
5. 找到**提取模式**配置项， 根据您的日志格式， 单击**修改**并选择相应的结构化提取模式。

6. 结构化提取模式

单行-完全正则格式

单行完全正则格式通常用来处理结构化的日志，指将一条完整日志按正则方式提取多个 key-value 的日志解析模式。
假设一条日志原始数据为：

```
10.135.46.111 - - [22/Jan/2019:19:19:30 +0800] "GET /my/course/1 HTTP/1.1" 127.0.0.1 200 782 9703
"http://127.0.0.1/course/explore?
```

```
filter%5Btype%5D=all&filter%5Bprice%5D=all&filter%5BcurrentLevelId%5D=all&orderBy=studentNum"
"Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0) Gecko/20100101 Firefox/64.0"  0.354 0.354
```

配置的自定义正则表达式为：

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

系统根据 `()` 捕获组提取对应的 key-value 后，您可以自定义每组的 key 名称如下所示：

```
body_bytes_sent: 9703
http_host: 127.0.0.1
http_protocol: HTTP/1.1
http_referer: http://127.0.0.1/course/explore?
filter%5Btype%5D=all&filter%5Bprice%5D=all&filter%5BcurrentLevelId%5D=all&orderBy=studentNum
http_user_agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0) Gecko/20100101 Firefox/64.0
remote_addr: 10.135.46.111
request_length: 782
request_method: GET
request_time: 0.354
request_url: /my/course/1
status: 200
time_local: [22/Jan/2019:19:19:30 +0800]
upstream_response_time: 0.354
```

多行-完全正则格式

多行-完全正则模式适用于日志文本中一条完整的日志数据跨占多行（例如 Java 程序日志），可按正则表达式提取为多个 key-value 键值的日志解析模式。若不需要提取 key-value，请参阅 多行全文格式 进行配置。配置多行-完全正则模式时，您需要先输入日志样例，再自定义正则表达式。配置完成后，系统将根据正则表达式中的捕获组提取对应的 key-value。
假设一条日志原始数据为：

```
[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)
```

配置的自定义正则表达式为：

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

行首正则表达式为：

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

根据提取的 key，采集到日志服务的数据为：

```
time：  2018-10-01T10:30:01,000`
level：  INFO`
msg：java.lang.Exception: exception happened
    at TestPrintStackTrace.f(TestPrintStackTrace.java:3)
    at TestPrintStackTrace.g(TestPrintStackTrace.java:7)
```

```
    at TestPrintStackTrace.main(TestPrintStackTrace.java:16)
```

JSON 格式

JSON 格式日志会自动提取首层的 key 作为对应字段名，首层的 value 作为对应的字段值，以该方式将整条日志进行结构化处理，每条完整的日志以换行符\n为结束标识符。
假设一条 JSON 日志原始数据为：

```
{"remote_ip":"10.135.46.111","time_local":"22/Jan/2019:19:19:34
+0800","body_sent":23,"responsetime":0.232,"upstreamtime":"0.232","upstreamhost":"unix:/tmp/php-
cgi.sock","http_host":"127.0.0.1","method":"POST","url":"/event/dispatch","request":"POST
/event/dispatch HTTP/1.1","xff":"-","referer":"http://127.0.0.1/my/course/4","agent":"Mozilla/5.0
(Windows NT 10.0; WOW64; rv:64.0) Gecko/20100101 Firefox/64.0","response_code":"200"}
```

经过日志服务结构化处理后，该条日志将变为如下：

```
agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:64.0) Gecko/20100101 Firefox/64.0
body_sent: 23
http_host: 127.0.0.1
method: POST
referer: http://127.0.0.1/my/course/4
remote_ip: 10.135.46.111
request: POST /event/dispatch HTTP/1.1
response_code: 200
responsetime: 0.232
time_local: 22/Jan/2019:19:19:34 +0800
upstreamhost: unix:/tmp/php-cgi.sock
upstreamtime: 0.232
url: /event/dispatch
xff: -
```

分隔符格式

分隔符日志是指一条日志数据可以根据指定的分隔符将整条日志进行结构化处理，每条完整的日志以换行符 \n 为结束标识符。日志服务在进行分隔符格式日志处理时，您需要为每个分开的字段定义唯一的 key。
假设您的一条日志原始数据为：

```
10.20.20.10 ::: [Tue Jan 22 14:49:45 CST 2019 +0800] ::: GET /online/sample HTTP/1.1 ::: 127.0.0.1 :::
200 ::: 647 ::: 35 ::: http://127.0.0.1/
```

当日志解析的分隔符指定为 `:::` ，该条日志会被分割成八个字段，并为这八个字段定义唯一的 key，如下所示：

```
IP: 10.20.20.10 -
bytes: 35
host: 127.0.0.1
length: 647
referer: http://127.0.0.1/
request: GET /online/sample HTTP/1.1
status: 200
time: [Tue Jan 22 14:49:45 CST 2019 +0800]
```

索引配置相关

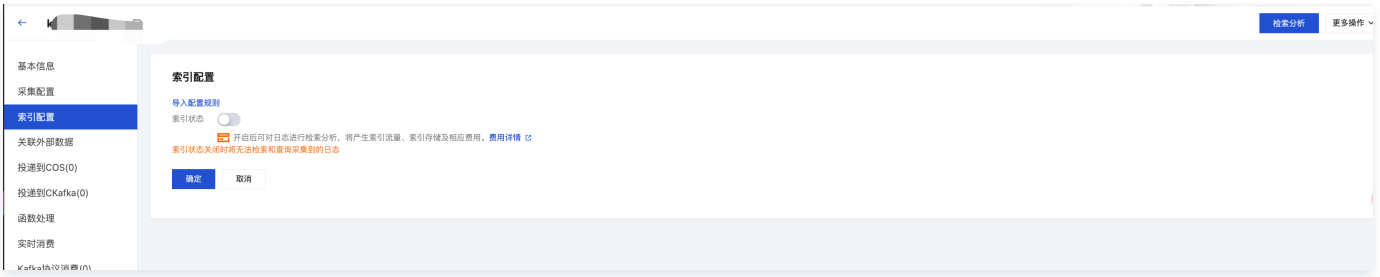
最近更新时间：2025-04-03 17:39:52

如何开启索引配置?

索引配置 是使用日志服务（Cloud Log Service，CLS）进行检索分析的必要条件，未开启索引将无法对日志进行检索分析。 强烈建议开启。

操作步骤

1. 登录 日志服务控制台。
2. 在左侧导航栏中，选择**日志主题**，进入日志主题列表页面。
3. 单击需要配置索引的日志主题 ID/名称，进入日志主题管理页面。
4. 选择**索引配置**页签， 单击**编辑**，进入编辑索引配置页面。



5. 单击**索引状态**开关， 开启索引。
6. 开启索引后， 您可进一步配置全文索引或键值。

- **全文索引**：支持对日志进行全文检索。
- **键值索引**：支持基于日志内容中的指定字段进行检索。该配置的前提是日志采集配置中的提取模式为结构化提取模式（即将日志解析为键值对）。

如何开启键值索引配置?

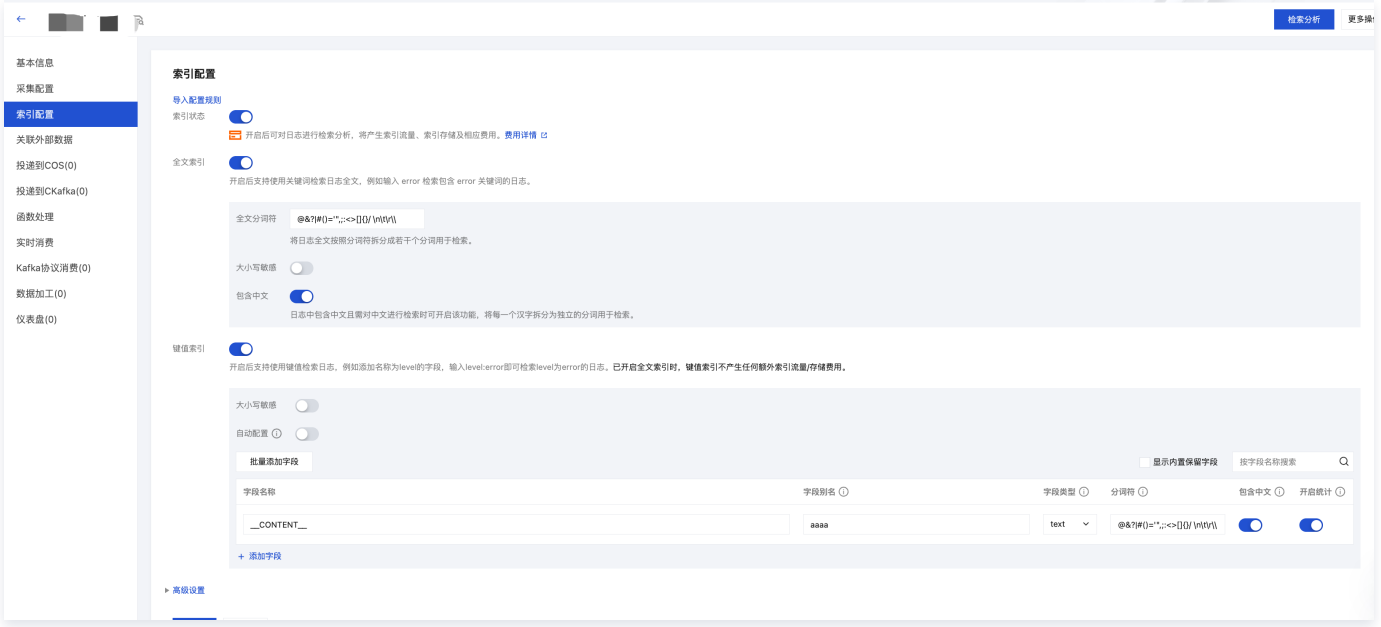
键值索引指的是在全文索引的基础上，进一步将原始日志按字段（即 key:value）分别切分为多个分词进行索引构建，检索时基于键值方式进行检索（即键值检索）。强烈建议开启，以最大化提升日志检索的效率。

> **注意：**
> 开启键值索引的前提条件是日志采集配置中的提取模式为结构化提取模式（即将日志解析为键值对）。

操作步骤

1. 登录 日志服务控制台。
2. 在左侧导航栏中，选择**日志主题**，进入日志主题列表页面。
3. 单击需要配置索引的日志主题 ID/名称，进入日志主题管理页面。
4. 选择**索引配置**页签， 单击**编辑**，进入编辑索引配置页面。

5. 单击**键值索引**开关，开启键值索引。
6. 开启键值索引后，您还可以单击自动配置，系统将自动获取采集到的最近1条日志作为样例，并将其中的字段解析为键值索引。您可以在自动配置的基础上进行微调，快速获取最终的索引配置信息。



计费相关

最近更新时间：2025-05-16 10:47:52

免费额度相关

CLS 有免费额度吗?

2022年9月5日以后所有开通 CLS 的新用户，可在开通 CLS 服务时免费领取10U * 3个月的资源包，可用于抵扣按量计费产生的费用。更多资源包信息，请参见 资源包（预付费）介绍 。

资源包相关

如何查看资源包使用情况?

您可以前往 CLS 控制台 > 资源包 > 资源包管理 页面，查看已购买的资源包的生效时间、到期时间以及使用量情况。

资源包包含哪些费用?

CLS 资源包可用于抵扣中国站所有地域的所有按量计费项，超出资源包规格或者资源包到期后，您使用 CLS 服务产生的费用将自动转为按量计费。

资源包将如何抵扣?

CLS 每个计费项抵扣消耗资源包额度（U）的比例，与该计费项的按量付费的单价完全一致。例如，中国大陆的写日志流量的按量付费价格为0.18元/GB，则每GB写日志流量，抵扣0.18U。更多详情参见 资源包（预付费）介绍 。

CLS 资源包是否可以叠加使用?

CLS 资源包可以叠加使用。但仅资源包的规格叠加，有效时长不叠加。关于资源包的购买示例、生效周期以及生效范围，请参见 资源包（预付费）介绍 。

CLS 资源包支持续费、升级或者退费吗?

CLS 资源包支持续费与退费，暂不支持升级。续费指引请参见 资源包续费 ，退费指引请参见 资源包退费 。

CLS 资源包到期后日志数据会丢失吗? 需要迁移数据吗?

资源包已用完或资源包到期后会**自动转为后付费模式**，即从账号余额中扣费，因此不会导致数据丢失，**无需对数据进行迁移操作**。只有当您系统欠费15天的情况下，数据才会被销毁，详情请参见 欠费说明 文档。

扣费相关

已购买资源包/已领取新手体验包为何仍会欠费?

1. 超出额度的资源使用费用
   例如，CLS 赠送给您的是10U的新手免费体验资源包，但是您实际使用了更多的用量，超出了资源包可抵扣的范围并进行了按量计费，因此产生了扣费。
2. 资源包到期
   CLS 为首次开通服务的用户提供3个月有效期的10U资源包。新手资源包过期后，所有费用将按量计费。您可以登录 CLS 控制台，进入 资源包 > 资源包管理 页面查看新手体验资源包或者您购买的资源包的有效期。

已清理日志服务资源，为何第二天仍会产生费用?

CLS 的计费方式为先使用，后付费。按照各**计费项**的实际用量，以天为单位，每日进行计量、结算、扣费和出账。第二天结算前一天的账单，所以清理了日志服务资源后，第二天需要结算前一天的账单，进行一次扣费，此后不再产生费用。

欠费停服相关

如何清理日志服务资源以停止计费?

