保密等级：内部公开





**腾讯专有云 Paas 平台**

**Tencent Cloud-native Suite**

**微服务平台(TSF)**

**升级指南**

文档版本：01

发布日期：2022年09月15日

|  |
| --- |
| 腾讯云计算（北京）有限责任公司 |

**版权声明**

本文档著作权归腾讯云计算（北京）有限责任公司（以下简称“腾讯云”）单独所有，未经腾讯云事先书面许可，任何主体不得以任何方式或理由使用本文档，包括但不限于复制、修改、传播、公开、剽窃全部或部分本文档内容。

本文档及其所含内容均属腾讯云内部资料，并且仅供腾讯云指定的主体查看。如果您非经腾讯云授权而获得本文档的全部或部分内容，敬请予以删除，切勿以复制、披露、传播等任何方式使用本文档或其任何内容，亦请切勿依本文档或其任何内容而采取任何行动。

**商标声明**



“腾讯”、“腾讯云”及其它腾讯云服务相关的商标、标识等均为腾讯云及其关联公司各自所有。若本文档涉及第三方主体的商标，则应依法由其权利人所有。

**免责声明**

本文档旨在向客户介绍本文档撰写时，腾讯云相关产品、服务的当时的整体概况，部分产品或服务在后续可能因技术调整或项目设计等任何原因，导致其服务内容、标准等有所调整。因此，本文档仅供参考，腾讯云不对其准确性、适用性或完整性等做任何保证。您所购买、使用的腾讯云产品、服务的种类、内容、服务标准等，应以您和腾讯云之间签署的合同约定为准，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 文档版本 | 发布日期 | 修订人 | 修订内容 |
| 01 | 2022-04-15 | p\_jiangfan | 第一次正式发布。 |
| 02 | 2022-09-16 | xfenggeng | tcs230版本升级（合入tsf-1.29.8） |

目录

目录

前言

文档目的

目标读者

符号约定

1 文档内容

2 变更概况

2.1 变更时间

2.2 变更影响范围

2.3 变更范围

2.4 兼容性评估

2.5 升级新增资源评估

3 部署架构变更说明

3.1 部署架构主要区别

3.2 访问链路升级前后对比

4 前期准备

4.1 获取TCS Master 节点地址

4.2 获取生产节点部署架构&ip信息

4.3 验证老版本功能是否正常

4.4 获取产品依赖的服务列表

4.5 升级前TSF使用功能检查和确认

4.6 创建demo测试服务

4.7 确认产品生产节点dns配置

4.8 物料提前下载

4.9 备份2.1.0版本现有物料运行版本

4.9.1 生产组件收集

4.9.2 镜像组件收集

4.9.3 yunapi组件收集

4.9.4 preset组件收集

4.9.5 dbsql组件收集

4.10 物料包更新

4.11 检查2.3.0物料包

4.11.1 确认yunapi物料包版本&个数

4.11.2 生成2.3.0镜像列表

4.11.3 确认yaml物料包版本&个数

4.11.4 确认对应配置文件是否存在

4.11.5 确认iplist正确性

4.11.6 提前load镜像

4.12 备份数据库及检查项

4.13 各组件当前vip 查询记录

4.14 检查vpc路由规则

5 管控变更

5.1 配置更新

命令:

预期结果:

异常处理:

回滚方案:

备注:

5.2 dbsql 更新（350以上原地升级需要）

参考命令:

命令:

预期结果:

异常处理:

回滚方案:

5.3 yunapi更新

命令:

预期结果:

异常处理:

回滚方案:

备注:

5.4 preset更新

命令:

预期结果:

异常处理:

回滚方案:

备注:

5.5 镜像更新

命令：

预期结果:

异常处理:

回滚方案:

备注:

5.6 product组件更新(2.3.0前端分产品进行更新, 填写在这里.)

命令：

预期结果:

异常处理:

回滚方案:

备注:

6 生产节点滚动升级变更

6.1 获取升级程序

命令：

预期结果:

6.2 升级 tsf-consul-access

命令：

预期结果:

异常处理:

回滚方案:

6.3 升级tsf-consul-register

命令：

预期结果:

异常处理:

回滚方案:

备注:

6.4 升级tsf-consul-config

命令：

预期结果:

异常处理:

回滚方案:

6.5 升级tsf-consul-authen

命令：

预期结果:

异常处理:

回滚方案:

6.6 更新容器镜像

命令：

预期结果:

回滚方案:

6.7 切换组件tsf-ctsdb/tsf-elasticsearch

命令：

预期结果:

回滚方案:

6.8 升级license-server

命令：

预期结果:

回滚方案:

6.9 组件检查确认

命令：

预期结果:

6.10 TSF consul-access进程下线

命令：

回滚方案:

7 验证阶段

7.1 [管控面] 登录验证

7.2 [管控面] 管控功能验证

7.3 [数据面] 运行状态验证

## 前言

适用于 TSF TCS2.1 升级 TCS2.3.0 版本。

## 文档目的

TCS TSF 提供升级步骤，和指导升级。

## 目标读者

* 行业架构师
* 研发架构师
* 研发工程师
* 产品经理
* 网络工程师
* 部署工程师
* 运维工程师
* 安全工程师
* 测试工程师
* 技术支持工程师

符号约定

本文档可能采用的符号约定如下所示：

|  |  |
| --- | --- |
| 符号 | 说明 |
|  | 表示是正文的附加信息，是对正文的强调和补充。 |
|  | 表示有低度的潜在风险，主要是用户必读或较关键信息，若用户忽略注意消息，可能会因误操作而带来一定的不良后果或者无法成功操作。 |
|  | 表示有中度的潜在风险，例如用户应注意的高危操作，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。 |
|  | 表示有高度潜在危险，例如用户应注意的禁用操作，如果不能避免，会导致系统崩溃、数据丢失且无法修复等严重问题。 |

## 文档内容

包括TCS版本下TSF云产品升级流程，以及验证和回退内容。

## 变更概况

### 变更时间

( 针对实际项目情况进行填写，可以联系项目负责人. 行业架构师, 行业PM, 现场PM )

示例:

时间 2022年9月10号20:00 - 2022年9月10号23:00

耗时 4h

### 变更影响范围

( 针对实际产品情况进行填写 )

管控端: TSF yunapi数据将被刷新,管控升级期间 2.3.0版本的租户端及运营端在升级期间不可用.

数据面: 2.3.0 tsf consul access 组件已经容器化，先拉起管控端 oloud-tsf-consul-access

### 变更范围

* TCS2.1.0 dbsql、preset、yunapi、frontend-imgcache等组件都是Global组件，只需要在TCS master（物料机）操作升级更新即可。

|  |  |  |
| --- | --- | --- |
| **类型** | **数量** | **组件名** |
| yaml | 39 | ocloud-tsf-adaptor  ocloud-tsf-consul-access  ocloud-tsf-controler  ocloud-tsf-license-server  ocloud-tsf-manager-dispatch  ocloud-tsf-manager-operation  ocloud-tsf-master  ocloud-tsf-masterapi  ocloud-tsf-mesh-apiserver  ocloud-tsf-mesh-mixs  ocloud-tsf-mesh-pilot  ocloud-tsf-ratelimit-master  ocloud-tsf-repository-access  ocloud-tsf-repository-server  ocloud-tsf-task-access  ocloud-tsf-task-server  ocloud-tsf-task-trigger  tcloud-tsf-analyst  tcloud-tsf-apm  tcloud-tsf-auth  tcloud-tsf-billing  tcloud-tsf-dcfg  tcloud-tsf-dispatch  tcloud-tsf-event  tcloud-tsf-gateway  tcloud-tsf-metrics  tcloud-tsf-monitor  tcloud-tsf-ms  tcloud-tsf-nameserver  tcloud-tsf-privilege  tcloud-tsf-ratelimit  tcloud-tsf-record  tcloud-tsf-resource  tcloud-tsf-route  tcloud-tsf-scalable  tcloud-tsf-task-console  tcloud-tsf-task-ops  tcloud-tsf-template  tcloud-tsf-token |
| image | 39 | 同上 |
| dbsql | 2 | dbsql-tsf-backend  dbsql-tsf-task-backend |
| yunapi | 1 | yunapi-tsf |
| product | 3 | product-tsf-mock  product-frontend-imgcache-tcs-ocloud-tsf  product-frontend-imgcache-tcs-ocloud-tsf-manager |
| preset | 1 | preset-tsf-cam-ocloud-tsf  preset-tsf-cam-tcloud-tsf |

## 兼容性评估

TSF产品2.3.0的管控与2.1.0的生产组件不兼容.

且2.3.0的生产组件与2.1.0的管控不兼容.

**需要管控 & 生产一起升级**

## 升级新增资源评估

2.1.0升级到2.3.0, 涉及consul access 组件容器化，涉及容器资源新增(增加大小?????)。

支撑资源无变化。

## 部署架构变更说明

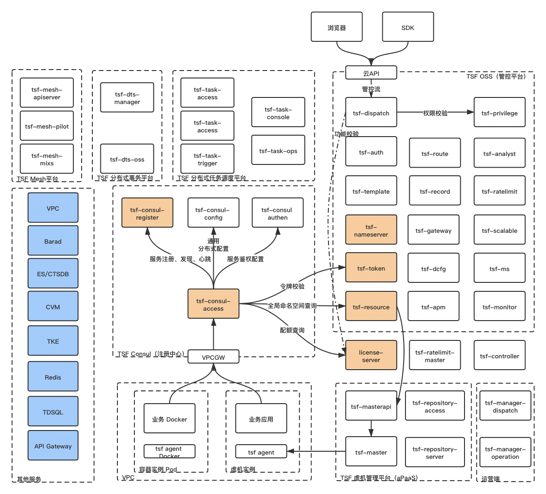
(针对实际产品情况进行填写)

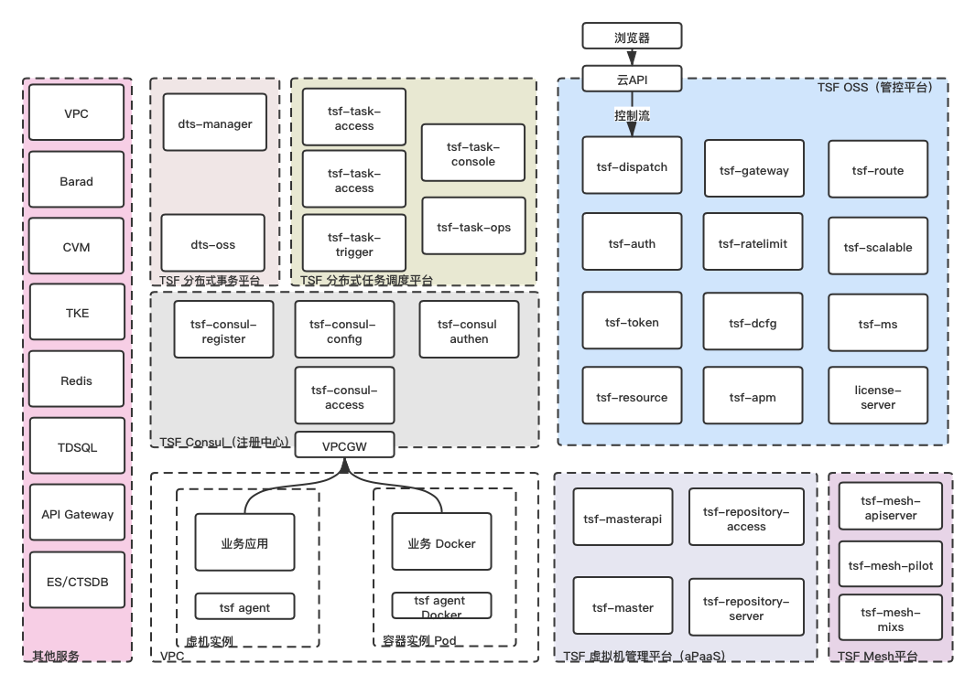
部署架构主要区别

|  |  |
| --- | --- |
| 版本 | 区别 |
| 2.1.0部署架构 | tsf-consul-access进程与consul其他组件混部署在虚机上 |
| 2.3.0部署架构 | tsf-consul-access进程现为kubernetes中的pod部署, 不再需要虚拟机进行部署 |



访问链路升级前后对比





### 前期准备

### 升级前需要确认TSF子集群已经升级完成

[TCS 2.1.2 子集群升级 2.3.0](https://doc.weixin.qq.com/doc/w3_APAA1wb6ACgSzrC3dodSoKoFVHGDQ?scode=AJEAIQdfAAolMpv6SeANAAygbhAD4)

### 获取TCS Master 节点地址

\*\*\* 重要 \*\*\* 可以联系项目负责人. 行业架构师, 行业PM, 现场PM 获取现场环境信息.

### 获取生产节点部署架构&ip信息

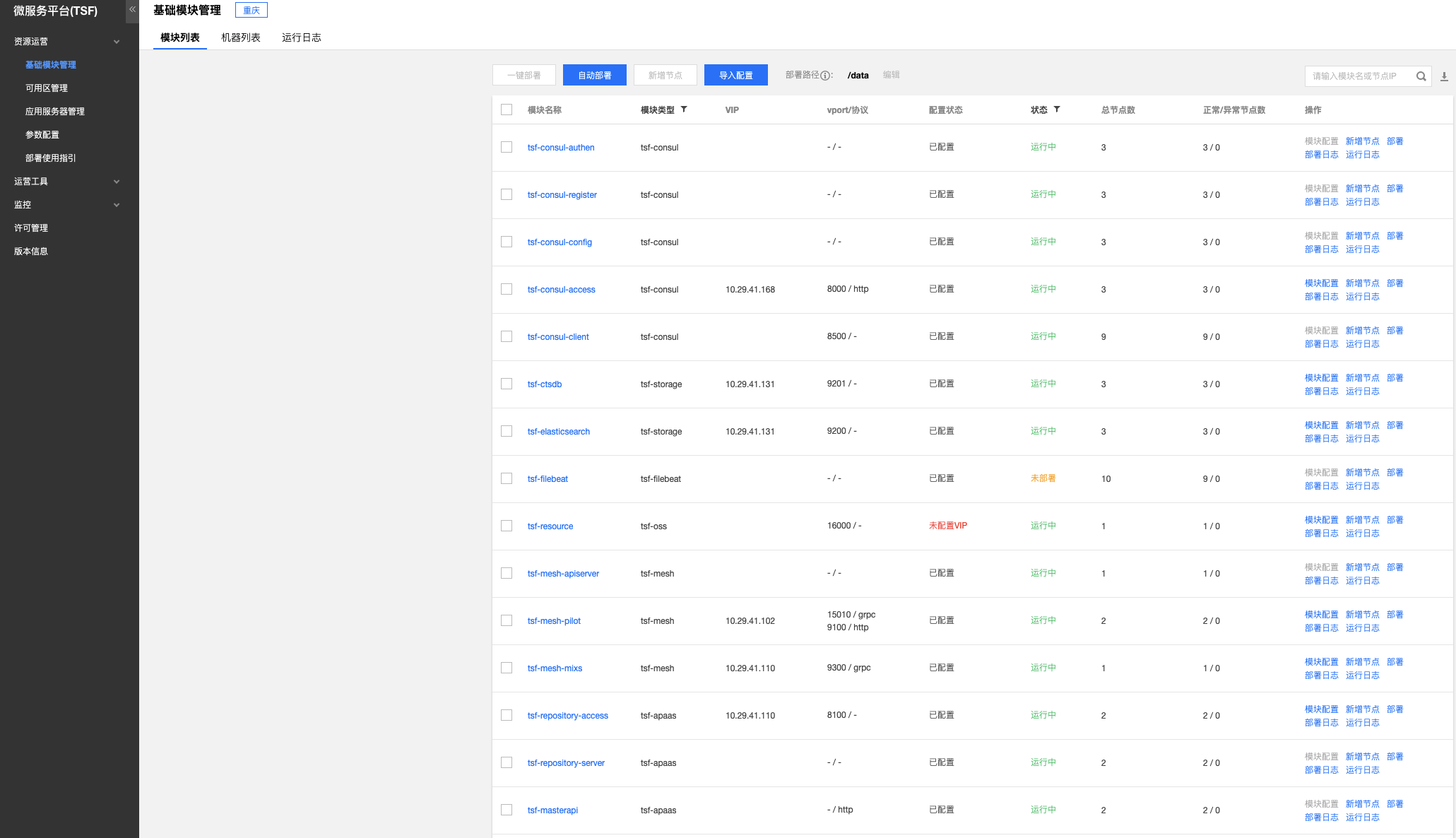
\*\*\* 重要 \*\*\* 可以联系项目负责人. 行业架构师, 行业PM, 现场PM 获取现场环境信息.

