云日志服务

最佳实践

文档版本	01
发布日期	2025-02-17

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 LTS 最佳实践总览

以下罗列了云日志服务（LTS）相关的最佳实践。

表 1-1 最佳实践总览

分类	最佳实践	场景说明
日志接入	采集第三方云厂商、互联网数据中心、华为云其他Region云主机日志到LTS	本实践主要介绍将阿里云主机日志采集到华为云LTS的操作步骤，互联网数据中心和华为云上跨Region采集日志的操作方式与采集阿里云主机日志的方式类似。
日志接入	采集第三方云厂商、互联网数据中心、华为云其他Region的Kubernetes日志到LTS	本实践主要介绍将阿里云Kubernetes日志采集到华为云LTS的操作步骤，互联网数据中心和华为云上跨Region采集日志的操作方式与采集阿里云主机日志的方式类似。
日志接入	采集Syslog汇聚服务器日志到LTS	本实践主要介绍通过Syslog协议将日志上传到日志服务的操作步骤。您需要购买ECS作为Syslog汇聚服务器，Linux服务器默认自带Syslog，目前华为云主机默认未配置接收远程Syslog写入，需要手动开启。
日志接入	将自建ELK日志导入云日志服务LTS	本实践主要介绍使用自定义Python脚本和LTS采集器ICAgent，协助用户将日志从Elasticsearch（简称ES）迁移到LTS中。
日志接入	使用Flume采集器上报日志到LTS	本实践主要介绍使用Flume系统采集日志，并且通过LTS侧提供的KAFKA协议方式上报日志。
日志接入	通过ECS接入LTS采集Zabbix数据	本实践介绍将Zabbix中的监控数据采集到云日志服务的日志流中。
日志接入	采集多渠道日志数据到LTS	本实践介绍采集多渠道日志数据到LTS。

分类	最佳实践	场景说明
日志搜索与分析	在LTS页面分析华为云ELB日志	本实践主要介绍将ELB日志接入LTS后，配置日志结构化后，即可进行日志搜索分析。
日志搜索与分析	通过LTS仪表盘可视化ELB日志分析结果	本实践主要介绍将ELB日志接入LTS后，配置日志结构化后，支持使用仪表盘将日志可视化，可以更直观的分析日志数据。
日志搜索与分析	在LTS页面分析华为云WAF日志	本实践主要介绍将WAF日志接入LTS后，配置日志结构化后，即可进行日志搜索分析。
日志搜索与分析	将LTS日志查询页面嵌入用户自建系统	本实践主要介绍通过统一身份认证服务IAM的联邦代理机制实现用户自定义身份代理，再将LTS登录链接嵌入至客户自建系统实现无需在华为云官网登录就可在自建系统界面查询LTS日志。
日志转储	批量修改LTS日志文件转储时区	本实践主要介绍通过Python脚本结合LTS API接口实现自定义的批量操作。
日志计费	通过日志流标签统计不同部门在LTS的费用开销	本实践主要介绍为了统计企业内部不同部门在LTS的费用开销情况，您可以在LTS的日志流上添加标签用于识别不同的业务部门，LTS在上传话单给费用中心时会带上这些标签信息。

2 日志接入

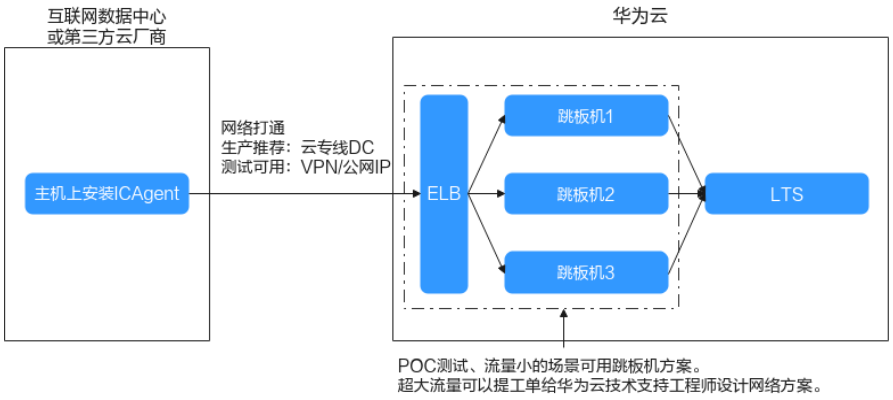
2.1 采集第三方云厂商、互联网数据中心、华为云其他Region 云主机日志到 LTS

方案概述

云上用户经常会遇到多云或者跨Region采集日志的场景，典型场景有两种，分别是：

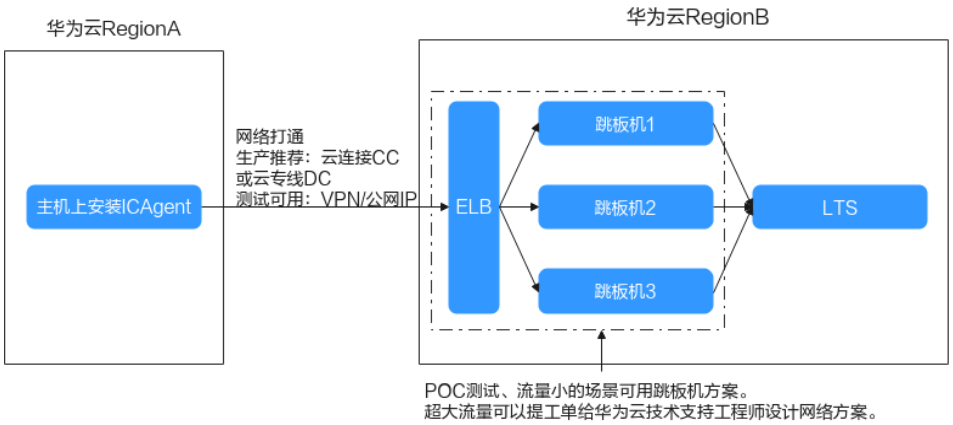
- 场景一：将互联网数据中心（Internet Data Center，以下简称IDC）或者第三方云厂商的日志采集到华为云LTS。

  图 2-1 第三方云厂商日志采集



- 场景二：将华为云RegionA的日志采集到华为云RegionB的LTS。

图 2-2 跨 Region 日志采集



对于场景一和场景二，您需要先打通网络，再安装ICAgent，最后按照日志接入向导即可以将日志采集到LTS。

- **ICAgent**：ICAgent是华为云日志服务的日志采集器，通过在主机上安装ICAgent，您可以将日志采集到LTS。安装ICAgent前，请确保本地浏览器的时间、时区与主机的时间、时区一致。
- **网络互通**：
  - 场景一：自建IDC或者第三方云厂商与华为云之间网络互通的典型方式是云专线DC，如果没有专线，您可以尝试VPN/公网IP方式。
  - 场景二：华为云不同Region之间网络互通典型的方式是云连接CC/云专线DC，您也可以使用VPN/公网IP方式。
- **跳板机**
  - 安装在自建IDC/第三方云厂商/跨华为云Region的ICAgent无法直接访问华为云管理面上报日志的网段，需要配置跳板机进行数据转发；当您在进行POC测试，或者日志流量并不大的情况下，可以使用跳板机的方案。对于大流量的日志场景，如果您希望在生产环境中去掉跳板机，请**提交工单**给华为云网络技术支持工程师帮您设计网络直通的方案。
  - 典型的跳板机配置是2vCPUs | 4GB，每台跳板机可以支持约30MB/s的流量转发，您可以根据自身的日志流量配置合理数量的跳板机，多台跳板机配置ELB进行流量分发。

本文将详细介绍将阿里云主机日志采集到华为云LTS的操作步骤，客户自建IDC和华为云上跨Region采集日志的操作方式与采集阿里云主机日志的方式类似。

以下将阿里云-华北二-北京局点的日志采集到华为云华东-上海一局点的LTS服务，云主机的操作系统为Linux环境。

资源规划

表 2-1 资源规划

区域	资源	资源说明
华东-上海一	弹性云服务器ECS	推荐CentOS 6.5 64bit及其以上版本的镜像， 最低规格为1vCPUs | 1GB，推荐规格为2vCPUs | 4GB。

区域	资源	资源说明
	弹性负载均衡 ELB	• 购买弹性负载均衡 ELB时，选择与弹性云服务器ECS相同的VPC。 • 新建弹性公网IP作为跳板机连接的IP。 • 根据业务量购买带宽，并进行适配 。

在华为云华东-上海一购买 ELB 和云主机作为跳板机

步骤1 登录云服务控制台，购买弹性云服务器 ECS。

非华为云上的服务器安装ICAgent，需要先在华为云上购买一台弹性云服务器作为跳板机。

步骤2 购买弹性负载均衡 ELB，并添加TCP监听器和后端服务器组。

1. 添加监听器。分别为端口30200、30201、8149、8923、8102添加监听器，具体请参见**添加TCP监听器**。
2. 为后端服务器组添加跳板机，具体请参见**添加后端服务器**。

步骤3 配置跳板机安全组规则，并开通转发端口。

1. 修改跳板机ECS使用的安全组规则。
   a. 单击ECS主机名称进入详情页。
   b. 在安全组页签，单击具体的安全组名，进入安全组详情页。
   c. 在该安全组详情页，选择“入方向规则”页签，单击“添加规则 ”。将安全组的入方向端口8149、8102、8923、30200、30201、80开启，保证非华为云的VM到跳板机ECS的数据连通性。

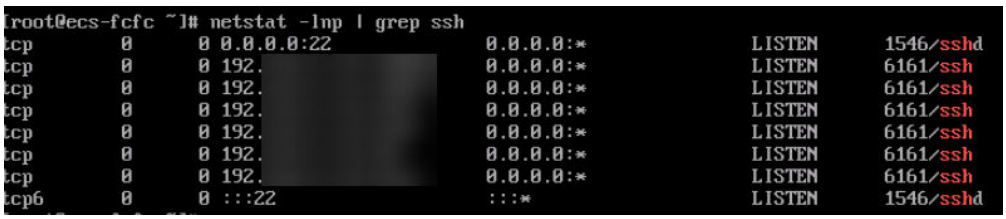
图 2-3 修改安全组规则

2. 在云日志服务控制台，选择“主机管理 > 主机”，单击“安装ICAgent”，进入安装ICAgent详情页面。按下图配置，“跳板机私有IP”填入ECS私有IP，生成安装命令。

图 2-4 安装 ICAgent

3. 复制命令以root用户登录跳板机，执行**SSH Tunnel**转发命令，根据命令提示输入root用户密码即可。
4. 执行**netstat -lnp | grep ssh**命令查看对应端口是否被侦听，如果返回结果如下图所示，说明TCP端口已开通。
   - 在浏览器地址栏里输入“http://跳板机ECS的IP地址”。如果访问成功，说明安全组规则已经生效。
   - 如果跳板机ECS掉电重启，请重新执行安装ICAgent页面生成的安装命令。如果您在生产环境中使用，请将上述端口转发命令配置为开机启动任务。

图 2-5 查看端口



----结束

在阿里云主机上安装 ICAgent

步骤1 获取AK/SK，请参考**如何获取访问密钥AK/SK**。

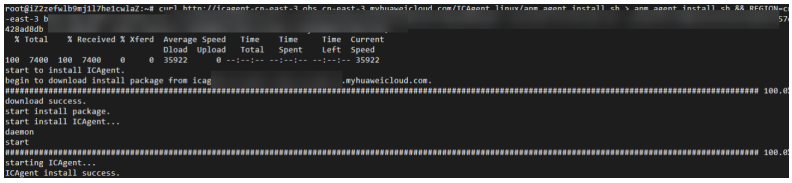
步骤2 输入跳板机连接IP（注意需要填入ECS弹性公网IP），生成ICAgent安装命令。

- 请确保替换正确的AK/SK，否则将无法安装ICAgent。
- 跳板机连接IP：使用EIP方式连接，为跳板机弹性公网IP，使用云专线VPC对等连接方式，为跳板机VPC内网IP。

