访问管理

产品文档

目录

运维管理指南
产品架构
核心功能组件
基础平台

组件ID	组件功能
tcloud-tcenter-project-org	项目管理/资源管理
ocloud-tcenter-location	平台部署信息服务

用户与权限

组件ID	组件功能
ocloud-tcenter-identityaccess	核心组件，负责账号读写，登录，校验，mfa认证
ocloud-tcenter-identityccdb	负责账号同步
ocloud-tcenter-open-identity	负责企业认证登录
ocloud-tcenter-mc-cas-portal	负责运营端cas登录

访问管理

组件ID	组件功能
tcloud-tcenter-cam	租户端cam
ocloud-tcenter-cam	运营端cam
product-tcenter-support-cam	虚拟机独立CAM（非必选）
tcloud-tcenter-support-cam	容器化独立CAM（非必选）

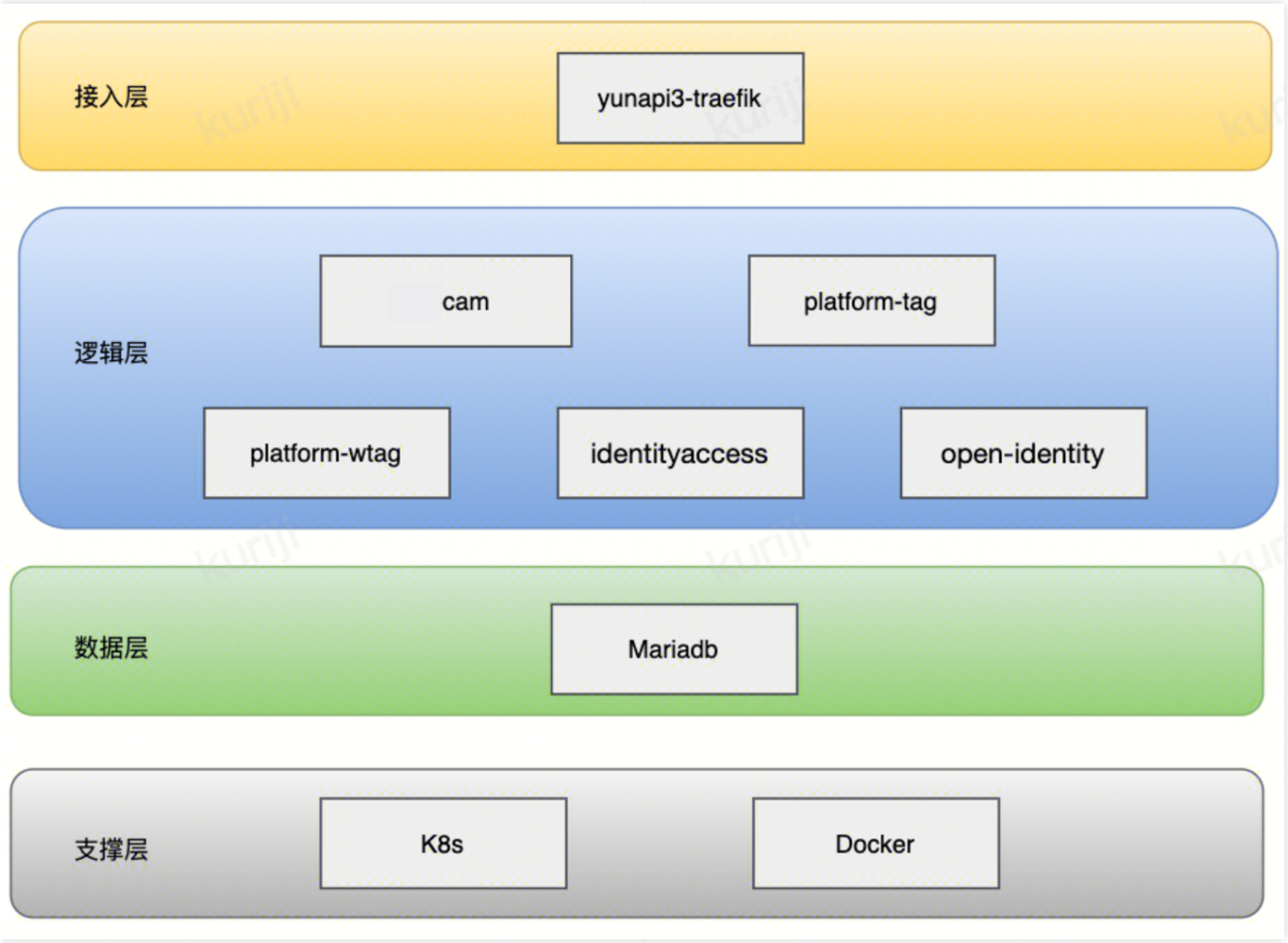
云API管理

组件ID	组件功能
ocloud-tcenter-yunapi3-traefik	云 API 运营端网关服务
tcloud-tcenter-yunapi3-traefik	云 API 租户端端网关服务
ocloud-tcenter-capi-internal	内部服务调用的网关服务
ocloud-tcenter-yunapi3-yuntu	负责运营端 云 API 管理
tcloud-tcenter-yunapi3-tyuntu	负责租户端云 API 管理
image-preset-yunapi	云 API预设数据导入导出工具apitools
ocloud-tcenter-location	负责管理平台产品部署信息

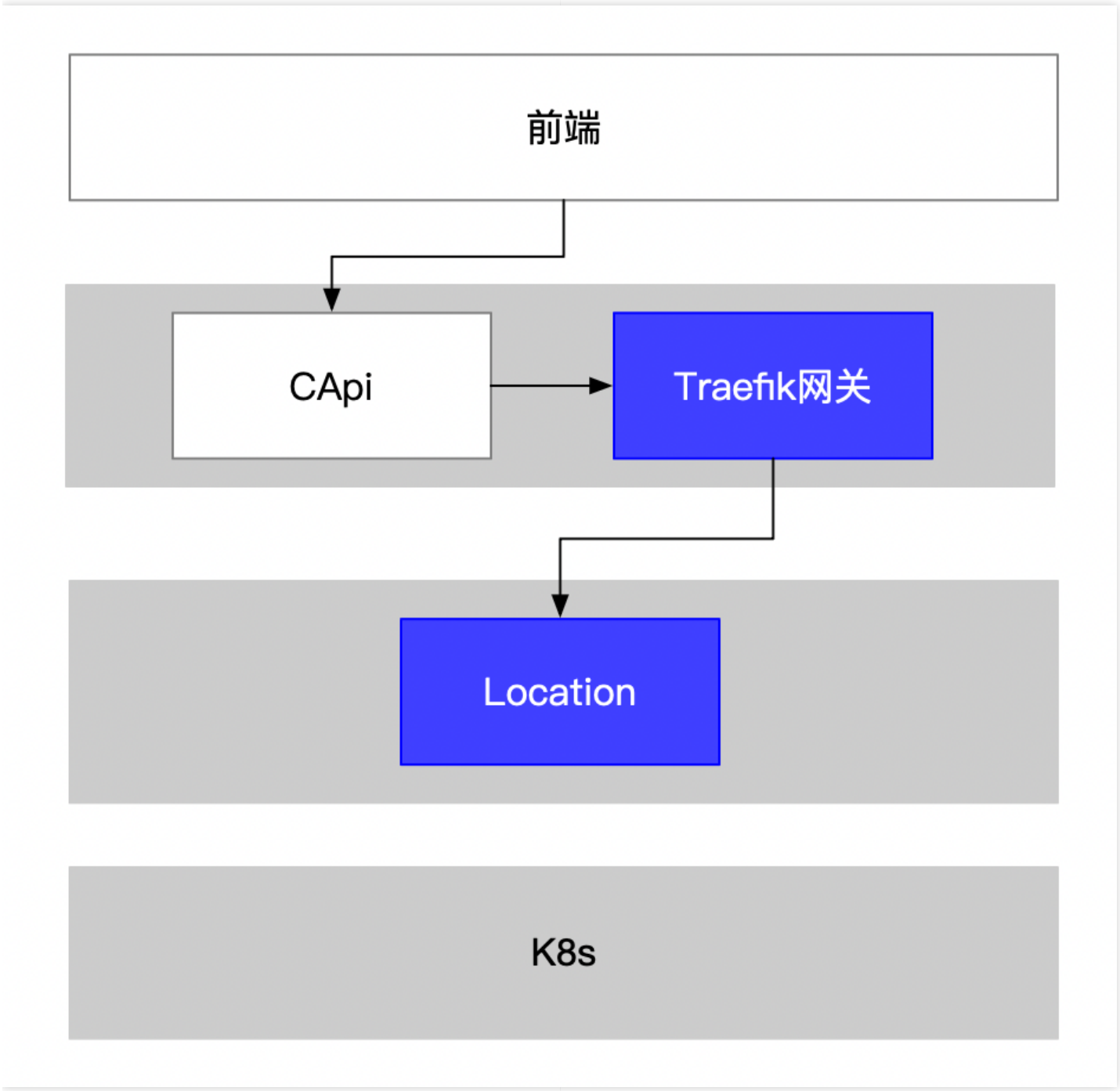
License管理

组件ID	组件功能
ocloud-tcenter-license	License后端服务组件
dbsql-tcenter-license	License服务数据库组件
ocloud-tcenter-license-crd	LicenseClient CRD注册组件

产品架构图
基础平台

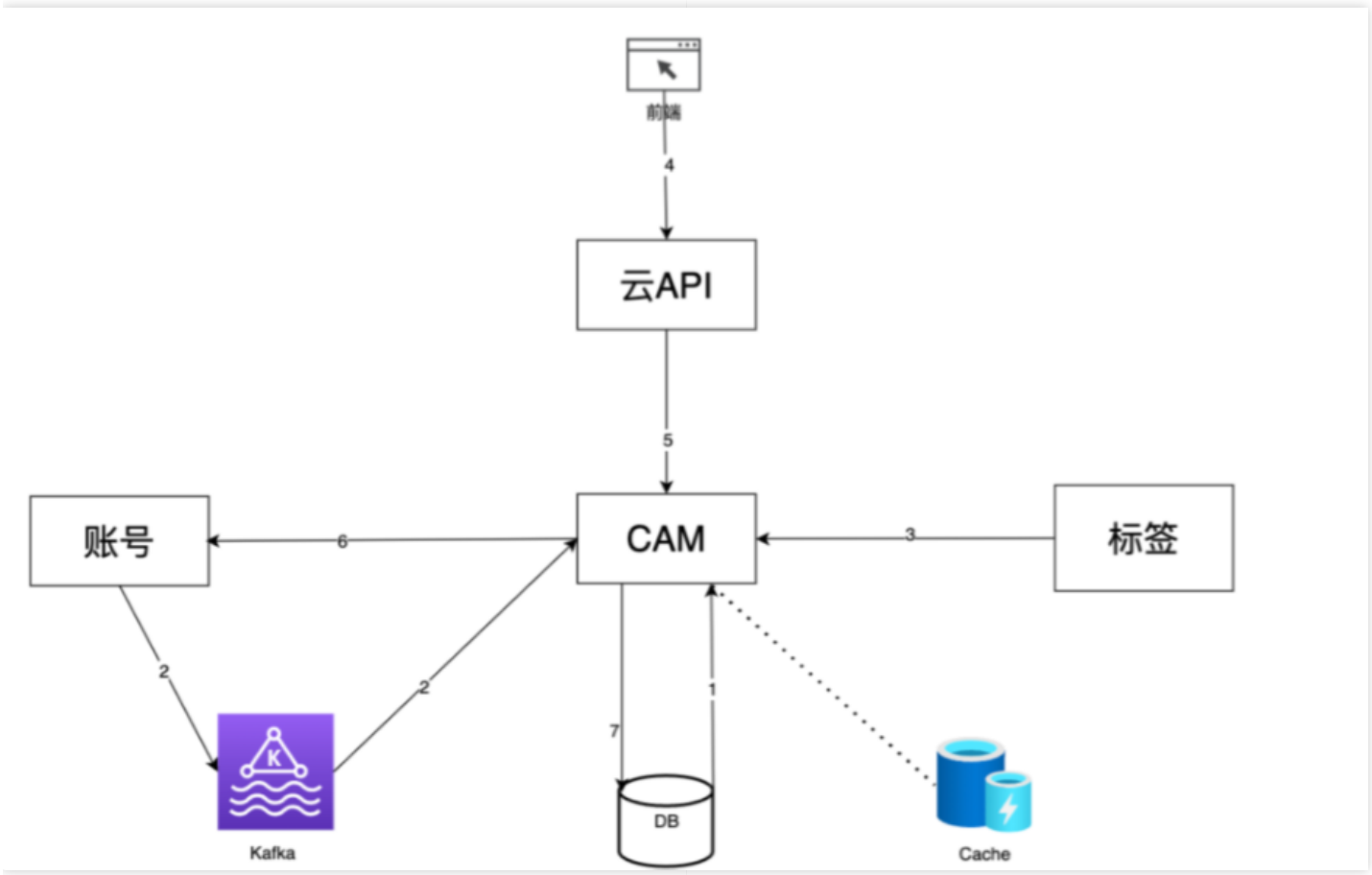


location 架构

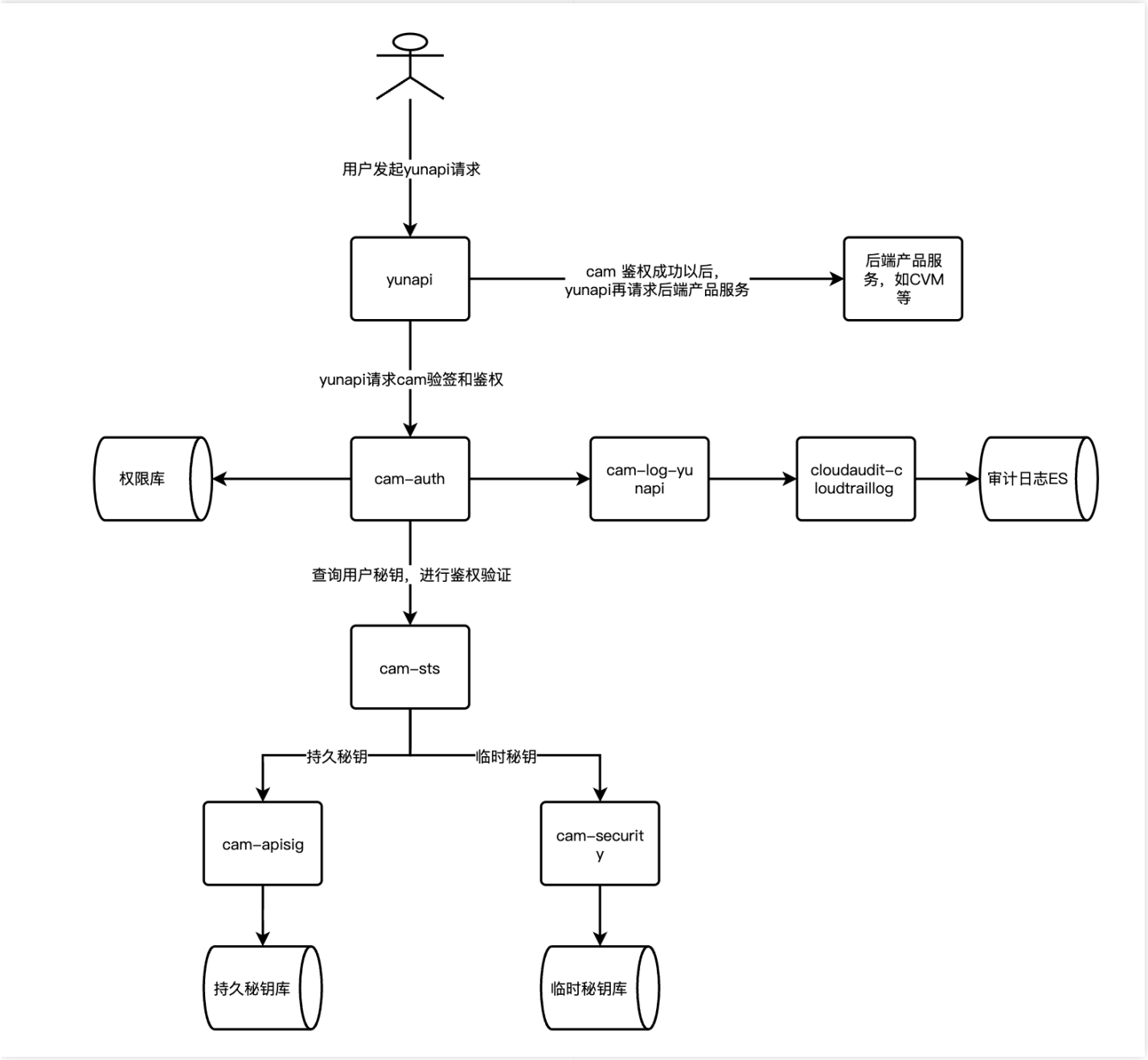

前端
CApi
Traefik网关
Location
K8s

访问管理

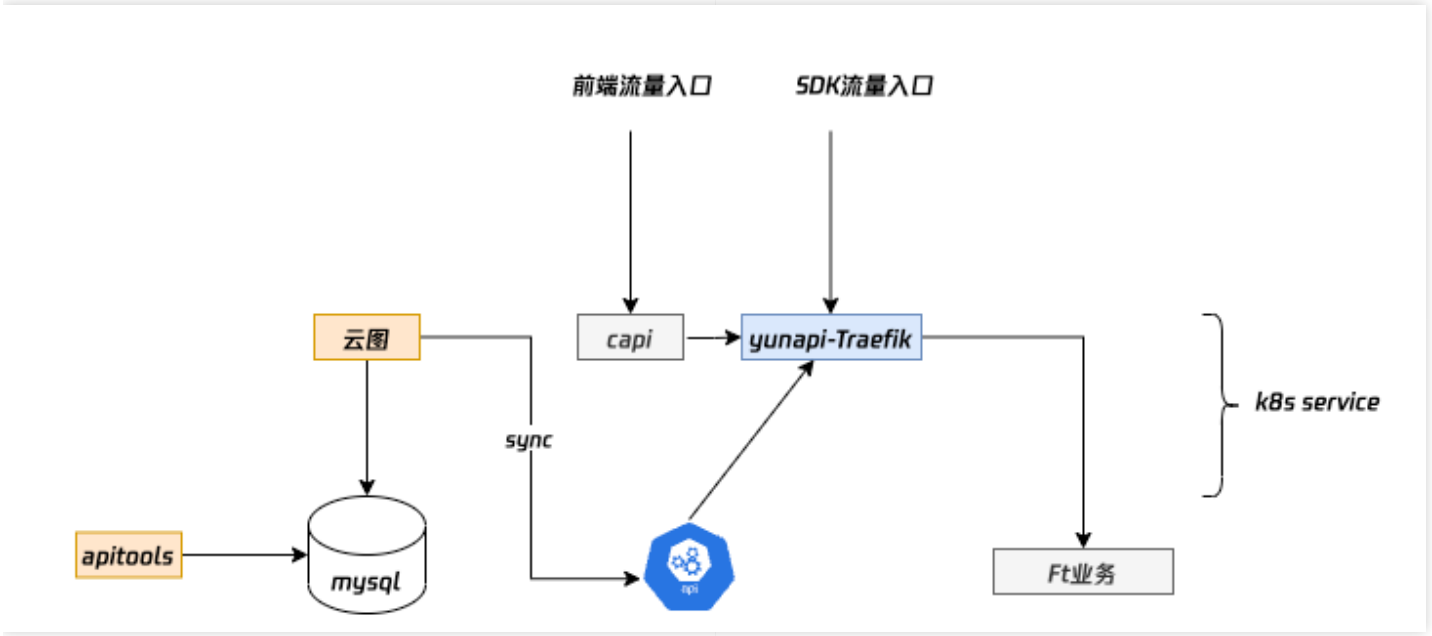
CAM主要依赖的服务包括账号、标签、TLSQL。



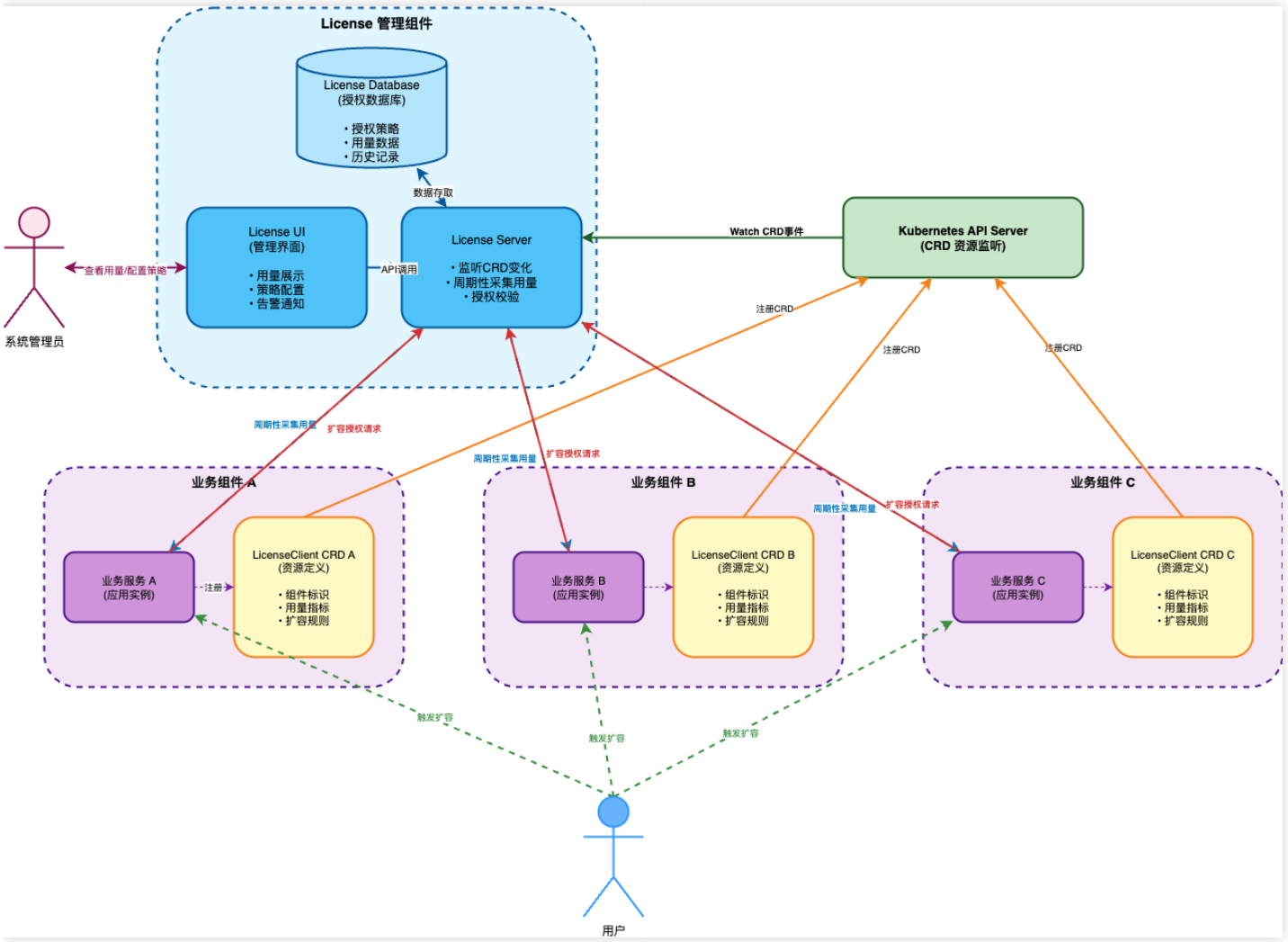
用户的鉴权请求会被云API转发到CAM，CAM查询用户密钥，验证鉴权。


用户发起yunapi请求
yunapi
cam 鉴权成功以后，
yunapi再请求后端产品服务
后端产品服务，如CVM等
yunapi请求cam验签和鉴权
权限库
cam–auth
cam–log–yunapi
cloudaudit–cloudtraillog
审计日志ES
查询用户秘钥，进行鉴权验证
cam–sts
持久秘钥
临时秘钥
cam–apisig
cam–security
持久秘钥库
临时秘钥库

云API管理


前端流量入口
SDK流量入口
云图
capi
yunapi-Traefik
k8s service
sync
apitools
mysql
Ft业务

License管理

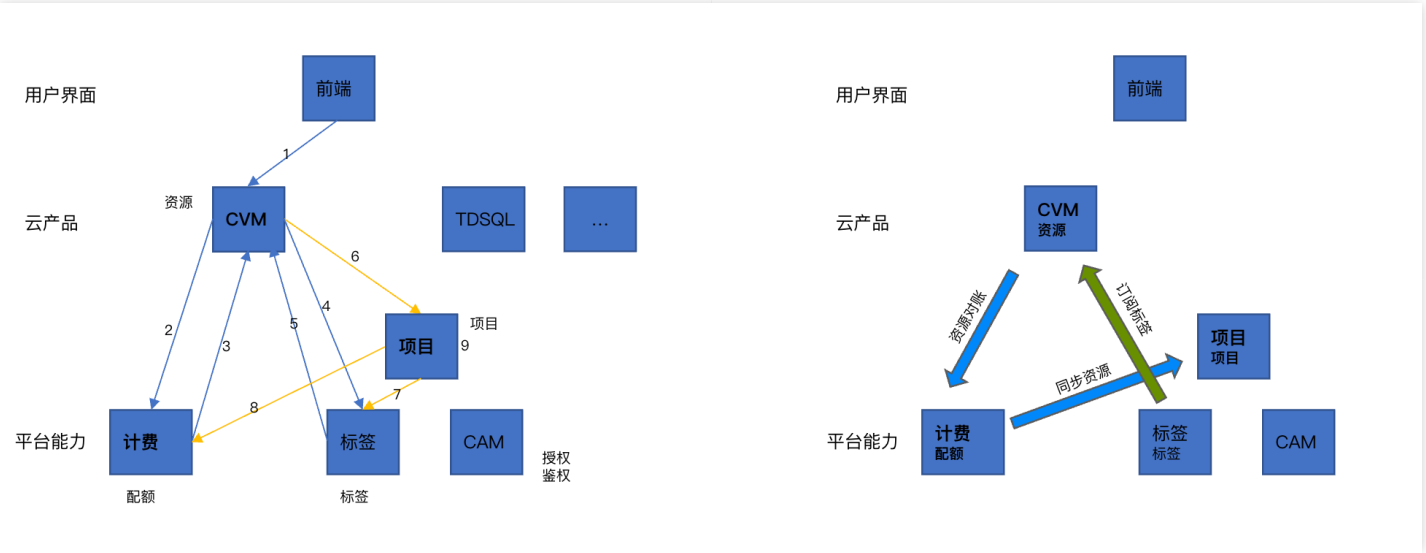

License 管理组件
License Database
(授权数据库)
• 授权策略
• 用量数据
• 历史记录
数据存取
License UI
(管理界面)
• 用量展示
• 策略配置
• 告警通知
API调用
License Server
• 监听CRD变化
• 周期性采集用量
• 授权校验
Watch CRD事件
Kubernetes API Server
(CRD 资源监听)
系统管理员
查看用量/配置策略
注册CRD
注册CRD
注册CRD
周期性采集用量
扩容授权请求
周期性采集用量
扩容授权请求
周期性采集用量
扩容授权请求
业务组件 A
业务组件 B
业务组件 C
业务服务 A
(应用实例)
注册
LicenseClient CRD A
(资源定义)
• 组件标识
• 用量指标
• 扩容规则
业务服务 B
(应用实例)
LicenseClient CRD B
(资源定义)
• 组件标识
• 用量指标
• 扩容规则
业务服务 C
(应用实例)
LicenseClient CRD C
(资源定义)
• 组件标识
• 用量指标
• 扩容规则
触发扩容
触发扩容
触发扩容
用户

部署架构图
访问管理

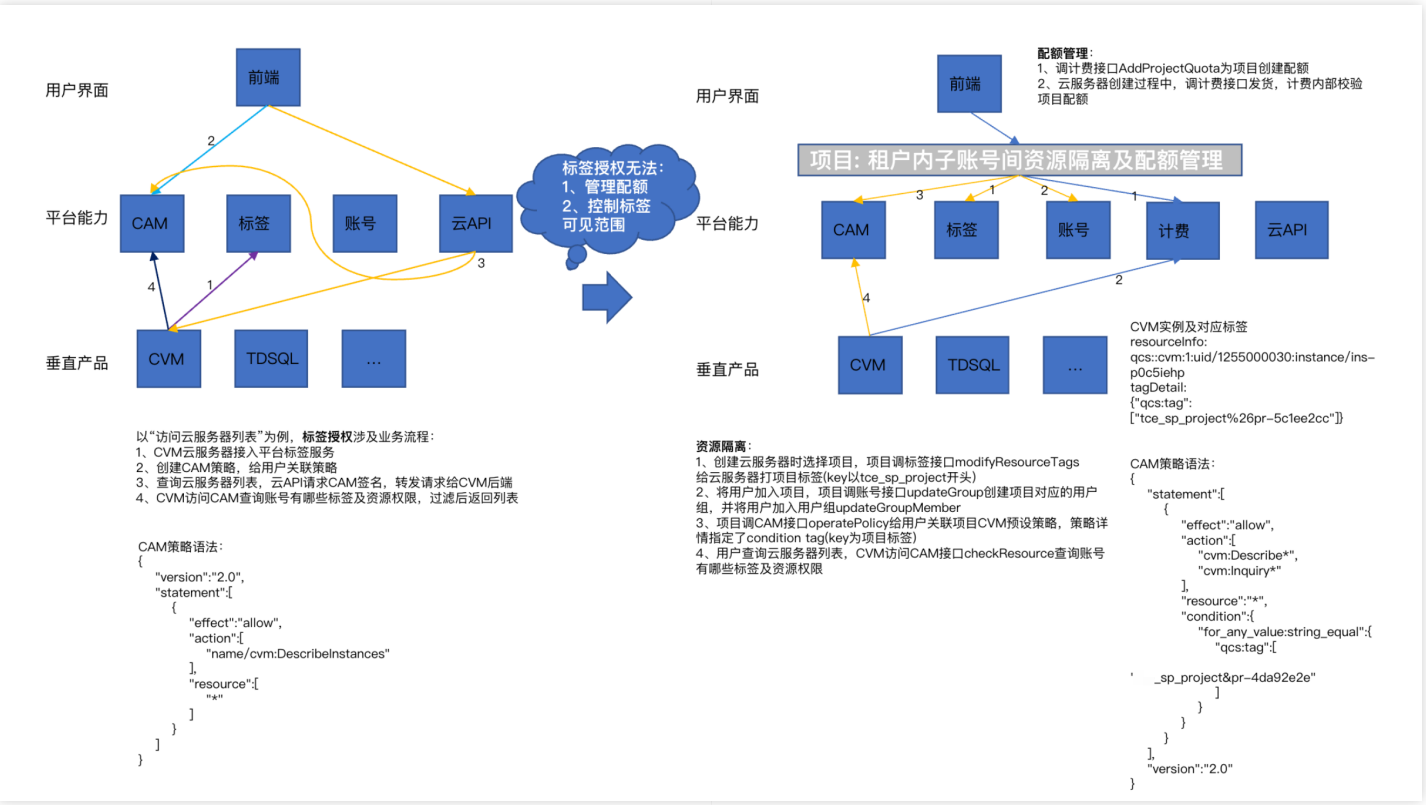
CAM 服务为 global 级别无状态服务（独立版CAM（组件名product-tcenter-support-cam，tcloud-tcenter-support-cam）为region级别服务），存储依赖支撑数据库，可均匀部署在所有地域所有可用区，所有组件可水平扩缩容。

业务流向图
基础平台

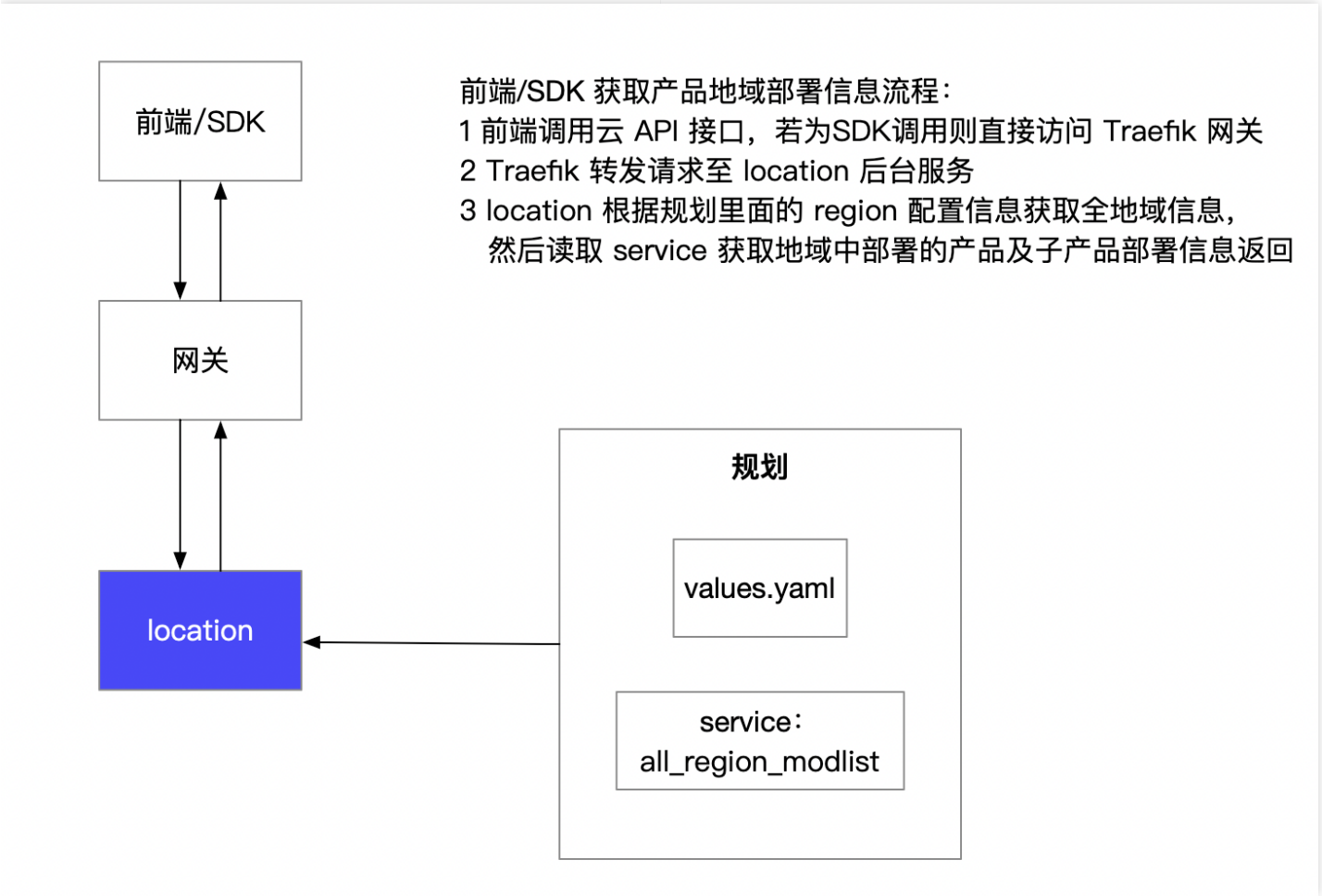
项目主要和计费、CAM、各类接入资源产生交互，基本业务流向如图：



以CVM为例，说明项目如何实现资源隔离：

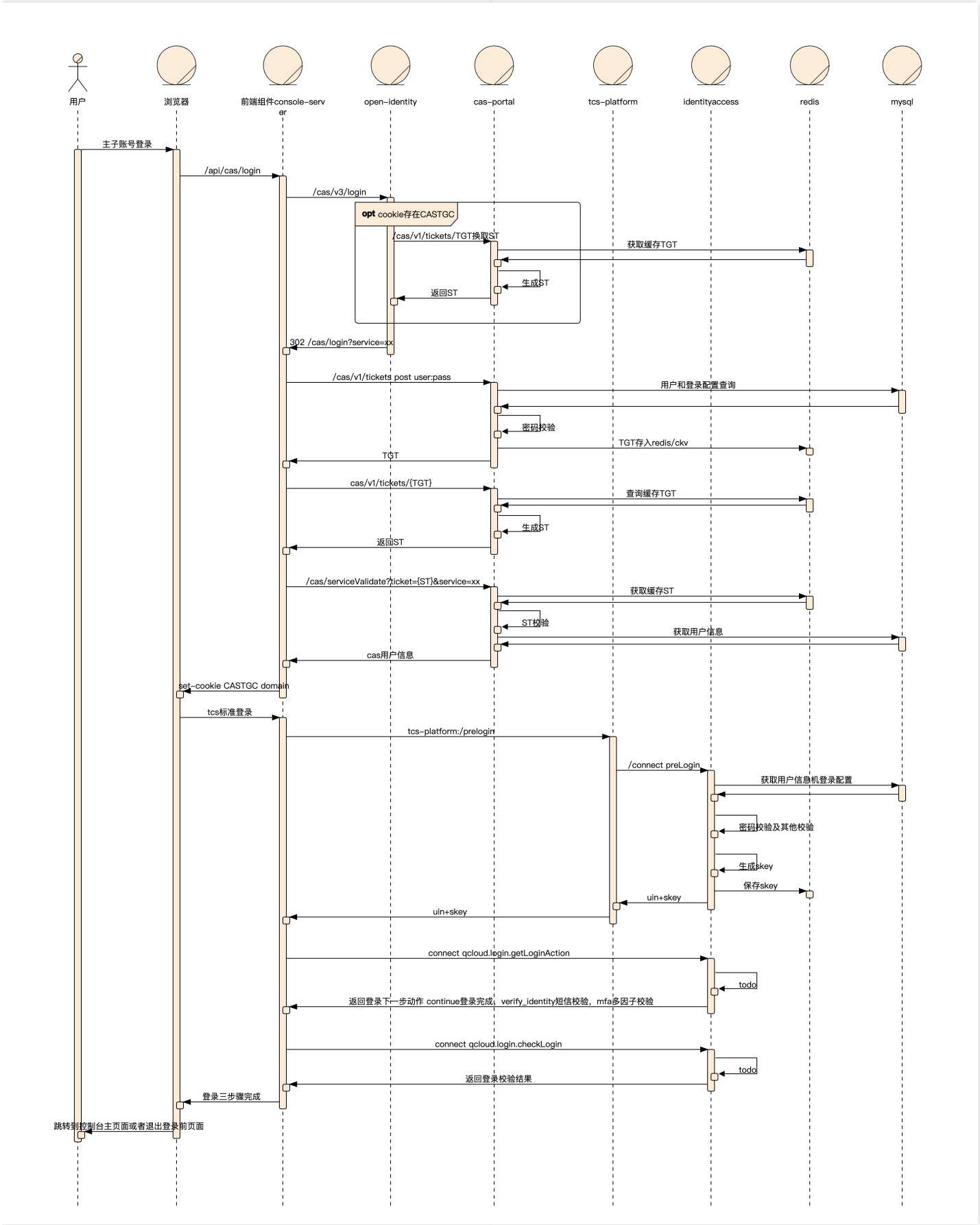


用户查询 location 服务获取产品地域部署信息流程：

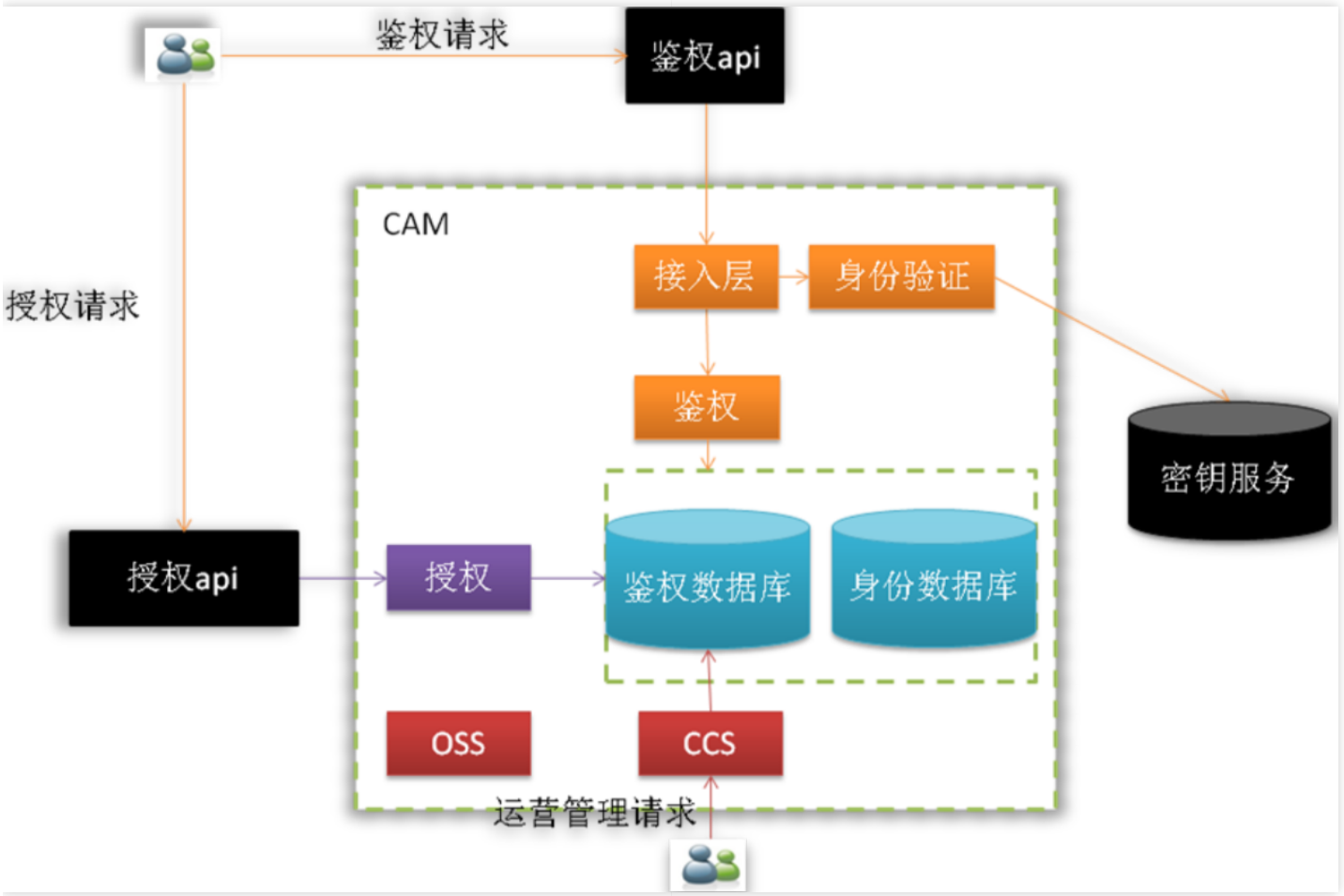

前端/SDK
网关
location
规划
values.yaml
service：
all_region_modlist
前端/SDK 获取产品地域部署信息流程：
1 前端调用云 API 接口，若为SDK调用则直接访问 Traefik 网关
2 Traefik 转发请求至 location 后台服务
3 location 根据规划里面的 region 配置信息获取全地域信息，
然后读取 service 获取地域中部署的产品及子产品部署信息返回