### 验证老版本功能是否正常

\*\*\* 重要 \*\*\*

(需要提供各产品检查方法. 1、多节点需要提供每个节点的验证方式. 2、基本功能验证方式.

1. 运营端--微服务平台（TSF），查看各组件运行状态及监控。(确认是否有显示异常或未部署的组件，TCS已升级步骤1、2可忽略）



1. 租户端检查集群、部署组、应用、服务治理、概览等菜单正常。
2. TCS 节点查看pod 运行状态，是否Running

kubectl get po -A |grep tsf-

1. consul 、es 、ctsdb 集群检查

# 登陆任意 consul master 节点，执行如下命令，检查集群leader 及 peers 输出是否包括所有master

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.23.0/bin && ./cluster\_health.sh

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.23.0/bin && ./cluster\_health.sh

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.23.0/bin && ./cluster\_health.sh

# elasticsearch 集群健康状态（选取ES其中一个节点执行）

curl -u root:changeme -XGET 127.0.0.1:9200/\_cluster/health?pretty

# ctsdb 集群健康状态（选取ctsdb其中一个节点执行）

curl -u root:changeme -XGET 127.0.0.1:9201/\_cluster/health?pretty

### 升级前TSF使用功能检查和确认

确认需要做2.1.0版本升级2.3.0版本的客户，是否有使用tsf-task、tsf-mesh。

# task 使用确认

##获取数据库登录信息

ip=`kubectl get secret -n tce dbsql-tsf-task-backend -o yaml |grep ipv4 |awk -F ' ' '{print $2}' |base64 -d`

port=`kubectl get secret -n tce dbsql-tsf-task-backend -o yaml |grep port |awk -F ' ' '{print $2}' |base64 -d`

user=`kubectl get secret -n tce dbsql-tsf-task-backend -o yaml |grep user |awk -F ' ' '{print $2}' |base64 -d`

pass=`kubectl get secret -n tce dbsql-tsf-task-backend -o yaml |grep pass |awk -F ' ' '{print $2}' |base64 -d`

# 登录数据库

mysql -h$ip -P$port -u$user -p$pass

# 进入指定库

use task\_schedule;

-- 检查是否有启用的任务

select \* from task\_record where state = 'ENABLED' limit 5;

# mesh 使用确认

# 查询是否有使用tsf-mesh 应用

ip=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep ipv4 |awk -F ' ' '{print $2}' |base64 -d`

port=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep port |awk -F ' ' '{print $2}' |base64 -d`

user=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep user |awk -F ' ' '{print $2}' |base64 -d`

pass=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep pass |awk -F ' ' '{print $2}' |base64 -d`

# 登录数据库

mysql -h$ip -P$port -u$user -p$pass

# 进入指定库

use tsf\_core;

select \* from tsf\_core.application where need\_mesh = '1';

# 查询是否有部署的mesh的机器，关注status状态， tsf-mesh":[{"name":"pilot-agent","status":"down"}]

select \* from appserver\_real where exists (select 1 from application where need\_mesh = '1' and instr(real\_status\_info,id) > 0 ) ;

### 确认产品生产节点dns配置

如果不涉及该项检查则删除示例内容，填无。

1. 确认产品生产节点中的dns是否配置正确

[root@tcs-10-0-9-53 ~]# cat /etc/resolv.conf

search global.chongqing chongqing

nameserver 10.202.0.49

nameserver 10.202.0.142

nameserver 10.202.0.47

options timeout:1 rotate

# DNS中IP显示为tcs master 节点

如果非以上预期结果, 需要检查当前配置是否正确.

1. 需替换 /etc/resolv.conf中nameserver为正确的dns服务器地址(当前TCS集群的master服务器三台IP)

### 物料提前下载

如果不涉及该项则填无。

提前下载物料放置部署机器，及检查物料版本。

# 物料包目录 /data/tce\_dc/software/

## 备份2.1.0版本现有物料运行版本

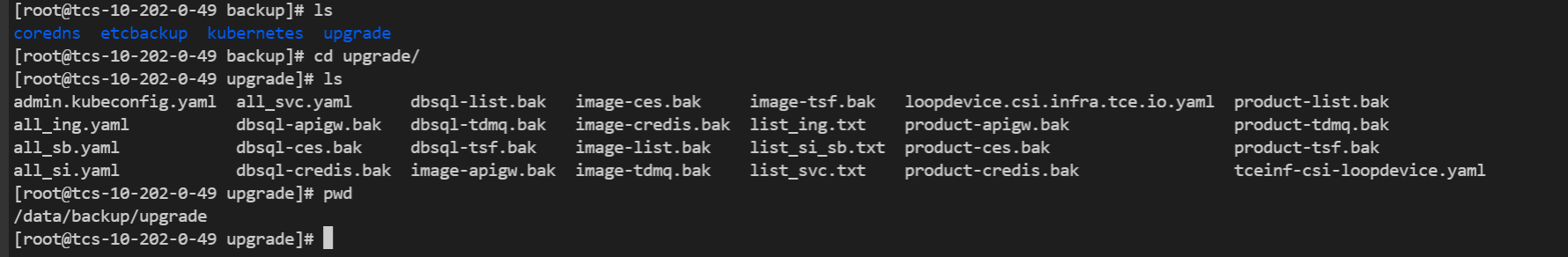
### 镜像组件收集

*回滚时，以当前运行image版本为准，收集当前运行的image版本，注意去重*

# 获取各镜像组件版本

若/data/backup/upgrade不为空且有组件的bak文件，可跳过镜像组件收集操作

预期结果：



# 在TCS Master 执行

kubectl get pods --all-namespaces -o jsonpath="{..image}" |tr -s '[[:space:]]' '\n' |sort | uniq|grep tsf > /data/tce\_dc/workspace/upgrade/tsf-image-$(date +%Y%m%d%H%M%S).bak

# 检查备份文件不为空

cat /data/tce\_dc/workspace/upgrade/tsf-image-$(date +%Y%m%d%H%M%S).bak

当前物料包的image版本

ls /data/tce\_dc/software/latest/image/\*tsf\*

对比两次查询的image版本结果，如不一致，联系现场架构师处理

yunapi组件收集

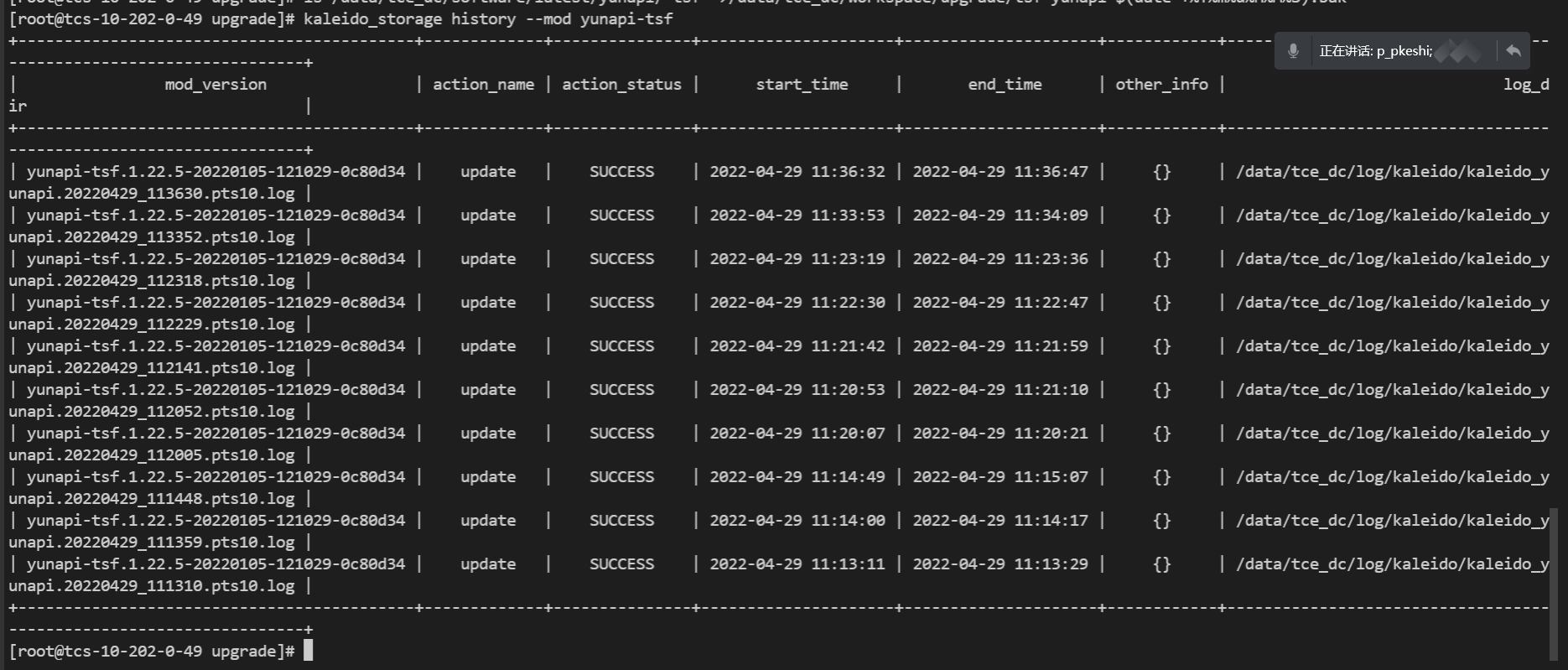
ls /data/tce\_dc/software/latest/yunapi/\*tsf\* >/data/tce\_dc/workspace/upgrade/tsf-yunapi-$(date +%Y%m%d%H%M%S).bak

# 检查备份文件不为空

cat /data/tce\_dc/workspace/upgrade/tsf-yunapi-$(date +%Y%m%d%H%M%S).bak

#查看tsf yunapi部署的历史版本并记录

kaleido\_storage history --mod yunapi-tsf



preset组件收集

收集当前消息预设数据物料包路径、cam预设策略物料包路径

ls /data/tce\_dc/software/latest/preset/preset-tsf-\* >/data/tce\_dc/workspace/upgrade/tsf-preset-$(date +%Y%m%d%H%M%S).bak

#查看 tsf cam-ocloud 和 cam-tcloud的历史版本记录

kaleido\_storage history --mod preset-tsf-cam-ocloud-tsf

#默认preset-tsf-cam-tcloud-tsf没有输出

kaleido\_storage history --mod preset-tsf-cam-tcloud-tsf

dbsql组件收集

tdsql涉及的dbsql包获取

ls /data/tce\_dc/software/latest/dbsql/\*tsf\* > /data/tce\_dc/workspace/upgrade/tsf-dbsql-$(date +%Y%m%d%H%M%S).bak

生产组件收集

收集现场物料包路径

ls /data/tce\_dc/software/latest/product/\*tsf\* > /data/tce\_dc/workspace/upgrade/tsf-product-$(date +%Y%m%d%H%M%S).bak

# 检查备份文件不为空

cat /data/tce\_dc/workspace/upgrade/tsf-product-$(date +%Y%m%d%H%M%S).bak

# 查看tsf mock部署的历史版本并记录

kaleido\_storage history --mod product-tsf-mock

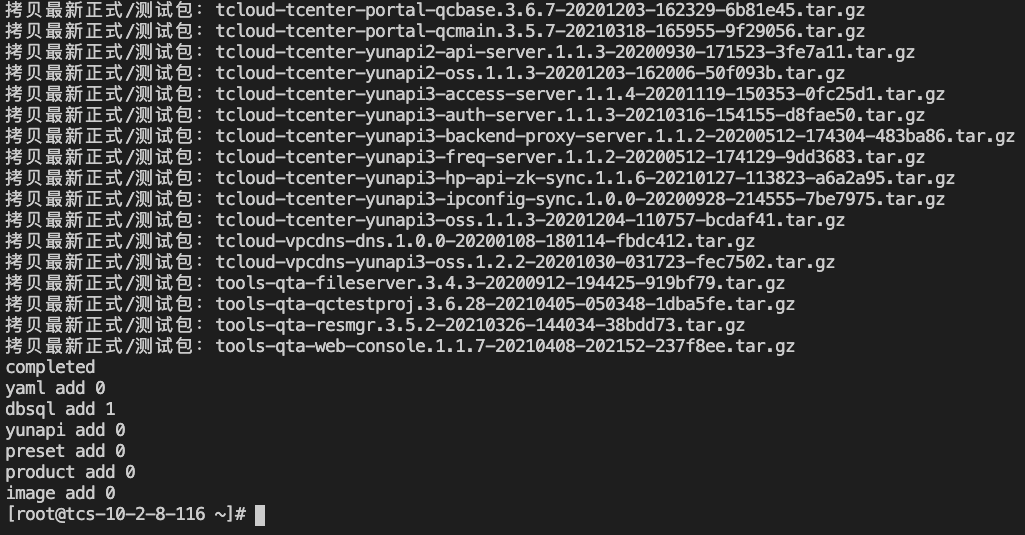


### 物料包更新

# 更新物料索引

kaleido\_refresh index

预期结果：



确认检查是否更新成功：

ll /data/tce\_dc/software/latest/image/\*tsf\* |grep 2.3.0

输出结果



异常处理:

示例: 正常情况下, 更新物料包不会出现非预期的场景, 如有异常, 请联系腾讯运维人员进行处理.

