

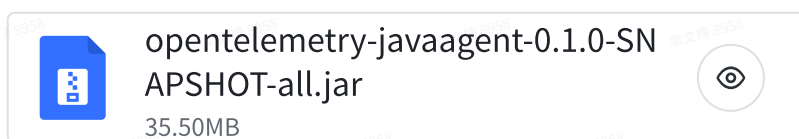
Hera接入文档

一、应用部署的依赖

接入Hera的应用，需要依赖探针、jvm参数、环境变量以及log-agent服务

1、探针

探针在构建完成后最终会得到一个opentelemetry相关的jar，我们需要在部署项目时，将jar文件放到服务器上。这是为了在jvm参数中，能够通过-javaagent使用到探针。我们基于开源版本默认打包了一版探针，可以直接使用：



2、jvm参数

`-javaagent:/opt/soft/opentelemetry-javaagent-all-0.0.1.jar`：用于表示javaagent探针jar包在服务器上的位置，我们一般习惯将探针的jar文件更名为 `opentelemetry-javaagent-all-0.0.1.jar`，并放在服务器 `/opt/soft` 目录下。

`-Dotel.resource.attributes=service.name=1-test`：用于表示当前服务的应用名。应用名应当与名为 `mi-one.app.name` 的环境变量保持一致

`-Dotel.traces.exporter=log4j2`：用于表示trace export方式，是通过log4j2将trace输出到日志文件中

`-Dotel.metrics.exporter=prometheus`：用于表示metrics export方式

`-Dotel.javaagent.debug=false`：用于表示是否开启debug日志，一般线上服务不建议开启

`-Dotel.exporter.prometheus.nacos.addr=${nacosurl}`：用于表示nacos地址。需要将nacos地址端口进行配置，例如：127.0.0.1:80

`-Dotel.javaagent.exclude-classes=com.dianping.cat.*`：过滤不被探针拦截的包。如果使用到了cat，需要将cat所在的目录进行过滤

`-Dotel.exporter.log.isasync=true`：用于表示是否开log4j2启异步日志，一般出于性能考虑，会是true

`-Dotel.exporter.log.pathprefix=/home/work/log/`：用于表示log4j2的日志位置。这里log4j2的日志会优先输出到名为 `MI-ONE_LOG_PATH` 的环境变量所表示的位置，如果没有这个环境变量，则会输出到 `-Dotel.exporter.log.pathprefix \ -Dotel.resource.attributes=service.name` 目录下

`-Dotel.propagators=tracecontext`：用于表示trace传输的处理类型，目前只用到了tracecontext

3、环境变量

`host.ip`：用于记录当前物理机IP，展示在trace的process.tags里。在k8s里获取的是pod的IP

`node.ip`：用于记录k8s当前node节点的IP

`MIONE_LOG_PATH`：用于将trace span信息存放在\${MIONE_LOG_PATH}/trace/trace.log

`mione.app.name`：用于记录应用id与应用名称，格式是appId-appName。eg: 1-test, 1是appId, test是appName。如果为空，程序里默认使用none。应用是Hera中非常重要的元数据，可观测数据展示都与应用有关。

`TESLA_HOST`：同host.ip。用于注册nacos、jvm metrics里的serverIp标签。

`JAVAAGENT_PROMETHEUS_PORT`：当前物理机可用端口号，用于提供给Prometheus拉取jvm metrics的httpServer使用。如果为空，程序里默认使用55433。

`hera.buildin.k8s`：用于记录是否是k8s部署的服务，如果是k8s的服务，标记为1。如果非k8s部署，可不填

`MIONE_PROJECT_ENV_NAME`：当前部署环境的名称，eg: dev、uat、st、preview、production。如不填，默认使用default

`MIONE_PROJECT_ENV_ID`：当前部署环境的ID。环境id与环境名称，在Hera的指标监控中可以根据不同环境来看监控图表

`APPLICATION`：是将 `mione.app.name` 的值所有中划线，转为下划线，用于容器监控的扩展指标

`serverEnv`：与 `MIONE_PROJECT_ENV_NAME` 的值相同，用于容器监控的扩展指标

4、log-agent

目前Hera的trace、log都使用log-agent来收集，log-agent需要以sidecar的方式与应用部署在同一容器中，并且需要被采集的日志文件目录与trace日志目录同时挂载出来，以能够被log-agent采集。

5、应用Dockerfile与K8S yaml文件示例

下面是我们demo项目所使用的dockerfile与k8s的yaml文件，可以参考一下hera-demo-client应用中关于探针、jvm、环境变量与log-agent的配置

Dockerfile

```
1 FROM openjdk:8-jre
2
3 COPY ./hera-demo-client-1.0.0-SNAPSHOT.jar /home/work/hera-demo-client-1.0.0-SNAPS
HOT.jar
```

```
4 COPY ./opentelemetry-javaagent-0.1.0-SNAPSHOT-all.jar /opt/soft/opentelemetry-java
agent-0.1.0-SNAPSHOT-all.jar
5
6 ENTRYPOINT ["java","-javaagent:/opt/soft/opentelemetry-javaagent-0.1.0-SNAPSHOT-a
l.jar","-Dotel.resource.attributes=service.name=2-hera-demo-client","-Dotel.trace
s.exporter=log4j2","-Dotel.metrics.exporter=prometheus","-Dotel.javaagent.debug=fa
lse","-Dotel.exporter.prometheus.nacos.addr=nacos:80","-Dotel.javaagent.exclude-cl
asses=com.dianping.cat.*","-Dotel.exporter.log.isasync=true","-Dotel.exporter.log.
pathprefix=/home/work/log/","-Dotel.propagators=tracecontext","-Xloggc:/home/work/
log/gc.log","-Duser.timezone=Asia/Shanghai","-XX:+HeapDumpOnOutOfMemoryError","-X
X:HeapDumpPath=/home/dum/oom.dump","-jar","/home/work/hera-demo-client-1.0.0-SNAPS
HOT.jar","&&","tail","-f","/dev/null"]
```

K8S yaml

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: hera-demo-client
5   namespace: demo
6   labels:
7     app: hera-demo-client
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: hera-demo-client
13  template:
14    metadata:
15      labels:
16        app: hera-demo-client
17    spec:
18      containers:
19        - name: hera-demo-client-container
20          image: demo/hera-demo-client:v1
21          env:
22            - name: host.ip
23              valueFrom:
24                fieldRef:
25                  apiVersion: v1
26                  fieldPath: status.podIP
27            - name: node.ip
28              valueFrom:
29                fieldRef:
30                  apiVersion: v1
```

```

31         fieldPath: status.hostIP
32     - name: MIONE_LOG_PATH
33       value: /home/work/log/hera-demo-client
34     - name: mione.app.name
35       value: 2-hera-demo-client
36     - name: TESLA_HOST
37       valueFrom:
38         fieldRef:
39           apiVersion: v1
40           fieldPath: status.podIP
41     - name: hera.buildin.k8s
42       value: '1'
43     - name: MIONE_PROJECT_ENV_NAME
44       value: dev
45     - name: JAVAAGENT_PROMETHEUS_PORT
46       value: '55433'
47     - name: MIONE_PROJECT_ENV_ID
48       value: '2'
49     - name: APPLICATION
50       value: 2_hera_demo_client
51     - name: serverEnv
52       value: dev
53   volumeMounts:
54     - name: log-path
55       mountPath: /home/work/log
56   resources:
57     limits:
58       cpu: '500m'
59       memory: 2Gi
60
61   - name: log-agent
62     image: demo/log-agent:v1
63     volumeMounts:
64     - mountPath: "/home/work/log"
65       name: log-path
66     resources:
67       limits:
68         cpu: '300m'
69         memory: 1Gi
70   volumes:
71   - name: log-path
72     emptyDir: {}