CLS 无一键停用服务的功能。若您不再使用 CLS 服务，您可以选择将您的日志主题及分区等资源完全删除以避免继续计费，无需注销账号（如有使用其他腾讯云服务，注销账号会受到影响）。操作请参见 清理日志服务资源 。

采集相关
机器组常见问题

最近更新时间：2024-10-14 16:10:41

机器组添加机器不生效

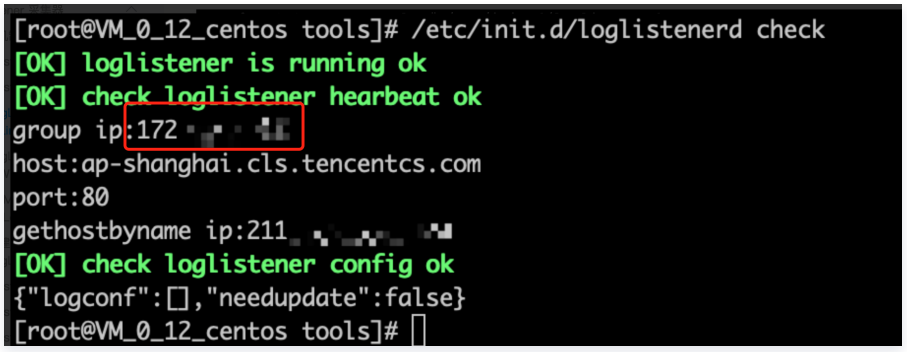
通过 IP 添加机器不生效

1. 检查 IP 对应机器是否已安装 LogListener，安装指引请参见 LogListener 安装和部署 。
2. 参考 如何使用采集自检工具，检查 LogListener 是否存在异常。
3. 检查机器组中添加的 IP 是否与 LogListener 识别到的 IP 一致。您可通过执行以下代码查看 LogListener 检测到的 IP。

Linux

```
/etc/init.d/loglistenerd check
```

在返回数据中，通过查看 `group ip` 查看 LogListener 识别到的 IP， 如下图：



Windows

以安装路径 `C:\Program Files (x86)\Tencent\LogListener` 为例，以管理员身份运行 Windows PowerShell，在安装路径下，执行以下命令检查 LogListener 心跳及配置：

```
.\loglistener_work.exe check
```

在返回数据中， 通过查看 `group_id` 查看 LogListener 识别到的 IP， 如下图：

通过机器标识添加机器不生效

1. 检查目标机器是否已安装 LogListener，安装指引请参考 LogListener 安装和部署 。
2. 参考 如何使用采集自检工具，检查LogListener是否存在异常。
3. 检查 LogListener 的机器标识是否与机器组中配置的机器标识一致。您可参考以下步骤查看 LogListener 的机器标识。

Linux

以安装路径 `/usr/local/` 为例，您可在 `/usr/local/loglistener/etc` 路径下，运行以下命令查看并配置 LogListener 配置文件。

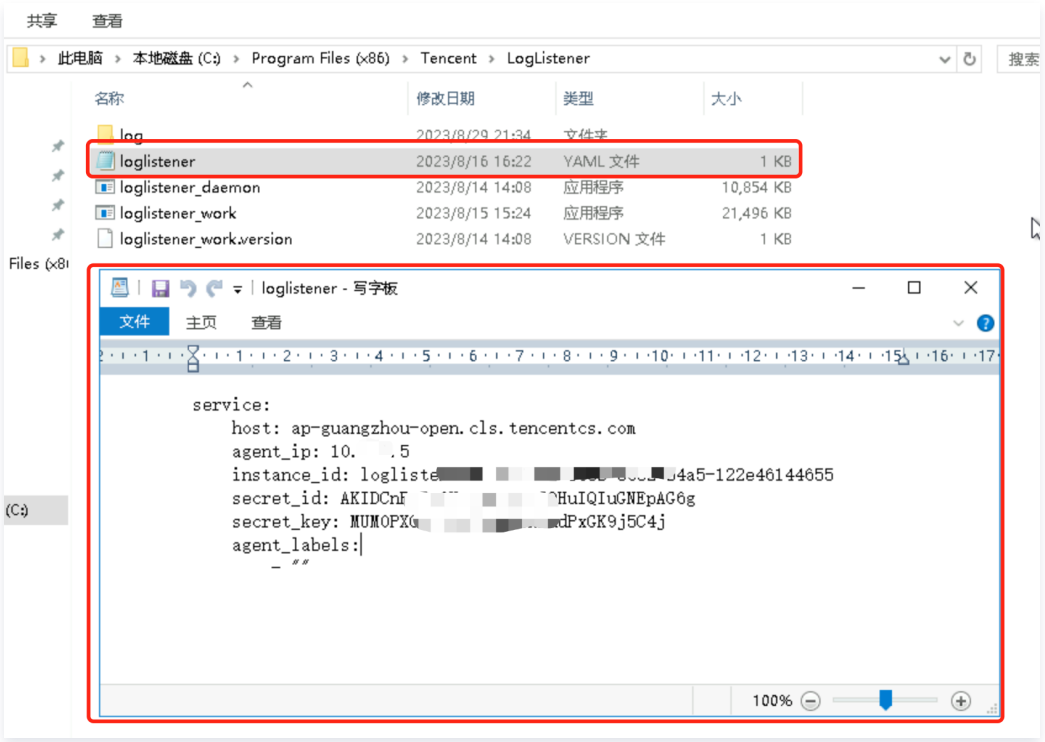
```
vim loglistener.conf
```

```
<root>
    proxy_host = ap-guangzhou-open.cls.tencentcs.com
    proxy_port = 
    secret_id = 
    sec
    # This switch use to enable
ted, default false. (
true/false)
    encryption =
    group_ip = 10.0.0.9
    # define loglistener by labels, separated by ,
    group_label =
    instance_id = loglistener-d69d2c27-d2f4-d584-c45d-954cc83d8489
    pos_file = data/pos.dat
    checkpoint_cache_file = data/checkpoint_cache
    checkpoint_window_size = 1024
    # Max file break points. Warn: modify this value, history breaskpoints will lost.
    max_file_breakpoints = 8192
```

其中 group_label 则为 Loglistener 的机器标识，请确保该标识与机器组中配置的一致。

Windows

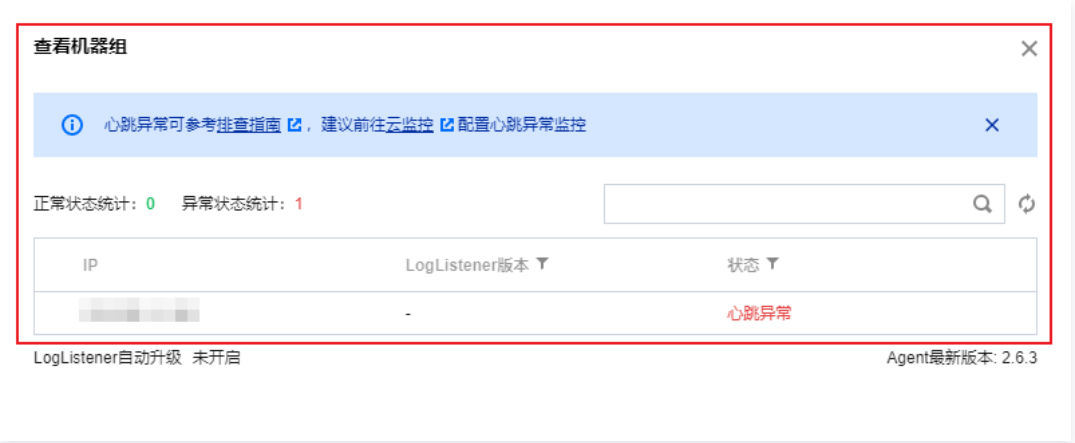
以安装路径 `C:\Program Files (x86)\Tencent\LogListener` 为例，在安装目录下找到 LogListener 配置文件 loglistener.yaml，并用任意编辑器打开，如下图：



其中 agent_label 则为 Loglistener 的机器标识，请确保该标识与机器组中配置的一致。

机器组机器状态异常问题

完成机器组配置后，在机器组管理页中，点击机器组名称可以查看与该机器组关联的机器心跳。状态正常说明机器心跳正常上报，若不为正常（如下图），则说明该机器上的 LogListener 存在异常。



心跳异常

当状态为心跳异常时， 说明 LogListener 与日志服务后端连接中断，会导致 LogListener 无法正常上传日志。您可通过以下方式进一步排查连接中断的原因：

> **说明**
> 仅适合于 LogListener 2.2.4 及以上版本，其他请参见 低版本 LogListener 异常状态排查 。

1. 使用 LogListener 快速诊断工具

LogListener 快速诊断工具可以快速诊断 LogListener 是否启动、心跳是否正常、配置拉取是否正常。
在命令行下执行如下指令：

```
/etc/init.d/loglistenerd check
```

若 LogListener 运行正常，诊断工具返回的结果如图所示：

```
[root@VM_30_69_centos etc]# sudo /etc/init.d/loglistenerd check
[OK] loglistener is running ok
[OK] check loglistener hearbeat ok
group ip:
host:ap-chengdu.cls.myqcloud.com
port:80
gethostbyname ip:
[OK] check loglistener config ok
{"logconf":[],"needupdate":false}
```

LogListener 进程异常
如果出现如下图所示 “[ERROR] loglistener is not running” 字样，表示 LogListener 没有启动。执行 `/etc/init.d/loglistenerd start` 启动，更多操作指令参见 LogListener 常用操作指令 。

```
]# /etc/init.d/loglistenerd check
[ERROR] loglistener is not running
]#
[root@VM-0-7-centos ~]#
```

LogListener 心跳异常
如果出现如下图所示 “[ERROR] check loglistener heareat fail” 字样，表示 LogListener 心跳异常。

```
]# /etc/init.d/loglistenerd check
[ERROR] check loglistener heareat fail
]#
~]#
```

引起 LogListener 心跳异常的原因有很多，最常见的情况有：

- 网络异常
  执行以下命令检查网络环境是否连通，命令中的 “cls domain name” 为 CLS 服务域名，请参见 可用地域 文档填写。

```
telnet <cls domain name> 80
```

- 密钥信息错误
  检查 LogListener 密钥信息是否正确，进入到 LogListener 安装目录执行如下命令。

  > **说明：**
  > 若无特殊指定，LogListener 安装目录通常为 `/usr/local/loglistener` 。

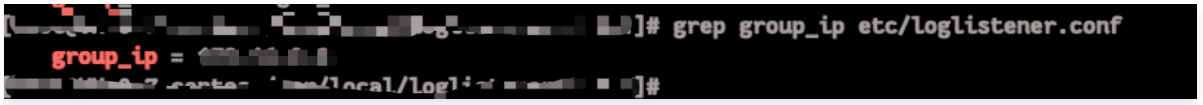
```
grep secret etc/loglistener.conf
```

2. 检查机器组 IP 配置

检查机器组所添加的 IP 地址是否为 LogListener 安装过程中获取的 IP 地址。进入到 LogListener 安装目录执行如下命令检查 LogListener 配置的 IP 地址：

> **说明：**
> 若无特殊指定，LogListener 安装目录通常为 `/usr/local/loglistener` 。

```
grep group_ip etc/loglistener.conf
```



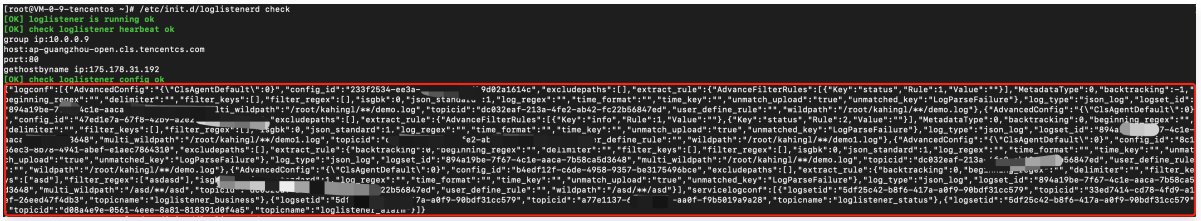
登录 日志服务控制台，单击**机器组管理**，查看机器组配置的 IP 地址，机器组所配置的 IP 地址必须与 LogListener 获取的 IP 地址完全一致。

目录超限

当状态为目录超限时，说明 LogListener 监听的采集目录数量已超过规格限制5000，会导致 LogListener 部分超出部分的目录的日志无法被采集。您可通过以下方式进行修复：

1. 执行以下命令查看当前 LogListener 关联的采集配置。

```
/etc/init.d/loglistenerd check
```



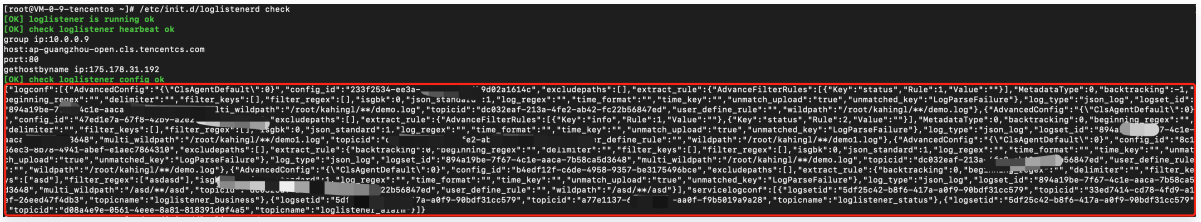
2. 以上返回结果中，logconfig 数组包含了当前 LogListener 的所有采集配置，您可针对每个采集配置，检查 wildpath 是否可能模糊命中大量无需采集的目录，例如 `/root/**/error.log` 这类跟目录的模糊路径。
3. 找到目标可优化的采集配置后， 从步骤1的返回数据中查看该采集配置关联的 topicid，并基于该 topicid 前往 CLS 控制台找到目标日志主题，优化或删除该采集配置。

文件超限

当状态为文件超限时，说明 LogListener 监听的采集文件数量已超过规格限制10000，会导致 LogListener 部分超出部分的目录的日志无法被采集。您可通过以下方式进行修复：

1. 执行以下命令查看当前 LogListener 关联的采集配置。

```
/etc/init.d/loglistenerd check
```



2. 以上返回结果中，logconfig 数组包含了当前 LogListener 的所有采集配置，您可针对每个采集配置，检查 wildpath 是否可能模糊命中大量无需采集的文件，例如 `/root/**/*.log` 这类从根目录模糊匹配任意名称的日志文件。
3. 找到目标可优化的采集配置后，从步骤1的返回数据中查看该采集配置关联的 topicid，并基于该 topicid 前往 CLS 控制台找到目标日志主题，优化或删除该采集配置。