检查2.3.0物料包

根据实际产品进行确认，如果不涉及该项检查则删除示例内容，填无。

#### 确认yunapi物料包版本&个数

(后续其他物料包检查, 需要仿照 yunapi 的检查结果, 给出预期示例。)

# 检查yunapi组件个数

ls -l /data/tce\_dc/software/latest/yunapi/yunapi-tsf\*.tgz |grep 2.3.0 | wc -l

预期：1个

# 检查dbsql组件个数

ls -l /data/tce\_dc/software/latest/dbsql/dbsql-tsf\*.tgz |grep 2.3.0 | wc -l

预期：2个

# 检查preset组件个数

ls -l /data/tce\_dc/software/latest/preset/preset-tsf\*.tgz |grep 2.3.0 | wc -l

预期：2个

# 检查image组件个数（tsf-alarm 组件不用）

ls -l /data/tce\_dc/software/latest/image/\*-tsf-\*.tar.gz |grep 2.3.0 |wc -l

预期：39个

# 检查product组件个数

ls -l /data/tce\_dc/software/latest/product/product-tsf-\*.tgz |grep 2.3.0 | wc -l

预期：1个

# 检查imgcache组件个数

ls -l /data/tce\_dc/software/latest/product/product-frontend-imgcache\*tsf\*.tgz |grep 2.3.0 | wc -l

预期：2个

正常情况：可以看到latest下的物料包软链到哪里，输出物料数量为xx个

异常情况：没有对应的物料包

异常处理：找运维人员查询是否存是出包遗漏，并出包到现场

### 生成2.3.0镜像列表

需要根据实际情况进行填写，如无镜像组件，则填无。

示例:

2.3.0变更涉及需更新镜像

ocloud-tsf-adaptor

ocloud-tsf-consul-access

ocloud-tsf-controler

ocloud-tsf-license-server

ocloud-tsf-manager-dispatch

ocloud-tsf-manager-operation

ocloud-tsf-master

ocloud-tsf-masterapi

ocloud-tsf-mesh-apiserver

ocloud-tsf-mesh-mixs

ocloud-tsf-mesh-pilot

ocloud-tsf-ratelimit-master

ocloud-tsf-repository-access

ocloud-tsf-repository-server

ocloud-tsf-task-access

ocloud-tsf-task-server

ocloud-tsf-task-trigger

tcloud-tsf-analyst

tcloud-tsf-apm

tcloud-tsf-auth

tcloud-tsf-billing

tcloud-tsf-dcfg

tcloud-tsf-dispatch

tcloud-tsf-event

tcloud-tsf-gateway

tcloud-tsf-metrics

tcloud-tsf-monitor

tcloud-tsf-ms

tcloud-tsf-nameserver

tcloud-tsf-privilege

tcloud-tsf-ratelimit

tcloud-tsf-record

tcloud-tsf-resource

tcloud-tsf-route

tcloud-tsf-scalable

tcloud-tsf-task-console

tcloud-tsf-task-ops

tcloud-tsf-template

tcloud-tsf-token

将镜像列表写入message.list

cd /data/tce\_dc/workspace/upgrade

vim tsf.list

预期结果:

cat /data/tce\_dc/workspace/upgrade/tsf.list

ocloud-tsf-adaptor

ocloud-tsf-consul-access

ocloud-tsf-controler

ocloud-tsf-license-server

ocloud-tsf-manager-dispatch

ocloud-tsf-manager-operation

ocloud-tsf-master

ocloud-tsf-masterapi

ocloud-tsf-mesh-apiserver

ocloud-tsf-mesh-mixs

ocloud-tsf-mesh-pilot

ocloud-tsf-ratelimit-master

ocloud-tsf-repository-access

ocloud-tsf-repository-server

ocloud-tsf-task-access

ocloud-tsf-task-server

ocloud-tsf-task-trigger

tcloud-tsf-analyst

tcloud-tsf-apm

tcloud-tsf-auth

tcloud-tsf-billing

tcloud-tsf-dcfg

tcloud-tsf-dispatch

tcloud-tsf-event

tcloud-tsf-gateway

tcloud-tsf-metrics

tcloud-tsf-monitor

tcloud-tsf-ms

tcloud-tsf-nameserver

tcloud-tsf-privilege

tcloud-tsf-ratelimit

tcloud-tsf-record

tcloud-tsf-resource

tcloud-tsf-route

tcloud-tsf-scalable

tcloud-tsf-task-console

tcloud-tsf-task-ops

tcloud-tsf-template

tcloud-tsf-token

# 数量确认，预期39个

cat /data/tce\_dc/workspace/upgrade/tsf.list | wc -l

#### 确认yaml物料包版本&个数

# 核对yaml文件个数是否和镜像列表一致

cat /data/tce\_dc/workspace/upgrade/tsf.list |wc -l

ls /data/tce\_dc/software/latest/image/ |grep tsf |wc -l

异常情况：yaml文件数对不上

处理方法：检查对比出那个yaml文件缺少，找出包同学查极光平台上出包结果，并出包

#### 确认对应配置文件是否存在

1. 镜像配置

cd

/data/tce\_dc/config/cc/mod\_config/region\_50000001/zone\_50010001/gaia\_1/mod\_config/image

# 根据上面image组件列表执行，确认每个组件包对应的配置都存在

ls -l \*tsf\*.json |wc -l

1. yunapi配置

cd /data/tce\_dc/config/cc/mod\_config/region\_50000001/zone\_50010001/gaia\_1/mod\_config/yunapi

# 根据上面yunapi组件列表执行，确认每个组件包对应的配置都存在

ls -l yunapi-tsf.json

1. dbsql配置

cd /data/tce\_dc/config/cc/mod\_config/region\_50000001/zone\_50010001/mod\_config/product

# 根据上面dbsql组件列表执行，确认每个组件包对应的配置都存在

ls -l dbsql-tsf\*.json

1. preset配置

cd /data/tce\_dc/config/cc/mod\_config/region\_50000001/zone\_50010001/gaia\_1/mod\_config/preset

# 根据上面preset组件列表执行，确认每个组件包对应的配置都存在

ls -l preset-tsf\*.json

1. product配置

cd /data/tce\_dc/config/cc/mod\_config/region\_50000001/zone\_50010001/mod\_config/product

# 根据上面product组件列表执行，确认每个组件包对应的配置都存在

ls -l product-tsf\*.json

正常情况：存在对应的配置

异常情况：缺少配置

ls: 无法访问xxxx\*: 没有那个文件或目录

异常处理：找出配置包的同学确认是否存在该配置文件

### 如果组件个数较多可以通过列表方式查看，需要在上面配置组件列表

cd /data/tce\_dc/software/latest/image/

cat /data/tce\_dc/workspace/upgrade/tsf.list | awk '{print "ls -l " $1 "\*"}' | sh

#### 确认iplist正确性

# TCS 机器每个region下检查iplist文件地址是否正确，齐全

grep tsf /data/tce\_dc/config/self/iplist.csv

#### 提前load镜像

# 检查镜像是否导入：

kaleido\_image check push /data/tce\_dc/software/latest/yaml/ tsf

# 如果返回失败，则需要导入镜像：

# 增加passage work 节点 30+

kubectl scale deploy -n passage worker --replicas=45

kaleido\_image load /data/tce\_dc/software/latest/image --filter cloud-tsf

#### 环境底层信息收集确认

#资源信息收集会在升级TCS时操作，这里确认该目录中存在且文件不为空

cd /data/backup/pod-resource/

#收集当前所使用的images版本

cd /data/upgrade\_script

bash upgrade image\_version tsf

#查看当前所使用的image组件版本

cat /data/backup/pod-image-version/tcs-tsf.list

### 备份数据库及检查项

* 数据库备份

# TCS master上进行操作

若/data/backup/db下不为空，且包含要备份的数据库可跳过备份操作

预期结果：



mkdir -p /data/tce\_dc/workspace/upgrade/db\_backup

cd /data/tce\_dc/workspace/upgrade/db\_backup

#通过查看mysql获取登录信息

ip=`kubectl get si -n tce mariadb.db-2 -o yaml |grep host |awk -F ' ' '{print $2}' | sed -n '2p'`

port=`kubectl get si -n tce mariadb.db-2 -o yaml |grep port |awk -F ' ' '{print $2}' |tail -n1`

user=`kubectl get si -n tce mariadb.db-2 -o yaml |grep user |awk -F ' ' '{print $2}'`

pass=`kubectl get si -n tce mariadb.db-2 -o yaml |grep pass |awk -F ' ' '{print $2}' |tail -n1`

mysqldump -u$user -p$pass -h$ip -P$port

--databases tsf\_analyst tsf\_apm tsf\_auth tsf\_billing tsf\_core tsf\_dcfg tsf\_dispatch tsf\_gateway tsf\_lic tsf\_monitor tsf\_ms tsf\_nameserver tsf\_privilege tsf\_ratelimit tsf\_record tsf\_resource tsf\_route tsf\_scalable tsf\_template tsf\_token tsf\_tx tsfmanager\_operation > alltsfbak.sql

# 手工单独备份表：tsf\_module、tsf\_module\_port、tsf\_module\_vport、tsf\_parameter\_definition， 方便升级后进行数据核对(**此处需要手动操作**)

# tsf\_module 数据库备份

use tsfmanager\_operation;

create table tsf\_module\_bak like tsf\_module;

insert into tsf\_module\_bak select \* from tsf\_module;

# tsf\_module\_port 数据库备份

create table tsf\_module\_port\_bak like tsf\_module\_port;

insert into tsf\_module\_port\_bak select \* from tsf\_module\_port;

# tsf\_module\_vport 数据库备份

create table tsf\_module\_vport\_bak like tsf\_module\_vport;

insert into tsf\_module\_vport\_bak select \* from tsf\_module\_vport;

# tsf\_parameter\_definition数据库备份

create table tsf\_parameter\_definition\_bak like tsf\_parameter\_definition;

insert into tsf\_parameter\_definition\_bak select \* from tsf\_parameter\_definition;

#各组件当前vip 查询记录 并记录VIP信息

select \* from tsfmanager\_operation.tsf\_module where length(trim(external\_vip))>0;

mysql -u$user -p$pass -h$host -P$port

-e 'select \* from tsfmanager\_operation.tsf\_module where length(trim(external\_vip))>0;

' > /data/tce\_dc/workspace/upgrade/mydb\_user.txt

预期结果：



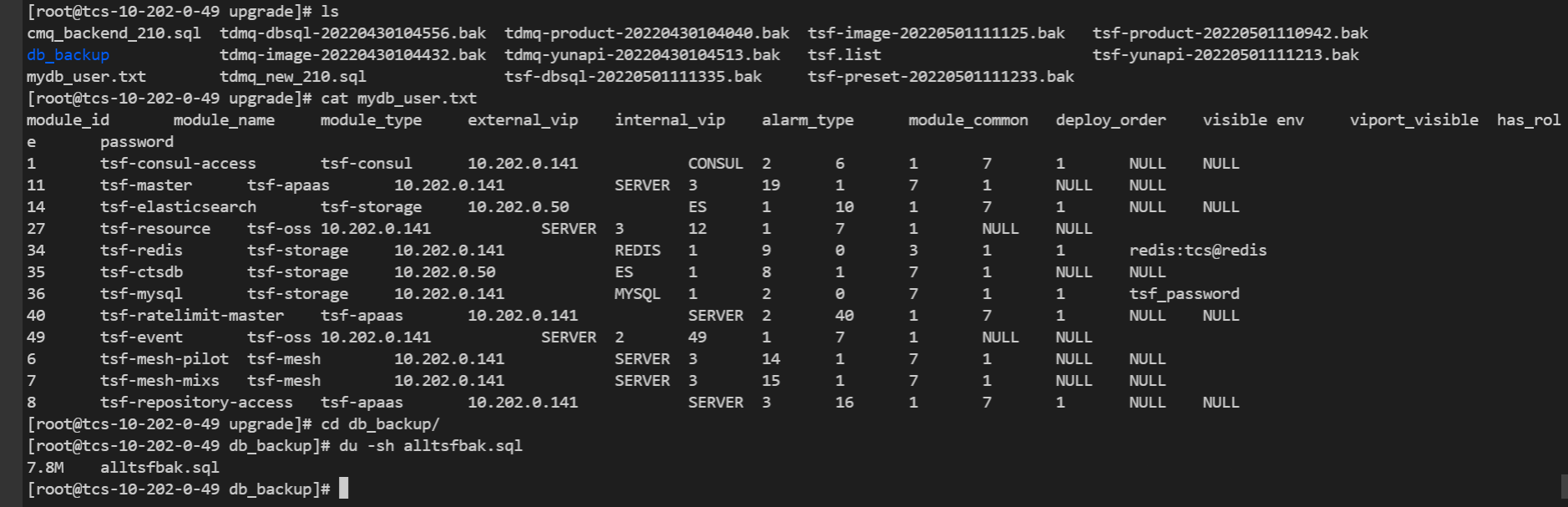
* 确认备份数据（每个region 做检查）

cd /data/tce\_dc/workspace/upgrade/db\_backup

检查备份文件大小，如果文件大于5M则正常，否则进行下面的步骤

du -sh alltsfbak.sql

预期结果：



## 管控变更

### 配置更新

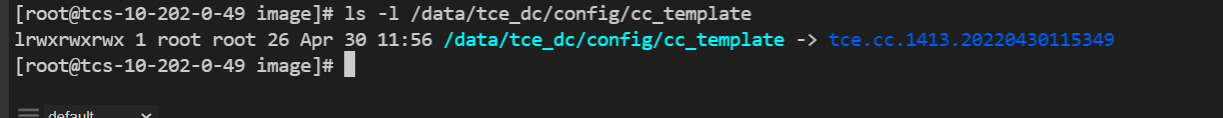
#### 操作命令:

1. 检查升级镜像列表是否都是tsf镜像名称

cat /data/tce\_dc/workspace/upgrade/tsf.list

1. 配置包备份：记录当前的配置包软链(截图保留）

ls -l /data/tce\_dc/config/cc\_template

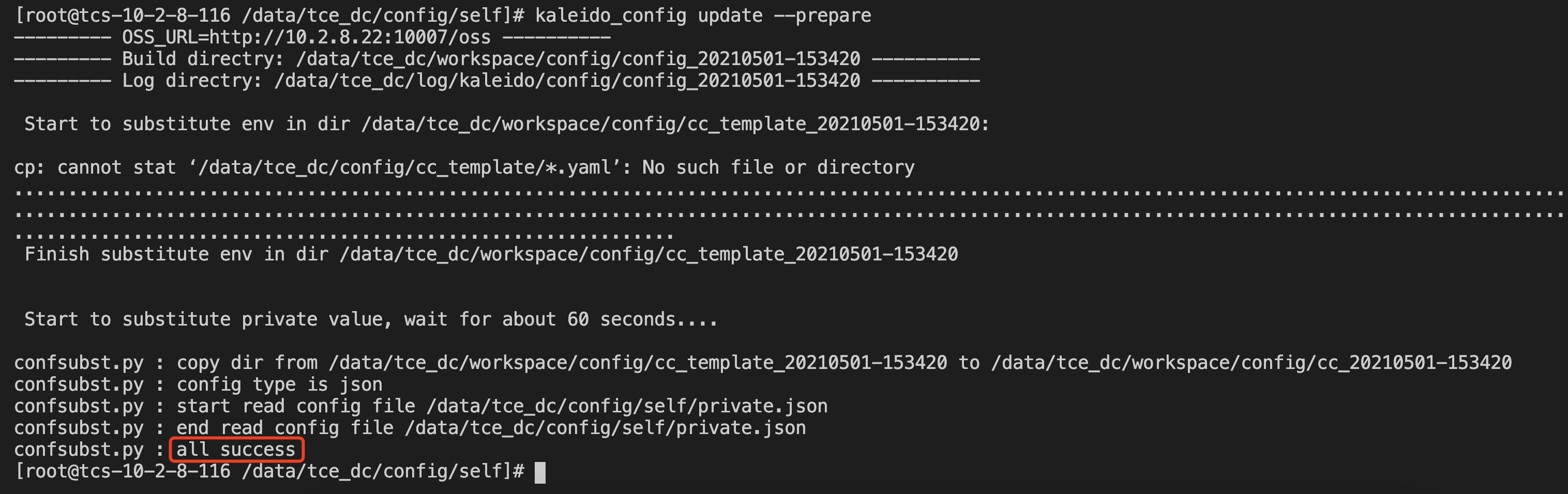


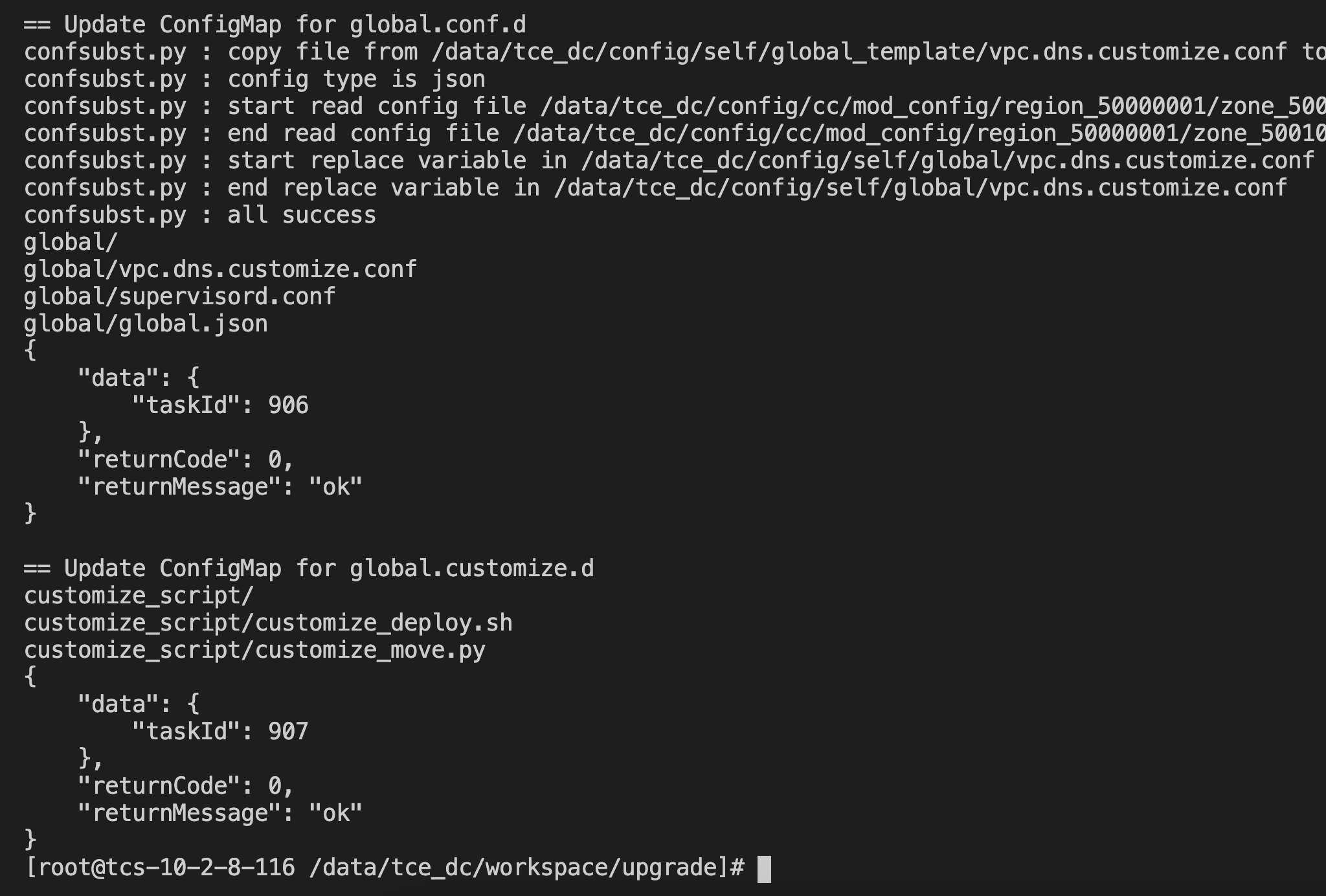
1. 配置更新

kaleido\_config update --prepare

kaleido config update --mod /data/tce\_dc/workspace/upgrade/tsf.list

#### 预期结果:





#### 异常处理:

联系运维人员确认。

#### 回滚方案:

1. 配置回滚

cd /data/tce\_dc/config;unlink cc\_template

1. 软件到配置采集时的配置包

ln -s ${tce\_cc} cc\_template

1. 刷新配置

kaleido\_config update --prepare

### dbsql 更新

根据实际情况删除示例部分，不涉及手动DBSQL变更则填无。

升级dbsql数据库

#### 操作命令:

kaleido dbsql sync #上传dbsql组件物料到node节点上

kaleido\_dbsql update ${dbsql\_component}

# dbsql\_component dbsql组件名称

**示例**： dbsql-tsf-backend组件拉起失败(此处只为示例，具体根据现场环境)

# 若dbsql拉起失败，查看对应生成的pod日志

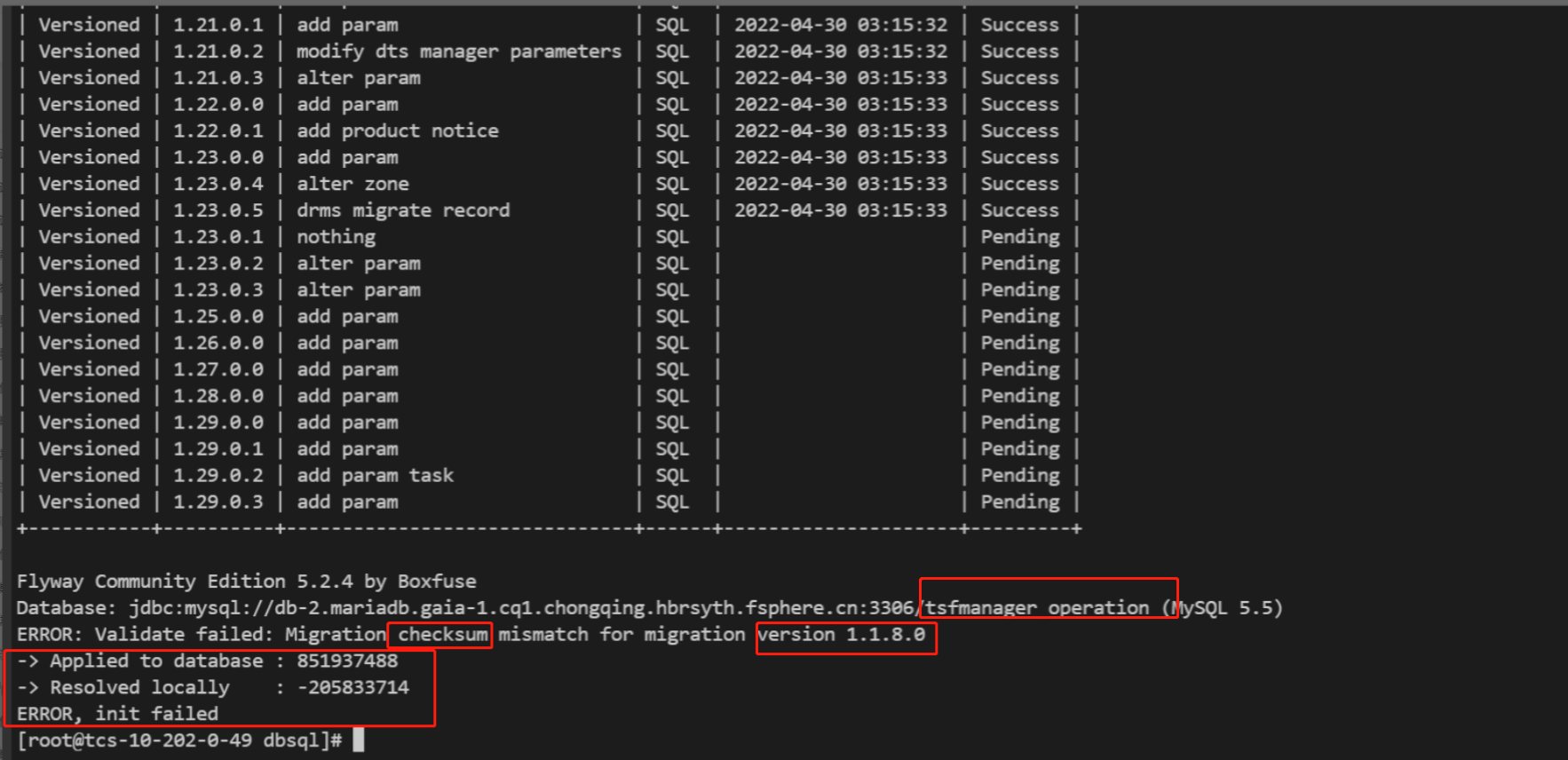
name=`kubectl get pod -n tce |grep tcloud-tsf-monito |awk '{print $1}'`

kubectl logs -f -n tce $name -c init

若查询报错如下图类型：

进行报错sql的替换，

需要替换的文件为 数据库为tsfmanager\_operation 版本为version 1.1.8.0 的sql文件



# 解压dbsql-tsf-backend组件包

dbsql-old=`ls /data/tce\_dc/software/{部署时组件包文件夹}/dbsql/ |grep dbsql-tsf-backend`

tar -zxvf $dbsql-old

dbsql-new=`ls /data/tce\_dc/software/latest/dbsql/ |grep dbsql-tsf-backend`

tar -zxvf $dbsql-new

#旧的sql包替换新的sql包

cp /data/tce\_dc/software/{部署时组件包文件夹}/dbsql/dbsql-tsf-backend/tsfmanager\_operation/{报错对应sql文件} /data/tce\_dc/software/latest/dbsql/dbsql-tsf-backend/tsfmanager\_operation/

#重新打包kaleido\_dbsql update组件包

cd /data/tce\_dc/software/latest/dbsql/

tar -czvf $dbsql-new dbsql-tsf-backend

#失败dbsql重新初始化

kaleido\_dbsql update dbsql-tsf-backend

dbsql 更新通过拉起pod 自动执行

#### 预期结果:

执行无报错。

#### 异常处理:

联系运维人员确认。

#### 回滚方案:

无

### yunapi更新

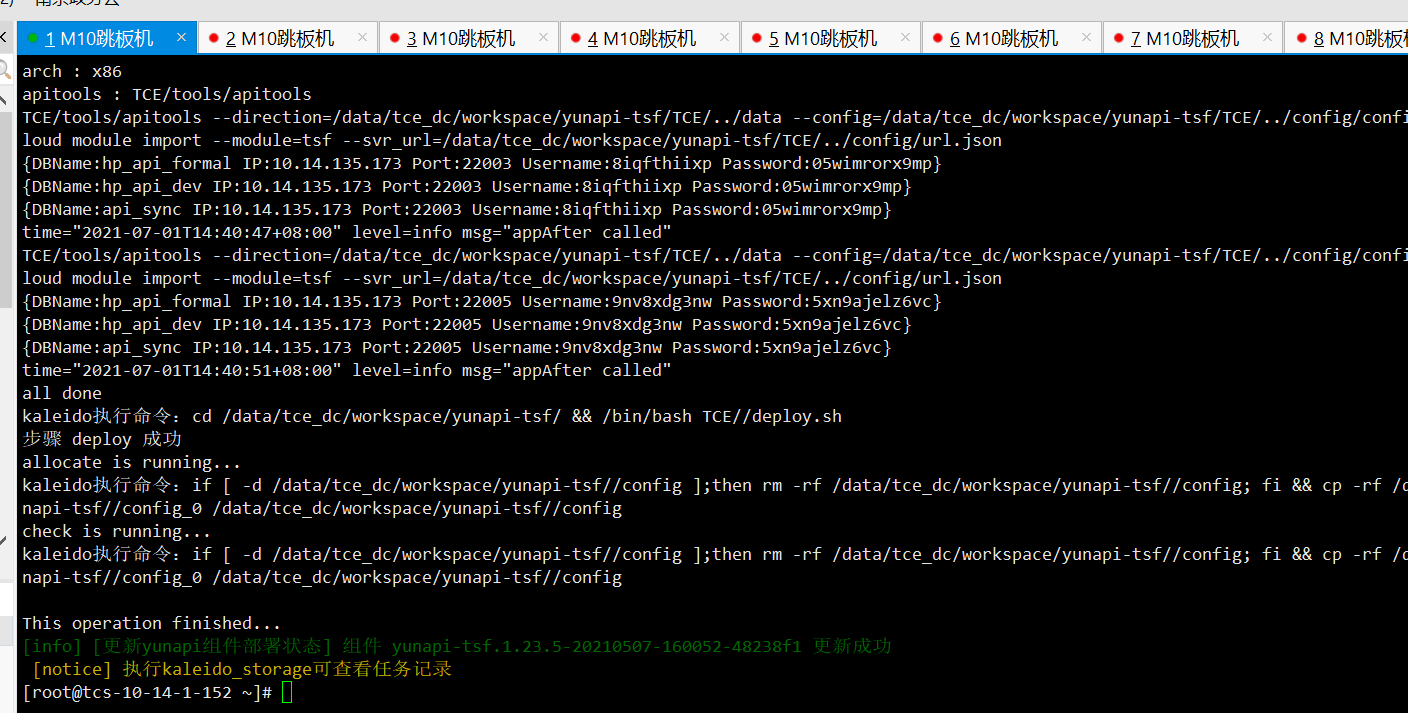
#### 操作命令:

kaleido\_yunapi update yunapi-tsf

## 若kaleido拉起失败，可使用以下命令

bash tad-deploy.sh yunapi-tsf

#### 预期结果:



#### 异常处理:

如果更新失败，则联系运维人员处理。

#### 回滚方案:

kaleido\_yunapi update /data/tce\_dc/software/TCS2.1.0.TXXX/yunapi/yunapi-tsf.xxx.tgz

将记录的2.1.0版本包回退，包路径参考 “前期准备 4.9.3yunapi组件收集”中生成的tsf-yunapi.bak 文件

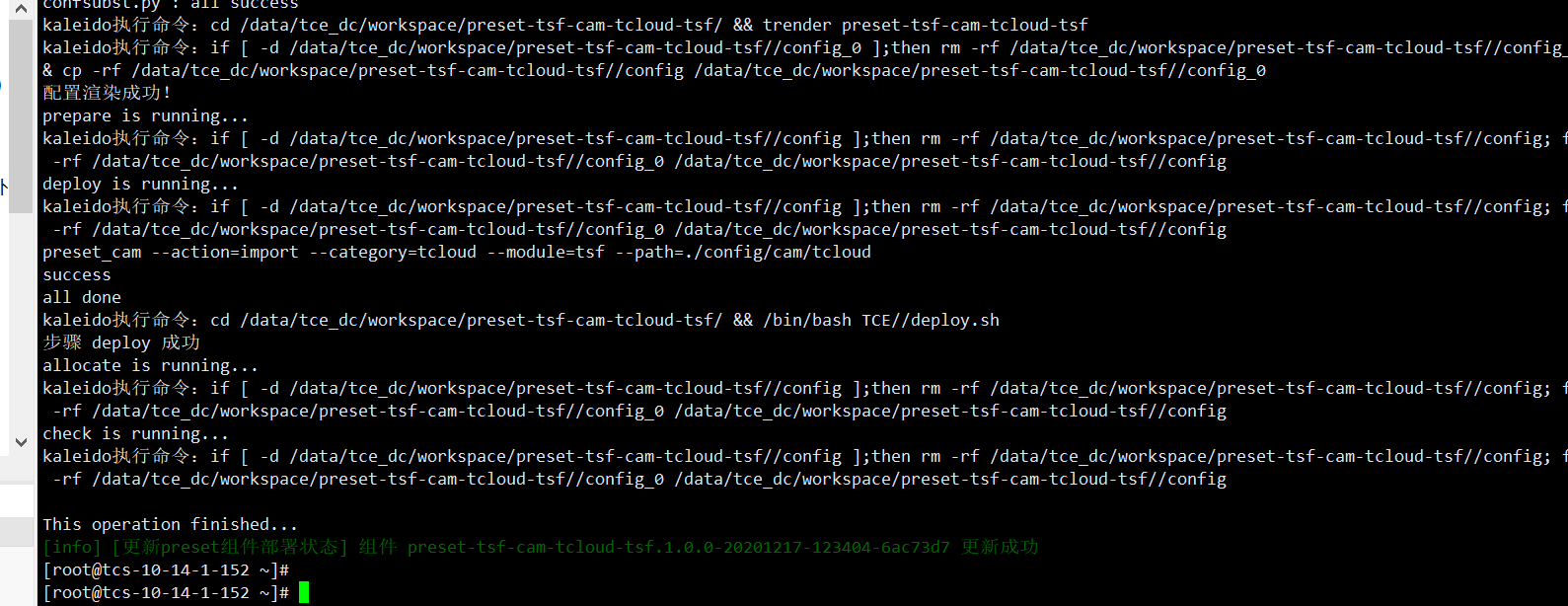
### preset更新

#### 操作命令:

kaleido\_preset update preset-tsf-cam-ocloud-tsf

kaleido\_preset update preset-tsf-cam-tcloud-tsf

#### 预期结果:



#### 异常处理:

联系运维人员处理

#### 回滚方案:

kaleido\_preset update /data/tce\_dc/software/TCS.TXXX/preset/preset-tsf-cam-ocloud-tsf.XXX.tgz

kaleido\_preset update /data/tce\_dc/software/TCS.TXXX/preset/preset-tsf-cam-tcloud-tsf.XXX.tgz