用户与权限

运营端核心登录链路如下：


用户
浏览器
前端组件console-server
open-identity
cas-portal
tcs-platform
identityaccess
redis
mysql
主子账号登录
/api/cas/login
/cas/v3/login
opt cookie存在CASTGC
/cas/v1/tickets/TGT换取ST
获取缓存TGT
生成ST
返回ST
302 /cas/login?service=xx
/cas/v1/tickets post user:pass
用户和登录配置查询
密码校验
TGT存入redis/ckv
TGT
cas/v1/tickets/{TGT}
查询缓存TGT
生成ST
返回ST
/cas/serviceValidate?ticket={ST}&service=xx
获取缓存ST
ST校验
获取用户信息
cas用户信息
set-cookie CASTGC domain
tcs标准登录
tcs-platform:/prelogin
/connect preLogin
获取用户信息机登录配置
密码校验及其他校验
生成skey
保存skey
uin+skey
uin+skey
connect qcloud.login.getLoginAction
todo
返回登录下一步动作 continue登录完成，verify_identity短信校验，mfa多因子校验
connect qcloud.login.checkLogin
todo
返回登录校验结果
登录三步骤完成
跳转到控制台主页面或者退出登录前页面

访问管理

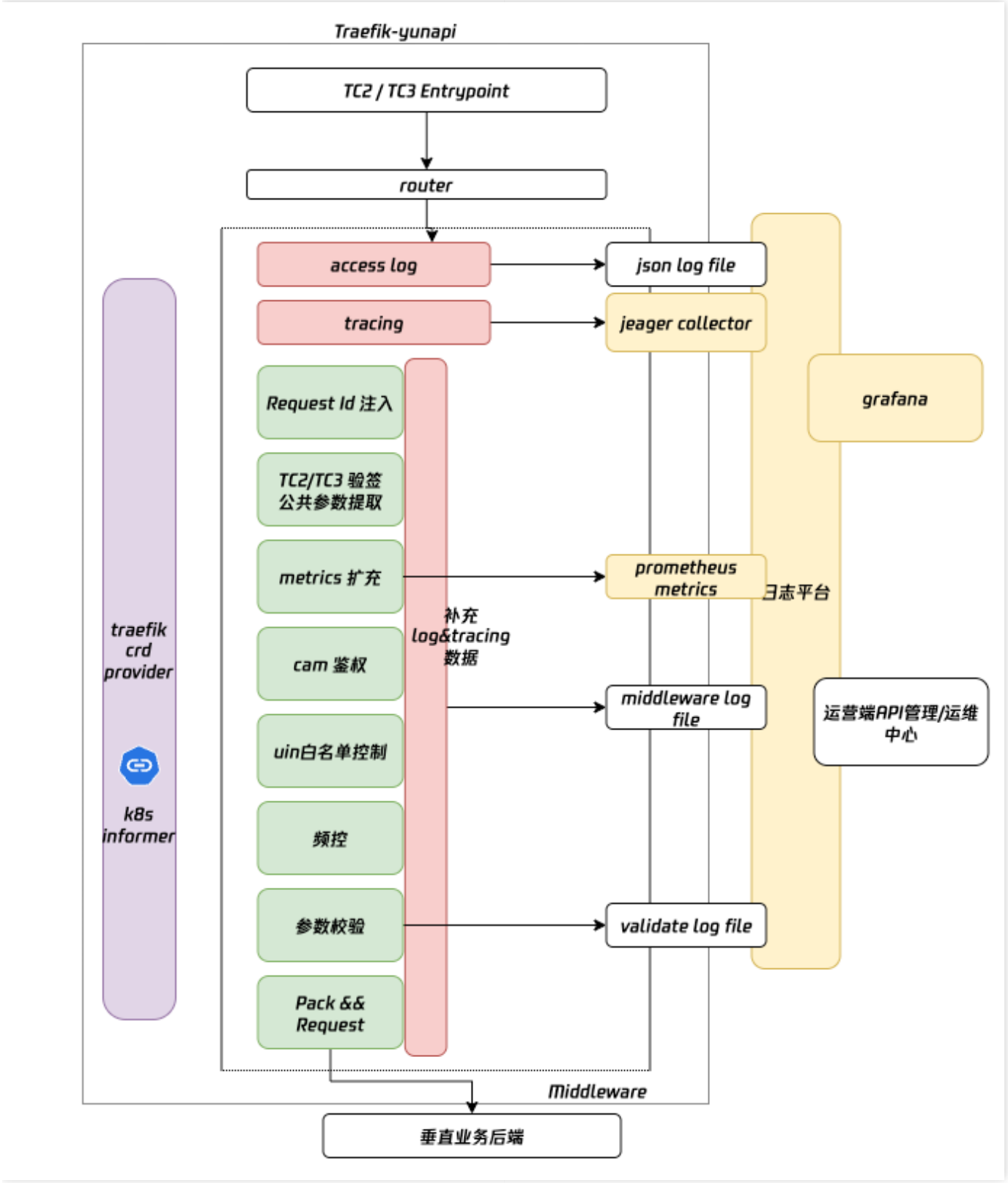


鉴权操作分为两个部分：

- 一是身份认证：通过密钥可以保证请求在中间过程中不被篡改；请求的URL包含密钥ID、签名KEY等，在服务端或根据用户的密钥ID获取用户的密钥Key，重新计算签名，判断两个签名一致则请求是合法的。
- 二是权限控制：权限可以到资源级别，也可以精确到接口级别。

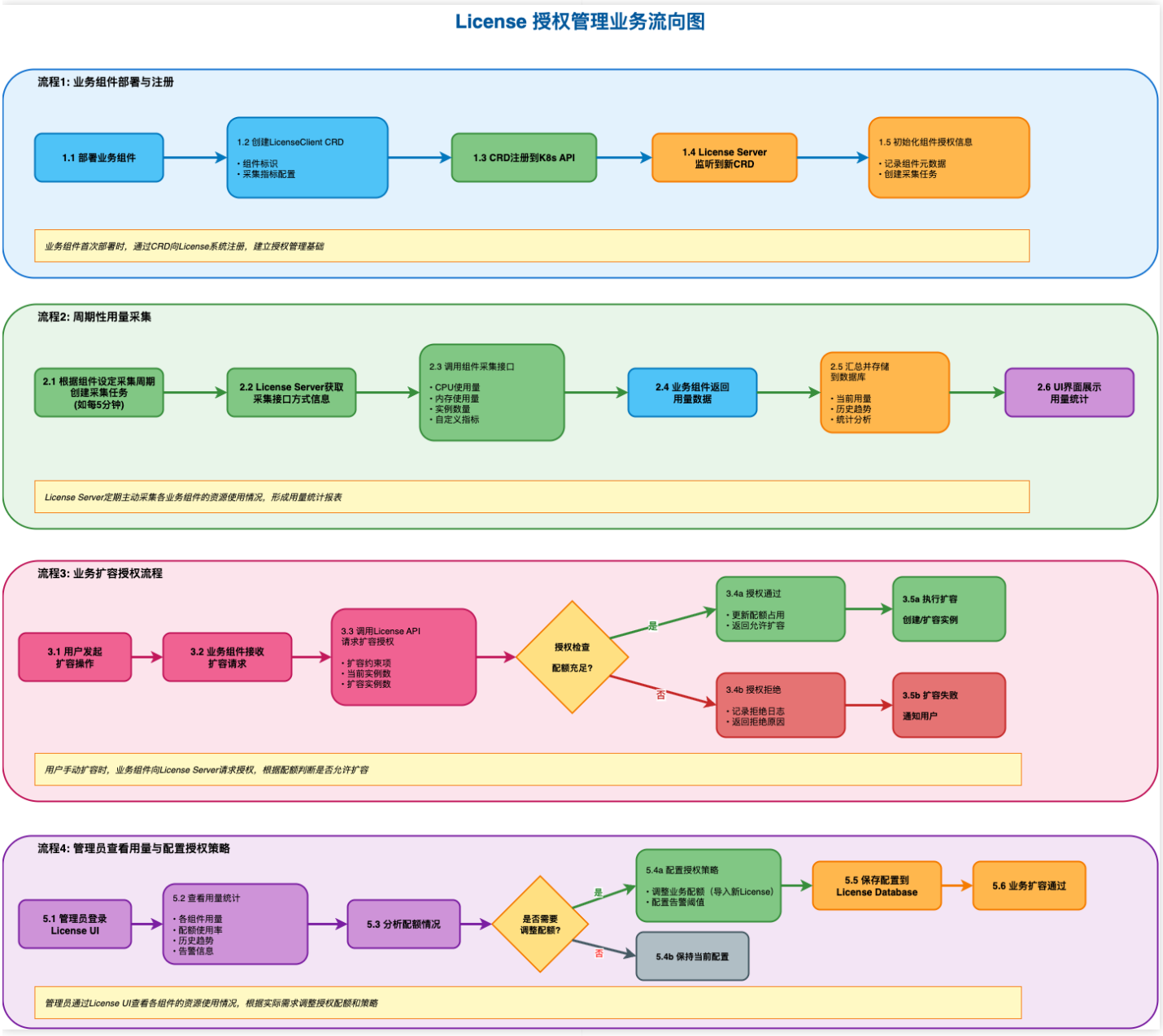
云API管理

核心访问链路为：前端->capi->traefik->ft 后台服务。



License管理

业务流向图



核心逻辑概述
基础平台

用户查询资源列表核心逻辑

1. 查看有权限的资源列表接口，logic.cam.checkResource 其中调用cam查询用户加入的所有**tce_project_${ProjectName}**用户组，得到加入的项目列表，tagList，key=tce_sp_project，value=项目名称）。
2. 如果用户具备相应的权限，根据tagList查出拉取标签资源的接口，将有相关标签的资源同时加载出来qcloud.tag.getResourcesByTags。
3. 根据上述步骤获得的资源情况，根据情况（deny allow）与主账户下的全部资源做筛选，得出应该显示的资源，其中对标签key是tce_sp_project的资源，验证其策略时，tag必须与项目名对应。

添加用户及授权

1. 项目对应的用户组 tce_project_${项目名} 添加用户，再给相应的用户赋予策略权限。
2. 添加成员页面只能添加未加入项目的用户，已加入项目的成员，只能通过表格进行编辑策略。

> 注意：
>
> 策略列表由项目的元数据维护，展示可添加的策略列表。在实际给用户授权时查询cam，若该项目的此策略不存在则新建后再授权给用户。cam 策略名称约定: tce_sp_project_${项目名}_${策略名}, cam 对 tce_project 前缀的策略做特殊处理。在实际展示策略名时去除前缀，即需要列出所有tce_sp_project_开头的策略。

用户查询 location 服务获取产品地域部署信息逻辑

1. 用户查询地域可用全产品列表。
2. 如果用户中指定了特定的地域则从全量地域配置中筛选出这些指定的地域。
3. 根据不同的条件选择并应用相应的策略：
    - 如果 input.ProductID 是特殊产品 ID，则支持“全量”列表。
    - 如果 input.ProductID 为空，则返回全产品列表，并处理地域白名单：
        - 如果不是 CMGT 请求，应用白名单策略（WhitelistStrategy）。
    - 如果是运营端请求，应用部署策略（DeploymentStrategy）。
    - 默认情况下，同时应用部署策略（DeploymentStrategy）和白名单策略（WhitelistStrategy）。DeploymentStrategy：根据产品和子产品的部署信息，过滤掉那些没有部署相关产品和子产品的区域和可用区。

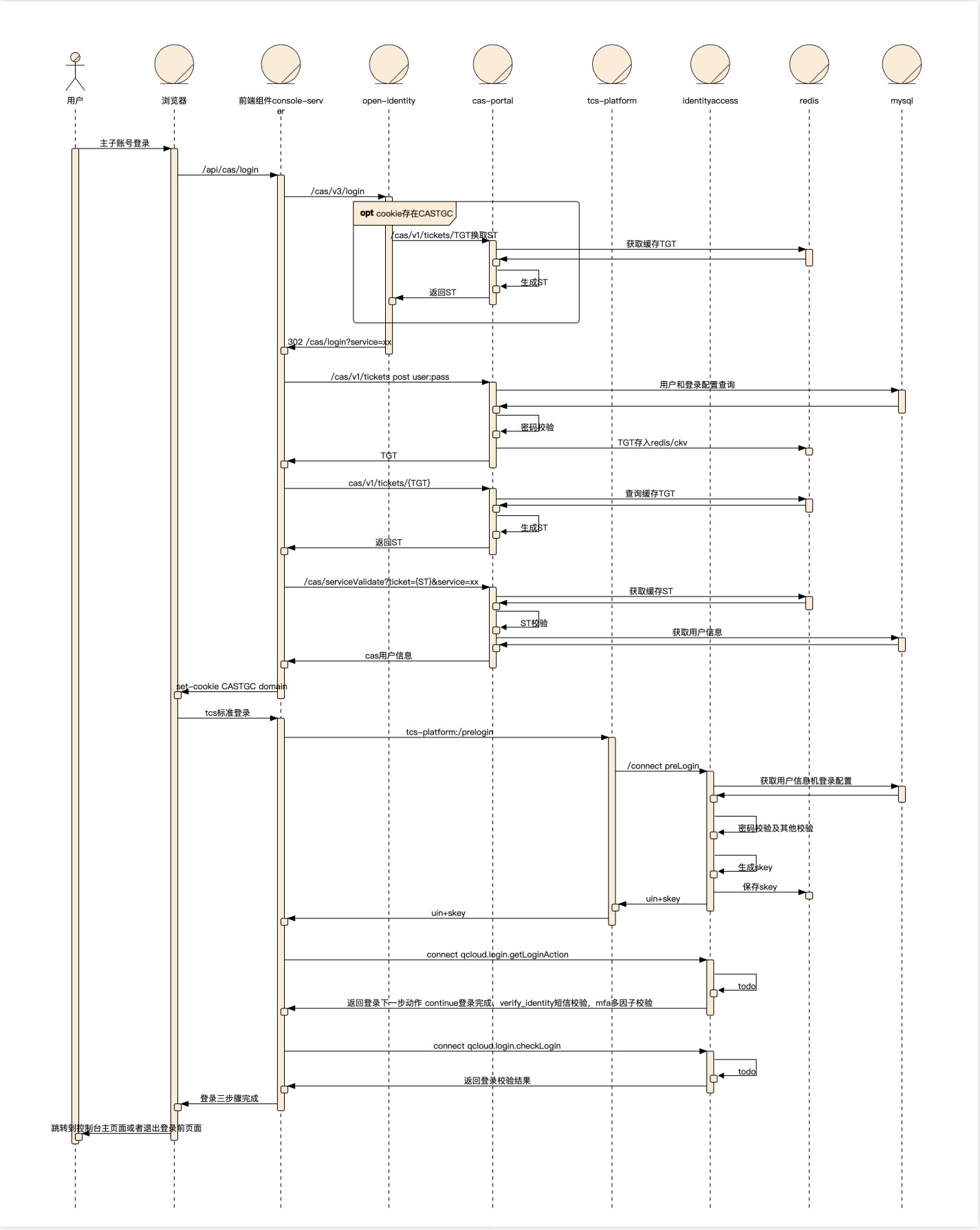
WhitelistStrategy：根据白名单配置过滤和更新区域和可用区的状态。

- 未配置白名单: 不受白名单约束，全部放通
- 已配置全量：所有uin都可以访问
- 已配置生效中：只有在白名单中uin才能访问
- 已配置无效：所有uin都不能访问

  ==> 等价转换为:
- 已配置生效中: 不在白名单中的uin, 不能访问
- 已经配置无效: 所有uin不能访问

用户与权限

登录核心逻辑：



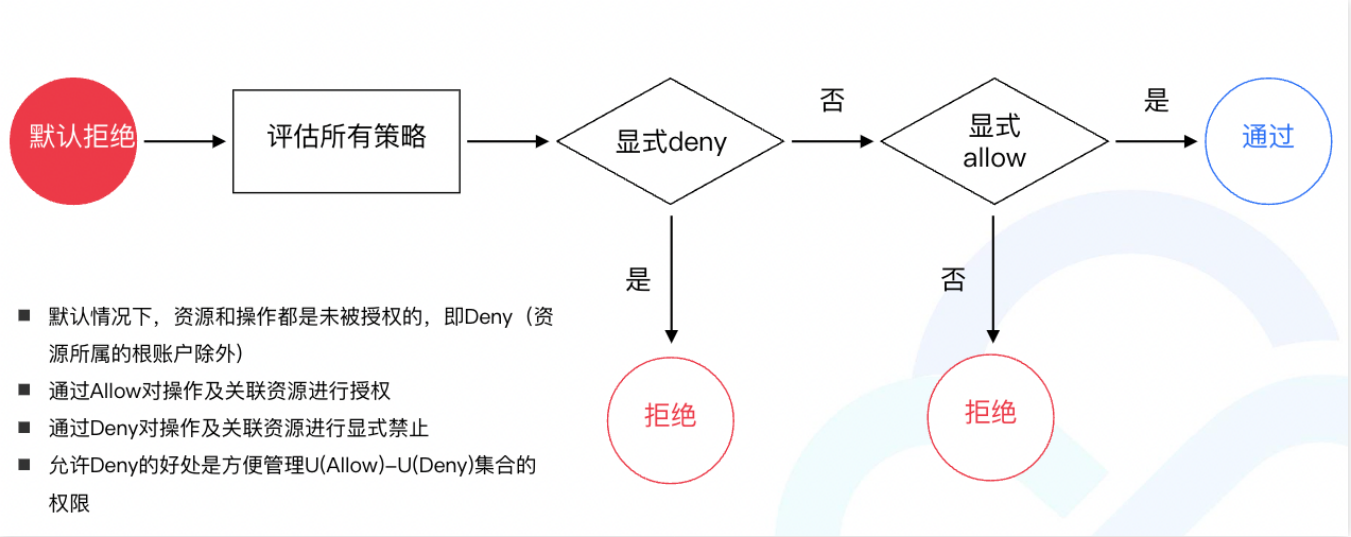
登录时，首先判断cookie是否存在TGT，如果存在TGT，通过TGT生成ST，使用ST换取用户消息，调用登录接口直接

登录进控制台。
如果cookie不存在TGT，跳转到登录页，走云平台标准登录流程。
云平台登录流程中，检测账号是否被锁定，检测账号密码是否正确，密码是否过期，是否开启登录保护。
登录过程中调 TDSQL，登录后将 skey 写入 Redis®，将登录信息上报给 Kafka。

访问管理

垂直产品按照规范接入云API，根据需要将云API接口设置为签名鉴权 或者 只签名不鉴权：

1. 对于开启了 签名鉴权 的云API接口，由云API访问CAM进行签名鉴权。
2. 对于设置为 只签名不鉴权 的云API接口，云API直接将请求透传给垂直产品，垂直产品可选择自行访问CAM进行签名鉴权。
   云API或垂直产品提取请求中的 账号/接口/资源/条件 等信息，调用CAM接口判断当前登录账号是否有权限通过云API接口访问特定资源。
   CAM支持常规鉴权、匿名鉴权、跨账号鉴权、token鉴权、标签鉴权、列表鉴权等，通用鉴权流程如下：

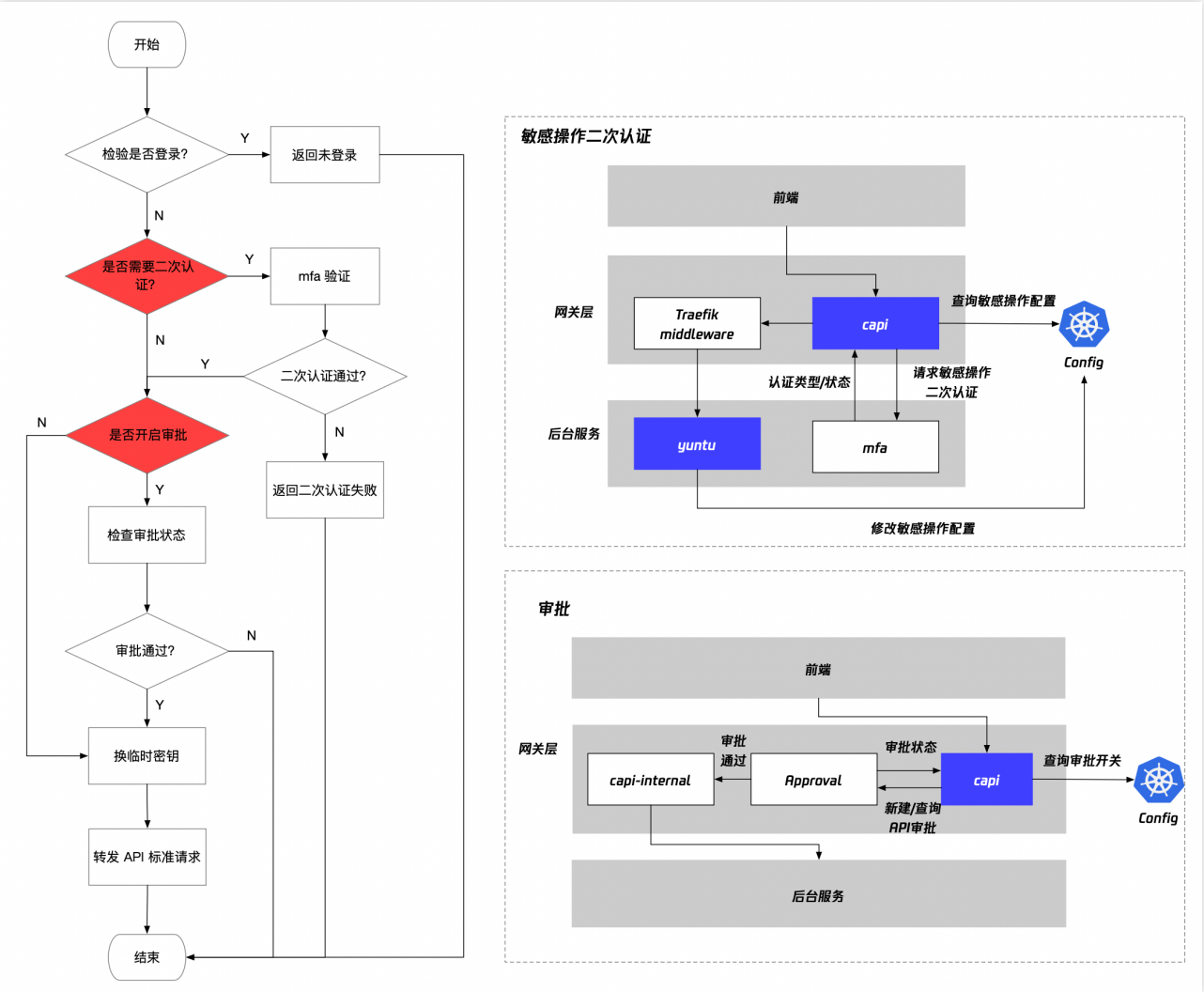


其中列表鉴权利用标签实现资源隔离，垂直产品直接调用CAM checkResource接口查询当前账号有权限访问的资源实例及标签列表。

云API管理

capi 控制云 API 请求核心流程：

- 登录检查流程：根据 uni 和 SKey 请求 auth 服务，检查登录状态。
- 敏感操作二次认证流程：网关查询API 是否敏感操作且开启二次认证，如果开启需要返回前端特殊错误码，前端弹出窗口引导用户输入 code在次请求，网关透传 code 给 mfa 验证。
- 审批检查流程：
  - 网关查询审批开关是否打开，如果打开则调用审批服务新建审批记录，后续请求会调用审批服务检查审批状态是否通过。
  - 审批通过则会调用capi-internal 模拟一次前端请求到网关，请求生效。



License管理

组件注册与授权初始化

目的：自动发现和注册新业务组件
流程：

1. License Controller 监听 Kubernetes API Server 上的 LicenseClient CRD 资源变化事件
2. 从 CRD 读取组件标识、采集配置等信息
3. 将授权信息、配额配置写入 License Database，更新 CRD status 为已注册
4. 通过事件机制通知注册结果

周期性用量采集

目的：定期采集各组件的资源使用情况
流程：

1. License Server 定时调度器每 5 分钟触发采集任务
2. 从 Kubernetes 获取所有已注册的 LicenseClient 资源列表
3. 并发调用各业务组件暴露的采集接口，获取约束项用量（CPU、Memory、实例数等）
4. 汇总采集结果，计算配额使用率，按时间序列写入 License Database
5. License UI 读取数据库展示统计报表和趋势图

扩容授权检查

目的：在组件扩容前验证是否有足够的配额
流程：

1. 业务组件发起扩容请求（包含组件 ID、目标实例数等）
2. License Server 验证请求合法性
3. 从 License Database 查询组件的总配额和当前使用量
4. 执行配额检查： 新增用量 + 当前使用量 ≤ 总配额 → 允许，否则拒绝
5. 返回允许/拒绝结果，记录审计日志

四、配额策略与告警

告警规则：

- 预警（≥80%）：黄色告警，建议尽快扩容
- 告警（≥100%）：红色告警，禁止扩容，必须立即处理

通知渠道：站内信、邮件、短信、公告

管理员操作：

1. 登录 License UI 查看实时用量、历史趋势、告警状态
2. 修改组件配额、调整告警阈值、设置有效期
3. License Server 监听策略变更事件，实时加载新配置
4. 所有变更操作记录在 License Database 供审计追踪

五、关键设计原则

原则	说明
事件驱动	基于 CRD 事件自动触发注册流程
异步并发	采集时并发调用各组件接口，提高效率
配额优先	扩容前必须检查配额，防止资源超用
多层告警	预警和告警分级处理，及时通知管理员
完整审计	所有操作（注册、采集、检查、变更）都有日志
实时同步	策略变更立即生效，无需重启系统

六、核心数据流

CRD 变化事件
↓
Server 监听 → 读取配置 → 写入数据库 → 更新 CRD 状态
↓
定时采集触发
↓
Server 获取组件列表 → 并发采集接口 → 汇总用量 → 写入数据库
↓
告警规则检查 → 触发通知（平台公告/邮件/微信）
↓
UI 展示 → 管理员查看/配置 → 策略变更 → Server 同步新配置

产品依赖
用户与权限

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑tdsql	数据库	依赖serviceid ： dbsql-ocloud-tcenter-identityaccess dbsql-ocloud-tcenter-open-identity 具体数据库： t_auth2 存储运营端用户账号相关信息信息	高
支撑kafka	消息队列	依赖serviceid kafka-ocloud-tcenter-identityaccess - identity-group-topic - sen_cloudaudit 审计信息存储	高
支撑Redis®	Redis® 缓存	依赖serviceId ckv-tcenter-identityaccess 存储用户登录缓存信息	高
CAM	cam鉴权	创建账号依赖cam服务授权等	高
cas	cas登录	运营端依赖cas登录	高
国密配置	国密配置依赖	依赖存储加解密，支撑配置加解密，传输加解密，签名验签	高
foyo	foyo配置中心	依赖foyo配置，运营端配置管理可以查看	中

审计日志

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑kafka	消息队列	作用：存储审计日志 依赖serviceid ： kafka-tcenter-ocloud-cam kafka-cam 相关topic： kafka-cam - auth_cloudaudit - signature_cloudaudit - sen_cloudaudit - cloudapi_req_production_json_log - audit_track - update_org_members kafka-tcenter-ocloud-cam - auth_ocloud_cloudaudit - signature_ocloud_cloudaudit - sen_ocloud_cloudaudit - cloudapi_ocloud_req_production_json_log	高
支撑ES	索引管理	作用：存储审计日志 依赖serviceid： es-ocloud-cloudaudit es-cloudaudit	高
支撑 Redis®	数据缓存	作用：存储缓存数据 依赖serviceid： ckv-tcenter-ocloud-cam-log-yunapi ckv-cam-log-yunapi	高
CAM	数据依赖	上报具体审计日志	高

访问管理

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑tdsql	数据库	依赖serviceid ： dbsql-ocloud-tcenter-cam dbsql-tcloud-tcenter-cam 具体数据库： cam	高
支撑kafka	消息队列	作用：存储鉴权、审计日志信息 依赖serviceid ： kafka-tcloud-tcenter-identityaccess kafka-ocloud-tcenter-identityaccess kafka-tcenter-ocloud-cam kafka-cam 相关topic： kafka-tcloud-tcenter-identityaccess - tcloud-identity-group-topic - tcloud_sen_cloudaudit kafka-ocloud-tcenter-identityaccess - identity-group-topic - sen_cloudaudit kafka-tcenter-ocloud-cam - auth_ocloud_cloudaudit - signature_ocloud_cloudaudit - sen_ocloud_cloudaudit - cloudapi_ocloud_req_production_json_log kafka-cam - auth_cloudaudit - signature_cloudaudit - sen_cloudaudit - cloudapi_req_production_json_log - audit_track - update_org_members	高
标签Tag	接口依赖	依赖serviceid ： tcloud-tcenter-platform-tag	高
账号	接口依赖	依赖serviceid ： tcloud-tcenter-platform-account	高

标签管理

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑tdsql	数据库	依赖serviceid ： dbsql-tcenter-platform-account 具体数据库： qcloudTag	高
支撑cmq	消息队列	依赖service ID： cmq-tcenter-wtag 服务通过订阅cmq获取tag信息	高

云API管理

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑tdsql hp_api_formal	数据库	依赖serviceid ： dbsql-tcenter-tcloud_api3, dbsql-tcenter-ocloud_api3 具体数据库： hp_api_formal 存储yunapi接口和后端服务地址信息 api_sync CRD同步信息	高
Kafka	消息队列	依赖服务：cloudaudit_kafka topic： cloudapi_ocloud_req_production_json_log（运营端） cloudapi_req_production_json_log（租户端） 用于云审计需求	高
CAM服务	cam鉴权服务	serviceid： ocloud-tcenter-cam-auth（运营端） tcloud-tcenter-cam-auth（租户端） 开启yunapi鉴权依赖cam服务进行鉴权 cam 写 ocloud-tcenter-cam-grant-write cam 读： ocloud-tcenter-cam-grant 内部接口调用cam 角色读写	高
CAM 服务	cam 换临时密钥	serviceid： tcloud-tcenter-cam-sts（租户端） ocloud-tcenter-cam-sts（租户端） 获取临时密钥	高
账号	账号登录认证	serviceid： tcloud-tcenter-tcs-authn-80（租户端） ocloud-tcenter-platform-verify（运营端） 请求云 API 时进行登录状态验证	高
账号	敏感操作二次认证	serviceid： tcloud-tcenter-platform-mfa（租户端） ocloud-tcenter-platform-mfa（运营端）	高

依赖产品名称	依赖类型	依赖关系描述	影响程度
审批	审批服务	serviceid： ocloud-tcenter-approval（租户端） ocloud-bsp-approval-svr（运营端）	高
密钥	密钥服务	serviceid: passwd-secret	低

License管理

依赖产品名称	依赖类型	依赖关系描述	影响程度
支撑TDSQL	数据库	依赖service id：dbsql-tcenter-license 数据存储	高
秘钥	密钥服务	依赖serviceid: passwd-secret 依赖存储加解密，支撑配置加解密，传输加解密，签名验签	高
消息中心	消息服务	依赖service id:ocloud-tcenter-message-svr 预警消息发送	中

故障处理
故障影响范围
标签管理

序号	场景	对云产品数据面影响	对平台控制面影响
1	标签页面报错	无	影响客户查看标签页面
2	产品页面给对应资源添加标签后不显示	无	影响客户对特定的资源赋予标签

License管理

序号	场景	对云产品数据面影响	对云产品业务面影响	对平台控制面影响
1	License 后端服务挂掉	无	License 挂掉云产品扩容默认放行	影响客户查看License相关信息
2	License 后端服务响应慢	无	云产品扩容卡顿或者超时	无
3	License 采集数据异常	部分依赖License采集用量数据做显示的产品显示会受影响	部分依赖License采集用量数据的产品扩容计算会出现错误（已使用数据不更新）	无，异常产品会显示异常

故障恢复
场景1：运营端的用户与权限页面打开报错或新创建账号报错

故障现象场景描述

运营端-用户与权限页面打开报错或新创建账号报错。

故障影响范围

- 对云平台产品管控的影响：影响现有账号展示和新创建账号、可能会影响运营端登录。
- 对用户生产业务的影响：无。

故障定位分析

1. 相关pod状态异常。
2. 服务渲染异常、数据库异常等。

故障应急处置步骤

1. 为快速恢复业务，重启相关pod服务。

```
kubectl get pod -n tce | grep -E 'cloud-tcenter-identityaccess|cloud-tcenter-open-identity'
kubectl delete pod -n tce ocloud-tcenter-identityaccess-xxxx/ocloud-tcenter-open-identity-xxx
```

2. 重启pod不能恢复业务需要进一步进到pod内查看日志进行故障排查。

```
kubectl get pod -n tce | grep -E 'cloud-tcenter-identityaccess|cloud-tcenter-open-identity'
kubectl exec -it -n tce ocloud-tcenter-identityaccess-xxxx bash
# 检查服务主进程是否运行
ps -ef
cd /data/log/
#通过报错reqid 查询日志
grep -ir $reqid app.log
```

```
[root@ocloud-tcenter-identityaccess-5cf646794d-lr4kh log]# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0 32 Sep29 ?        2-17:29:21 /identityaccess --configPath=/config/config.yaml
root          97       0  0 10:18 pts/0    00:00:00 bash
root         138      97  0 10:23 pts/0    00:00:00 ps -ef
[root@ocloud-tcenter-identityaccess-5cf646794d-lr4kh log]#
```

