阿里云
日志服务

常见问题

文档版本：20181218

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
	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。	 **禁止：** 重置操作将丢失用户配置数据。
	该类警示信息可能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，恢复业务所需时间约10分钟。
	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。	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。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 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。	`swich` *`{stand | slave}`*

目录

1 基本问题

为您介绍日志服务的基本问题。

日志服务是什么？

日志服务（Log Service，简称LS）是对日志收集、存储、订阅平台化服务。服务提供各种类型日志的实时收集，中心化管理、消费功能。

日志服务可以用来做什么事情？

- 多种方式（API、SDK及Logtail接入服务）的日志写入途径。
- 通过Logtail自由定义日志的收集以及解析方式。
- 利用机器组管理数以千计机器上的日志收集。
- 提供实时日志消费与订阅功能。
- 简单易用的控制台配置方式，所有操作都可以在Web端完成。
- 后台与阿里云多个云产品无缝对接。

日志服务的基本概念有哪些？

- 核心概念为：Project（项目、管理日志基础单元）、Logstore（日志库）、Shard（分区）、Topic（主题、对于Logstore二级分类）、Log（日志条数）、LogGroup（日志组）。
- 日志收集概念：Logtail配置（定义如何收集日志配置）、机器分组（分组）。

日志服务有哪几部分组成？

主要有日志收集客户端、服务端以及其他系统。客户端目前有Windows、Linux版本日志收集Agent（Logtail），服务端处理日志服务API读写、以及配置操作，其它系统包括OSS等阿里云产品，即支持向OSS等云产品同步日志数据。

日志服务如何定义一条日志？

日志包含三部分：时间（必填），日志内容（Key：Value对组成），以及元数据（Source，日志来源IP）。

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。

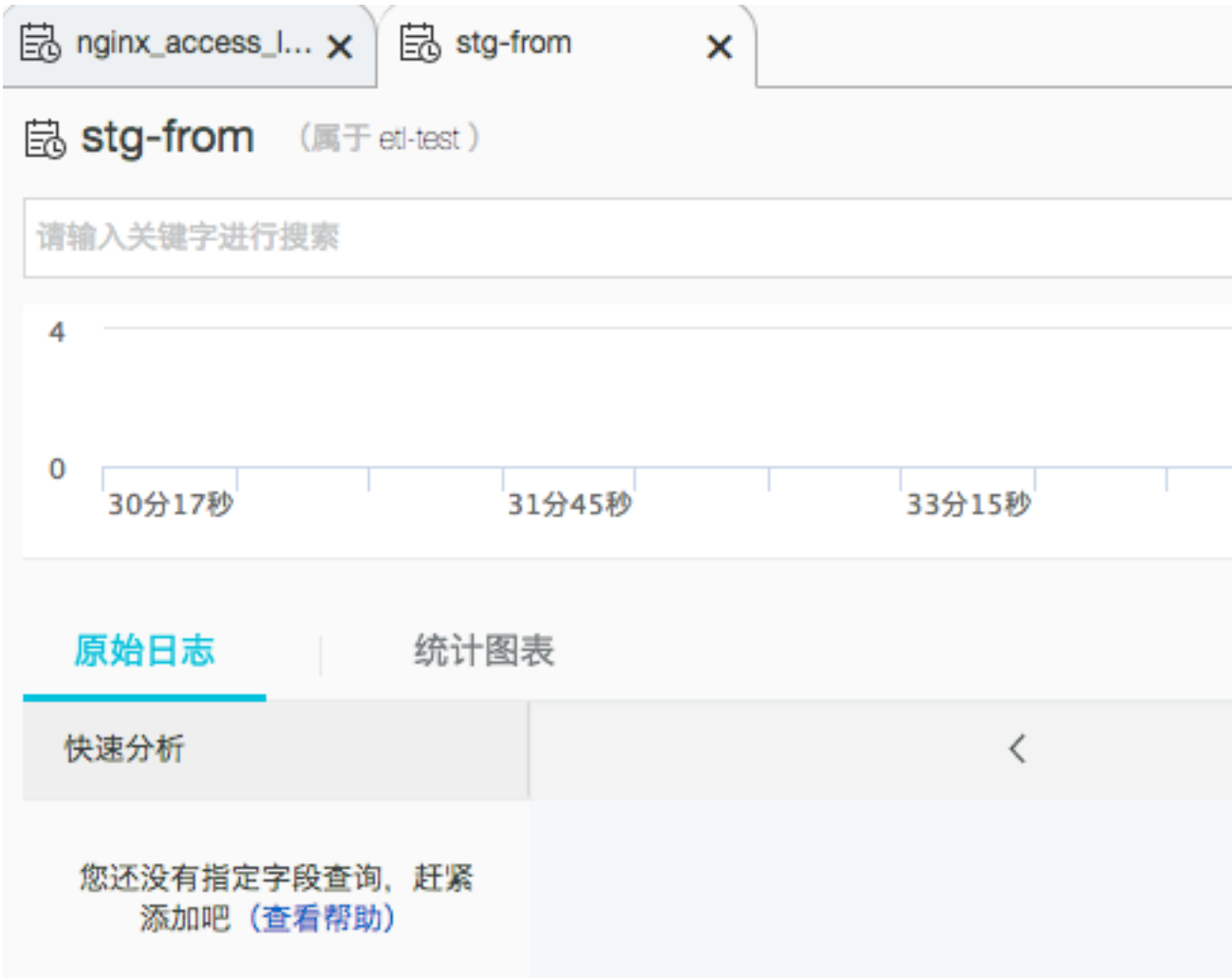
注意事项

- 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。

  参考操作*Logstore*，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。

  1. 开启*OSS*投递功能，将Logstore数据准实时投递到OSS保存。
  2. 在**Logstore**列表页，单击**查询**。

3. 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，了解更多请参考计费方式。

3 日志采集

3.1 Logtail排查简介

配置Logtail采集日志后，如果预览页面为空，或查询页面显示无数据，可以根据本文步骤进行排查。

背景信息

使用Logtail采集日志时，可能预览页面为空

操作步骤

1. 确认机器组内Logtail心跳是否正常。

   在日志服务控制台查看Logtail机器组心跳状态，详细步骤请参考*#unique_8/unique_8_Connect_42_section_ijx_mp1_ry*。

   如果心跳OK则直接进入下一步；如果心跳FAIL则需要进一步排查。

   详细排查步骤请参考*Logtail*机器组心跳失败检查。

2. 确认是否创建了Logtail采集配置。

   确认Logtail客户端状态正常后，则需要确认是否已创建Logtail配置。务必确认日志监控目录和日志文件名与机器上文件相匹配。目录结构支持完整路径和通配符两种模式。

3. 确认Logtail配置是否已应用到机器组。

   查看*Logtail*机器组，选择**管理配置**，确认目标配置是否已经应用到机器组。

4. 检查采集错误。

   确认Logtail配置正常后，您需要确认日志文件是否实时产生新数据。Logtail只采集增量数据，存量文件如果没有更新则不会被读取。如日志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。

   - **诊断采集错误**

     进行*#unique_10*，根据Logtail上报的错误类型处理。

   - **查看Logtail日志**

     客户端日志会记录关键INFO以及所有WARNING、ERROR日志。如果想了解更为实时完整的错误，可以在以下路径查看客户端日志：

     - Linux：*/usr/local/ilogtail/ilogtail.LOG*

- Linux：/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*

- 用量超限

  如果有大日志量或者大文件数据量的采集需求，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。参考*#unique_11*。

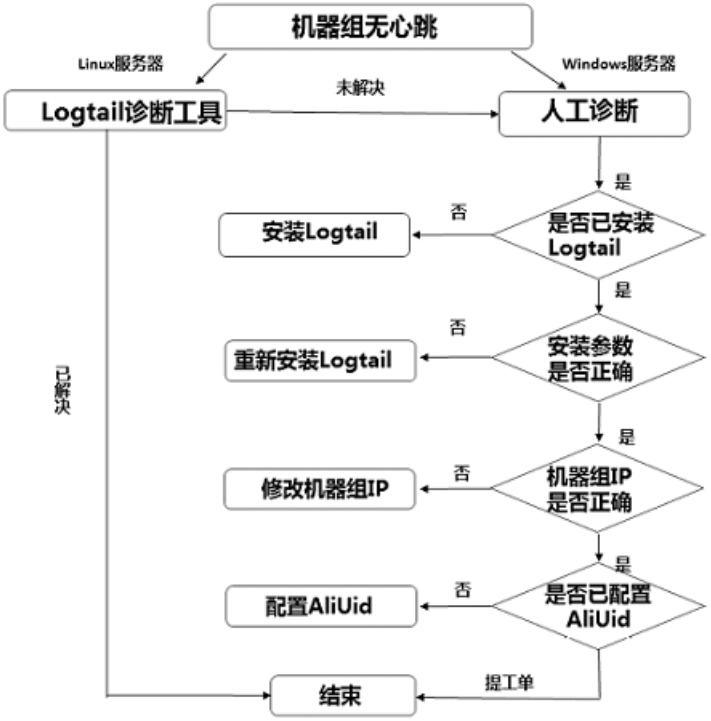
在完成以上三步检查后如问题仍未解决，请工单联系日志服务，并附上排查过程中发现的关键信息。