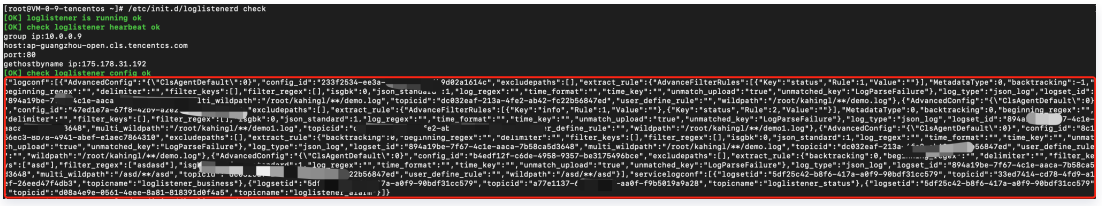
CPU 超限

当状态为 CPU 超限时，说明 LogListener 所占用 CPU 已超过预先设置的阈值。您可通过以下方式进行排查与修复：

优化目标采集目录

1. 执行以下命令查看当前 LogListener 关联的采集配置。

```
/etc/init.d/loglistenerd check
```

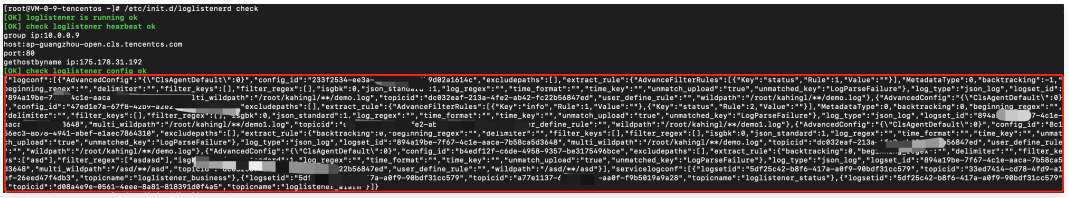


2. 以上返回结果中，logconfig 数组包含了当前 LogListener 的所有采集配置，您可针对每个采集配置，检查 wildpath 是否可能模糊命中大量无需采集的文件，例如 `/root/**/*.log` 这类从根目录模糊匹配任意名称的日志文件。
3. 找到目标可优化的采集配置后，从步骤1的返回数据中查看该采集配置关联的 topicid，并基于该 topicid 前往 CLS 控制台找到目标日志主题，优化或删除该采集配置。

优化正则表达式

1. 执行以下命令查看当前 LogListener 关联的采集配置。

```
/etc/init.d/loglistenerd check
```



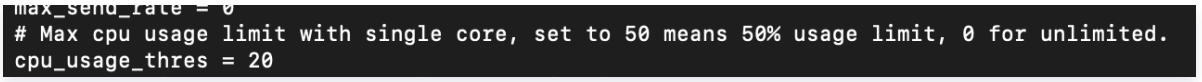
2. 以上返回结果中，logconfig 数组包含了当前 LogListener 的所有采集配置，您可针对每个带有 fullregex 或 multi_fullregex 字段的采集配置，检查正则是否过于复杂。
3. 找到目标可优化的采集配置后，从步骤1的返回数据中查看该采集配置关联的 topicid，并基于该 topicid 前往 CLS 控制台找到目标日志主题，优化或删除该采集配置。

提升 LogListener CPU 占用的限制

1. 以 LogListener 安装目录为 `/usr/local/loglistener` 为例，执行以下命令打开 LogListener 配置文件：

```
vim  etc/loglistener.conf
```

2. 在 LogListener 配置文件中， 找到 `cpu_usage_thres` 并按需提升 CPU 利用率限制， 如下图：



3. 完成配置后，保存并 重启 LogListener 。

提升 CPU 资源配额

K8s 场景下，请确保 `DeamonSet:tke-log-agent` 下 `Pod: loglistener` 的 CPU 配额充足。

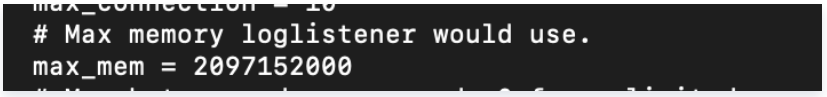
内存超限

当状态为内存超限时，说明 LogListener 所占用内存已超过预先设置的阈值。您可通过以下方式进行修复：

1. 以 LogListener 安装目录为 `/usr/local/loglistener` 为例，执行以下命令打开 LogListener 配置文件：

```
vim  etc/loglistener.conf
```

2. 在 LogListener 配置文件中，找到 `max_mem` 并按需提升内存占用限制，如下图：



3. 完成配置后，保存并 重启 LogListener 。

鉴权失败

确认密钥有效

当状态为鉴权失败时，说明 LogListener 中配置的密钥错误或已失效。您可通过以下方式进行修复：

1. 以 LogListener 安装目录为 `/usr/local/loglistener` 为例，执行以下命令打开 LogListener 配置文件：

```
vim  etc/loglistener.conf
```

2. 在 LogListener 配置文件中，找到 `secrete_id` 和 `secrete_key` 并判断是否正确有效，若无效则修改为正确的密钥信息，如下图：

3. 完成配置后，保存并 重启 LogListener 。

服务器时间正确

请确保服务器时间准确，否则可能导致鉴权失败。

LogListener 常见问题

最近更新时间：2024-10-15 18:50:51

为何单个文件不支持上报至多个 topic?

LogListener 采集程序的策略是，单个文件，只能上传至一个 topic。
例如，您有 topicA 和 topicB 两个文件，并对这两个文件进行如下设置。

- 设置 topicA 的采集路径为：`/data/log/**/*.log`
- 设置 topicB 的采集路径为：`/data/log/test/**/*.log`、`/data/log/**/test*.log`、`/data/log/**/*.log`，或者其他与 topicA 采集路径类似的路径。

此场景的配置，虽对应到两条采集路径交集的文件，但只会将数据上传至其中一个 topic 中。因此，我们建议针对不同的日志主题，应承载不同业务类型的日志，并在设置采集配置路径时，尽量精确配置信息。如果仍需要将某一个文件上传至不同的 topic，可使用软链接实现。对同一个采集目标创建不同的软链接，不同的 topic 分别采集不同的软链接路径/文件。

如何进行采集路径的设置?

目前采集路径的配置为：路径前缀+ “/**/” +通配文件名的形式。例如：`/data/log` + `/**/` + `*.log` ==> `/data/log/**/*.log`。
由于设置通配采集路径时，需要将采集路径（前缀部分）设置的尽量准确，从而使采集器能够更有效率地提供服务。

> **注意：**
> 如果采集路径前缀部分设置不准确，可能会导致采集路径匹配到的路径数量巨大，从而造成采集器进入异常状态，无法工作。
> 例如，采集路径设置为：//**/*.log，其前缀部分为"/"，这种情况下采集程序会扫描整个根目录，导致采集程序无法工作。

推荐的日志轮转方案是什么?

对于日志的轮转，推荐方式为轮转后的文件名，不要被采集通配路径覆盖到。
例如，配置的采集路径是 `/var/log/xxxx/**/*.log`，需要采集的日志是 `test.log`。当 `test.log` 轮转成 `test.2021-07-13.4.log` 时，LogListener 能够识别 `test.2021-07-13.4.log` 是 `test.log` 的轮转文件，对它仍按照 `test.log` 来标记。因此，LogListener 保存的位点文件中，没有 `test.2021-07-13.4.log` 这个文件的采集记录。
而当 LogListener 重启后，LogListener 会按照 `/var/log/xxxx/**/*.log` 去扫描文件，并发现 `test.2021-07-13.4.log` 文件符合匹配规则，且没有采集记录，是一个新文件，然后重新采集。
所以建议，**采集通配路径不要匹配到轮转文件，避免造成重启后重新采集轮转文件**。
我们推荐的日志轮转方案是，如果您需要采集 `test.log`，建议将轮转后的文件名命名为 `test.log.2021-07-13.xxx`，可以有效不被 `*.log` 覆盖到。

LogListener 升级说明?

在 LogListener 迭代过程中，采集路径的接口参数做过变更，比较老的版本（ver< 2.2.8）设置的采集路径，在新版本中不再被支持。
因此，如需对存量老版本（ver < 2.2.8）进行升级，在采集程序升级之后，需要在控制台重新以通配路径的方式，再次设置采集路径。

LogListener 采集配置如何使用正则采集模式?

在控制台设置采集配置时，如果选择正则相关的采集模式，控制台虽提供正则 kv 提取小工具，但此工具暂不提供对中文内容的正则自动生成功能。如需对中文文本进行正则提取，可以自行编写正则表达式，在控制台进行验证，或者使用其他第三方工具进行验证。

初次使用 LogListener 采集器接入时，发现无日志上传，怎么办?

可能是采集器配置不正确导致，常见情况如下：

- 配置的服务端域名不匹配，采集器拉取不到当前地域的采集配置，无采集业务进行。
- 采集器加入了 IP 机器组，但是采集器中又配置了标签 label 信息，导致在当前地域拉取不到采集配置，无采集业务进行。
- 采集器中配置的 secret ID/KEY 不正确，或者权限不足，导致无法上传日志。
- 环境问题（如 VPC 子网内，外网未开启），如果配置了跨地域上传，是不生效的，采集器实际上还是与本地域服务端进行通信。
- 通常这种情况下，可以登录采集器所在机器，进入采集器安装目录，并执行 `./bin/check` 命令，检查如下内容：
- 域名是否正确。
- 心跳上报是否正常。
- 采集配置是否正确拉取。

机器组使用混用，导致采集不采集，怎么办?

目前机器组分为两类，其使用方法相互独立：

- IP 机器组，机器 IP 需要在控制台上手动加入机器组，对应机器上 loglistener.conf 的group_label 需为空。

- 标签机器组，控制台设置机器组标签，对应机器上 loglistener.conf的group_label 需要设置为相同的标签。

如上两种用法不兼容，如果混合使用，采集机器将拉取不到正确的采集配置，造成不采集的现象。

LogListener 的采集策略是什么?

一堆文件排队进行 LogListener 采集时，队首文件先采集，且要求其在某个时刻读取到文件尾才会让出队首位置。即：不是每个文件在单位时间内，都能均等的享受采集资源。
当单个文件始终写入大于采集速度，且采集速度慢导致始终消费不到文件最新位置时，会出现某个文件长时间或一直霸占采集资源，从而导致其他的文件无法进入采集流程。

Topic 采集阻塞严重怎么办?

在某段时间内，如果单个文件的产生速度大于采集速度，LogListener 会持续一直在采集这一个文件，陷入了对其他文件的采集阻塞场景。

过滤器的规则是什么?

过滤器的规则是匹配后采集，而不是匹配后丢弃。对于未能匹配的日志，LogListener 将不会进行采集。

如何使用非 root 权限启动 LogListener?

建议用户使用 root 权限安装启动。如需在非 root 权限下使用 LogListener，可以参见 设置非 root 权限启动 LogListener 。

如何对 LogListener 的进程进行绑核?

使用 taskset 工具进行绑核， `taskset -cp ${cpu number} ${pid>}` 。

如何处理 LogListener 占用内存过高，控制资源的使用?

- 建议升级到最新 LogListener 版本 ，并设置 `memory_tight_mode = true` 。
- 使用 CGroup 限制 CPU 和内存使用。

LogListener 是否支持软链接方式采集?

LogListener 低于2.3.0版本不支持监听软连接方式的日志文件和 NFS、CIFS 等共享文件目录上的日志文件，以上版本均可支持。

LogListener 可以向多个日志主题上传数据吗?

- LogListener 可以为同地域的多个日志主题采集数据，但不支持为异地多个日志主题采集。
- 同一个日志文件只支持采集到一个主题。

LogListener 初始化的时候是否可以自动加入机器组?

标识机器组支持，参见文档 管理机器组 。

LogListener 日志上传策略是什么?

- 缓存的日志量超过4M。
- 缓存的日志条数超过10000条。
- 读到文件末尾。

LogListener 支持的最大性能是多少?

- 单行全文日志最大处理能力为115MB/s。
- 多行全文日志最大处理能力为40MB/s。
- JSON 格式日志最大处理能力为25MB/s。
- CSV 格式日志最大处理能力为50MB/s。
- 完全正则格式日志最大处理能力为18MB/s (和正则的复杂度有关)。

服务器更换 IP 地址后，LogListener 应该如何适配?

- 若服务器通过机器标识绑定机器组，用户无需变更 LogListener 配置。若服务器 IP 需要频繁变更，建议用户使用 机器标识 配置机器组。
- 若服务器通过 IP 地址绑定机器组，用户需要完成以下配置变更：
  a. 修改配置文件中 group_ip 选项，填入变更后的 IP 地址，例如：

```
sed -i '' "s/group_ip *=.*/group_ip = ${group_ip}/" etc/loglistener.conf
```

b. 重启 LogListener。

```
/etc/init.d/loglistenerd restart
```

c. 如果使用的是 IP 机器组，登录 日志服务控制台，在左侧导航栏中，单击**机器组管理**，修改该服务器绑定的机器组配置，使用新 IP 替换原机器 IP 地址并确定。

LogListener 安装异常问题

最近更新时间：2024-10-18 10:20:11

如何安装及使用日志服务 LogListener，请参见 LogListener 安装指南 文档，并了解 LogListener 机制 。

哪些原因可能导致无法正确安装 LogListener?

1. 内核版本不是64位。
2. 安装方式出错。
3. 最新特性功能依赖较高版本 LogListener。

如何处理 LogListener 安装异常?