将记录的2.1.0版本包回退，包路径参考 “前期准备 2.9.4 preset组件收集”中生成的tdsql-preset文件

### 镜像更新

操作命令：

# 检查镜像是否导入：

kaleido\_image check push /data/tce\_dc/software/latest/yaml/ tsf

# 拉起pod, 注意根据预期结果中的命令检查最终的镜像版本

for i in `cat /data/tce\_dc/workspace/upgrade/tsf.list`;do tad-deploy.sh $i;done

# 检查dbsql 更新成功:completed，如有异常，则参考下面异常处理步骤

kubectl get serviceinit -ntce |grep tsf

# dbsql 执行成功后，配置信息核对，以免拉起pod后配置被改错，如果有差异数据，需要核对确认是否要回退为原配置（一般以升级前的配置为准）

#通过查看dbsql secret的yaml获取登录信息

ip=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep ipv4 |awk -F ' ' '{print $2}' |base64 -d`

port=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep port |awk -F ' ' '{print $2}' |base64 -d`

user=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep user |awk -F ' ' '{print $2}' |base64 -d`

pass=`kubectl get secret -n tce dbsql-tsf-backend -o yaml |grep pass |awk -F ' ' '{print $2}' |base64 -d`

mysql -u$user -p$pass -h$ip -P$port

# 信息对比 如有不同，升级后运营端进行修改

use tsfmanager\_operation;

select \* from tsf\_module a,tsf\_module\_bak b where a.module\_id = b.module\_id and a.external\_vip <> b.external\_vip;

select \* from tsf\_module\_port a,tsf\_module\_port\_bak b where a.module\_id = b.module\_id and a.port\_type = b.port\_type and a.port <> b.port;

select \* from tsf\_module\_vport a,tsf\_module\_vport\_bak b where a.module\_id = b.module\_id and a.vport\_type = b.vport\_type and a.vport <> b.vport;

select a.parameter\_id,a.parameter\_name,a.parameter\_value,b.parameter\_value from tsf\_parameter\_definition a,tsf\_parameter\_definition\_bak b where a.parameter\_id = b.parameter\_id and a.parameter\_value <> b.parameter\_value;

# 查看部署状态，如果是未部署，改成运行（此处需要操作）。

select \* from tsf\_module\_instance where status != '运行';

示例：

update tsf\_module\_instance set status = '运行' where module\_id = 47

预期结果:

* # 查看拉起状态为Running，无pending

kubectl get pod -ntce | grep tsf

# 查看镜像版本为最新2.1.0版本(跟latest下yaml 文件比对是否存在）

for image in `cat /data/tce\_dc/workspace/upgrade/tsf.list`;do kubectl get pods -ntce -o jsonpath="{..image}" |tr -s '[[:space:]]' '\n' |sort | uniq|grep $image |sed 's/registry.tce.com\/tce\///' |sed 's/:/./'|sort | awk '{print "ls -l /data/tce\_dc/software/latest/yaml/" $1 "\*"}' |sh ;done

# 若有组件版本不对，手动进行更新，若tad更新失败，可使用kaledio 再次拉起

tad-deploy.sh 组件名

#例如： kaleido\_image upgrade --yaml /data/tce\_dc/software/latest/yaml/ocloud-tsf-masterapi.2.3.0-20220323-014530-12e236c.tgz

注意⚠️：如果出现文件不存在，则表示正在运行的的2.1.0镜像版本比2.3.0的新，需要再指定镜像版本更新

异常处理:

# pod 拉起失败原因检查

https://docs.qq.com/sheet/DWlNDb2xIWUFGb3Na?tab=p4059t

# curl 查询pajero 服务注册信息

curl -X GET 127.0.0.1:30150/api/v1alpha1/service/instances?serviceID=secret-tsf

kubectl delete pod sd-controller-xxxx -ntcs-system

# 检查注册情况，有数据返回则注册成功，可以查看config 是否ready

curl -X GET 127.0.0.1:30150/api/v1alpha1/service/instances?serviceID=secret-tsf

回滚方案:

回滚配置中心，参考配置变更回滚步骤

将记录的2.1.0版本包回退，包路径参考 “前期准备 2.9.2镜像组件收集”中生成的tsf-image.bak文件

回滚命令:

kaleido\_image upgrade /data/tce\_dc/software/TCE3.6.0.Txxx/yaml/ocloud-tsf-resource.xxx.tgz

kaleido\_image upgrade

### product组件更新

操作命令：

#运营端tsf

kaleido\_product install product-frontend-imgcache-tcs-ocloud-tsf

#租户端tsf

kaleido\_product install product-frontend-imgcache-tcs-ocloud-tsf-manager

kaleido\_product install product-tsf-mock

预期结果:

运营端、租户端TSF 各菜单页面显示正常，无报错。

异常处理:

咨询运维人员。

回滚方案:

#运营端tsf

kaleido\_product install product-frontend-imgcache-ocloud-tsf --action=back

#租户端tsf

kaleido\_product install product-frontend-imgcache-tcloud-tsf

--action=back

## 生产节点滚动升级变更

### 获取升级程序

操作命令：

在TCS master 上执行如下脚本，获取需要升级的程序包。

vim get\_app.sh

#!/bin/bash

mkdir -p /root/tsf-pkgs/ && cd /root/tsf-pkgs/

PDNAME=`kubectl get pod -n tce |grep ocloud-tsf-manager-operation |awk '{print $1}'|head -n1`

FILE\_PATH=("/opt/repository/tsf/tsf-consul/tsf-consul-access/" "/opt/repository/tsf/tsf-consul/tsf-consul-authen/" "/opt/repository/tsf/tsf-consul/tsf-consul-register/" "/opt/repository/tsf/tsf-consul/tsf-consul-config/")

for APP\_PATH in ${FILE\_PATH[@]};do

PDFILE=`kubectl exec ${PDNAME} -n tce -- ls ${APP\_PATH}`

kubectl cp tce/${PDNAME}:${APP\_PATH}${PDFILE} ${PDFILE}

done

# 执行获取升级程序

sh get\_app.sh

查看目录下程序包 ll /root/tsf-pkgs/

#分发组件

dptool copy "/root/tsf-pkgs" "/data/" "10.0.8.165"

### 升级tsf-consul-register

操作命令：

* [生产节点] 流量切换

# ssh consul-register-slave1

# 注释crontab 任务，crontab -e，先取消crontab，以防leader 被自动回切

sed '/register/s/^/#/g' /var/spool/cron/root -i

# crontab -l 确认

# \*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/log/crontab.log

# 确认该10.29.12.51节点是非leader，确认命令

#获取本机ip

registry\_ip=10.29.12.51

curl $registry\_ip:8050/v1/status/leader

curl $registry\_ip:8050/v1/status/peers

curl http://$registry\_ip:8050/v1/operator/raft/configuration?pretty

# 如果升级节点当前为leader，执行切换，将 10.29.12.51 的leader 切换给节点10.29.12.52

registry\_ip=10.29.12.51

new\_leader=10.29.12.52

# 执行leader 切换,理论上只执行一次

curl -v "http://$registry\_ip:8050/v1/operator/raft/transfer?address=$new\_leader:8054" -X POST

# 切换后确认

curl http://$registry\_ip:8050/v1/operator/raft/configuration?pretty

# 期望输出结果ip是 new\_leader ip





# 确认该节点为follower后，进行 tsf-consul-register 流量摘除

# slave 节点不提供服务，同如下操作，不需要切换leader

# mysql 登录

update tsf\_core.consul\_server set used = 0 where ip = '10.0.9.53' and port = '8050';

# 摘除流量后 等待一分钟观察现象，无问题后执行下一步

# 可以查看摘除节点的log，不会刷新日志；查看leader日志，确保没有异常

tailf /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/log/consul-xxxx.log

* [生产节点] 程序包升级

# 登陆解压

cd /data && unzip tsf-consul-register-v1.29.7.zip

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-register

netstat -anp |grep 8050

# 备份程序

mkdir -p /data/tsf-back

cp -rf /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0 /data/tsf-back/tsf-consul-register-v1.26.0\_bak

# 更新程序

# unalias cp

cp -rf /data/tsf-consul-register-v1.29.7/bin /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0

# 修改文件夹版本为1.29

mv /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/ /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.29.7/

# 启动进程

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.29.7/bin && sh start.sh

# 查看是否加入集群

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.29.7/bin && sh cluster\_health.sh

# 测试该节点是否正常

registry\_ip=10.29.12.51

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8050/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://$registry\_ip:8050/v1/kv/foo?raw

# 测试通过后，恢复流量

update tsf\_core.consul\_server set used=1 where  ip='10.0.0.0' and port=8050

# 取消注释crontab 任务，并修改1.23.0 --》1.26.0 crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.29.7/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.29.7/log/crontab.log

# 使用替换

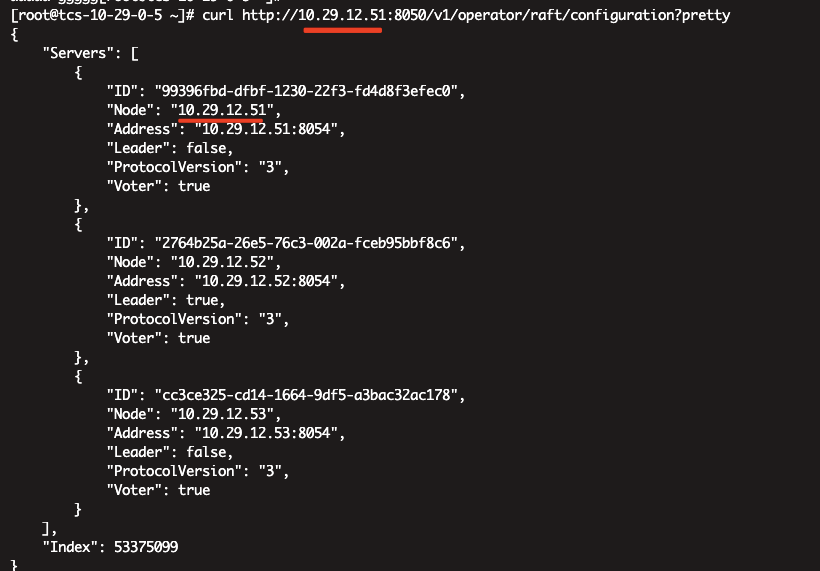
:% s/tsf-consul-register-v1.26.0/tsf-consul-register-v1.29.7/g

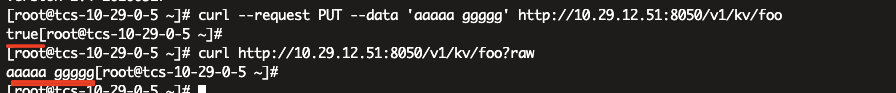
# 检查修改的是否正确

crontab -l |grep tsf-consul-register

* 验证无问题后，滚动升级其他节点。

预期结果:





异常处理:

联系运维人员处理。

回滚方案:

恢复程序及启动老服务。

# 注释crontab 任务，crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/log/crontab.log

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-register

# 恢复程序

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0 && rm -rf bin

cp -r /data/tsf-back/tsf-consul-register-v1.23.0/bin /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.26.0/

cd /data/tsf/tsf-consul/tsf-consul-register/tsf-consul-register-v1.23.0/bin && sh start.sh

# 测试该节点是否加入集群

curl http://$registry\_ip:8050/v1/operator/raft/configuration?pretty

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8050/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://$registry\_ip:8050/v1/kv/foo?raw

备注:

无

### 升级tsf-consul-config

命令：

* [生产节点] 流量切换

# ssh consul-config-slave1

# 注释crontab 任务，crontab -e，先取消crontab，以防leader 被自动回切

sed '/config/s/^/#/g' /var/spool/cron/root -i

# crontab -l 确认

# \*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/log/crontab.log

# 确认该10.29.12.51节点是非leader，确认命令

#

curl $registry\_ip:8060/v1/status/leader

curl $registry\_ip:8060/v1/status/peers

curl http://$registry\_ip:8060/v1/operator/raft/configuration?pretty

# 如果升级节点当前为leader，执行切换，将 10.29.12.51 的leader 切换给节点10.29.12.52

curl -v "http://$registry\_ip:8060/v1/operator/raft/transfer?address=10.29.12.52:8064" -X POST

# 切换后确认

curl http://$registry\_ip:8060/v1/operator/raft/configuration?pretty

# 确认该节点为follower后，进行 tsf-consul-config 流量摘除

# slave 节点不提供服务，同如下操作，不需要切换leader

update tsf\_core.consul\_server set used=0 where  ip='$consul-config-ip' and port=8060

# 摘除流量后 等待一分钟观察现象，无问题后执行下一步

# 可以查看摘除节点的log，不会刷新日志；查看leader日志，确保没有异常

tailf /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/log/consul-xxxx.log

* [生产节点] 程序包升级

# 登陆解压

cd /data && unzip tsf-consul-config-v1.29.7.zip

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-config

netstat -anp |grep 8060

# 备份程序

mkdir -p /data/tsf-back

cp -r /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0 /data/tsf-back/tsf-consul-config-v1.26.0\_bak

# 更新程序

cp -rf /data/tsf-consul-config-v1.29.7/bin /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/

# 修改文件夹版本为1.29

mv /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/ /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.29.7/

# 启动进程

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.29.7/bin && sh start.sh

# 查看是否加入集群

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.29.7/bin && sh cluster\_health.sh

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8060/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://$registry\_ip:8060/v1/kv/foo?raw

# 测试通过后，恢复流量

update tsf\_core.consul\_server set used=1 where  ip='$consul-config-ip' and port=8060

# 取消注释crontab 任务，并修改1.23.0 --》1.26.0 crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.29.7/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.29.7/log/crontab.log

:s/tsf-consul-config-v1.29.7/tsf-consul-config-v1.29.7/g

# 检查修改的是否正确

crontab -l |grep tsf-consul-config

* 验证无问题后，滚动升级其他节点。

预期结果:

# 测试该节点是否加入集群

curl http://$registry\_ip:8060/v1/operator/raft/configuration?pretty

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8060/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

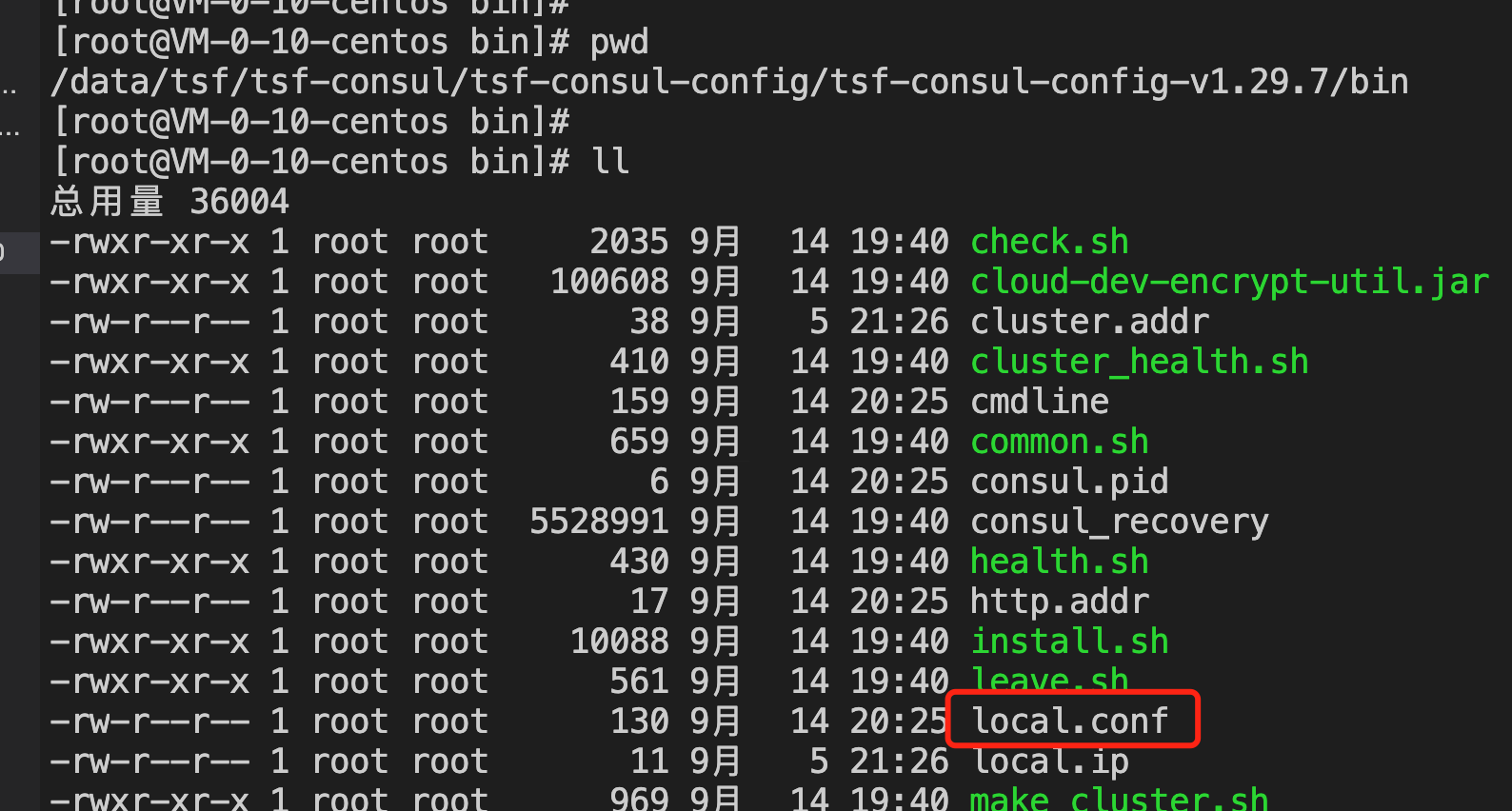
curl http://$registry\_ip:8060/v1/kv/foo?raw