3.2 Logtail 机器无心跳

配置Logtail采集日志数据后，如果Logtail机器组心跳状态不正常，可使用Logtail自动诊断工具或人工诊断的方式排查问题。

如果使用Logtail采集日志，在服务器上安装Logtail之后，Logtail会定时向服务端发送心跳包。如果机器组状态页面显示机器无心跳，说明客户端和服务端未成功联通。日志服务提供*Logtail*自动诊断工具和人工诊断步骤，您可以根据需求选择排查方式。

- 自动诊断：日志服务提供Linux版Logtail自动诊断工具，排查步骤请参考*Logtail*自动诊断工具。
- 人工诊断：Logtail诊断工具未检查出问题、或服务器为Windows服务器，请参考本文档逐步排查。



1. 检查是否已安装**Logtail**

请执行以下命令查看客户端状态，如未安装Logtail客户端，请参考*Linux*和*Windows*，务必按照您日志服务Project所属Region以及网络类型进行安装。

查看Logtail安装状态：

- Linux：

```
sudo /etc/init.d/ilogtaild status
```

如果显示`ilogtail is running`，表示已安装Logtail，例如：

```
[root@****************~]# sudo /etc/init.d/ilogtaild status
ilogtail is running
```

- Windows：

1. 在控制面板中单击管理工具，并单击服务。
2. 查看LogtailDaemon、LogtailWorker两个Windows Service的运行状态。如果正在运行，表示已安装Logtail。

如果Logtail正在运行，请执行下一步检查。

2. 检查**Logtail**安装参数是否正确

安装Logtail时，需要为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*表1*，并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，可能会导致Logtail机器无心跳。

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，该文件的路径为：

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

1. 检查安装参数。

   检查文件*ilogtail_config.json*中客户端连接的网络入口所属Region是否与您Project所在Region一致。

   例如以下回显信息表明Logtail安装在华东一（杭州）地域的ECS中。

   图 **3-1:** 检查安装参数

```
[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
}
```

2. 检查安装方式。

Telnet测试*ilogtail_config.json*中配置的域名，检查是否根据服务器所属网络环境选择了正确的安装方式。

例如，*ilogtail_config.json*中记录Logtail配置的域名为`cn-hangzhou-intranet`，则可以执行`telnet logtail.cn-hangzhou-intranet.log.aliyuncs.com 80`检查连通性，如果已联通，说明安装方式正确。

例如，查看Linux ECS的网络连接性：

```
[root@********** ~]# telnet logtail.cn-hangzhou-intranet.log.
aliyuncs.com 80
Trying 100*0*7*5...
Connected to logtail.cn-hangzhou-intranet.log.aliyuncs.com.
Escape character is '^]'.
```

如果telenet失败，说明安装时选择了错误的参数，以至于执行了错误的安装命令。请参考*Linux*和*Windows*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。

3. 检查机器组配置的IP地址是否正确

机器组中配置的IP地址必须和Logtail获取到的服务器地址一致，否则机器组无心跳、或无法采集到日志数据。

Logtail在机器上获取IP的方式：

- 如果没有设置主机名绑定，会取服务器的第一块网卡IP。
- 如果在文件*/etc/hosts*中设置了主机名绑定，需要确认绑定的IP。执行命令hostname可以查看主机名。

排查步骤

1. 查看Logtail获取的IP地址。

   文件*app_info.json*的`ip`字段中记录了Logtail获取的IP地址，该文件的路径为：

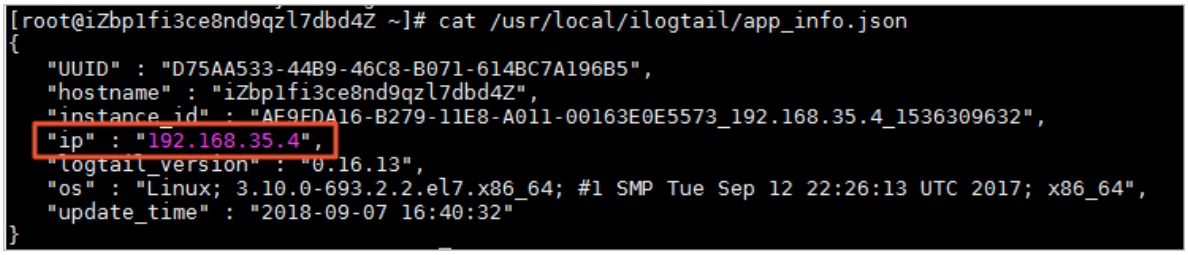
   - Linux：*/usr/local/ilogtail/app_info.json*
   - Windows x64：*C:\Program Files (x86)\Alibaba\Logtail\app_info.json*
   - Windows x32：*C:\Program Files\Alibaba\Logtail\app_info.json*

   说明：

- 如果app_info.json文件中ip字段为空，Logtail无法工作。此时需为服务器设置IP地址并重启Logtail。
- 文件app_info.json仅做记录，修改该文件并不会改变Logtail获取的IP地址。

图 3-2: 查看Logtail获取的IP地址



2. 查看机器组中配置的地址。

机器组中配置的IP地址，请在日志服务控制台机器组列表页面单击查看状态。

图 3-3: 查看机器组

如果服务端机器组内填写的IP与客户端获取的IP不一致，则需要修改。

- 若服务端机器组填写了错误IP，请修改机器组内IP地址并保存，等待1分钟再查看心跳状态。

- 若修改了机器上的网络配置（如修改*/etc/hosts*），请重新启动Logtail以获取新的IP，并根据app_info.json文件中的ip字段修改机器组内设置的IP地址。

重启**Logtail**的方式：

- Linux：

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

- Windows：在控制面板中单击管理工具 > 服务，找到并重启LogtailWorker。

机器组配置的IP地址的确为Logtail获取的IP地址，请执行下一步检查。

4. 检查非本账号下**ECS**是否已配置**AliUid**

如果您的ECS服务器和日志服务Project不在同一账号下、或服务器为其他云厂商服务器、自建IDC，则需要在服务器上配置*AliUid*，为安装Logtail的机器授权。

检查*/etc/ilogtail/users*目录下是否有账号ID同名文件。

如果没有，请参考文档为非本账号*ECS*、自建*IDC*配置*AliUid*进行操作。

说明：

账号ID请在阿里云控制台个人信息中查看。

图 3-4: 查看账号**ID**

如果您的问题仍未解决，请提工单到日志服务。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。

3.3 诊断采集错误

使用Logtail收集日志时，会遇到诸如正则解析失败、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日志收集错误。

操作步骤

1. 登录日志服务控制台，单击Project名称。
2. 在**Logstore**列表的日志收集模式列中单击诊断。

   图 **3-5:** 诊断

3. 查看日志收集错误。

   进入诊断页面后，即可查看指定Logstore的所有Logtail收集日志错误列表，鼠标移至**错误类型**可以查看错误详情。

   具体错误类型请查看日志采集错误类型。

   图 **3-6:** 查看收集错误

4. 查询指定机器收集错误。

   在**日志采集错误**页面中，在输入框输入指定机器IP地址，查看指定机器的所有收集错误。其中Logtail上报错误信息的时间间隔为5分钟。

   处理问题完毕后，根据错误统计时间可以查看业务恢复正常后是否仍有报错。历史报错在过期前仍可显示，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即可。

   > **说明：**
   > 如需查看所有解析失败而丢弃的完整日志行，请登录服务器查看*/usr/local/ilogtail/ilogtail.LOG*。

3.4 日志采集错误类型

在**Logstore**列表页面单击诊断可以查看当前Logstore的所有日志采集报错，本文档介绍具体错误类型的说明及对应的处理方式。

若您遇到其他问题，请提交工单处理。

错误类型	错误说明	处理方式
LOGFILE_PERMINSSION_ALARM	Logtail无权限读取指定文件。	检查服务器Logtail的启动账户，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`.*`。
MULTI_CONFIG_MATCH_ALARM	同一个文件，只能被一个Logtail的配置收集，不支持同时被多个Logtail配置收集。	检查一个文件是否在多个配置中被收集，并删除多余的配置。
REGEX_MATCH_ALARM	正则表达式解析模式下，日志内容和正表达式不匹配。	复制错误内容中的日志样例重新尝试匹配，并生成新的新的解析正则式。
PARSE_LOG_FAIL_ALARM	JSON、分隔符等解析模式下，由于日志格式不符合定义而解析失败。	请单击错误信息，查看匹配失败的详细报错。
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，其它错误请提工单解决。
LOGTAIL_CRASH_ALARM	Logtail因超过服务器资源使用上限而崩溃。	请参考*#unique_11*修改CPU、内存使用上限，如有疑问请提工单。
REGISTER_INOTIFY_FAIL_ALARM	Linux下注册日志监听失败，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。	请参考*#unique_11*修改CPU使用上限或网络发送并发限制，如有疑问请提工单解决。
SEND_DATA_FAIL_ALARM	• 主账号未创建任何AcessKey。 • Logtail客户端机器与日志服务服务端无法连通或者网络链路质量较差。 • 服务端写入配额不足。	• 主账号创建AK。 • 检查本地配置文件*/usr/local/ilogtail/ilogtail_config.json*，执行`curl <服务地址>`，查看是否有内容返回。