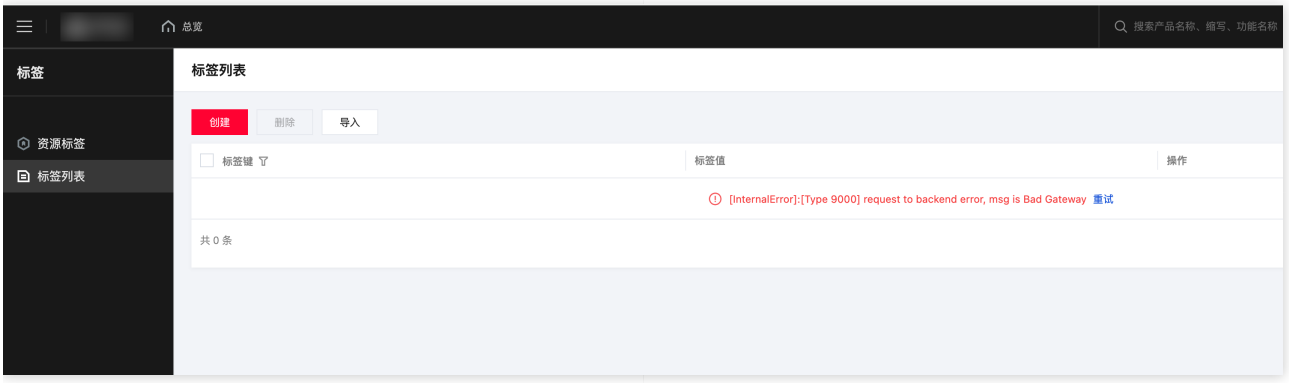
故障恢复验证

页面报错消失或者服务功能恢复。

场景2：标签服务页面报错

故障现象场景描述

标签服务页面报错：



故障影响范围

- 对云平台产品管控的影响：影响标签查询，修改等。
- 对用户生产业务的影响：无影响

故障定位分析

后端服务异常，容器化部署：

1. 服务pod状态异常。
2. 服务单个接口功能异常。

故障应急处置步骤

1. 检查服务pod状态：检查pod是否正常拉起，pod状态是否正常。

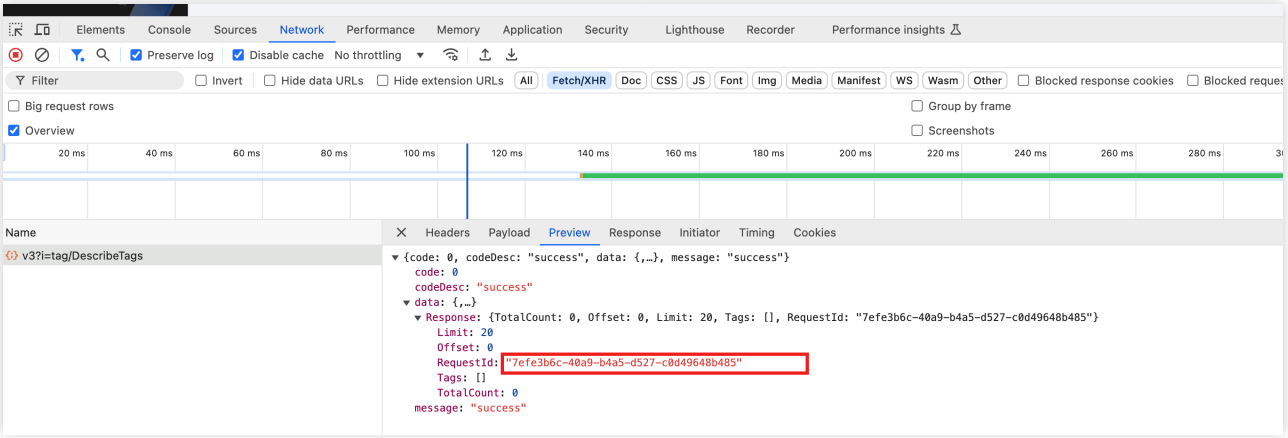
   kubectl get pod -n tce | grep tag

   - pod 状态不正常，删除重启pod 。

     kubectl delete pod -n tce tcloud-tcenter-platform-tag-xxxxx

   - pod状态正常，通过页面RequestId 查询服务日志排查根因。

```
kubectl get pod -n tce | grep tcloud-tcenter-platform-tag
kubectl exec -it -n tce tcloud-tcenter-platform-tag-xxxx bash
cd /data/log/swoole_tag
grep -ir "{RequestId}" ./*
```



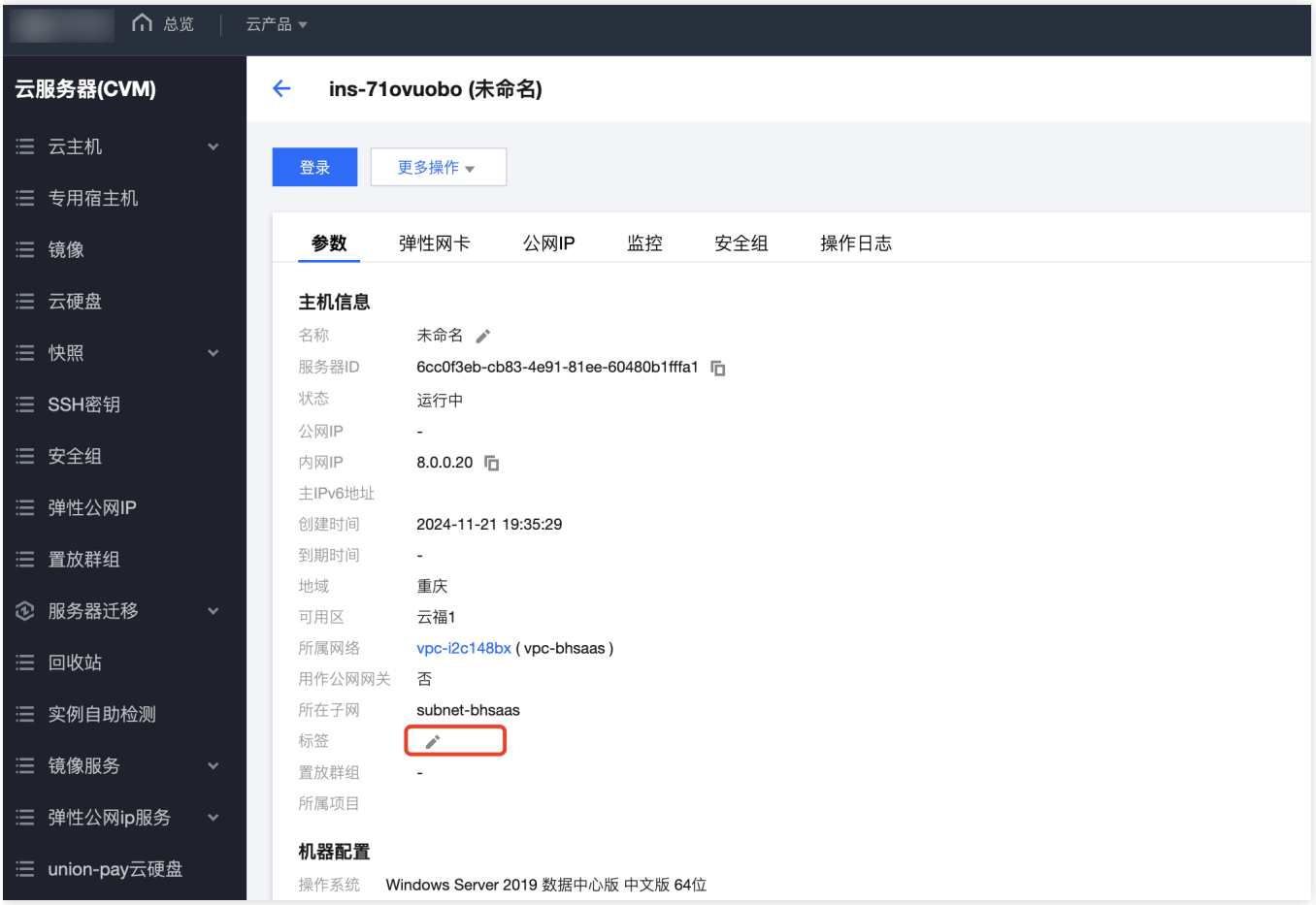
故障恢复验证

页面功能恢复。

场景3：产品页面给对应资源添加标签后不显示

故障现象场景描述

例如cvm页面给某个机器赋予标签后，页面提示成功无异常，但cvm页面不显示标签信息。



故障影响范围

- 对云平台产品管控的影响： 影响客户对特定的资源赋予标签。
- 对用户生产业务的影响：无。

故障定位分析

1. 确认cmq服务是否正常：
   kubectl get pod -n sso | grep cmq

```
[root@tcs-10-25-0-154 ~]# kubectl get pod -n sso | grep cmq
cmq-tce-tce-support-cmq-1-1-756757645c-hvdxg                    1/1     Running            9          148d
cmq-tce-tce-support-cmq-1-1-756757645c-xm5bn                    1/1     Running            3          113d
exporter-default-cls-redis-test-5f44dc54c5-cmqpv                1/1     Running            0          105d
rabbitmq-exporter-tce-tce-support-cmq-1-1-76cb5f4bc6-t9l8n      1/1     Running            0          42d
rabbitmq-tce-tce-support-cmq-1-1-0                              1/1     Running            0          125d
rabbitmq-tce-tce-support-cmq-1-1-1                              1/1     Running            0          125d
rabbitmq-tce-tce-support-cmq-1-1-2                              1/1     Running            0          125d
```

2. 查看tag服务内cmq配置正常：

   #获取pod名称

   kubectl get pod -n tce | grep wtag

   #任选一个Running的pod查看配置

   kubectl exec -n tce tcloud-tcenter-platform-wtag-69b5f84465-6ccwm -- cat /data/release/swoole_tag_write/application/config/idc/hosts_config.php | grep MQ_API_HOST

   #将获取到的MQ_API_HOST设置为环境变量

```
[root@tcs-10-25-0-154 ~]# kubectl get pod -n tce | grep wtag
tcloud-tcenter-platform-wtag-69b5f84465-6ccwm                   1/1     Running            0          4d11h
tcloud-tcenter-platform-wtag-69b5f84465-lthrv                   1/1     Running            0          4d11h
tcloud-tcenter-platform-wtag-69b5f84465-pdbpn                   1/1     Running            0          4d11h
[root@tcs-10-25-0-154 ~]# kubectl exec -n tce tcloud-tcenter-platform-wtag-69b5f84465-6ccwm -- cat /data/release/swoole_tag_write/application/config/idc/hosts_config.php | grep MQ_API_HOST
    const MQ_API_HOST = "http://tce-support-cmq-1-1.mr-50000001.tce2az31011.fsphere.cn:5672/api/v1";
[root@tcs-10-25-0-154 ~]# MQ_API_HOST="http://tce-support-cmq-1-1.mr-50000001.tce2az31011.fsphere.cn:5672/api/v1";
[root@tcs-10-25-0-154 ~]# echo $MQ_API_HOST
http://tce-support-cmq-1-1.mr-50000001.tce2az31011.fsphere.cn:5672/api/v1
```

3. 查看mq订阅情况：

   #确认其中存在tag-resource-topic，若不存在，则提工单到平台侧处理

   curl $MQ_API_HOST/topics

```
[root@tcs-10-25-0-154 ~]# curl $MQ_API_HOST/topics
{"code":200,"message":"list topic success","data":[{"topicName":"tag-resource-ocloud-topic"},{"topicName":"tag-resource-topic"}]}[
```

   #查看订阅情况，例如本次则查看cvm订阅情况，若现场没有订阅需提工单到cvm产品侧处理

   curl $MQ_API_HOST/subs/tag-resource-topic | python -m json.tool | grep -C3 -i cvm

```
[root@tcs-10-25-0-154 ~]# curl $MQ_API_HOST/subs/tag-resource-topic | python -m json.tool | grep -C3 -i cvm
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5928    0  5928    0     0   216k      0 --:--:-- --:--:-- --:--:--  222k
        },
        {
            "endpoint": "http://v3.tcloud-cbs-cgw.mr-50000001.tce2az31011.fsphere.cn/cbs/tag",
            "filterTag": "mr-50000001.cvm.snapshot",
            "protocol": "http",
            "subscriptionName": "cvm-snapshot-mr-50000001",
            "topicName": "tag-resource-topic"
        },
        {
            "endpoint": "http://v3.tcloud-cbs-cgw.mr-50000001.tce2az31011.fsphere.cn/cbs/tag",
            "filterTag": "mr-50000001.cvm.volume",
            "protocol": "http",
            "subscriptionName": "cvm-volume-mr-50000001",
            "topicName": "tag-resource-topic"
        },
        {
--
            "topicName": "tag-resource-topic"
        },
        {
            "endpoint": "http://tcloud-cvm-event.cvm:80",
            "filterTag": "mr-50000001.cvm.instance",
            "protocol": "http",
            "subscriptionName": "cvm-instance-mr-50000001",
            "topicName": "tag-resource-topic"
        },
        {
--
        },
        {
            "endpoint": "http://tcloud-vpc-cgw-v3.vpc:8520/tag/",
            "filterTag": "mr-50000001.cvm.sg",
            "protocol": "http",
            "subscriptionName": "cvm-sg-mr-50000001",
            "topicName": "tag-resource-topic"
        },
        {
--
            "topicName": "tag-resource-topic"
        },
```

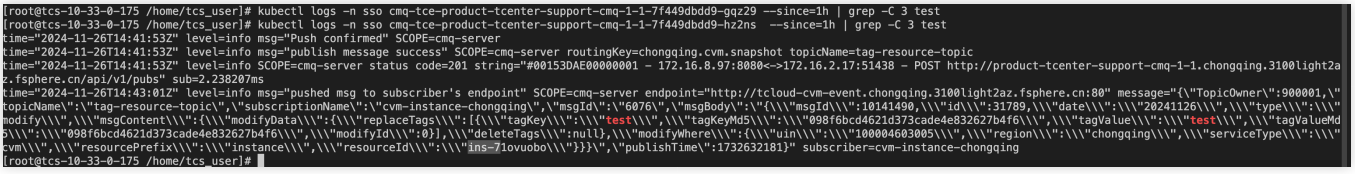
#确认订阅正常后，确认cmq回调日志是否触发

#首先查找cmq pod

kubectl get pod -n sso | grep cmq

#查看日志，此处since=1h表示过去一小时的日志，test为测试用的标签名称，也可用ins id等信息过滤，多个pod均需过滤，出现下图日志代表回调成功，进一步原因提单cvm处理；若未出现日志，提单平台侧处理

kubectl logs -n sso cmq-tce-tce-support-cmq-1-1-xxx --since=1h | grep -C 3 test



故障应急处置步骤

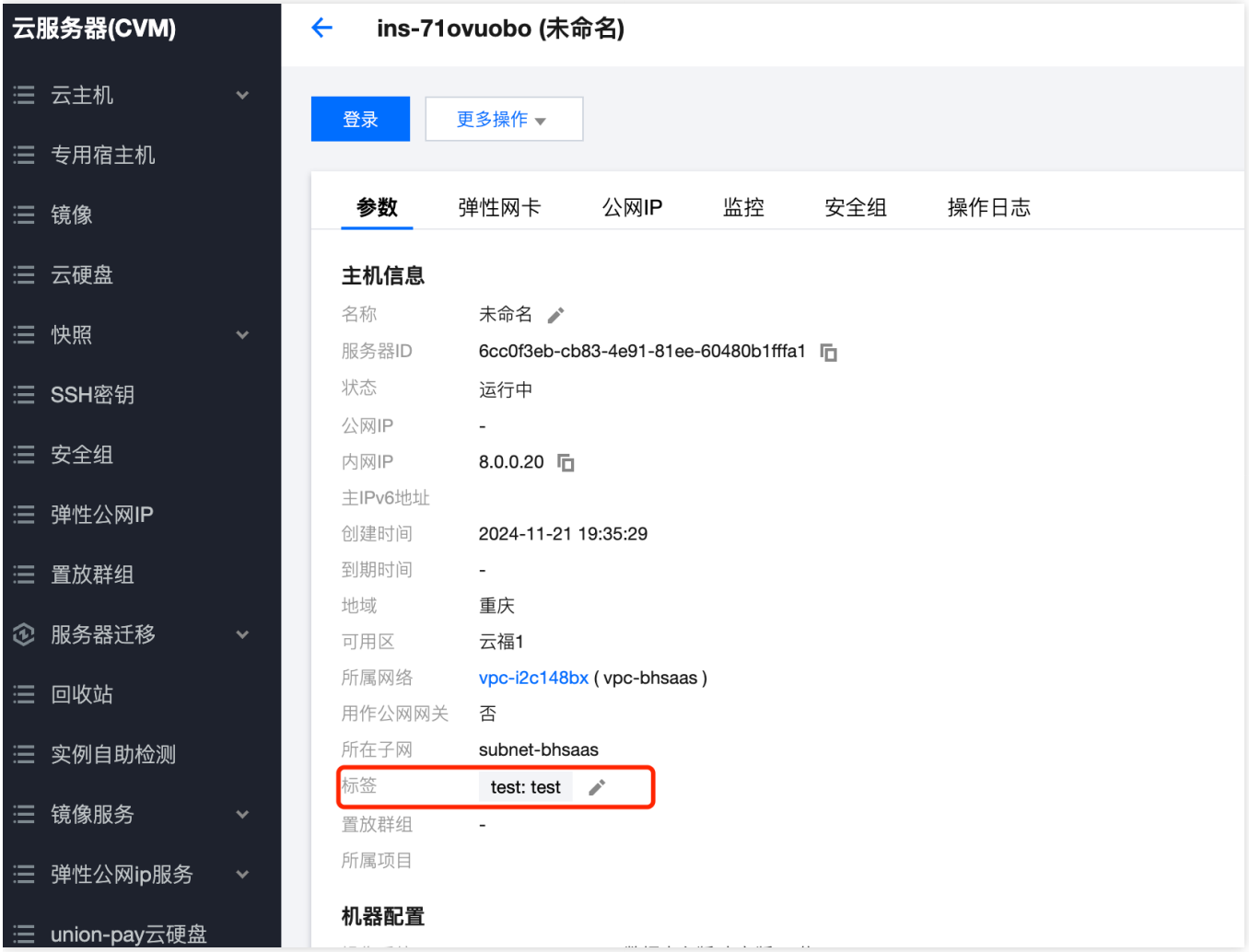
1. 若cmq topic丢失，可手动创建：

   curl -X POST -d '{"topicName":"tag-resource-topic"}' $MQ_API_HOST/topics

2. 若cmq缺失订阅，需找相关产品侧确认是否重启哪些组件可以重新订阅。
3. 若前两者均正常，cmq无回调消息，现场临时可考虑重启tag相关以及cmq的pod临时确认是否可恢复：
   kubectl rollout restart deploy -n tce tcloud-tcenter-platform-tag
   kubectl rollout restart deploy -n tce tcloud-tcenter-platform-wtag
   kubectl rollout restart deploy -n sso cmq-tce-tce-support-cmq-1-1

故障恢复验证

产品页面给对应资源添加标签后可以正常显示。



场景4：云api产品不可用，影响平台管控业务

故障现象场景描述

通过yunapi调用接口都报错。

故障影响范围

- 对云平台产品管控的影响：管控不可用，所用通过yunapi调用接口不可用。
- 对用户生产业务的影响：不影响客户已有存在的数据业务使用。

故障定位分析

- yunapi pod状态是否正常。
- 环境网络是否正常。

故障应急处置步骤

1. 查看pod状态是否正常，pod状态不正常可以先删除重启快速恢复业务。

```
kubectl get pod -n tce | grep tcenter-yunapi3
kubectl delete pod -n tce xxxx--tcenter-yunapi3-traefik-xxxxx
```

2. 重启不能恢复业务，进到容器内查看yunapi-traefik 到其它服务网络是否正常。网络异常修复网络，网络异常进一步通过查看服务日志排查问题。进一步恢复业务。

```
kubectl exec -it -n tce tcloud-tcenter-yunapi3-traefik-xxxx bash
cd /usr/local/services/traefik/log
#通过页面reqid过滤日志
grep -ir "reqid" traefik.log
 grep -ir "reqid" *
```

故障恢复验证

页面通过yunapi调用接口成功。

日常巡检
云产品巡检
查看巡检项

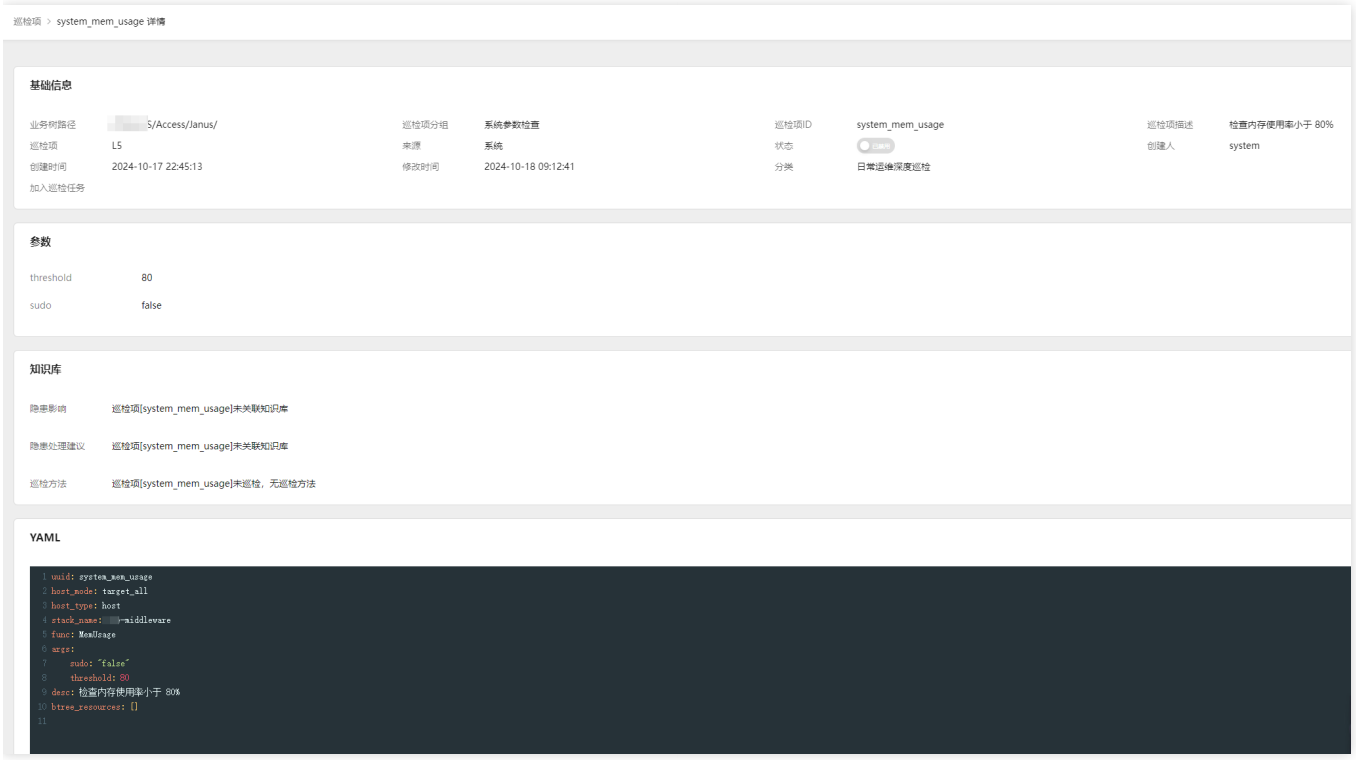
从【巡检平台】-【巡检项管理】进入，已为云产品内置了巡检项，可以通过巡检项来检查云产品状态。

1 巡检项列表

此模块汇总当前平台的巡检项清单，可对巡检项进行启用/停用、编辑、新增和删除；下拉框选择业务树路径、巡检项分组可以过滤巡检项，以及通过巡检项描述或巡检项分组等关键字搜索巡检项。

2 查看巡检项详情信息

在巡检项列表，点击要查看的【巡检项ID】，进入该巡检项详情页面；页面展示巡检项信息，包含：基础信息、参数、知识库、YAML。



3 编辑巡检项

在巡检项管理页面，点击【编辑】进行操作。


巡检平台
巡检控制台
巡检报告
巡检任务
巡检项管理
操作记录
巡检项 > 编辑巡检项
巡检项产品
un
巡检项分组
依赖服务检查
巡检项ID
tcs_system_check_defunct_process1
巡检项描述
检查服务器-僵死进程数小于20
巡检项分级
L2
巡检项来源
自定义
分类
日常运维
参数配置
analysis
{"cmd": "ps aux|grep defunct|grep -v grep|sort -rn -k3", "desc": "\u539f\u56e0\u5206\u6790-\u83b7\u53d6\u673a\u5668\u50f5\u
cmd_list[0].cmd
ps -A -ostat,ppid,pid,cmd |grep -e '^[Zz]'|wc -l
cmd_list[0].desc
检查服务器-僵死进程数小于20
cmd_list[0].expect_equation
le
cmd_list[0].expect_value
20
是否启用
是
否
加入巡检任务
v让对方v
test001
日常巡检-全量任务
提交巡检项

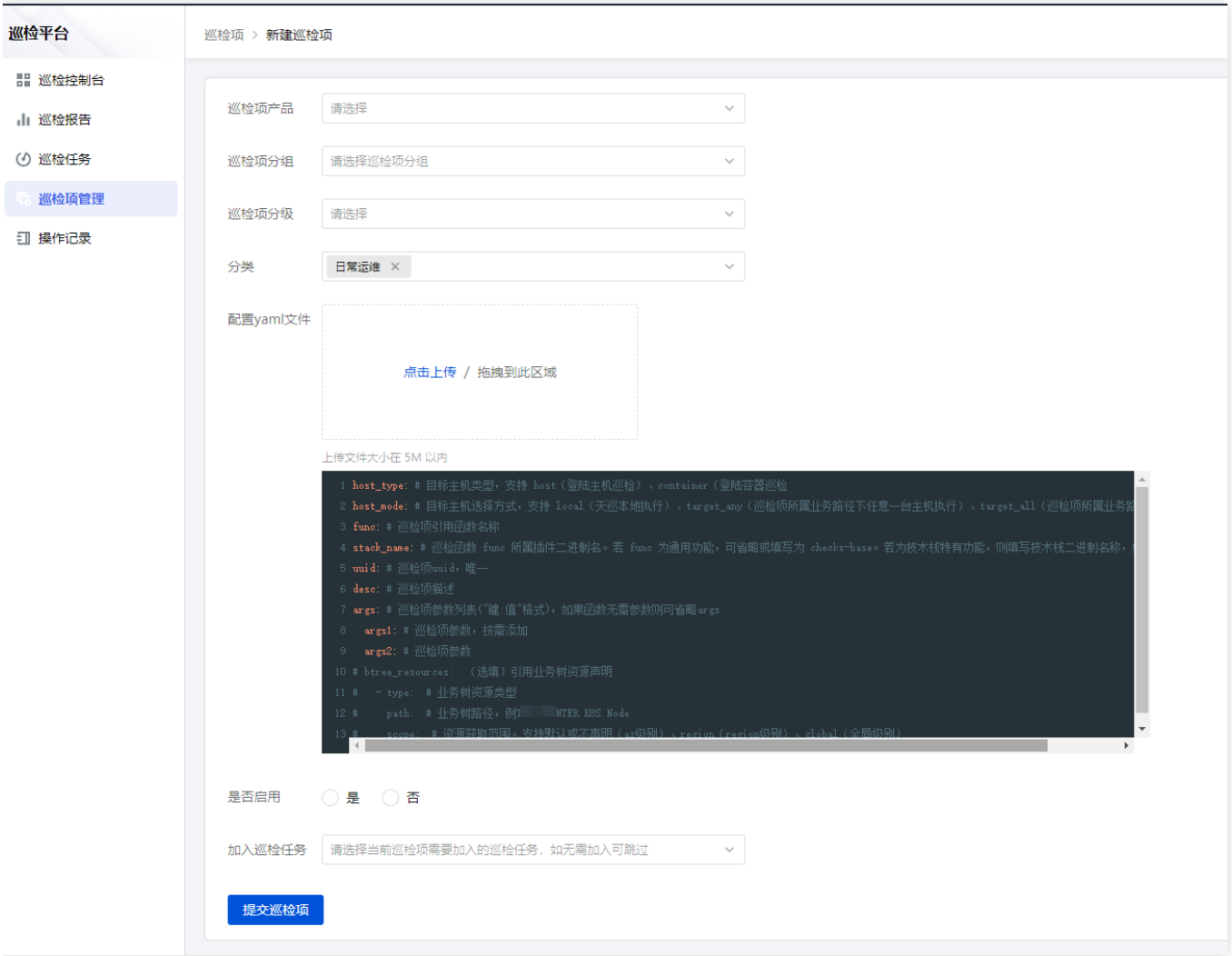
4 启用/停用巡检项

启用：点击未启用状态的按钮开关，对已停用巡检项进行启用，启用后已加入的巡检任务下一次执行时该巡检项会被执行。
停用：点击已启用状态的按钮开关，对启用中的巡检项进行停用，停用后已加入的巡检任务下一次执行时该巡检项不会执行，相关异常项自动消除。

5 添加自定义巡检项

支持巡检项白屏化管理，以下提供新建自定义巡检项的指引。

开始添加自定义巡检项
从【巡检平台】-【巡检项管理】进入，点击【新建巡检项】。



编写巡检项的流程为：

1. 巡检项产品：选择当前巡检项的检查对象
2. 巡检项分组：选择巡检项分组名称
3. 巡检项分级：选择巡检项分级
4. 分类：选择分类（默认是日常运维）
5. 配置yaml文件
    i. 根据巡检途径，找到合适的巡检插件、巡检函数。
        i. 远程执行命令通常使用 system 插件的 system_exec_command 函数
        ii. 执行 SQL 通常使用 mysql 插件的 check_field_value 函数
        iii. ……
    ii. 为巡检函数声明参数
    iii. 声明描述信息、uuid（uuid 尽量使用具有实际含义的英文名称）
6. 启用巡检项，并加入巡检任务。

配置yaml文件

(1)基础格式
yaml文件包含以下字段：

```
host_type: host  # 目标主机类型，支持 host（登录主机巡检）、container（登录容器巡检，当前路径是 k8s-master-ip 或者 tianxun 的时候，host_type 要写成 container）
host_mode: target_all  # 目标主机选择方式，支持 local（本地执行）、target_any（选取当前巡检项所属业务路径下任意一台主机执行）、target_all（选取当前巡检项所属业务路径全部主机执行）
func: MysqlExec  # 巡检项引用函数名称
stack_name: checks-base  # 巡检函数 func 所属插件二进制名。若 func 为通用功能，可省略此字段或填写为 checks-base。若为技术栈特有功能，则需填写技术栈二进制名称如 tce-compute
uuid: tx_cvm_podcheck_1 # （必填）巡检项 uuid
desc: 巡检项描述 # （必填）巡检项描述
args: # 巡检项参数列表("键:值"格式)，如果 func 无需参数则可省略 args
   database: yhimage
   sql: show tables;
# btree_resources:  （选填）引用业务树资源声明
#   - type: host  # 业务树资源类型
#     path: Platform.TCENTER.K8S.Node  # 业务树路径
#     scope: region  # 资源获取范围。支持默认或不声明（az级别）、region（region级别）、global（全局级别）
```

(2)引用业务树资源（btree_resources）

```
host_type: host
host_mode: local
func: ExecuteShResultNum
btree_resources:  # 此处声明业务树 Platform.TCENTER.K8S.Node 路径下当前 region 内 host 资源
   - type: host
     path: Platform.TCENTER.K8S.Node
     scope: region
args:
```

```
  sh_list:
    - sh_name: "tgwadm_whitelist_check.sh"
      params: '{{ .btree_resources.host | map(.ip) | join(" ") }}' # 此处使用 .btree_resources.host 来引用已声明的资源。语法为 jq 格式
      expect_equation: eq
      expect_value: 1
```

(3)常用函数格式

- 执行 shell 命令，比较输出结果与预期数字。

```
stack_name: checks-base
func: SystemCmdResultNum
args:
  cmd_list:
    cmd: "uptime | awk '{print $3/365}'"
    desc: '服务器运行时间是否超过3年'
    expect_equation: 'le'
    expect_value: 3
```

- 执行 shell 命令，比较输出结果与预期字符串。

```
stack_name: checks-base
func: SystemCmdResultStr
args:
  cmd_list:
    cmd: kubectl get secret -n cert-manager cert-manager-webhook-tls -o json | jq '.data["tls.crt"]' | x
args echo | base64 -d | openssl x509  -checkend 3888000
    desc: '检查cert-manager的证书过期时间在45天以后'
    expect_equation: 'eq'
    expect_value: 'Certificate will not expire'
```

- 检查 pod 是否存活

```
stack_name: checks-base
func: PodActiveCheck
args:
  pod_name: 'tcloud-argus2-adp-synchronizer'
```

- 检查 pod 是否存活，并在 pod 内执行 shell 命令。

```
stack_name: checks-base
func: PodActiveCheck
args:
  cmd_list:
```

```
    - cmd: ps aux | grep 'gw-controller-monitor' | grep -v grep | awk '{print $11}' |sort | wc -l
      desc: 检查ocloud-vpc-nfv-gw-controller gw-controller-monitor 进程运行正常
      expect_equation: ge
      expect_value: 1
  pod_name: ocloud-vpc-nfv-gw-controller
```

- 复制 shell 脚本并执行，检查执行结果。

```
stack_name: checks-base
func: ExecuteShResultNum
args:
  sh_list:
    - sh_name: "tgwadm_whitelist_check.sh"
      params: '"ip1" "ip2"'
      expect_equation: eq
      expect_value: 1
```

(4)完整函数列表

- checks-base

```
CheckOsVersion
ClusterAllocationCheck
CpuFrequency
CpuUsage
CrontabGrep
DiskCapacity
DiskMount
DiskUsage
ElasticsearchCheck
EtcdCheck
ExecuteShResultNum
ExecuteShResultNumProxy
ExecuteShResultStr
ExecuteShResultStrProxy
GetK8sCert
GetOcloudCert
GetServerCert
GetTcloudCert
HttpApiResult
HttpApiResultNum
HttpApiResultStr
IOUsage
LoadAverage
MemSwapUsage
MemUsage
```

MysqlCheck
MysqlCheckDemo
MysqlCheckFieldValue
MysqlExec
PartitionUsage
PingCheck
PodActiveCheck
PodImageCheck
PodLogCheck
PodNodeCheck
PodReplicaCheck
PodRestartCheck
PodUsageCheck
PortCheck
ProcessCheck
PvcUsageCheck
RedisConfigCheck
RedisExecCheck
ServiceActiveCheck
SyncDate
SysLogCheck
SystemCmdResult
SystemCmdResultNum
SystemCmdResultNumProxy
SystemCmdResultStr
SystemCmdResultStrProxy
SystemModeCheck
ZookeeperCheck

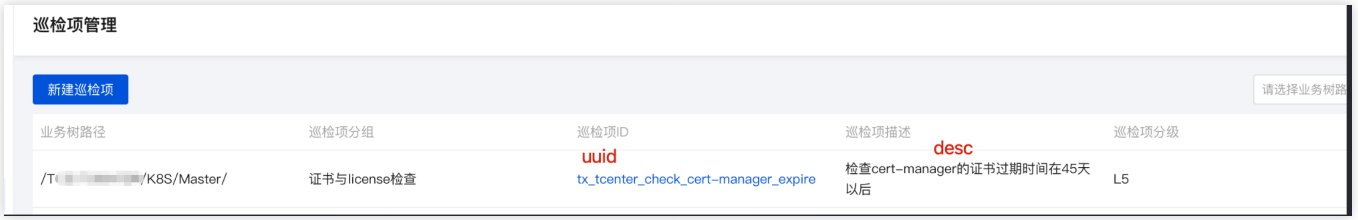
(5)示例

示例1：tx_tcenter_check_cert-manager_expire

```
uuid: tx_tcenter_check_cert-manager_expire
host_mode: target_all
host_type: host
stack_name: tce-base
func: SystemCmdResultStr
args:
  cmd_list:
    - cmd: kubectl get secret -n cert-manager cert-manager-webhook-tls -o json | jq '.data["tls.crt"]' | xargs
echo | base64 -d | openssl x509  -checkend 3888000
      desc: 检查cert-manager的证书过期时间在45天以后
      expect_equation: eq
      expect_value: Certificate will not expire
desc: 检查cert-manager的证书过期时间在45天以后
btree_resources: []
```

以上述巡检项为例：

- 该巡检项 uuid 为 tx_tcenter_check_cert-manager_expire，desc 为 '检查cert-manager的证书过期时间在45天以后'。uuid 和 desc 会展示在巡检列表页。



- 该巡检项的 host_mode 为 target_all，表示该巡检项最终会在所属业务树路径的所有机器上执行。如果是 target_any 的话则会在所属业务树路径的任一机器上执行。
- 该巡检项的 host_type 为 host，表示该巡检项的目标巡检对象为主机。除了 host 外，巡检平台还支持 container 类型的 host_type，表示目标巡检对象为容器或特殊节点。
- 该巡检项的 stack_name 为 tce-base，func 为 SystemCmdResultStr。表示巡检的实现位于名为 tce-base 的二进制文件中，被调用函数名称为 SystemCmdResultStr。
- 后续的 args 字段是需要传递给 SystemCmdResultStr 的参数。与 2.2 类似，SystemCmdResultStr 函数支持执行多个 shell 命令，每个 shell 命令使用 cmd 指定，预期输出值为 'Certificate will not expire'。

示例2：tx_cbs_pod_processor

```
uuid: tx_cbs_pod_processor
host_mode: target_any
host_type: container
stack_name: tce-storage
func: PodActiveCheck
args:
  cmd_list:
    - cmd: df -h | grep '/' |tr -d '%' | awk '{if ($5>80)print}' | wc -l
      desc: 检查pod容器磁盘使用率>80%的挂载分区个数为0
      expect_equation: eq
      expect_value: 0
  pod_name: ocloud-cbs-processor
desc: pod ocloud-cbs-processor 容器状态为running，磁盘使用率小于80%
btree_resources: []
```

在此只列出此巡检项与前一个巡检项不同的部分：

- 当前巡检项 host_mode 为 target_any，host_type 为 container。表示此巡检只需执行一次，同时需要到容器（pod）内执行。
- 当前巡检项 func 为 PodActiveCheck，args 为传递给 PodActiveCheck 的参数，此函数也支持多 shell 命令同时声明。