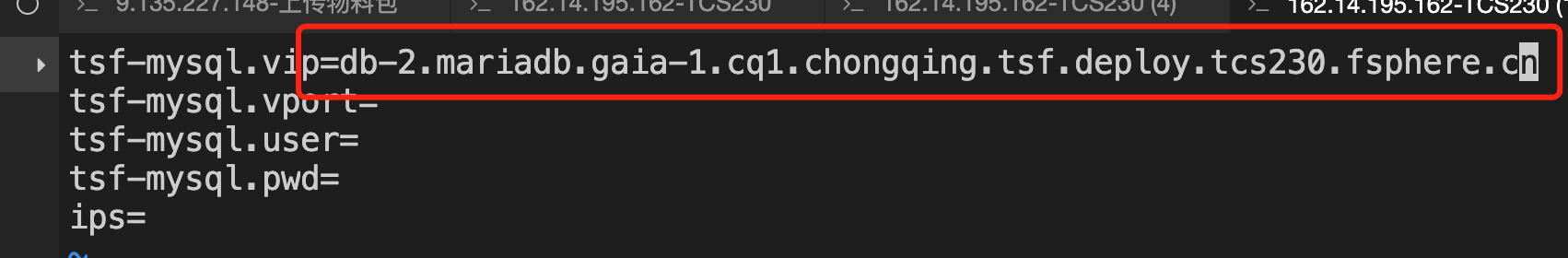
异常处理:

数据库连接异常：



修改local.conf配置文件：





改完之后重启服务，进入bin目录：

sh stop.sh

sh start.sh

联系运维人员处理。

回滚方案:

恢复程序及启动老服务。

# 注释crontab 任务，crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/log/crontab.log

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-config

# 恢复程序

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0 && rm -rf bin

cp -r /data/tsf-back/tsf-consul-config-v1.23.0/bin /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.26.0/bin

cd /data/tsf/tsf-consul/tsf-consul-config/tsf-consul-config-v1.23.0/bin && sh start.sh

# 测试该节点是否加入集群

curl http://10.29.12.51:8060/v1/operator/raft/configuration?pretty

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://10.29.12.51:8060/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://10.29.12.51:8060/v1/kv/foo?raw

### 升级tsf-consul-authen

命令：

* [生产节点] 流量切换

# ssh consul-authen-slave1

# 注释crontab 任务，crontab -e，先取消crontab，以防leader 被自动回切

sed '/authen/s/^/#/g' /var/spool/cron/root -i

# crontab -l 确认

# \*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/log/crontab.log

# 确认该10.29.12.51节点是非leader，确认命令

curl $registry\_ip:8070/v1/status/leader

curl $registry\_ip:8070/v1/status/peers

curl http://$registry\_ip:8070/v1/operator/raft/configuration?pretty

# 如果升级节点当前为leader，执行切换，将 10.29.12.51 的leader 切换给节点10.29.12.52

curl -v "http://$registry\_ip:8070/v1/operator/raft/transfer?address=$new\_leader:8074" -X POST

# 切换后确认

curl http://$registry\_ip:8070/v1/operator/raft/configuration?pretty

# 确认该节点为follower后，进行 tsf-consul-authen 流量摘除

# slave 节点不提供服务，同如下操作，不需要切换leader

update tsf\_core.consul\_server set used=0 where  ip='$consul-authen-ip' and port=8070

# 摘除流量后 等待一分钟观察现象，无问题后执行下一步

# 可以查看摘除节点的log，不会刷新日志；查看leader日志，确保没有异常

tailf /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/log/consul-xxxx.log

* [生产节点] 程序包升级

# 登陆解压

ssh root@$consul\_authen\_ip

cd /data && unzip tsf-consul-authen-v1.29.7.zip

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-authen

netstat -anp |grep 8070

# 备份程序

mkdir -p /data/tsf-back

cp -r /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0 /data/tsf-back/tsf-consul-authen-v1.26.0\_bak

# 更新程序

cp -r /data/tsf-consul-authen-v1.29.7/bin /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/

# 修改文件夹版本为1.29

mv /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/ /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.29.7/

# 启动进程

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.29.7/bin && sh start.sh

# 查看是否加入集群

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.29.7/bin && sh cluster\_health.sh

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8070/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://$registry\_ip:8070/v1/kv/foo?raw

# 测试通过后，恢复流量

update tsf\_core.consul\_server set used=1 where  ip='$consul-authen-ip' and port=8070

# 取消注释crontab 任务，并修改1.23.0 --》1.26.0 crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/log/crontab.log

:s/tsf-consul-authen-v1.23.0/tsf-consul-authen-v1.26.0/g

# 检查修改的是否正确

crontab -l |grep tsf-consul-authen

DOC\_LINK "{\"fileUrl\":\"https://drive.weixin.qq.com/s?k=AJEAIQdfAAodF0vc5pAKwAXQY5ABo\",\"fileId\":\"i.1970325010981265.1688850300231084\_f.663141801t3wD\",\"fileExt\":1000,\"fileName\":\"1.29.6~1.29.7升级指导.md\",\"fileType\":\"Wedrive\"}"1.29.6~1.29.7升级指导.md

* 验证无问题后，滚动升级其他节点。

预期结果:

# 测试该节点是否加入集群

curl http://$registry\_ip:8070/v1/operator/raft/configuration?pretty

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://$registry\_ip:8070/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://$registry\_ip:8070/v1/kv/foo?raw

异常处理:

联系运维人员处理。

回滚方案:

恢复程序及启动老服务。

# 注释crontab 任务，crontab -e

\*/1 \* \* \* \* cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin;./check.sh &> /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/log/crontab.log

# 停止服务

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin && sh stop.sh

# 检查进程或监听端口不存在

ps -ef |grep consul-authen

# 恢复程序

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0 && rm -rf bin

cp -r /data/tsf-back/tsf-consul-authen-v1.23.0/bin /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.26.0/bin

cd /data/tsf/tsf-consul/tsf-consul-authen/tsf-consul-authen-v1.23.0/bin && sh start.sh

# 测试该节点是否加入集群

curl http://10.29.12.51:8070/v1/operator/raft/configuration?pretty

# 测试该节点是否正常

curl --request PUT --data 'aaaaa ggggg' http://10.29.12.51:8070/v1/kv/foo

# 获取数据，预期返回结果：aaaaa ggggg

curl http://10.29.12.51:8070/v1/kv/foo?raw

### 升级tsf-ctsdb

升级步骤（以ctsdb为例子，es也参考此步骤）

**逐个节点滚动升级操作**

1. 替换二进制：

#上传tsf-ctsdb到目标节点如opt目录并解压在当前目录

unzip tsf-ctsdb-1.29.8.zip && cd tsf-ctsdb/ctsdb && unzip elasticsearch.zip

#备份原组件/data/tsf-ctsdb/elasticsearch目录下的lib 、modules、plugins三个目录

mkdir -p /data/tsf-back/ctsdb

cp -r /data/tsf-ctsdb/elasticsearch/lib /data/tsf-back/ctsdb

cp -r /data/tsf-ctsdb/elasticsearch/modules /data/tsf-back/ctsdb

cp -r /data/tsf-ctsdb/elasticsearch/plugins /data/tsf-back/ctsdb

#移除原组件/data/tsf-ctsdb/elasticsearch目录下的lib 、modules、plugins三个目录

rm -rf /data/tsf-ctsdb/elasticsearch/lib

rm -rf /data/tsf-ctsdb/elasticsearch/modules

rm -rf /data/tsf-ctsdb/elasticsearch/plugins

#替换二进制

cd /data/tsf-ctsdb/elasticsearch

cp -ar /opt/tsf-ctsdb/ctsdb/elasticsearch/lib ./

cp -ar /opt/tsf-ctsdb/ctsdb/elasticsearch/modules/ ./

cp -ar /opt/tsf-ctsdb/ctsdb/elasticsearch/plugins/ ./

#修改所有者权限

chown -R c\_log:c\_log /data/tsf-ctsdb/elasticsearch

2）修改配置（此步骤只需ctsdb需要操作）：

vim /data/tsf-ctsdb/elasticsearch/config/elasticsearch.yml

#将cluster\_type改为clog；如cluster\_type: clog

3）重启进程：

cd /$deploy\_dir/common/tsf-storage/tsf-ctsdb/tsf-ctsdb/bin/ && sh stop.sh && sh start.sh

4）等待集群恢复green:

curl x.x.x.x:9201/\_cluster/health?pretty

等待集群恢复green后执行另外一个节点升级操作，剩下节点参考如上步骤

5）执行pipeline

以下脚本只需ctsdb执行:

[ctsdb-add.sh](https://drive.weixin.qq.com/s?k=AJEAIQdfAAolFboWsnAKwAXQY5ABo)

sh ctsdb-add.sh ctsdb\_node\_ip es\_port

[metric\_pipeline.sh](https://drive.weixin.qq.com/s?k=AJEAIQdfAAoZwqtOKDAKwAXQY5ABo)

sh metric\_pipeline.sh ctsdb\_node\_ip ctsdb\_port

[tsf-trace-pipeline.sh](https://drive.weixin.qq.com/s?k=AJEAIQdfAAoY0QuPTlAKwAXQY5ABo)

sh tsf-trace-pipeline.sh ctsdb\_node\_ip ctsdb\_port

### 升级tsf-elasticsearch

升级步骤（以ctsdb为例子，es也参考此步骤）

**逐个节点滚动升级操作**

1. 替换二进制：

#上传tsf-ctsdb到目标节点如opt目录并解压在当前目录

unzip tsf-elasticsearch-1.29.8.zip && cd tsf-elasticsearch/elasticsearch && unzip elasticsearch.zip

#备份原组件/data/tsf-ctsdb/elasticsearch目录下的lib 、modules、plugins三个目录

mkdir -p /data/tsf-back/elasticsearch

cp -r /data/tsf-elasticsearch/elasticsearch/lib /data/tsf-back/elasticsearch

cp -r /data/tsf-elasticsearch/elasticsearch/modules /data/tsf-back/elasticsearch

cp -r /data/tsf-elasticsearch/elasticsearch/plugins /data/tsf-back/elasticsearch

#移除原组件/data/tsf-ctsdb/elasticsearch目录下的lib 、modules、plugins三个目录

rm -rf /data/tsf-elasticsearch/elasticsearch/lib

rm -rf /data/tsf-elasticsearch/elasticsearch/modules

rm -rf /data/tsf-elasticsearch/elasticsearch/plugins

#替换二进制

cd /data/tsf-elasticsearch/elasticsearch

cp -ar /opt/tsf-elasticsearch/elasticsearch/elasticsearch/lib ./

cp -ar /opt/tsf-elasticsearch/elasticsearch/elasticsearch/modules/ ./

cp -ar /opt/tsf-elasticsearch/elasticsearch/elasticsearch/plugins/ ./

#修改所有者权限

chown -R c\_log:c\_log /data/tsf-elasticsearch/elasticsearch

2）修改配置（此步骤只需ctsdb需要操作）：

vim /data/tsf-elasticsearch/elasticsearch/config/elasticsearch.yml

#将cluster\_type改为clog；如cluster\_type: clog

3）重启进程：

cd /$deploy\_dir/common/tsf-storage/tsf-elasticsearch/tsf-elasticsearch/bin/ && sh stop.sh && sh start.sh

4）等待集群恢复green:

curl x.x.x.x:9200/\_cluster/health?pretty

等待集群恢复green后执行另外一个节点升级操作，剩下节点参考如上步骤

5）执行pipeline

以下脚本只需es执行:

[elasticsearch-add.sh](https://drive.weixin.qq.com/s?k=AJEAIQdfAAoKuHz88aAKwAXQY5ABo)

sh elasticsearch-init.sh es\_node\_ip es\_port

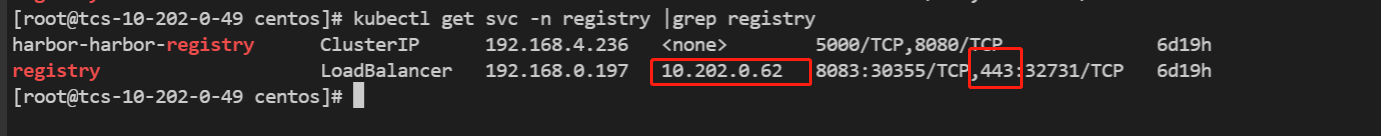
[tsf-oss-pipeline.sh](https://drive.weixin.qq.com/s?k=AJEAIQdfAAoW7mfvGtAKwAXQY5ABo)

sh tsf-oss-pipeline.sh es\_node\_ip es\_port

### TSF镜像迁移

#查询harbor登录地址

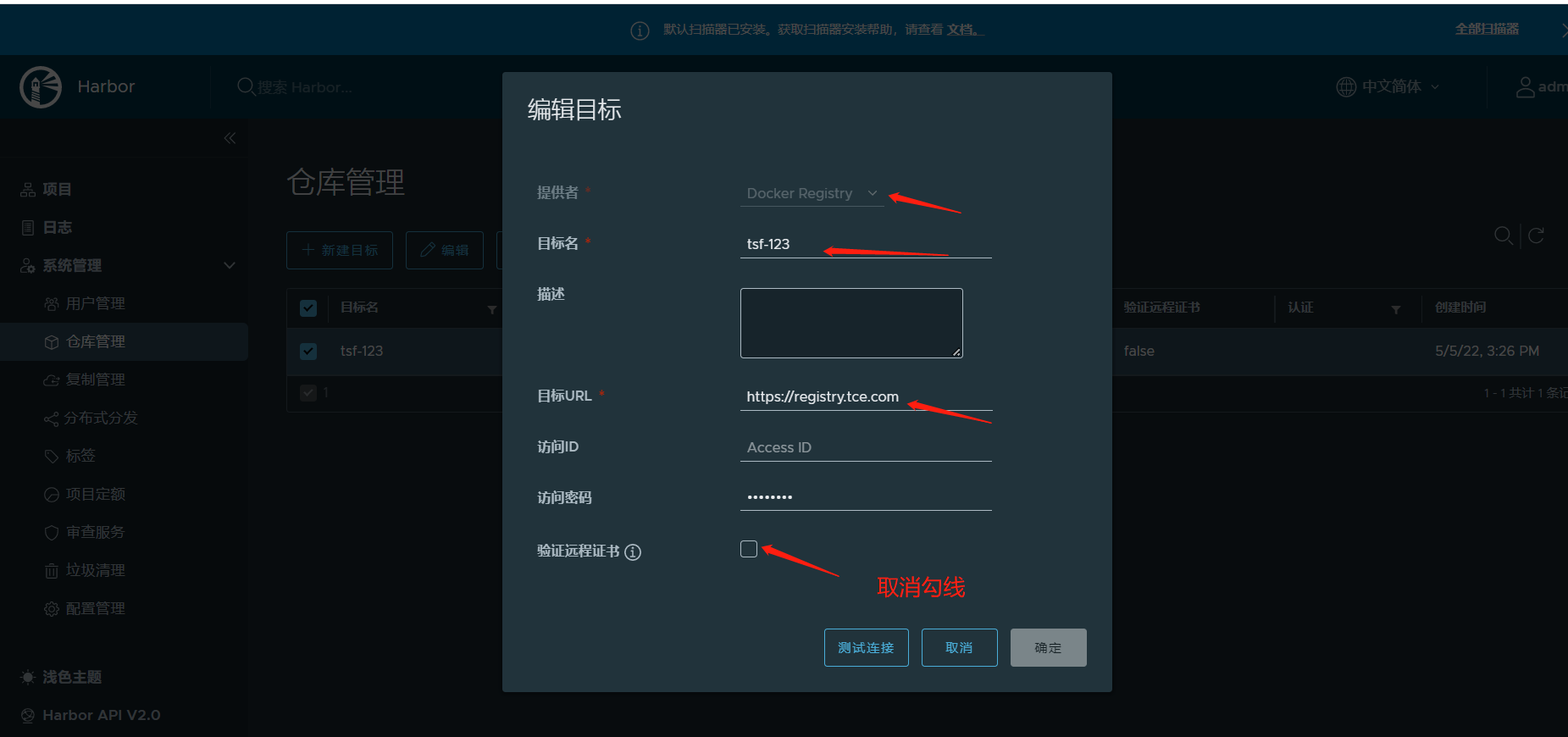
kubectl get svc -n registry |grep registry