1. 确认内核版本。
   LogListener 安装目录下的 bin 目录中的可执行文件只支持 Linux 64位内核，执行命令 `uname -a` ， 确认内核版本是否为 x86_64。
2. 确认安装执行命令是否正确。
   具体请参见 LogListener 安装指南 文档进行操作。
3. 确认 Loglistener 版本。
   日志服务最新特性可能依赖新版 LogListener，若确认是使用新特性异常，请下载 LogListener 最新版本。LogListener 下载及详细安装步骤请参见 LogListener 安装指南 。
4. 验证 LogListener 成功安装。
   参见如何使用 LogListener 快速诊断工具 检查 LogListener 进程、心跳和采集配置拉取是否正常。

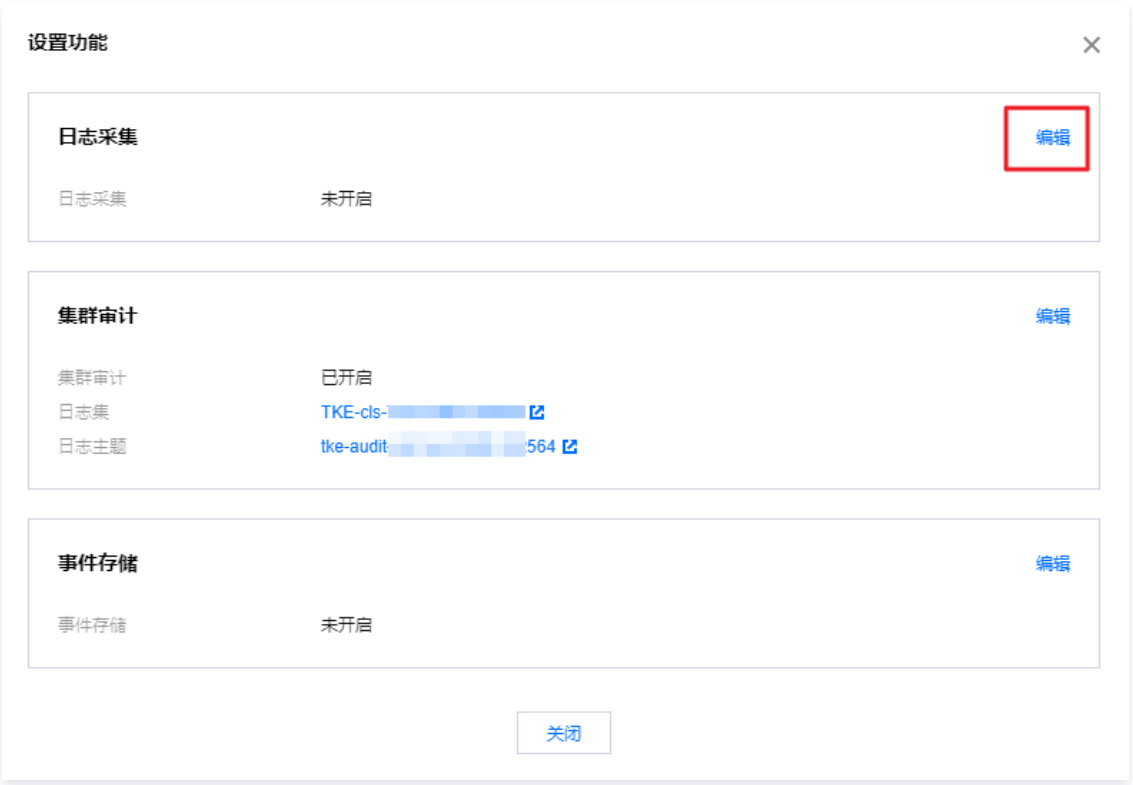
容器日志采集常见问题

最近更新时间：2025-04-24 10:34:12

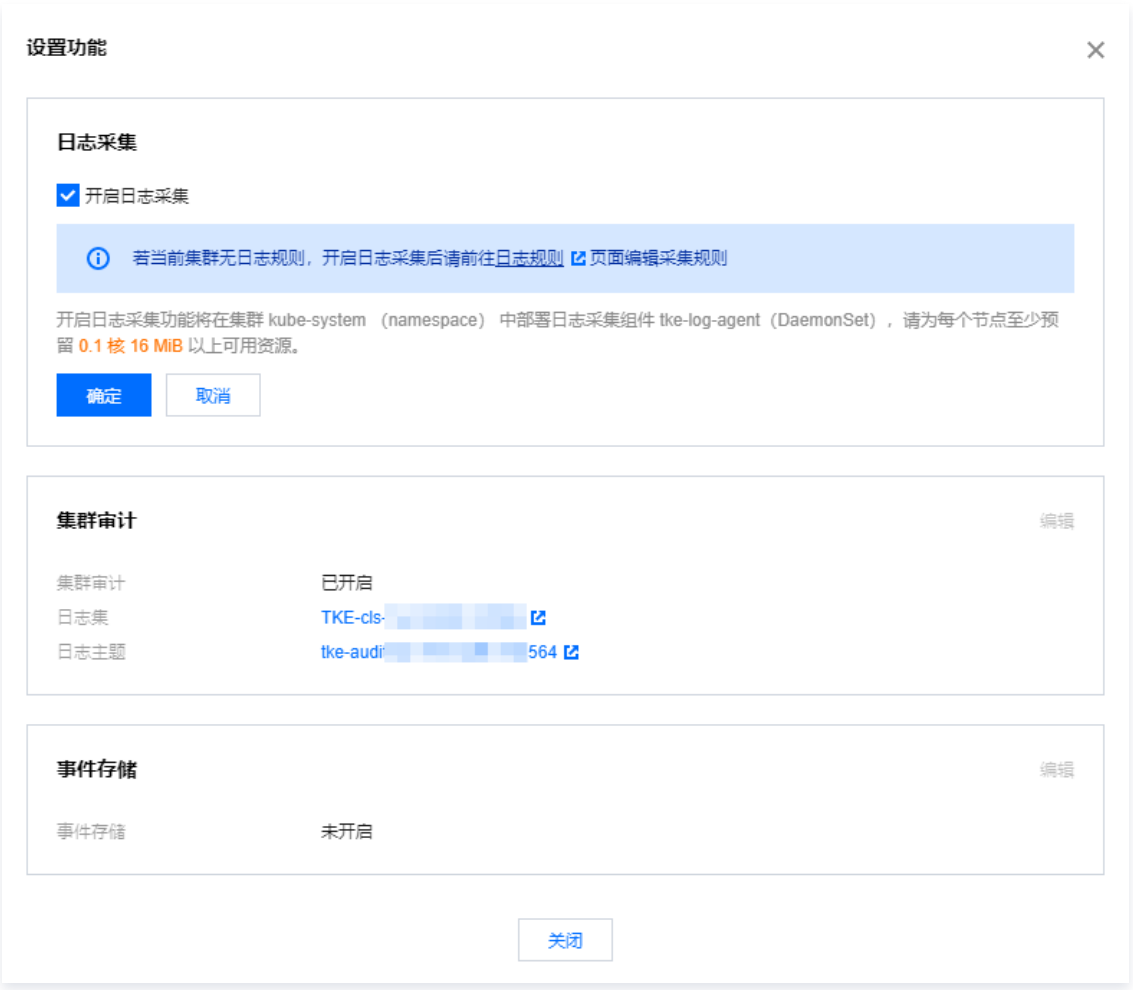
安装与升级相关

如何在 TKE 集群中部署日志采集组件?

1. 登录 容器服务控制台 。
2. 在左侧导航栏中，单击**运维功能管理**，进入功能管理页面。
3. 找到待操作的集群，单击**设置**。
4. 在弹出的窗口中，单击**日志采集**栏的**编辑**。



5. 勾选**开启日志采集**，单击**确定**。



6. 单击**关闭**。

如何在 TKE 集群中升级日志采集组件?

1. 登录 容器服务控制台 。
2. 在左侧导航栏中，单击**运维功能管理**，进入功能管理页面。
3. 找到可升级组件的集群，单击**设置**。

4. 在弹出的窗口中，单击**日志采集**栏的**编辑**。

5. 单击**升级组件**。

网络与权限相关

云 API 域名不通，怎么办?

容器服务（Tencent Kubernetes Engine，TKE）日志采集组件和日志服务（Cloud Log Service，CLS）通信的组件 cls-provisioner 使用腾讯云 API 域名，需要保证域名的可连通性。如果存在组件部署失败等问题，并在日志中看到有如下图所示报错，即表示网络域名不通。

如上图所示，cls-provisioner 启动异常。通过查看日志发现，cls.internal.tencentcloudapi.com 域名不通。
在腾讯云上的机器，默认腾讯云的内外网域名都是联通的。通常导致此类问题的原因是 TKE 节点的 DNS 配置被修改过，您可通过如下两种方法进行修复：

- TKE 节点机器的 DNS 配置添加腾讯云默认 DNS。

- 如果宿主机上的 DNS 服务器是 core-dns，在 coredns 上添加腾讯云 DNS 解析即可。

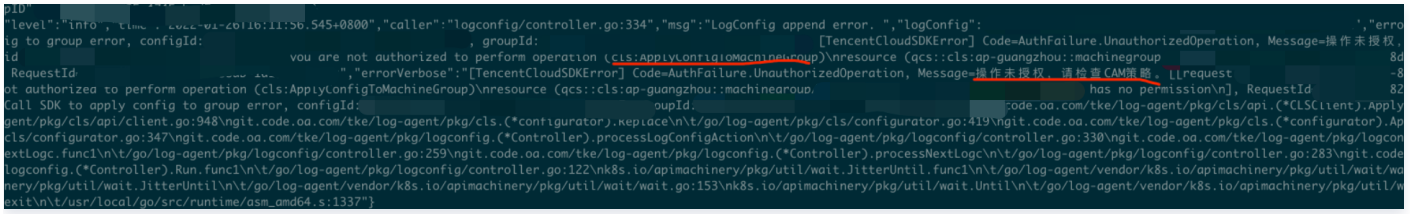
> **说明：**
> 建议在 TKE 集群中遇到域名不通的问题优先检查 TKE 节点 DNS 配置。

CLS 日志上传域名不通，怎么办？

日志上传的域名和云 API 域名不同，日志上传的域名为 `<region>.cls.tencentcs.com`（外网）和 `<region>.cls.tencentyun.com`（内网），更多详情请参考 域名 文档。
修复方案：
在集群节点机器打通域名的访问。

在 cls-provisioner 和 CLS 的通信中提示未授权，怎么办？

在 cls-provisioner 和 CLS 的通信中，一般会有类似如下的报错：



修复方案：
在创建 TKE 集群的账号下预设策略，即在 TKE_QCSRole 角色中关联 QcloudAccessForTKERoleInOpsManagement 策略。

采集相关

采集到 CLS 的日志被截断了，怎么办？

在某些情况下，用户日志的输出类型是标准输出，但采集到 CLS 的日志发现被截断了。因为 Docker 的默认日志存储 json-tool，对单行日志大小有限制，所以超过16K的日志会进行截断。
修复方案：
修改日志输出，单行日志打印不要超过16K。

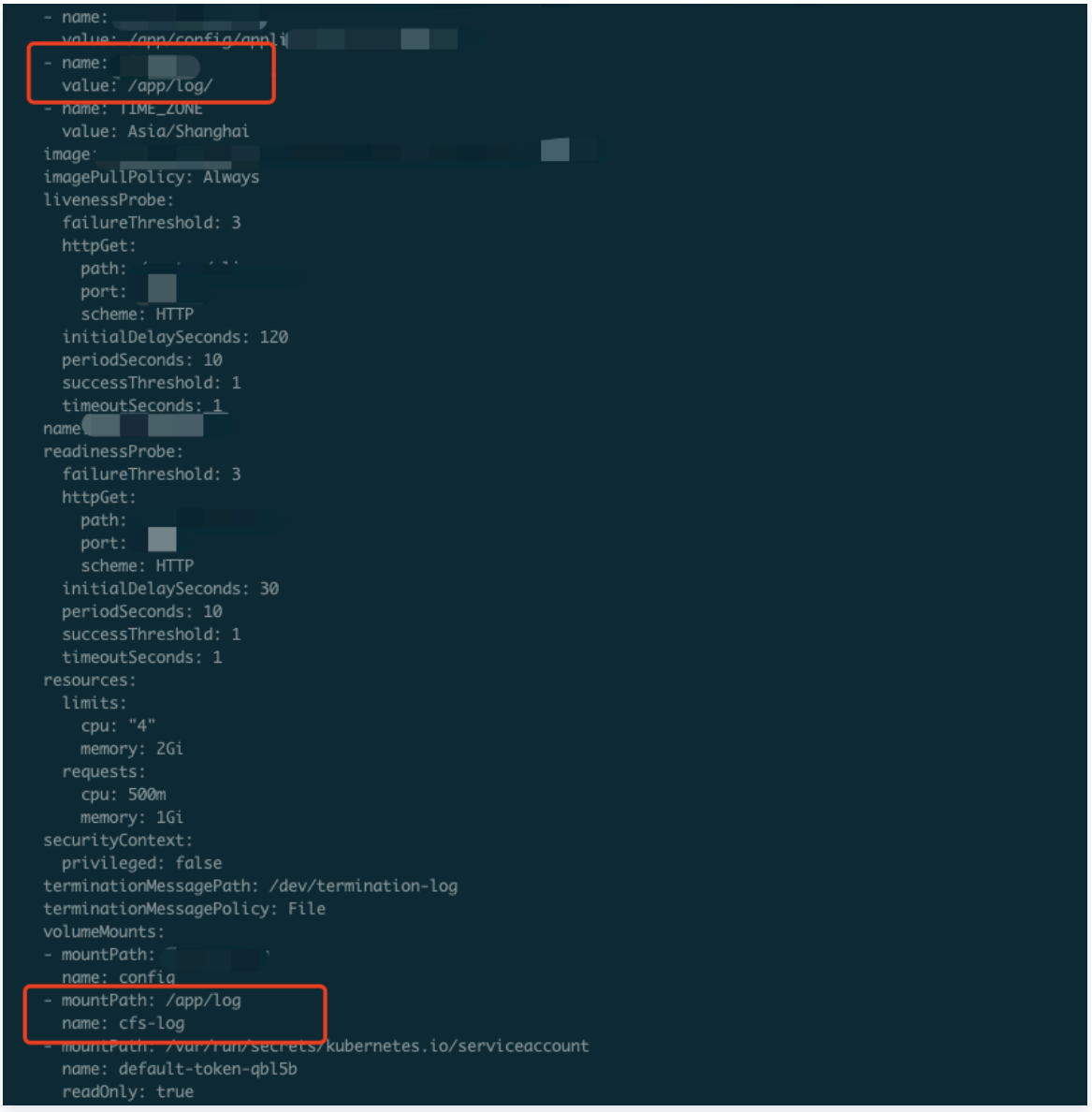
日志重复采集，怎么办？

用户通过 CLS 控制台检索，发现有些日志出现了重复采集。此时，可以优先检查日志输出的路径，确认日志是否输出到 PV/PVC 创建的持久化存储上。
如果日志输出到持久化存储上，当业务 Pod 重建时，会导致日志会被重新采集。可以使用如下命令，查看 Pod 的 yaml 定义：

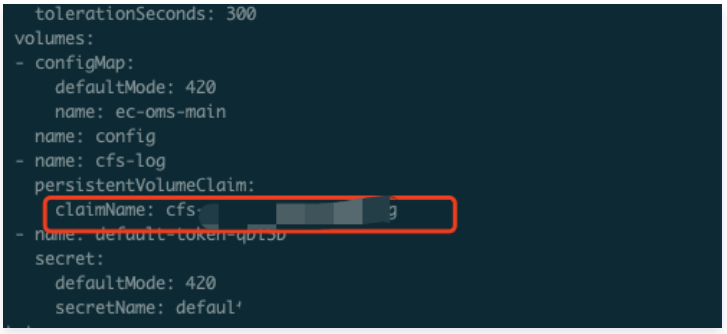
```
kubectl get pods <pod_name> -n <namespace> -o yaml | less
```