步骤3 复制命令root用户登录阿里云ECS，执行ICAgent安装命令进行安装，当显示“ICAgent install success”时，表示安装成功。

如果是华为云跨Region采集日志，例如：将华为云华东-上海一局点采集日志到华为云华南-广州局点，需要在华南-广州局点购买ELB和跳板机ECS，然后将ICAgent的安装命令在华东-上海一局点的跳板机ECS上执行。

图 2-6 查看 ICAgent 安装状态



步骤4 安装成功后，在云日志服务左侧导航栏中选择“主机管理 > 主机”，查看该服务器中ICAgent状态为“运行”。

----结束

将日志接入 LTS

步骤1 登录华为云云日志服务控制台，在左侧导航栏中，选择“主机管理 > 主机组”，单击“新建主机组”，填写“主机组名称”并勾选主机，即可创建完成主机组。

步骤2 配置日志接入规则。具体操作请参考**ECS接入**。

----结束

查看日志流

在云日志服务（LTS）的日志管理页面，单击目标日志流名称进入详情页面，查看有日志即可证明阿里云主机日志已成功上报到LTS。

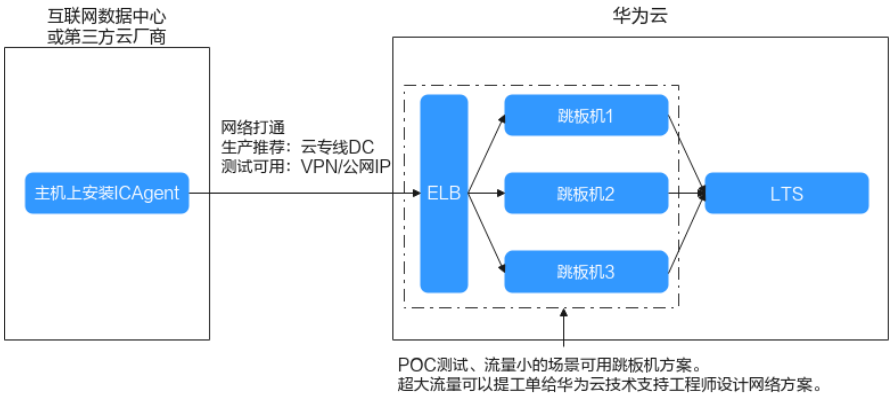
2.2 采集第三方云厂商、互联网数据中心、华为云其他 Region 的 Kubernetes 日志到 LTS

方案概述

云上用户经常会遇到多云或者跨Region采集Kubernetes日志场景，典型场景有两种，分别是：

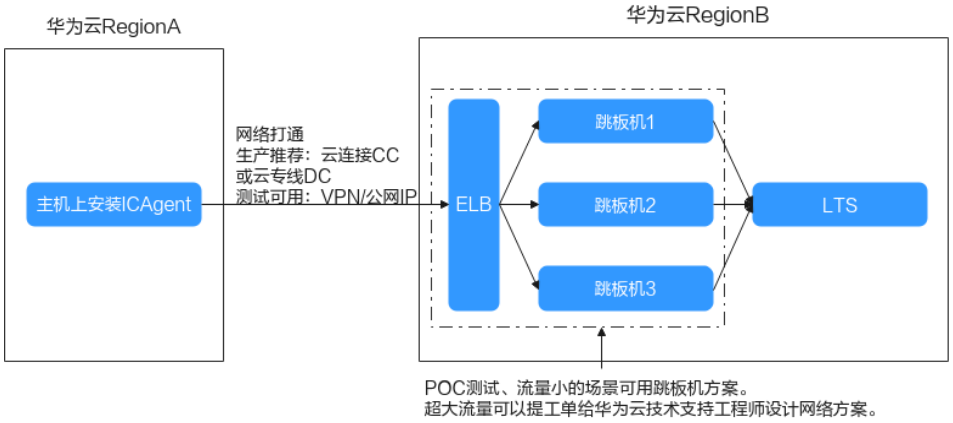
- 场景一：将互联网数据中心（Internet Data Center，以下简称IDC）或者第三方云厂商的日志采集到华为云LTS。

图 2-7 第三方云厂商日志采集



- 场景二：将华为云RegionA的日志采集到华为云RegionB的LTS。

图 2-8 跨 Region 日志采集



对于场景一和场景二，您需要先打通网络，再安装ICAgent，最后按照日志接入向导即可以将日志采集到LTS。

- **ICAgent**：ICAgent是华为云日志服务的日志采集器，通过在主机上安装ICAgent，您可以将日志采集到LTS。安装ICAgent前，请确保本地浏览器的时间、时区与主机的时间、时区一致。
- **网络互通**：
  - 场景一：自建IDC或者第三方云厂商与华为云之间网络互通的典型方式是云专线DC，如果没有专线，您可以尝试VPN/公网IP方式。
  - 场景二：华为云不同Region之间网络互通典型的方式是云连接CC/云专线DC，您也可以使用VPN/公网IP方式。
- **跳板机**
  - 安装在自建IDC/第三方云厂商/跨华为云Region的ICAgent无法直接访问华为云管理面上报日志的网段，需要配置跳板机进行数据转发；当您在进行POC测试，或者日志流量并不大的情况下，可以使用跳板机的方案。对于大流量的日志场景，如果您希望在生产环境中去掉跳板机，请**提交工单**给华为云网络技术支持工程师帮您设计网络直通的方案。
  - 典型的跳板机配置是2vCPUs | 4GB，每台跳板机可以支持约30MB/s的流量转发，您可以根据自身的日志流量配置合理数量的跳板机，多台跳板机配置ELB进行流量分发。

本文将详细介绍将阿里云主机日志采集到华为云云日志服务（LTS）的操作步骤，客户自建IDC和华为云上跨region采集日志的操作方式与采集阿里云主机日志的方式类似。

以下将阿里云-华北二北京局点的日志采集到华为云华东-上海一局点的LTS服务，云主机的操作系统为Linux环境。

资源规划

表 2-2 资源规划

区域	资源	资源说明
华东-上海一	弹性云服务器 ECS	推荐CentOS 6.5 64bit及其以上版本的镜像， 最低规格为1vCPUs | 1GB，推荐规格为2vCPUs | 4GB。
	弹性负载均衡 ELB	• 购买弹性负载均衡 ELB时，选择与弹性云服务器ECS相同的VPC。 • 新建弹性公网IP作为跳板机连接的IP。 • 根据业务量购买带宽，并进行适配 。

在华为云华东-上海一购买 ELB 和云主机作为跳板机

步骤1 登录云服务控制台，购买弹性云服务器 ECS。

非华为云上的服务器安装ICAgent，需要先在华为云上购买一台弹性云服务器作为跳板机。

步骤2 购买弹性负载均衡 ELB，并添加TCP监听器和后端服务器组。

1. 添加监听器。分别为TCP端口30200、30201、8149、8923、8102添加监听器，具体请参见**添加TCP监听器**。
2. 为后端服务器组添加跳板机，具体请参见**添加后端服务器**。

步骤3 配置跳板机安全组规则，并开通转发端口。

1. 修改跳板机ECS使用的安全组规则。
   a. 单击ECS主机名称进入详情页。
   b. 在安全组页签，单击具体的安全组名，进入安全组详情页。
   c. 在该安全组详情页，选择“入方向规则”页签，单击“添加规则 ”。将安全组的入方向端口8149、8102、8923、30200、30201、80开启，保证非华为云的VM到跳板机ECS的数据连通性。

图 2-9 修改安全组规则

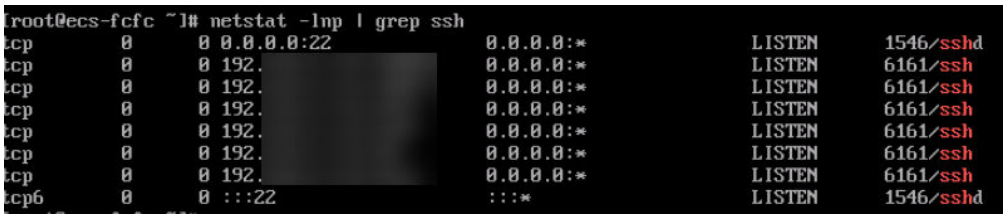
2. 在云日志服务控制台，选择“主机管理 > 主机”，单击“安装ICAgent”，进入安装ICAgent详情页面。按下图配置，“跳板机私有IP”填入ECS私有IP，生成安装命令。

图 2-10 安装 ICAgent

3. 复制命令以root用户登录跳板机，执行**SSH Tunnel**转发命令，根据命令提示输入root用户密码即可。

4. 执行**netstat -lnp | grep ssh**命令查看对应端口是否被侦听，如果返回结果如下图所示，说明TCP端口已开通。
   - 在浏览器地址栏里输入“http://跳板机ECS的IP地址”。如果访问成功，说明安全组规则已经生效。
   - 如果跳板机ECS掉电重启，请重新执行安装ICAgent页面生成的安装命令。如果您在生产环境中使用，请将上述端口转发命令配置为开机启动任务。

图 2-11 查看端口



----结束

配置日志接入

Kubernetes集群在一个节点上安装成功即可，不需要所有节点重复操作。

请提前获取AK/SK，请参考**如何获取访问密钥AK/SK**。

步骤1 配置跳板机。

1. 在弹性云服务器管理控制台找到已创建的机器，获取私有IP。

图 2-12 获取私有 IP

2. 在云日志服务LTS（LTS）控制台，选择“主机管理 > 主机”，单击“安装ICAgent”，进入安装ICAgent详情页面，“跳板机私有IP”填入ECS私有IP，生成安装口令并复制。

图 2-13 安装 ICAgent

3. 登录跳板机ECS，执行上一步的命令并根据提示输入节点密码，若无报错，表示安装成功。

图 2-14 执行生成的安装命令

4. 在“安装ICAgent”页面，“跳板机连接IP”输入跳板机的公网ip，确保已勾选“为了避免泄漏您的AK/SK，请勾选此处，执行关闭历史记录命令。”
5. 复制ICAgent安装命令并在跳板机ECS上执行，根据提示输入当前账号的AK、SK，当显示“ICAgent install success”时，表示安装成功。

步骤2 配置日志接入规则。具体操作请参考**自建K8s应用日志接入LTS**。

----结束

查看日志流

在云日志服务（LTS）的日志管理页面中，单击目标日志流名称进入详情页面，查看有日志即可证明阿里云Kubernetes日志已成功上报到LTS。

2.3 采集 Syslog 汇聚服务器日志到 LTS

背景信息

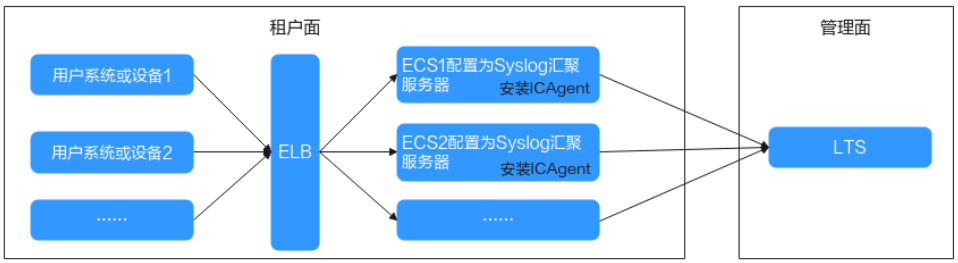
Syslog是网络上各种设备将日志收集到日志服务器的一种数据协议，它几乎被所有的网络设备支持，并且能够记录多种事件类型的日志消息，支持syslog的设备常见的有路由器、交换机、打印机等，甚至unix-like的服务器也可以支持产生syslog消息，用以记录用户的登录、防火墙事件、apache或者nginx access日志等。

Syslog主要是基于RFC5424和RFC3164定义相关格式规范，RFC3164协议是2001年发布的，RFC5424协议是2009年发布的升级版本。因为新版兼容旧版，且新版本解决了很多问题，因此推荐使用RFC5424协议。

本文介绍通过Syslog协议将日志上传到日志服务的操作步骤。您需要购买ECS作为Syslog汇聚服务器，Linux服务器默认自带Rsyslog，目前华为云主机默认未配置接收远程Syslog写入，需要手动开启。