#### 登录harbor管理界面



新建一个仓库，目标名任意，其他信息如图所示

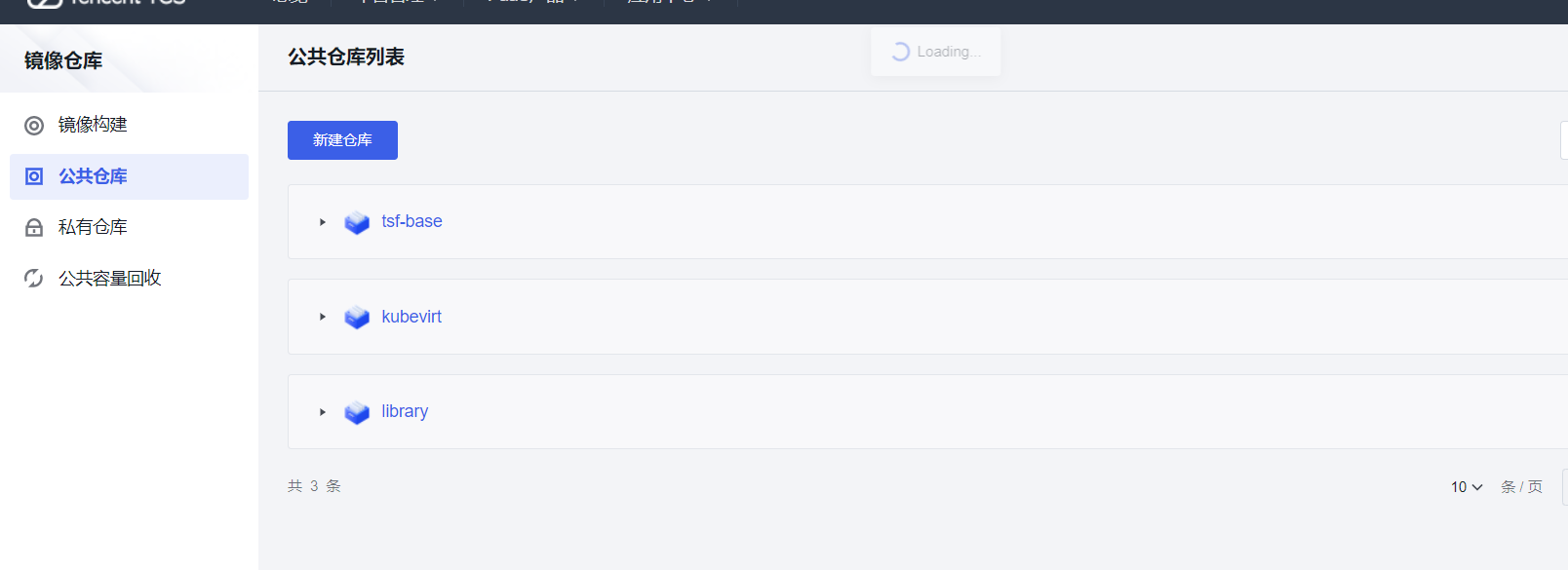


#### TCS镜像仓库进行仓库创建

#新建私有仓库 tsf-909619400



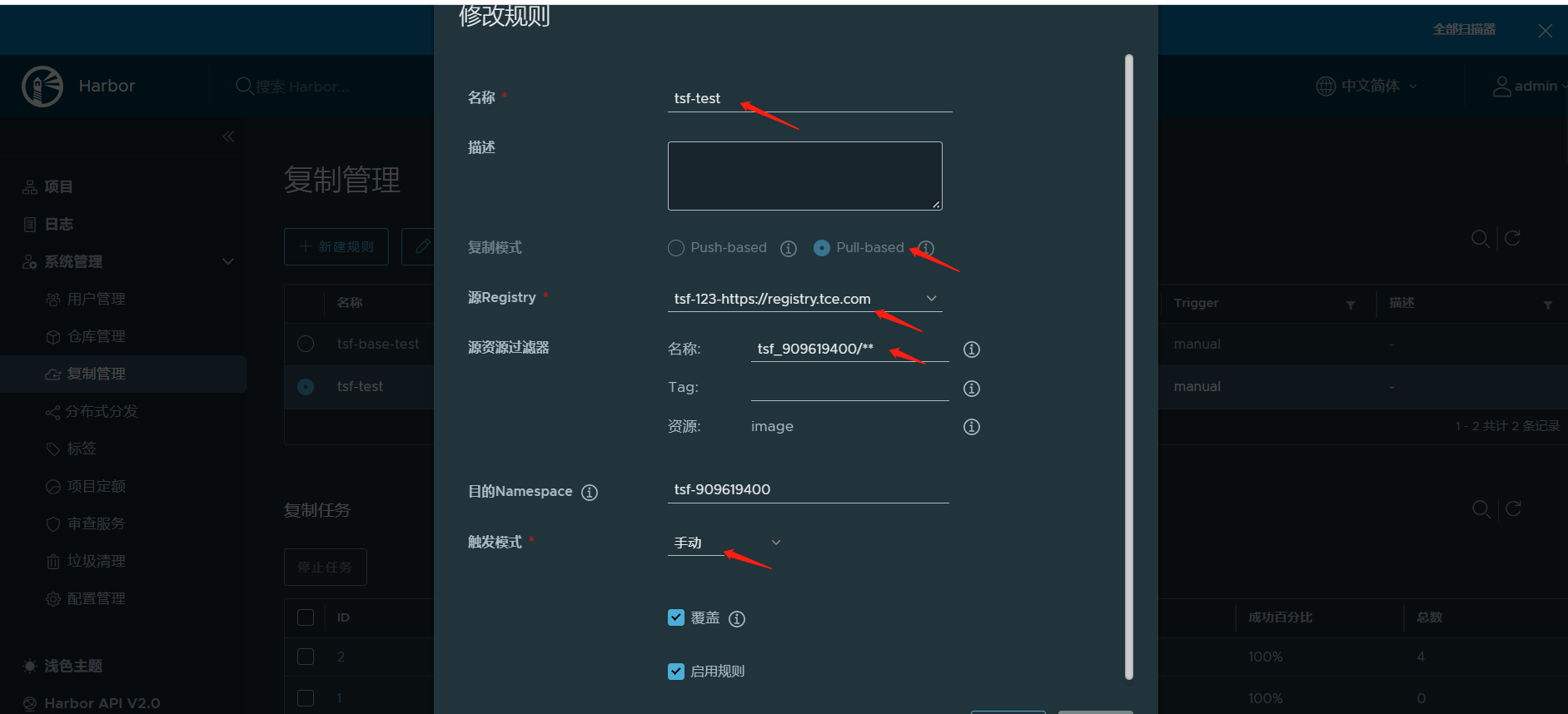
#新建公共仓库 tsf-base



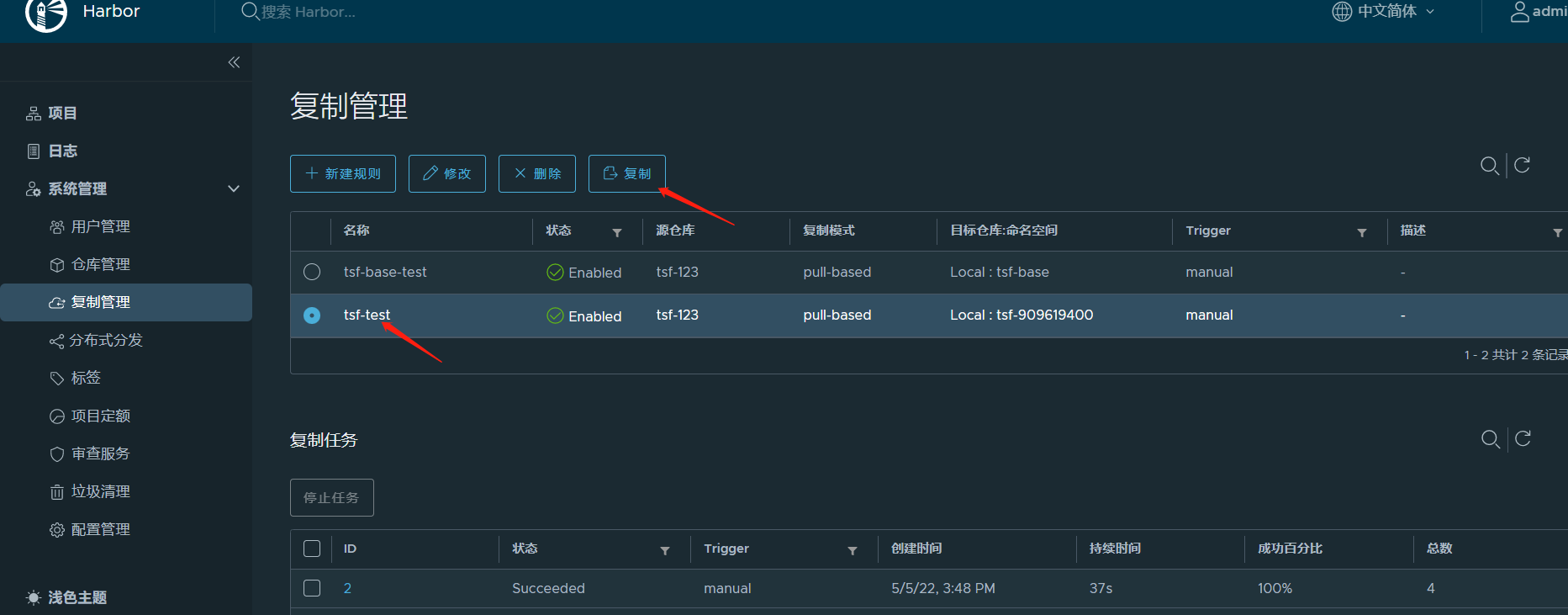
# harbor镜像仓库创建复制规则



# 新建规则填写信息如下：



#选择创建的规则进行复制/迁移

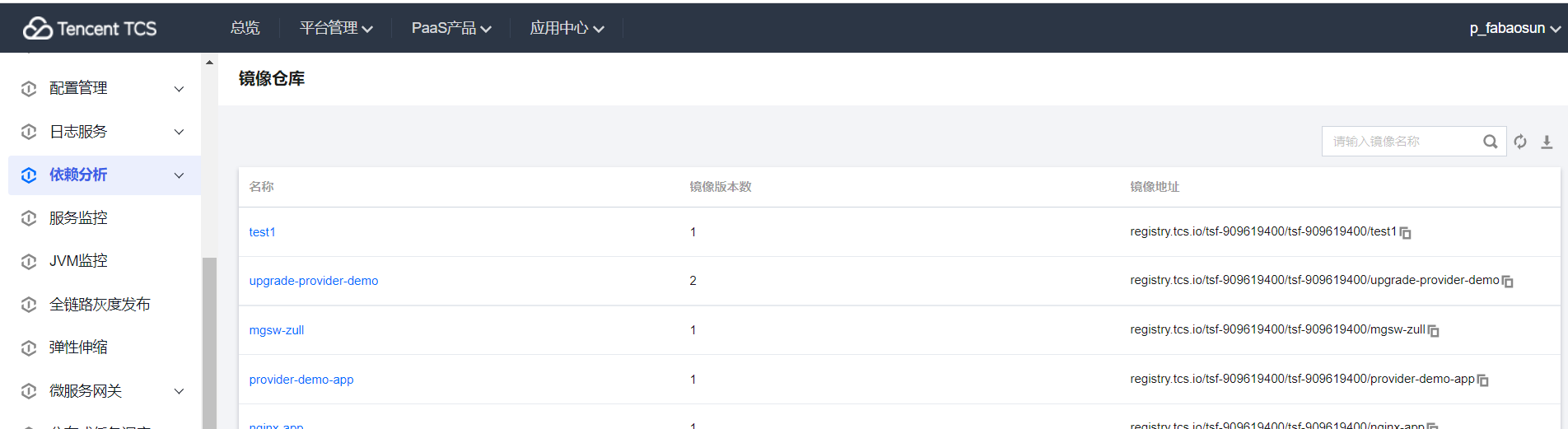


预期结果：

#tcs镜像仓库与tsf镜像仓库即可查询到镜像







### 更新容器镜像

命令：

# 使用TKE 可以上传公共镜像的underlay 机器操作

docker login -u tceadmin -paaaaaaaa1！

# 拷贝tcs 上 agent-1.21.1.tar.gz、istio-proxy-image-20200508\_17.tar.gz 到导入镜像仓库

cd /data/tce\_dc/workspace/product-tsf-mock/software/

docker load -i agent-image-1.29.4.tar.gz

docker load -i istio\_proxy\_1.14.8.1.tar.gz

docker tag <agent-image-id> registry.tcs.io/tsf-base/agent:1.29

docker tag <istio-proxy-image-id> registry.tcs.io/tsf-base/istio\_proxy:1.29

docker push registry.tcs.io/tsf-base/agent:1.29

docker push registry.tcs.io/tsf-base/istio\_proxy:1.29

# 更新系统参数 （此处可在运营端查看，若参数配置中已经修改，不需要操作此步骤）

use tsfmanager\_operation

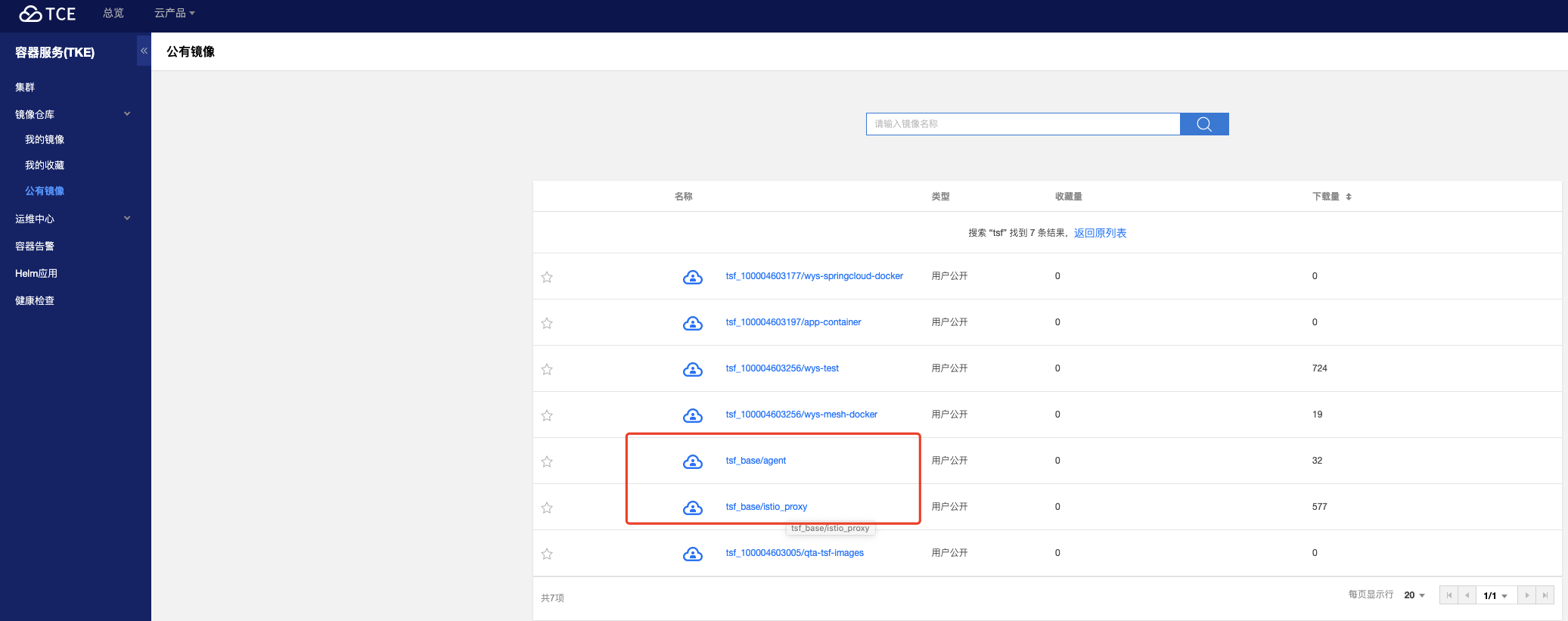
select \* from tsf\_parameter\_definition where parameter\_name in ('tsf.container.image.agent','tsf.container.image.istio-proxy');