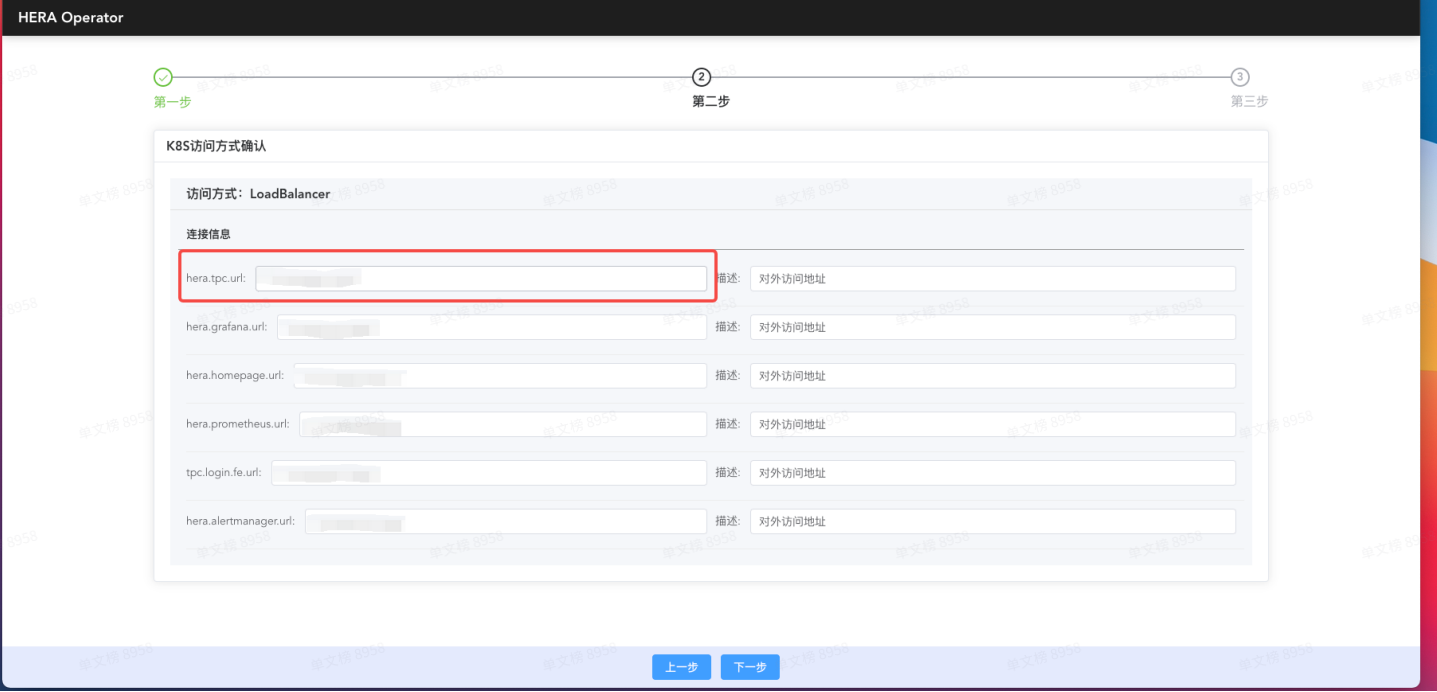
```

二、录入TPC

Hera通过TPC系统来录入、同步、管理应用的元数据信息，以及应用与人的权限关系

1、注册并登录tpc

我们需要访问通过hera-operator生成的tpc的页面链接

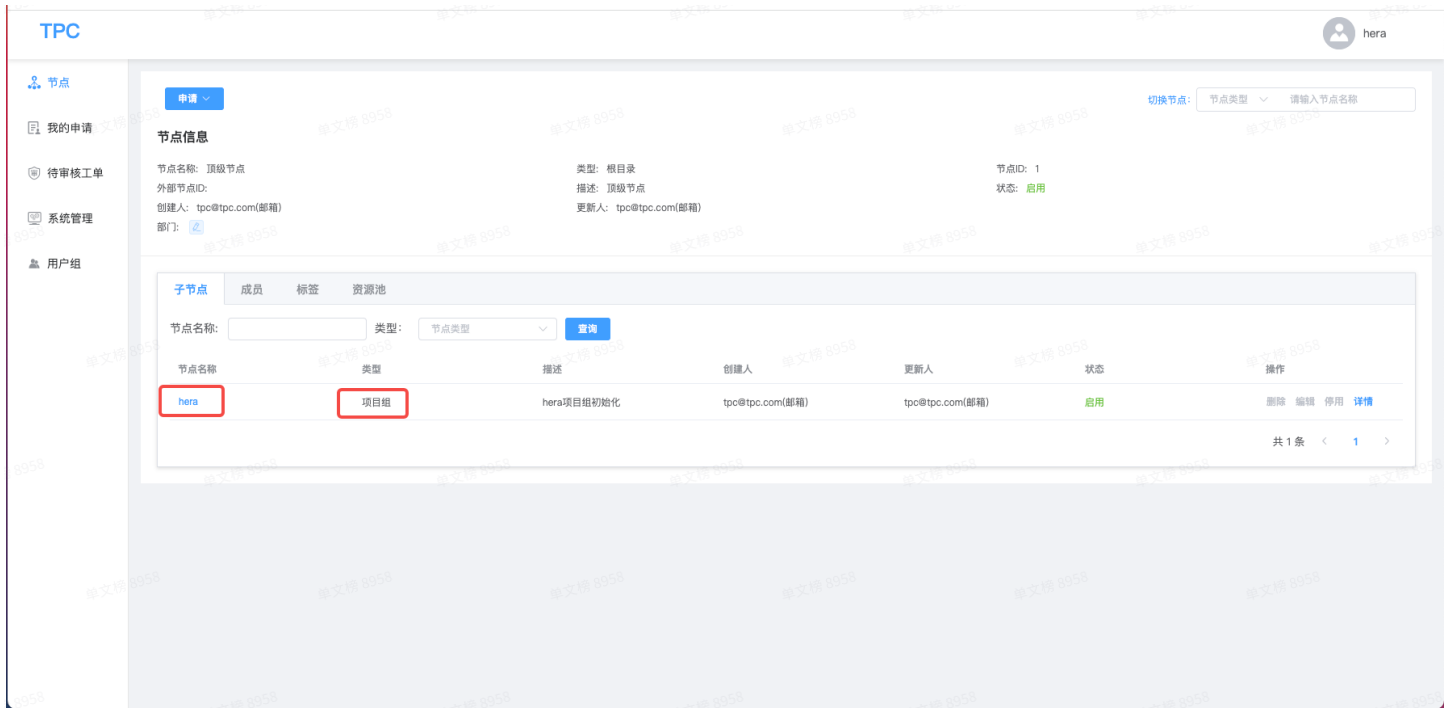


未登录访问时，会跳转到tpc登录页面，如果是第一次访问，我们需要先注册tpc超级管理员账号，该账号用于创建项目、授权等操作。超级管理员账号名必须为：tpc@tpc.com，密码可以自定义。注册完成后，我们使用tpc@tpc.com进行登录。

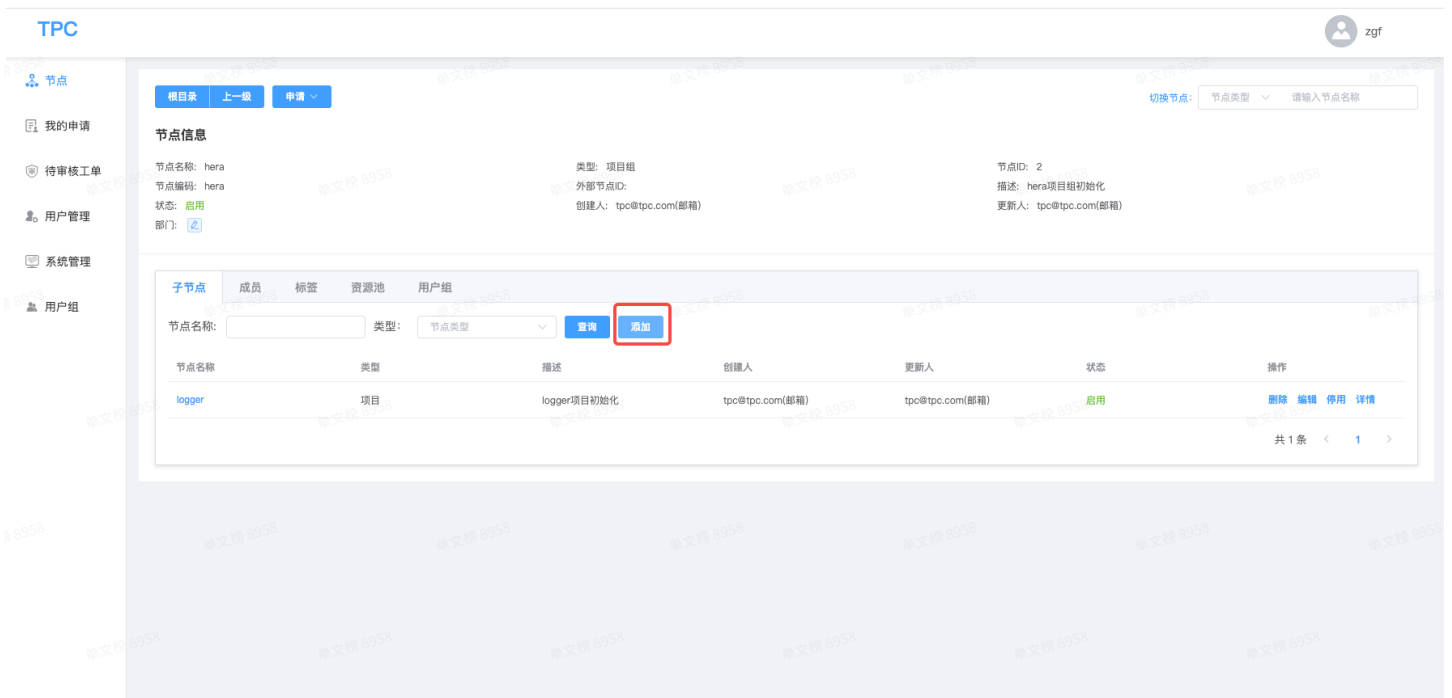


2、录入应用

登录成功后，就进入了tpc的首页。在tpc首页的节点信息--子节点中，默认会创建一个名为hera的节点，这个节点是项目组，接入Hera的应用需要先在这个项目组下维护应用信息。



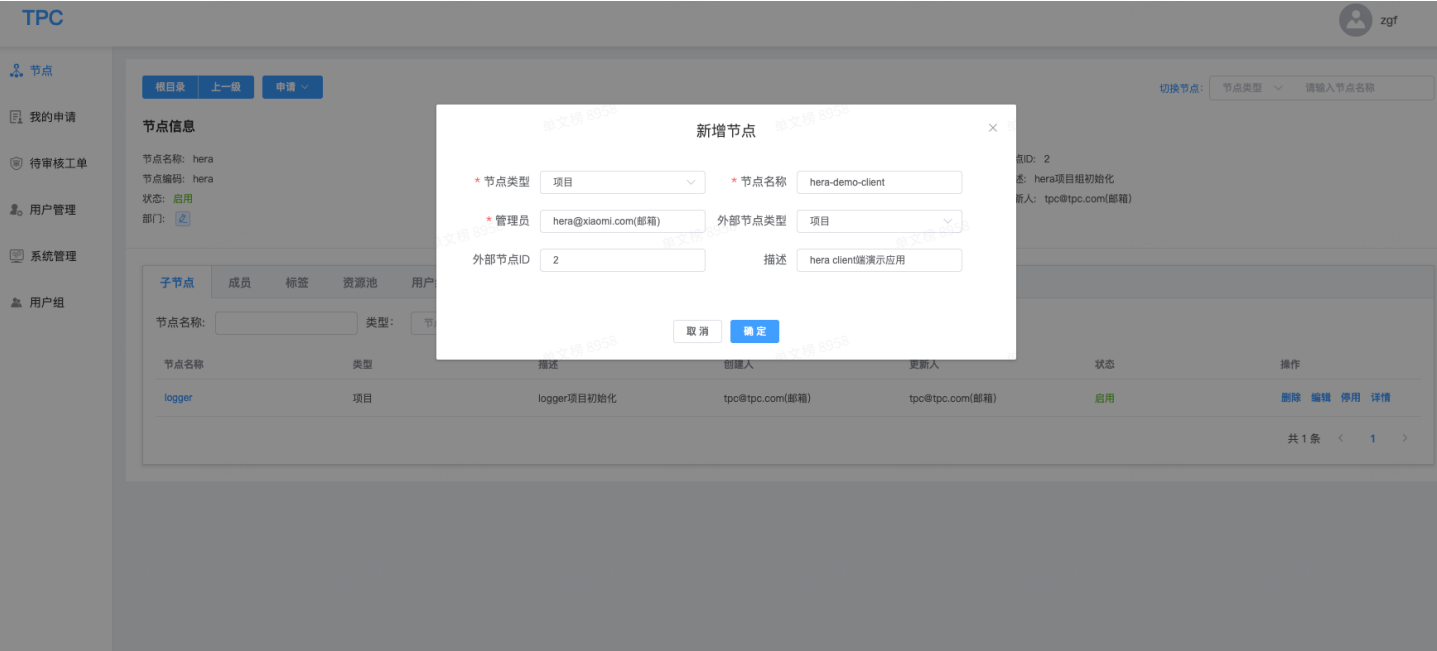