方案概述

图 2-15 方案流程图



- 你可以购买Linux云主机，配置为Syslog汇聚服务器，用于接收其他设备发送的日志数据；Syslog服务器默认接收日志大小为1024字节，超过会截断。
- 单台Syslog服务器处理日志能力为10MB/s，如果您的日志量较大，或者希望可靠性更高，可以购买多台ECS配置为Syslog服务器，并配置ELB负载均衡分发流量。
- 您需要在Syslog服务器上安装ICAgent，并配置日志采集规则，就可以将日志采集到LTS。

资源规划

购买两台ECS机器，一台ECS作为Syslog汇聚服务器，另一台用为业务ECS模拟客户系统或者设备发送日志。

购买弹性云服务器

步骤1 登录管理控制台，选择“计算 > 弹性云服务器 ECS”。

步骤2 购买一台弹性云服务器作为Syslog汇聚服务器。

推荐CentOS 6.5 64bit及其以上版本的镜像， 推荐规格为2vCPUs | 4GB。

步骤3 使用root用户登录Syslog服务器，安装ICAgent。

1. syslog服务器安全组出方向打开TCP协议的30200、30201、8149、8923、8102等端口，入方向需打开UDP的514端口作为syslog服务器默认监听端口。
2. 在云日志服务管理控制台，左侧导航栏选择“主机管理 > 主机”，进入“主机”页面。
3. 单击“安装ICAgent”， 安装系统选择“Linux”，主机类型选择“区域内主机”，“安装方式”选择“获取AK/SK凭证”，单击“复制命令”复制ICAgent安装命令，并手动替换AK/SK。
4. 使用root用户登录syslog服务器，执行ICAgent安装命令进行安装，当显示“ICAgent install success”时，表示安装成功。安装成功后，在云日志服务左侧导航栏中选择“主机管理 > 主机”，查看该服务器中ICAgent的状态。

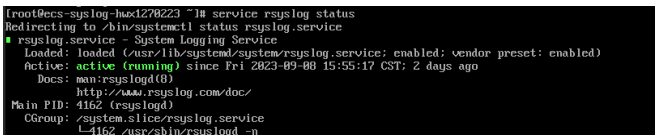
步骤4 打开RSyslog服务器监听接收功能。

目前华为云主机rsyslog服务器默认未配置接收远程syslog写入，需要手动开启。

1. 已登录云主机。
2. 修改rsyslog配置文件。
   ```
   vi /etc/rsyslog.conf
   ```
3. 配置文件中添加以下内容，开启tcp udp远程接收。
   ```
   # Provides UDP syslog reception
   $ModLoad imudp
   $UDPServerRun 514
   # Provides TCP syslog reception
   $ModLoad imtcp
   $InputTCPServerRun 514
   ```
4. 保存后重启rsyslog服务器，单击云服务器“更多 > 重启”。
5. 执行以下任意一条命令显示正常代表服务运行正常。

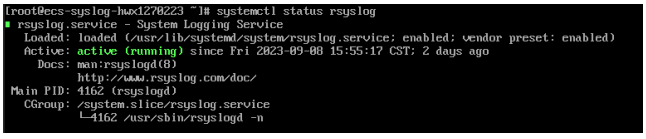
   执行“service rsyslog status”命令检查rsyslog运行状态为running。

   图 2-16 检查 rsyslog 运行状态

   执行“systemctl status rsyslog”命令检查rsyslog运行状态为running。

   图 2-17 rsyslog 运行状态

   执行“netstat -anp | grep 514”命令查看是否打开监听。

   图 2-18 查看是否打开监听

步骤5 配置采集syslog日志。

1. 创建主机组 ，在云日志服务 LTS控制台左侧导航栏中选择“主机管理 > 主机组”，单击“新建主机组”，设计“主机组名称”，勾选主机即可。
2. 左侧导航栏中选择“日志接入 > 云主机ECS-文本日志”。
3. 选择日志流。
4. 选择主机组。
5. 采集配置，配置采集路径为/var/log/messages。

步骤6 登录业务ecs验证。

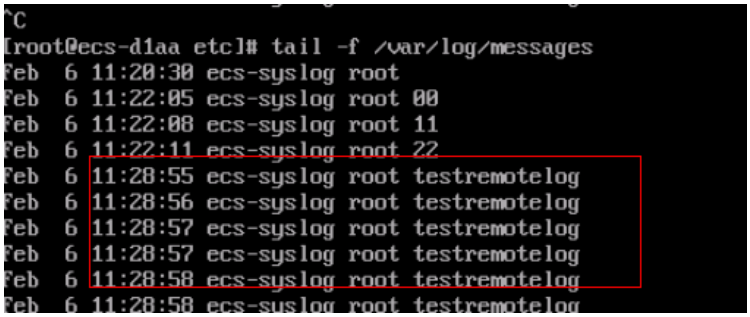
当您的业务系统或者设备输出日志后，即可在LTS界面查看。登录业务ecs使用logger -n x.x.x.x -P 514 testremotelog命令发送syslog至汇聚服务器，x.x.x.x为syslog服务器ip（公网ip/私有ip）地址，testremotelog为发送日志内容，可以自定义。

执行成功后，配置的日志组日志流里可以查看到该条日志。

或登录syslog汇聚服务器，查看/var/log/messages中是否存在testremotelog日志。

```
tail -f /var/log/messages
```

图 2-19 查看是否存在 testremotelog 日志



步骤7 通过多台syslog服务器和elb实现负载均衡。

目前采集单台syslog服务器处理日志能力为10MB/s，如果客户日志业务量较大，可以使用多台syslog服务器和elb实现扩容和负载均衡。

1. 创建syslog汇聚服务器和安装ICAgent。
2. 请参考**实现单个Web应用的负载均衡**创建负载均衡器。
3. 分别为TCP/UDP的端口和514端口添加监听器，请参考**添加TCP监听器**。
4. 为后端添加服务器组，请参考**后端服务器组**。

----结束

2.4 将自建 ELK 日志导入云日志服务 LTS

方案概述

ELK是Elasticsearch、Logstash和Kibana的简称，它们组合起来提供了业界最常用的日志分析和可视化工具。

- Elasticsearch是一个基于Lucene的开源、分布式、RESTful搜索和分析引擎。

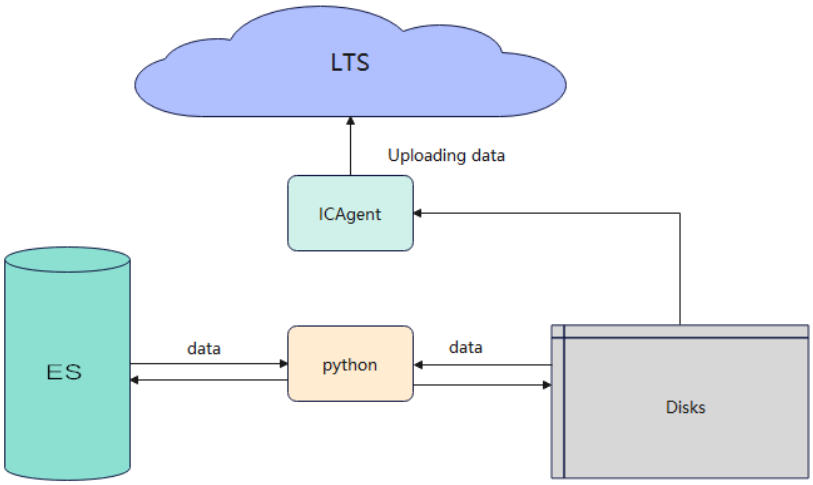
- Logstash是一个开源的、服务器端的数据处理管道，能够同时从多个来源实时接收、转换并将数据发送到用户选择的“存储库”。通常用于日志的收集、过滤和转发。
- Kibana是一个开源的分析和可视化平台，用于数据的可视化、仪表盘创建和搜索查询。通常与Elasticsearch一起使用。

华为云日志服务LTS在功能丰富度、成本、性能方面优于开源ELK方案，具体对比可以参考**云日志服务LTS对比自建ELK Stack有什么优势?** 。本文提供最佳实践，使用自定义Python脚本和LTS采集器ICAgent，协助用户将日志从Elasticsearch（简称ES）迁移到LTS中。

当前华为云支持ECS机器通过安装ICAgent来采集日志文件，因此可以基于该功能实现Elasticsearch日志导入云日志服务。

Elasticsearch数据先通过python脚本将数据落盘到ECS，然后通过LTS服务的日志接入功能，将落盘的日志文件采集到LTS服务。

图 2-20 方案流程图



将自建 ELK 日志导入云日志服务 LTS

步骤1 登录**云日志服务控制台**。

步骤2 请参考**安装ICAgent**在ECS主机安装ICAgent。

步骤3 配置ECS日志接入云日志服务，请参考**ECS接入**。

步骤4 脚本执行前期准备。以下示例仅供参考，请以实际信息填写为准。

- 首次使用python，需要安装python环境。
- 首次使用ES时需要安装对应ES版本的python数据包，本次方案测试使用的elasticsearch为7.10.1版本。

```
pip install elasticsearch==7.10.1
```

- 方案测试使用的ES为华为云CSS服务创建的ES。

步骤5 执行用来构造索引数据的python脚本，如果索引已经有数据，忽略这一步，直接执行**步骤6**。

python脚本需执行在ECS机器，脚本命名为xxx.py格式，构造数据请参考如下示例：

以下斜体字段需按照实际情况进行修改，参考示例是插入1000条数据，内容为：This is a test log,Hello world!!!\n；

- **index**：要创建的索引名称，参考示例为: *test*。
- **链接ES**：ES的访问url，参考示例为：*http://127.0.0.1:9200*。

```
from elasticsearch import Elasticsearch
def creadIndex(index):
    mappings = {
        "properties": {
            "content": {
                "type": "text"
            }
        }
    }
    es.indices.create(index=index, mappings=mappings)
def reportLog(index):
    i = 0
    while i < 1000:
        i = i + 1
        body = {"content": "This is a test log,Hello world!!!\n"}
        es.index(index=index,body=body)
if __name__ == '__main__':
    #索引名称
    index = 'test'
    #链接ES
    es = Elasticsearch("http://127.0.0.1:9200")
    creadIndex(index)
    reportLog(index)
```

步骤6 构建python读写脚本，用来将ES数据写入磁盘，输出文件路径需与配置日志接入规则一致。

脚本需执行在ecs机器，命名xxx.py格式，写入磁盘数据脚本请参考如下示例：

以下斜体字段需按照实际情况进行修改。

- **index**：字段为索引名，参考示例为: *test*。
- **pathFile** ： 为数据写入磁盘绝对路径，参考示例为: */tmp/test.log*。
- **scroll_size** ： 为索引滚动查询大小，参考示例为： *100*。
- **链接ES**：ES的访问url，参考示例为：*http://127.0.0.1:9200*。

```
from elasticsearch import Elasticsearch
def writeLog(res, pathFile):
    data = res.get('hits').get('hits')
    i = 0
    while i < len(data):
        log = data[i].get('_source').get('content')
        file = open(pathFile, 'a', encoding='UTF-8')
        file.writelines(log)
        i = i + 1
    file.flush()
    file.close()
if __name__ == '__main__':
    #索引名称
    index = 'test'
    #输出文件路径
    pathFile = '/tmp/' + index + '.log'
    #滚动查询一次滚动大小，默认为100
    scroll_size = 100
    #链接ES
    es = Elasticsearch("http://127.0.0.1:9200")
    init = True
    while 1:
        if (init == True):
            res = es.search(index=index, scroll="1m", body={"size": scroll_size})
```



```
        init =False
    else:
        scroll_id = res.get("_scroll_id")
        res = es.scroll(scroll="1m", scroll_id=scroll_id)
    if not res.get('hits').get('hits'):
        break
    writeLog(res, pathFile)
```

步骤7 执行命令前，请确保python已安装成功，在ECS执行如下命令，将ES索引数据写入磁盘。

```
python xxx.py
```

步骤8 查看数据是否成功查询及写入磁盘。

参考示例demo写入磁盘路径为：*/tmp/test.log*，操作时需要填写实际使用的路径，执行如下命令可以查看数据写入磁盘情况。

```
tail -f /tmp/test.log
```