6 删除巡检项

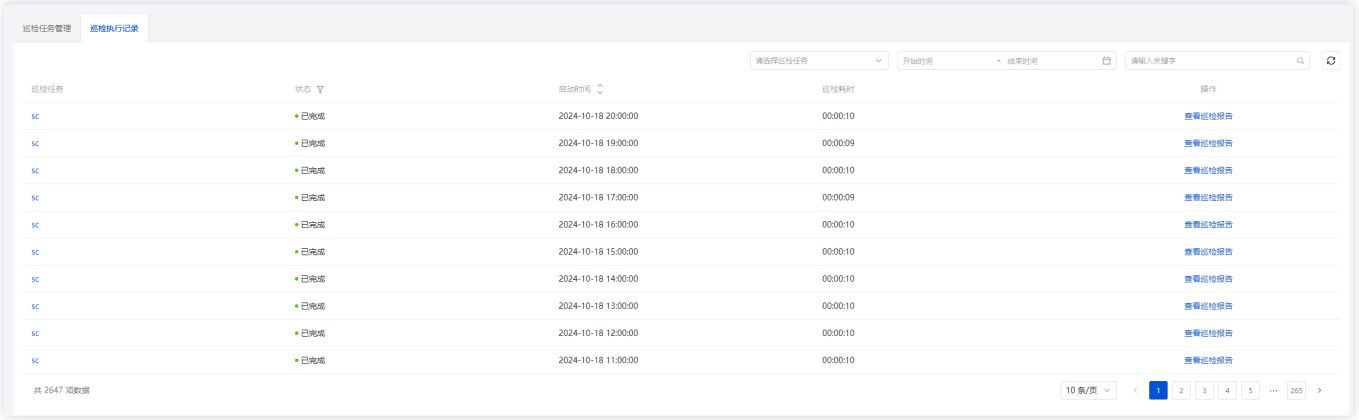
点击【删除】对自定义巡检项进行删除；系统巡检项不支持删除，如果不使用，则可以进行停用。

查看巡检任务

从【巡检平台】-【巡检任务】进入，里面包含了系统内置的巡检任务以及用户自定义的巡检任务。

1 巡检任务列表

此模块管理平台的所有巡检任务和巡检任务执行记录，支持巡检任务执行白屏化管理，直接在界面查看巡检任务执行记录，操作巡检任务创建、启用/禁用、立即执行、编辑和删除。



">

2 巡检任务详情页面

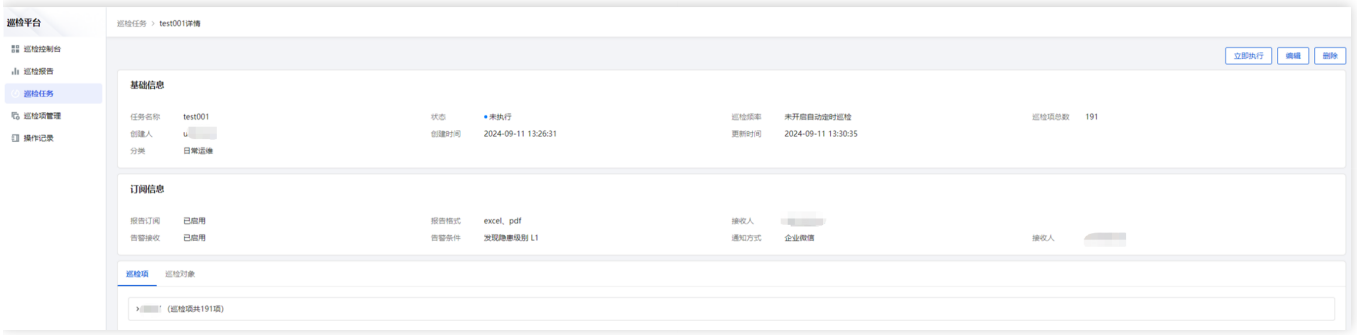
在巡检任务列表，点击要查看的【任务名称】，进入该巡检项详情页面巡检任务详情页面。

(1)基础信息展示巡检任务的任务名称、状态、巡检频率、创建时间等信息。

(2)订阅信息展示巡检任务的报告订阅以及告警订阅配置情况。

(3)巡检项列表展示（根据产品节点显示任务中的巡检项）。

(4)巡检对象列表展示（根据产品节点显示任务中的巡检对象）。



3 执行巡检任务

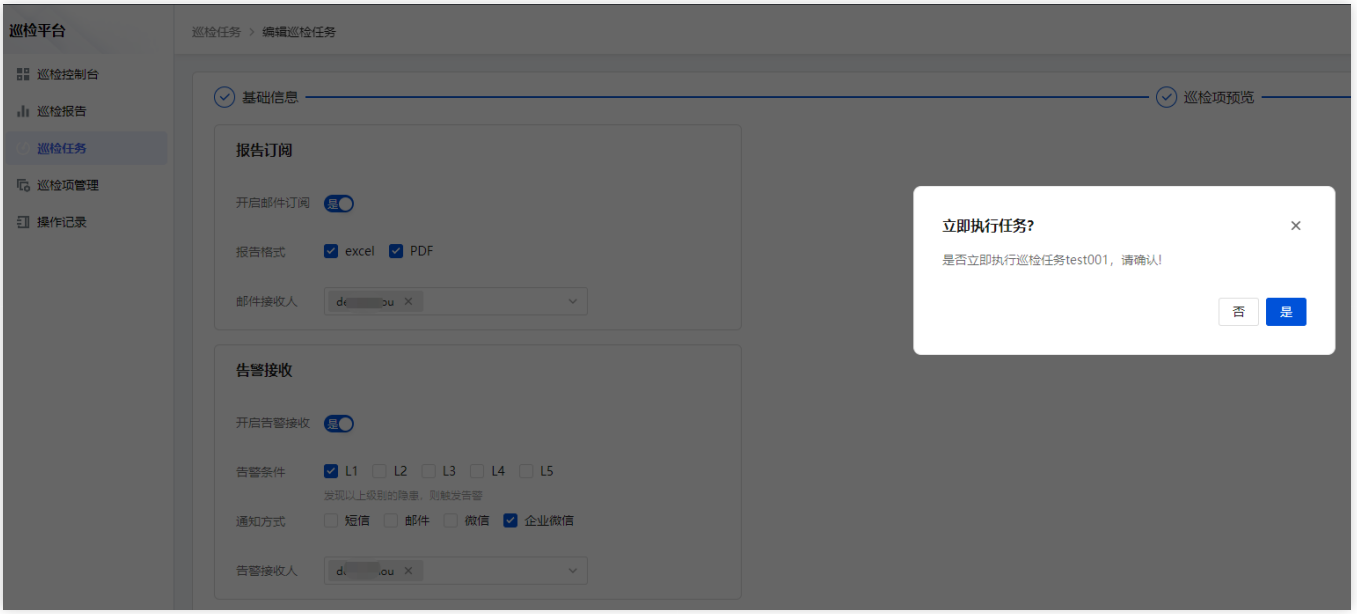
点击巡检任务的【立即执行】按钮，确认后会立即触发执行该巡检任务，执行后巡检结果将展示在巡检控制台和巡检报告模块

4 编辑巡检任务

点击巡检任务的【编辑】按钮对该巡检任务进行编辑/修改

完成提交修改时选择立即执行任务：

- 是：提交后会立即执行修改的巡检任务同时按修改后的信息执行。
- 否：修改的巡检任务在下一次执行时开始生效。



5 启用/禁用巡检任务

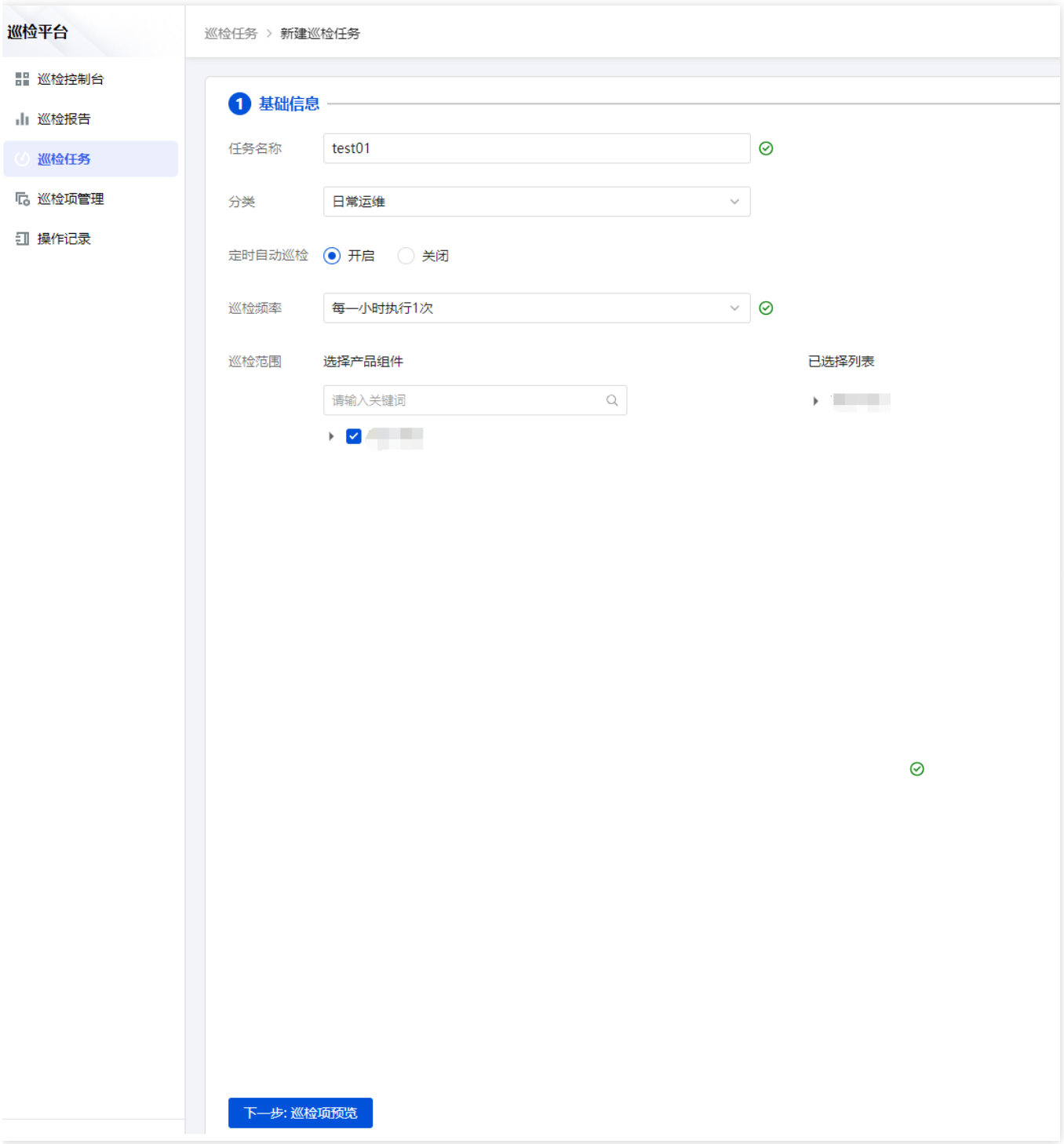
启用：点击未启用状态的按钮开关，对已停用的巡检任务启用，巡检任务将按巡检频率进行巡检
禁用：点击已启用状态的按钮开关，对已启用的巡检任务停用，巡检任务将不再执行

6 新建巡检任务

支持巡检任务白屏化管理，以下提供新建自定义巡检任务的指引，
从【巡检平台】-【巡检任务】进入，点击【新建巡检任务】

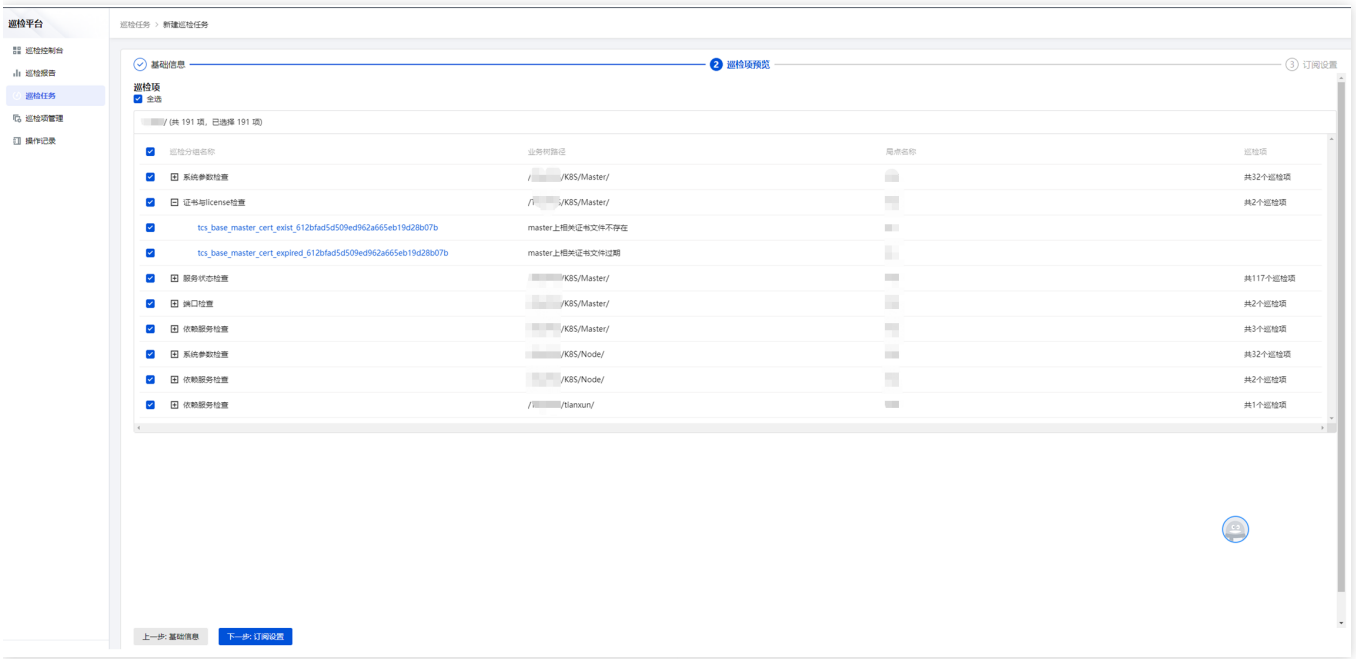
(1)填写巡检任务基础信息

任务名称：输入巡检任务名称；
分类：下拉框中选择；
定时自动巡检：选择是否开启定时巡检；
巡检频率：选择巡检任务自动巡检的频率，下拉框中选择；
巡检范围：选择此巡检任务的巡检业务树范围，按产品集群、产品节点等进行选择，需要选择有机器的业务树节点。



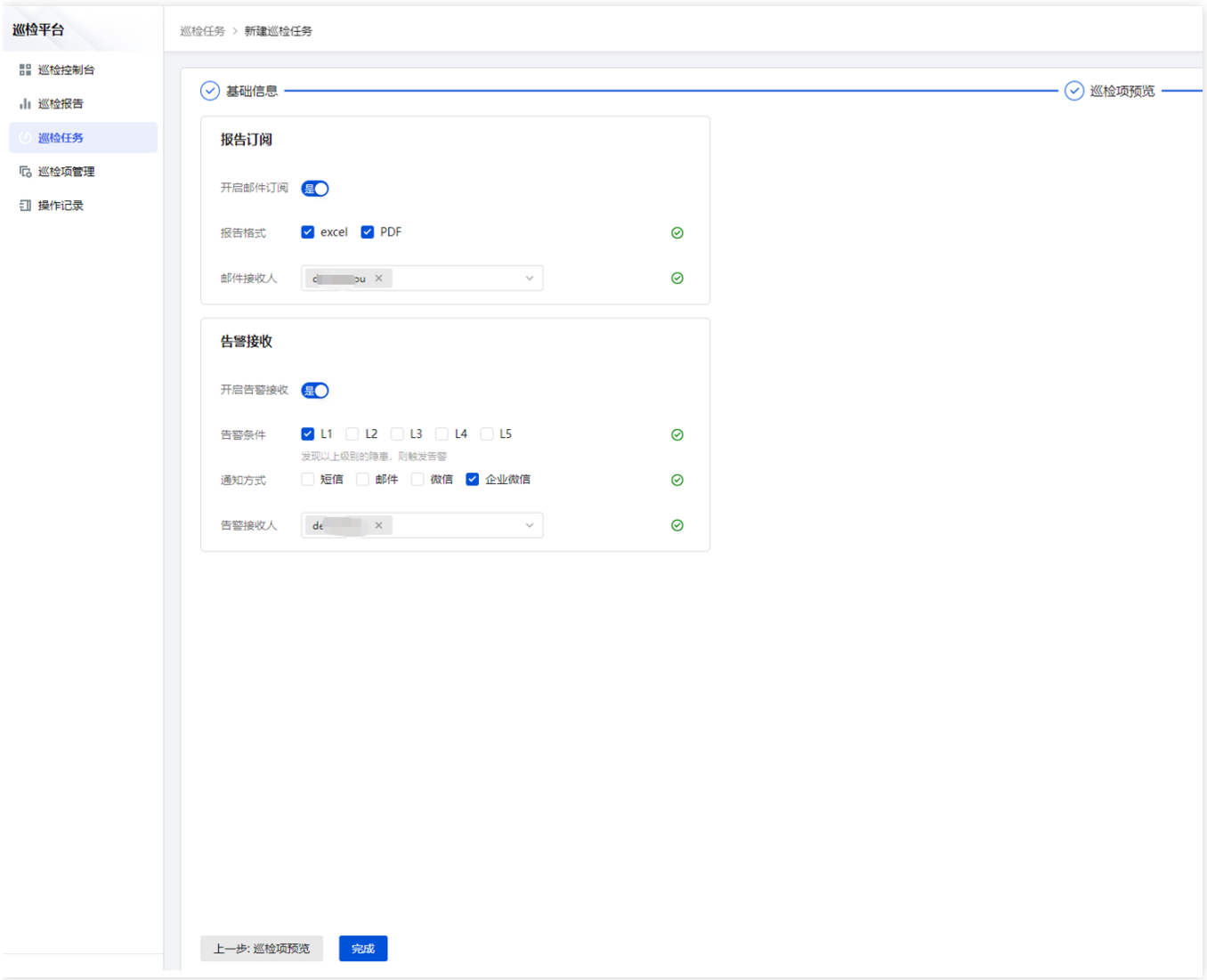
(2)选择巡检项

勾选要执行的巡检项



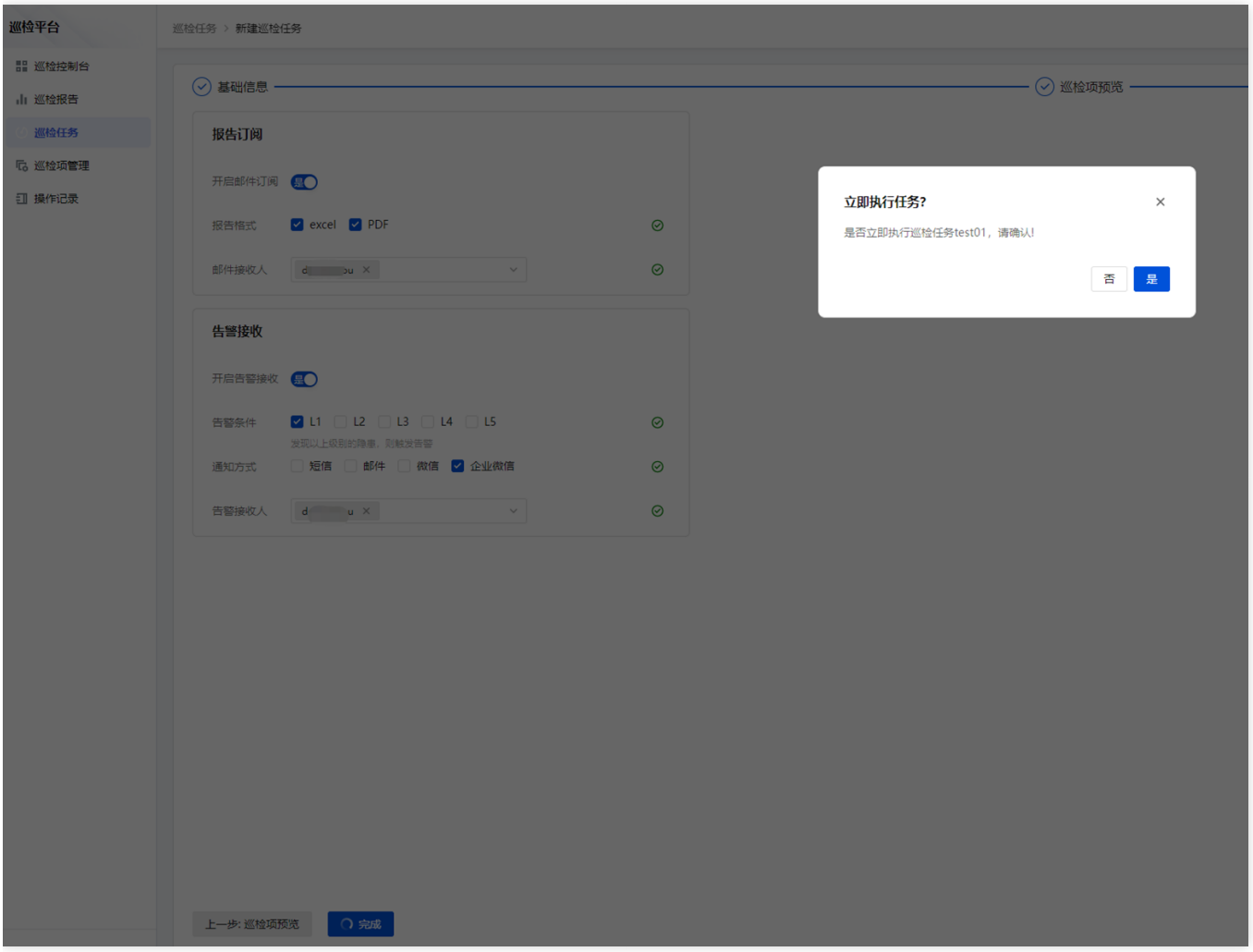
(3)订阅设置

根据自身需求，设置“报告订阅”和“告警接收”的相关信息



(4)提交巡检任务

提交巡检任务时可以选择是否立即执行，选择立即执行提交后，该巡检任务会立即执行



7 删除任务

点击巡检任务的【删除】按钮即可删除选中的巡检任务，删除后无法恢复，请谨慎操作，巡检任务已经执行生成的巡检报告不会被删除。（注：系统初始的默认巡检任务无法删除）

查看巡检结果

从【巡检平台】-【巡检报告】进入，巡检任务的执行结果会以巡检报告的形式全部展示出来，可直接在前端页面查看巡检结果，或者下载巡检报告查看巡检结果。

1 巡检报告列表

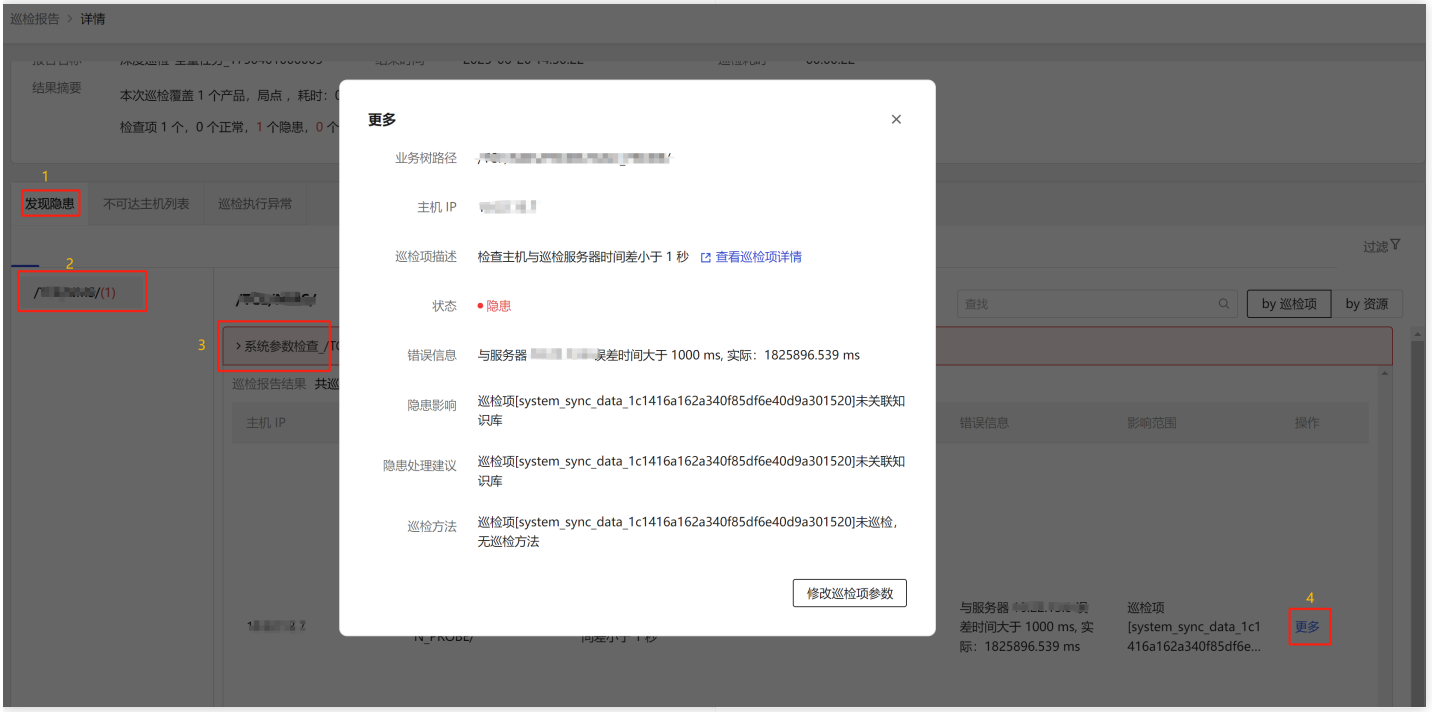
可查看巡检报告列表数据，包含报告名称、巡检覆盖范围、发现隐患比、结果摘要、完成时间、巡检耗时；在页面上可根据巡检任务、局点、巡检产品、时间搜索条件查询数据。

2 巡检报告详情页面

点击巡检报告列表中的【报告名称】进入该巡检报告详情页面，页面展示当前巡检摘要、发现隐患、不可达主机列表、巡检执行异常。

(1)发现隐患

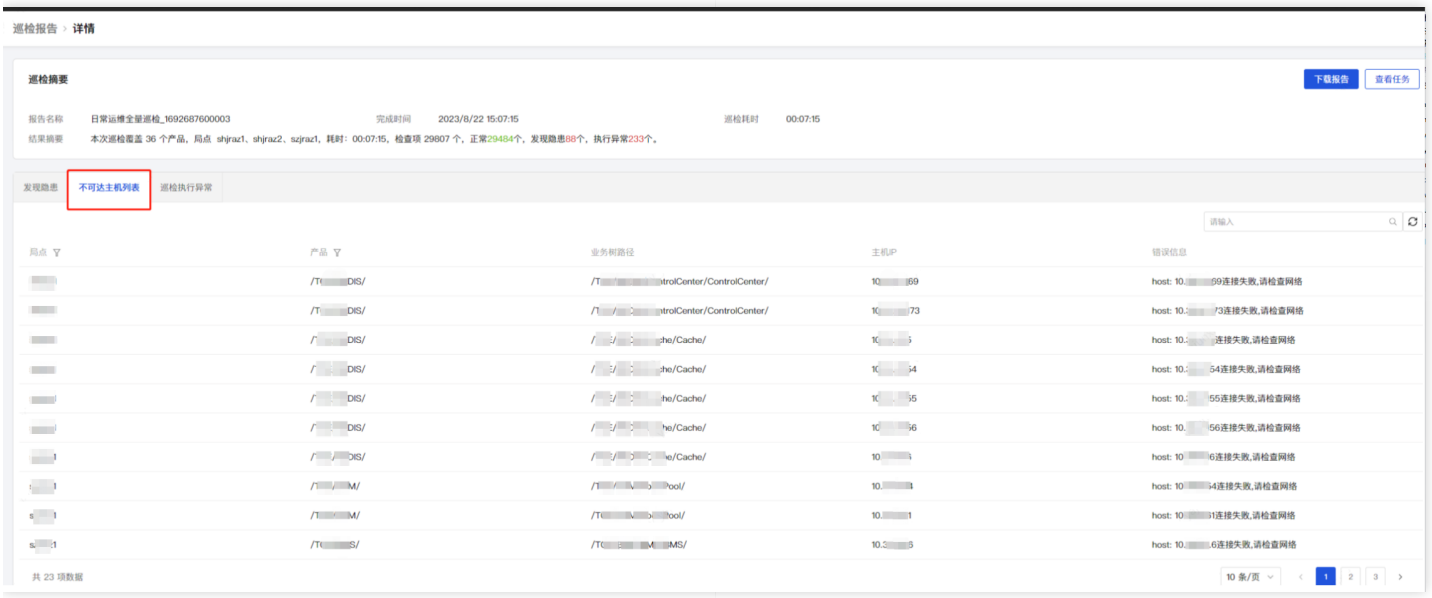
当前页面的产品名称后出现不为0的红色数量时，说明当前巡检任务发现被巡检产品存在隐患，可点击对应【产品】，展开红色的巡检组，点击异常状态巡检项的【更多】，可查看详细的巡检结果内容。



巡检报告详情页面可以通过勾选【巡检结果状态】和【巡检项分级】，按条件进行过滤。

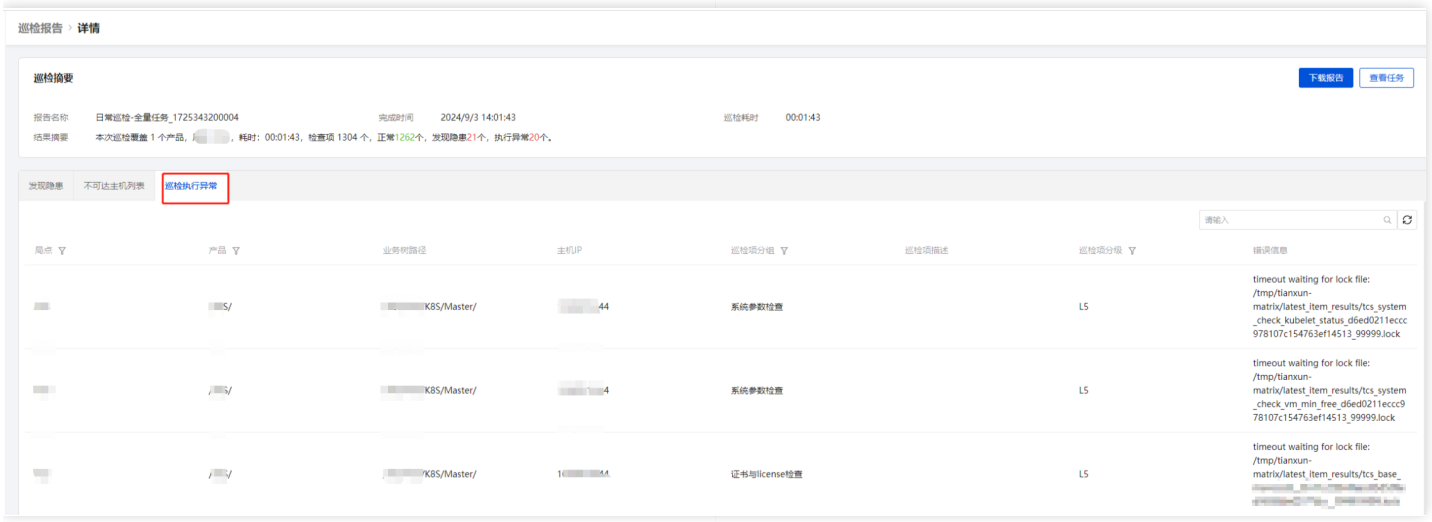
(2)不可达主机列表

点击巡检报告详情页面的【不可达主机列表】，该列表展示当前巡检任务未连接成功的主机信息，根据列表中的错误信息提示解决，检查网络或免密配置信息等。



(3)巡检执行异常

点击巡检报告详情页面的【巡检执行异常项】，该列表中汇总当前巡检报告中，拉取主机失败或巡执行失败的巡检项。



3 下载巡检报告

进入巡检报表页面，点击报告格式，即可下载相应格式的巡检报告。


巡检平台
巡检报告
巡检控制台
巡检报告
巡检任务
巡检项管理
操作记录
巡检报告
请选择巡检任务
请选择局点
请选择产品
请选择日期
-
请选择日期
报告名称
覆盖范围
发现隐患比
结果摘要
结束时间
巡检耗时
下载报告
深度巡检-全量任务_1750386600004
1/1
巡检产品 1 个，检查项 1 个，0 个正常，发现隐患 1 个。
2025-06-20 10:30:22
00:00:22
报告格式
EXCEL
JSON
DOC
运营端监控完整性-监控数据检验_1750384800008
天巡
0/0
巡检产品 0 个，检查项 0 个，0 个正常，发现隐患 0 个。
2025-06-20 10:00:10
00:00:10
日常巡检-全量任务_1750384800007
1/107
巡检产品 6 个，检查项 107 个，105 个正常，发现隐患 1 个。
2025-06-20 10:00:52
00:00:52

TCENTER 巡检项清单

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
check_ntp_hwclock	检查hwclock硬件时钟偏差是否超过10秒	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/Zookeeper/instance TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/TDSQL/Instance TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
drms_tce-tcenter-hdfs_check_enable_topology_4f970b7a	检查机架感知是否开启	容灾检查	容灾检查	TCENTER/HDFS/namenode
drms_tce-tcenter-k8s_check_apiserver_39c40585	检查 apiserver 的 rdata 配置	容灾检查(单机)	容灾检查	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_cluster_global_dnsnameservers_178906da	检查 cluster global 配置中的 dnsNameservers	容灾检查(单机)	容灾检查	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_coredns_c728ad1d	检查 coredns 的健康检查配置	容灾检查(单机)	容灾检查	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_kube_dns_d5dbbfbf	检查kube DNS	容灾检查(单机)	容灾检查	TCENTER/K8S/Master
drms_tce-tcenter-k8s_check_registry_1bd407c1	检查 tcs-internal-registry	容灾检查(单机)	容灾检查	TCENTER/K8S/Master
system_cpu_usage	检查 CPU 使用率高于90%	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_disk_mount	检查磁盘正确挂载	系统参数检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_io_usage	检查 IO 使用率低于 90%	系统参数检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Master TCENTER/ ImgCache/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_load_average	检查最近1分钟/5分钟/15分钟系统平均负载是否小于CPU核数	系统参数检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_mem_usage	检查内存使用率小于 80%	系统参数检查	日常运维,深度巡检	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
system_partition_usage	检查所有挂载分区使用率小于 80%	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
system_sync_data	检查主机与巡检服务器时间差小于 1 秒	系统参数检查	日常运维,深度巡检	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_barad_es_ping	检查Elasticsearch节点ping延迟小于 500 ms	依赖服务检查	日常运维	TCENTER/Elasticsearch/etcd
tx_barad_zk_config_state	检查Zookeeper集群节点配置	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_barad_zk_node_count	检查Zookeeper集群kafka目录数为3	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_barad_zk_node_exist	检查Zookeeper集群/目录下存在有storm110，kafka，kfkSpout目录	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_barad_zk_partition_usage	检查Zookeeper集群磁盘 / 和/data 使用率是否小于阈值	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_barad_zk_snapshot_state	检查Zookeeper集群快照大小	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_barad_zk_state	检查Zookeeper集群节点状态	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_check_mysql_status_cam_auth	TCenter cam auth组件 数据库检查	脏数据检查	日常运维	TCENTER/k8s-master-ip

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_check_mysql_status_cam_grant	TCenter cam grant组件 数据库检查	脏数据检查	日常运维	TCENTER/k8s-master-ip
tx_check_mysql_status_cam_list	TCenter cam list组件 数据库检查	脏数据检查	日常运维	TCENTER/k8s-master-ip
tx_check_mysql_status_cam_sts	TCenter cam sts组件 数据库检查	脏数据检查	日常运维	TCENTER/k8s-master-ip
tx_check_mysql_status_platform_account	TCenter platform account组件 数据库检查	脏数据检查	日常运维	TCENTER/k8s-master-ip
tx_get_svc-ignore-sync-ip	检查svc中是否配置了infra.tce.io和ignore-sync-ip	安全类检查	日常运维	TCENTER/K8S/Master
tx_kafka_check_replicationfactor	检查物理机kafka副本数不为1	服务状态检查	日常运维	TCENTER/Kafka/kafka
tx_pass_tmp_safe	用户root和tunnel_user密码已入库且密码风险检查	安全类检查	日常运维	TCENTER/K8S/Master
tx_Redis®_base_cc_sql_interface_balance_status	检查newcc库中interfce分布不均	脏数据检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_interface_procs_capacity	检查newcc库中interface使用量高于80%	脏数据检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_Redis®_procs_capacity	检查newcc库中Redis®进程使用量高于80%	脏数据检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_base_cc_sql_status_task_state	检查newcc库中事件任务正常	脏数据检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_base_interface_status_cli_num	[采集]interface的连接数高于5000	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_latency	[采集]interface的时延小于200ms	机房网络检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_maxfd	[报警]interface fd最大limit配置大于10000	系统参数检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_mem	[采集]interface的内存大于5G	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_rafficlimit	[报警]interface出现限流limit日志	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_slowlog	[报警]interface 1小时内高于50条慢查询	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_stats_avg_latency_6	[报警]interface 单进程请求耗时超过500ms	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_stats_clirates	[报警]interface 单进程连接数利用率超过80%	容量检查	日常运维	TCENTER/CRedis/interface

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_Redis®_base_interface_status_stats_input_limit_rate	[报警]interface 单进程输入带宽利用率超过80%	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_stats_output_limit_rate	[报警]interface 单进程输出带宽利用率超过80%	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_interface_status_stats_qps	[报警]interface 单进程QPS大于50000	容量检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_base_server_status_check_client_biggest_input_buf_status	[采集]Redis®输入缓冲区最大buffer大于100Mb	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_client_longest_output_list_status	[采集]Redis®输出缓冲区对象个数大于1000	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_connected_clients_ratio_status	[报警]Redis®的clients链接数高于上限50%	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_input_kbps_status	[采集]Redis®的输入bps高于500Mb	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_ops_per_sec_status	[采集]Redis®的请求qps高于6万	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_instantaneous_output_kbps_status	[采集]Redis®的输出bps高于500Mb	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_latest_fork_usec_status	[采集]Redis®的fork时延高于1s	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_logdiskbusy_status	[报警]Redis®的落盘出现busy阻塞	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_master_aof_enable	[报警]Redis®的master实例开启AOF,有性能影响	服务配置检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_Redis®_limit_maxfd	[报警]Redis®的limit配置fd限制大于10000	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_repl_offset_status	[报警]Redis®的主从偏移量大于10Mb	服务配置检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_slowlog_status	[采集]Redis®的慢查询高于20条	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_base_server_status_check_used_memory_status	[采集]Redis®的内存使用量高于30G	容量检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_cc_agent_check_cc_agent_timer	Redis® cc 依赖agent进程存活检查-cc_agent_timer	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_cc_monitor_py	Redis® cc 依赖agent进程存活检查-cc_monitor_py	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_Redis®_cc_agent_check_defunct_proc	Redis® cc cc_monitor_py假死	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_influxdb	Redis® cc 依赖agent进程存活检查-influxd	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_mul_server	Redis® cc 依赖agent进程存活检查-mul_server	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_qds_center	Redis® cc 依赖agent进程存活检查-qds_center	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_check_web	Redis® cc 依赖agent进程存活检查-web	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_agent_version_statics_cc_monitor_py	[采集]获取cc管控cc_monitor_py版本	进程检查	日常运维	TCENTER/Product/product-tcenter-support-cRedis®/oss_nodes
tx_Redis®_cc_sql_status_app_nums	检查newcc库中实例存在脏实例数据	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_app_state	检查newcc库中实例状态正常	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_interface_machine_status	检查newcc库中interface机器状态出现非1	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_interface_proc_status	检查newcc库中interface进程状态出现非1	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_machine_salerate	检查newcc库中机器售卖率高于80%	数据库检查	日常运维	TCENTER/CRedis/oss