点击hera节点，进入hera项目组中。hera项目组中默认会创建logger项目，这个项目是用于Hera日志中，我们可以忽略。点击图中标注的“添加”按钮，会弹出创建项目的弹框。



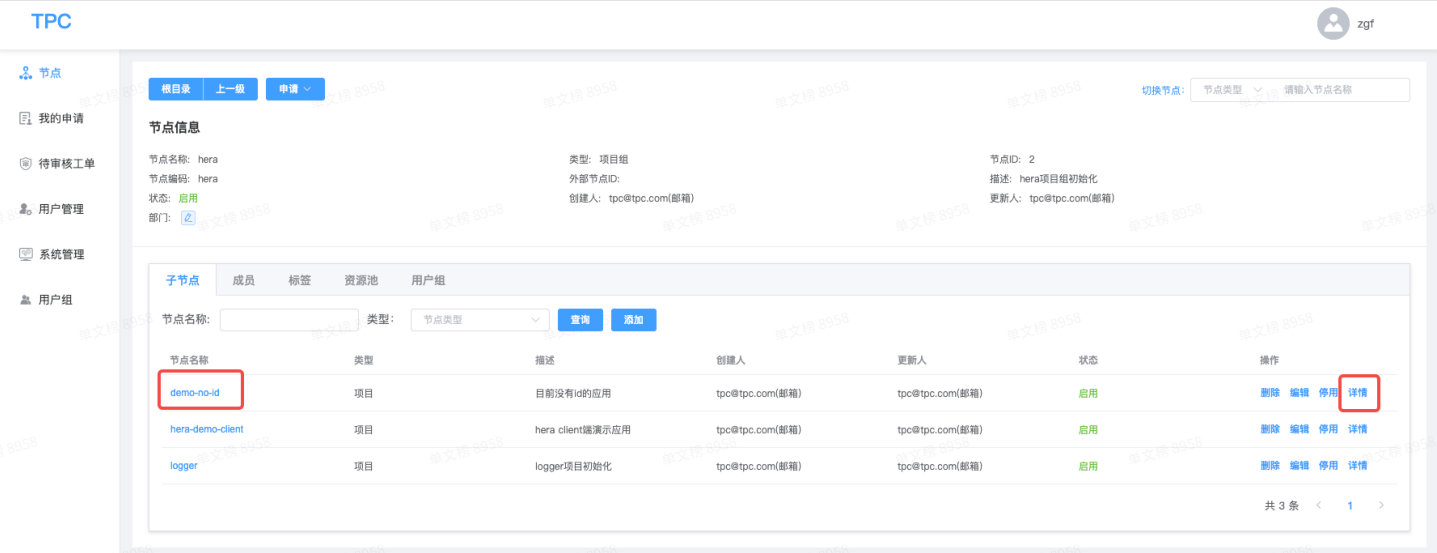
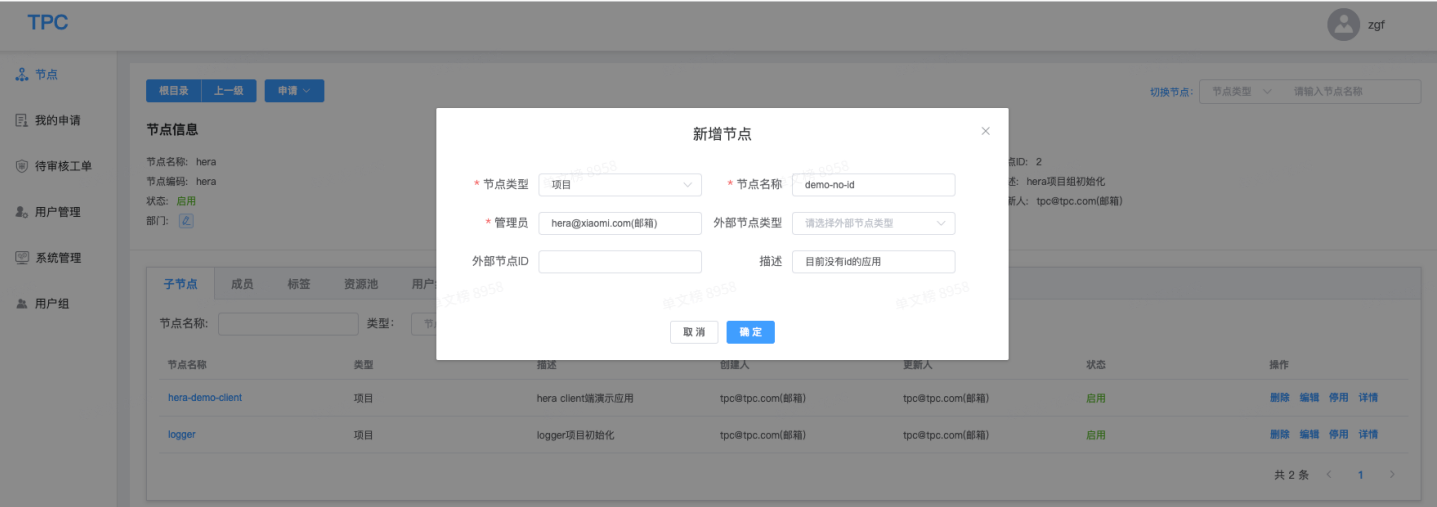
节点类型选择“项目”，节点名称就填写我们的应用名称，管理员就填写当前应用的成员账号，当然，这个账号需要提前在tpc上注册。

我们录入应用信息时，需要分两种情况：

一是，我们目前已经有了应用的数据，并且应用的数据中有id，此时我们可以通过在tpc维护外部节点id，来将我们已有的应用id录入。这里我使用之前“一、应用部署的依赖”中的“应用Dockerfile与K8s yaml文件示例”中的应用来进行录入，如下图所示：



二是，我们目前只有应用的名称，但是没有应用的id，此时我们不需要填写“外部节点类型”与“外部节点ID”，tpc在创建应用之后，会自动生成该应用的id，点击该应用节点详情按钮，在该应用“节点信息”中，我们可以看到该应用的id。如下图所示：

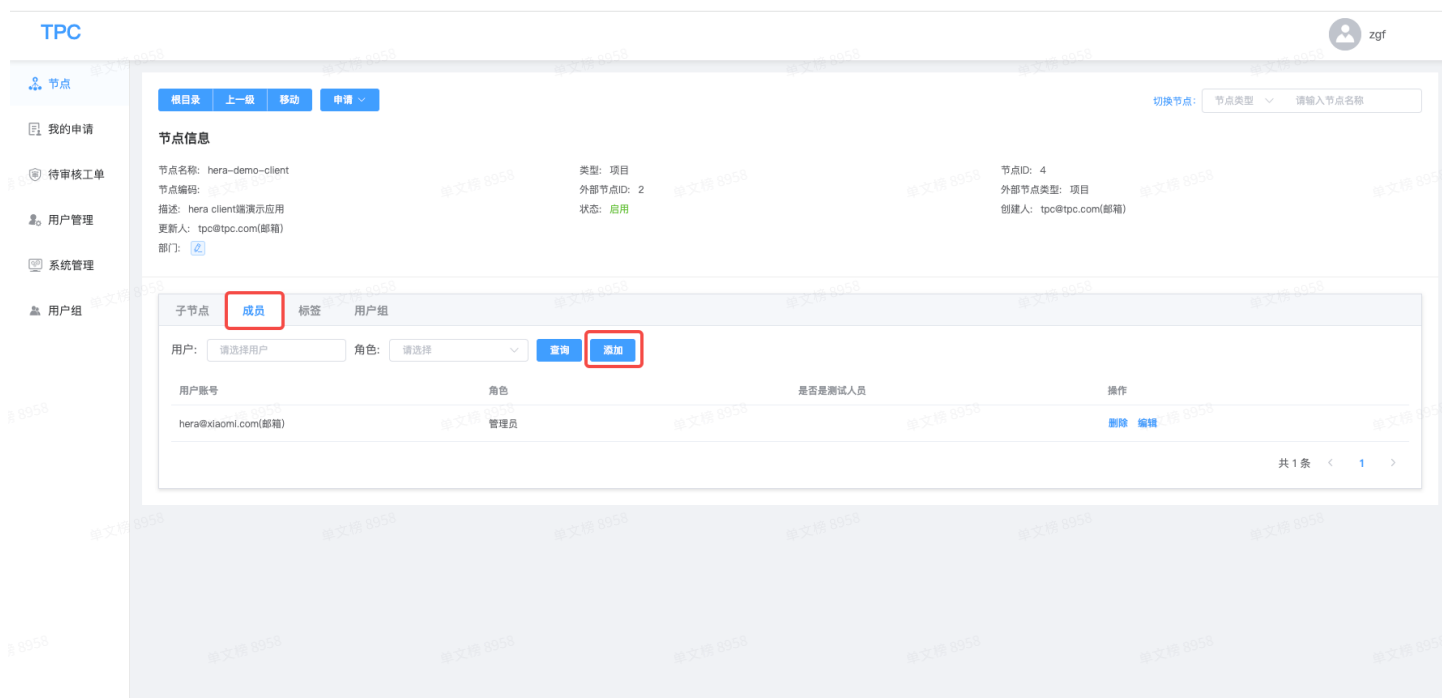