返回类似如下信息即表示日志输出到持久化存储上。

- 业务使用了 CFS，且 CFS 挂载到容器上。



- 使用了 CFS 声明。



修复方案：

- 如果日志不需要保存在持久化存储上，可以在容器集群中开启日志采集，即可将日志采集到 CLS 内。
- 如果需要将日志文件保存在持久化存储上，可以在 CLS 控制台中配置 LogListener 采集规则时，将采集策略修改为**增量**采集，但是增量采集不保证会采集到全部日志。

日志漏采集，怎么办？

Loglistener 当前不支持保存在 NFS 上的日志。Loglistener 获取文件的更新信息是通过订阅 Linux 内核事件来的，并不是主动去扫描目标文件。NFS 文件更新信息是在 NFS 服务端完成，无法在 client 本地的内核产生事件，因此无法被感知，所以 NFS 文件无法实时采集，会存在漏采集。

采集配置与 Pod 不匹配，怎么办？

您可以通过如下操作进行排查：

1. 使用如下命令，确认 Pod 的 label 是否与采集配置匹配。

```
kubectl get pods <pod_name> -n <namespace> --show-labels
```

   - 匹配，请执行下一步。
   - 不匹配，请对照正确内容修改采集配置。

2. 检查 Pod 的 workload（Deployment 或者 statefulsets 等）是否和采集配置匹配。

```
kubectl get pods -n <namespace> |grep testa
```

   - 匹配，请执行下一步。
   - 不匹配，请对照正确内容修改采集配置。

3. 使用如下命名，查看 Pod 的 yaml 定义，确认 container 的名字和采集配置中指定的容器名是否匹配。

```
kubectl get pods <pod_name> -n <namespace> -o yaml
```

   - 匹配，任务完成。
   - 不匹配，请对照正确内容修改采集配置。

采集路径不正确怎么办？

在采集容器文件或者采集宿主机文件的场景下，确认采集的目录路径正确，并且有符合采集规则的日志。

日志文件可以使用软链接（符号链接）吗？

容器文件的采集场景下，不支持匹配到的日志文件存在软链接的场景。
Kubernetes 场景下，CLS 的实现是解析容器文件在宿主机的位置。由于容器中的软链接目标指向的是容器内的路径，如果匹配到的采集文件存在软链接，将不能正确可达。
修复方案：

修改采集规则的路径和匹配文件，即采集日志文件实际所在路径和文件，避免匹配到软链接。

触发采集规格限制是什么？

由于 Loglistener 采集在容器场景下，资源受到限制，所以对监听的目录和文件有个数限制：

- 监控目录数：5000
- 监控文件数：10000

在采集容器文件或者采集宿主机文件场景下，可能会遇到此类问题。通常情况下，由于用户没有清理过期的日志文件，在查看 Loglistener 日志时，日志中有类似如下图信息：

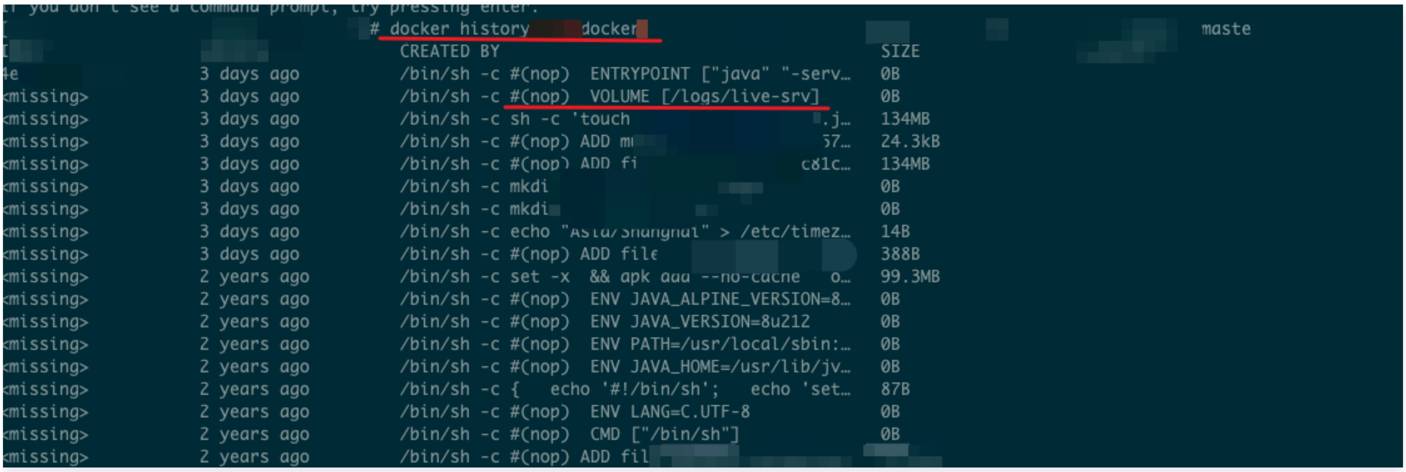
对于超过最大限制的文件和目录，Loglistener 将不会纳入监听，导致有些用户预期内的日志文件没有采集。
更多详情请参考 资源与性能限制 文档。
修复方案：
在日志目录下，使用 `tree` 命令，查看整个目录下当前的目录数量和文件数量是否达到 Loglistener 限制。

- 如果没有达到限制，在业务容器所在宿主机的 /var/log/tke-log-agent 下执行 `tree -L 5`，检查整体机器粒度是否达到限制。
  因为 Loglistener 的限制是机器粒度的，如果某一个容器文件数量没有达到阈值，但是可能整体机器粒度所有的容器文件达到限制。
- 如果达到限制，请将过期日志及时归档，减少 Loglistener 监控的目录和文件所消耗的资源。

在 Dockerfile 定义了 volume，怎么办？

- Docker 场景：使用 `docker history $image` 命令，查看镜像在构建信息。
- Containerd 场景：使用 `crictl inspecti $image` 命令，查看镜像在构建信息。

返回如下信息，可以看出，在 Dockerfile 中，用户自定义了一个 volume /logs/live-srv，刚好是日志所在目录。该操作会干扰日志采集组件找到正确的日志文件。



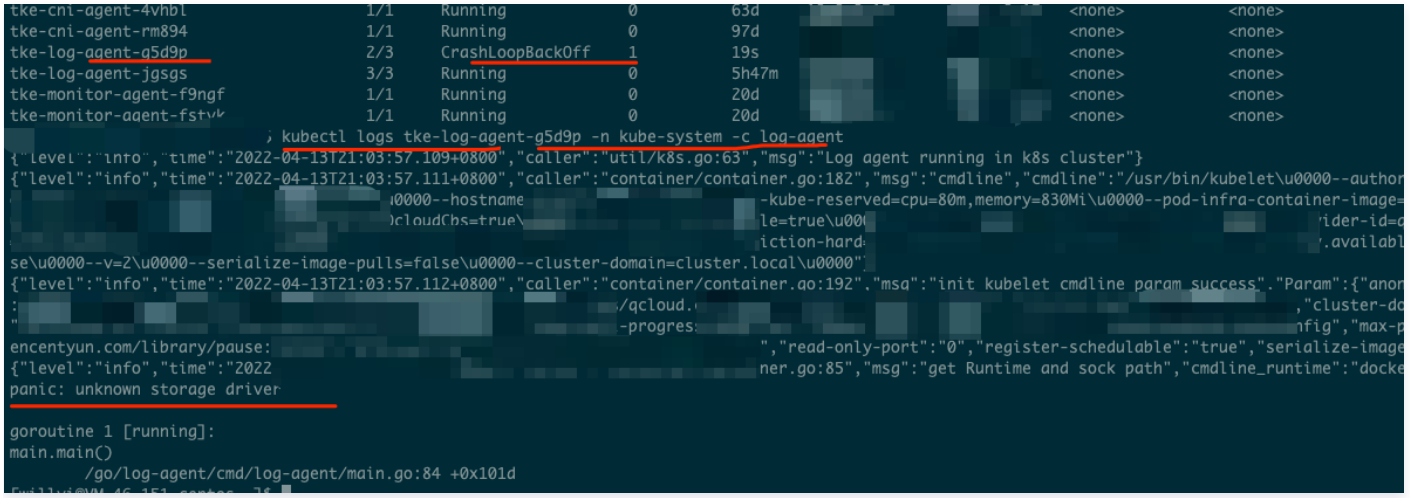
修复方案：

- 修改 Dockerfile，去掉 volume，然后重新构建镜像，重新部署服务。
- 修改服务日志写入目录，不要写入 Dockerfile 中定义的 volume 路径中。

其他问题

容器引擎类型识别错误，怎么办？

在 Docker 场景下，一些场景会触发老版本的一些 bug，导致日志采集组件不能正常启动，出现 panic 日志。
其主要原因是由于用户自定义了 TKE 集群节点的 Docker 配置，从而导致出现如下图所示的错误：



修复方案：

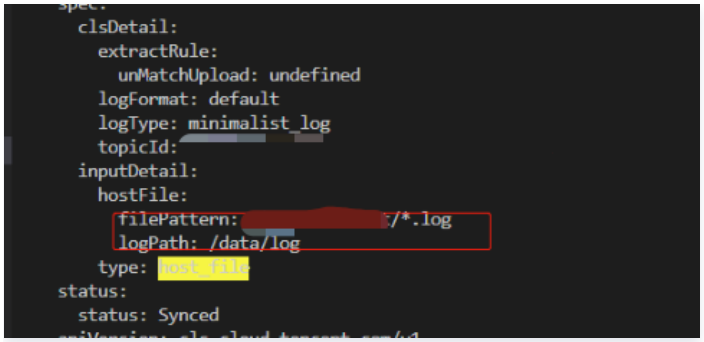
- 在 /etc/docker/daemon.json 配置中，添加 `"storage-driver": "overlay2"` 。如下图所示：

```
[root@VM-0-5-centos ~]# cat /etc/docker/daemon.json
{
  "bridge": "none",
  "debug": false,
  "default-runtime": "runc",
  "exec-opts": [],
  "exec-root": "",
  "graph": "/var/lib/docker",
  "group": "",
  "insecure-registries": [],
  "ip-forward": true,
  "ip-masq": false,
  "iptables": false,
  "ipv6": false,
  "labels": [],
  "live-restore": true,
  "log-driver": "json-file",
  "log-level": "warn",
  "log-opts": {
    "max-file": "10",
    "max-size": "100m"
  },
  "max-concurrent-downloads": 10,
  "registry-mirrors": [
    "https://mirror.ccs.tencentyun.com"
  ],
  "runtimes": {},
  "selinux-enabled": false,
  "storage-driver": "overlay2",
  "storage-opts": [
    "overlay2.override_kernel_check=true"
  ]
}
```

- 在 TKE 控制台升级日志采集组件版本。因为新版本采集组件已经修复此问题，无需修改 Docker 配置。

filePattern 设置了子目录，怎么办?

如下图所示，用户在 filePattern 字段参数中设置了子目录。此操作会导致日志不能正常采集。



修复方案：

在 logPath 参数中设置日志文件目录，filePattern 中只设置文件类型参数，不设置子目录。

自建 K8S 日志采集排查指南

最近更新时间：2024-05-29 11:46:11

如何对自建 K8S 日志采集进行排查?

按照自建 K8S 集群安装 LogListener 部署完成后，就可以通过创建 LogConfig 或者通过控制台去设置采集配置，开始日志采集了。
如果出现日志采集异常，首先按照以下流程进行自查。

1. 确认 logconfig 状态

- 执行 `kubectl get logconfig` 命令查看集群所有的采集配置。
- 执行 `kubectl get logconfig xxx -o yaml` 命令查看具体某一个采集配置。
- 查看 logconfig 同步的状态，status 非 Synced 状态都是异常的，异常信息会在 reason 里面，正常都是 success 的状态。
  如上 logconfig 的状态同步是成功的，那么采集异常的原因就是其他方面的。如下图所示：

```
[root@VM-48-16-centos ~]# kubectl get logconfig [redacted] -o yaml
apiVersion: cls.cloud.tencent.com/v1
kind: LogConfig
metadata:
  creationTimestamp: "2022-12-09T07:02:41Z"
  generation: 2
  name: willyi-test
  resourceVersion: "33746388884"
  selfLink: /apis/cls.cloud.tencent.com/v1/logconfigs/willyi-test
  uid: [redacted]
spec:
  clsDetail:
    [redacted]
    extractRule:
      backtracking: "0"
      jsonStandard: "true"
      unMatchUpload: "false"
    logFormat: default
    logType: minimalist_log
    maxSplitPartitions: 0
    storageType: ""
    topicId: [redacted]
  inputDetail:
    containerStdout:
      container: log-agent
      containerOperator: notin
      includeLabels:
        app: cls-provisioner,tke-log-agent
      namespace: kube-public,kube-system
      nsLabelSelector: ""
    type: container_stdout
status:
  code: success
  reason: success
  status: Synced
[root@VM-48-16-centos ~]#
```

想要进一步了解同步错误的原因，可以查看 cls-provisioner 的日志。

2. 查看 cls-provisioner 日志

- 执行以下命令确定 cls-provisioner 的 Pod 名称。

```
kubectl get pods -n kube-system -o wide |grep cls-provisioner
```

- 获取到 cls-provisioner 的 Pod 名称后， 执行以下命令查看 cls-provisioner 日志，并定位错误的具体原因。

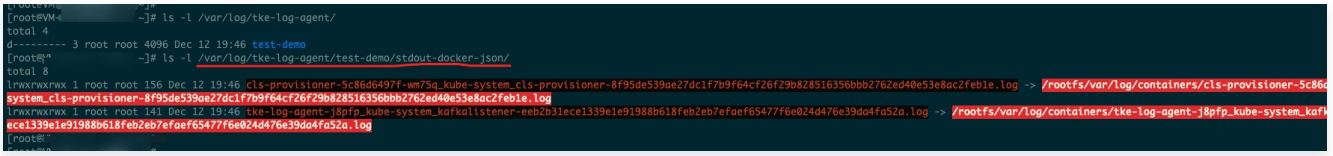
```
kubectl logs cls-provisioner-xxx -n kube-system
```

> ⚠ **注意**
> cls-provisioner 组件的作用是和 CLS 服务端通信，将 logconfig 采集配置经过转换，同步到 CLS 服务端，这样采集器才能从服务端获取到采集配置，进而进行正常日志采集。