步骤9 登录云日志服务控制台，在“日志管理”页面，单击日志流名称进入详情页面，在“日志搜索”页签查看到日志数据，即代表采集成功。

----结束

2.5 使用 Flume 采集器上报日志到 LTS

Flume是一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统，Flume支持在日志系统中定制各类数据发送方，用于收集数据；同时，Flume提供对数据进行简单处理，并写到各种数据接受方的能力。

用户使用Flume系统采集日志，并且通过LTS侧提供的KAFKA协议方式上报日志。以下是部分常用数据采集场景示例：

1. **使用Flume采集文本日志上报到LTS**
2. **使用Flume采集数据库表数据并且上报至LTS**
3. **使用Flume采集syslog协议传输的日志上报到LTS**
4. **通过Flume采集TCP/UDP协议传输的日志上报到LTS**
5. **通过Flume采集SNMP协议上报的设备管理数据并发送到LTS**
6. **使用默认拦截器处理日志**
7. **自定义拦截器处理日志**
8. **使用外部数据源丰富日志内容并上报到LTS**

前提条件

- 用户机器已经安装了JDK。
- 用户已经安装Flume，并且需要在Flume中配置文件中配置JDK路径。

使用 Flume 采集文本日志上报到 LTS

支持使用Flume采集文本日志内容上报至LTS，参考如下示例添加采集文本日志的conf文件。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

```
#Named
a1.sources = r1
a1.channels = c1
a1.sinks = k1
```



```
#Source
a1.sources.r1.type = TAILDIR
a1.sources.r1.channels = c1
a1.sources.r1.filegroups = f1
a1.sources.r1.filegroups.f1 = /tmp/test.txt
a1.sources.r1.fileHeader = true
a1.sources.r1.maxBatchCount = 1000

#Channel
a1.channels.c1.type = memory
a1.channels.c1.capacity = 10000
a1.channels.c1.transactionCapacity = 100

#Sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";

#Bind
a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

使用 Flume 采集数据库表数据并且上报至 LTS

使用Flume采集数据库表数据并且上报至LTS，实现对表数据变动监控。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

步骤1 在**https://github.com/keedio/flume-ng-sql-source**页面下载flume-ng-sql-source插件，转换为jar包并取名为flume-ng-sql-source.jar，打包前注意将pom文件中的flume-ng-core 版本与flume安装版本保持一致，并且将jar包放在安装Flume包路径的lib目录下面，例如**FLUME_HOME/lib**目录下（例子中的**FLUME_HOME**为Flume安装路径，仅供参考，请以实际安装路径为准）。

步骤2 添加MySQL驱动到**FLUME_HOME/lib**目录下：

1. 下载MySQL驱动。
   ```
   wget https://dev.mysql.com/get/Downloads/Connector-J/mysql-connector-java-5.1.35.tar.gz
   ```
2. 将驱动包解压并打为jar包。
   ```
   tar xzf mysql-connector-java-5.1.35.tar.gz
   ```
3. 将jar包存放在FLUME_HOME/lib/路径。
   ```
   cp mysql-connector-java-5.1.35-bin.jar  FLUME_HOME/lib/
   ```

步骤3 添加采集MySQL的conf文件。

```
# a1表示agent的名称
# source是a1的输入源
# channels是缓冲区
# sinks是a1输出目的地，本例子sinks使用了kafka
a1.channels = c1
a1.sources = r1
a1.sinks = k1

#source
a1.sources.r1.type = org.keedio.flume.source.SQLSource
# 连接mysql的一系列操作，{mysql_host}改为您虚拟机的ip地址，可以通过ifconfig或者ip addr查看,
{database_name}改为数据库名称
# url中要加入?useUnicode=true&characterEncoding=utf-8&useSSL=false，否则有可能连接失败
a1.sources.r1.hibernate.connection.url = jdbc:mysql://{mysql_host}:3306/{database_name}?
useUnicode=true&characterEncoding=utf-8&useSSL=false
```

```
# Hibernate Database connection properties
# mysql账号，一般都是root
a1.sources.r1.hibernate.connection.user = root
# 填入您的mysql密码
a1.sources.r1.hibernate.connection.password = xxxxxxxx
a1.sources.r1.hibernate.connection.autocommit = true
# mysql驱动
a1.sources.r1.hibernate.dialect = org.hibernate.dialect.MySQL5Dialect
a1.sources.r1.hibernate.connection.driver_class = com.mysql.jdbc.Driver
# 存放status文件
a1.sources.r1.status.file.path = FLUME_HOME/bin
a1.sources.r1.status.file.name = sqlSource.status
# Custom query
# 填写需要采集的数据表名{table_name}，也可以使用下面的方法：
a1.sources.r1.custom.query = select * from {table_name}


#Sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";


a1.channels.c1.type = memory
a1.channels.c1.capacity = 10000
a1.channels.c1.transactionCapacity = 10000
a1.channels.c1.byteCapacityBufferPercentage = 20
a1.channels.c1.byteCapacity = 800000
```

步骤4 启动Flume后，即可开始采集数据库中的表数据到LTS。

----结束

使用 Flume 采集 syslog 协议传输的日志上报到 LTS

Syslog协议是一种用于在IP网络中传输日志消息的协议，通过Flume将syslog协议传输的日志采集并上报到LTS。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

- 接收UDP日志，参考如下示例添加采集Syslog协议的conf文件。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1

a1.sources.r1.type=syslogudp
#host_port为syslog服务器的端口
a1.sources.r1.port = {host_port}
#host_ip为syslog服务器的ip地址
a1.sources.r1.host = {host_ip}
a1.sources.r1.channels = c1

a1.channels.c1.type = memory

#Sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
```



```
required username="${projectId}" password="${accessKey}#${accessSecret}";
a1.sinks.k1.channel = c1
```

- 接收TCP日志，参考如下示例添加采集Syslog协议的conf文件。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1

a1.sources.r1.type=syslogtcp
#host_port为syslog服务器的端口
a1.sources.r1.port = {host_port}
#host_ip为syslog服务器的ip地址
a1.sources.r1.host = {host_ip}
a1.sources.r1.channels = c1

a1.channels.c1.type = memory

#Sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";
a1.sinks.k1.channel = c1
```

通过 Flume 采集 TCP/UDP 协议传输的日志上报到 LTS

通过Flume采集TCP/UDP协议传输的日志上报到LTS。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

- 采集TCP端口日志，参考如下示例添加采集端口的conf文件。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1

a1.sources.r1.type = netcat
a1.sources.r1.bind = 0.0.0.0
a1.sources.r1.port = {host_port}

a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";

a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

- 采集UDP端口日志，参考如下示例添加采集端口的conf文件。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1

a1.sources.r1.type = netcatudp
a1.sources.r1.bind = 0.0.0.0
a1.sources.r1.port = {host_port}
```

```
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";

a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

通过 Flume 采集 SNMP 协议上报的设备管理数据并发送到 LTS

通过Flume采集SNMP协议上报的设备管理数据并发送到LTS。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

- 监听SNMP协议通信端口号161。参考如下示例添加SNMP协议接受日志的conf。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1

a1.sources.r1.type = netcatudp
a1.sources.r1.bind = 0.0.0.0
a1.sources.r1.port = 161

a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username="${projectId}" password="${accessKey}#${accessSecret}";

a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

- 监听SNMP协议陷阱(Trap)通信的端口号162，参考如下示例添加SNMP协议接受日志的conf。

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1


a1.sources.r1.type = netcatudp
a1.sources.r1.bind = 0.0.0.0
a1.sources.r1.port = 162


a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic = ${logGroupId}_${logStreamId}
a1.sinks.k1.kafka.bootstrap.servers = ${ip}:${port}
a1.sinks.k1.kafka.producer.acks = 0
a1.sinks.k1.kafka.producer.security.protocol = SASL_PLAINTEXT
a1.sinks.k1.kafka.producer.sasl.mechanism = PLAIN
a1.sinks.k1.kafka.producer.compression.type = gzip
a1.sinks.k1.kafka.producer.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
```

```
required username="${projectId}" password="${accessKey}#${accessSecret}";

a1.channels.c1.type = memory
a1.channels.c1.capacity = 1000
a1.channels.c1.transactionCapacity = 100

a1.sources.r1.channels = c1
a1.sinks.k1.channel = c1
```

使用默认拦截器处理日志

使用Flume采集器时，拦截器是简单的插件式组件，设置在Source和Channel之间。Source接收到的事件Event，在写入Channel之前，拦截器都可以进行转换或者删除这些事件。每个拦截器只处理同一个Source接收到的事件。

- 时间戳拦截器

  该拦截器的作用是将时间戳插入到flume的事件报头中。如果不使用任何拦截器，flume接收到的只有message。时间戳拦截器的配置， 参数默认值描述type，类型名称timestamp，也可以使用类名的全路径preserveExisting为false。如果设置为true，若事件中报头已经存在，不会替换时间戳报头的值。source连接到时间戳拦截器的配置：

  ```
  a1.sources.r1.interceptors = timestamp
  a1.sources.r1.interceptors.timestamp.type=timestamp
  a1.sources.r1.interceptors.timestamp.preserveExisting=false
  ```

- 正则过滤拦截器

  在日志采集的时候，可能有一些数据是不需要的，添加过滤拦截器可以过滤掉不需要的日志，也可以根据需要收集满足正则条件的日志。参数默认值描述type，类型名称REGEX_FILTER。excludeEvents为false时默认收集匹配到的事件。如果为true，则会删除匹配到的event，收集未匹配到的。source连接到正则过滤拦截器的配置：

  ```
  a1.sources.r1.interceptors = regex
  a1.sources.r1.interceptors.regex.type=REGEX_FILTER
  a1.sources.r1.interceptors.regex.regex=(today)|(Monday)
  a1.sources.r1.interceptors.regex.excludeEvents=false
  ```

  这样配置的拦截器就只会接收日志消息中带有today或者Monday的日志。

- 搜索并替换拦截器

  拦截器基于Java正则表达式提供简单的基于字符串的搜索和替换功能。配置如下：

  ```
  # 拦截器别名
  a1.sources.r1.interceptors = search-replace
  # 拦截器类型，必须是search_replace
  a1.sources.r1.interceptors.search-replace.type = search_replace

  # 删除事件正文中的字符，根据正则匹配event内容
  a1.sources.r1.interceptors.search-replace.searchPattern = today
  # 替换匹配到的event内容
  a1.sources.r1.interceptors.search-replace.replaceString = yesterday
  # 设置字符集，默认是utf8
  a1.sources.r1.interceptors.search-replace.charset = utf8
  ```

自定义拦截器处理日志

在Flume中自定义拦截器的方式主要流程如下（以java语言为例），以下示例中的FLUME_HOME表示Flume的安装路径，例如/tools/flume（仅供参考），实际配置的时候，请以用户安装Flume的实际路径为准。

步骤1 创建MAVEN工程项目，引入Flume依赖。

根据集群中的 Flume 版本，引入 Flume 依赖，如下所示：

```
<dependencies>
    <dependency>
      <groupId>org.apache.flume</groupId>
      <artifactId>flume-ng-core</artifactId>
      <version>1.10.1</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

无需将该依赖打包进最后的JAR包中，故将其作用域设置为provided。

步骤2 创建类实现拦截器接口Interceptor，并且实现相关方法。

- initialize() 方法：初始化拦截器操作，读取配置信息、建立连接等。
- intercept(Event event) 方法：用于拦截单个事件，并对事件进行处理。接收一个事件对象作为输入，并返回一个修改后的事件对象。
- intercept(List<Event> list) 方法：事件批处理，拦截事件列表，并对事件列表进行处理。
- close() 方法：关闭拦截器，在这里释放资源、关闭连接等。

```
import org.apache.flume.Event;
import org.apache.flume.interceptor.Interceptor;

import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;

public class TestInterceptor implements Interceptor {
  @Override
  public void initialize() {

  }
  @Override
  public Event intercept(Event event) {

    // 获取事件数据
    String eventData = new String(event.getBody(), StandardCharsets.UTF_8);
    // 检查事件数据中是否包含指定字符串
    if (eventData.contains("hello")) {
      // 如果包含指定字符串，则过滤掉该事件，返回 null
      return null;
    }

    return event;
  }

  @Override
  public List<Event> intercept(List<Event> events) {
    // 创建一个新的列表，存储处理过后的事件
    List<Event> interceptedEvents = new ArrayList<>();
    for (Event event : events) {
      Event interceptedEvent = intercept(event);
      if (interceptedEvent != null) {
        interceptedEvents.add(interceptedEvent);
      }
    }
    return interceptedEvents;
  }