错误类型	错误说明	处理方式
		• 为Logstore增加Shard数目，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。	请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。	在控制台扩容分区。
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。	请参考Logtail*#unique_11*修改CPU使用上限或网络发送并发限制，如有疑问请提工单。
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。	请参考Logtail*#unique_11*修改CPU使用上限或网络发送并发限制，如有疑问请提工单。
LOGDIR_PERMINSSION_ALARM	没有日志监控目录读取权限。	请检查日志监控目录是否存在，若存在请检查目录权限设置。
ENCODING_CONVERT_ALARM	编码转换失败。	请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。如存在按照READ_LOG_DELAY_ALARM处理方式解决，若不存在请检查时间字段配置。 • 检查时间字段配置。若时间字段配置正确，请检查日志记录程序时间输出是否正常。 如有疑问请提工单。
STAT_LIMIT_ALARM	日志采集配置目录中的文件数超限。	检查采集配置目录是否有较多的文件和子目录，合理设置监

错误类型	错误说明	处理方式
		控的根目录和目录最大监控深度。
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毕的日志。	该报错通常为采集严重阻塞导致，请参考Logtail#unique_11修改CPU使用上限或网络发送并发限制，如有疑问请提工单。
INPUT_COLLECT_ALARM	输入源采集异常。	请参考错误提示处理。
HTTP_LOAD_ADDRESS_ALARM	http输入的address不合法。	请检查address合法性。
HTTP_COLLECT_ALARM	http采集异常。	请根据错误提示排查，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。	请根据错误提示排查。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在修改表的schema。其他情况请提工单。
MYSQL_INIT_ALARM	MySQL初始化异常。	请参考错误提示处理。
MYSQL_CHECKPOING_ALARM	MySQL checkpoint格式异常。	请确认是否修改该配置中的checkpoint相关配置，其他情况请提工单。
MYSQL_TIMEOUT_ALARM	MySQL查询超时。	请确认MySQL服务器和网络是否异常。
MYSQL_PARSE_ALARM	MySQL查询结果解析失败。	请确认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。	这种情况是由于数据发送过快，若真实数据量很大，则无需关心。

错误类型	错误说明	处理方式
ANCHOR_FIND_ALARM	anchor插件错误、配置错误或存在不符合配置的日志。	请单击错误查看详细报错，报错根据内容分为以下几类，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	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。	请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。	请单击错误查看详细报错，根据错误信息进行进一步地排查，错误一般与所连接的MySQL master相关。
CHECKPOINT_INVALID_ALARM	插件内Checkpoint解析失败。	请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。	请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。

错误类型	错误说明	处理方式
DOCKER_FILE_MATCH_ALARM	无法在Docker容器中查找到指定文件。	请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	docker stdout插件错误，根据配置中的BeginLineRegex构建正则表达式失败。	请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	docker stdout采集初始化失败。	请单击错误查看详细报错，报错根据内容分为以下几类： • host...version...error：请检查配置中指定的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。
DOCKER_STDOUT_START_ALARM	Docker stdout初始化采集时，stdout文件大小超过限制。	一般由于首次采集时stdout文件已存在，可忽略。
DOCKER_STDOUT_STAT_ALARM	Docker stdout无法检查到stdout文件。	一般由于容器退出时无法访问到文件，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	geoip插件错误。	请单击错误查看详细报错，报错根据内容分为以下几类： • 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。

错误类型	错误说明	处理方式
		• 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	http插件错误，配置中指定的ResponseStringMatch正则表达式编译错误。	请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	http插件错误，获取HTTP响应失败。	请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。	请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。	请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。	请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。	请单击错误查看详细报错，报错根据内容分为以下几类： • 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

错误类型	错误说明	处理方式
		• 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。	请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。	请单击错误查看详细报错，报错根据内容分为以下几类： • 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。	请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。	请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。	请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。	请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。
PARSE_DOCKER_LINE_ALARM	service_docker_stdout插件错误，解析日志失败。	请单击错误查看详细报错，报错根据内容分为以下几类：

错误类型	错误说明	处理方式
		• 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。	请单击错误查看详细报错，报错根据内容分为以下几类，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	regex插件错误，编译配置中指定的Regex正则表达式失败。	请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。

错误类型	错误说明	处理方式
		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。	请单击错误查看详细报错，对其中报错的URL进行检查。
REGEX_FIND_ALARM	regex 插件错误，无法从日志中找到配置中SourceKey指定的字段。	请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	regex插件错误，匹配失败。	请单击错误查看详细报错，报错根据内容分为以下几类，请根据具体的错误信息进行进一步地排查，比如检查配置是否正确。 • 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。	请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。	请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。	请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	插件内通过LogFileReader对象进行文件采集时发生错误。	请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。

错误类型	错误说明	处理方式
SERVICE_SYSLOG_INIT_ALARM	service_syslog插件错误，初始化失败。	请单击错误查看详细报错，检查配置中的Address是否正确。
SERVICE_SYSLOG_STREAM_ALARM	service_syslog插件错误，通过TCP采集时发生错误。	请单击错误查看详细报错，报错根据内容分为以下几类，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。	请单击错误查看详细报错，报错根据内容分为以下几类，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。

3.5 排查容器日志采集异常

当您使用Logtail采集容器（标准容器、Kubernetes）的日志时，如果采集状态异常，可以根据本文进行排查问题、检查运行状态等运维操作。

排查异常：

- 排查机器组心跳异常
- 排查容器日志采集异常

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#stdout#
- 查看Kubernetes集群中日志相关组件状态
- 查看Logtail的版本号信息、IP地址和时间
- 误删CRD创建出的Logstore后#应如何处理

排查机器组心跳异常

您可以通过检查机器组心跳状态的方式判断容器上的Logtail是否已正确安装。

1. 查看机器组心跳状态。
   a. 登录日志服务控制台，单击Project名称。
   b. 在左侧导航栏中单击**Logtail机器组**。
   c. 找到要查看状态的机器组，并单击**查看状态**。

      记录心跳状态为**OK**的节点数。
2. 检查集群中Worker节点数。

   执行命令`kubectl get node | grep -v master`，查看集群中Worker节点数。

```
$kubectl get node | grep -v master
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**。
     - 若您使用#unique_30，请参考参数说明检查*${your_region_name}*、*${your_aliyun_user_id}*和*${your_machine_group_user_def*否填写正确。

- 若您使用*#unique_32/unique_32_Connect_42_section_lnl_jpr_zdb*，请提交工单至日志服务。
- 若您使用*#unique_32/unique_32_Connect_42_section_kdx_bqr_zdb*，请参考参数说明，检查*{your-project-suffix}*、*{regionId}*、*{aliuid}*、*{access-key-id}*和*{access-key-secret}*是否已正确填写，若填写错误请执行命令`helm del --purge alibaba-log-controller`删除安装包，并重新安装。

- 机器组心跳数少的节点于集群中Worker节点数。

  1. 判断是否使用yaml文件手动部署DaemonSet。

     执行命令`kubectl get po -n kube-system -l k8s-app=logtail`，若有返回结果，表示您之前使用yaml文件手动部署了DaemonSet。

  2. 下载最新版本的*DaemonSet*模板。
  3. 将其中的*${your_region_name}*、*${your_aliyun_user_id}*、*${your_machine_group_name}*替换为您的真实内容。
  4. 执行命令`kubectl apply -f ./logtail-daemonset.yaml`更新DaemonSet yaml文件。

  其他情况，请提交工单至日志服务。

排查容器日志采集异常

如果您在控制台的**预览**或**查询**页面未查看到日志数据，说明日志服务并未采集到您的容器日志数据。请确认容器状态后，执行以下检查。

1. 确认机器组状态是否正常。
2. 检查采集配置标识是否正确。

   检查配置中**IncludeLabel**、**ExcludeLabel**、**IncludeEnv**、**ExcludeEnv**是否和您需要采集的容器配置匹配。

   > **说明：**
   > 其中的Label为容器Label（**docker inspect**中的Label信息），并不是Kubernetes中定义的Label。 您可以通过将**IncludeLabel**、**ExcludeLabel**、**IncludeEnv**和**ExcludeEnv**配置临时去除，查看是否可以正常采集到日志，若可以采集则可以确定为标记配置问题。

3. 检查其他注意事项。

若您配置采集容器内文件，请注意：

- 若下发采集配置后文件没有修改事件，Logtail则不采集。
- 采集容器日志文件，只支持采集容器默认存储或挂载到本地的文件，暂不支持其他存储方式。

登录Logtail容器

- **普通Docker**

1. 在宿主机执行`docker ps | grep logtail`搜索Logtail容器。
2. 执行`docker exec -it ****** bash`登陆。