3. 查看采集端日志

如果采集配置同步正常，但是日志还是采集有异常，可以具体看下采集端的相关日志。

- 查看软连是否建立成功。
  我们以采集标准输出为例：
  会在/var/log/tke-log-agent/<采集配置名称(logconfig 名称)>/stdout-docker-json 下创建需要采集的 Pod 的标准输出日志的软连，创建好之后才能正常采集。



我们是以 Docker 为例的，如果 runtime 是 containerd，那么路径是/var/log/tke-log-agent/<采集配置名称(logconfig 名称)>/stdout-containerd。
采集 contianer file 的软连建立方式如下：
/var/log/tke-log-agent/<采集配置名称(logconfig 名称)>/
如下图：

确认按照上面示例的软连是否建立 OK，如果未建立，则是有异常的。如果建立成功，则要继续看下采集器 loglistener 的日志。

- 执行以下命令，查看采集器 loglistener 日志。

```
kubectl get pods -n kube-system -o wide |grep tke-log-agent
```

首先找到日志采集异常 Pod 对应宿主机上的 tke-log-agent 的 Pod名称，然后执行以下命令查看 loglistener 日志。

```
kubectl logs tke-log-agent-xxx -n kube-system -c loglistener
```

如果已经采集到了 topic，但是检索不到，可以先看下是否打开 topic 的全文索引。

采集配置常见问题

最近更新时间：2024-05-29 14:28:51

完整日志内容如何通过完全正则采集配置？

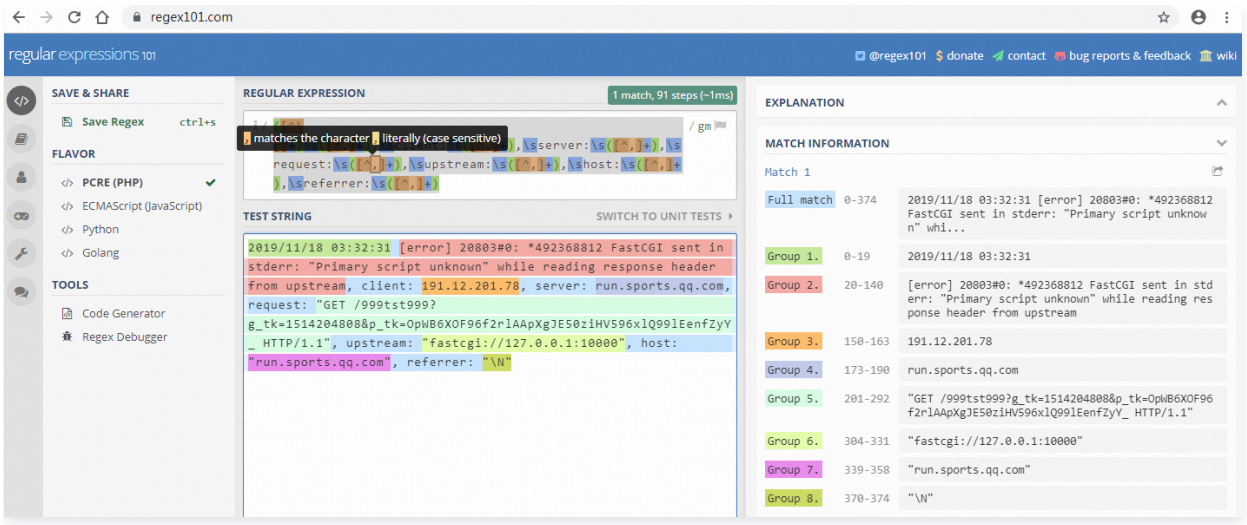
原始日志中，完整的日志格式如下：

```
2019/11/18 03:32:31 [error] 20803#0: *492368812 FastCGI sent in stderr: "Primary script unknown" while reading response header from upstream, client: 191.12.201.78, server: run.sports.qq.com, request: "GET /999tst999?g_tk=1514204808&p_tk=OpWB6XOF96f2rlAApXgJE50ziHV596xlQ99lEenfZyY_ HTTP/1.1", upstream: "fastcgi://127.0.0.1:10000", host: "run.sports.qq.com", referrer: "\N"
```

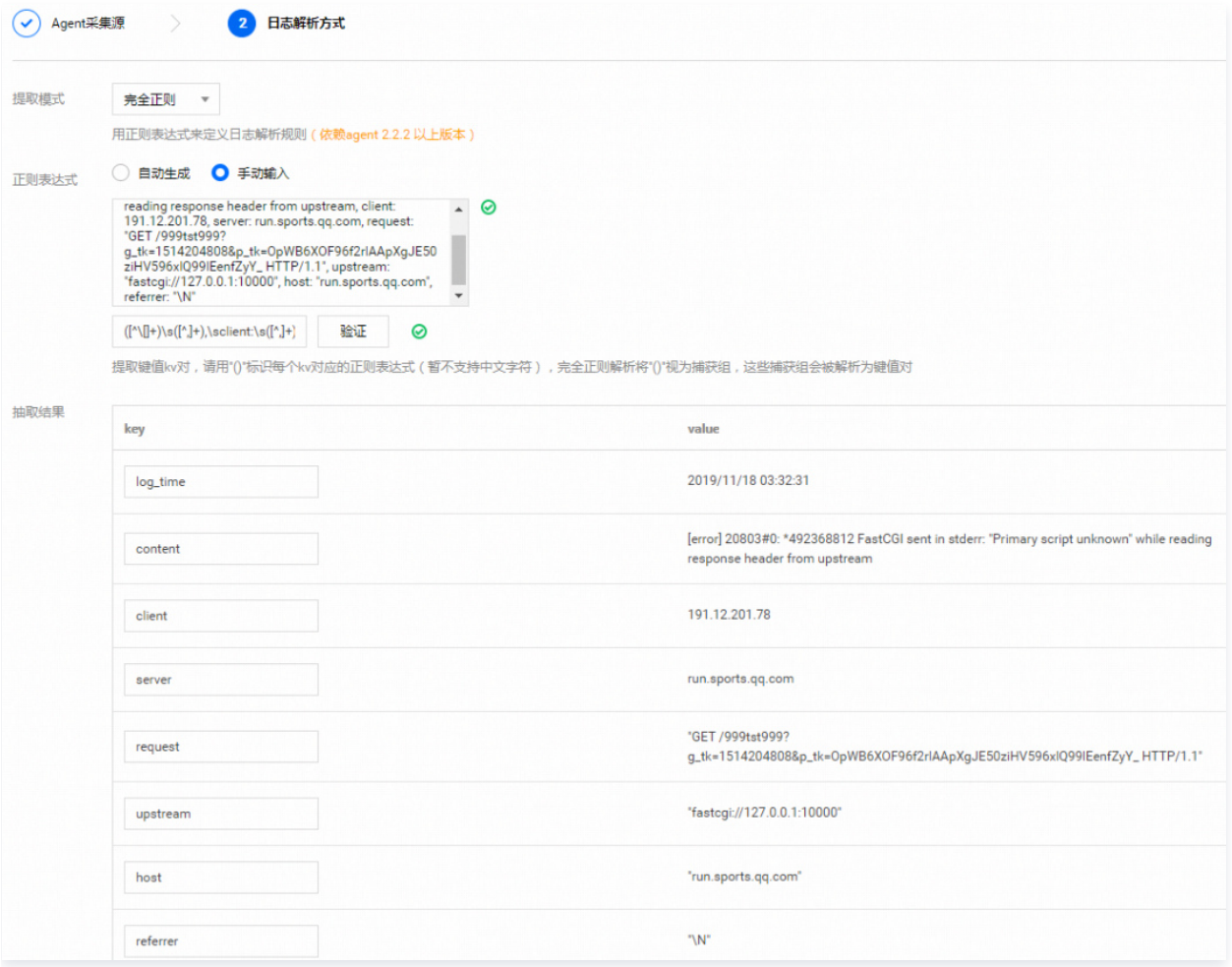
通过如下正则表达式，可以提取出对应的字段：

```
([^\
[]+)\s([^,]+),\sclient:\s([^,]+),\sserver:\s([^,]+),\srequest:\s([^,]+),\supstream:\s([^,]+),\shost:\s([^,
]+),\sreferrer:\s([^,]+)
```

原始日志格式化的字段如下（推荐使用 regex 构建自定义的正则表达式）：



各字段依次命名为：log_time、content、client、server、request、upstream、host、referrer。



日志内容字段部分缺失如何采集?

但是在某些情况下，日志中的 upstream、referrer 字段会缺失。
例如，无 upstream 有 referrer：

```
2019/11/18 04:02:38 [error] 20802#0: *492391323 access forbidden by rule, client: 45.71.63.206, server: admin.sports.qq.com, request: "GET /index HTTP/1.1", host: "admin.sports.qq.com", referrer: "http://admin.sports.qq.com/index"
```

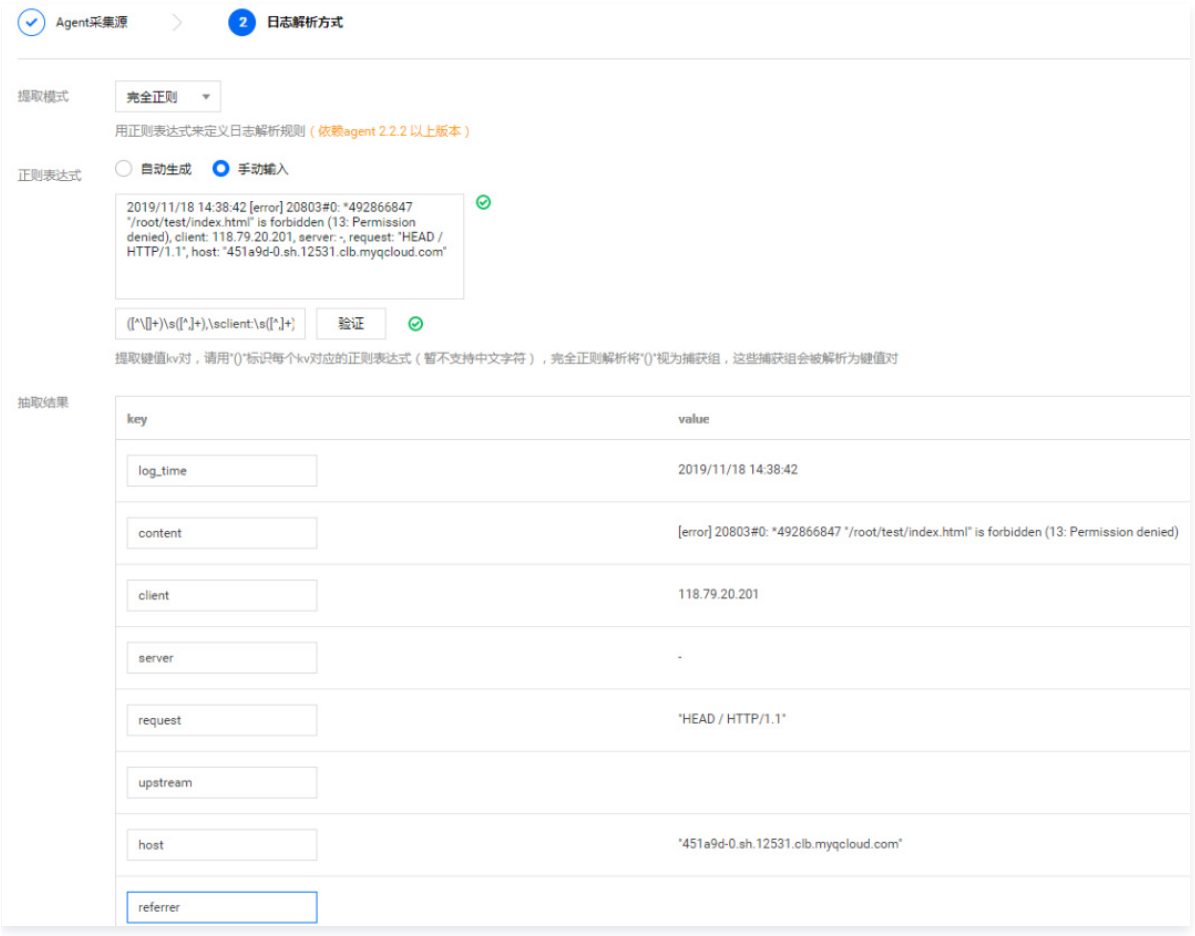
又或者，无 upstream 无 referrer：

```
2019/11/18 14:38:42 [error] 20803#0: *492866847 "/root/test/index.html" is forbidden (13: Permission denied), client: 118.79.20.201, server: -, request: "HEAD / HTTP/1.1", host: "451a9d-0.sh.12531.clb.myqcloud.com"
```

此时，前文提到的正则表达式就无法适用这两种缺失的场景。
通过正则表达式中的组合的 `非捕获括号语法(?:x)` 以及 `匹配0次或1次?` 语法，既能兼容 upstream 和 referrer 缺失的场景，又能保证 `捕获括号` 提取的序号一致。
完整的正则表达式如下：