  @Override
  public void close() {

  }
```

```
}
```

步骤3 构建拦截器，拦截器的创建和配置通常是通过 Builder 模式来完成的，完整的代码如下所示：

```
import org.apache.flume.Context;
import org.apache.flume.Event;
import org.apache.flume.interceptor.Interceptor;

import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;

public class TestInterceptor implements Interceptor {
    @Override
    public void initialize() {
    }
    @Override
    public Event intercept(Event event) {
        // 获取事件数据
        String eventData = new String(event.getBody(), StandardCharsets.UTF_8);
        // 检查事件数据中是否包含指定字符串
        if (eventData.contains("hello")) {
            // 如果包含指定字符串，则过滤掉该事件，返回 null
            return null;
        }
        return event;
    }
    @Override
    public List<Event> intercept(List<Event> events) {
        List<Event> interceptedEvents = new ArrayList<>();
        for (Event event : events) {
            Event interceptedEvent = intercept(event);
            if (interceptedEvent != null) {
                interceptedEvents.add(interceptedEvent);
            }
        }
        return interceptedEvents;
    }

    @Override
    public void close() {

    }

    // 拦截器构建
    public static class Builder implements Interceptor.Builder {

        @Override
        public void configure(Context context) {

        }

        @Override
        public Interceptor build() {
            return new TestInterceptor();
        }

    }

}
```

步骤4 转换为jar包，并且将其上传至Flume安装路径下的lib文件夹下（请以用户安装Flume的实际路径为准）。

步骤5 编写配置文件，需要将自定义的拦截器配置进去。

拦截器全类名配置时需要注意，格式为拦截器的全类名 + $Builder。


```
# 拦截器配置
# 拦截器定义
a1.sources.r1.interceptors = i1
# 拦截器全类名
a1.sources.r1.interceptors.i1.type = TestInterceptor$Builder
```

步骤6 运行Flume即可。

----结束

KV解析日志：用空格分隔字符串并且转换为Map类型字符串。

```
public class TestInterceptor implements Interceptor {
    @Override
    public void initialize() {
    }

    @Override
    public Event intercept(Event event) {
        // 获取事件数据
        String eventData = new String(event.getBody(), StandardCharsets.UTF_8);
        Map<String, Object> splitMap = new HashMap<>();
        String[] splitList = eventData.split(" ");
        for (int i = 0; i < splitList.length; i++) {
            splitMap.put("field" + i, splitList[i].trim());
        }
        eventData.setBody(splitMap.toString().getBytes(StandardCharsets.UTF_8));
        return event;
    }

    @Override
    public List<Event> intercept(List<Event> events) {
        List<Event> interceptedEvents = new ArrayList<>();
        for (Event event : events) {
            Event interceptedEvent = intercept(event);
            if (interceptedEvent != null) {
                interceptedEvents.add(interceptedEvent);
            }
        }
        return interceptedEvents;
    }

    @Override
    public void close() {
    }
}
```

使用外部数据源丰富日志内容并上报到 LTS

Flume数据传输的基本单元，以Event的形式将数据从源头传输至目的地。Event由Header和Body两部分组成，Header用来存放该Event的一些属性，为K-V结构，Body用来存放该条数据，形式为字节数组。

有外部数据源时，如果您需要丰富日志内容，例如修改日志内容、添加字段、删除内容等操作，将修改内容添加至Event的body中，Flume才能将日志内容上报到LTS。例如使用Java自定义扩展日志内容。以下示例中的参数介绍请参考**使用KAFKA协议上报日志**。

```
import com.alibaba.fastjson.JSONObject;

import org.apache.flume.Context;
import org.apache.flume.Event;
import org.apache.flume.interceptor.Interceptor;

import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.List;
```



```
public class TestInterceptor implements Interceptor {
    @Override
    public void initialize() {
    }

    @Override
    public Event intercept(Event event) {
        // 获取事件数据，将原数据转换为json字符串并且添加额外字段
        String eventData = new String(event.getBody(), StandardCharsets.UTF_8);
        JSONObject object = new JSONObject();
        object.put("content", eventData);
        object.put("workLoadType", "RelipcaSet");
        eventData = object.toJSONString();
        eventData.setBody(eventData.getBytes(StandardCharsets.UTF_8));
        return event;
    }

    @Override
    public List<Event> intercept(List<Event> events) {
        List<Event> interceptedEvents = new ArrayList<>();
        for (Event event : events) {
            Event interceptedEvent = intercept(event);
            if (interceptedEvent != null) {
                interceptedEvents.add(interceptedEvent);
            }
        }
        return interceptedEvents;
    }

    @Override
    public void close() {
    }
}
```

2.6 通过 ECS 接入 LTS 采集 Zabbix 数据

Zabbix作为常用的开源监控系统，提供了丰富的告警规则用于系统监控。云日志服务LTS支持将Zabbix中的监控数据采集到日志流中。本文介绍通过ECS接入将Zabbix数据采集到云日志服务的操作步骤。

前提条件

- 准备好需要采集日志的ECS主机，详细请参考**购买弹性云服务器**。如果您已有可用的ECS主机，可重复使用，不需要再次创建。
- 已下载及在ECS机器上安装Zabbix，详细请参考**下载与安装Zabbix**。

步骤一：配置监控数据存储路径

Zabbix会将监控数据保存在其所在的机器上，您可以根据如下步骤设置监控数据的存储路径。

1. 登录Zabbix所在服务器。
2. 打开zabbix_server.conf文件。
   ```
   vim /etc/zabbix/zabbix_server.conf
   ```
3. 在zabbix_server.conf文件中，设置数据存储路径。
   ```
   ExportDir=/tmp/
   ```
4. 重启Zabbix服务，使配置生效。
   ```
   systemctl restart zabbix-server
   ```
   配置生效后，Zabbix会在/tmp目录下生产文件（文件名后缀为.ndjson），用于保存监控数据。

步骤二：配置 ECS 接入 LTS

步骤1 在左侧导航栏中，选择“接入 > 接入中心”，单击“云主机 ECS-文本日志”进行主机接入配置。

步骤2 进入选择日志流页面。

1. 单击“所属日志组”后的目标框，在下拉列表中选择具体的日志组（例如lts-group-ECS）。
2. 单击“所属日志流”后的目标框，在下拉列表中选择具体的日志流（例如lts-topic-ECS）。
3. 单击“下一步：选择主机组（可选）”。

步骤3 选择主机组，单击“下一步：采集配置”。

步骤4 采集配置，路径配置为/tem/**/*.ndjson，其余参数按照界面默认即可。更多设置请参考**云主机ECS文本日志接入LTS**。

图 2-21 采集配置

步骤5 单击“下一步：索引配置”，进入索引配置页面，按照界面默认参数配置即可，通过配置索引后，可对日志进行查询和分析操作。更多信息请参考**索引配置**。

步骤6 单击“提交”，日志接入成功，可以单击“返回接入配置列表”查看日志接入，在接入管理页签，则会生成一条接入配置信息。

步骤7 完成日志接入配置后，可以在云日志服务控制台实时查看上报的日志。

单击目标日志接入任务“所属日志流”列的日志流名称，即可进入日志流详情页。

步骤8 单击“实时日志”页签，查看实时日志。

日志大约每隔5秒钟上报一次，在日志消息区域，您最多需要等待5秒钟左右，即可查看实时上报的日志。

----结束

2.7 采集多渠道日志数据到 LTS

日志接入是指通过各种手段采集应用程序或服务在运行时产生的系统运行状态、错误信息、用户操作记录等日志信息，并将这些日志信息存储到指定的位置，以便后续分析和使用。这些日志信息对于系统运维、故障排查以及业务分析都具有重要意义。

云日志服务LTS提供实时日志接入功能，通过ICAgent插件、云服务接入、自建软件接入、API接入、SDK接入等多种方式将采集到的日志上报到LTS，满足不同场景日志接入的诉求。

- 支持40+华为云服务日志对接LTS。
- 支持Web、移动端iOS、Android和小程序日志接入LTS。
- 支持采集Logstash、Flume、Beats开源软件中的数据。
- 支持通过HTTP、HTTPS、Syslog、Kafka等标准协议接入数据。

应用场景

某个外卖平台型电商网站（包括用户、餐厅、配送员等），用户可以在网页、App、微信、支付宝等进行下单点菜，商家拿到订单后开始加工，并自动通知周围的快递员，最后由快递员将外卖送到用户手中。

在数据化运营的过程中，发现数据采集困难，例如：

- 多渠道：各种渠道数据分散，例如广告商、推广人员等。
- 多终端：网页版、公众账号、手机、浏览器（Web、移动端页面）等。
- 异构网：VPC、用户自建IDC，华为云ECS等。
- 多开发语言：核心系统Java、前端Nginx服务器、后台支付系统C++。
- 设备不同：商家有不同平台（X86、ARM）设备。

如果需要把散落在外部、内部的日志收集起来，可以通过LTS实现**统一管理日志**。

- **采集用户推广日志**
- **采集终端用户日志**
- **采集服务器日志**
- **采集不同网络环境下的日志数据**

统一管理日志

1. 在云日志服务LTS控制台创建日志组，详细操作请参考**管理日志组**。
2. 在云日志服务LTS控制台为不同数据源产生的日志创建日志流，详细操作请参考**管理日志流**，以下示例日志仅供参考，请以实际应用日志为准。
   - wechat-server（用于存储微信服务器访问日志）
   - wechat-app（用于存储微信服务器应用日志）
   - wechat-error（用于存储错误日志）
   - alipay-server
   - alipay-app
   - deliver-app（用于存储送货员App状态）

- deliver-error（用于存储错误日志）
- web-click（用于存储H5页面点击量的日志）
- server-access（用于存储服务端访问日志）
- server-app（用于存储服务器应用的日志）
- coupon（用于存储应用优惠券的日志）
- pay（用于存储支付日志）
- order（用于存储订单日志）

采集用户推广日志

商家一般通过以下方式推广新用户：

- 在注册网站时直接投放优惠券。
- 在扫描传单二维码、网页二维码或其他渠道的二维码，投放优惠券。

推广前先设置如下注册服务器地址，生成二维码（传单、网页）供用户注册扫描。用户通过扫码进行注册时，就可以得知用户是通过特定来源进入的，并记录日志。