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_Redis®_cc_sql_status_master_port	检查newcc库中master进程的端口配置正常	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_master_slave	检查newcc库中Redis®_procs_t中主从不一致进程	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_Redis®_machine_status	检查newcc库中Redis®机器状态出现非1	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_Redis®_proc_status	检查newcc库中Redis®进程状态出现非1	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_replication_check	检查newcc库中副本数量一致性检查	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_cc_sql_status_scale_check	检查newcc库中分片数量一致性检查	数据库检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_check_agent_status	检查是否存在 agent 未拉起	服务状态检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_check_cluster_fail_nodes_status	检查 cluster 是否存在 fail状态未forget成功节点	服务状态检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_interface_version_statics	[采集]interface版本信息	服务状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_hwcheck_disk	检查最近30分钟内日志中存在磁盘异常日志	硬件状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_hwcheck_eth	检查最近30分钟内日志中存在有网卡异常日志	硬件状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_hwcheck_mem	检查最近30分钟内日志中是存在内存异常日志	硬件状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_hwcheck_nvme	检查最近30分钟内日志中存在有nvme异常日志	硬件状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_hwcheck_raid	检查最近30分钟内日志中存在有raid卡异常日志	硬件状态检查	日常运维	TCENTER/CRedis/interface
tx_Redis®_machine_survival_status	检查机器存活状态	服务状态检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_master_slave_consistency	检查是否存在主从一致性脏数据	服务状态检查	日常运维	TCENTER/CRedis/oss
tx_Redis®_server_status_cluster_fail_nodes	[报警]Redis®的cluster中存在fail状态节点	服务状态检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_server_status_cluster_slots_ok	[报警]Redis®的cluster槽位状态异常	服务状态检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_server_status_cluster_state	[报警]Redis®的cluster info状态异常	服务状态检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_server_status_memused_rate	[报警]Redis®的内存使用率大于80%	服务状态检查	日常运维	TCENTER/CRedis/cache

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_Redis®_server_status_stand_proc	[报警]Redis®存在单点进程	服务状态检查	日常运维	TCENTER/CRedis/cache
tx_Redis®_server_version_statics	[采集]Redis®的cache进程版本采集	服务状态检查	日常运维	TCENTER/CRedis/cache
tx_tcenter_ckv_check_port	检查 ckv 端口存在	依赖服务检查	日常运维	TCENTER/CRedis/cache
tx_tcenter_ckv_check_time	检查 ckv 时间正常	依赖服务检查	日常运维	TCENTER/CRedis/cache
tx_tcenter_csp_mgmt_pod_nginx	pod csp-pod-mgmt 检查nginx进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mgmt_pod_pm2	pod csp-pod-mgmt 检查PM2进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_ceph_mgr	检查ceph-mgr进程运行正常	服务状态检查	日常运维	TCENTER/CSP/monitor_hosts
tx_tcenter_csp_mon_ceph_mon	检查ceph-mon进程运行正常	服务状态检查	日常运维	TCENTER/CSP/monitor_hosts
tx_tcenter_csp_mon_moira_server	检查moira server进程运行正常	服务状态检查	日常运维	TCENTER/CSP/monitor_hosts
tx_tcenter_csp_mon_moira_server_api	检查moira server api服务正常	服务状态检查	日常运维	TCENTER/CSP/monitor_hosts
tx_tcenter_csp_mon_pod_ceph_mgr	pod csp-pod-mon 检查ceph-mgr进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_ceph_mon	pod csp-pod-mon 检查ceph-mon进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_moira_server	pod csp-pod-mon 检查moira server进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_moira_server_api	pod csp-pod-mon 检查moira server api服务正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_node_exporter	pod csp-pod-mon 检查node_exporter进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_mon_pod_prometheus	pod csp-pod-mon 检查prometheus进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_moira_agent	检查moira agent运行正常	服务状态检查	日常运维	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_nginx	检查csp_nginx运行正常	服务状态检查	日常运维	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_pod_moira_agent	pod csp-pod-rgw 检查moira agent运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_nginx	pod csp-pod-rgw 检查csp_nginx运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_node_exporter	pod csp-pod-rgw 检查node_exporter进程运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_rgw_pod_rgw_api	pod csp-pod-rgw 检查rgw api服务正常	容器检查	日常运维	TCENTER/k8s-master-ip

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_csp_rgw_radosgw	检查rgw进程运行正常	服务状态检查	日常运维	TCENTER/CSP/gw_hosts
tx_tcenter_csp_rgw_rgw_api	检查rgw api服务正常	服务状态检查	日常运维	TCENTER/CSP/gw_hosts
tx_tcenter_csp_store_ambari_agent	检查ambari-agent服务运行正常	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_ambari_agent_crontab	检查crontab中配置ambari-agent定时任务	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_down	检查ceph osd没有down	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_health	检查集群状态健康	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_size	检查csp集群的副本数	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_ceph_usage	检查csp集群的使用率	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_crond	检查crond运行正常	服务状态检查	日常运维	TCENTER/CSP/ips
tx_tcenter_csp_store_pod_ambari_agent	pod csp-pod-osd 检查ambari-agent服务运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_ambari_agent_crontab	pod csp-pod-osd 检查crontab中配置ambari-agent定时任务	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_ceph_down	pod csp-pod-osd 检查ceph osd没有down	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_ceph_health	pod csp-pod-osd 检查集群状态健康	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_csp_store_pod_crond	pod csp-pod-osd 检查crond运行正常	容器检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_es_allocation_oss	检查Elasticsearch集群磁盘使用率未超过阈值	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_breaker_oss	检查熔断器 breaker 参数设置是否 < 80%	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_check_agent	agent 进程运行正常	进程检查	日常运维,深度巡检	TCENTER/Elasticsearch/data
tx_tcenter_es_check_api	api能够正常访问	API检查	日常运维,深度巡检	TCENTER/Elasticsearch/oss
tx_tcenter_es_check_ces_node_info	检查.ces_node_info文件3天没有更新	服务配置检查	日常运维	TCENTER/Elasticsearch/data

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_es_check_etcd_cluster_health	检查etcd集群健康	容器检查	日常运维,深度巡检	TCENTER/K8S/Master
tx_tcenter_es_check_etcd_cluster_health	检查etcd集群健康	服务状态检查	日常运维,深度巡检	TCENTER/Elasticsearch/etcd
tx_tcenter_es_check_master	master 进程运行正常	进程检查	日常运维,深度巡检	TCENTER/Elasticsearch/oss
tx_tcenter_es_check_master_cron	master计划任务存在	服务配置检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_check_node_status	每个节点正常	服务状态检查	日常运维	TCENTER/Elasticsearch/etcd
tx_tcenter_es_cluster_each_shards_oss	检查Elasticsearch集群磁盘使用率未超过阈值	服务状态检查	日常运维,深度巡检	TCENTER/Elasticsearch/oss
tx_tcenter_es_cluster_total_shards_oss	检查Elasticsearch集群磁盘使用率未超过阈值	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_health_oss	检查Elasticsearch集群健康状态为green	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_jvm_mem_oss	检查Elasticsearch集群JVM内存使用率未超过阈值	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_node_oss	检查Elasticsearch集群健康状态为green	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_es_shards_store_oss	检查支撑ES节点单个分片小于50g	服务状态检查	日常运维	TCENTER/Elasticsearch/oss
tx_tcenter_etcd_dbsize	检查dbSize小于1.6g	服务状态检查	日常运维	TCENTER/K8S/Master TCENTER/K8S/Etcd
tx_tcenter_etcd_member_status	检查member状态正常	服务状态检查	日常运维	TCENTER/K8S/Etcd
tx_tcenter_hdfs_check_datanode_cluster_state	测试 hdfs datanodes 集群正常	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_hdfs_check_dfs_usage	检测 hdfs 集群 DFS 使用率小于 80%	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_kinit_state	检查hdfs是否kinit执行成功	服务状态检查	日常运维	TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_missing_blocks	检测 hdfs 集群不存在 Missing Blocks	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_namenode_cluster_state	测试 hdfs namenodes 集群正常	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_read_state	测试 hdfs 读正常	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_safe_mode	检测hdfs是否进入安全模式	系统参数检查	日常运维,深度巡检	TCENTER/HDFS/namenode TCENTER/HDFS/datanode
tx_tcenter_hdfs_check_write_state	测试 hdfs 写正常	服务状态检查	日常运维,深度巡检	TCENTER/HDFS/datanode
tx_tcenter_hdfs_crontab_check_hdfs_journalnode	检测hdfs-jn计划任务	服务配置检查	日常运维	TCENTER/HDFS/journalnode
tx_tcenter_hdfs_crontab_check_kinit	检查hdfs是否配置kinit计划任务	服务状态检查	日常运维	TCENTER/HDFS/journalnode
tx_tcenter_hdfs_kerberos	kerberos进程检查	服务状态检查	日常运维	TCENTER/HDFS/namenode
tx_tcenter_hdfs_keytable	keytabe检查	服务状态检查	日常运维	TCENTER/HDFS/namenode
tx_tcenter_hdfs_keytable_check_node	检查所有节点的keytab文件	服务状态检查	日常运维	TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				datanode
tx_tcenter_hdfs_process_check_journalnode	检测hdfs-jn进程正常	进程检查	日常运维	TCENTER/HDFS/journalnode
tx_tcenter_k8s_apiserver-etcd-client_expire	apiserver-etcd-client证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_apiserver-kubelet-client_expire	apiserver-kubelet-client证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_apiserver-loopback-client_expire	apiserver-loopback-client证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_apisever_expire	apisever证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_ca_expire	ca证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_cert-manager_expire	cert-manager证书将在45天内过期	证书与license检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_k8s_etcd-ca_expire	etcd-ca证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Etcd
tx_tcenter_k8s_etcd-peer_expire	etcd-peer证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Etcd
tx_tcenter_k8s_etcd-server_expire	etcd-server证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Etcd
tx_tcenter_k8s_kubelet-ca_expire	kubelet-ca证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_kubelet-client_expire	kubelet-client证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_kubelet_expire	kubelet证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_mtu_check	mtu没有风险	服务配置检查	日常运维,深度巡检	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_k8s_ocloud-tcenter-base-vpc-dns_analysis	ocloud-tcenter-base-vpc-dns能够解析域名	服务配置检查	日常运维	TCENTER/k8s-master-ip

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_k8s_registry-ca_expire	registry-ca证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_k8s_registry_expire	registry证书将在45天内过期	证书与license检查	日常运维	TCENTER/K8S/Master
tx_tcenter_kafka_check_data_size	kafka 检查	脏数据检查	日常运维	TCENTER/Kafka/kafka
tx_tcenter_kafka_check_port	kafka 检查	端口检查	日常运维	TCENTER/Kafka/kafka
tx_tcenter_kafka_check_topic	kafka 检查	脏数据检查	日常运维	TCENTER/Kafka/kafka
tx_tcenter_kafka_partition_leader	kafka的partition leader不为-1	服务状态检查	日常运维	TCENTER/Kafka/kafka
tx_tcenter_kubelet_check_docker	检查docker进程状态	进程检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_kubelet_check_kubelet_status	检查kubelet进程状态	进程检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_kubelet_tcs_cgroup_memory_nokmem	检查tcs所有节点完成cgroup.memory=nokmem配置	服务配置检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcenter_mongo_cluster_config_state	mongo 集群检查	脏数据检查	日常运维	TCENTER/MongoDB/instance
tx_tcenter_mongo_cluster_read_state	mongo 集群检查	脏数据检查	日常运维	TCENTER/MongoDB/instance
tx_tcenter_mongo_cluster_shard_state	mongo 集群检查	脏数据检查	日常运维	TCENTER/MongoDB/instance
tx_tcenter_mongo_cluster_write_state	mongo 集群检查	脏数据检查	日常运维	TCENTER/MongoDB/instance
tx_tcenter_mq_check_cluster_state	集群状态正常	服务状态检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_check_cron	检查mq的定时任务	服务状态检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_check_cron	检查mq的定时任务	服务配置检查	日常运维	TCENTER/Product/product-tcenter-support-mq/mq_nodes

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_mq_check_list_queues_state	mq list_queues 状态正常	服务状态检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_cluster_brainsplit	mq 集群状态没有出现脑裂	服务状态检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_report_state	mq节点检查	服务状态检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_report_state	mq节点检查	脏数据检查	日常运维	TCENTER/Product/product-tcenter-support-mq/mq_nodes
tx_tcenter_mq_state	mq节点检查	服务状态检查	日常运维,深度巡检	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_mq_state	mq节点检查	脏数据检查	日常运维,深度巡检	TCENTER/Product/product-tcenter-support-mq/mq_nodes
tx_tcenter_mq_version_check_3100	云平台物理支撑rabbitmq版本检查	服务配置检查	日常运维	TCENTER/RabbitMQ/RabbitMQ
tx_tcenter_ntp_check_crontab_ntpdate	检查crontab中没有配置ntpdate定时任务	时间服务检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_ntp_check_ntpd_is_enabled	检查ntpd自启动脚本正常	时间服务检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_ntp_check_ntpd_offset	检查ntpd上次同步offset在100ms以内	时间服务检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ journalnode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/ gw_hosts TCENTER/ CRedis/oss TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_ntp_check_ntpd_when	检查ntpd上次同步时间在1024s以内	时间服务检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_ntp_check_process_ntpd_and_chrony	检查是否使用ntpd或chrony	时间服务检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/journalnode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/oss TCENTER/CRedis/interface TCENTER/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				CRedis/cache
tx_tcenter_pod_workload_node_check	检查k8s集群中workload是否有node重叠	容器检查	日常运维	TCENTER/K8S/Master
tx_tcenter_system_check_defunct_process	检查服务器-僵尸进程数小于50	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_cpu_idle	检查服务器-cpuidle大于30%	系统参数检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips TCENTER/CSP/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				gw_hosts TCENTER/ CRedis/ interface TCENTER/ CRedis/cache
tx_tcenter_system_disk_read_write	检查服务器磁盘读写可用	系统参数检查	日常运维	TCENTER/ Zookeeper/ zookeeper TCENTER/ TDSQL/ scheduler TCENTER/ TDSQL/proxy TCENTER/ TDSQL/oss TCENTER/ TDSQL/ monitor TCENTER/ TDSQL/db TCENTER/ TDSQL/chitu TCENTER/ RabbitMQ/ RabbitMQ TCENTER/ Product/ product- tcenter- support-mq/ mq_nodes TCENTER/ Kafka/kafka TCENTER/K8S/ Node TCENTER/K8S/ Master TCENTER/ ImgCache/ imgcache TCENTER/ HDFS/ namenode TCENTER/ HDFS/ datanode TCENTER/ Elasticsearch/ oss TCENTER/ Elasticsearch/ etcd TCENTER/ Elasticsearch/ data TCENTER/CSP/ web_hosts TCENTER/CSP/ monitor_hosts TCENTER/CSP/ ips

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_fd_usage	检查服务器-fd用量低于80%	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_inode_usage	检查服务器-inode使用率低于80%	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_netdev_recv_pkg	检查服务器-网卡入包量小于5G	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts TCENTER/CSP/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_netdev_send_pkg	检查服务器-网卡出包量小于5G	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/web_hosts

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_network_sockets	检查服务器-网络socket链接小于20万	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data TCENTER/CSP/

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_tcenter_io_await	检查服务器-ioawait小于1s	系统参数检查	日常运维	TCENTER/Zookeeper/zookeeper TCENTER/TDSQL/scheduler TCENTER/TDSQL/proxy TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/db TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Node TCENTER/K8S/Master TCENTER/ImgCache/imgcache TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/Elasticsearch/oss TCENTER/Elasticsearch/etcd TCENTER/Elasticsearch/data

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
				TCENTER/CSP/web_hosts TCENTER/CSP/monitor_hosts TCENTER/CSP/ips TCENTER/CSP/gw_hosts TCENTER/CRedis/interface TCENTER/CRedis/cache
tx_tcenter_system_tcenter_io_util	检查服务器-ioutil小于60%	系统参数检查	日常运维	TCENTER/TDSQL/scheduler TCENTER/TDSQL/oss TCENTER/TDSQL/monitor TCENTER/TDSQL/chitu TCENTER/RabbitMQ/RabbitMQ TCENTER/Product/product-tcenter-support-mq/mq_nodes TCENTER/Kafka/kafka TCENTER/K8S/Master TCENTER/HDFS/namenode TCENTER/HDFS/datanode TCENTER/CSP/monitor_hosts TCENTER/CSP/gw_hosts
tx_tcenter_tcs_cgroup_memory_nokmem_status	检查tcs所有节点完成cgroup.memory=nokmem生效	服务配置检查	日常运维	TCENTER/K8S/Master
tx_tcenter_tcs_check_node_mem_cpu	检查node的cpu/mem分配率是否高于90%	容器检查	日常运维	TCENTER/K8S/Master
tx_tcenter_tcs_check_node_mem_cpu	检查node的cpu/mem分配率是否高于90%	服务状态检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_tcs_check_node_status	检查node是否ready	容器检查	日常运维,深度巡检	TCENTER/K8S/Master

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_tcs_check_node_status	检查node是否ready	服务状态检查	日常运维,深度巡检	TCENTER/k8s-master-ip
tx_tcenter_tcs_master_disk_usage	检查磁盘使用率是否小于60%	进程检查	日常运维	TCENTER/K8S/Master
tx_tcenter_tdsql_check_proc	检查ewp进程&计划任务配置	进程检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_ioutil	检查磁盘的util值是否超过60%	容量检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_load_average	检查最近1分钟/5分钟/15分钟系统平均负载是否小于CPU核数	容量检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_login	判断用户tdsql是否存在&tdsql是否可以登录	系统参数检查	日常运维	TCENTER/TDSQL/Instance
tx_tcenter_tdsql_mem_swap_usage	检查swap使用是否超过60%	容量检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_mem_usage	检查内存使用率是否超过阈值	容量检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_partition_usage	检查 /和/data 挂载分区使用率是否小于阈值	容量检查	日常运维	TCENTER/TDSQL/oss TCENTER/TDSQL/Instance
tx_tcenter_tdsql_rc_local	rc.local是否有执行权限检查	系统参数检查	日常运维	TCENTER/TDSQL/Instance
tx_tcenter_tke_check_cls_fail_by_k8s_api	检查是否有创建失败或长期创建中的tke集群	服务状态检查	日常运维	TCENTER/k8s-master-ip
tx_tcenter_vpcdns_dig_pod	检查 base-vpc-dns 域名解析	服务状态检查	日常运维	TCENTER/KubeResource/ocloud-tcenter-base-vpc-dns
tx_tcenter_zk_check_conn_num	zk连接数检查是否小于500	依赖服务检查	日常运维	TCENTER/Zookeeper/instance

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_zk_check_conn_num	zk连接数检查是否小于500	服务状态检查	日常运维	TCENTER/Zookeeper/zookeeper
tx_tcenter_zk_check_status	zk状态检查	依赖服务检查	日常运维,深度巡检	TCENTER/Zookeeper/instance
tx_tcenter_zk_check_zk_outstanding_requests	zk排队请求数量是否超过10	依赖服务检查	日常运维,深度巡检	TCENTER/Zookeeper/instance
tx_tcenter_zk_check_zk_outstanding_requests	zk排队请求数量是否超过10	服务状态检查	日常运维,深度巡检	TCENTER/Zookeeper/zookeeper
tx_tcenter_zk_check_znode_num	znode数量检查超过10W	依赖服务检查	日常运维,深度巡检	TCENTER/Zookeeper/instance
tx_tcenter_zk_check_znode_num	znode数量检查超过10W	服务状态检查	日常运维,深度巡检	TCENTER/Zookeeper/zookeeper
tx_tcenter_zk_config	检查Zookeeper集群节点配置	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_tcenter_zk_multi_version	检查Zookeeper集群是否有多个版本混部	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_tcenter_zk_no_sync_data	zk判断未同步数据是否大于10	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_tcenter_zk_no_sync_data	zk判断未同步数据是否大于10	服务状态检查	日常运维	TCENTER/Zookeeper/zookeeper
tx_tcenter_zk_partition_usage	检查Zookeeper集群磁盘 / 和/data 使用率是否小于阈值	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_tcenter_zk_snapshot	检查Zookeeper集群快照大小	依赖服务检查	日常运维	TCENTER/Zookeeper/instance
tx_tcenter_zk_snapshot_size	zk快照体积是否超过1G	依赖服务检查	日常运维,深度巡检	TCENTER/Zookeeper/instance

巡检项ID	巡检项描述	巡检项分组	分类	业务树路径
tx_tcenter_zk_snapshot_size	zk快照体积是否超过1G	服务状态检查	日常运维,深度巡检	TCENTER/Zookeeper/zookeeper
tx_tcenter_zk_state	检查Zookeeper集群节点状态	依赖服务检查	日常运维,深度巡检	TCENTER/Zookeeper/instance
tx_tcs_check_pajero	检查 pajero 运行状态	服务配置检查	日常运维,深度巡检	TCENTER/K8S/Master
tx_tcs_check_pvc_usage	检查是否存在 pvc 使用率超过 85%	服务配置检查	日常运维	TCENTER/K8S/Node TCENTER/K8S/Master
tx_tcs_check_tunl_vip	检查 tcs 隧道完整性	服务配置检查	日常运维,深度巡检	TCENTER/K8S/Node
tx_tcs_dig_underlay	检查 tcs underlay 域名解析	进程检查	日常运维	TCENTER/K8S/Master

巡检处理
巡检项1：基础平台功能巡检

操作场景

tcenter部分的组件都在容器中运，且部分容器身都有健康探测机制。所以先确认容器的运是否健康很重要。

前提条件

已获取租户端控制台及运营端控制台登录地址和账号密码。

操作步骤

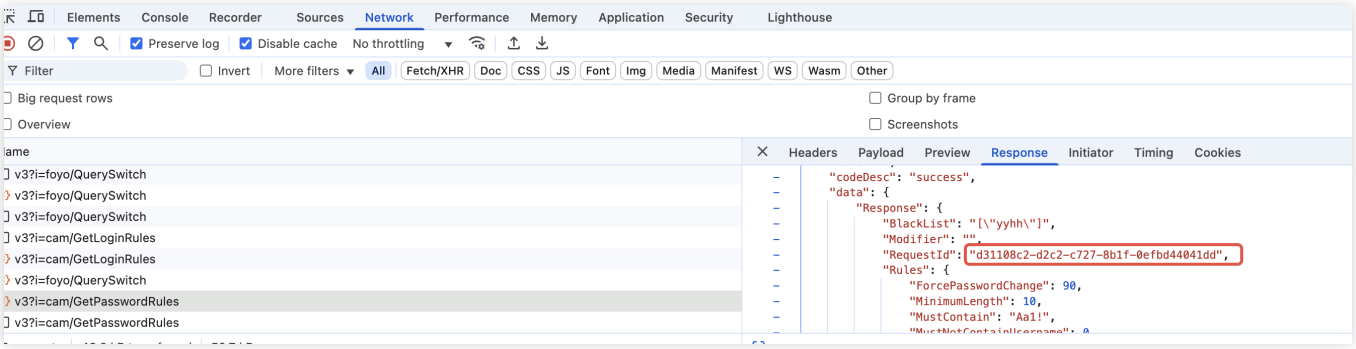
1. 登录租户端控制台，确认主页面无报错
2. 选择个人信息 > 账户中心/访问管理/安全设置，进入基础平台租户端页面，在左侧栏中依次单击所有子项等，查看页面是否正常展示。
3. 登录运营端控制台，确认主页面无报错
4. 选择平台管理 > 用户与权限/云API管理/CAM管理，进入基础平台运营端页面，在左侧栏中依次单击所有子项等，查看页面是否正常展示。
   如果存在页面显示异常、页面出现报错的情况，请参考异常处理来处理。

异常处理

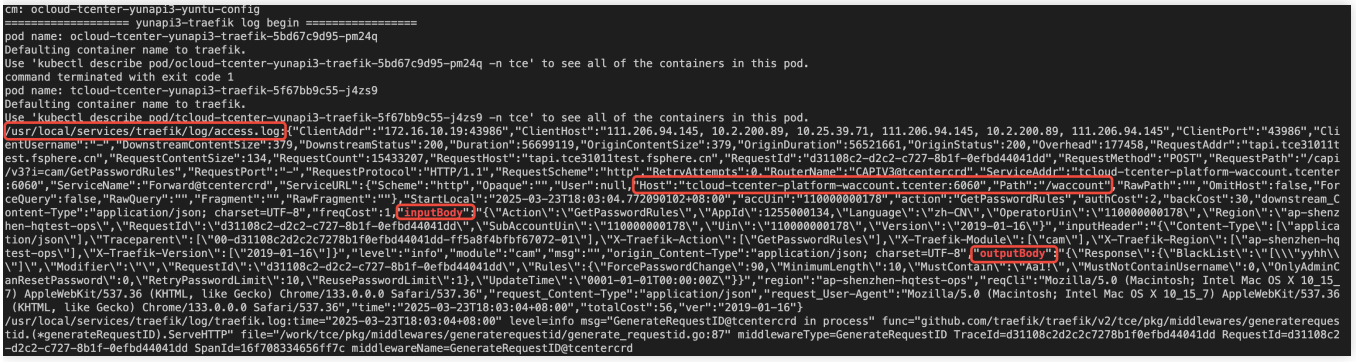
1. 检查服务pod状态

```
# 检查pod是否正常拉起，pod状态是否正常
kubectl get pod -n tce | grep cloud-tcenter
# 由于tcenter服务均为无状态服务，若存在异常状态pod，可通过delete命令快速恢复，此处以ocloud-tcenter-identityaccess为例
kubectl delete pod -n tce ocloud-tcenter-identityaccess-xxx
```

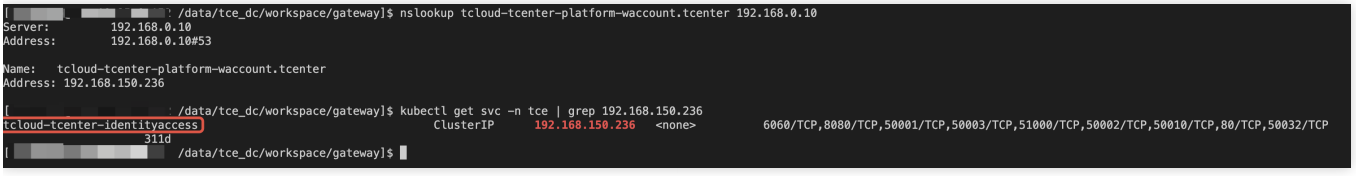
1. 查看服务日志
   由于涉及页面较多，且几乎均为容器化组件提供服务，均通过云API向后转发请求，故提供通用排查方案



获取云API日志搜索脚本
mkdir -p /data/tce_dc/workspace/gateway/
cd /data/tce_dc/workspace/gateway/
kubectl -ntce cp `kubectl get pod -n tce | grep ocloud-tcenter-yunapi3-yuntu | awk '{print $1}' | tail -n 1`:/usr/local/services/yuntu/bin/traefik_tool.sh /data/tce_dc/workspace/gateway/traefik_tool.sh
chmod +x traefik_tool.sh
./traefik_tool.sh log ${reqid}

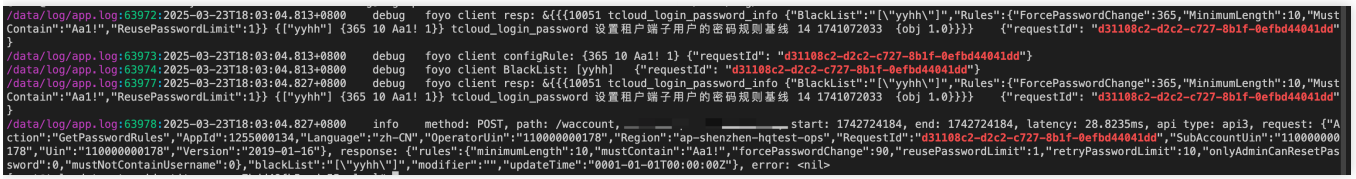


截图中红框部分需要关注的主要为
1）请求后端地址（tcloud-tcenter-platform-waccount.tcenter:6060/waccount）
2）入参（{"Action":"GetPasswordRules","AppId":1255000134,"Language":"zh-CN","OperatorUin":"110000000178","Region":"ap-hqtest-ops","RequestId":"d31108c2-d2c2-c727-8b1f-0efbd44041dd","SubAccountUin":"110000000178","Uin":"110000000178","Version":"2019-01-16"}）
进一步排查日志方式需进入后端业务容器查看，例如



nslookup tcloud-tcenter-platform-waccount.tcenter 192.168.0.10
kubectl get svc -n tce | grep 192.168.150.236
kubectl get pod -n tce | grep tcloud-tcenter-identityaccess

kubectl exec -it -n tce tcloud-tcenter-identityaccess-xxxx bash
grep -rn ${reqid} /data/log/



巡检项2：组件状态巡检

操作场景

tcenter部分的组件都在容器中运，且部分容器身都有健康探测机制。所以先确认容器的运是否健康很重要。

前提条件

可登录k8s集群节点，并能执行kubectl命令。

操作步骤

1. 登录k8s集群节点，执行如下命令确认服务状态是否正常。

```
# 检查pod是否正常拉起，pod状态是否正常
kubectl get pod -n tce | grep cloud-tcenter
```

异常处理

若存在容器运行状态异常，可通过如下方式确认异常原因。

```
# 此处以tcloud-tcenter-console-web-xx为例
kubectl describe pod -n tce tcloud-tcenter-console-web-xxx | grep -A 5  -E 'Last State:|Events:'
```

```
    Last State:     Terminated
      Reason:       Completed
      Exit Code:    0
      Started:      Sun, 23 Mar 2025 18:24:17 +0800
      Finished:     Sun, 23 Mar 2025 18:25:47 +0800
    Ready:          False
--
Events:
  Type     Reason     Age                  From     Message
  ----     ------     ----                 ----     -------
  Warning  Unhealthy  13m (x6959 over 4d)  kubelet  (combined from similar events): Liveness probe failed: [2025-03-23T18:16:47] cwp_cloudapi healthchk exit 1
```

此处主要关注点为Last State.Reason和Exit Code，此处若容器由于资源限制问题，例如oom导致容器异常，则显示为oomkilled，code为137；图中状态非137，在云平台场景下基本均为服务自身健康探测机制失败导致，结合events部分可以进一步确认，该容器异常原因为存活探测失败。

```
# 查看容器资源限制大小，健康检查设置
kubectl describe pod -n tce tcloud-tcenter-console-web-xxx | grep -A 4 -E 'Limits|Liveness|Readiness'
```

```
    Limits:
      cpu:     4
      memory:  16G
    Requests:
      cpu:        1
--
    Liveness:     exec [/bin/bash /tce/healthchk.sh] delay=30s timeout=1s period=30s #success=1 #failure=3
    Readiness:    exec [/bin/bash /tce/healthchk.sh] delay=30s timeout=1s period=30s #success=1 #failure=3
    Environment:  <none>
```

1）若容器由于资源限制问题可通过调整组件规格大小临时解决，长期方案可提单相关技术栈确认是否为资源使用不合理或者出厂规格设置不合理，此处一般处理方案如下：

```
# 此处以tcloud-tcenter-console-web-xx为例,修改其中limits部分调整对应规则信息
# 常规k8s组件默认修改deploy、sts、ds等cr即可，云平台场景下应用采用oam模型，常规服务上层控制器均为comp
kubectl edit comp -n tce tcloud-tcenter-console-web
# 确认容器正常启动
kubectl get pod -n tce | grep tcloud-tcenter-console-web
# 确认组件历史资源使用情况，可通过登录运营端 > 监控系统 > 云产品指标 > tcs/pod > 监控指标选择pod_cpu_limit_usage/pod_memory_nocache_limit_usage,集群选择global,命名空间tce,容器选择tcloud-tcenter-console-web查看
```

2）若容器由于健康检查问题则需查看服务日志进一步确认无法启动的原因，此处一般处理方案如下，初步排查后可提单相关技术栈支持。

```
kubectl get pod -n tce | grep tcloud-tcenter-console-web
# 当容器处于0/1 Running状态的时候进入
kubectl exec -it -n tce tcloud-tcenter-console-web-xxx bash
tail -n 100 /data/log/console/console.server.log
# 若容器一直处于CrashLoopBackOff状态，也可通过kubectl logs命令尝试查看日志（需要服务支持将日志打印至标准输出）
kubectl logs -n tce tcloud-tcenter-console-web-xxx
```

日常监控
常用监控指标

指标/事件名称	分类	单位	指标描述	阈值	预案
Pod发生重启	容器监控		容器发生重启		-
Pod cpu使用率(占limit)超过阈值	容器监控	%	Pod cpu使用率较高，超过了limit设置的阈值。	95%	-
Pod 实际内存使用率(占limit)超过阈值	容器监控	%	Pod 实际内存使用率较高，超过了limit设置的阈值。	95%	-

告警处理
Pod 发生重启处理预案

告警说明

Pod发生重启：

- 基础平台业务相关容器（ocloud-osp-passwd-svr）由于某些异常原因，导致pod发生重启。
- 审计日志业务相关容器（t|ocloud-cloudaudit）由于某些异常原因，导致pod发生重启。
- CAM业务相关容器（t|ocloud-tcenter-cam/tcloud-tcenter-support-cam）由于某些异常原因，导致pod发生重启。
- 标签业务相关容器（tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag）由于某些异常原因，导致pod发生重启。
- 云API管理业务相关容器（cloud-tcenter-yunapi3）由于某些异常原因，导致pod发生重启。

告警属性

所属模块	名称	事件级别	手工清除/自动清除
tcs-kube_state指标监控默认告警策略	Pod 发生重启	2级告警	自动清除

对系统的影响

无影响。

可能原因

序号	原因	说明

序号	原因	说明
1	资源限制	- 内存限制: 此处一般伴随着（Pod 实际内存使用率(占limit)超过阈值）告警，如果容器使用的内存超过了定义的限制，Kubernetes 会杀死该容器，导致重启，此处内存不足的原因需要联系技术支持寻求服务帮助，可能原因如下： - 实际分配内存资源限制过小，需调整大小。 - 应用程序可能存在内存泄漏，导致内存使用量逐渐增加，最终超出限制，需优化程序内存回收机制。 - CPU 限制: 此处一般伴随着（Pod cpu使用率(占limit)超过阈值）告警，虽然 CPU 限制不会直接导致容器重启，但如果容器因 CPU 资源不足而无法正常工作，可能会导致崩溃，此处cpu不足的原因需要联系技术支持寻求服务帮助，可能原因如下： - 实际分配cpu资源限制过小，需调整大小。 - 应用程序可能存在异常，导致cpu使用量逐渐增加，最终超出限制，需优化程序逻辑。
2	健康检查失败	Liveness Probe: 如果配置了活跃探针（liveness probe），并且探针检测到容器不健康，Kubernetes 会重启该容器，此处容器不健康需要联系技术支持寻求服务帮助确认业务服务是否正常，容器不健康的根因。
3	节点问题	此处若出现节点故障，节点上所有服务均会异常，该场景由TCS层面统一处理，业务侧无需关注。

处理步骤

1. 查询pod最近一次运行结束时的状态信息：
    - 基础平台业务：kubectl describe pod -n tce ocloud-osp-passwd-svr-xxx | grep -A 5 'Last State:'
    - 审计日志业务：*kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Last State:'*
    - CAM业务：kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 5 'Last State:'
    - 标签业务：kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Last State:'
    - 云API管理业务：kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Last State:'

    ```
    Last State:     Terminated
      Reason:       Error
      Exit Code:    137
      Started:      Sun, 01 Dec 2024 16:55:11 +0800
      Finished:     Sun, 01 Dec 2024 16:57:21 +0800
    Ready:          True
    ```

    > 说明：
    >
    > 此处主要查看reason和exit code部分：
    >
    > - exitCode：容器的退出代码。
    > - reason：终止的原因，Error（发生错误）、OOMKilled（因内存不足被杀死）等。

2. 查看pod事件状态：

- 基础平台业务：kubectl describe pod -n tce ocloud-osp-passwd-svr-xxx | grep -A 10 'Events:'
- 审计日志业务：kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 10 'Events:'
- CAM业务：kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 10 'Events:'
- 标签业务：kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 10 'Events:'
- 云API管理业务：kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 10 'Events:'

3. 收集故障时间点日志信息，联系技术支持寻求服务帮助，日志路径： /data/log

Pod cpu使用率(占limit)超过阈值处理预案

告警说明

Pod cpu使用率(占limit)超过阈值：

- 基础平台业务相关容器（cloud-tcenter）由于某些异常原因，导致cpu使用率较高，超过了limit设置的95%。
- 审计日志业务相关容器（t|ocloud-cloudaudit）由于某些异常原因，导致cpu使用率较高，超过了limit设置的95%。
- CAM业务相关容器（t|ocloud-tcenter-cam/tcloud-tcenter-support-cam）由于某些异常原因，导致cpu使用率较高，超过了limit设置的95%。
- 标签业务相关容器（tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag）由于某些异常原因，导致cpu使用率较高，超过了limit设置的95%。
- 云API管理业务相关容器（cloud-tcenter-yunapi3）由于某些异常原因，导致cpu使用率较高，超过了limit设置的95%。