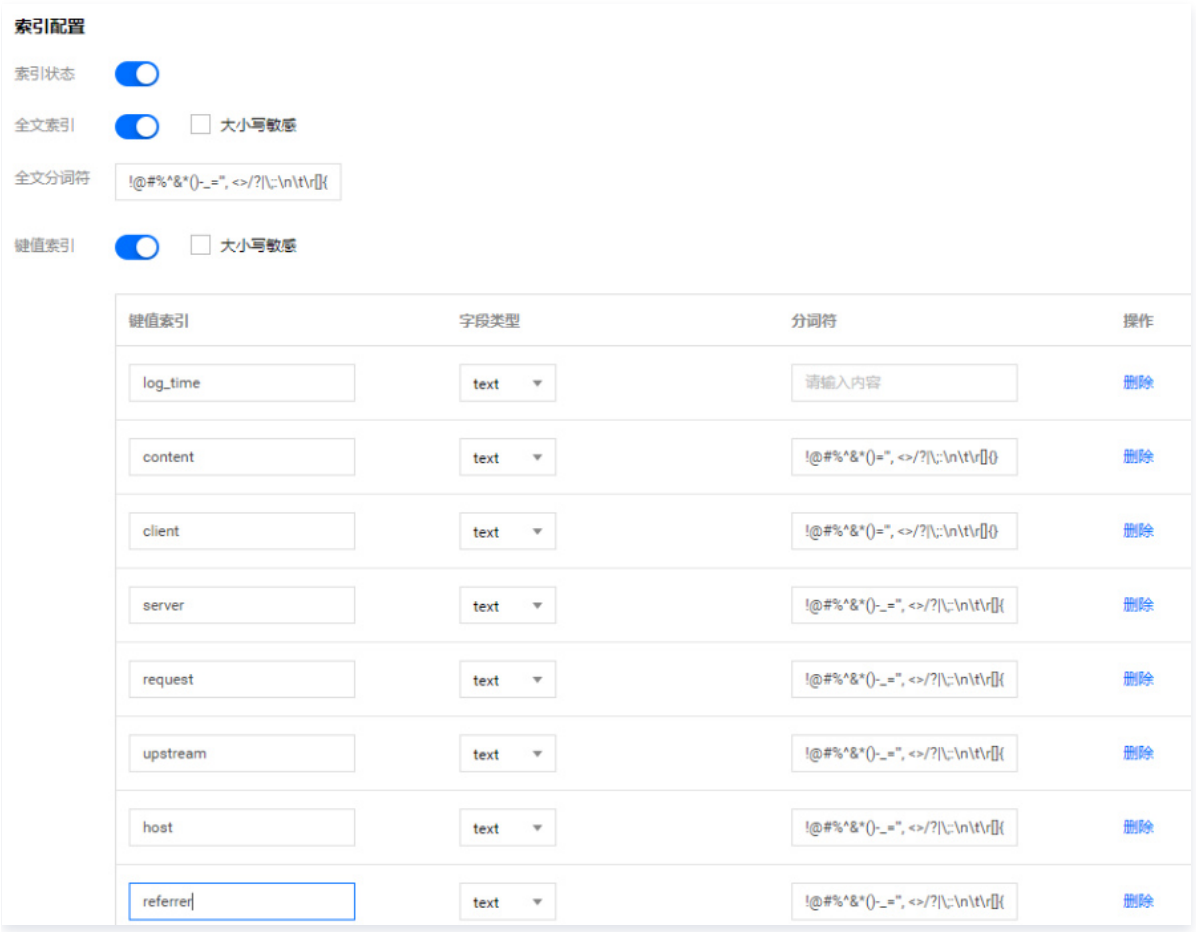
```
([^\[]+)\s([^,]+),\sclient:\s([^,]+),\sserver:\s([^,]+),\srequest:\s([^,]+)(?:,\supstream:\s([^,]+))?,\shost:\s([^,]+)(?:,\sreferrer:\s([^,]+))?
```

针对无 upstream 和 referrer 的日志进行测试，CLS 采集页面上的提取截图如下：



如何进行索引配置?

开启索引，并配置全文和键值索引。



如何检索日志?

在日志服务控制台检索日志，如下所示。
检索关键词：`referrer:admin*`

如何使用采集自检工具

最近更新时间：2024-10-18 10:20:11

如何在非 TKE 环境下使用 Loglistener 日志采集自检工具?

下载工具

下载地址：`https://mirrors.tencent.com/install/cls/loglistener-check/linux/loglistener-check`
请选择对应的版本，下载对应平台的二进制文件。

开始使用

相关命令

```
loglistener check tool

Usage:
  loglistener [flags]

Flags:
  -h, --help              help for loglistener
      --root_dir string   (optional) install loglistener root dir (default is /usr/local/loglistener-*)
      --topic string      (optional) topic ID
```

主要参数说明如下：

- --root_dir：指定 Loglistener 安装根目录，默认是 /usr/local/loglistener-*，例如安装了2.7.2版本，根目录就是/usr/local/loglistener-2.7.2。
- --topic：指定需要检查的日志采集的 topic，如果指定了topic，检查的范围会更加精确。

示例

使用检查工具，查看检查工具提示的问题，并基于提供的解决方案解决问题。
例如，使用默认参数检查结果示例。

```
[root@          centos ~]# ./loglistener-check
                          CLS loglistener check result

1: Check loglistener version                                                    [Warning]
suggestion: Installed version: v2.6.3,latest version: v2.7.2,current install version is not the latest version, recommended to upgrade to the latest version
2: Check machine time zone                                                      [OK]
3: Check machine inotify                                                        [Warning]
suggestion: Check that the parameter fs.inotify.max_user_instances is 128, we recommend that this parameter is greater than or equal to 8192
4: Check loglistener process                                                    [OK]
5: Check Dead lock                                                              [OK]
6: Check Loglistener OOM                                                        [OK]
7: Check log trucate                                                            [OK]
8: Check inotify event lose                                                     [OK]
9: Check loglistener mem limit                                                  [OK]
10: Check loglistener file or dir limit                                          [OK]
11: Check server connection                                                      [OK]
12: Check some key unsupported                                                   [OK]
```

检查结果：

1. 当前机器安装的 Loglistener 版本过低，检查工具发现新版本，推荐用户升级；新版本修复了诸多问题和添加很多新特性，推荐用户升级。
2. inotify 参数设置不符合推荐值，建议用户修改，否则有可能会影响日志采集。

如何在 TKE 环境下使用 Loglistener 日志采集自检工具?

下载工具

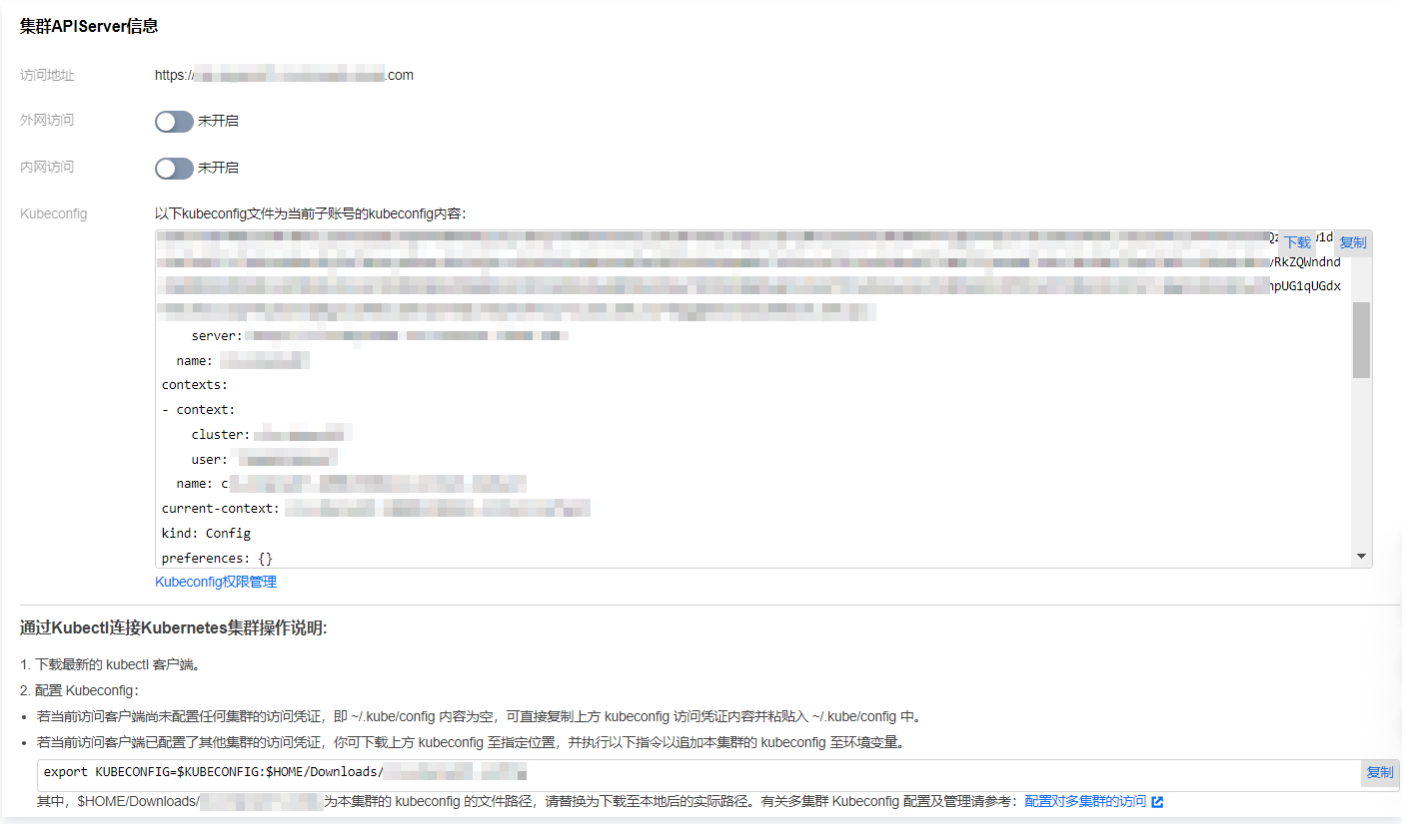
下载地址：`https://mirrors.tencent.com/install/cls/detect-tool`
请选择对应的版本，下载对应平台的二进制文件。

开始使用

连接 TKE 集群

1. 登录 容器服务控制台 。
2. 在左侧导航栏中，单击**集群**，进入集群管理页面。

3. 找到需要连接的集群，单击其 ID/名称，进入该集群管理页面。
4. 在左侧导航栏中，单击**基本信息**，进入基本信息页面。
5. 在“集群APIServer信息”栏中，按照 TKE 的指引，使用 kubectl 连接到 TKE 集群。具体操作可参见 配置 Kubeconfig 。



相关命令

```
Usage of ./detect-tool:
      --kubeconfig string    (optional) Path to a kubeconfig file, specifying how to connect to the API
server. (default "/root/.kube/config")
      --logConfig string     (optional) logconfig name
      --namespace string     (optional) pod name namespace
      --pod string           (optional) pod name
      --topic string         (optional) topic id
```

- 主要参数说明如下：
  - kubeconfig：用于连接 Kubernetes 的配置文件，通常 kubectl 正常连接到集群后，无需特殊设置和指定。
  - logConfig：采集配置名称，指定要检测的采集配置。
  - pod：Pod 名称，指定需要检测的 Pod 的名称。若指定了 Pod，则必须指定 namespace。
  - namespace：指定需要检测的 Pod 所在的 namespace。若指定了 namespace，则必须指定 Pod。
  - topic：指定需要检测 topic 关联的采集配置。一个集群可能会有多个采集配置关联到同一个 topic。

示例

使用检查工具，查看检查工具提示的问题，并基于提供的解决方案解决问题。

- 示例1：指定采集配置和 Pod 检查

```
./detect-tool --logConfig=<采集配置名称> --pod=<pod名称>  --namespace=<pod所在ns>
```

```
% ./detect-tool --logConfig          --pod=cls-provisioner-5849c7d8d7-qp47p --namespace=kube-system
    TKE log collection check result

1: Check log collection componet version                                                    [OK]
2: Check Running status of the cluster log collection component                             [OK]
3: Check Logconfig                                                                          [OK]
4: Check Logconfig matches the Pod                                                          [Error]
suggestion: The specified Pod is not found in the cluster, please check whether the specified Pod exists
5: Check PVC/NFS for Pod                                                                    [OK]
```

检查结果：指定的 pod 和采集配置规则不匹配。

- 示例2：指定 topic 检查

```
./detect-tool --topic=<日志主题ID>
```

```
% ./detect-tool --topic=
    TKE log collection check result

1: Check log collection componet version                                                    [Warning]
suggestion: Installed version: v1.0.8.1, The current install version is not the latest version, it is recommended to upgrade to the latest version in the TKE console
2: Check Running status of the cluster log collection component                             [OK]
3: Check Logconfig                                                                          [Warning]
suggestion: There are multiple Logconfig associated with the specified topic 9                  917, please confirm that multiple Logconfig do not collect the same log file
```

检查结果：

- 建议升级日志组件版本，当前集群不是最新的版本。
- 有两个采集配置关联了一个 topic，建议避免多个采集配置采集同一个文件。

检索分析相关
检索不到日志

最近更新时间：2024-05-29 14:28:51

检索语句为空时，为什么检索不到任何日志?

检索语句为空时，代表无任何过滤条件、查询所有日志。如果未能查看到日志，可按以下步骤排查：

1. **选择较大的时间范围（例如最近30天），保持检索语句为空，确定是否有日志。**
2. **无日志：**
   - 可能为当前日志主题未开启索引，请在 索引配置 中为该主题开启索引。
   - 可能为之前日志上报时未开启索引，日志上报结束后才开启的索引，请对历史数据 重建索引，或在开启索引后重新采集日志。
   - 可能为日志上报失败，使用 LogListener 采集日志时，请检查 机器组状态是否出现异常，使用 SDK 等方式上报日志时请检查 SDK 自身输出日志是否存在错误。
   - 可能为日志时间错误，日志上报至未来时间或已超出日志主题存储时长的时间。使用 LogListener 采集日志时，请检查日志所在服务器时间是否正确，如果使用了自定义日志时间，请检查 时间格式 是否配置正确。使用 SDK 等方式上报日志时，请检查日志时间参数传参是否正确。
3. **部分时间段有日志，部分时间段无日志：**
   - 可能为日志上报失败，使用 LogListener 采集日志时，请检查 机器组状态是否出现异常，使用 SDK 等方式上报日志时请检查 SDK 自身输出日志是否存在错误。
   - 可能为该时间段确实无日志，请检查自身业务在指定时间段是否产生日志。

检索语句为空时能够检索到日志，为什么输入检索语句后检索不到任何日志?

有以下几种可能会导致该问题：

- 部分日志上报失败：使用 LogListener 采集日志时，请检查 机器组状态是否出现异常，使用 SDK 等方式上报日志时请检查 SDK 自身输出日志是否存在错误。
- 索引配置中分词不正确：检索基于日志分词，原始日志需在分词后包含检索条件指定的“词”才能被检索到，例如通过 error 检索不到 errorMessage，因为 errorMessage 本身是一个单独的词，不等于 error，需使用通配符 error*才能检索该日志。关于分词的更多说明及示例参见 分词与索引 。
- 历史日志未采用最新的索引配置：修改索引配置后，仅针对新采集的日志生效。您可以在 索引配置 页面查看上一次索引修改时间，若被检索日志位于该时间前，则有可能按照当前的索引配置无法检索该历史日志，请对历史数据 重建索引 。
- 使用 LogListener 采集日志时，某些日志在采集环节出现解析异常，存储在了 LogParseFailure 字段下，请尝试全文检索或对 LogParseFailure 字段使用键值检索（需为 LogParseFailure 开启键值索引）。
- 某些日志在创建索引的过程中出现 异常，存储在了__RAWLOG__字段下，请尝试全文检索。

使用日志服务控制台能够检索到日志，为什么使用 API 检索不到日志?