```
http://example.com/login?source=10012&ref=kd4b
```

当服务端接受请求时，服务器输出如下日志：

```
2024-06-20 19:00:00 e41234ab342ef034,102345,5k4d,467890
```

支持通过以下方式采集用户推广日志：

- 应用程序输出日志到硬盘，通过ICAgent采集日志，详细请参考**云主机ECS文本日志接入LTS**。
- 应用程序日志通过SDK写入LTS，详细请参考**SDK概述**。

采集终端用户日志

端侧日志全面采集接入LTS，例如Web浏览器、IOS、安卓、百度小程序、微信小程序、钉钉小程序、快应用等多类端侧日志。LTS提供了多种移动端SDK，实现了缓存发送、异常重试、批量发送等稳定功能，用户快速集成即可全面采集移动端日志到LTS。

采集方案：

- 移动端：可以使用移动端iOS SDK，Android SDK接入。
- ARM设备：ARM平台可以使用Native C交叉编译。
- 商家平台设备：X86平台设备可以用SDK、ARM平台可以使用Native C交叉编译。

支持通过以下方法采集终端用户日志：

- 将日志写到本地文件，通过创建日志采集任务写到指定日志流中。
- Docker中产生的日志可以使用CCE接入LTS进行采集，详细请参考**云容器引擎CCE应用日志接入LTS**。
- C#、Python、Java、PHP、C等可以使用SDK写入LTS，详细请参考**SDK概述**。

采集服务器日志

服务器日志参考如下：以下示例日志仅供参考，请以实际日志为准。

- Syslog日志
  ```
  Aug 31 11:07:24 zhouqi-mac WeChat[9676]: setupHotkeyListenning event NSEvent: type=KeyDown loc=(0,703) time=115959.8 flags=0 win=0x0 winNum=7041 ctxt=0x0 chars="u" unmodchars="u" repeat=0 keyCode=32
  ```
- 应用程序Debug日志
  ```
  __FILE__:build/release64/sls/shennong_worker/ShardDataIndexManager.cpp
  __LEVEL__:WARNING
  __LINE__:238
  __THREAD__:31502
  offset:816103453552
  saved_cursor:1469780553885742676
  seek count:62900
  seek data redo
  log:pangu://localcluster/redo_data/41/example/2016_08_30/250_1472555483
  user_cursor:1469780553885689973
  ```
- Trace日志
  ```
  [2013-07-13 10:28:12.772518]  [DEBUG] [26064]  __TRACE_ID__:661353951201  __item__:[Class:Function]_end__  request_id:1734117  user_id:124 context:.....
  ```

支持通过以下方法采集服务器日志：

- 将日志写到本地文件，通过创建采集任务写到指定日志流中。
- Docker中产生的日志可以使用CCE接入LTS进行采集，详细请参考**云容器引擎CCE应用日志接入LTS**。
- C#、Python、Java、PHP、C等日志可以使用SDK写入LTS，详细请参考**SDK概述**。

采集不同网络环境下的日志数据

LTS在各Region提供访问点，每个Region提供三种方式接入LTS：

- 内网（经典网络）：本Region内服务访问，带宽链路质量好（推荐使用该方式）。
- 公网（经典网络）：可以被任意访问，访问速度取决于链路质量、传输安全保障建议使用HTTPS。
- 私网（专有网络VPC）：本Region内VPC网络访问。

3 日志搜索与分析

3.1 在 LTS 页面分析华为云 ELB 日志

方案概述

ELB在分发外部流量时，详细的记录HTTP(S)的访问日志，如URI请求、客户端IP和端口、状态码。

ELB日志可用于审计，也可用于通过时间和日志中的关键词信息搜索日志，同时也可以通过各种SQL聚合函数来分析某段时间内的外部请求统计数据，比如统计1天内所有URI请求404的错误条数；分析1周内的UV（用户实际单击网站次数）或PV（网站的业务访问量），掌握真实用户的网站使用频率等。

资源规划

购买并使用华为云ELB实例。

限制条件

LTS ELB日志当前仅支持七层独享型负载均衡和七层共享型负载均衡，不支持四层共享型负载均衡。

在 LTS 页面分析华为云 ELB 日志

步骤1 将ELB访问日志对接至云日志服务详细操作请参见**访问日志**。

步骤2 在系统首页左上角单击 ☰ ，选择“管理与监管 > 云日志服务 LTS”。

步骤3 在日志管理页面，单击日志流名称进入日志流详情页面，单击右上角的 ⚙ 按钮，进入设置页面，在“云端结构化解析”页签，“自动配置索引和快速分析”默认开启，选择“结构化模板”提取方式，勾选ELB系统模板，单击“保存”，关于快速分析的更多详情请参见**快速分析**。

图 3-1 配置结构化模板

步骤4 在日志流详情页面，单击“日志分析”页签，进行SQL查询与分析，如需要多样化呈现查询结果，请参考**设置云端结构化解析日志**进行配置。

- 统计1周内的PV，具体SQL查询分析语句如下所示：
  ```
  select count(*) as pv
  ```
- 统计1周内的UV，具体SQL查询分析语句如下所示：
  ```
  select count(distinct remote_port) as uv
  ```
- 统计1天所有URI返回请求2xx/3xx/4xx/5xx（返回码），了解业务的执行结果。具体的SQL查询分析语句如下所示：
  ```
  select host, router_request_uri as url, count(*) as pv,
  sum(case when status >= 200 and status < 300 then 1 else 0 end ) as "2xx times",
  sum(case when status >= 300 and status < 400 then 1 else 0 end ) as "3xx times",
  sum(case when status >= 400 and status < 500 then 1 else 0 end ) as "4xx times",
  sum(case when status >= 500 and status < 600 then 1 else 0 end ) as "5xx times"
  group by host, router_request_uri
  order by pv desc
  limit 100
  ```
  查询结果有表格、柱状图、折线图、饼图和数字图等呈现形式。更多信息请参考**统计图表**。

----结束

3.2 通过 LTS 仪表盘可视化 ELB 日志分析结果

当ELB日志接入云日志服务后，您可以通过SQL语句查询分析日志，将日志结果保存为多种图表，并将图表保存至仪表盘，从而使用仪表盘实时分析ELB日志数据。

前提条件

- 已采集ELB日志，具体操作，请参见**ELB接入**。
- 已对日志内容完成结构化配置，具体操作请参考**日志结构化概述**。

限制条件

- 一个日志流最多可创建100个图表。
- 一个仪表盘最多可创建50个图表。

- 一个仪表盘最多可添加10个过滤器。
- 一个华为账号最多可创建100个仪表盘。
- 一个华为账号最多可创建100个仪表盘模板。
- 一个华为账号最多可创建200个仪表盘分组。
- 一个华为账号最多可创建200个仪表盘模板分组。

操作流程

1. **新建可视化图表**
2. **将图表添加到仪表盘**
3. **添加过滤器**

新建可视化图表

步骤1 SQL查询与分析。

1. 登录云日志服务控制台，在左侧导航栏中，选择“日志管理”。
2. 在日志列表中，单击日志组名称前对应的 ，选择目标日志流，进入日志详情页面。
3. 选择“日志分析”，在SQL查询条件框中，选择对应时间并输入SQL语句，单击“查询”进行**日志搜索**。
   - 当设置时间范围内日志量超过10亿行时会触发迭代查询，可以通过迭代查询分多次完成全部日志的查询，界面会显示“查询状态：结果精确”。
   - 根据SQL查询返回的数据，依照您的业务需求选择不同图表类型，呈现查询结果。
   - 关于更多SQL查询的说明，请参见**SQL分析语法介绍**。

步骤2 新建图表。

1. 单击“新建”，新建可视化图表。
   或单击“保存”，将当前查询结果新建为可视化图表。
2. 在创建图表页面中，配置相关参数。
   如果开启“同时添加到仪表盘”按钮，新建图表可以直接添加到仪表盘中。
3. 完成后，单击“确定”。

步骤3 查看可视化图表。

单击“展开图表”，查看可视化图表。

----结束

将图表添加到仪表盘

将图表添加到仪表盘有两种方式：

方式一：

步骤1 鼠标悬浮可视化图表名称，单击 选择“添加到仪表盘”。

步骤2 在弹出的移动图表页面中，选择待存放的仪表盘。

步骤3 完成后，单击“确定”。

----结束

方式二：

步骤1 创建仪表盘。

1. 在左侧导航栏中，选择“仪表盘”。
2. 在仪表盘下方，选择仪表盘分组。
3. 单击“添加仪表盘”，在创建仪表盘页面，配置相关参数。
   关于仪表盘参数的说明，请参见**使用仪表盘将日志可视化**。

步骤2 将图表添加到仪表盘。

1. 在创建仪表盘页面，单击“开始添加图表”，进入添加可视化图表界面，选择目标日志新建的**可视化图表**。
2. 完成后，单击“确定”。

----结束

添加过滤器

根据设置的变量添加过滤器的操作步骤如下：

步骤1 在左侧导航栏中，选择“仪表盘”。

步骤2 在仪表盘下方，选择仪表盘分组。

步骤3 单击待操作的仪表盘名称，进入仪表盘详情页面。

步骤4 单击  ，在过滤器页面中，配置相关参数，单击“确定”。

关于过滤器参数的说明，请参见**添加过滤器**。

步骤5 调整页面布局，单击“保存设计”。

步骤6 验证过滤结果。

----结束

3.3 在 LTS 页面分析华为云 WAF 日志

方案概述

WAF（Web应用防火墙）通过对HTTP(S)请求进行检测，识别并阻断SQL注入、跨站脚本攻击、网页木马上传、命令/代码注入等攻击，所有请求流量经过WAF时，WAF会记录攻击和访问的日志，可实时决策分析、对设备进行运维管理以及业务趋势分析。

通过 LTS 分析 WAF 日志

WAF接入LTS后，支持**通过LTS分析WAF日志**。

3.4 将 LTS 日志查询页面嵌入用户自建系统

LTS支持将日志查询界面嵌入到客户自建系统。通过统一身份认证服务IAM的联邦代理机制实现用户自定义身份代理，再将登录链接嵌入至客户自建系统实现无需在华为云官网登录就可在自建系统界面查询LTS日志。

应用场景

- 该功能主要用于用户可以在自建系统免密登录LTS的场景，但是登录华为云LTS控制台还是需要账号密码。
- 用户在外部系统中（例如公司内部运维或运营系统）快速集成云日志服务的查询、分析能力。
- 无需管理众多华为子账户，方便将日志数据进行分享查看。

将 LTS 日志查询页面嵌入用户自建系统

您需要先在IAM服务为用户自定义创建身份代理并创建委托，然后再将LTS日志查询页面嵌入用户自建系统。

步骤1 使用DomainA（该账号仅供参考，请以实际账号为准）登录统一身份认证服务控制台。

步骤2 在用户组页面创建IAM用户组（用户组名以GroupC为例）并授予全局服务中的Agent Operator权限，该权限仅能切换角色至委托方账号中，访问授权的服务，具体方法请参见：**创建用户组并授权**。

步骤3 在用户页面创建IAM用户（用户名以UserB为例），并加入GroupC用户组中，具体方法请参见：**用户组添加用户**。

请确认该IAM用户支持**编程访问**和**管理控制台访问**云日志服务。如需修改IAM用户访问方式，请参考：**修改IAM用户信息**。

步骤4 在左侧导航栏选择“委托”，单击右上方的“创建委托”。

步骤5 在创建委托页面，设置委托参数。

1. “委托名称”以“iam_for_lts”为例，“委托类型”必须选择“普通账号”，“委托的账号”填写“DomainA”，“持续时间”选择“永久”，单击“下一步”。
2. 设置最小授权范围，选择“LTS ReadOnlyAccess”权限（该权限为LTS云日志服务的只读权限，只能查询LTS服务的数据，不能对LTS服务做设置修改），单击“下一步”。

   图 3-2 选择策略

3. 选择授权范围方案，勾选“指定区域项目资源”，根据需要勾选对应区域，单击“确定”。

步骤6 使用postman等工具获取X-Subject-LoginToken参数。（以下示例截图仅供参考，请以实际获取的参数为准）

1. 通过账号密码获取UserB用户的X-Subject-Token。

   接口类型：POST

   接口url：https://Endpoint/v3/auth/tokens，参数选择自定义格式，并输入如下参数，其中name从上而下依次为租户名称、用户名称、租户名称，password为用户密码。

   终端节点（Endpoint）即调用API的**请求地址**，不同服务在不同区域的终端节点不同，您可以从**地区和终端节点**中查询IAM服务的终端节点。

```
{
    "auth" : {
        "identity" : {
            "methods" : [
                "password"
            ],
            "password" : {
                "user" : {
                    "domain" : {
                        "name" : "xxxxxxx"
                    },
                    "name" : "xxxxxx",
                    "password" : "xxxxxx"
                }
            }
        },
        "scope" : {
            "domain" : {
                "name" : "xxxxxx"
            }
        }
    }
}
```

   在返回结果中获取响应头中X-Subject-Token字段。

   图 3-3 返回结果

```
General:
    Request URL: https://iam.myhuaweicloud.com/v3/auth/tokens
    Request Method: POST
    Status Code: 201