```
$docker ps | grep logtail
223fbd3ed2a6e        registry.cn-hangzhou.aliyuncs.com/log-service/
logtail                              "/usr/local/ilogta..."   8 days
ago          Up 8 days                                    logtail-iba
$docker exec -it 223fbd3ed2a6e bash
```

- **Kubernetes**

1. 执行`kubectl get po -n kube-system | grep logtail`，搜索Logtail的Pod。
2. 执行`kubectl exec -it -n kube-system ****** bash`，登陆Pod。

```
$kubectl get po -n kube-system | grep logtail
logtail-ds-g5wgd                                              1/1
   Running    0          8d
logtail-ds-slpn8                                              1/1
   Running    0          8d
$kubectl exec -it -n kube-system logtail-ds-g5wgd bash
```

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。
2. 打开Logtail容器目录*/usr/local/ilogtail/*。

```
cd /usr/local/ilogtail
```

3. 查看文件*ilogtail.LOG*和*logtail_plugin.LOG*。

```
cat ilogtail.LOG
```

```
cat logtail_plugin.LOG
```

Logtail的容器标准输出（stdout）

容器stdout并不具备参考意义，请忽略以下标准输出（stdout）：

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3
c7869500fbe2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser
 to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de
1f85d25009528daa749c1bf8c16edff44beab6e69718/merged: must be superuser
 to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df
72ad0c52fac800cd56c6e880dc4e8a640b1e16c22dbe/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
```

查看Kubernetes集群中日志相关组件状态

执行命令`helm status alibaba-log-controller`可以查看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

相关信息存储在Logtail容器中的*/usr/local/ilogtail/*目录中的*app_info.json*，示例如下：

```
kubectl exec logtail-ds-gb92k -n kube-system cat /usr/local/ilogtail/
app_info.json
{
   "UUID" : "",
   "hostname" : "logtail-gb92k",
   "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_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CRD创建出的Logstore后，应如何处理

若您删除了CRD自动创建出的Logstore，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名。
- 重新启动POD *alibaba-log-controller*。该POD可通过命令`kubectl get po -n kube-system | grep alibaba-log-controller`查找。

3.6 查询本地采集状态

1. *#unique_34/unique_34_Connect_42_section_wyc_hjs_zdb*
2. *#unique_34/unique_34_Connect_42_section_ebd_jjs_zdb*
   a. *#unique_34/unique_34_Connect_42_section_wmp_rjs_zdb*
   b. *#unique_34/unique_34_Connect_42_section_yml_cks_zdb*
   c. *#unique_34/unique_34_Connect_42_section_uxb_kks_zdb*
   d. *#unique_34/unique_34_Connect_42_section_png_vks_zdb*
   e. *#unique_34/unique_34_Connect_42_section_rzj_cls_zdb*
3. *#unique_34/unique_34_Connect_42_section_ff4_hls_zdb*
4. *#unique_34/unique_34_Connect_42_section_ysz_pls_zdb*
   a. *#unique_34/unique_34_Connect_42_section_jcw_rls_zdb*
   b. *#unique_34/unique_34_Connect_42_section_tsq_sls_zdb*
   c. *#unique_34/unique_34_Connect_42_section_uxz_tls_zdb*
   d. *#unique_34/unique_34_Connect_42_section_gg1_vls_zdb*

简介

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。

使用指南

确认已安装支持状态查询功能的Logtail客户端之后，在客户端输入相对命令即可查询本地采集状态。安装Logtail参见*#unique_35*。

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。
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，按需调整#unique_11修改CPU使用上限或网络发送并发限制。
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，按需调整#unique_11修改CPU使用上限或网络发送并发限制。
send_error	日志数据上传失败。	高	参考#unique_10解决。

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：

- 命令logstore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。注意：`--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
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

关键字	含义	单位
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，按需调整*#unique_11*修改CPU使用上限或网络发送并发限制。
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，按需调整*#unique_11*修改CPU使用上限或网络发送并发限制。

状态	含义	处理方式
sender_invalid	日志数据发送异常	检查网络状态，若网络正常，参考#unique_10解决。

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$same
file_path : /disk2/test/normal/access.log
file_dev : 64800
file_inode : 22544456
file_size_bytes : 17154060
file_offset_bytes : 17154060
read_bytes : 65033430
parse_success_lines : 230615
parse_fail_lines : 0
last_read_time : 1503977170
read_count : 667
avg_delay_bytes : 0
/etc/init.d/ilogtaild status logfile --format=json 1 sls-zc-test
release-test-same /disk2/test/normal/access.log
{
    "avg_delay_bytes" : 0,
    "config" : "##1.0##sls-zc-test$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
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

关键字	含义	单位
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    device
inode  file_size  read_offset
 17-08-29 11:26:11             ok   62.12MB         231000
0   17-08-29 11:36:11         671         0B     64800  22544459    18
.22MB      18.22MB
 17-08-29 11:16:11             ok   62.02MB         230615
0   17-08-29 11:26:10         667         0B     64800  22544456    16
.36MB      16.36MB
 17-08-29 11:06:11             ok   62.12MB         231000
0   17-08-29 11:16:11         687         0B     64800  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_queue
network_error  quota_error  discard_error  send_success  send_block
send_valid          max_unsend          min_unsend    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)`，具体参见*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。

返回值	类型	输出	问题排查
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。
   - 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。
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。
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。

3.7 Logtail 快速诊断工具

当日志采集发生异常时，您可以通过Logtail自助检测工具查看客户端是否存在异常情况，根据工具提示快速定位并解决问题。

> **说明：**
> 本工具目前仅支持Linux系统的服务器。

准备工作

1. 下载检测工具脚本。

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

> **说明：**
> 如果无法正常下载，请通过以下备用地址重试。

```
wget http://logtail-corp.oss-cn-hangzhou-zmf.aliyuncs.com/linux64/
checkingtool.sh
```

```
                    -O checkingtool.sh
```

2. 安装curl工具。

   检查工具需要使用curl进行网络连通性检查，请确保机器已安装curl工具。

运行诊断工具

1. 执行以下命令运行诊断工具：

```
chmod 744 ./checkingtool.sh
./checkingtool.sh
sh checkingtool.sh
```

   回显信息：

```
[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 :
```

2. 请根据提示输入1或2，脚本会根据您的选择执行不同检查流程。

   其中：

   - 1表示执行机器组心跳检查，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。

   选择检查项目后，诊断工具会自动执行对应检查流程。

诊断流程

机器组心跳检查

选择机器组心跳检查流程后会进行下述一系列的检查：

1. 基础环境检查。

   - 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
```

   若其中检查出现`Error`信息，请参考提示进行处理。

2. 确认是否非本人ECS。

   基础环境检查通过后，请确认您的服务器是否为ECS、是否由本账号购买。

若此服务器不是ECS或者ECS购买账号和日志服务账号不同，输入y，否则输入N。

```
[Input]: Is your server non-Alibaba Cloud ECS or not belong to the
same account with the current Project of Log Service ? (y/N)
```

当输入y后，检查工具会输出本地配置的AliUid信息，请确认其中是否包含了您的AliUid，若未包含请参考文档创建*AliUid*标识。

```
[Input]:  Is your server non-Alibaba Cloud ECS or not belong to the
same account with the current Project of Log Service ? (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Region。

   请确认您的Project所在区域是否和Logtail安装时所选区域一致，若不一致请重新安装*Logtail*。

   ```
   [Input]: please make sure your project is in this region : { cn-
   hangzhou } (y/N) :
   ```

4. 检查IP配置。

   请确认您机器组配置的IP和Logtail工作IP一致，若不一致请参考*IP*地址机器组修改。

   若您配置的是自定义标识机器组，请确认本地配置的标识与服务端配置一致，若不一致请参考自定义标识机器组修改。

   ```
   [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) :
   ```

日志采集检查

选择日志未采集检查流程后会进行下述一系列的检查：

1. 确认IP配置。

   请确认您机器组配置的ip和Logtail工作ip一致且心跳正常，若不一致请修改机器组。

   ```
   [Input]:  please make sure your machine group's ip is same with : {
   11.XX.XX.187 } (y/N) :
   ```

2. 确认采集配置应用。

请确认您的采集配置已经成功应用到该机器组中，如何查看机器组应用配置参见管理机器组。

```
[Input]: please make sure you have applied collection config to the
machine group (y/N) :Y
```

3. 检查日志文件。

   检查时请输入您需要检查的日志文件全路径，若未找到匹配项，请确认配置的路径信息可以匹配给定的日志文件。

   若配置错误请重新修改采集配置并保存，1分钟后再次执行此脚本重新检查。

```
[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

若所有的检查全部通过，但采集依然出现异常，请在脚本最后的选择中输入y并回车确认。

请您将检查脚本输出的信息作为附件，提交工单给我们的售后工程师。

```
[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
```