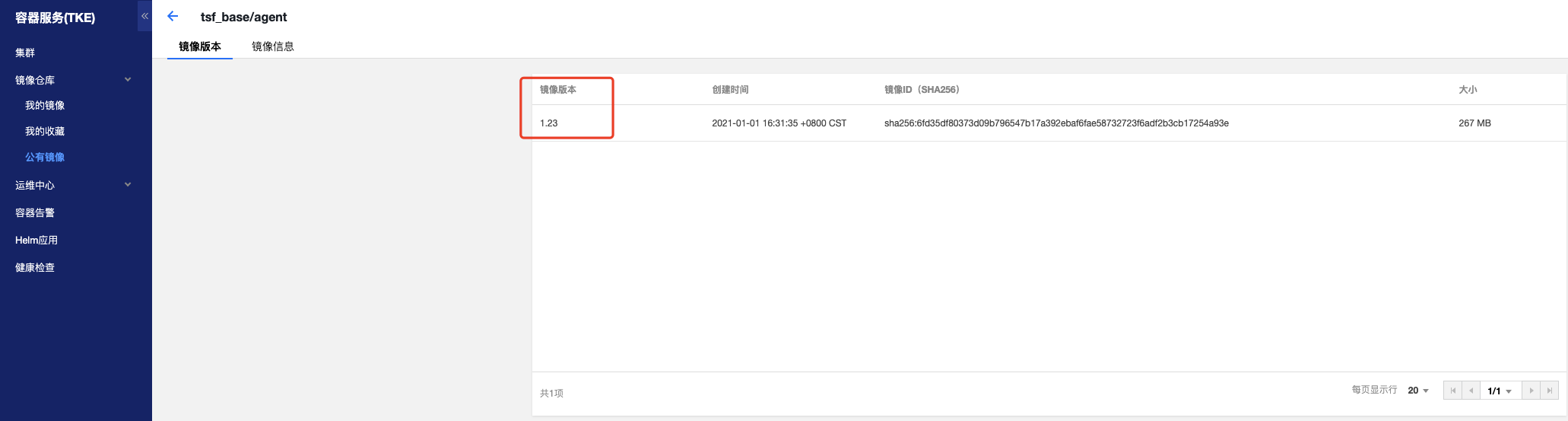
update tsf\_parameter\_definition set parameter\_value = 'tsf\_base/agent:1.29' where parameter\_name in ('tsf.container.image.agent');

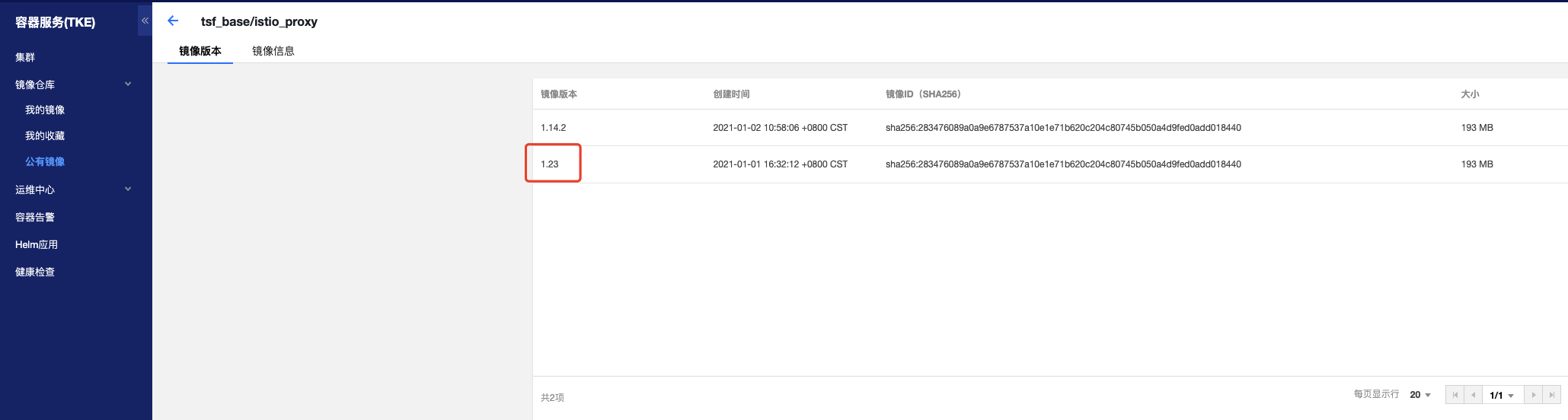
update tsf\_parameter\_definition set parameter\_value = 'tsf\_base/istio\_proxy:1.29' where parameter\_name in ('tsf.container.image.istio-proxy');

预期结果:

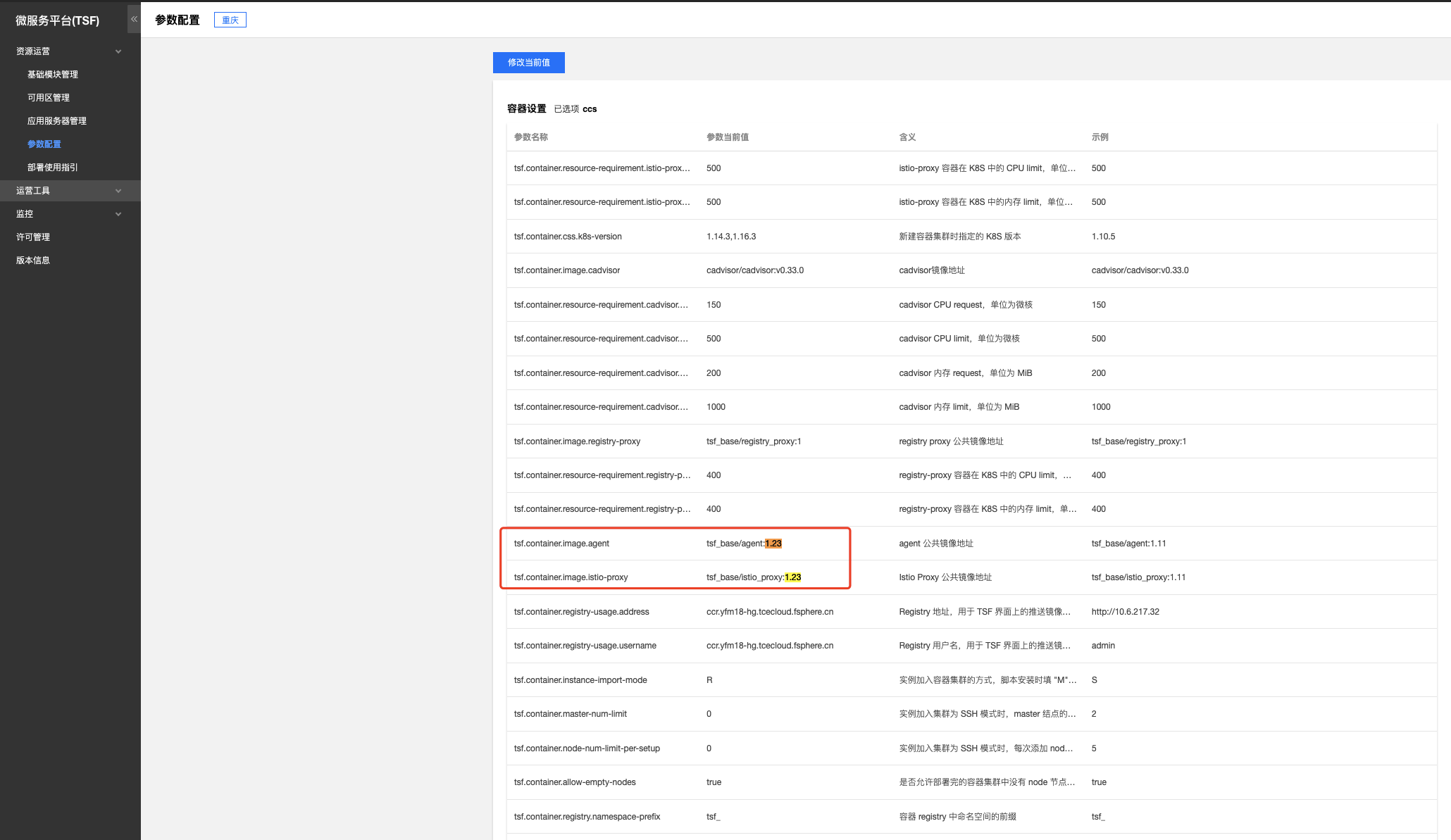
租户端--镜像仓库--公共镜像，查看tsf 如下镜像版本是否有1.29版本







运营端，系统参数检查：



回滚方案:

# 更新系统参数

use tsfmanager\_operation

update tsf\_parameter\_definition set parameter\_value = 'tsf\_base/agent:1.23' where parameter\_name in ('tsf.container.image.agent');

update tsf\_parameter\_definition set parameter\_value = 'tsf\_base/istio\_proxy:1.23' where parameter\_name in ('tsf.container.image.istio-proxy');

### 升级license-server

license-server pod 拉起后，pod内会因为没有授权而租户端偶尔报错，影响服务发布，不影响存量服务调用。（不需要重新导入license）

命令：

# 拉起license-server 组件镜像

kaleido\_image upgrade --yaml /data/tce\_dc/software/latest/yaml/ocloud-tsf-license-server.1.0.1-xxx.tgz

-- 查询删除cvm license-server 节点，不影响租户端使用

use tsfmanager\_operation;

delete from tsf\_module\_instance where module\_id = (select distinct module\_id from tsf\_module where module\_name = 'license-server');

# 申请授权

1、 运营端--基础模块管理，license-server 节点管理，查看应该包括cvm 节点 + pod 节点。（如果拉起pod 后仅包括cvm节点，先界面移除后，刷新则会从consul获取到注册成功的license-server 节点，含cvm 节点 + pod 节点）

2、运营端--许可管理，申请授权（确保运营端可以看到cvm 和 pod 的license-server 节点，再申请）。

3、运营端，导入授权。

4、导入license 授权后，所有license pod内执行确认

curl -i -m 3 -k 'https://localhost:7100/license/list'





预期结果:

# consul

# 导入license 授权后，所有license pod内执行确认

curl -i -m 3 -k 'https://localhost:7100/license/list'

# 1是等待生效、2是已生效、3是已过期、4是已作废、5是无效。

# 运营端和租户端测试，无缺失授权的报错

回滚方案:

kubectl get deployment -ntce |grep ocloud-tsf-license-server

kubectl scale -ntce deployment/ocloud-tsf-license-server --replicas=0

### 组件检查确认

命令：

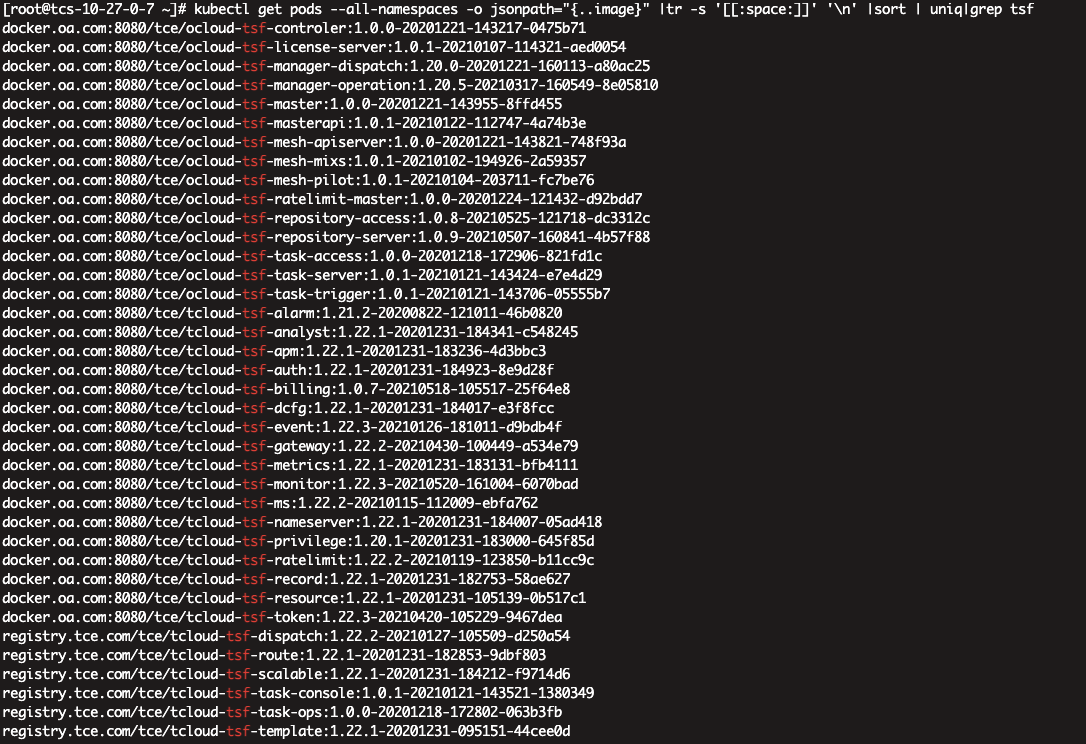
# 预期38个pod 组件，去重多副本

kubectl get pods --all-namespaces -o jsonpath="{..image}" |tr -s '[[:space:]]' '\n' |sort | uniq|grep tsf |wc

# 同时确认镜像版本是否更新为2.1.0

kubectl get pods --all-namespaces -o jsonpath="{..image}" |tr -s '[[:space:]]' '\n' |sort | uniq|grep tsf

预期结果:



### 查看组件是否更新成功：

#执行对比脚本

cd /data/upgrade\_script && bash upgradeCheck paas apigw

#脚本执行完成后会在该目录生成一个tcs-apigw.html的文件

ls /data/backup/pod-image-now/

### 验证阶段

[管控面] 登录验证

无。

[管控面] 管控功能验证

运营端、租户端各菜单功能验证。

[数据面] 运行状态验证

按照前期准备，创建demo测试服务验证所有功能。