不管是哪种情况，注意这里的应用id与应用名称，都要与应用部署中的jvm参数：`-Dotel.resource.attributes=service.name` 与环境变量中的 `mione.app.name` 保持一致。

3、应用与人的权限

如果我们想给某一个应用添加管理员，让他可以在hera上有配置该应用的权限，我们可以在tpc中找到这个应用，点击应用名称，进入节点详情，在该应用节点的“成员”标签页中，点击添加按钮，可以选择将某个人的账号添加为管理员



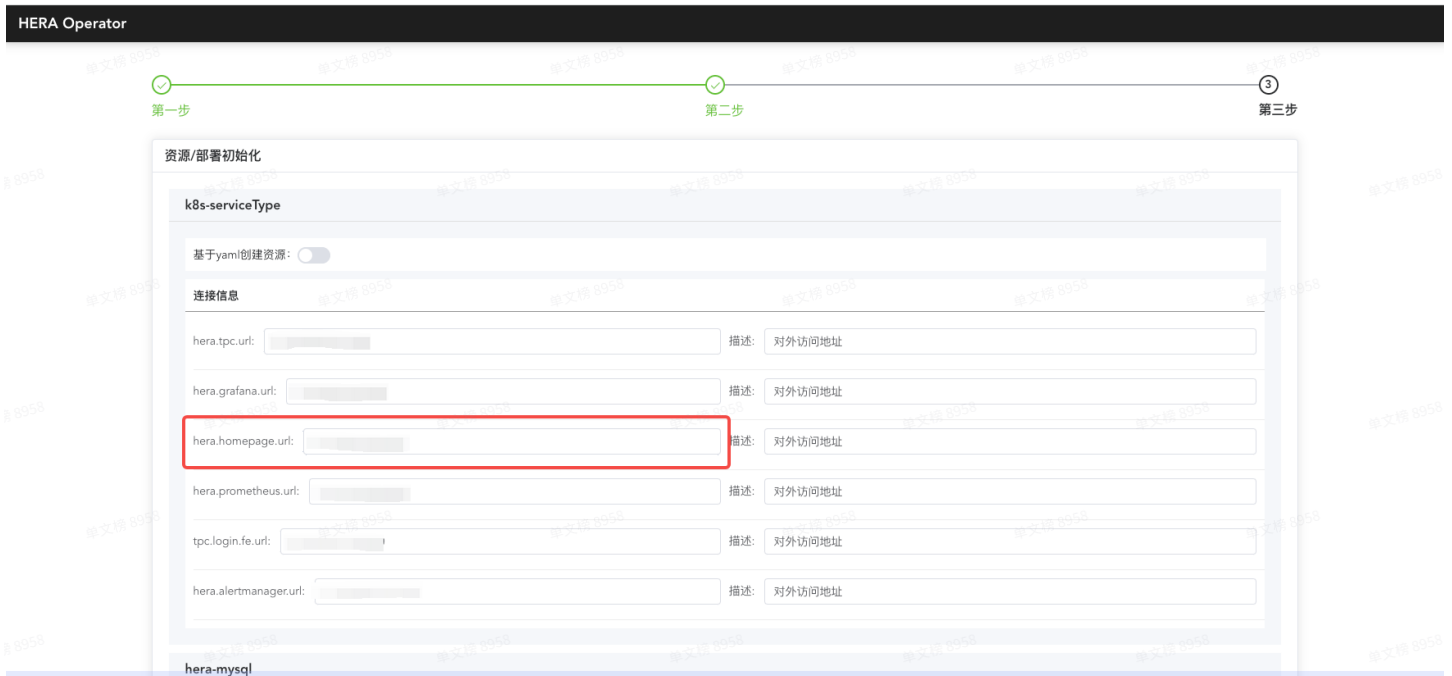


三、配置日志采集

由于Hera中的日志、trace数据都是由log-agent采集，所以我们需要在Hera日志配置中配置采集信息，告诉log-agent去哪采集，以及采集后发送给谁。

1、访问Hera日志页面

我们通过访问operator生成的hera首页的链接，在首页头部title中，点击“日志服务”，进入Hera日志的界面。





2、配置日志资源（后续会自动创建默认资源）

在日志页面左侧菜单栏，点击资源管理，填写RocketMQ与ES资源信息。

（1）RocketMQ