快速检查

快速检查运行时无需确认，可用于二次封装自定义检查脚本。

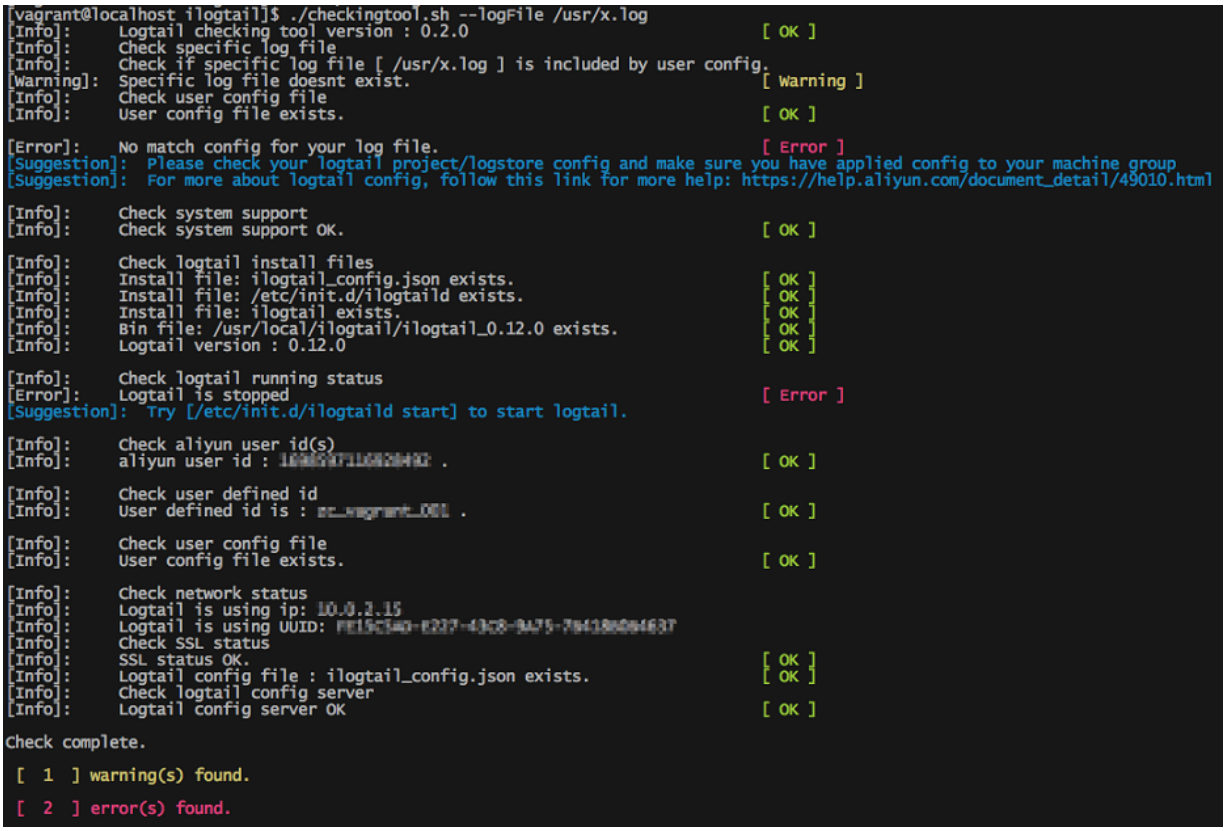
> 说明：
> 快速检查运行时会输出客户端配置的阿里云ID和动态机器组/自定义标识，不存在时并不会给出告警，如果客户端需要阿里云ID或动态机器组/自定义标识的配置，请查看工具的输出和您配置的是否一致，不一致时按照以下方法重新配置：创建*AliUid*标识、自定义标识机器组。

操作步骤

请运行脚本`./checkingtool.sh --logFile [LogFileFullPath]`进行检查。 检测脚本发现异常时，请根据脚本提示进行处理。

说明：

若指定日志文件检查通过且Logtail运行环境正常，建议进入阿里云控制台中查看该日志服务配置项的异常日志，参见诊断采集错误。



Logtail采集异常的常见问题

运行Logtail快速诊断工具后，可以诊断出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	使用命令`/etc/init.d/ilogtaild start`开启Logtail。
多个Logtail进程	使用命令`/etc/init.d/ilogtaild stop`关闭Logtail，然后执行命令`/etc/init.d/ilogtaild start`开启。
443端口被禁用	防火墙开放443端口。

常见问题	解决方法
无法找到配置服务器	确认是否已正确安装 *Linux Logtail*，若安装错误，重新执行安装命令。
不存在用户配置	确认是否已执行以下操作： 1. 控制台已经创建好Logtail配置。 2. 机器组中包含该服务器。 3. 已经将配置应用到机器组。
没有匹配指定日志文件	确认是否正确配置了Logtail。
指定日志文件匹配多次	匹配多次时Logtail会随机选择一个配置，建议去重。

检测工具常用参数

常用参数	说明
`--help`	查看帮助文档。
`--logFile [LogFileFullPath]`	检测Logtail是否收集路径为`LogFileFullPath`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userID、网络连通性等）。
`--logFileOnly [LogFileFullPath]`	只检测Logtail是否收集路径为`LogFileFullPath`的日志。
`--envOnly`	只检测Logtail运行环境。

3.8 日志采集功能与 Kafka对比

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，各用户可以根据需要搭建Kafka集群。

日志服务(Log Service)是基于飞天Pangu构建的针对日志平台化服务。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。通过标准话的Restful API对外提供服务。

Log Service Loghub提供公共的日志采集、分发通道，用户如果不想自己搭建、运维kafka集群，可以使用Log Service LogHub功能。

概念	Kafka	Loghub
存储对象	topic	logstore
水平分区	partition	shard

概念	Kafka	Loghub
数据消费位置	offset	cursor

Loghub & Kafka 功能比较

功能	Kafka	LogHub
使用依赖	自建或共享Kafka集群	Log Service服务
通信协议	TCP 打通网络	Http (restful API)，80端口
访问控制	无	基于云账号的签名认证+访问控制
动态扩容	暂无	支持动态shard个数弹性伸缩(Merge/Split)，对用户无影响
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	暂不开放，默认3份拷贝
failover/replication	调用工具完成	自动，用户无感知
扩容/升级	调用工具完成，影响服务	用户无感知
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、无需关心
保存时间	配置确定	根据需求动态调整

成本对比

参见*#unique_43*中LogHub部分。

3.9 完整正则模式采集日志

3.9.1 如何调试正则表达式

在配置Logtail采集文本日志时，如果选择完整正则模式解析、采集日志，需要您根据自己的日志样例配置正则表达式。本文档介绍如何调试正则表达式。

如果您希望对您所在日志服务控制台所设置的正则表达式进行调试，您可以直接在界面上使用验证按钮所提供的功能来进行检查：

- 对于行首正则表达式，检查一下当前设置能否正确匹配出您期望的日志数量。
- 对于提取字段，检查一下各个字段中的值是否是您所希望的。

如果您希望进行更多的验证乃至调试正则表达式，您可以利用诸如 *Regex101*、*RegexTester* 之类的在线工具，将控制台为您自动生成的正则表达式拷贝粘贴到这些工具上，然后填充您的实际日志来进行检查、调试。

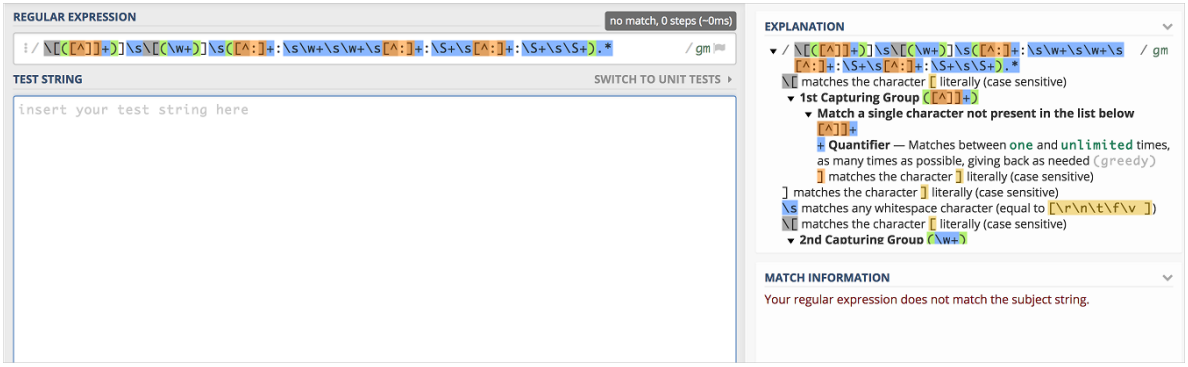
完整正则模式提供自动生成功能，可能会为多行日志的message字段生成不合适的正则。本文档以Regex101为例，对该正则进行以下检查。

操作步骤

1. 拷贝日志服务根据日志样例自动生成的完整正则。
2. 打开网站*Regex101*。
3. 在**REGULAR EXPRESSION**中粘贴自动生成的完整正则：