告警属性

所属模块	名称	事件级别	手工清除/自动清除
tcs-pod指标监控默认告警策略	Pod cpu使用率(占limit)超过阈值	2级告警	自动清除

对系统的影响

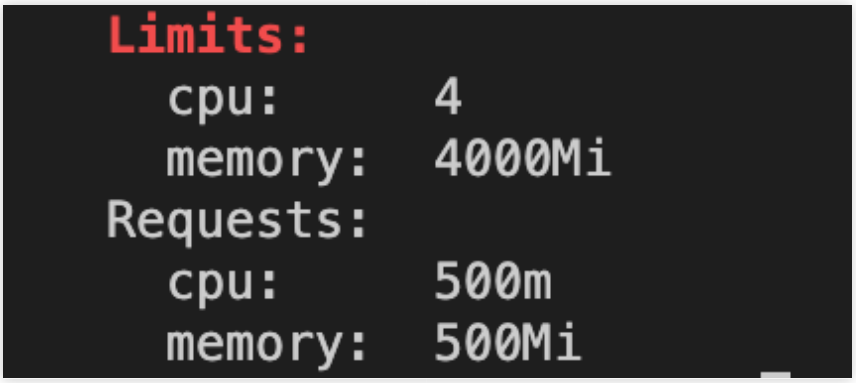
无影响。

可能原因

- 实际分配cpu资源限制过小，需调整大小。
- 应用程序可能存在异常，导致cpu使用量逐渐增加，最终超出限制，需优化程序逻辑。

处理步骤

1. 查询pod的资源限制信息：
    - 基础平台业务***：kubectl describe pod -n tce cloud-tcenter-xxx | grep -A 5 'Limits:'***
    - 审计日志业务：*kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Limits:'*
    - CAM业务：*kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx | grep -A 5 'Limits:'*
    - 标签业务：*kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Limits:'*
    - 云API管理业务：*kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Limits:'*



2. 查看pod历史监控信息：
   运营端-监控系统-云产品指标-TCS/POD内查询告警触发的pod在告警时间点附近或者更久的cpu使用率监控，确认是否为正常业务行为。

时间范围	故障时间点当天
监控指标	pod_cpu_limit_usage
Cluster	global
Namespace	tce
Pod	◦ 基础平台业务: cloud-tcenter-xxx ◦ 审计日志业务：Pod: t|ocloud-cloudaudit-xxx ◦ CAM业务：Pod: t|ocloud-tcenter-cam/tcloud-tcenter-support-cam-xxx

	○ 标签业务：Pod: tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx ○ 云API管理业务：Pod: cloud-tcenter-yunapi3-xxx

3. 收集相关信息，联系技术支持寻求服务帮助。

Pod 实际内存使用率(占limit)超过阈值处理预案

告警说明

Pod 实际内存使用率(占limit)超过阈值：

- 基础平台业务相关容器（cloud-tcenter）由于某些异常原因，导致内存使用率较高，超过了limit设置的95%。
- 审计日志业务相关容器（t|ocloud-cloudaudit）由于某些异常原因，导致内存使用率较高，超过了limit设置的95%。
- CAM业务相关容器（t/ocloud-tcenter-cam/tcloud-tcenter-support-cam）由于某些异常原因，导致内存使用率较高，超过了limit设置的95%。
- 标签业务相关容器（tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag）由于某些异常原因，导致内存使用率较高，超过了limit设置的95%。
- 云API管理业务相关容器（cloud-tcenter-yunapi3）由于某些异常原因，导致内存使用率较高，超过了limit设置的95%。

告警属性

所属模块	事件名称	事件级别	手工清除/自动清除
tcs-pod指标监控默认告警策略	Pod 实际内存使用率(占limit)超过阈值	2级告警	自动清除

对系统的影响

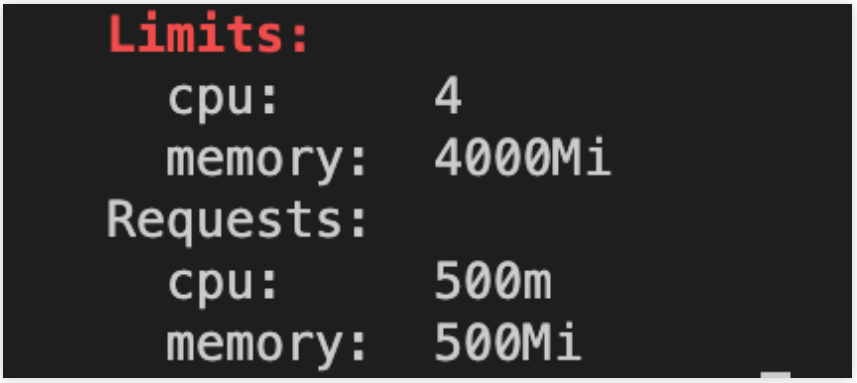
无影响。

可能原因

- 实际分配内存资源限制过小，需调整大小。
- 应用程序可能存在内存泄漏，导致内存使用量逐渐增加，最终超出限制，需优化程序内存回收机制。

处理步骤

1. 查询pod的资源限制信息：
    - 基础平台业务：*kubectl describe pod -n tce cloud-tcenter-xxx | grep -A 5 'Limits:'*
    - 审计日志业务：*kubectl describe pod -n tce t|ocloud-cloudaudit-xxx | grep -A 5 'Limits:'*
    - CAM业务：*kubectl describe pod -n tce t|ocloud-tcenter-cam/tcloud-tcenter-support-cam | grep -A 5 'Limits:'*
    - 标签业务：*kubectl describe pod -n tce tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx | grep -A 5 'Limits:'*
    - 云API管理业务：*kubectl describe pod -n tce cloud-tcenter-yunapi3-xxx | grep -A 5 'Limits:'*



2. 查看pod历史监控信息：
   运营端-监控系统-云产品指标-TCS/POD内查询告警触发的pod在告警时间点附近或者更久的内存使用率监控，确认是否为正常业务行为。

时间范围	故障时间点当天
监控指标	pod_memory_nocache_limit_usage
Cluster	global
Namespace	tce

Pod	○ 基础平台业务： cloud-tcenter-xxx ○ 审计日志业务： t|ocloud-cloudaudit-xxx ○ CAM业务： t|ocloud-tcenter-cam/tcloud-tcenter-support-cam ○ 标签业务：tcloud-tcenter-platform-tag/tcloud-tcenter-platform-wtag-xxx ○ 云API管理业务： cloud-tcenter-yunapi3-xxx

3. 收集相关信息，联系技术支持寻求服务帮助。

应急预案
用户与权限：绑定产品权限后仍然无权限访问产品页面

故障现象场景描述

绑定产品权限后仍然无权限访问产品页面。

故障定位分析

1. 确认权限策略是否正常：运营端查看对应的策略语法是否包含页面访问的动作和接口。

2. 确认数据库中策略信息是否正常：
   登录cam数据库，在cExtension表中根据uin查该用户绑定的策略，查看 actionName 中，有无用户所需的接口；若无，则在 cServicePerm 表中，根据serviceType查找接口对应服务的数据，查看apiEnName中，有无用户所需的接口。

   ```
   dbsql_info dbsql-tcenter-cam-cauth/dbsql-tcenter-ocloud-cam-cauth
   select * from cExtension where uin='xxx' and actionName like "%xxx%"\G
   select * from cExtension where uin=xxx and strategyId in (xxx);
   select * from `cServicePerm` where serviceType='xxx'\G
   select * from cServicePerm where serviceType in (xxx);
   select * from cServicePerm where permId in (xxx) and apiEnName like '%xxx%'\G
   select distinct permId from cStrategyPerm where strategyId=xxx;
   ```

3. 查看策略信息是否正常：在 cStrategy 表中查看所绑定的策略名详情信息。

   ```
   set names latin1
   select * from cStrategy where strategyName = 'xxx访问管理权限'\G
   select * from cStrategy where  strategyInfo like '%xxx%'\G
   select * from cStrategy where strategyId="xxx";
   ```

4. 检查策略绑定任务记录信息：
   通过 jobdetail 表，查看策略绑定的记录是否有异常。

   ```
   select * from cExtendJobDetail where uin=xxx;
   select * from cExtendJobDetail where uin=xxx order by updateTime desc limit 10;
   ```

   同时确认通过用户组添加的权限策略记录是否有异常。

   ```
   select * from cReceiverInfo where uin=xxx; // uid
   ```

```
select * from cGroupMemberInfo where uid=xxx; // groupId
select * from cExtendJobDetail where groupId in (xxx) order by updateTime desc limit 20; //
select * from cRelatedStrategy where groupId=xxx;
```

5. 进一步通过请求复现问题进行分析：

   i. 使用出问题的账号操作出问题的动作，然后使用没有权限的requestId在cam-auth组件中查询日志。
   ii. 待查询到日志，修改日志的内容，填加 "debug": 1。
   iii. curl 127.0.0.1:9502 -d'' ，使用修改后的参数发起请求。


```
{
  "version": 1,
  "componentName": "API",
  "eventId": "4627b684-bcfb-4206-a72a-a46a9ac173a1",
  "timestamp": 1744355915,
  "interface": {
    "para": {
      "version": "2.0",
      "method": "POST",
      "mode": 0,
      "debug": 1,
      "url": "cbs.yunapi3.3100arm-baseline.fsphere.cn/?
Action=DescribeDepotTransferTaskOverview&DepotIds.0=16777291&Language=zh-
CN&Nonce=1744355915&OIDCIdToken=&OIDCRefreshToken=&Region=chongqing&Request
Client=CAPI_GO_SDK&RequestSource=&SecretId=AKIDB0IgYdiELni7MCTKBrV0nqcNqLeRIt
yd&SignatureMethod=HmacSHA256&Timestamp=1744355915&Token=81d4a1873a606507
d8b5db2a915acdce447cb8d710001&Version=2017-03-12",
      "header": "authorization:q-sign-algorithm=api_sha256&q-
ak=AKIDB0IgYdiELni7MCTKBrV0nqcNqLeRItyd&q-sign-time=1744355615;1744356215&q-
key-time=1744355615;1744356215&q-header-list=&q-url-param-
list=Action;DepotIds.0;Language;Nonce;OIDCIdToken;OIDCRefreshToken;Region;RequestClie
nt;RequestSource;SecretId;SignatureMethod;Timestamp;Token;Version&q-
signature=c045bda9b59c3e362d94853433304322ddc9ddaa8a9119ed9c355a9692859c6
```


故障应急处置步骤

1. 若预设数据权限策略缺失，需手动更新preset组件。

```
kaleido_preset update /data/tce_dc/software/lastest/preset/preset-xxx
```

2. 若数据存在脏数据，需进行清理。

i. 备份原数据：

```
tad job dbBackup --servicebindings=dbsql-tcenter-cam-cauth
tad job dbBackup --servicebindings=dbsql-tcenter-ocloud-cam-cauth
```

ii. 删除cServicePerm中脏数据。

iii. 重刷preset组件：

```
kaleido_preset update preset-
```

iv. 检查cStrategyPerm和cExtension中是否引用正确。

附录A 常用操作
常用操作

重启pod

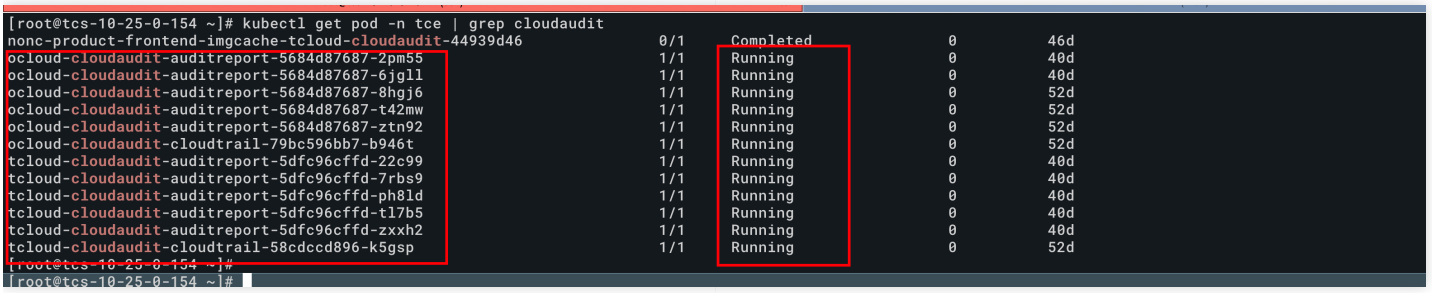
操作描述

适用于页面有报错、pod状态不正常时为快速恢复业务进行尝试性操作 。

处理步骤

查询审计服务pod，删除需要删除的pod。

```
kubectl get pod -n tce | grep cloudaudit
kubectl delete pod -n tce tcloud-cloudaudit-xxxxx
```



查看服务日志

操作描述

排查问题时，查看服务日志。

处理步骤

查询pod，进入容器，查看日志。

```
kubectl get pod -n tce | grep cloudaudit

kubectl exec -it -n tce tcloud-cloudaudit-auditreport-xxxxx bash
cd /data/log/python/
vim xxxxxxx-log.txt

kubectl exec -it -n tce tcloud-cloudaudit-cloudtrail-xxxxx bash
```

```
cd /data/log/cloudtrail/detail
vim daily_xxxxxx.log
```

获取服务日志

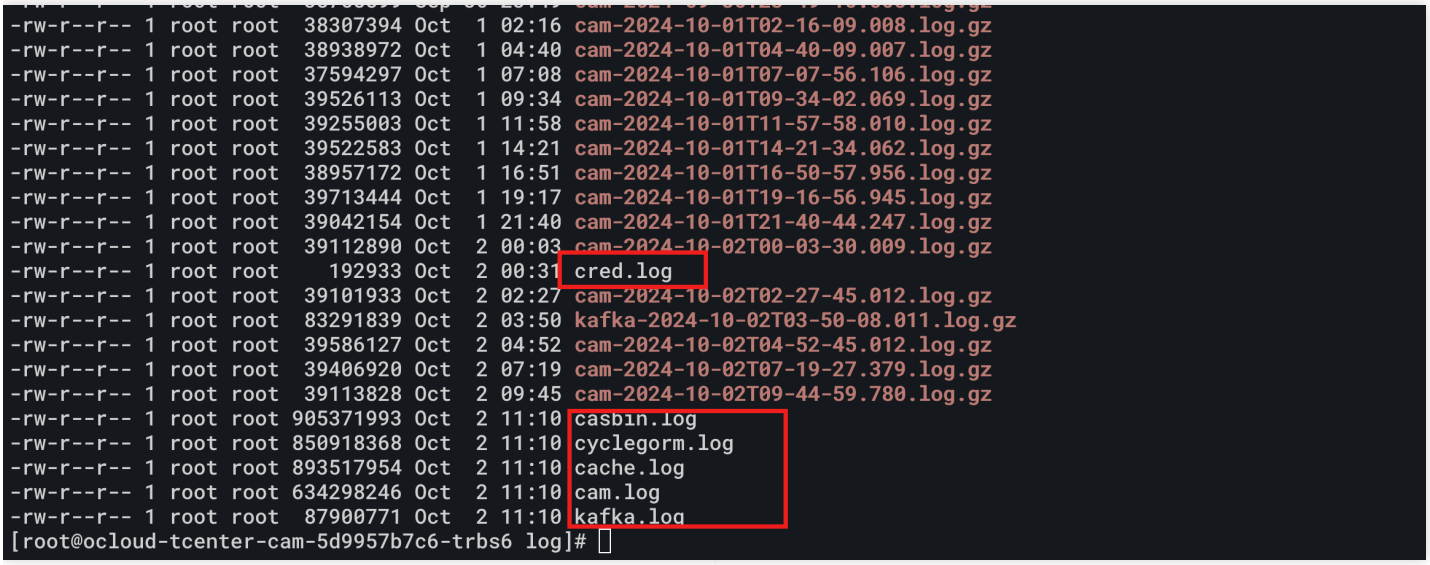
操作描述

cam 服务异常时，需要通过查看日志进行问题处理。

处理步骤

容器化部署，进入tcloud-tcenter-cam/ocloud-tcenter-cam，进入**/data/log**目录查看 cam.log 等日志。

kubectl get pod -n tce | grep cloud-tcenter-cam
kubectl exec -it -n tce ocloud-tcenter-cam-xxxxxxx bash
cd /data/log/
tail -f cam.log 等日志



查看pod渲染状态

操作描述

pod日常状态巡检。

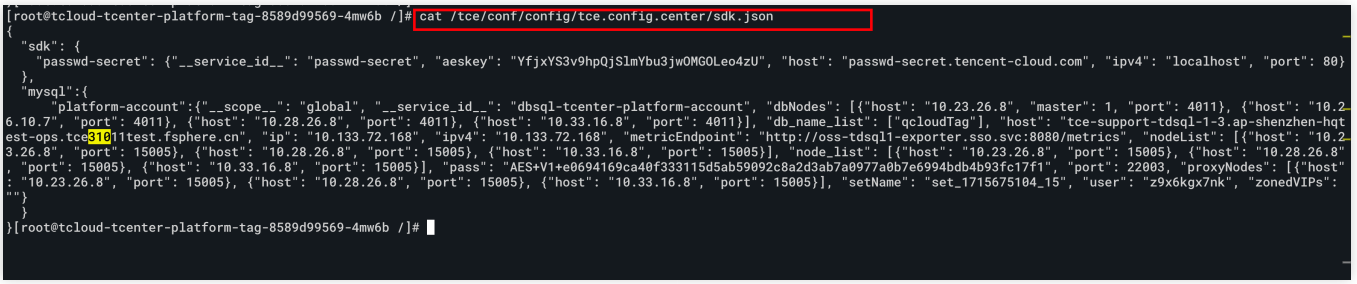
处理步骤

1. 进入pod内查看渲染状态：

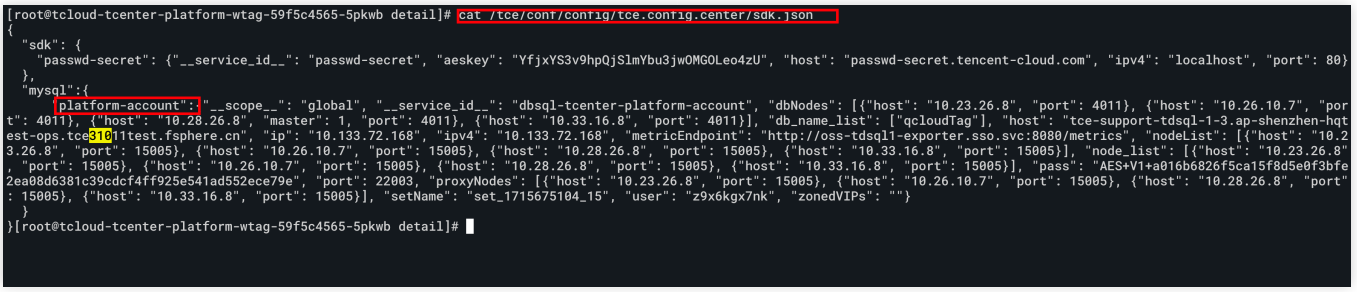
kubectl get pod -ntce | grep -E 'tcenter-platform-tag|tcenter-platform-wtag'
#进入pod内查看渲染状态
kubectl exec -it -n tce tcloud-tcenter-platform-tag-xxxxx bash
cat /tce/conf/config/tce.config.center/sdk.json

kubectl exec -it -n tce tcloud-tcenter-platform-tag-wtag-xxxxx bash
#进入pod内查看渲染状态
cat /tce/conf/config/tce.config.center/sdk.json
cat /data/release/swoole_tag/application/config/idc/hosts_config.php

2. tcloud-tcenter-platform-tag 服务数据库渲染：



3. *tcloud-tcenter-platform-tag-wtag* 数据库渲染是否正常：



4. cmq渲染是否正常：

```
[root@tcloud-tcenter-platform-wtag-59f5c4565-5pkwb detail]# cat /data/release/swoole_tag_write/application/config/idc/hosts_config.php
<?php

class HostsConfig{
    public static function getResourceUrl(){
        $config = array();
        $config['gz']['cvm']['cvm'] = 'http:///interfaces/interface.php';
        return $config;
    }
    const ACCOUNT_URL = "http://tcloud-tcenter-platform-account.tcenter:6060/account";
    const WHITE_LIST_URL = "http://tcloud-tcenter-platform-whitelist.tcenter:6060/whitelist";
    const CAM_TAG_URL = "http://tcloud-tcenter-cam-grant-write.tcenter/camTag";
    const CMQ_DOMAIN = "";
    const CMQ_URL = "";
    const CMQ_SECRET_ID = "";
    const CMQ_SECRET_KEY = "";
    const CMQ_TOPIC_NAME = "tag-resource-topic";
    const CMQ_TOPIC_NAME1 = "tag-resource-topic";
    const CMQ_TOPIC_NAME_PROJECT = "tag-project-resource-topic";
    const MQ_API_HOST = "http://tce-support-cmq-1-1.ap-shenzhen-hqtest-ops.tce3101test.fsphere.cn:5672/api/v1";
    const CMSI_URL = "";

    const CAM_AUTH_URL = "http://tcloud-tcenter-cam-auth.tcenter:9502";
}


[root@tcloud-tcenter-platform-wtag-59f5c4565-5pkwb detail]#
```

获取操作日志

操作描述

查看pod，进入pod查看日志。

处理步骤

```
kubectl get pod -n tce | grep tcenter-yunapi3
kubectl exec -it -n tce t/ocloud-tcenter-yunapi3-traefik-xxxx bash
cd /usr/local/services/traefik/log
#通过页面reqid过滤日志
grep -ir "reqid" traefik.log
 grep -ir "reqid" *
```

技术指标
用户与权限

- 用户创建成功率99.9%。
- 用户组查询成功率99.9%。
- 策略创建成功率99.9%。
- 策略绑定成功率99%。

性能指标
用户与权限

性能指标	• 用户列表查询 pct99: 300ms • 新建 pct99: 1s • 用户组列表 pct99: 300ms • 新建用户组 pct99: 1s • 策略列表 pct99:200ms • 新建自定义策略 pct99: 1.5s • 策略绑定用户 pct99: 800ms • cpu使用率 5% • 内存占用率 10%
产品限制参数	• 主账号最多可创建1000个自定义策略 • 单个用户最多可关联5000个策略

产品白皮书
产品概述

基础平台和运营平台（基础版）是为企业客户构建的一站式运营管理平台。平台通过提供统一的账号权限管理、资源管理、产品管理等核心能力，实现对专有云环境的集中化、自动化管理，帮助客户简化人员、资源、产品的管理流程，从而提升云上业务的运营效率，让企业能够更加专注于核心业务创新与发展。

产品优势

精细运营

- 为客户提供云产品的用量、费用数据，并提供配额管控能力。
- 支持对资源做分组管理，解决单个租户内的资源分组和授权管理的复杂性问题。
- 可匹配企业内的多级组织，实现企业内多租户的统一运营管理。

安全运营

- 可通过登录保护、操作保护、审批管理功能实现对高危操作的管控。
- 云审计可实现对所有操作的审计。
- 提供细粒度的权限管理，以满足企业内部的权限控制要求。

开放

所有产品的 API 全量开放，支持客户基于 API 做二次开发。

应用场景

1、统一运营

场景描述：针对中小型企业，不分部门，管理相对集中。

方案描述：企业可新建两个租户，一个作为测试环境，一个作为生产环境。在测试租户或生产租户下，可按需新建项目，不同资源和账号加入到对应项目里，项目里的资源只能被项目内成员管理。该方案可帮助客户实现租户内资源的分组管理。

2、安全运营

场景描述：金融、保险行业需满足等保要求，实现操作可追溯、权限精细化。

方案描述：云审计可记录用户的所有操作记录；权限管理可帮助客户实现操作级别和资源级别的权限控制；登录保护、操作保护、审批管理帮助客户实现对高危操作的管控。

产品架构

基础平台和运营平台（基础版）主要提供用户管理、资源管理、产品管理的能力，具体功能架构如下。

功能特性

- 为客户提供安全合规的用云管云流程：提供账号、登录、操作保护、权限管理、审批管理、云审计、项目管理、消息通知能力，保障客户安全合规用云管云。
- 提供精细化运营能力：通过计费计量、配额管理、容量平台，实现对资源的精细化管理。
- 开放：云 API 全面开放，支持客户二次开发。

  账号：平台支持多租户，租户间的资源隔离，单租户内分主账号子账号，子账号的权限需由管理员来分配。

  登录：平台提供登录保护功能，当用户登录云平台时，需要验证身份（如手机验证码、token 等），避免了即使他人盗取您的密码，也无法登录您的账号的情况，能够最大限度地保证账号安全。

  操作保护：操作保护可有效保障账号的资源安全。云用户执行敏感操作前，需要先完成身份验证（如手机验证码、token 等），身份验证通过后方可进行相关操作。

  权限管理：专有云提供接口粒度和资源粒度的权限管理，帮助客户安全、便捷地管理对云服务和资源的访问。

  审批管理：支持自定义编排审批流。高危操作需通过管理员审批之后才能被执行。

  云审计：用户通过云控制台、API、SDK 对云上产品和服务的访问和使用行为均可被审计，以满足安全合规的审计要求。

  项目管理：解决单个租户账号内的资源分组、授权管理以及按项目分账的复杂性问题。

  计费计量：支持统计各租户的资源实例用量信息，并提供统一的聚合统计、展示、查询、筛选等功能。

  配额管理：可通过配额管理，统一为租户分配各云产品的资源配额。

  资源容量平台：为客户提供容量可视化、容量告警、容量预测的功能，便于客户直观感知整个资源池下的资源使用情况，及时根据业务情况做出决策（如扩容等）

  报表平台：为客户提供统一、自动化的报表服务，将云平台分散的资源数据转化为标准化报表，助力客户定期、清晰地掌握资源基础情况。

  产品上下架：可对云产品设置上架、灰度上架和下架操作，上架时，所有租户可在云平台访问到对应产品；灰度上架时，产品只对名单内的租户可见；下架时，该产品在控制台对所有租户都不可见。

  产品目录：可为指定的租户设置租户可见的产品目录。

  API管理：平台所有产品的 API 全量开放，并提供 API 文档，支持客户基于 API 做二次开发。

  消息：可将运营消息或告警类消息及时发送给用户。

使用建议

基础平台

软件使用建议

指标名	默认值	取值范围
租户端会话超时时间	30分钟	30～1440分钟
运营端会话超时时间	30分钟	30～1440分钟
租户端账号临时锁定时间	60分钟	10～1440分钟
运营端账号临时锁定时间	60分钟	10～1440分钟
租户端登录时密码输入错误次数	10次	3～10次
运营端登录时密码输入错误次数	10次	3～10次
租户端多端登录开关	关闭	关闭、开启
运营端多端登录开关	关闭	关闭、开启
运营端登录图形验证码	关闭	关闭、开启
租户端登录图形验证码	关闭	关闭、开启
租户端水印	关闭	关闭、开启
运营端水印	关闭	关闭、开启

用户与权限

软件使用建议

指标名	默认值	取值范围

指标名	默认值	取值范围
单租户下的账号数量	10000个	不支持调整
单租户下的用户组数	1000个	不支持调整
子账号可加入的用户组数	300个	不支持调整
一个用户组中的用户数	1000个	不支持调整
单租户下的自定义策略数	1000个	不支持调整
一个用户、用户组或角色可关联的策略数	5000个	不支持调整
一个策略语法最大字符数	4096个字符	不支持调整
SSO协议	-	支持OAuth 2.0、SAML、CAS、LDAP协议
密码组成	小写字母、数字	大写字母、小写字母、数字、特殊符号，至少包含两项
最短密码长度	10个字符	10～32个字符
密码有效期	90天	0（不限制）～365天
重复限制	1次（默认不能与前一次密码相同）	1～24次

云API管理

软件使用建议

指标名	默认值	取值范围
云 API SDK 语言	-	云 API 提供多语言 SDK，支持 Java、Python、PHP、Golang和Node.js 开发语言。
API 超时时间	5000ms	10～30000ms
API 单租户限频值	20次/s	不支持调整。
单接口参数个数	-	0～20个

指标名	默认值	取值范围
单复杂对象参数个数	-	0～20个
请求包大小	1024KB	1024～10240KB

- 平台固有产品 API 只提供查询能力，不可编辑和删除。
- 用户自定义产品及接口，可以新增、编辑和删除。

计量管理

软件使用建议

已接入计量计费的产品范围如下，如项目上部署了计费管理，可实现对云产品的按量计费。

分类	产品
计算	- 云服务器（CVM） - 裸金属服务器（BMS）
网络	- 负载均衡（CLB） - 弹性公网IP（EIP） - NAT网关（NATGW） - NAT64网关 - VPN网关 - 专线接入（DC） - 对等连接（PC） - IDC带宽
存储	- 云硬盘（CBS） - 云硬盘快照Snapshot - 对象存储（COS） - 对象存储（CSP） - 文件存储（CFS） - 并行文件存储（TurboFS） - 备份中心（BRC） - 日志服务（CLS）
数据库	- TDSQL MYSQL版 - TDSQL PostgreSQL版 - 云数据库 Redis®

分类	产品
中间件	- 消息队列 Pulsar 版 - 消息队列 CKafka 版 - 消息队列 RocketMQ 版
安全	- 主机安全（CWP） - Web应用防火墙（WAF） - 安全运营中心（SOC） - 云堡垒机（CBH） - 数据安全审计SaaS版（DSAuditSaaS） - 密钥管理服务（KMS） - 凭据管理系统（SSM） - 云加密机（CloudHSM） - 云防火墙（CFW）

计量计费的产品配额范围如下：

云产品	配额项
云服务器（CVM）	实例数
	CPU核数
	内存大小
裸金属服务器（BMS）	裸金属服务器数
云硬盘（CBS）	实例数
	存储空间
云硬盘快照Snapshot	快照数
文件存储（CFS）	实例数
并行文件存储（TurboFS）	实例数
	存储空间
备份中心（BRC）	备份点数量
TDSQL MYSQL版	实例数
	CPU核数
	存储空间
TDSQL PostgreSQL版	实例数量

云产品	配额项
	CPU核数
	存储空间
云数据库 Redis®	实例数
	内存

云审计

软件使用建议

- 支持记录所有云API的操作。
- 支持查询最长一年内的操作记录。

审批管理

软件使用建议

已接入审批管理的产品范围如下：

分端	产品
租户端	云服务器（CVM）
	负载均衡（CLB）
	云硬盘（CBS）
	私有网络（VPC）
	容器服务（TKE）
	裸金属服务器（BMS）
	云数据库(MySQL MariaDB)（MariaDB）

分端	产品
	TDSQL MYSQL版
	消息队列 CKafka 版（CKafka）
	消息队列 Pulsar 版（TDMQ Pulsar）
	专线接入（DC）
	文件存储（CFS）
	资源管理
运营端	基础平台（Tcenter）
	裸金属服务器（BMS）
	基础设施（DCOS）
	负载均衡（CLB）
	私有网络（VPC）
	专线接入（DC）
	云服务器（CVM）
	作业工具
	主机安全（CWP）
	云硬盘（CBS）
	文件存储（CFS）
	云数据库 Redis®
	产品运维中心
	对象存储（COS）
	网络管理（NMS）
	密码库
	Agent管理
	Web应用防火墙（WAF）

产品简介
CAM概述

访问控制（Cloud Access Management，CAM）是云平台提供的Web服务，主要用于帮助客户安全管理云平台账户下资源的访问权限。用户可以通过CAM创建、管理和销毁用户(组)，并使用身份管理和策略管理控制其他用户使用云平台资源的权限。

产品功能

CAM提供以下功能支持：

根账号资源的授权访问

可以将根账号的资源授权给其他人员，包括子账号和其他根账号，而不需要分享根账号相关的身份凭证。

精细化的权限管理

可以针对不同的资源授权给不同的人员不同的访问权限。例如可以允许某些子账号拥有CVM某台虚拟机的操作权限，而另一些账号或者根账号可以拥有某个地域的CVM操作权限等。这里的资源、访问权限、用户都可以批量打包。

最终一致性

CAM目前支持云平台的多个地域，通过复制策略数据实现跨地域的数据同步，虽然CAM策略的修改会及时提交，不过跨地域的策略同步会导致策略生效的延迟；同时CAM使用缓存来提高性能（目前是一分钟缓存），更新需要在缓存过期后生效。

应用场景

企业子账号权限管理

企业内不同岗位的员工需要拥有该企业云资源的最小化访问权限。
场景：某个企业拥有很多云资源，包括CVM 、VPC实例、CDN实例、COS存储桶和对象等。该企业拥有很多员工，包括开发人员、测试人员、运维人员等。部分开发人员需要拥有其所在项目相关的开发机云资源的读写权限，测试人员需要拥有其所在项目的测试机云资源的读写权限，运维人员负责机器的购买和日常运营。当企业员工职责或参与项目发生变更，将终止对应的权限。

使用限制

限制项	限制值
一个主账号中的用户组数	300
一个主账号中的子账号数	2000
一个子账号可加入的用户组数	300
一个用户组中的子账号数	300
一个主账号可创建的自定义策略数	1000
一个策略语法最大字符数	4096

> 注意：
>
> 1. 一个主账号可创建的自定义策略数包含COS自定义策略数。如果您遇到「超过自定义策略条数上限（上限为1500条）」提示且CAM自定义策略数未达到上限，可前往COS存储桶列表-控制台 ，单击存储桶名称进入权限管理处查看 ACL（Access Control List）数目是否超过上限。
> 2. 直接关联到一个用户、用户组的策略数包含COS自定义策略数。如果您遇到「关联策略失败」提示且CAM内关联策略数未达到上限，可前往COS存储桶列表-控制台 ，单击存储桶名称进入权限管理处查看 ACL（Access Control List）数目是否超过上限。

支持CAM的产品

简介

访问管理已经支持对多数云产品服务进行权限管理。本文主要介绍支持访问管理CAM的产品服务的相关信息。具体维度包括授权粒度、控制台、根据标签进行授权、参考文档等。 以下列表分别罗列了云平台各大产品类别下已支持CAM的服务。 对表中信息进行如下定义：

- 服务：支持CAM的云服务的名称，单击链接至对应产品服务文档，方便您快速获取相关信息。
- 授权粒度：当前服务提供的最小授权粒度。
  其中授权粒度按照粒度粗细分为服务级、操作级和资源级三个级别。
- 服务级：定义对服务的整体是否拥有访问权限，分为允许对服务拥有全部操作权限或者拒绝对服务拥有全部操作权限。
- 操作级：定义对服务的特定接口（API）是否拥有访问权限，例如：授权某账号对云服务器服务进行只读操作。
- 资源级：定义对特定资源是否有访问权限，这是最细的授权粒度，例如：授权某账号仅读写操作某台云服务器。
- 控制台：是否支持子账号通过控制台访问当前服务，“✓”表示支持，“-”表示暂不支持。
- 根据标签进行授权：当前服务是否支持通过标签进行权限管理，“✓”表示支持，“-”表示暂不支持。
- 参考文档：当前服务与CAM相关的文档链接，“-”表示暂无。

计算

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
云服务器	资源级	✓	✓	-	访问管理指南
容器服务	资源级	✓	-	-	访问管理指南
裸金属服务器	资源级	✓	-	-	-

存储

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
对象存储	资源级	✓	✓	-	访问管理指南
文件存储	资源级	✓	✓	-	访问管理指南

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
云硬盘	资源级	✓	✓	-	-
日志服务(cls)	资源级	✓	✓	-	-

网络

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
负载均衡	资源级	✓	✓	-	访问管理指南
私有网络	资源级	✓	✓	-	访问管理指南

数据库

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
TDSQL MYSQL版	资源级	✓	-	-	访问管理指南
云数据库 Redis®	资源级	✓	-	-	-
TDSQL PostgreSQL版	资源级	✓	-	-	-
文档数据库(MongoDB)	资源级	✓	-	-	-
云数据库(MySQL MariaDB)	资源级	✓	-	-	-
数据传输服务（DTS）	资源级	✓	-	-	-

管理与审计

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
访问管理	操作级	✓	-	-	访问管理指南
云审计	操作级	✓	-	-	-

监控与运维

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
云监控	操作级	✓	-	-	-

公共服务

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
VPC域名解析（VPCDNS）	服务级	✓	-	-	-

大数据

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
Elasticsearch Service （ES）	服务级	✓	-	-	-

安全

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
堡垒机(dabs)	服务级	✓	-	-	-
数据加密服务(cloudhsm)	服务级	✓	-	-	-
主机安全(CWP)	服务级	✓	-	-	-
密钥管理服务(KMS)	服务级	✓	-	-	-
凭据管理服务(SSM)	服务级	✓	-	-	-
Web应用防火墙(WAF)	服务级	✓	-	-	-
数据安全审计	服务级	✓	-	-	-
敏感数据处理	服务级	✓	-	-	-
云防火墙（CFW）	服务级	✓	-	-	-

中间件

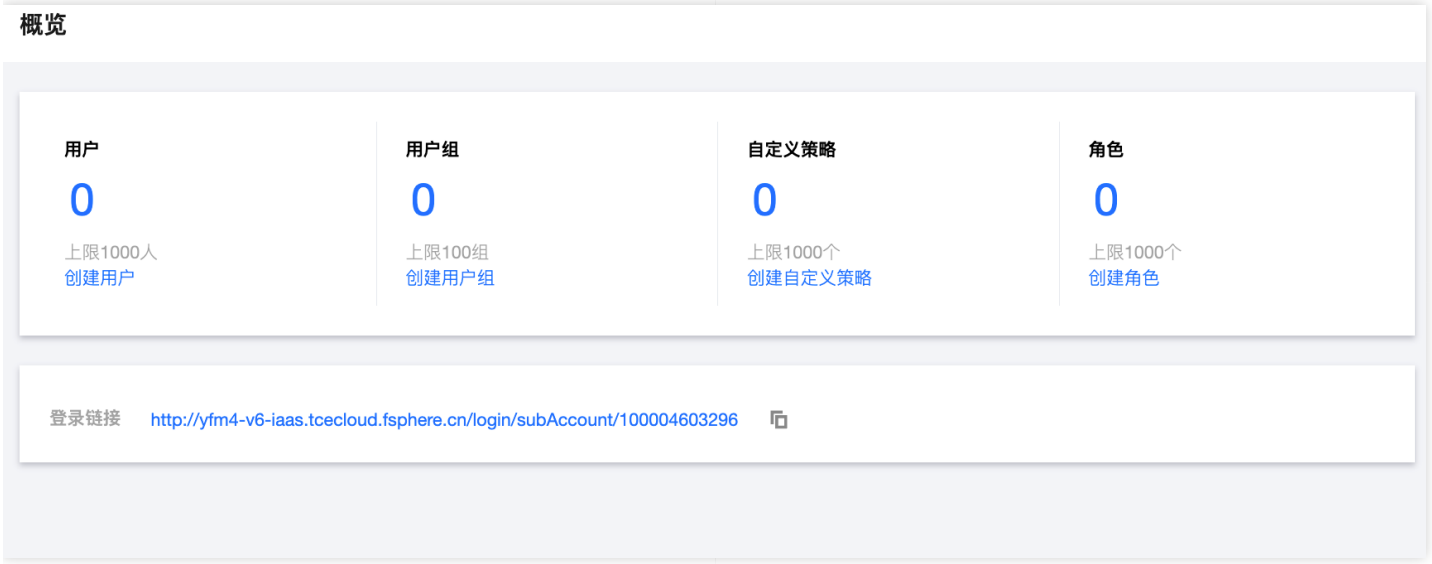
服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
微服务平台(TSF)	服务级	✓	-	-	-
消息队列 Pulsar 版	服务级	✓	-	-	-
消息队列 CKafka 版	服务级	✓	-	-	-