别名：RocketMQ资源别名，可以自定义

区域：选择“大陆机房”

mq地址：RocketMQ的地址，格式为host:port

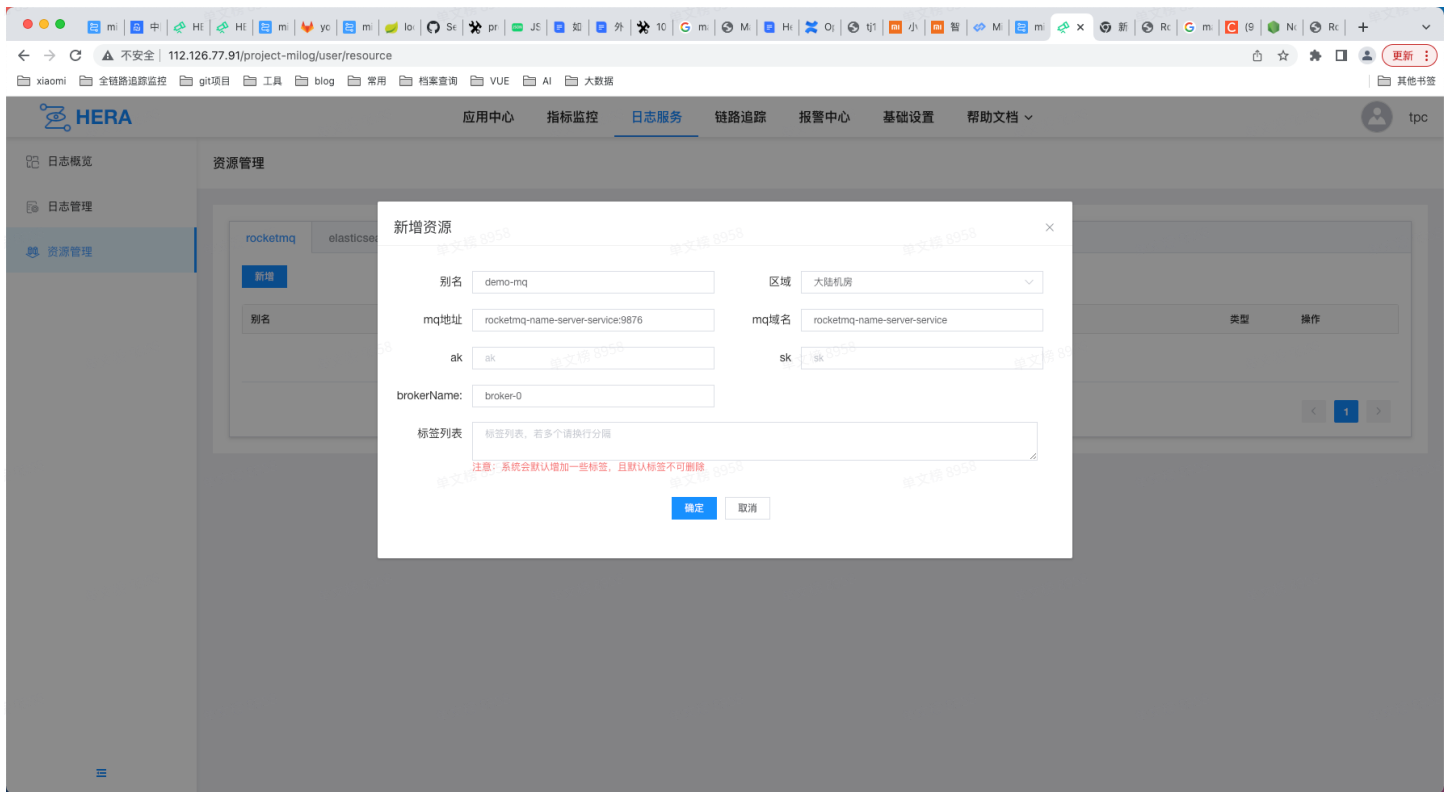
mq域名：RocketMQ地址的域名

ak：RocketMQ的accessKey，没有可不填

sk：RocketMQ的secretKey，没有可不填

brokerName：填写RocketMQ的brokerName，可以在RocketMQ的dashboard中查看，目的是为了创建日志所需要的topic，默认的为broker-0

标签列表：RocketMQ的tag列表，没有可不填



(2) ES

别名：ES资源别名，可以自定义

区域：选择“大陆机房”

es集群名称：同别名

esApi地址：ES的地址，格式为host:port

连接方式：选择连接ES的方式，分为“用户名和密码”与“Token”，用户名、密码、Token没有可以不填

标签列表：没有可不填

es索引组：我们会为每种日志类型默认创建一个索引组，但是需要用户将这四个索引组进行分别配置：

多行应用日志：mione_hera_log_multiple_app_log01