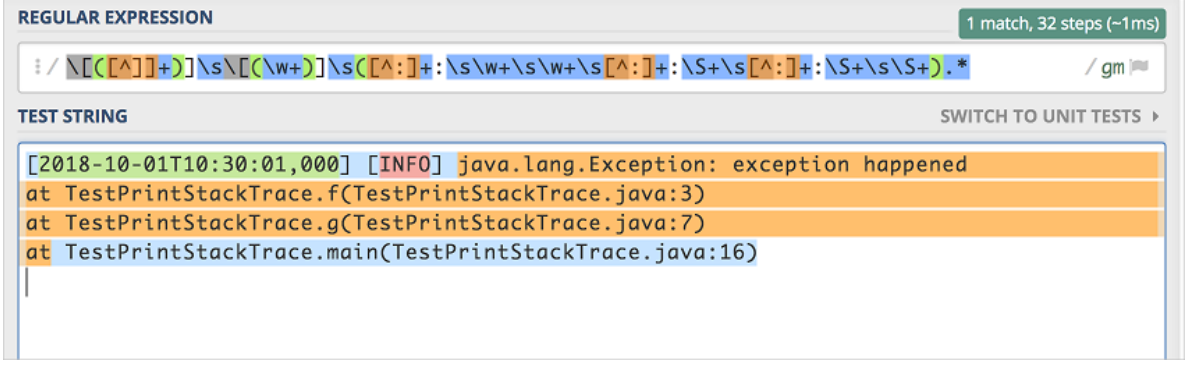
```
\[([^]]+)]\s\[(\w+)]\s([^:]+:\s\w+\s\w+\s[^:]+:\S+\s[^:]+:\S+\s\S
+).*
```

在界面的右侧，您还可以看到该正则的含义。



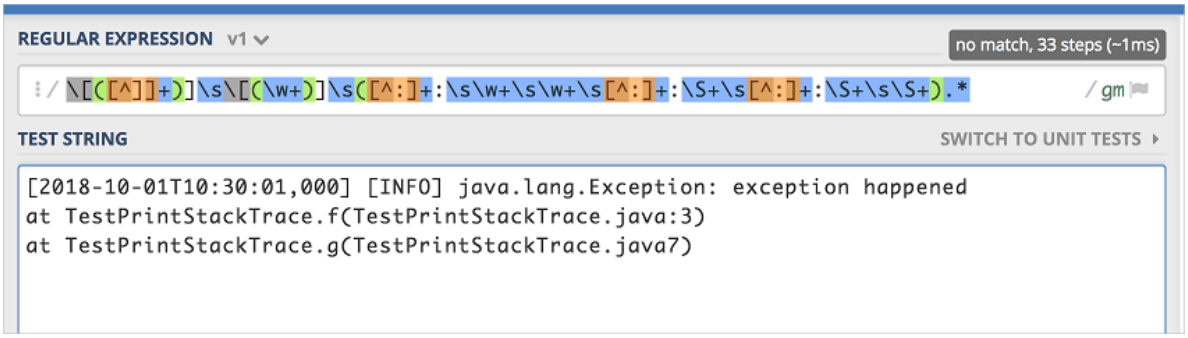
4. 在**TEST STRING**中贴入日志样例中的日志。

以下示例表示正则式与日志部分匹配，`at` 之后的内容并没有被包含到 `message` 字段中（表示为橘色和蓝色），因此该表达式不完全匹配样例日志，即对于该样例日志来说，这条正则表达式是错误的，使用这条正则表达式无法正常采集到所有日志数据。



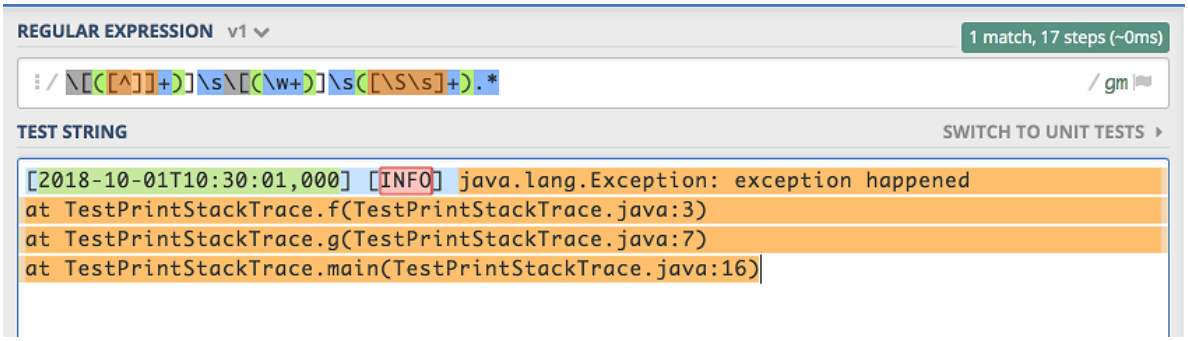
5. 验证另一个错误：如果日志中只有两个冒号的情况。

   以下示例表示匹配失败。

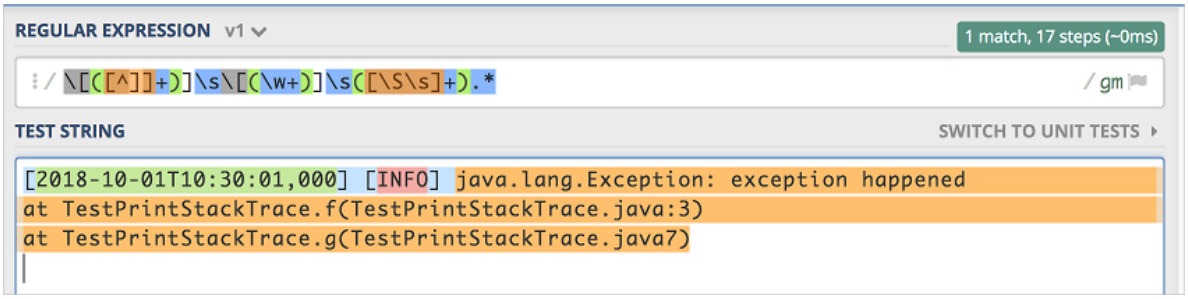


6. 将最后一个正则表达式替换为 `[\S\s]+` ，并再次尝试检查匹配程度。

   `at` 之后的内容如下：



   只有两个冒号的日志：



您也可以按照以上方法来对您的正则表达式进行调试、修改，最终应用于Logtail采集配置中。

3.9.2 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。

3.9.3 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
```

对此，有两种方案可以考虑：

- Schema-On-Write：为同样的一份日志应用多个采集配置，每个采集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：

  Logtail 不支持对同一个文件同时应用多个采集配置，您需要为该文件所在的目录建立多个软链接，每个配置作用于不同的软链接目录来间接实现多个配置同时采集一个文件。

- Schema-On-Read：使用它们共同的正则表达式来采集。

比如说采用多行日志采集，将时间和日志等级作为行首正则，剩余的部分都作为 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（比如千万级）。

3.9.4 如何配置时间格式

在Logtail采集配置中设置时间格式，请遵循以下注意事项。

- 日志服务的时间戳只支持到秒，所以时间格式只需配置到秒，无需配置毫秒、微秒等信息。
- 只需time字段中前面能解析出时间的部分即可，后面无需配置。

常见日志格式配置示例如下：

```
自定义1 2017-12-11 15:05:07
%Y-%m-%d %H:%M:%S
自定义2 [2017-12-11 15:05:07.012]
[%Y-%m-%d %H:%M:%S
RFC822     02 Jan 06 15:04 MST
%d %b %y %H:%M
RFC822Z    02 Jan 06 15:04 -0700
%d %b %y %H:%M
RFC850      Monday, 02-Jan-06 15:04:05 MST
%A, %d-%b-%y %H:%M:%S
RFC1123     Mon, 02 Jan 2006 15:04:05 MST
%A, %d-%b-%y %H:%M:%S
RFC3339     2006-01-02T15:04:05Z07:00
%Y-%m-%dT%H:%M:%S
RFC3339Nano 2006-01-02T15:04:05.999999999Z07:00
%Y-%m-%dT%H:%M:%S
```

4 日志查询

4.1 日志查询常见问题

如何在日志数据中搜索IP地址？

在日志数据中搜索IP地址，支持全部匹配的方式检索。您可以直接在日志数据中直接搜索指定IP地址相关的日志信息，比如包含指定IP地址、过滤指定IP地址等。但是目前尚不支持部分匹配的方式检索，即不能直接搜索IP地址的一部分，因为小数点不是日志服务默认的分词项。如果需要的话，建议自行过滤，比如用SDK先下载数据，然后在代码里用正则或者用string.indexof等方法判断。

例如，在日志服务的Project中搜索条件如下。