Response Headers:
    cache-control: no-cache, no-store, must-revalidate
    connection: keep-alive
    content-length: 5482
    content-type: application/json; charset=UTF-8
    date: Tue, 26 Sep 2023 07:29:37 GMT
    expires: Thu, 01 Jan 1970 00:00:00 GMT
    pragma: no-cache
    server: CloudWAF
    strict-transport-security: max-age=31536000; includeSubdomains;
    x-content-type-options: nosniff
    x-download-options: noopen
    x-frame-options: SAMEORIGIN
    x-iam-trace-id: token_cn-north-4_null_f8530fd2e48e21cc953d48988219b639
    x-request-id: f8530fd2e48e21cc953d48988219b639
    x-subject-token:
    MIIRSQYJKoZIhvcNAQcCoIIROjCCETYCAQExDTALBglghkgBZQMEAgEwgg9bBgkqhkiG9w0BBwGggg9MBIIPSHsidG9r
    a3jvRMc3TudvocOQBq+4-QIhbppckgY1M3LS7pFv0vW2rGJEPAyrK9V+tb5zBaH5RwE1rfMI99PxmsGSFhLh9EUH6WMr9
    +Zk1Y26HpaQqrrTkKOG9+PYPRw02ktSvgPaDjoeWIMiyF-5T0Ng3BT3srVfWZb3uPWjhM0Ls2r6w==
    x-xss-protection: 1; mode=block;
```

2. 根据1获取的用户X-Subject-Token获取临时访问密钥。

   在请求header添加X-Auth-Token字段，value为1中获取到的X-Subject-Token。

   图 3-4 获取临时访问密钥

Key	Value
X-Auth-Token	MIIRSQYJKoZIhvcNAQcCoIIROjCCETYCAQExDTALBglghkgBZQMEAgEwgg9bBgkqhkiG9w0BBwGggg9MBIIPSHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMjMtMDktMj

   接口类型：POST

接口url：https://Endpoint/v3.0/OS-CREDENTIAL/securitytokens，参数选择自定义格式，并输入如下参数，其中具体参数含义分别为：agency_name为委托名称、domain_name为租户名称、duration_seconds为token过期时间（单位秒）、name为用户名。

```
{
    "auth" : {
        "identity" : {
            "methods" : [
                "assume_role"
            ],
            "assume_role" : {
                "agency_name" : "iam_for_lts",
                "domain_name" : "xxxxxx",
                "duration_seconds" : 86400,
                "session_user" : {
                    "name" : "xxxxxx"
                }
            }
        }
    }
}
```

在返回结果中获取响应体中获取临时访问密钥。

图 3-5 获取临时访问密钥

3. 根据2获取的临时访问密钥获取登录X-Subject-LoginToken。

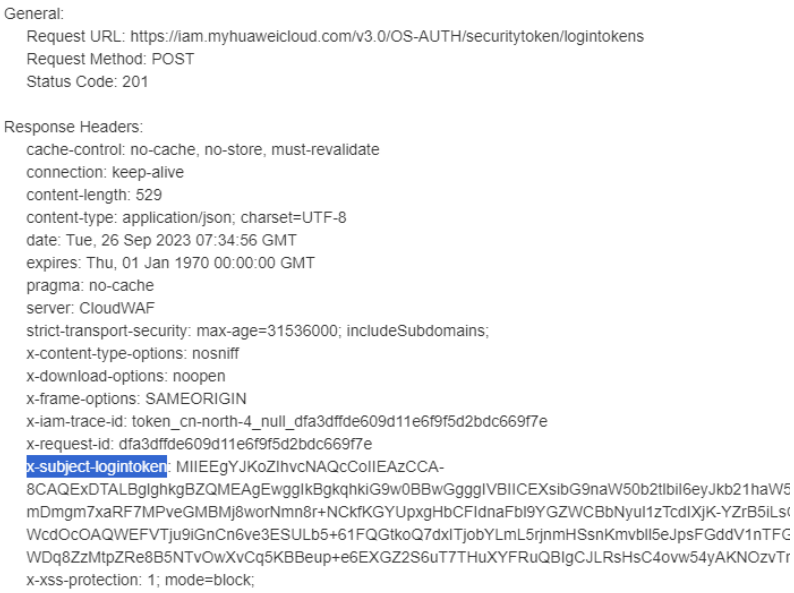
   接口类型：POST

   接口url：https://Endpoint/v3.0/OS-AUTH/securitytoken/logintokens，参数选择自定义格式，并输入如下参数，其中access、secret、id对应的值分别为2返回的access、secret、securitytoken，duration_seconds为token过期时间，单位为秒。

   ```
   {
       "auth" : {
           "securitytoken" : {
               "access" : "xxxxxx",
               "secret" : "xxxxxx",
               "id" : "xxxxxx",
               "duration_seconds" : 43200
           }
       }
   }
   ```

   在返回结果中获取响应头中的X-Subject-LoginToken字段。

图 3-6 获取 X-Subject-LoginToken



步骤7 根据3获取的X-Subject-LoginToken构建代理URL，完成免密登录。

代理登录地址的构建规则为：

https://auth.huaweicloud.com/authui/federation/login?service={target_console_url}&logintoken={logintoken}&idp_login_url={enterprise_system_loginURL}

表 3-1 URL 参数说明

参数名称	说明
{target_console_url}	云日志服务地址说明的urlencode编码结果。详细请参考**云日志服务地址说明**。
{logintoken}	为3获取的X-Subject-LoginToken的urlencode编码结果。
{enterprise_system_loginURL}	选填参数，为客户的页面地址的urlencode编码结果，当loginToken验证失效时会跳转到该页面。

- 以上三个参数都需要进行urlencode编码，否则可能导致免密登录失败。
- urlencode编码操作方式：打开浏览器按F12进入开发者模式，选择console（控制台），输入“encodeURIComponent("*")”，*为需要编码的信息，单击“Enter”,查看返回的urlencode编码值。

{target_console_url}的值为LTS前台服务URL地址的urlencode编码，编码前URL如下，具体参考**表3-2**。

https://console-intl.huaweicloud.com/lts/?region={regionId}&cfModuleHide=header_sidebar_floatlayer#/lts/logEventsLeftMenu/events?groupId={groupId}&topicId={topicId}&epsId={epsId}&condition={condition}

表 3-2 参数说明

参数名称	说明
{regionId}	区域ID，登录console页面后，在浏览器地址栏中获取。
{groupId}	日志组ID。
{topicId}	日志流ID。
{epsId}	日志流所属的企业项目ID，若无企业项目，值为0。
{condition}	日志查询条件，如name:a and age:12 and addr:xx。 ● 非必填 ● 单个关键词形式为key:value ● 多个关键词用 and 隔开 ● 关键词中不能包含英文分号（;）、英文冒号（:） ● 关键词中含有其他特殊字符如（+、=、？ 、#、%、& ）需转换为十六进制，即%加字符的ASCII码（ %2B、%3D、%3F、%23、%25、%26 ）

步骤8 完成以上步骤，即可实现在自建系统免密登录云日志服务LTS。

使用如下iframe嵌套，其中src的值为7中获得的代理URL。

iframe内嵌功能依赖浏览器允许第三方Cookie，具体设置方法请以实际使用的浏览器为准，例如chrome浏览器参考如下方法：设置->隐私和安全->第三方Cookie->允许第三方Cookie。

```
<body>
    <iframe src="target_url" width="100%" height="96%" id="ltsIframePage"></iframe>
</body>
```

----结束

云日志服务地址说明

1. 云日志服务首页，基础URL为：

   ```
   https://console-intl.huaweicloud.com/lts/?
   region={regionId}&cfModuleHide=header_sidebar_floatlayer_rightsidebar#/cts/manager/groups
   ```

   表 3-3 参数说明表

参数名称	必填	类型	描述
regionId	是	String	区域ID，登录console页面后在浏览器的地址栏中获取。

2. 日志搜索界面，基础URL为：

   ```
   https://console-intl.huaweicloud.com/lts/?
   region={regionId}&cfModuleHide=header_sidebar_floatlayer_rightsidebar#/cts/logEventsLeftMenu/
   events?
   groupId={groupId}&topicId={topicId}&epsId={epsId}&hideHeader={hideHeader}&fastAnalysisCollapsed
   ={fastAnalysisCollapsed}&hideDashboard={hideDashboard}&hideFeedback={hideFeedback}&isFoldLabe
   l={isFoldLabel}&hideStreamName={hideStreamName}&showK8sFilter={showK8sFilter}&clusterId={clus
   terId}&hideBarChart={hideBarChart}&hideTabs={hideTabs}&condition={condition}
   ```


表 3-4 参数说明表

参数名称	必填	类型	默认值	描述
regionId	是	String	无	区域ID，登录console页面后在浏览器的地址栏中获取。
groupId	是	String	无	日志组ID。
topicId	是	String	无	日志流ID。
epsId	否	String	无	日志流所属的企业项目ID，若无企业项目，值为0。
hideHeader	否	Boolean	false	是否隐藏左侧列表及顶部横向日志流列表，如需隐藏，该参数值为true。 使用iframe内嵌场景才会生效，不使用iframe内嵌场景则不生效。
fastAnalysisCollapsed	否	Boolean	false	是否收起快速分析，如需默认收起，该参数值为true。
hideDashboard	否	Boolean	false	是否隐藏创建仪表盘图标，如需隐藏，该参数值为true。
hideFeedback	否	Boolean	false	是否隐藏评价按钮，如需隐藏，该参数值为true。
isFoldLabel	否	Boolean	true	控制日志表格中的label字段是否换行，如需换行展示，该参数值为true。
hideStreamName	否	Boolean	false	是否隐藏日志流名称，如需隐藏，该参数值为true。
showK8sFilter	否	Boolean	false	是否展示容器日志筛选，容器日志搜索场景下，可选择该参数为true，控制是否展示容器日志筛选条件。
clusterId	否	String	无	集群ID，showK8sFilter参数为true时，该参数必填。
hideBarChart	否	Boolean	false	是否默认收起日志条数统计图，如需默认收起，该参数值为true。
hideTabs	否	Boolean	false	是否隐藏“日志搜索、日志分析、实时日志”标签tabs，默认不隐藏。如需隐藏，该参数值为true。
hideShare	否	Boolean	false	是否隐藏“分享”按钮，默认不隐藏。如需隐藏，该参数值为true。（当前仅华北-北京四局点支持该参数）
keepOnline	否	Boolean	false	是否保持登录状态。如需一直保持登录状态，不退出登录，该参数值为true。

参数名称	必填	类型	默认值	描述
condition	否	String	无	日志查询条件，如name:a and age:12 and addr:xx。 ● 非必填 ● 单个关键词形式为key:value ● 多个关键词用 and 隔开 ● 关键词中不能包含英文分号（;）、英文冒号（:） ● 关键词中含有其他特殊字符如（+、=、？ 、#、%、&）需转换为十六进

3. 可视化日志搜索界面，基础URL为：

```
https://console-intl.huaweicloud.com/lts/?region={regionId}&cfModuleHide=header_sidebar_floatlayer_rightsidebar#/cts/logEventsLeftMenu/events?visualization=true&groupId={groupId}&topicId={topicId}&epsId={epsId}&sql={sql}
```

表 3-5 参数说明表

参数名称	必填	类型	默认值	描述
regionId	是	String	无	区域ID，登录console页面后在浏览器的地址栏中获取。
groupId	是	String	无	日志组ID。
topicId	是	String	无	日志流ID。
epsId	否	String	无	日志流所属的企业项目ID，若无企业项目，值为0。
hideHeader	否	Boolean	false	是否隐藏左侧列表及顶部横向日志流列表，如需隐藏，该参数值为true。
sql	否	String	无	SQL查询语句，如SELECT count (*)。

4. 仪表盘界面，基础URL为：