运营平台

服务	授权粒度	控制台	根据标签进行授权	服务角色	参考文档
云审计	服务级	✓	-	-	-
访问管理	服务级	✓	-	-	-
标签	服务级	✓	-	-	-

操作指南
概览

登录访问管理控制台，并在左侧导航栏中，选择【概览】，进入概览页面，显示用户、用户组、自定义策略、角色数量，和创建入口以及访问管理指引。



用户管理
用户类型

CAM 用户为您在云平台中创建的一个实体，每一个CAM用户仅同一个云账户关联。您注册的云账号身份为主账号，您可以通过用户管理来创建拥有不同权限的子账号协助您。子账号的类型分为子用和消息接收人。

账号类型	主账号	子账号	
		子用户	消息接收人
定义	拥有云所有资源，可以任意访问其任何资源。 不建议使用主账号对资源进行操作，应创建子账号并按照最小权限原则赋予策略，使用权限范围有限的子账号操作您的云资源。	由主账号创建，完全归属于创建该子用户的主账号。	仅拥有消息接收功能。
控制台访问	✔	✔	-
编程访问	默认已拥有全部策略	✔	-
策略授权	✔	✔	-
消息通知	✔	✔	✔

主账号

操作场景

本文档介绍主账号权限设置，消息接收，您可以通过以下步骤了解主账号权限以及如何修改消息接收方式。

前提条件

已注册云平台账号即主账号。

操作步骤

主账号无需授权

主账号默认拥有账号下云平台所有资源，无需授权，可以任意访问其任何资源。因此，不建议您使用主账号对资源进行操作，应创建子账号并按照最小权限原则赋予策略，使用权限范围有限的子账号操作您的云资源。

主账号消息接收

您注册云平台主账号时登记的安全手机、安全邮箱将同时作为初始消息接收方式。若您在账号中心 > 控制台内修改了安全手机或安全邮箱，您在访问管理（CAM）> 控制台用于云平台消息通知的联系手机或联系邮箱不会同步修改。

> 注意：为避免您因消息遗漏造成的损失，请您及时前往【访问管理】控制台确认用于消息订阅的联系手机或联系邮箱是否符合预期。

子账号
创建子用户

操作场景

本文档介绍如何创建及设定子用户的权限，子用户将在获得的权限范围内管理主账号下的资源。

操作指南

通过控制台创建

您可以通过登录云控制台，通过控制台创建子用户并设定权限。

1. 登录访问管理控制台，并在左侧导航栏中，选择【用户管理】>【用户】，进入用户页面。
2. 在用户页面，单击【新建用户】，弹出【选择新建用户类型】对话框，选择【子用户】，进入【新建子用户】页面。
3. 在填写用户信息页面，在“设置用户信息”下填写用户名（必填）、昵称、手机、邮箱信息。

   > 说明 - 单击【新增用户】可一次最多创建10个用户。

- 因子用户登录使用用户名，用户名一经确定将无法更改。

4. 在“访问类型”下设置子用户的访问方式。
   - 编程访问：启用SecretId和SecretKey，子用户将通过云API、SDK和其他开发工具管理权限范围内的主账号资源。
   - 云平台管理控制台访问：启用密码，子用户将通过登录到云控制台方式管理权限范围内的主账号资源。

     > 说明：为了保证您的账号安全，建议您开启登录保护和操作保护。

5. 设置【控制台密码】、【需要重置密码】、【登录保护】、【操作保护】。
6. 单击【创建】，创建子用户。创建成功后，系统自动进入【设定权限】页面。
7. 在【设定权限】中，可以选择：
   - 添加到现有用户组：新建用户组，并将子用户添加进来，或者选择加入现有用户组。
   - 复制现有用户权限：复制现有用户，使得当前用户具有已选择用户的权限。
   - 从策略列表中授权：选择策略，给当前子用户添加对应的策略权限。
8. 单击【下一步：完成】，进入设置用户权限页面。

9. 在设置用户权限页面，根据您的实际需求，选择不同的方式为当前新建的子用户设定权限，关联策略后子用户将获得策略描述的权限。
    - 添加至现有用户组：把子用户添加到组是按工作职能来管理用户权限的最佳做法，您可以通过随组关联获得权限。将子用户添加到现有用户组或新建用户组，子用户可以随组关联到该组附加的策略。
    - 复制现有用户权限：通过复制现有用户的权限为子用户关联策略，单击【复制现有用户权限】，勾选需要复制的用户，子用户可以关联到被复制用户附加的策略。
    - 从策略列表中授权：单击【从策略列表中授权】，勾选需要关联的策略。
10. 单击【下一步：完成】。
11. 在完成页面，单击【完成】完成创建子用户操作，进入提示新建子用户成功页面。
12. 进入提示新建子用户成功页面，您可以通过以下方法获取子用户信息。

- 单击【下载安全凭证】通过 excel 文件将部分信息保存至本地。

通过API创建

您可以通过访问密钥调用 AddUser 接口添加子用户并设定权限。

子用户权限设置

操作场景

本文档介绍如何授权和解除子用户关联的策略，子用户将在获得的权限范围内管理主账号下的资源。

操作步骤

为子用户授权关联策略

直接关联

您可以直接为用户关联策略以获取策略包含的权限。

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入 用户管理页面。
2. 在用户管理页面，点击用户，进入用户详情页。
3. 单击【已经关联的策略】可以查看已经关联的策略。
4. 点击关联策略，在弹出的策略列表选择策略。
5. 单击【确定】完成直接为子用户授权关联策略操作。

随组关联

您可以将用户添加至用户组，用户将自动获取该用户组所关联策略的权限，通过此种方法获取的策略类型为随组关联。如需解除随组关联策略，需将用户移出相应用户组。

1. 登录访问管理控制台，选择【用户管理】>【用户组】，进入 用户管理页面。
2. 在用户组管理页面，点击用户组，进入用户组详情页。
3. 单击【已经关联的策略】可以查看已经关联的策略。
4. 点击关联策略，在弹出的策略列表选择策略。
5. 单击【确定】完成直接为用户组授权关联策略操作。

为子用户解除关联策略

直接解除子用户关联策略

您可以直接解除用户关联的策略以解除用户关联的权限。

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入用户组管理页面。
2. 在用户管理页面，点击用户，进入用户详情页。
3. 单击【已经关联的策略】可以查看已经关联的策略。
4. 点击解除策略。
5. 单击【确定】完成直接为子用户授权解除策略操作。

从组中移出用户

您可以从组中移出用户以解除用户关联的策略

1. 登录访问管理控制台，选择【用户管理】>【用户组】，进入用户组管理页面。
2. 在用户组管理页面，点击用户组，进入用户详情页。
3. 单击【已经关联的策略】可以查看已经关联的策略。
4. 点击解除策略。
5. 单击【确定】完成直接为子用户授权解除策略操作。

子用户API密钥管理

操作场景

本文档介绍如何授管理子用户的云API，子用户可通过云API密钥来调用云API。

说明：只可为子用户创建密钥，无法为主账号创建密钥，在实际项目中，不建议使用主账号。

操作步骤

为子用户新建密钥

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入 用户管理页面。
2. 在用户管理页面，点击用户，进入用户详情页，单击【API密钥管理】。
3. 点击“新建密钥”按钮，选择密钥的创建方式：系统生效或自定义创建。
4. 密钥列表里会增加一条新创建的密钥。

禁用子用户密钥

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入 用户管理页面。
2. 在用户管理页面，点击用户，进入用户详情页，单击【API密钥管理】。
3. 在密钥列表的操作列，点击禁用按钮。
4. 在二次确认弹框里点确认禁用按钮。
5. 禁用后，用户无法通过该密钥来调用云API.

启用子用户密钥

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入 用户管理页面。
2. 在用户管理页面，点击用户，进入用户详情页，单击【API密钥管理】。
3. 在密钥列表的操作列，点击启用按钮。
4. 启用后，用户可通过该密钥来调用云API.

删除子用户密钥

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入 用户管理页面。
2. 在用户管理页面，点击用户，进入用户详情页，单击【API密钥管理】。
3. 在密钥列表的操作列，点击删除按钮。注意，删除密钥之前，需先禁用密钥。
4. 删除后，用户无法通过该密钥来调用云API.

子用户安全凭证
子用户登录

操作场景

本文档介绍如何登录子用户和企业微信子用户，登录成功后子用户将在权限范围内管理主账号下的资源。

操作步骤

子用户登录

通过账号、密码登录

您可以通过以下步骤使用账号、密码的方式登录 自定义创建的子用户，登录成功后，可在设置的权限范围内管理主账号下的资源。

1. 进入 子用户登录 页面进行账号登录。

2. 在子用户登录页面，输入主账号 ID、子用户名、登录密码信息， > 主账号 ID 即子用户所属主账号 ID。账号 ID（例如：100001234567）是账号在云平台的唯一标识，请联系主账号在 账号信息处查看。

3. 单击【登录】，完成通过账号、密码方式登录子用户操作。

为子用户重置登录密码

操作场景

本文档介绍如何修改子用户密码，修改之后可以通过新的密码登录子用户管理主账号下资源。

操作步骤

1. 在【用户管理】>【用户】中选择需要修改密码的子用户，选择具体【用户名称】，进入用户详情页。
2. 在用户详情页中选择【安全设置】>【控制台密码】，单击【管理密码】。
3. 在弹出的【管理控制台访问】窗口中，设置当前用户密码。
    - 若您当前子用户需要通过登录控制台访问云，请将【控制台访问】设置为【启用】。
    - 若您需要为子用户设置新密码，您可以通过以下两种方式。
    - 若您在【控制台密码】中选择【自动生成的密码】，系统会自动生成控制台登录密码。您可以复制保存，如有需要可以单击【下载.csv】保存密码。
    - 若您在【访问密码】中选择【自定义密码】，输入您为该子用户设置的控制台登录密码。
    - 若您需要当前用户自行重置密码，可勾选【用户必须在下次登录时重置密码】，子用户在下次登录成功后将被要求重新设置控制台登录密码。

为子用户设置安全保护

操作场景

本文档介绍如何开启和关闭子用户的安全保护，子用户将根据设置判断是否进行安全验证。

操作步骤

为子用户开启操作保护

1. 登录访问管理控制台，并在左侧导航栏中，选择用户管理 - 用户 ，进入用户列表页面。
2. 在用户列表页面，找到需要设置操作保护的子用户。
3. 单击用户名称，进入用户详情页面。
4. 在用户详情页面，单击安全设置，进入安全设置页面。
5. 在安全设置页面，单击 账号保护 - 操作保护后的编辑按钮。如下图所示：

6.在弹出的操作保护的窗口中，勾选需要开启的保护类型，为当前子用户开启相应的操作保护。

7.单击确定，完成为子用户开启操作保护。

为子用户关闭操作保护

1.参考 开启操作保护 中的步骤1 - 步骤5，进入身份安全窗口。
2.在弹出的身份安全窗口中，将操作保护的类型设置为不开启。

3.单击确定，完成为子用户关闭操作保护。

为子用户开启登录保护

1. 登录访问管理控制台，并在左侧导航栏中，选择用户管理 - 用户 ，进入用户列表页面。
2. 在用户列表页面，找到需要设置登录保护的子用户。
3. 单击用户名称，进入用户详情页面。
4. 在用户详情页面，单击安全设置，进入安全设置页面。
5. 在安全设置页面，单击 账号保护 - 登录保护后的编辑按钮。如下图所示：

6.在弹出的登录保护的窗口中，勾选需要开启的保护类型，为当前子用户开启相应的登录保护。

7.单击确定，完成为子用户开启登录保护。

为子用户关闭登录保护

1.参考 开启操作保护 中的步骤1 - 步骤5，进入身份安全窗口。
2.在弹出的身份安全窗口中，将登录保护类型设置为不开启。

3.单击确定，完成为子用户关闭登录保护。

子用户订阅消息

操作场景

本文档介绍如何为子用户验证消息渠道以及设置订阅消息。如需子用户接收消息，需子用户验证通过消息渠道，为其订阅消息后该用户已验证通过的消息渠道即可接收相关的消息提醒。

操作指南

设置订阅消息

1. 进入消息中心，点击消息订阅
2. 选择消息类型
3. 点击添加接收人
4. 在弹出的“添加接收人”窗口，勾选需订阅的的接收人/接收组。
5. 单击【确定】，完成设置订阅消息操作。

删除子用户

操作场景

本文档介绍如何删除单个或者多个子用户，删除之后，子用户将不再拥有该主账号的管理权限。

前提条件

已登录访问管理控制台，选择【用户管理】>【用户】，进入用户管理页面。

操作步骤

删除单个子用户

1. 在用户管理页面，找到需要删除的子用户。
2. 单击右侧操作列的【删除】。
3. 在弹出的删除用户窗口，确认当前子用户下的 API 密钥已禁用且删除，详细请参考访问密钥。
4. 单击【确认删除】，完成删除单个子用户操作。

删除多个子用户

1. 在用户列表管理页面，左侧勾选需删除的子用户。
2. 单击左上方的【删除】。
3. 在弹出的删除用户窗口，确认已勾选子用户下的 API 密钥已禁用且删除，详细请参考 访问密钥。
4. 单击【确认删除】，完成删除多个子用户操作。

用户信息

操作场景

本文档介绍如何查看和修改子账号用户名、备注、手机等信息。

查看用户信息

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入用户管理页面，找到需要查看用户信息的子账号。
2. 单击【用户名称】，进入用户详情页面。
3. 可在页面上方查看当前子账号的用户信息（含用户名、备注、手机、邮箱、是否允许微信通知）。

修改用户信息

1. 登录访问管理控制台，选择【用户管理】>【用户】，进入用户管理页面，找到需要修改用户信息的子账号。
2. 单击【用户名称】，进入用户详情页，单击右上角【编辑】。
3. 在弹出的编辑信息窗口，修改相应的用户信息。
    - 修改当前用户的用户名，子用户因登录使用用户名，无法修改。
    - 联系手机：修改当前子账号绑定手机信息，该手机可以用于接收主账号消息通知及敏感操作前的身份验证。
    - 联系邮箱：修改当前子账号绑定邮箱信息，该邮箱可以用于接收主账号消息通知。
4. 单击【确定】，完成修改用户信息操作。您可以通过修改之后的用户名、手机、邮箱在 用户列表管理页面，搜索到您的子账号。

用户组
新建用户组

新建用户组

1. 登录访问管理控制台，选择【用户管理】>【用户组】，进入用户组管理页面。
2. 单击【新建用户组】，进入填写用户组信息页面。
3. 在填写用户组信息页面，填写用户组名和备注，其中用户组名为必填项。 > 说明： 在用户组列表中您可以搜索用户组名或备注，在众多用户组中快速准确定位到对应的用户组。
4. 单击【确定】，创建用户组成功。
5. 单击新建的用户组名称，进入设置用户组信息页面。
6. 在设置【已关联的策略】页面，单击【关联策略】，进入管理策略页面。
7. 勾选需要授权的策略（可多选），单击【确定】，即可关联策略成功。
8. 在【已关联的策略】，您可以查看用户组的相关设置，如有误可点击【解除】修改。

关联文档

如果您想了解如何通过用户组管理子用户进行分组授权，请参阅 用户管理、用户组权限设置。
如果您想了解如何创建子用户，请参阅 自定义创建子用户。

用户管理

操作场景

本文档介绍如何新为用户组添加或者删除单个、多个用户。

前提条件

已登录访问管理控制台，选择【用户管理】>【用户组】，进入用户组页面。

操作步骤

为用户组添加用户

单个用户组添加用户

1. 在用户组页面，找到要添加用户的用户组。
2. 单击右侧操作列的【添加用户】。
3. 在弹出的添加用户窗口，勾选要添加的用户。
4. 单击【确定】，完成为用户组添加用户操作。

多个用户组添加用户

1. 在用户组页面，左侧勾选需要添加的用户组。
2. 单击左上角【添加用户】。
3. 在弹出的添加用户窗口，勾选要添加的用户。
4. 单击【确定】，完成为用户组添加用户操作。

为用户组删除用户

为用户组删除单个用户

1. 在用户组页面，找到要删除用户的用户组。
2. 单击用户组名称，进入用户组详情页。
3. 在用户组详情页，单击【已添加的用户】，进入用户列表页面。
4. 在用户列表页面找到要删除的用户，单击右侧操作列的【移出该组】。
5. 单击【移出用户】，完成为用户组删除单个用户操作。

为用户组删除多个用户

1. 在用户组页面，找到要删除用户的用户组。
2. 单击用户组名称，进入用户组详情页。
3. 在用户组详情页，单击【已添加的用户】，进入用户列表页面。
4. 在用户列表页面，勾选需要删除的用户。
5. 单击【移出用户】>【确认移出】，完成为用户组删除多个用户操作。

用户组权限设置

操作场景

本文档介绍如何授权和解除用户组关联的策略，用户组下的子账号将在获得的权限范围内管理主账号下的资源。

前提条件

已登录访问管理控制台，选择【用户管理】>【用户组】，进入用户组管理页面。

操作步骤

为用户组添加策略

1. 在用户组管理页面，单击用户组名称，进入用户组详情页。
2. 在用户组详情页，单击【已关联的策略】，进入权限管理页面。
3. 在权限管理页面，单击【关联策略】。
4. 在弹出框勾选要添加的策略（可多选），单击【确定】，完成为用户组添加策略操作。

为用户组解除策略

1. 在用户组管理页面，单击用户组名称，进入用户组详情页。
2. 在用户组详情页，单击【已关联的策略】，进入权限管理页面。
3. 在列表中找到需要解除的策略，单击右侧的【解除】。
4. 确认无误后单击【确认解除】，完成为用户组解除策略操作。

删除用户组

操作场景

本文档介绍如何删除用户组，删除之后，用户组下的子账号将不再拥有通过用户组获得的权限。

操作步骤

删除单个用户组

1. 选择【用户管理】>【用户组】，进入用户组管理控制台页面。
2. 在用户组管理控制台页面，找到需删除的用户组。
3. 单击右侧操作列的【删除】，完成删除用户组的操作。

用户设置
密码规则

操作场景

本章节介绍如何设置子用户的密码有效期。

操作步骤

> 说明
>
> - 该步骤所设定的密码规则仅适用于使用密码登录的子用户。
> - 登录密码失效后子用户将无法通过其他登录方式进行登录，须重置登录密码。

1. 选择【用户管理】> 【用户设置】，进入安全设置页面。
2. 点击【密码规则设置】模块的编辑按钮，可以修改密码规则、密码长度、密码的有效期限以及密码重复次数。
    - 密码规则：至少包含大写字母、小写字母、数字、特殊规则中的两项密码规则
    - 最短密码长度：默认最短10个字符，最长可设置32个字符
    - 密码有效期：0天表示不限制，最长可设置 365 天
    - 重复限制：限制新密码与历史密码的重复，默认与前1次不重复。（最大24次密码不重复）
    - 密码黑名单：用户设置密码，禁止包含黑名单中设置的字符串，最多可设置10个
3. 单击【确定】按钮完成密码复杂度设置。从上一次修改密码开始计算，到达有效期需要重置密码。

登录策略

本章节介绍如何设置子用户的会话超时时间。

前提条件

必须使用主账号或者有管理员权限的子账号登录租户端。

操作步骤

1. 选择【用户管理】> 【用户设置】，进入用户设置页面。
2. 在【登录策略】区域，单击【会话超时时间】模块的编辑按钮，修改会话超时时间。
3. 修改完成后，单击【确定】。

策略管理
相关定义
权限

权限是描述在某些条件下允许或拒绝执行某些操作访问某些资源。

默认情况下，主账号是资源的拥有者，拥有其名下所有资源的访问权限；子账号没有任何资源的访问权限；资源创建者不自动拥有所创建资源的访问权限，需要资源拥有者进行授权。

策略是定义和描述一条或多条权限的语法规范。CAM 支持两种类型的策略，预设策略和自定义策略。预设策略是由云平台创建和管理的一些常见的权限集合，如超级管理员、云资源管理员等，这类策略只读不可写。自定义策略是由用户创建的更精细化的描述对资源管理的权限集合。预设策略不能具体描述某个资源，粒度较粗，而自定义策略可以灵活的满足用户的差异化权限管理需求。

通过给用户或者用户组绑定一个或多个策略完成授权。被授权的策略既可以是预设策略也可以是自定义策略。

策略

策略是用于定义和描述一条或多条权限的语法规范。云的策略类型分为预设策略和自定义策略。CAM 从不同角度切入，为您提供了多种方法来创建和管理策略。若您需要向CAM用户或组添加权限，您可以直接关联预设策略，或创建自定义策略后将自定义策略关联到CAM用户或组。每个策略允许包含多个权限，同时您可以将多个策略附加到一个CAM用户或组。

预设策略

预设策略由云创建和管理，是被用户高频使用的一些常见权限集合，如超级管理员、资源全读写权限等。操作对象范围广，操作粒度粗。预设策略为系统预设，不可被用户编辑。

自定义策略

由用户创建的更精细化的描述对资源管理的权限集合，允许作细粒度的权限划分，可以灵活的满足用户的差异化权限管理需求。例如，为某数据库管理员关联一条策略，使其有权管理云数据库实例，而无权管理云服务器实例。

授权指南
创建自定义策略

操作场景

本文档介绍如何通过不同的创建方式创建自定义策略，自定义策略允许作细粒度的权限划分，可以灵活满足用户的差异化权限管理需求。

前提条件

已登录访问管理控制台，进入策略管理页面。

操作步骤

按策略生成器创建

按策略生成器创建的策略，通过从策略向导中选择服务和操作，并定义资源，自动生成策略语法，简单灵活，优先推荐使用。

1. 在策略管理页面，单击左上角的【新建自定义策略】。

2. 在弹出的选择创建方式窗口中，单击【按策略生成器创建】，进入选择服务和操作页面。

3. 在选择服务和操作页面，补充以下信息。

   - 服务（必选）：选择需要添加的产品。
   - 操作（必选）：选择您要授权的操作。
   - 资源（必填）：填入您要授权的资源的资源六段式。授权粒度为操作级、服务级的云产品不支持填写具体资源六段式，填「*」即可，授权粒度为资源级的云产品资源描述方式请参阅 支持 CAM 的产品 中对应产品的「访问管理指南」文档。云产品支持的授权粒度请参阅 支持 CAM 的产品 中的「授权粒度」。
   - 条件（选填）：设置子账号上述授权的生效条件。详细可参阅 生效条件。

     > 说明：
     >
     > 一条策略中可以添加多条声明。

4. 单击【添加声明】>【下一步】，进入编辑策略页面。

5. 在策略编辑页面，补充策略名称、描述信息，确认策略内容，其中策略名称和策略内容由控制台自动生成。

> 说明：
>
> - 策略名称默认为 "policygen" ，后缀数字根据创建日期生成。您可进行自定义。
> - 策略内容与第 3 步的服务和操作对应，您可根据实际需求进行修改。

6. 单击【完成】，完成按策略生成器创建自定义策略的操作。

按标签授权

按标签授权的策略，将具有一类标签属性的资源快速授权给用户或用户组。

1. 在策略管理页面，单击左上角的【新建自定义策略】。
2. 在弹出的选择创建方式窗口中，单击【按标签授权】，进入按标签授权页面。
   - **赋予用户/用户组：**勾选需要授权的用户/用户组。（可选其一）
   - 对绑定标签：单击添加，在弹出的输入框中属于标签健和标签值。（必填项）
   - 的下列服务资源具有对应操作权限：选择服务和操作权限，默认为管理权限。
3. 在按标签授权页面选择以下信息，单击【下一步】，进入检查页面。
4. 在检查页面，确认策略名称、策略内容后单击【完成】，完成按标签授权创建自定义策略操作。其中默认的策略名称和策略内容由控制台自动生成，策略名称默认为 "policygen"，后缀数字根据创建日期生成。

授权管理

操作场景

本文档介绍如何通过策略关联用户/用户组/角色和如何通过用户/用户组/角色关联策略。关联成功后，用户/用户组/角色将通过策略获得对应的权限。

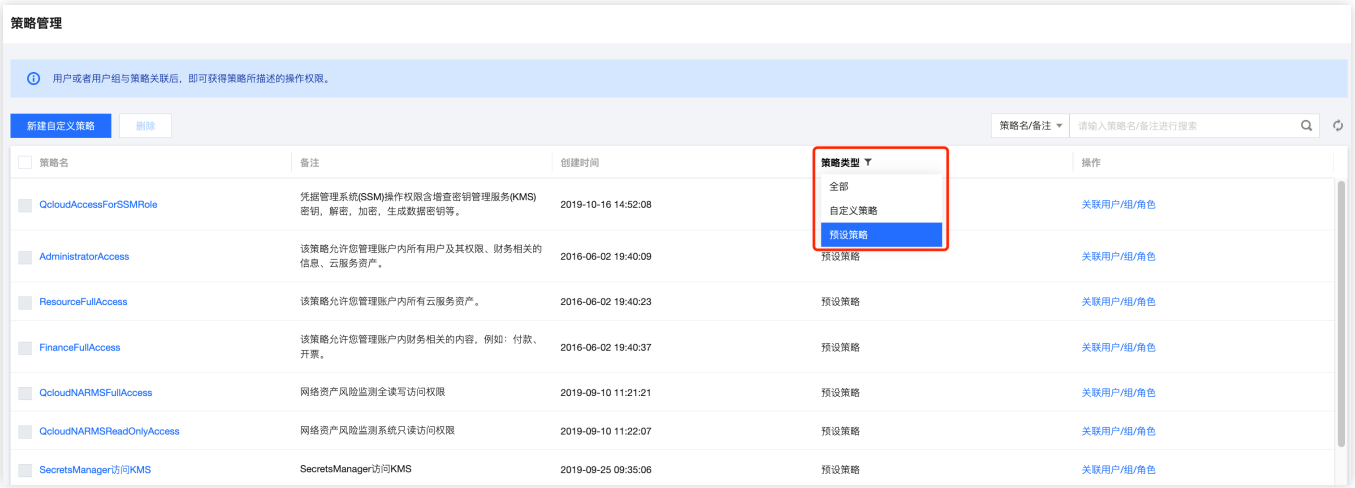
前提条件

已登录 访问管理控制台。

操作步骤

通过策略关联用户/用户组/角色

1. 在访问管理控制台，单击左侧【策略管理】，进入【策略管理】页面。
2. 在【策略管理】页面，根据业务需要在【策略类型】中选择预设策略或自定义策略。



3. 找到需要授权的预设策略，单击右侧【操作】列的【关联用户/组/角色】。
4. 在弹出的【关联用户/组/角色】对话框，根据业务需要在【类型】中选择用户、用户组或角色。

5. 勾选要关联的用户、用户组或角色。
6. 单击【确定】，完成通过策略关联用户操作。

通过用户/用户组关联策略

通过用户关联策略

1. 在访问管理控制台，单击左侧【用户管理】，进入【用户管理】页面。
2. 选择指定用户，单击用户名称，进入【用户详情】页面。
3. 在【已关联的策略】下，单击【关联策略】，打开【关联策略】对话框。
4. 在【关联策略】对话框中，勾选需要授权的策略。
5. 单击【确定】，完成通过用户关联自定义策略操作。

通过用户组关联策略

1. 在访问管理控制台，单击左侧【用户组】，进入【用户管理】页面。
2. 找到需要授权的用户组，单击用户组名称，进入用户组详情页。
3. 在【已关联的策略】下，单击【关联策略】，打开【关联策略】对话框。
4. 在弹出的【关联策略】对话框中，勾选需要授权的策略。
5. 单击【确定】，完成通过用户组关联预设策略操作。

限制IP访问

操作场景

本文档介绍如何通过自定义策略限制子账号访问 IP，设置成功后，子账号将通过所设置的 IP 管理主账号下的资源，或者拒绝子账号通过设置的 IP 管理主账号下资源。

前提条件

需要设置的产品支持按 IP 限制业务访问，详细可参考 常见问题。

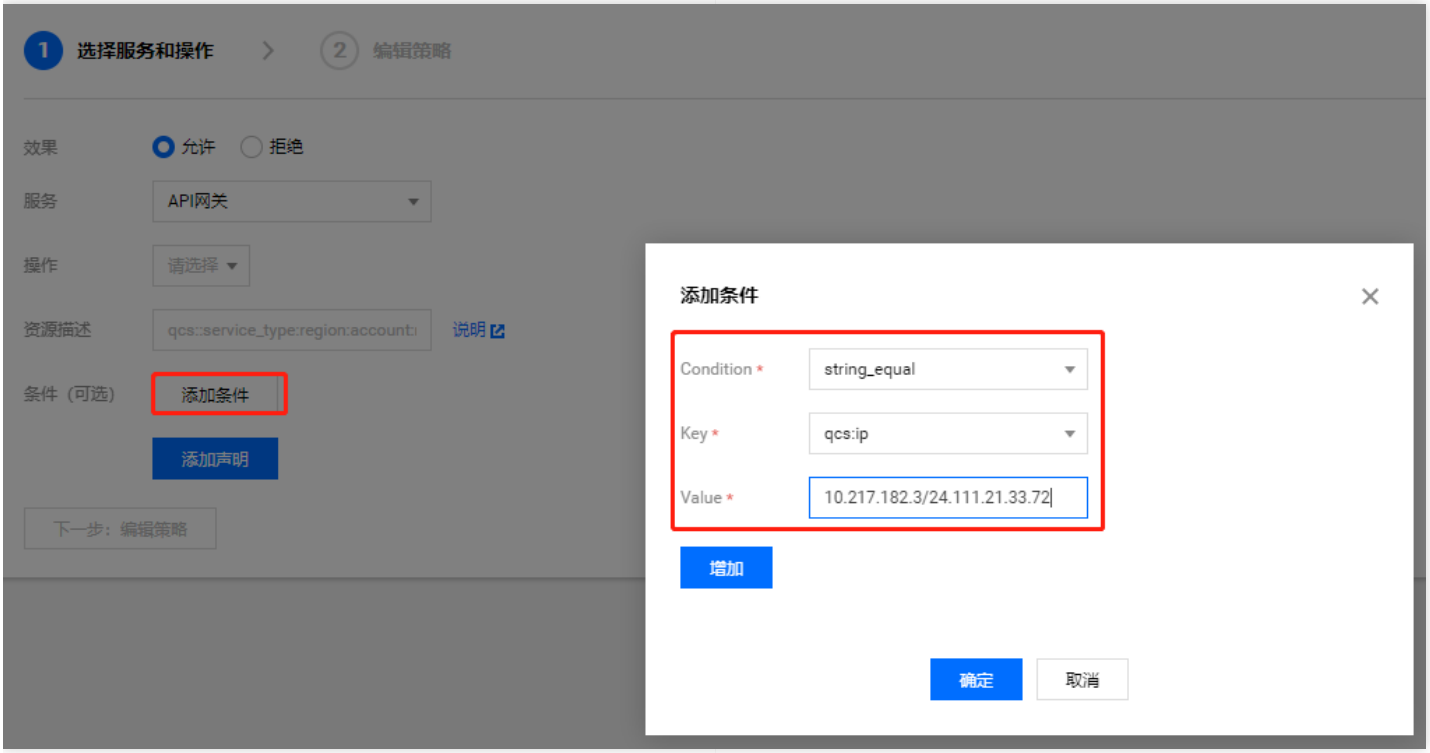
操作步骤

1. 进入 策略管理页面，单击左上角的【新建自定义策略】。
2. 在弹出的选择创建方式窗口中，单击【按策略生成器创建】，进入选择服务和操作页面。
3. 在选择服务和操作页面，补充以下信息。

- 效果：必填项，选择 "允许"。如选择 "拒绝"，用户或用户组不能获取授权。
- 服务：必填项，选择需要添加的产品。
- 操作：必填项，根据您的需求勾选产品权限。
- 资源：必填项，您可以参考 资源描述方式填写。
- 条件：根据您的需求选择条件，输入 IP 地址。可以添加多条限制。例如，效果选择"允许"，仅限使用该 IP 地址的用户或组获取授权。

使用示例

以下示例表示用户必须在 10.217.182.3/24 或者 111.21.33.72/24 网段才能调用云 API 访问 cos:PutObject，如下图：



策略语法如下：

```
{
    "version": "2.0",
    "statement": [
        {
            "effect": "allow",
            "action": "cos:PutObject",
            "resource": "*",
            "condition": {
                "ip_equal": {
                    "qcs:ip": [
                        "10.217.182.3/24",
                        "111.21.33.72/24"
                    ]
                }
            }
        }
    ]
}
```

语法逻辑
元素参考

策略(policy)由若干元素构成，用来描述授权的具体信息。核心元素包括委托人(principal)、操作(action)、资源(resource)、生效条件(condition)以及效力(effect)。元素保留字仅支持小写。它们在描述上没有顺序要求。对于策略没有特定条件约束的情况，condition 元素是可选项。在控制台中不允许写入 principal 元素，仅支持在策略管理 API 中和策略语法相关的参数中使用 principal。

1.版本(version)

描述策略语法版本。该元素是必填项。目前仅允许值为"2.0"。

2.委托人(principal)

描述策略授权的实体。包括用户（开发商、子账号、匿名用户）、用户组，未来会包括角色、联合身份用户等更多实体。仅支持在策略管理API中策略语法相关的参数中使用该元素。

3.语句(statement)

描述一条或多条权限的详细信息。该元素包括 action、resource、condition、effect 等多个其他元素的权限或权限集合。一条策略有且仅有一个statement 元素。

4.操作(action)

描述允许或拒绝的操作。操作可以是 API（以name前缀描述）或者功能集（一组特定的 API，以 permid 前缀描述）。该元素是必填项。

5.资源(resource)

描述授权的具体数据。资源是用六段式描述。每款产品的资源定义详情会有所区别。有关如何指定资源的信息，请参阅您编写的资源声明所对应的产品文档。该元素是必填项。

6.生效条件(condition)

描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。有些服务允许您在条件中指定其他值。该元素是非必填项。

7.效力(effect)

描述声明产生的结果是“允许”还是“显式拒绝”。包括 allow(允许)和deny(显式拒绝)两种情况。该元素是必填项。

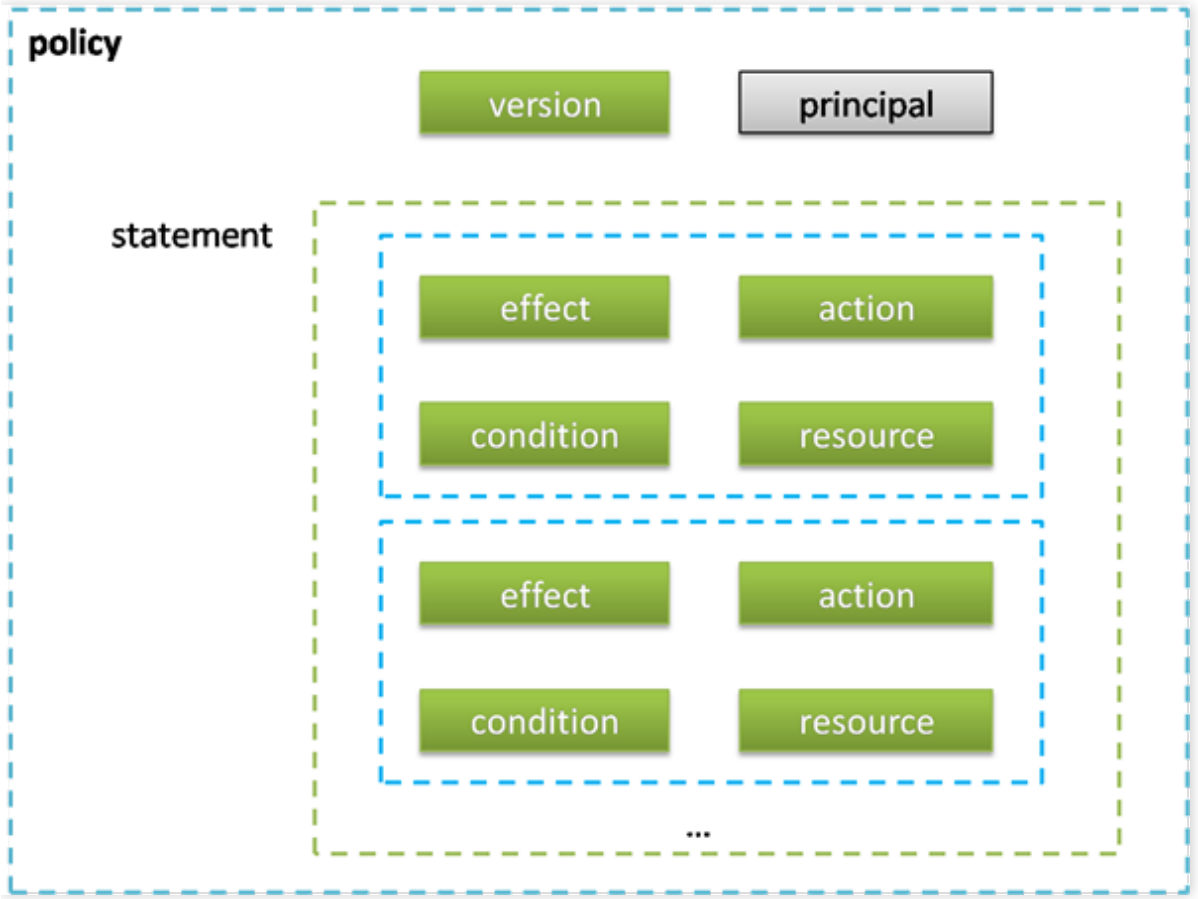
8.策略样例

该样例描述为：允许属于开发商 ID 1238423下的子账号 ID 3232523以及组 ID 18825， 对地域1的cos存储桶 bucketA和地域2的 cos 存储桶 bucketB 下的对象 object2，在访问 IP 为10.121.2.*网段时，拥有所有 cos 读 API 的权限以及写对象的权限，以及可以发送消息队列的权限。

```
{
    "version": "2.0",
    "principal": {
        "qcs": [
            "qcs::cam::uin/1238423:uin/3232523",
            "qcs::cam::uin/1238423:groupid/18825"
        ]
    },
    "statement": [
        {
            "effect": "allow",
            "action": [
                "name/cos:PutObject",
                "permid/280655"
            ],
            "resource": [
                "qcs::cos:bj:uid/1238423:prefix//1238423/bucketA/*",
                "qcs::cos:gz:uid/1238423:prefix//1238423/bucketB/object2"
            ],
            "condition": {
                "ip_equal": {
                    "qcs:ip": "10.121.2.10/24"
                }
            }
        },
        {
            "effect": "allow",
            "action": "name/cmqqueue:Sendmessages",
            "resource": "*"
        }
    ]
}
```