```
not ip:121.42.0 not status:200 not 360jk not DNSPod-Monitor not status
:302 not jiankongbao
        not 301 and status:403
```

搜索结果中仍会出现121.42.0网段地址。因为日志服务会认为121.42.0.x是一个词，所以只有搜121.42.0.x能搜到结果，而121.42.0的话不会搜到这个结果，同理加上not也就不会过滤该地址。

如何在日志中搜索包含空格的关键字？

搜索包含空格的关键字时，如果直接搜索，则会得到包含空格左侧关键字或右侧关键字的所有日志。建议您在查询的包含空格的关键字时，把关键字用双引号包裹起来，将引号中的内容作为一个关键字进行搜索，搜索结果就是符合条件的日志内容。

例如，在以下日志中搜索包含关键字`POS version`的日志。

```
post():351]:&amp;nbsp;device_id:&amp;nbsp;BTAddr&amp;nbsp;:&amp;nbsp;B6:xF:xx:65:xx:A1
&amp;nbsp;IMEI&amp;nbsp;:&amp;nbsp;35847xx22xx81x9&amp;nbsp;WifiAddr&amp;nbsp;:&amp;nbsp;4c:xx
:0e:xx:4e:xx&amp;nbsp;|&amp;nbsp;user_id:&amp;nbsp;bb07263xxd2axx43xx9exxea26e39e
5f&amp;nbsp;POS&amp;nbsp;version:903
```

如果直接搜索`POS version`，则会得到包含POS或者`version`的所有日志，不符合搜索要求。如果搜索`"POS version"`，则会得到包含关键字`POS version`的所有日志。

如何完成双重条件检索？

双重条件检索时，只需同时输入两个语句即可。

例如，需要在Logstore中搜索数据状态不是OK或者Unknown的日志。直接搜索`not OK not Unknown`即可得到符合条件的日志。

日志服务提供哪些渠道查询采集的日志？

日志服务提供了三种方式查询日志：

1. 通过日志服务控制台查询。
2. 通过SDK查询。
3. 通过Restful API查询。

日志服务提供什么样的查询能力？

- 提供组合条件过滤查询，查询语法参见查询语法。
- 能够提供单次查询10亿/S日志的能力。用户可以根据一定的条件筛选出需要的日志，读取命中日志在时间维度上的分布，或者拿到原始日志。
- 查询提供了cache的功能，第二次查询相同的条件获得更加完整的查询结果。

日志查询有什么限制？

- 最多能够查询30个词组成的组合条件。
- 单次查询结果最多获取100行原始数据，可以通过翻页下载更多日志。
- 单次查询1秒内可以处理10亿行数据。

4.2 查询不到日志数据

在使用日志服务产品的日志查询功能时，如果查询不到日志数据，请按照以下原因进行排查。

1. 未成功采集日志数据

如果并未成功采集日志数据到日志服务，则无法查询到目标日志。请在预览界面查看是否有日志数据。

如果有日志数据，说明日志数据已成功采集到日志服务中，建议您排查其他原因。

如果没有日志数据，可能是以下原因造成，请进一步排查。

- 日志源没有生产日志数据。

  日志源没有日志产生的情况下，没有日志可以投递到日志服务。请检查您的日志源。

- Logtail无心跳。

  请在机器组状态页面中查看机器是否有心跳。没有心跳请参考*Logtail* 机器无心跳。

- 监控文件没有实时写入。

如果监控文件有实时写入，您可以打开*/usr/local/ilogtail/ilogtail.LOG*查看报错信息。常见错误如下：

- parse delimiter log fail：分割符收集日志出错。
- parse regex log fail：正则收集日志出错。

2. 分词设置错误

查看已设置的分词符，检验根据分词符对日志内容进行分割后，是否刚好得到关键字。例如分割符为默认的`,;=()[]{}?@&<>/:’`那么用户的日志里如果是有abc”defg,hij会被分割成abc”defg和hij两部分，用abc就搜不到这条日志。

同时支持模糊查询，具体查询语法，请参考查询语法。

> 说明：
>
> - 为了节约您的索引费用，日志服务进行了索引优化，配置了键值索引的Key，不进全文索引。例如，日志中有名为message的key，并且配置了键值索引，加了空格做分词（加空格做分词，请把空格加到分词字符串的中间）。”message: this is a test message“可以用 key:value 的格式 message:this 查到，但是直接查this查询不到，因为配置了键值索引的key，不进全文索引。
> - 创建索引或者对索引做任何更改，只对新进的数据有效，旧数据一律无效。
>
> 您可以点开索引属性，检查已设置的分词是否符合要求。

3. 其他原因

如果日志有产生，可以先在查询处修改查询的时间范围。另外由于日志预览的功能数据是实时的，但是查询的功能是有最多1分钟的延迟的，所以用户可以在日志产生后等1分钟再查。

如您的问题仍未解决，请提工单联系我们。

4.3 日志消费与查询区别

日志服务提供日志消费和查询的功能，都属于对日志的读操作，区别在于消费提供收集和分发通道，查询提供日志查询功能。

日志消费与日志查询区别

日志服务提供了两项功能都和“读”有关：

日志收集与消费（LogHub)：提供公共的日志收集、分发通道。全量数据顺序（FIFO）读写，提供类似Kafka的功能

- 每个LogStore有一个或多个Shard，数据写入时，随机落到某一个shard中
- 可以从指定shard中，按照日志写入shard的顺序批量读取日志
- 可以根据server端接收日志的时间，设置批量拉取shard日志的起始位置（cursor）

日志查询（**Search/Analytics**）：在LogHub基础上提供海量日志查询+分析功能，根据条件进行日志查询与统计

- 通过查询条件查找符合要求的数据
- 支持关键词 AND、NOT、OR的布尔组合和结果SQL统计
- 数据查询不区分shard

两者区别：

功能	日志查询(LogSearch)	日志收集与消费(LogHub)
关键词查找	支持	不支持
小量数据读取	快	快
全量数据读取	慢(100条日志100ms，不建议通过该方式读取数据)	快（1MB日志10ms，推荐方式）
读取是否区分topic	区分	不区分，只以shard作为标识
读取是否区分shard	不区分，查询所有shard	区分，单次读取需要指定shard
费用	较高	低
适用场景	监控、问题调查与分析等场景	流式计算、批量处理等全量处理场景

4.4 日志查询分析常见报错

日志查询分析的常见报错如下。

1. line 1:44: Column ‘my_key_field’ cannot be resolved;please add the column in the index attribute

报错原因：`my_key_field`这个Key不存在，所以您在query中无法引用该Key。

解决方案：在查询页面，右上角查询分析属性里，添加该字段为键值索引，同时打开统计功能。

2. Column ‘xxxxline‘ not in GROUP BY clause;please add the column in the index attribute

报错原因：您在查询中使用了GROUP BY语法，但是在Select中引用了一个非agg字段，该字段没有出现在GROUP BY中。例如`select key1, avg(latency) group by key2` ，key1没有出现在GROUP BY中。

解决方案：正确语法是`select key1,avg(latency) group by key1,key2`。

3. sql query must follow search query,please read syntex doc

报错原因：没有指定filter条件，例如`select ip,count(*) group by ip`。

解决方案：正确的写法为`*|select ip,count(*) group by ip`。

4. please read syntex document,and make sure all related fields are indexed. error after select . error detail:line 1:10: identifiers must not start with a digit; surround the identifier with double quotes

报错原因：SQL中引用到的列名、变量名等以数字开头，不符合规范。

解决方案：建议更改该名称，以字母开头。

5. please read syntex document,and make sure all related fields are indexed. error after select . error detail:line 1:9: extraneous input ‘’ expecting

报错原因：有单词拼写错误。

解决方案：请根据报错中指出的错误位置，修改至正确。

6. key (category) is not config as key value config,if symbol : is in your log,please wrap : with quotation mark "

报错原因：category字段未配置字段索引，不能在分析语句中使用。

解决方案：请在查询分析属性中设置该字段的索引。详细说明请参考开启并配置索引。

7. Query exceeded max memory size of 3GB

报错原因：当前query使用服务端内存超过3 GB。通常原因为GROUP BY查询的去重后value太多。

解决方案：请优化GROUP BY的查询语句，减少GROUP BY的Key的个数。

5 日志投递

5.1 日志投递MaxCompute后，如何检查数据完整性

在日志服务数据投递MaxCompute场景下，需要在MaxCompute表分区维度上检查数据完整性，即MaxCompute表中某个分区中数据是否已经完整。

使用保留字段__partition_time__作为表分区列，如何判断分区数据是否已完整

`__partition_time__`由日志的time字段计算得到，由日志真实时间按照时间格式字符串向下取整得出。其中，日志真实时间既不是投递数据的时间，也不是日志写入服务端时间。

举例：日志时间为2017-05-19 10:43:00，分区字段格式字符串配置为`yyyy_MM_dd_HH_mm`，每1小时投递一次。那么：无论该日志是什么时刻写入服务端，这条日志会存入MaxCompute的`2017_05_19_10_00`分区，计算细节参考*MaxCompute*投递字段说明。

如果不考虑写入了历史数据等问题，在日志实时写入的情况下，有以下两种方法判断分区数据是否已完整：

- 通过控制台或API/SDK判断（推荐）

  使用*API*、*SDK*或者控制台获取指定Project/Logstore投递任务列表。例如API返回任务列表如下。控制台会对该返回结果进行可视化展示。