```
https://console-intl.huaweicloud.com/lts/?region={regionId}&cfModuleHide=header_sidebar_floatlayer_rightsidebar#/cts/manager/dashboard?dashboardId={dashboardId}&hideDashboardList={hideDashboardList}&showCurrentdashboardGroup={showCurrentdashboardGroup}&streamId={streamId}&streamDisabled={streamDisabled}&readonly={readonly}&filter=key1:value1,value2;key2:value3,value4&autoFresh={autoFresh}
```

表 3-6 参数说明表

参数名称	必填	类型	默认值	描述	示例
regionId	是	String	无	区域ID，登录console页面后在浏览器的地址栏中获取。	region=xx-xx-xx
dashboardId	否	String	无	需要展示的仪表盘ID，默认值为""。 使用场景：当用户需要默认展示某仪表盘时，可添加此参数。	dashboardId=xxxxxxxx
hideDashboardList	否	Boolean	false	是否隐藏仪表盘选择下拉列表：默认不隐藏，true表示隐藏。 **使用场景：**当用户需要隐藏仪表盘下拉列表时，可通过添加该参数且值为true来实现。	hideDashboardList=true
showCurrentdashboardGroup	否	Boolean	false	是否只展示当前仪表盘所在分组/模板：默认值为false。 **使用场景：**当用户只需要展示当前仪表盘分组/模板的仪表盘时，可通过添加该参数且值为true来实现。 注意：若hideDashboardList参数值为true时，当前参数无效。	showCurrentdashboardGroup=true
streamId	否	String	无	日志流ID：默认值为""。 **使用场景：**只适用于仪表盘模板。当用户需要默认选中某日志流时，可添加该参数。	streamId=xxxxxx
streamDisabled	否	Boolean	false	日志流下拉框：默认可选择，true标识不可选择 **使用场景：**只适用于仪表盘模板。当用户需要置灰日志流下拉框时，可添加该参数。	streamDisabled=true

参数名称	必填	类型	默认值	描述	示例
filter	否	String	无	过滤器参数，值为要选中的过滤器的名称及选中项。 key1、key2为过滤器名称，value1、value2为过滤器key1需要选中的值，value3、value4为过滤器key2需要选中的值。多个过滤器按照;分隔，多个选中项按照,分隔。 **使用场景：**当用户内嵌仪表盘界面需要默认选中某些过滤器的key、value时，可添加该参数。	filter=key1:value1,value2;key2:value3,value4
readonly	否	Boolean	false	是否是只读场景，只读场景下，操作类按钮会被隐藏。例如：新建过滤器、添加/修改/删除仪表盘等。 **使用场景：**当用户只需要展示仪表盘，不需要操作权限时，可添加该参数。	readonly=true
autoFresh	否	String	无	定时刷新时长，默认值为""。 **使用场景：**当用户需要指定默认定时刷新时长时，可添加此参数，当前支持的定时刷新时长参数取值为：0m，1m，5m，15m之一，对应：不定时刷新、1分钟定时刷新、5分钟定时刷新、15分钟定时刷新。	autoFresh=1m

4 日志转储

4.1 批量修改 LTS 日志文件转储时区

您在LTS上经常会执行日志接入、日志告警、日志转储等配置操作。有些操作是需要重复多次配置，但目前LTS还没有提供控制台批量操作功能，这时您可以通过Python脚本结合LTS API接口实现自定义的批量操作。

使用场景

当用户创建了1000条日志转储的OBS规则，但转储时文件时区全部选择(UTC) 协调世界时间，现在需要根据实际情况修改为(UTC+08:00) 北京。目前LTS控制台没有提供批量修改功能，如果手动修改每条转储规则，会导致人工耗时非常长。

前提条件

1. Linux系统的主机。
2. 查询API相关接口文档。
   - 通过查询日志转储API获取到所有转储任务的信息。
   - 通过更新日志转储API将转储任务配置的时区修改。
3. 在API Explorer中测试API功能，API Explorer提供API检索及平台调试能力。
4. 参考API Explorer示例代码，在主机上安装Python SDK。
   - Python的**SDK依赖包地址**以及**SDK使用说明**。
     ```
     pip install huaweicloudsdklts
     ```
   - API Explore提供Python调用API的示例代码，以下示例仅供参考：
     ```
     # coding: utf-8
     import os
     from huaweicloudsdkcore.auth.credentials import BasicCredentials
     from huaweicloudsdklts.v2.region.lts_region import LtsRegion
     from huaweicloudsdkcore.exceptions import exceptions
     from huaweicloudsdklts.v2 import *
     if __name__ == "__main__":
         # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
         # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
     /* 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量
     ```



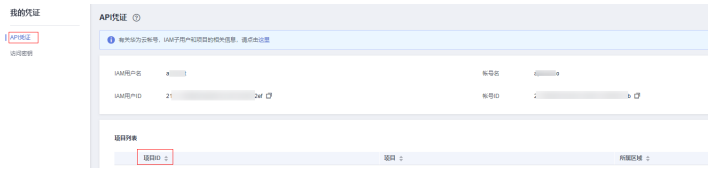
```
中密文存放, 使用时解密, 确保安全;
    ak = os.environ["CLOUD_SDK_AK"]
    sk = os.environ["CLOUD_SDK_SK"]
    credentials = BasicCredentials(ak, sk)
    client = LtsClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(LtsRegion.value_of("xx")) \
        .build()
    try:
        request = ListTransfersRequest()
        response = client.list_transfers(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

批量修改 LTS 日志文件转储时区

步骤1 获取参数，并将实际值替换到代码中。

- **如何获取访问密钥AK/SK**
- 获取项目ID（ project id ），详细步骤请参考**API凭证**。

图 4-1 获取 Project_ID



- 请参考**地区和终端节点**获取Region&iam_Endpoint。

表 4-1 Endpoint 表

区域名称	区域	终端节点（Endpoint）	协议类型
亚太-曼谷	ap-southeast-2	lts.ap-southeast-2.myhuaweicloud.com	HTTPS
亚太-新加坡	ap-southeast-3	lts.ap-southeast-3.myhuaweicloud.com	HTTPS
中国-香港	ap-southeast-1	lts.ap-southeast-1.myhuaweicloud.com	HTTPS

- 获取时区和时区ID。

表 4-2 常用时区表

时区	时区ID
UTC-12:00	Etc/GMT+12
UTC-11:00	Etc/GMT+11
UTC-10:00	Pacific/Honolulu
UTC-09:00	America/Anchorage
UTC-08:00	America/Santa_Isabel
UTC-07:00	America/Chihuahua
UTC-06:00	America/Chicago
UTC-05:00	America/New_York
UTC-04:00	America/Santiago
UTC-03:00	America/Sao_Paulo
UTC-02:00	Etc/GMT+2
UTC-01:00	Atlantic/Azoresjavik
UTC+00:00	Europe/London
UTC+01:00	Europe/Parist
UTC+02:00	Europe/Istanbul
UTC+03:00	Europe/Minsk
UTC+04:00	Europe/Moscow
UTC+05:00	Asia/Tashkent
UTC+06:00	Asia/Almaty
UTC+07:00	Asia/Bangkok
UTC+08:00	Asia/Shanghai
UTC+09:00	Asia/Tokyo
UTC+10:00	Asia/Yakutsk
UTC+11:00	Asia/Vladivostok
UTC+12:00	Pacific/Fiji
UTC+13:00	Pacific/Apia

步骤2 在主机上执行如下命令确认是否已安装huaweicloudsdkcore和huaweicloudsdklts包。

```
pip list | grep huaweicloudsdk
```

- 如果已安装，执行结果会返回huaweicloudsdk相关信息。如果没有安装，则不会返回任何内容。

- 若未安装在主机上请执行如下操作进行安装：
  ```
  pip install huaweicloudsdkcore huaweicloudsdklts
  ```

步骤3 在主机上执行“vi lts_python.py”新建lts_python.py文件，将如下代码复制到该文件中，用于实现批量修改OBS转储文件时区。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdklts.v2 import *
from huaweicloudsdklts.v2.region.lts_region import LtsRegion
/* 认证用的ak和sk硬编码到代码中或者明文存储都有很大的安全风险, 建议在配置文件或者环境变量中密文存放, 使用时解密, 确保安全;
if __name__ == "__main__":
AK = "your ak"
SK = "your sk"
PROJECT_ID = "your project id"
REGION = "your region"
IAM_ENDPOINT = "iam_endpoint"

OBS_TIME_ZONE = "the time_zone you want to change"
OBS_TIME_ZONE_ID = "time_zone_id"

credentials = BasicCredentials(AK, SK, PROJECT_ID).with_iam_endpoint(IAM_ENDPOINT)

client = LtsClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(LtsRegion.value_of(REGION)) \
    .build()

# 1.get obs transfer task
try:
    request = ListTransfersRequest()
    request.log_transfer_type = "OBS"
    response = client.list_transfers(request)
    obs_transfer_num = len(response.log_transfers)
    task_list = response.log_transfers
    print("#### get {} obs transfer task ####".format(obs_transfer_num))
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)

# 2.set obs transfer task obs_time_zone to UTC+08:00
CNT = 1
while len(task_list):
    transfer_task = task_list.pop()
    print("There are still {} progress: \n".format(len(task_list)), transfer_task)
    try:
        if transfer_task.log_transfer_info.log_transfer_detail.obs_time_zone == OBS_TIME_ZONE:
            CNT += 1
            continue
        request = UpdateTransferRequest()
        transfer_task.log_transfer_info.log_transfer_detail.obs_time_zone = OBS_TIME_ZONE
        transfer_task.log_transfer_info.log_transfer_detail.obs_time_zone_id = OBS_TIME_ZONE_ID
        request.body = UpdateTransferRequestBody(
            log_transfer_info=transfer_task.log_transfer_info,
            log_transfer_id=transfer_task.log_transfer_id
        )
        response = client.update_transfer(request)
        CNT += 1
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
        task_list.append(transfer_task)
    except exceptions.ServerResponseException as e:
```



```
        print({
            "target": transfer_task.log_streams,
            "reason": e
        })
        task_list.append(transfer_task)
```

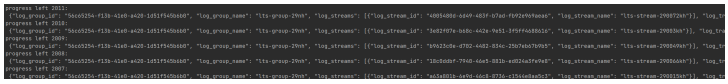
步骤4 在主机上执行Python脚本，批量修改OBS转储文件时区。

```
nohup python lts_python.py > lts_python.log &
```

步骤5 查看执行日志确认Python脚本运行成功，文件时区已修改。

```
tail -f lts_python.log
```

图 4-2 查看执行日志



----结束

5 日志计费

5.1 通过日志流标签统计不同部门在 LTS 的费用开销

为了统计企业内部不同部门在LTS的费用开销情况，您可以在LTS的日志流上添加标签用于识别不同的业务部门，LTS在上传话单给费用中心时会带上这些标签信息。您可以在“费用 > 费用账单 > 消费详情”下载LTS的明细账单信息，然后基于资源标签来统计不同部门的费用开销，为企业内部的费用分摊提供依据。

前提条件

按日志流维度上报话单功能仅支持白名单用户，若您需要使用日志流标签统计不同部门在LTS的费用开销，请**提交工单**申请开通。

方案介绍

日志流是通过日志组管理的，给日志组添加标签时，默认开启应用到日志流，这样日志流就自动添加标签。即可通过日志流统计不同部门在LTS的费用开销。

本实践以aa和bb部门为例子，首先在aa部门的日志组添加group=groupaa标签，bb部门的日志组添加group=groupbb标签，然后在费用明细中导出账单，通过Excel进行统计分析。

以下提到的价格仅为示例，实际计算请以**价格计算器**中的价格为准。

通过日志流标签统计不同部门在 LTS 的费用开销

步骤1 登录云日志服务控制台。

步骤2 在“日志管理”页面，将鼠标悬浮在aa部门创建的日志组“标签”列单击  。

步骤3 在弹出的编辑标签页面，单击“添加标签”，填写aa部门的标签键group和标签值groupaa，默认开启“应用到日志流”，将日志组标签同步到日志流，单击“确定”。

图 5-1 aa 部门添加标签

步骤4 将鼠标悬浮在bb部门创建的日志组“标签”列单击 。

步骤5 在弹出的编辑标签页面，单击“添加标签”，填写bb部门的标签键group和标签值groupbb，默认开启“应用到日志流”，将日志组标签同步到日志流，单击“确定”。

图 5-2 bb 部门添加标签

步骤6 标签添加成功后预计需要等待一个小时才能生成话单。

步骤7 在控制台顶部菜单栏中选择“费用 > 费用账单”，进入“消费汇总”页面。

步骤8 左侧导航栏选择“消费详情”，统计维度选择“按使用量”，统计周期选择“明细”，在筛选条件中选择“产品类型：云日志服务”。详细操作请参考**账单详情**。

步骤9 单击“导出”，在导出页面自定义设置导出范围，将费用明细导出到本地查看，详细操作请参考**账单导出**。

步骤10 费用明细导出的Excel文件中，筛选“资源标签”，过滤标签名称，即可更直观的查看到aa和bb部门的消费明细详情。

实际计算请以**价格计算器**中的价格为准。

----结束