该问题是由于 API 传参错误导致，建议在日志服务控制台页面下使用浏览器开发者工具查看接口的请求参数，与您调用 API 的参数是否一致。最常见的问题是 SearchLog 接口的 From 和 To 参数使用的 Unix 时间戳单位错误，应当使用毫秒格式 Unix 时间戳，不支持秒格式。

检索分析报错

最近更新时间：2025-06-18 15:44:32

常见报错信息、原因及解决方案如下：

报错信息	报错原因	解决方案
SyntaxError [field: xxx, can not search on this field, since it is not indexed] QueryError [illegal_argument_exception.Cannot search on field [xxx] since it is not indexed.]	字段 xxx 未开启键值索引	为该字段开启键值索引，详情请参见 键值索引
SyntaxError [Full-Text can not search on this field, since it is not indexed] QueryError [illegal_argument_exception.Cannot search on Full-Text since it is not indexed.]	未开启全文索引	开启全文检索，详情请参见 全文索引
QueryError [line X:X:Column 'XXX' cannot be resolved]	SQL 语句中 xxx 字段未开启统计，不能用于统计分析	为该字段开启统计，详情请参见 键值索引 。另外请注意 SQL 中字符串需使用单引号包裹，双引号包括代表字段
QueryError [parse_exception.XXX]	查询语句语法错误	检查报错信息中指出的错误位置及错误原因，一般为检索条件语法错误（非 SQL）
QueryError [line X:X: XXX]	查询语句语法错误	检查报错信息中指出的错误位置及错误原因，一般为 SQL 语法错误
QueryError [line X:X:Function 'xxx' not registered]	SQL 语句中不支持 xxx 函数	检查函数名称是否拼写错误
QueryError [line X:X:Unexpected parameters (XXX) for function XXX]	SQL 语句中函数入参数据类型不正确	检查函数入参，可使用 类型转换函数 修改数据类型
QueryError [line X:X:'XXX' must be an aggregate expression or appear in GROUP BY clause]	SQL 语句中 GROUP BY 语法错误	一般为 SELECT 字段没有出现在 GROUP BY 子句中，详见 GROUP BY 语法
QueryError [Syntax error in query statement, please check]	查询语句语法错误	查询语句语法错误，但系统暂时不能明确指出具体错误位置，请检查完整的语句，或联系 技术支持
QueryError [illegal_argument_exception.syntax error on field [and|or|not], or full text search is closed]	检索条件不支持小写逻辑操作符，小写逻辑操作符会按照普通字段进行全文检索	使用大写逻辑操作符 AND|OR|NOT，如您并不需要逻辑操作，而是全文检索包含 and|or|not 的日志，请开启全文索引
QueryError [number_format_exception.For input string: ">"]	数值比较语句语法错误	检查数值比较符号周围是否存在空格等特殊符号，正确格式参见 status:>400
SyntaxError [query: *tes, prefix fuzzy query is not supported] QueryError [parse_exception.parse_exception: Cannot parse 'xxx': '*' or '?' not allowed as first character in WildcardQuery]	不允许使用前缀模糊查询，例如 content:*example	前缀模糊检索的使用限制及示例请参见 语法规则
QueryError [circuit_breaking_exception. Analysis data is too large,please reduce the scope of data query]	查询数据量过大	适当缩减查询时间范围，精确检索条件。如果仍旧报错，请联系 技术支持
Internal error. Please try again later RequestId: [7be994d4-xxxx-xxxxx-xxxx-9c38xxxx65de]	CLS 内部错误	请联系 技术支持，并提供报错信息中的 RequestId
The search is timed out. Please try again later. SearchTimeout	查询超时	适当缩小数据查询范围及 SQL 复杂度，或稍后再试

LimitExceeded.LogSearch	搜索并发超过限制	降低查询频率（包括 API），稍后再试。如当前查询频率并不高，仍旧报错，请联系 技术支持

统计分析(SQL)结果小数位数不正确

最近更新时间：2024-07-01 16:46:12

现象1：未保留小数位数，仅有整数部分

使用类似如下的检索分析语句时，期望能够保留小数点两位数，但结果只有整数部分：

- 检索分析语句：

```
* | select count_if(status>=400)*100/count(*) as "错误日志百分比"
```

- 结果：

```
0
```

即便使用 round 函数 指定需要保留的小数点位数，结果仍然只有整数部分：

- 检索分析语句：

```
* | select round(count_if(status>=400)*100/count(*),2) as "错误日志百分比"
```

- 结果：

```
0
```

原因

SQL 中，整数(bigint)之间进行除法运算时，结果仍为整数，对整数使用 round 函数也是无效的。

解决办法

- 方法1：

  在四则运算中，使用乘法将其中一个整数(bigint)转换为小数(double)，例如：

```
* | select round(count_if(status>=400)*100.0/count(*),2) as "错误日志百分比"
```

- 方法2：

  使用 类型转换函数 将其中一个整数(bigint)转换为小数(double)，例如：

```
* | select round(cast(count_if(status>=400)*100 as double)/count(*),2) as "错误日志百分比"
```

现象2：小数位数保留不正确

使用类似如下的检索分析语句时，期望能够最终保留小数点2位数，但结果保留了14位数：

- 检索分析语句：

```
* | select round(count_if(status>=400)*1.0/count(*),5)*100 as "错误日志百分比"
```

- 结果：

```
99.99900000000001
```

原因

`count_if(status>=400)*1.0/count(*)` 的结果类型为 double，使用 round 函数 后结果仍为 double，再进行运算时不能再保证小数位数的正确性。

解决方法

将所有的运算进行完毕后，最终再统一保留小数位数，例如：

```
* | select round(count_if(status>=400)*100.0/count(*),3) as "错误日志百分比"
```

其他问题

最近更新时间：2024-12-26 18:12:22

日志服务 CLS 是什么?

日志服务（Cloud Log Service，CLS）提供一站式的日志数据解决方案。您无需关注扩缩容等资源问题，五分钟快速便捷接入，即可享受日志的采集、存储、加工、检索分析、消费投递、生成仪表盘、告警等全方位稳定可靠服务。全面提升问题定位、指标监控的效率，大大降低日志运维门槛。
日志服务主要提供以下功能：

- 日志采集：便捷实时采集跨地域、多渠道、多平台、不同数据源的日志数据，轻松采集多种其他腾讯云产品日志。
- 日志存储：提供两种存储类型：实时存储和低频存储。
- 日志检索分析：使用关键词检索日志，帮助用户快速定位异常日志，同时支持使用 SQL 对日志进行统计分析，获取日志条数随时间变化趋势、错误日志比例等统计指标。
- 日志数据加工：日志过滤、清洗、脱敏、富化、分发、结构化。
- 日志投递与消费：投递到腾讯云存储、中间件，消费到流计算。
- 仪表盘：将检索分析结果快速生成自定义 Dashboard。
- 告警：异常日志秒级告警，支持通过电话、短信、邮件、微信、企业微信和自定义接口回调等方式通知用户。

日志服务如何定义一条日志?

日志（Log）是应用系统运行过程中产生的记录数据，如用户操作日志、接口访问日志、系统错误日志等。日志通常以文本的形式存储在应用系统所在的机器上，一条系统运行记录对应的日志可能为一行文本（单行日志），也可能为多行文本（多行日志）。
更多说明及示例请参见 日志与日志组 。

日志可以保存多长时间?

日志服务提供日志生命周期管理，在创建日志主题时可以指定日志的有效保存周期，支持保存1 – 3600天或永久保存，逾期后数据将会被清理且不会再产生存储费用。

如何删除日志?

日志服务不支持直接删除日志，只能在日志超出有效保存周期后被自动清理，您可以在日志主题的**基本信息**中查看或修改**日志保留时间**。

> ⚠ **注意：**
> 过期日志的自动清理不是实时的，存在一定延迟，因此缩短保存周期后再立刻恢复之前的保存周期，并不一定会触发自动清理。例如原本的保存周期为7天，10:00时刻修改为了1天，10:01时刻又修改为了7天，日志的清理可能并未触发。该延迟一般为2h，部分情况下会更大。

日志集和日志主题的区别是什么?

日志主题（Topic）是日志数据在日志服务（Cloud Log Service，CLS）平台进行采集、存储、检索和分析的基本单元，采集到的海量日志以日志主题为单元进行管理，包括采集规则配置、保存时间配置、日志检索分析以及日志下载/消费/投递等。
日志集（Logset）是对日志主题的分类，一个日志集可包含多个日志主题。日志集本身不存储任何日志数据，仅方便用户管理日志主题。
更多说明及示例请参见 日志主题与日志集 。

单个日志主题最多可采集多少日志?

为应对海量日志采集需求，单个日志主题包含多个主题分区，每个主题分区写请求最大为500QPS，写流量最大为5MB/s。采集日志量较大时，建议开启 主题分区自动分裂 功能（默认开启），单个日志主题最大可拥有50个分区，此时单个日志主题的写请求最大为50 * 500 = 25000QPS，写流量最大为50 * 5 = 250MB/s。
此处的写请求及写流量并不能简单的等于日志条数及日志量，日志上传时会将多条日志打包为一个 日志组 并进行压缩，实际支持的日志条数及日志量将远大于上述限制。使用 Loglistener 时将自动进行日志打包及压缩，您无需关注具体的打包策略。

什么是索引和分词?

索引配置是使用日志服务（Cloud Log Service，CLS）进行检索分析的必要条件，只有开启索引才能对日志进行检索分析。创建索引实际上是将原始日志按指定的符号切分为多个片段（即分词），并对分词进行 倒排索引 的过程。
更多说明及示例请参见 分词与索引 。

全文索引和键值索引有什么区别?

- 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 `error` 表示检索包含 `error` 关键词的日志。

- 键值索引：键值索引将原始日志按字段（即 key:value）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 `level:error` 表示检索 `level` 字段中包含 `error` 的日志。

更多说明及示例请参见 配置索引 。

检索和分析有什么区别?

- 检索：根据指定的条件查找匹配的原始日志，例如使用 `status:404` 检索响应状态码为404的应用请求日志。
- 分析：针对符合检索条件的日志使用 SQL 进行统计分析，例如使用 `status:404 | select count(*) as logCounts` 统计响应状态码为404的应用请求日志数量。

更多说明及示例请参见 检索分析概述及语法规则 。

日志检索分析性能如何?

- 检索性能：百亿级别日志，秒级返回结果。
- 分析性能：亿级别日志，秒级返回结果；百亿级别日志，1分钟内返回结果。该性能与分析使用的 SQL 复杂度有很大的关系，SQL 非常复杂时可能低于该性能指标。

从日志生成到可以检索到需要多久?

通过 Loglistener/Kafka 协议/API/SDK 等方式采集日志时，从日志生成到可以检索的时间均为秒级。

服务不在腾讯云上，可以使用日志服务吗?

可以使用。日志服务对日志源没有限制要求，只要日志源与日志服务的服务端之间网络可达，就可以将日志采集到日志服务中来。日志服务支持的地域及对应的域名请参见 可用地域 。

服务器更换 IP 地址后，Loglistener 应该如何适配?

- 若服务器通过机器标识绑定机器组，用户无需变更 Loglistener 配置。若服务器 IP 需要频繁变更，建议用户使用机器标识配置机器组。详情请参见 机器组 。
- 若服务器通过 IP 地址绑定机器组，用户需要完成以下配置变更：

1. 修改 Loglistener 安装目录下的 /etc/loglistener.conf 文件。此处安装目录以 /usr/local 为例：

```
vi /usr/local/loglistener-2.3.0/etc/loglistener.conf
```

2. 键盘按 i 键，进入编辑模式。
3. 修改配置文件中 group_ip 部分，填入变更后的 IP 地址。
4. 保存设置并退出编辑器，具体操作步骤：按 **Esc** 键，输入 **:wq**，按 **Enter** 键。
5. 执行如下命令，重启 Loglistener。

```
/etc/init.d/loglistenerd restart
```

6. 登录 日志服务控制台 ，在左侧导航栏中，单击**机器组管理**，修改该服务器绑定的机器组配置，使用新 IP 替换原机器 IP 地址，单击**确定**。

如何排查测试告警通知渠道报错或未收到测试消息?

情况1：页面显示“发送失败”

鼠标在“发送失败”上悬停可查看错误码及详细的失败原因，常见的错误码如下：

错误码	含义	排查方式
-1004	该通知渠道消息发送失败	一般是由于接收对象中的用户或用户组未配置或验证手机、邮箱或微信导致，可在 用户列表 中查看并配置。
-1005	该通知渠道部分消息发送失败，例如部分用户未成功发送，或部分渠道未成功发送	一般是由于接受对象中的部分用户或用户组未配置或验证手机、邮箱或微信导致，可在 用户列表 中查看并配置。
-1006	企业微信或自定义接口回调报错	根据详细的失败原因进一步排查，常见的错误包括： • invalid URI for request：不是一个合法的 URL 地址。 • i/o timeout：接口访问超时，请检查接口地址及能否通过公网直接访问。

		• callback custom error with status:xxx：接口响应报错，请检查接口地址及后端服务是否正常。 • http status code is 400：接口 HTTP 状态码为400 • ssrf attack：回调接口地址需为公网可直接访问的地址，接口地址为腾讯云内网时可能出现该错误。

说明：
测试“自定义接口回调”告警渠道时，平台采用固定的请求参数调用该自定义接口，可能会存在请求参数不符合接口要求而导致通知发送失败的情况，此时测试通知渠道功能无实际意义，您可直接在告警策略中按接口要求配置合适的请求头和请求内容，详情参见 自定义回调接收告警通知 。

情况2：页面显示“已发送”，但实际未收到测试消息

根据接收渠道，常见原因如下：

接收渠道	原因
邮件、短信、微信、电话	为避免重复通知干扰用户，一天内仅允许向同一用户使用同一渠道发送一次测试消息。
自定义接口回调（钉钉、飞书、Slack 等第三方平台地址）	测试消息不符合钉钉、飞书、Slack API 接口要求，消息被忽略，此时测试通知渠道功能无实际意义，您可直接在告警策略中按钉钉及飞书 API 要求配置合适的请求头和请求内容来发送告警，详情可参见 管理通知渠道组 。
自定义接口回调（其它地址）	CLS 以 HTTP 响应状态码来判断消息发送是否成功，请检查自定义接口是否存在 HTTP 响应状态码正常，但仍存在其他业务逻辑限制的情况。