```
{
  "count" : 10,
  "total" : 20,
  "statistics" : {
      "running" : 0,
      "success" : 20,
      "fail" : 0
  }
  "tasks" : [
      ...
      {
          "id" : "abcdefghijk",
          "taskStatus" : "success",
          "taskMessage" : "",
          "taskCreateTime" : 1448925013,
          "taskLastDataReceiveTime" : 1448915013,
          "taskFinishTime" : 1448926013
      },
      {
          "id" : "xfegeagege",
          "taskStatus" : "success",
          "taskMessage" : "",
          "taskCreateTime" : 1448926813,
          "taskLastDataReceiveTime" : 1448930000,
          "taskFinishTime" : 1448936910
```

```
        }
    ]
}
```

`taskLastDataReceiveTime`表示该批任务中最后一条日志到达服务端时的机器系统时间，对应控制台的接收日志数据时间，根据该参数判断时间为T以前的数据是否已经全部投递到MaxCompute表。

- 如果`taskLastDataReceiveTime < T + 300s`（300秒是为了容忍数据发送服务端发生错误重试）以前的每个投递任务状态都是success，说明T时刻的数据都已经入库。
- 如果任务列表中有ready/running状态任务，说明数据还不完整，需要等待任务执行结束。
- 如果任务列表中有failed状态任务，请查看原因并在解决后重试任务。您可以尝试修改投递配置，以解决投递问题。

- 通过MaxCompute分区粗略估计

  比如在MaxCompute中以半小时做一次分区，投递任务为每30分钟一次，当表中包含以下分区：

  ```
  2017_05_19_10_00
  2017_05_19_10_30
  ```

  当发现分区2017_05_19_11_00出现时，说明11:00之前分区数据已经完整。

  该方法不依赖API，判断方式简单但结果并不精确，仅用作粗略估计。

使用自定义日志字段作为表分区列，如何判断分区数据已完整

比如日志中有一个字段date，取值：20170518，20170519，在配置投递规则时将date列映射到表分区列。

这种情况下，需要考虑date字段与写入服务端时间差，结合使用保留字段方法，根据接收日志数据时间判断。

投递成功，但MaxCompute表数据有缺失，应如何解决

MaxCompute投递任务状态成功，但表中数据缺失，一般有以下原因：

- 表分区列映射的日志服务字段的名称不存在。此时投递过去的列值为null，而MaxCompute表不允许分区列值为null。
- 表分区列映射的日志服务字段的值包含/或其他特殊符号。MaxCompute将这些字符作为保留字，不允许在分区列中出现。

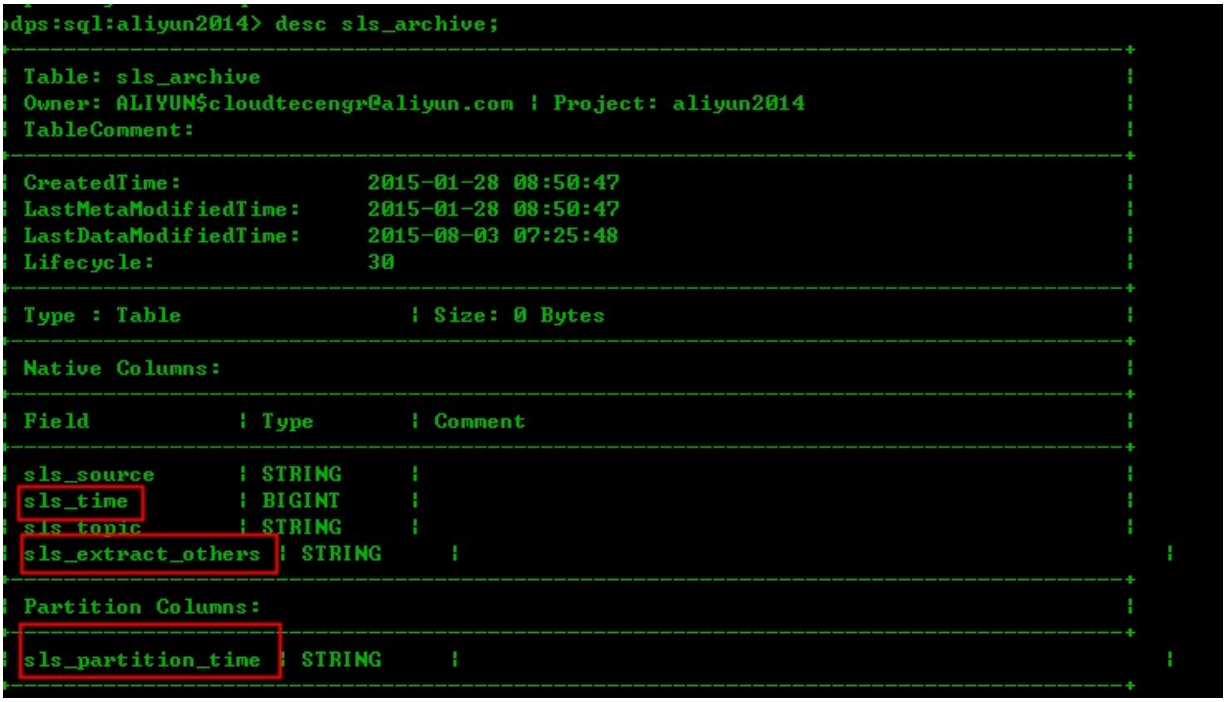
遇到这些情况时，投递策略为忽略异常的日志行，并继续任务，在该次任务中其它分区正确的日志行可以成功同步。

因此，在配置字段映射不当的情况下，可能出现任务成功但是表中缺少数据的情况，请修改分区列配置。建议使用保留字段__partition_time__作为分区。

更多细节请参考MaxCompute投递相关限制说明。

5.2 投递MaxCompute时的参数时间

日志服务投递日志至MaxCompute中有多个时间参数。



其中涉及到的时间参数及其含义如下：

时间参数	含义
sls_time	sls的服务器收到日志的时间
sls_extract_others	是用户真实的时间
sls-partition_time	MaxCompute归档的时间

说明：

sls_time一般都是略晚于sls_extract_others的时间。

如您的问题仍未解决，请联系售后技术支持。

6 命令行工具CLI

6.1 CLI的主要功能

日志服务命令行工具CLI支持以命令行工具完成常见功能，例如日志查询、完整性检查与自动分页，并支持多账户与跨域复制。

CLI有哪些主要功能？

日志服务命令行工具CLI的主要功能包括：

- 配置*CLI*
- 创建*Logtail*配置
- 跨域复制项目配置
- 拉取日志
- 查询日志
- *Elasticsearch* 数据迁移
- 支持的灵活时间格式
- 高速跨域日志复制、历史数据重新索引与数仓投递

7 计费

7.1 停用日志服务

若您不再需要使用日志服务，可以通过删除所有数据来停用日志服务。

背景信息

若您不再需要使用日志服务，删除所有Project和Logstore以清除所有日志数据。

> 说明：
>
> - 日志数据清除的当天仍会产生存储等费用，次日会收到账单。清除数据第三天开始不会收到日志服务的账单，因为从清除数据第二天开始没有产生任何计费项。
> - 当您的Project被删除后，其中的所有日志数据及配置信息都会永久释放，不可恢复。所以，在删除Project 前请慎重确认，避免数据丢失。

操作步骤

1. 登录日志服务控制台。
2. 在**Project**列表页面，找到需要删除的项目。
3. 单击项目名称右侧的**删除**。
4. 在弹出的对话框中，单击**获取验证码**。
5. 输入您收到的校验码并单击**确认** 。

7.2 计费常见问题

问题列表：

1. 因日志服务造成账号欠费#应如何处理#

1. 因日志服务造成账号欠费，应如何处理？

日志服务为使用后计费的模式，每日为您输出账单并自动扣费，账单内容为您昨日的资源使用项目。如您欠费时间超过24小时，则您的服务将自动停止，而您所占用的存储空间的这部分资源仍会继续扣费， 因此欠费余额会累计。建议您在欠费后24H内补足欠费，以免服务停止，对您的业务造成损失。在您补足金额后，可以继续操作日志服务。

2. 只创建了**Project**和**Logstore**，为什么会有账单？

您如果创建过Project和Logstore，则会默认创建Shard预留资源。正如创建Logstore时的页面提示，日志服务对Shard收取少量的资源预留费用。根据当前的计费策略，Shard的免费额度为31天*个，如您创建了2个Shard，则15天后开始计费。如果您不再需要该Shard，您可以直接删除Project和Logstore。删除当日的资源使用费用将会在次日为您发送账单，隔日后将不再有该Project的账单。

详细计费项目请参考***#unique_72***。

3. 如何关闭日志服务？

如果您不再需要使用日志服务，您可以直接删除账号下所有Project，相当于关闭了日志服务，次日开始不再计费。如果您的账号处于欠费状态，请补足欠费后再删除Project。如您账号下已没有任何日志服务业务和资源，隔日后您不会收到日志服务的账单。