单行应用日志：mione_hera_log_single_app_log01

nginx日志：mione_hera_log_nginx_app_log01

opentelemetry日志：mione_hera_log_other_app_log01

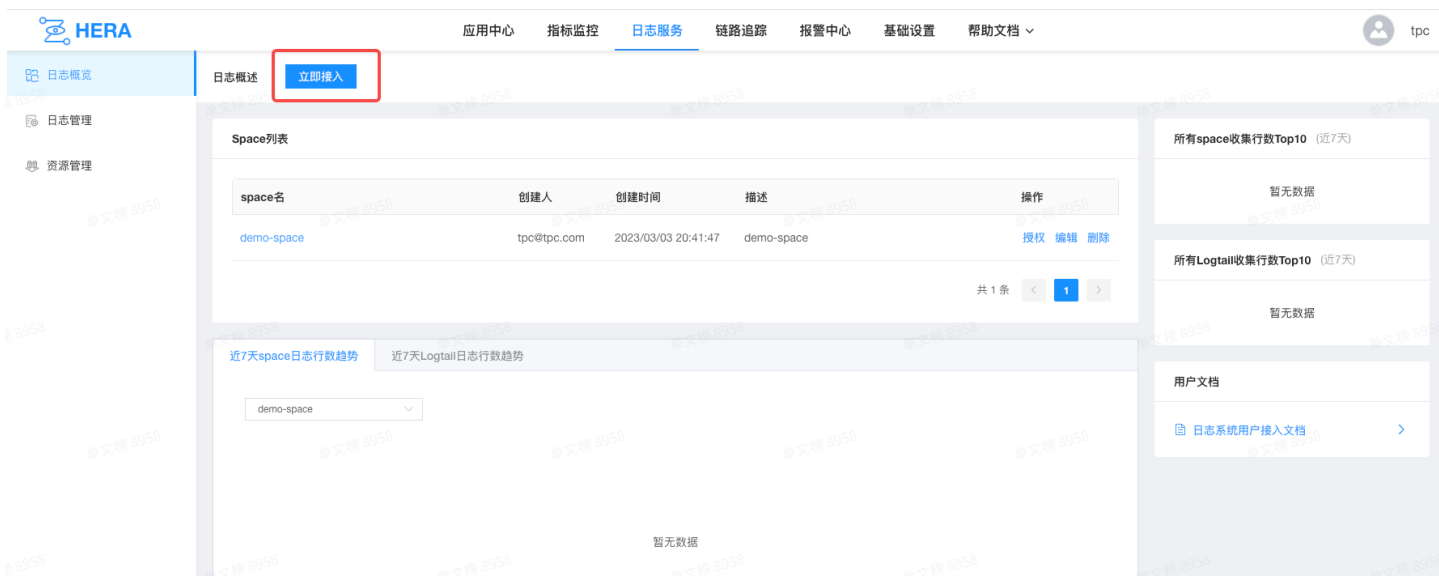


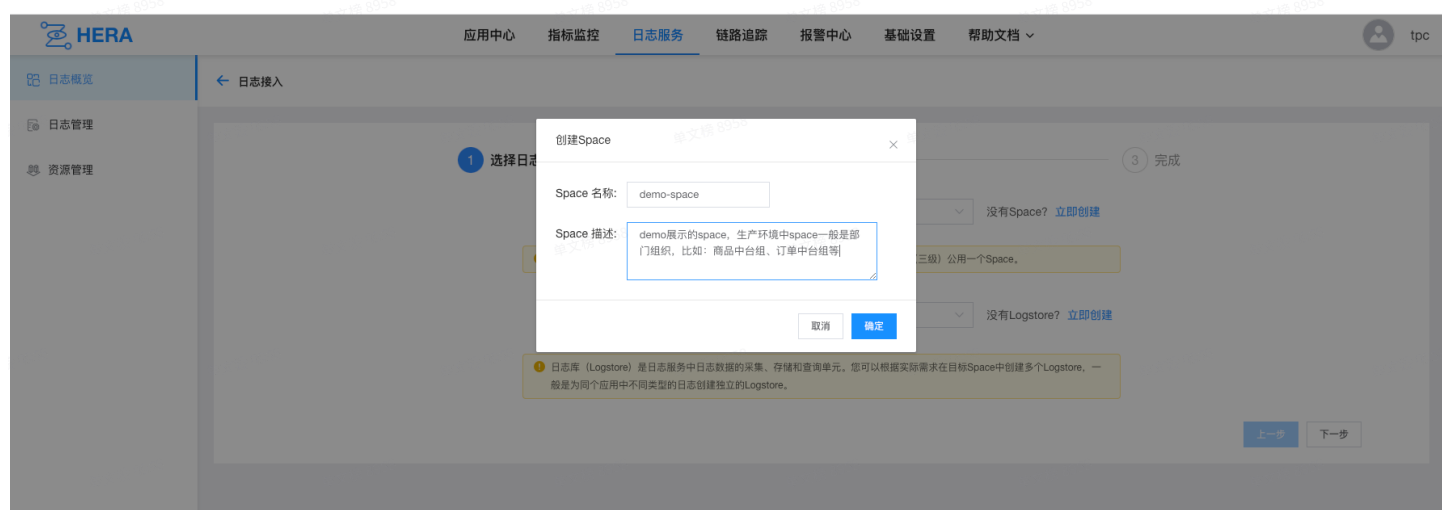
3、创建space

配置好资源后，我们需要创建space--store--tail这三个配置。

首先，在hera日志服务首页中，点击“立即接入”，进入日志接入页面。点击空间Space右侧的立即创建，创建一个Space。Space一般是部门组织级别。

注意：创建Space后，需要去tpc上hera--logger下对应的space节点下进行授权。

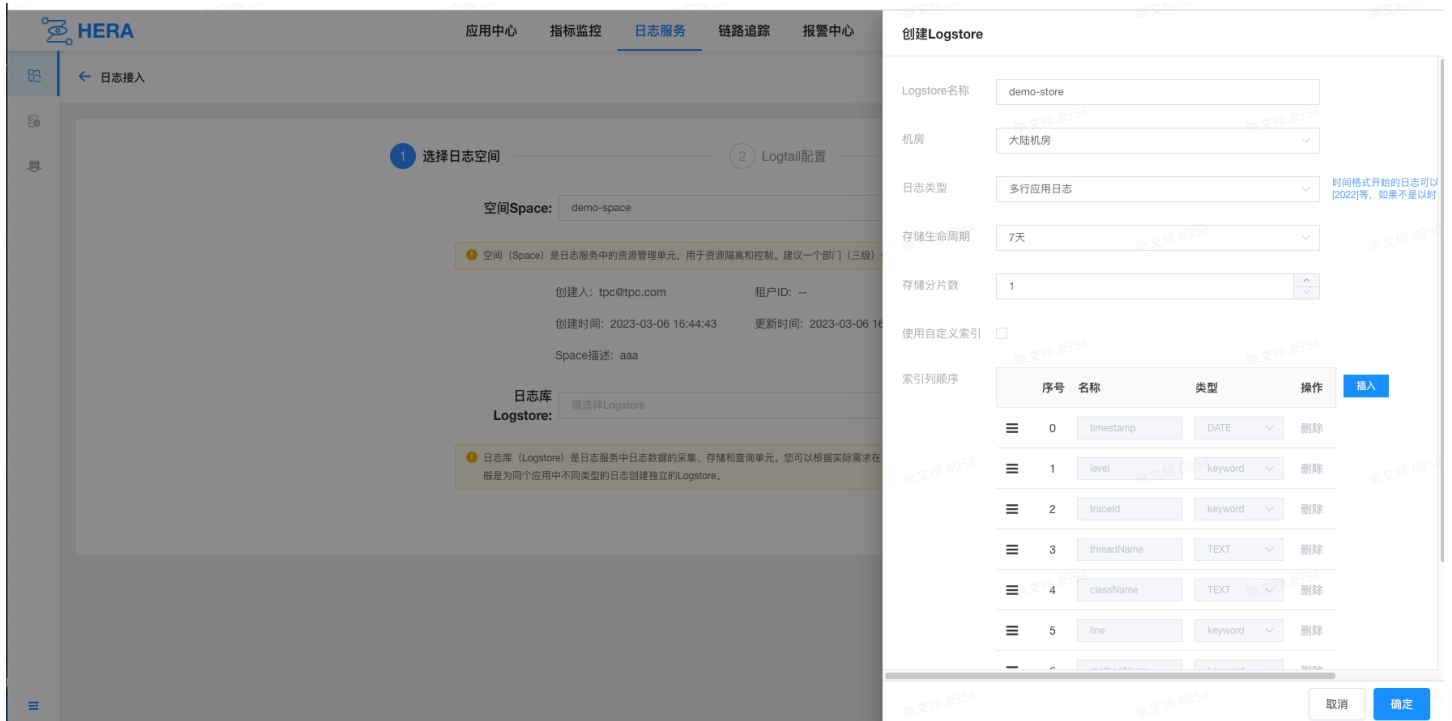




4、创建store

在创建Space后，我们在空间Space下拉框中选择刚刚创建的Space，点击日志库Logstore右侧的立即创建，创建store。





5、创建tail

应用类型：选择hera

部署方式：选择mione

服务应用名：选择需要接入的应用名

服务别名：可以自定义

环境分组：选择需要接入的环境

服务IP：选择环境后自动带出，需要进行勾选。如果没有，也可以手动进行添加

日志文件目录：应用日志文件的输出位置，具体到文件名

切分表达式：日志文件切分的命名规则

如果日志切分后和当前目录不在一起，且切分后的文件名的前缀不是日志为名的，则要自己手动填写切分表达式，例如：/home/work/logs/applogs/old/shopapi_application.log-*. 否则可不填。

行首正则表达式：如：^2022-*. *\$

日志解析类型：分割符、自定义、正则、json、nginx

解析脚本：

分割符：每一段是怎么分割的，输入分割你日志的分割符

自定义：如果是比较复杂的，可能每一部分都分割都不一样，则用自定义，例如：[%s]-[%s]-[%s]-[%s]-%s，会根据 %s 查找每一部分，每一部分的前缀和后缀

正则：输入正则表达式，会根据捕获组来提取字段

nginx：输入 nginx 日志格式配置，例如：

```

1 log_format milog2 '$time_iso8601\t$msec\t$host\t$request_method\t$scheme\t$status\t\t'
2 '$upstream_addr\t$upstream_cache_status\t$upstream_status\t'
3 '$server_addr\t$remote_addr\t$http_x_forwarded_for\t'
4 '$request_length\t$request_time\t$bytes_sent\t$upstream_response_time\t'
5 '$upstream_connect_time\t$upstream_header_time\t$upstream_response_length\t$upstream_bytes_received\t'
6 '$request_uri\t"$http_user_agent"\t'
7 '$sent_http_x_xiaomi_error_code';

```

收集速率：日志采集时多长时间发送一次，控制发送的速率

MQ配置：可不填，默认会将日志消息随机发送至某一个topic中

索引列：日志每一部分代表的含义，这个是从当前 store 中选择的，且顺序必须和日志中实际的日志中字段各部分的含义对应。store 的索引列必须是最多的，避免 tail 选择时没有。

HERA 应用中心 指标监控 日志服务 链路追踪 报警中心 基础设置 帮助文档 tpc

← 日志接入

1 选择日志空间 2 Logtail配置 3 完成

Logtail是HERA日志服务提供的日志采集Agent，用于采集各业务服务器上的日志。您可以根据实际需求在目标Logstore中创建多个Logtail，一般是为一次查看多个应用的日志创建独立的Logtail。

应用类型: hera

部署方式: mlone

服务应用名: china_hera-demo-client

服务别名: china_hera-demo-client

环境分组: dev

服务IP: 请输入服务IP 添加 全选

日志文件路径 (填写到文件名): /home/work/log/hera-demo-client/server.log

切分表达式: 切分表达式

行首正则表达式

将多行文本视为一个value，使用正则匹配首行

日志解析类型

分隔符

解析脚本

收集速率

常规收集

MQ配置

请选择

索引列

timestamp

level

threadName

traceId

className

line

message

序号	索引值	操作
0	timestamp	删除
1	level	删除
2	threadName	删除
3	traceId	删除
4	className	删除
5	line	删除
6	message	删除

测试以上配置

上一步

提交

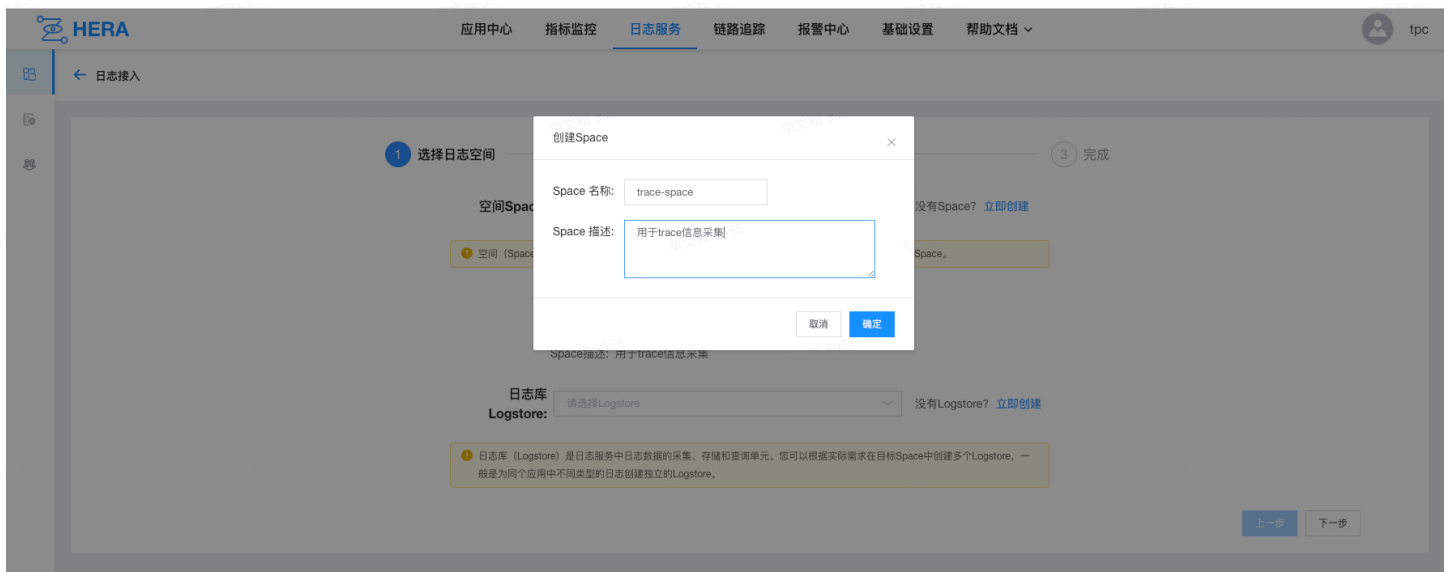
配置完成后，等待几秒钟的时间，就可以在日志页面查看日志了。



6、配置trace采集（后续可以自动创建）

在Hera中，trace信息是通过Hera日志中的log-agent进行收集的，我们是把trace信息当做是一种特殊的日志进行采集，所以还需要在Hera日志上配置trace的采集信息。只需要按照以下配置即可。

(1) 创建trace-space



(2) 创建trace-store

日志类型选择“opentelemetry日志”即可，其他的配置默认即可。



(3) 创建trace-tail

与一般的应用创建tail大致相同，只是配置相对来说比较固定。

服务应用名：只能选择china_log-agent

服务分组：选择default_env

服务IP：全选

日志文件路径：填写“/home/work/log/*/trace.log”。它会扫描所有/home/work/log下的trace.log文件进行采集

收集速率：选择快速收集

MQ配置：选择我们在日志资源中配置的RocketMQ集群，后面则是需要填写topic，如果无更改固定为：mone_hera_staging_trace_etl_server

HERA

应用中心 指标监控 日志服务 链路追踪 报警中心 基础设置 帮助文档

tpc

日志接入

选择日志空间

2 Logtail配置

3 完成

Logtail是HERA日志服务提供的日志采集Agent，用于采集各业务服务器上的日志。您可以根据实际需求在目标Logstore中创建多个Logtail，一般是为一次查看多个应用的日志创建独立的Logtail。

应用类型

hera

部署方式

mone

服务应用名

china_log-agent

服务别名

china_log-agent

环境分组

default_env

服务IP

请输入服务IP

添加

全选

10.53.129.76

10.53.128.247

日志文件路径

(具体到文件名)

/home/work/log/trace.log

切分表达式

切分表达式

行首正则表达式

将多行文本视为一个value，使用正则匹配首行

行首正则表达式

将多行文本视为一个value，使用正则匹配首行

日志解析类型

分隔符

解析脚本

|

收集速率

快速收集-尽快收集，占用一定资源

MQ配置

rocketmq / demo-mq

mone_hera_staging_trace_et

上一步

提交

四、首页关注应用

在首页添加应用到“我参与的应用”或者“我关注的应用”后，就可以查看监控、链路信息了。

HERA

应用中心指标监控日志服务链路追踪报警中心基础设置帮助文档

tpc

HERA - 云时代的系统可观测平台

请选择

请输入应用名称

log-agent

开源组参与暂无描述

+ 添加

hera-demo-client

开源组参与暂无描述

+ 添加

我参与的应用

我关注的应用

严重

警告

健康

HERA

应用中心指标监控日志服务链路追踪报警中心基础设置帮助文档

tpc

HERA - 云时代的系统可观测平台

已参与

请输入应用名称

我参与的应用

我关注的应用

严重

警告

健康

hera-demo-client

健康

开源组

0

0

0

0

指标报警

日志报警

调用异常

慢查询

Copyright © 2021 中国区&信息技术部&有品 保留所有权利

HERA

应用中心指标监控日志服务链路追踪报警中心基础设置帮助文档

tpc

指标监控

应用名: hera-demo-clientDashboard资源利用率调用异常概况(0)慢查询概况(0)接口大盘中间件

应用Dashboard

项目大盘列表

质量大盘列表

报警设置

应用健康度

应用报警策略

报警模板

PromQL模板

选择实例AllAll选择时间now-5m至now

应用健康度

可用性

100.0000%

80.0000%

60.0000%

40.0000%

20.0000%

0.0000%

21:54

21:55

21:56

21:57

21:58

sql100.0000%100.0000%

redis100.0000%100.0000%

HTTP99.9665%100.0000%

Dubbo调出99.8995%100.0000%

Dubbo调入100.0000%100.0000%

实例列表【主机数: 1】可以点击跳转到容器与物理机监控

实例(点击跳转到物理机监

状态存活

宿主机

容器启动时间(点击跳转到30.9 min

调用量变化(30s内总和)

300

250

200

150

100

50

0

21:53:30

21:54:00

21:54:30

21:55:00

21:55:30

21:56:00

21:56:30

21:57:00

21:57:30

21:58:00

HTTP271587

Dubbo调出18138.74

业务指标

HTTP调用 OPS

HTTP调用 PQO.DT