语法结构

整个策略的语法结构如下图所示。策略 policy 由版本 version 和语句 statement 构成，还可以包含委托人信息 principal，委托人仅限于策略管理 API 中策略语法相关的参数中使用。
语句 statement 是由若干个子语句构成。每条子语句包括操作 action、资源 resource、生效条件 condition 以及效力 effect 四个元素，其中 condition 是非必填项。



JSON 格式

策略语法以 JSON 格式为基础。创建或更新的策略不满足 JSON 格式时，将无法提交成功，所以用户必须要确保 JSON 格式正确。 JSON 格式标准在 RFC7159 中定义，您也可以使用在线 JSON 验证程序检查策略格式。

语法约定

语法描述中有如下约定：

- 以下字符是包含在策略语法中的 JSON 字符：

  { } [ ] " , :

- 以下字符是用于描述策略语法中的特殊字符，不包含在策略中：

  = < > ( ) |

- 当一个元素允许多个值时，使用逗号分隔符和省略号进行表示。例如：

  [<resource_string>, < resource_string>, ...]
  <principal_map> = { <principal_map_entry>, > <principal_map_entry>, ... }

允许多个值时，也可以只包含一个值。当元素只有一个值时，尾部的逗号必须去掉，且中括号"[]"标记可选。例如：

"resource": [<resource_string>]
"resource": <resource_string>

- 元素后的问号 (?) 表示该元素是非必填项。例如：

  <condition_block?>

- 元素是枚举值的情况下，枚举值之间用竖线 "|" 表示，并用 "()" 括号定义枚举值的范围。例如：

  ("allow" | "deny")

- 字符串元素用双引号包括起来。例如：

  <version_block> = "version" : "2.0"

语法描述

```
policy={
    <version_block><principal_block?>,
    <statement_block>
}<version_block>="version": "2.0"<statement_block>="statement": [
    <statement>,
    <statement>,
    ...
]<statement>={
    <effect_block>,
    <action_block>,
```

```
    <resource_block>,
    <condition_block?>
}<effect_block>="effect": ("allow"|"deny")<principal_block>="principal": ("*"|<principal_map>)<princi
pal_map>={
    <principal_map_entry>,
    <principal_map_entry>,
    ...
}<principal_map_entry>="qcs": [
    <principal_id_string>,
    <principal_id_string>,
    ...
]<action_block>="action": ("*"|[
    <action_string>,
    <action_string>,
    ...
])<resource_block>="resource": ("*"|[
    <resource_string>,
    <resource_string>,
    ...
])<condition_block>="condition": {
    <condition_map>
}<condition_map>{
    <condition_type_string>: {
        <condition_key_string>: <condition_value_list>
    },
    <condition_type_string>: {
        <condition_key_string>: <condition_value_list>
    },
    ...
}<condition_value_list>=[
    <condition_value>,
    <condition_value>,
    ...
]<condition_value>=("string"|"number")
```

语法说明：

- 一个策略 policy 可以包含多条语句 statement。
  策略的最大长度是 4096 个字符（不包含空格），具体信息请参阅 限制。
  各个块 block 的显示顺序无限制。例如，在策略中， version_block 可以跟在 effect_block 后面等。
- 当前支持的语法版本为 2.0。
- principal_block 元素在控制台中不允许写入，仅支持在策略管理 API 中和策略语法相关的参数中使用 principal 。
- 操作 action 和资源 resource 都支持列表，其中 action 还支持各产品定义的操作集 permid。
- 生效条件可以是单个条件，或者包括多个子条件块的逻辑组合。每个生效条件包括条件操作符

condition_type、条件键 condition_key，条件值 condition_value。

- 每条语句 statement 的效力 effect 为 deny 或 allow 。当策略中包含的语句中既包含有 allow 又包含有 deny 时，遵循 deny 优先原则。

字符串说明

语法描述的元素字符串说明如下：

action_string

由描述作用域、服务类型和操作名称组成。

```
//所有产品所有操作
"action":"*"
"action":"*:*"
// COS 产品所有操作
"action":"cos:*"
// COS 产品的名为 GetBucketPolicy 的操作
"action":"cos:GetBucketPolicy"
// COS 产品部分匹配 Bucket 的操作
"action":"cos:*Bucket*"
//操作集 ID 为 280649 的操作列表
"action":"permid/280649"
// cos 产品，名为 GetBucketPolicy\PutBucketPolicy\DeleteBucketPolicy 的操作列表
"action":["cos:GetBucketPolicy","cos:PutBucketPolicy","cos: DeleteBucketPolicy"]
```

其中，permid 为各产品定义的操作集合 ID ，具体信息请参阅各相关产品文档。

resource_string

资源通过六段式描述。

```
qcs: project :serviceType:region:account:resource
```

示例如下所示：

```
// COS 产品的 object 资源，地域1，资源拥有者的 uid 是10001234，资源名是 bucket1/object2 ，资源前缀是 prefix
qcs::cos:sh:uid/10001234:prefix//10001234/bucket1/object2
// CMQ 产品的队列，地域1，资源拥有者的 uin 是12345678，资源名是12345678/queueName1,资源前缀是 queueName
qcs::cmqqueue:sh:uin/12345678:queueName/12345678/queueName1
// CVM 产品的云服务器，地域1，资源拥有者的 uin 是12345678，资源名是 ins-abcdefg,资源前缀是 instance
qcs::cvm:sh:uin/12345678:instance/ins-abcdefg
```

具体信息请参阅各产品的 支持的资源级权限 页面的资源描述方法。

condition_type_string

条件操作符，描述测试条件的类型。例如 string_equal、string_not_equal、date_equal、date_not_equal、ip_equal、ip_not_equal、numeric_equal、numeric_not_equal 等。示例如下所示:

```
"condition":{
        "string_equal":{"cvm:region":["sh","gz"]},
        "ip_equal":{"qcs:ip":"10.131.12.12/24"}
}
```

condition_key_string

条件键，表示将对其值采用条件操作符进行操作，以便确定条件是否满足。CAM 定义了一组在所有产品中都可以使用的条件键，包括 qcs:current_time、qcs:ip 、qcs:uin 和 qcs:owner_uin 等。具体信息请参阅 生效条件。

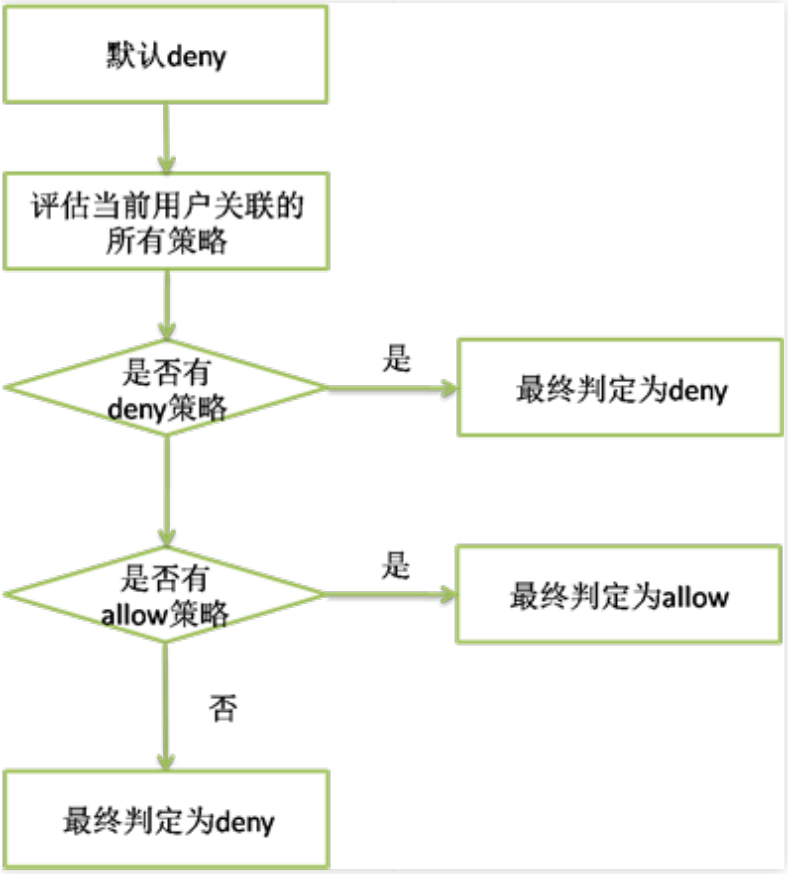
principal_id_string

对于 CAM 而言，用户也是它的资源。因此委托人 principal 也采用六段式描述。示例如下，具体信息请参阅 资源描述方式。

```
"principal":  {"qcs":["qcs::cam::uin/1238423:uin/3232",
          "qcs::cam::uin/1238423:groupid/13"]}
```

评估逻辑

云平台用户访问云资源时， CAM 通过以下评估逻辑决定允许或拒绝。



1. 默认情况下，所有请求都将被拒绝。
2. CAM 会检查当前用户关联的所有策略。
3. 判断是否匹配策略，是则进行下一步判断；否则最终判断为 deny ，不允许访问云资源。
4. 判断是否有匹配 deny 策略，是则最终判定为 deny ，不允许访问云资源；否则进行下一步判断。
5. 判断是否有匹配 allow 策略，是则最终判断为 allow ，允许访问云资源；否则最终判定为 deny，不允许访问云资源。
    - 对于根账号，默认拥有其名下所有资源的访问权限。
    - 有些通用策略，会默认关联所有 CAM 用户。具体请见下文的 通用策略表。
    - 其他策略都必须显式指定，包括 allow 和 deny 策略。
    - 对于支持跨帐号资源访问的业务，存在权限传递的场景，即根账号 A 授权根账号 B 下的某个子账号对其资源的访问权限。这个时候 CAM 会同时校验 A 是否授权给 B 该权限以及 B 是否授权给子帐号该权限，两者同时满足的前提下， B 的子账号才有权访问 A 的资源。
      目前支持的通用策略表如下：

策略说明	策略定义

策略说明	策略定义

查询密钥需要 MFA 验证	{
	"principal":" ",
	"action":"account:QueryKeyBySecretId",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

设置敏感操作需要 MFA 验证	{
	"principal":" ",
	"action":"account:SetSafeAuthFlag",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

绑定 token 需要 MFA 验证	{
	"principal":" ",
	"action":"account:BindToken",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

解绑 token 需要 MFA 验证	{
	"principal":" ",
	"action":"account:UnbindToken",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

修改邮箱需要 MFA 验证	{
	"principal":" ",
	"action":"account:ModifyMail",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

修改手机号需要 MFA 验证	{
	"principal":" ",
	"action":"account:ModifyPhoneNum",
	"resource":"",
	"condition":{"string_equal":{"mfa":"0"}}
	}

资源描述方式

资源 resource 元素描述一个或多个操作对象，如 CVM 资源、COS存储桶等。本文档主要介绍 CAM 的资源描述信息。

六段式

所有资源均可采用下述的六段式描述方式。每种产品都拥有其各自的资源和对应的资源定义详情。有关如何指定资源的信息，请参阅对应的产品文档。
六段式定义方式如下所示：

```
qcs:project_id:service_type:region:account:resource
```

其中：

- qcs 是 qcloud service 的简称，表示是云平台的云资源。该字段是必填项。
- project_id 描述项目信息，仅兼容 CAM 早期逻辑。当前策略语法禁止填写该信息。
- service_type 描述产品简称，如 CVM、CDN等，产品的检测具体细节请参考对应的产品文档。值为 * 的时候表示所有产品。该字段是必填项。
- region 描述地域信息。值为空的时候表示所有地域。云平台新版地域统一命名方式请参考 地域和可用区。
- account 描述资源拥有者的根账号信息。目前支持两种方式描述资源拥有者，uin 和 uid 方式。
- uin 方式，即根账号的 QQ 号，表示为 uin/${uin} ，如 uin/12345678；
- uid 方式，即根账号的 APPID，表示为 uid/${appid} ，如 uid/10001234。
- 值为空的时候表示创建策略的 CAM 用户所属的根账号。目前COS和CAS业务的资源拥有者只能用uid方式描述（如不涉及，无需关注），其他业务的资源拥有者只能用 uin 方式描述。
- resource 描述各产品的具体资源详情。

1. 有几种描述方式，该字段是必填项。

2. 表示某个资源子类下的资源 ID 。如 VPC 产品的 instance/ins-abcdefg 。

   ```
   <resource_type>/<resource_id>
   ```

3. 表示某个资源子类下的带路径的资源 ID 。如 COS 产品的 prefix//10001234/bucket1/object2 。该方式下，支持目录级的前缀匹配。如 prefix//10001234/bucket1/* ，表示 bucket1 下的所有 Object 。

   ```
   <resource_type>/<resource_path>
   ```

4. 表示某个资源子类下的所有资源。如 instance/* 。

<resource_type>/*

5. 表示某产品下的所有资源。

*

6. 在某些场景下，资源 resource 元素也可以用 * 来描述，含义定义如下，详细信息也请参阅对应的产品文档。
7. 操作 action 是需要关联资源的操作时，resource 定义为 * ，表示关联所有资源。
8. 操作 action 是不需要关联资源的操作时，resource 都需要定义为 * 。

CAM 的资源定义

CAM 包含了用户、组、策略等资源，CAM 资源的描述方式如下所示：

根账号：

qcs::cam::uin/164256472:uin/164256472

或

qcs::cam::uin/164256472:root

子账号：

qcs::cam::uin/164256472:uin/73829520

组：

qcs::cam::uin/164256472:groupid/2340

所有用户：

qcs::cam::anonymous:anonymous

或

*

策略：

qcs::cam:: uin/12345678:policyid/*

或

qcs::cam:: uin/12345678:policyid/12423

资源的重要说明

- 资源的拥有者一定是根账号。如果资源是子账号创建的，不会自动拥有资源的访问权限，需要由资源拥有者授权。
- COS、CAS等业务支持跨账号授权资源的访问权限。被授权账号可以通过权限传递方式将资源授权给其子账号。

策略变量

使用场景

场景假设：您希望给每个 CAM 用户授予其创建资源的访问权限。例如您想要设置 COS 资源的创建者默认拥有该资源的访问权限。
如果由资源拥有者(根账号)将资源逐个授权给资源创建者，授权成本很高，需要为每种资源都编写策略并授权给创建者。在这种情况下，您可以通过使用策略变量来实现您的需求。在策略的资源定义中增加占位符描述的创建人信息，该占位符即使策略变量。当鉴权时，策略变量将被替换为来自请求本身的上下文信息。

授予创建者资源读权限的策略描述方式如下：

```
{
      "version":"2.0",
      "statement":
       {
         "effect":"allow",
         "action":"name/cos:Read*",
         "resource":"qcs::cos::uid/1238423:prefix/${uin}/*"
       }
}
```

- 策略变量在每个资源的路径中带上创建人的 uin。如uin 为12356 的用户创建了名为 test 的 bucket，则其对应的资源描述方式为

qcs::cos::uid/1238423:prefix/12356/test

- uin 为 12356 的用户访问该资源时，鉴权过程中会把对应的策略信息的占位符替换为访问者，即

qcs::cos::uid/1238423:prefix/12356/

- 策略中的资源 qcs::cos::uid/1238423:prefix/12356/ 可以通过前缀匹配访问资源 qcs::cos::uid/1238423:prefix/12356/test 。

策略变量的位置

资源元素位置 ：策略变量可以用在资源六段式的最后一段。
条件元素位置 ：策略变量可以用在条件值中。

以下策略表示 VPC 创建者拥有访问权限。

```
{
        "version":"2.0",
        "statement":
        {
           "effect":"allow",
           "action":"name/vpc:*",
           "resource":"qcs::vpc::uin/12357:vpc/*"
           "condition":{"string_equal":{"qcs:create_uin":"${uin}"}}
        }
}
```

策略变量列表

目前支持的策略变量列表如下：

变量名	变量含义
${uin}	当前访问者的子账号 uin 。对于访问者是根账号的情况，它和根账号 uin 一致。
${owner_uin}	当前访问者所属的根账号 uin 。
${app_id}	当前访问者所属的根账号的 APPID 。

生效条件

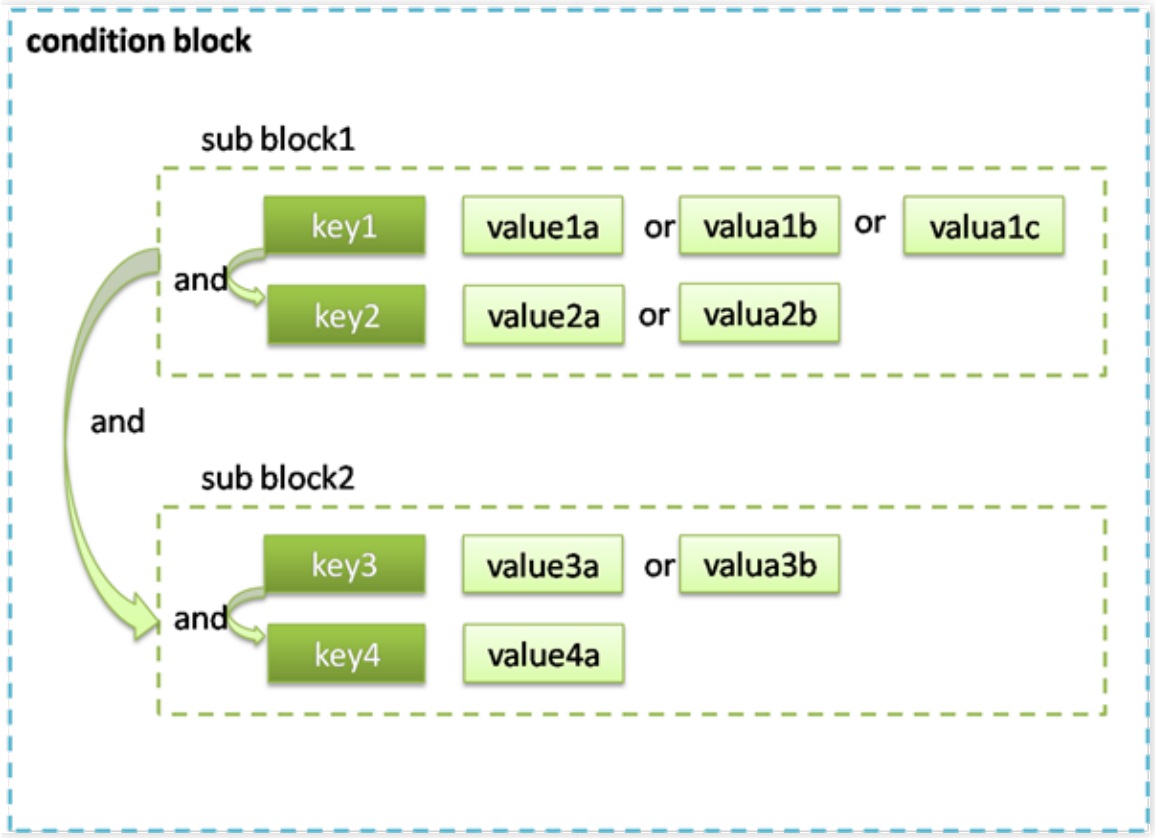
使用场景

在很多场景下，我们需要对创建的策略进一步约束生效的条件 condition 。

- 场景1： CAM 用户调用云 API 时，需要限制用户访问来源，则要求在现有的策略基础上加上 IP 条件。
- 场景2：当 CAM 用户在调用 VPC 对等连接 API 时，除了需要判断 CAM 用户是否拥有对等连接 API 和对等连接资源的访问权限外，还需要确认 CAM 用户是否拥有对等连接关联的 VPC 的访问权限。

语法结构

生效条件的语法结构如下图所示。一个条件块可以由若干个子条件块 sub block 构成，每个子条件块 sub block 对应一个条件操作符和若干个多个条件键，每个条件键对应了若干个条件值。



评估逻辑

条件生效的评估逻辑如下所述：

1. 条件键会对应到多个条件值，只要上下文信息中的对应键值在关联的条件操作符作用下满足其中任意一个条件值，则条件生效。
2. 对于一个子条件块中存在多个条件键的情况下，只有每个条件键对应的条件都生效时，该子条件块才生效。
3. 对于包含多个子条件块的情况，只有每个子条件块都生效时，整个条件才生效。
4. 对于包含 _if_exist 后缀的条件操作符，即使上下文信息中不包含条件操作符所关联的条件键，该条件依然生效。

5. 对于 for_all_value ：限定词约束的条件操作符，适用于上下文信息中的条件键包括多个值的场景。只有当上下文信息中的条件键的每个值在关联的条件操作符作用下生效时，整个条件才生效。
6. 对于 for_any_value ：限定词约束的条件操作符，适用于上下文信息中的条件键包括多个值的场景。只要上下文信息中的条件键的任意一个值在关联的条件操作符作用下生效时，整个条件就可以生效。

使用示例

1. 以下示例表示用户必须在 10.217.182.3/24 或者 111.21.33.72/24 网段才能调用云 API 。

```
{
 "version": "2.0",
 "statement": {
    "effect": "allow",
    "action": "cos:PutObject",
    "resource": "*",
    "condition": {
       "ip_equal": {
          "qcs:ip": [
             "10.217.182.3/24",
             "111.21.33.72/24"
          ]
       }
    }
 }
}
```

2. 以下示例描述允许 VPC 绑定指定的 NAT 网关， VPC 的地域必须是区域1。

```
{
 "version": "2.0",
 "statement": {
    "effect": "allow",
    "action": "name/vpc:AcceptVpcPeeringConnection",
    "resource": "qcs::vpc:sh::pcx/2341",
    "condition": {
       "string_equal_if_exist": {
          "vpc:region": "sh"
       }
    }
 }
}
```

条件操作符列表

下表是条件操作符、条件名以及示例的信息。每个产品自定义的条件键，请参阅对应的产品文档。

条件操作符	含义	条件名	举例
string_equal	字符串等于(区分大小写)	qcs:tag	{"string_equal":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_equal	字符串不等于(区分大小写)	qcs:tag	{"string_not_equal":{"qcs:tag/tag_name1":"tag_value1"}}
string_equal_ignore_case	字符串等于(不区分大小写)	qcs:tag	{"string_equal_ignore_case":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_equal_ignore_case	字符串不等于(不区分大小写)	qcs:tag	{"string_not_equal_ignore_case":{"qcs:tag/tag_name1":"tag_value1"}}
string_like	字符串匹配(区分大小写)	qcs:tag	{"string_like":{"qcs:tag/tag_name1":"tag_value1"}}
string_not_like	字符串不匹配等于(区分大小	qcs:tag	{"string_not_like":{"qcs:tag/tag_name1":"tag_value1"}}

条件操作符	含义	条件名	举例
	写)		
date_not_equal	时间不等于	qcs:current_time	{"date_not_equal":{"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_greater_than	时间大于	qcs:current_time	{" date_greater_than ":{"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_greater_than_equal	时间大于等于	qcs:current_time	{" date_greater_than_equal ":{"qcs:current_time":"2016-06-01T00:01:00Z"}}
date_less_than	时间小于	qcs:current_time	{" date_less_than ":{"qcs:current_time":"2016-06-01T 00:01:00Z"}}
date_less_than_equal	时间小于等于	qcs:current_time	{" date_less_than ":{"qcs:current_time":"2016-06-01T 00:01:00Z"}}
date_less_than_equal	时间小于等于	qcs:current_time	{"date_less_than_equal ":{"qcs:current_time":"2016-06-01T00:01:00Z"}}
ip_equal	ip等于	qcs:ip	{"ip_equal":{"qcs:ip ":"10.121.2.10/24"}}
ip_not_equal	ip不等于	qcs:ip	{"ip_not_equal":{"qcs:ip ":["10.121.2.10/24", "10.121.2.20/24"]}}
numeric_not_equal	数值不等于	qcs:mfa	{" numeric_not_equal":{"mfa":1}}
numeric_greater_than	数值大于		{"numeric_greater_than ":{"cvm_system_disk_size":10}}
numeric_greater_than_equal	数值大于等于		{"numeric_greater_than_equal ":{"cvm_system_disk_size":10}}
numeric_less_than	数值小于		{"numeric_less_than ":{"cvm_system_disk_size":10}}
numeric_less_than_equal	数值小于等于		{"numeric_less_than_equal ":{"cvm_system_disk_size":10}}

条件操作符	含义	条件名	举例
numeric_equal	数值等于	qcs:mfa	{" numeric_equal":{"mfa":1}}
numeric_greater_than	数值大于		{"numeric_greater_than ":{"some_key":11}}
bool_equal	布尔值匹配	-	-
null_equal	条件键为空匹配	-	-

说明：

1. 日期格式按照 ISO8601 标准表示，并需要使用 UTC 时间。
2. IP 格式要符合 CIDR 规范。
3. 条件操作符（null_equal除外）加上后缀 _if_exist，表示上下文信息中即便不包含对应的键值依然生效。
4. for_all_value ：限定词搭配条件操作符使用，表示上下文信息中条件键的每个值都满足要求时才生效。
5. for_any_value ：限定词搭配条件操作符使用，表示上下文信息中条件键的任意一个值满足要求时就可以生效。
6. 部分业务不支持条件，或仅支持部分条件。具体信息参考业务文档说明。

角色管理
角色概述

什么是角色

角色可以看作是TCloudFinanceZone的“虚拟账号”，角色同样可被授予策略，拥有在TCloudFinanceZone中允许执行和拒绝执行的权限。角色可以是任一TCloudFinanceZone账号代入，并不是唯一地与某个账号绑定关联。角色没有关联的持久证书（密码或访问密钥），主账号仅在申请角色时需要使用持久证书，在用户担任某个角色时，则会动态创建临时证书并为用户进相应访问时提供该临时证书，即可通过临时密钥签名调用基础服务的开放 API 来访问用户的云资源。

角色使用场景

云账号角色

云账号角色用于实现跨租户的资源访问。如一些代运维场景，租户A可通过角色来实现让租户B来访问A租户下特定的资源。

基本概念

在您开始使用角色前需要了解一些基本术语，包括角色、服务角色、自定义角色、权限策略等。

角色

拥有一组权限的虚拟身份。用于对角色载体授予TCloudFinanceZone中服务、操作和资源的访问权限。这些权限附加到角色，而不附加到具体的用户或者用户组。

CAM 支持以下 2 种类型的角色：

- 服务（预设）角色：由服务进行预定义的角色，服务角色需经过用户授权，服务即可通过扮演服务角色对用户资源进行访问操作。
- 自定义角色：由用户自行定义的角色，用户可以自由灵活地决定角色载体和角色权限。
  角色可由以下用户使用：
- 可作为角色的主账号。
- 可作为角色的子用户以及协作者。

服务角色

服务角色是各个产品服务直接提供的独特类型的 CAM 预设角色。服务角色的关联权限由相关产品服务预定义，一旦相关产品服务被您赋予服务角色，即该产品服务能够全权代表您调用服务角色权限范围内的其他产品服务。服务角色可以让您更轻松地使用服务，因为在赋予角色的流程中您不必手动添加权限，只需要选择是否给该服务授予服务角色的相关权限。

给相关产品服务赋予服务角色的流程中，服务角色的相关权限和角色载体已经被定义，除非另外定义，否则仅该服务可以代入角色。服务角色的预定义包括角色名称、角色载体、权限策略。

云账号角色

云账号角色是用户自己对 CAM 角色进行定义。自定义角色的角色名称、角色载体以及角色权限均由用户决定。自定义角色可以让您更自由灵活地对您云上资源的访问使用权限进行分配，角色载体为其他租户的主账号。

被您授予角色的对象仅在使用角色的过程中能够获得相关权限，避免给予持久密钥可能带来的安全问题。

权限策略

JSON 格式的权限文档。您可以在权限策略中定义角色可使用的操作和资源。该文档规则依赖于 CAM 策略语言规则。

信任策略

JSON 格式的权限文档。您可以在信任策略中定义可扮演角色的对象以及扮演角色时需满足的条件。该文档规则依赖于 CAM 策略语言规则。

创建角色

操作场景

本文档介绍如何创建角色。创建成功后，角色可以在获得的权限范围内管理主账号下的资源。

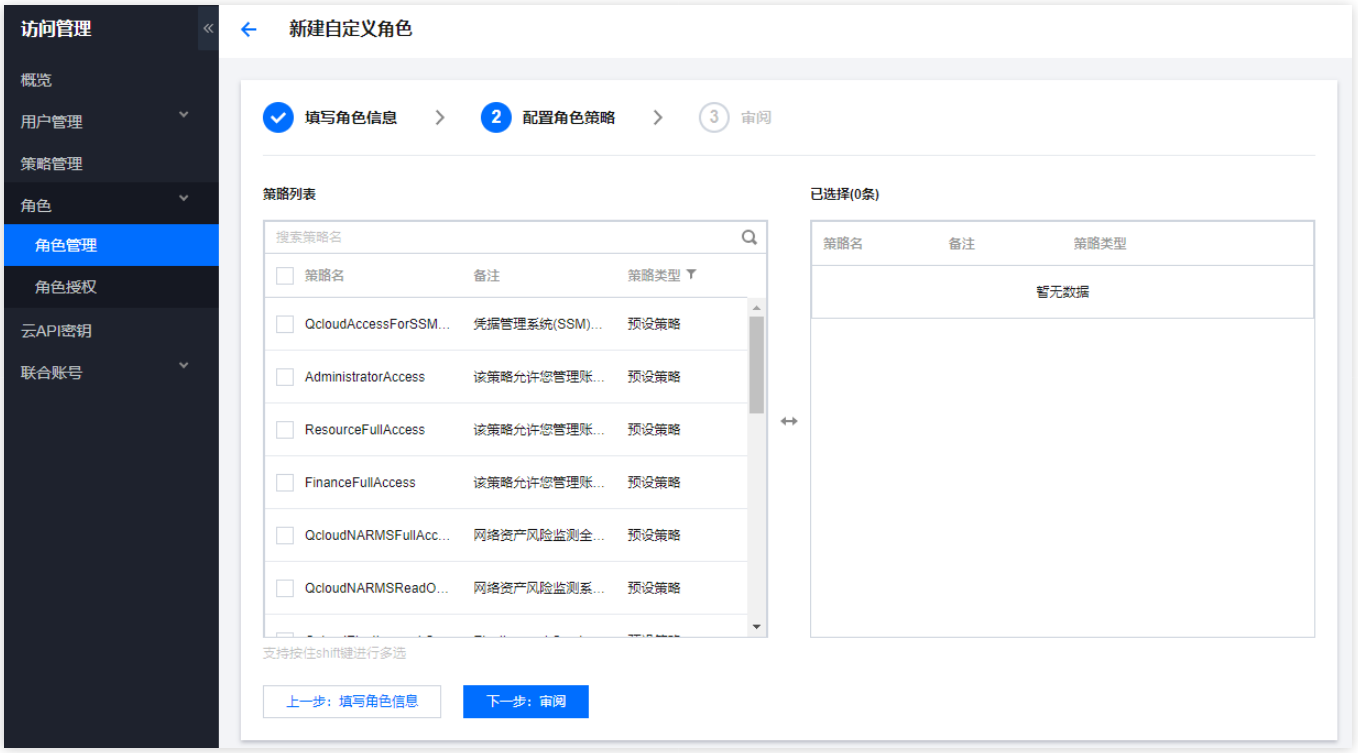
前提条件

已登录访问管理控制台。

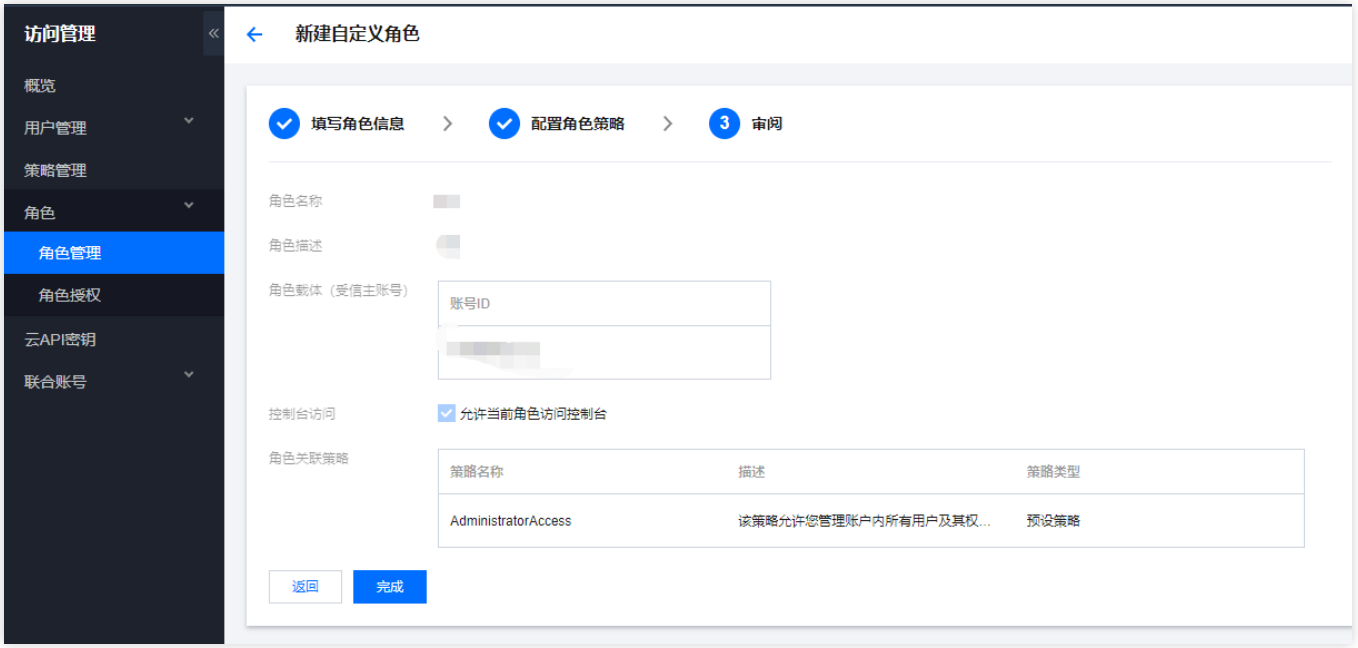
操作步骤

1. 在【访问管理】-【角色】-【角色管理】页面，单击【新建角色】，根据界面提示输入相应参数，单击【下一步：配置角色策略】。

2. 在【策略列表】内勾选您想要给当前角色添加的策略，单击【下一步：审阅】。



3. 对角色相关信息做确认后，单击【完成】。



修改角色

操作场景

本文档介绍如何修改角色关联策略。修改成功后，角色将根据当前设置在获得的权限范围内管理主账号下的资源。

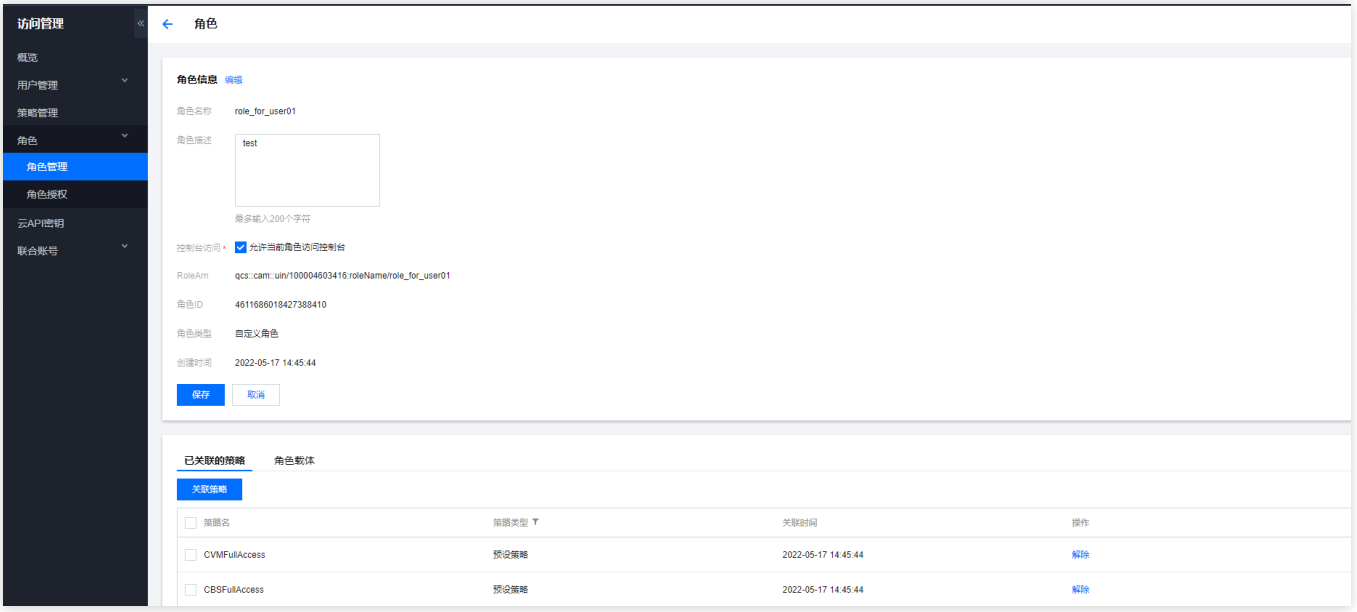
前提条件

已登录访问管理控制台。

操作步骤

修改角色描述

1. 在【角色管理】页面，单击待修改的角色名称，进入角色详情页。
2. 在【角色信息】区域，单击  ，编辑角色描述信息。



3. 单击【保存】，更新角色描述信息。

修改已关联策略

1. 在【角色管理】页面，单击待修改的角色名称，进入角色详情页。
2. 在【已关联的策略】页签，单击【关联策略】。
3. 在弹出的【关联策略】对话框，勾选想要给当前角色添加的策略，单击【确定】，添加角色关联策略。

解除已关联策略

1. 在【角色管理】页面，单击待修改的角色名称，进入角色详情页。
2. 在【已关联的策略】页签，单击待解除策略名对应【操作】列的【解除】。
3. 在弹出的【解除策略】对话框，单击【确认解除】，该角色将无法获得该策略所描述的相关权限。

删除角色

操作场景

本文档介绍如何删除角色。角色删除后，将无法获取相关权限管理账号下的资源。

操作步骤

1. 登录访问管理（CAM）控制台，进入【角色管理】页面。
2. 单击待删除的角色名对应【操作】列的【删除】。

3. 在弹出的【删除角色】对话框中，单击【确定】，删除该角色，同时解除该服务角色已关联的策略及授权关系。

授权角色

1. 在【访问管理】-【角色】-【角色授权】列表里，会列出当前租户可扮演的所有角色列表。
2. 管理员在操作列点授权扮演，可指定当前租户下，哪些子用户可以通过扮演指定角色来访问对应租户下的资源。
3. 被授权扮演之后的子用户，在登录控制台后，可以切换角色。
4. 点击切换角色之后，可以在页面选择要切换的角色名，这里只会列出该子用户有权限的角色列表。

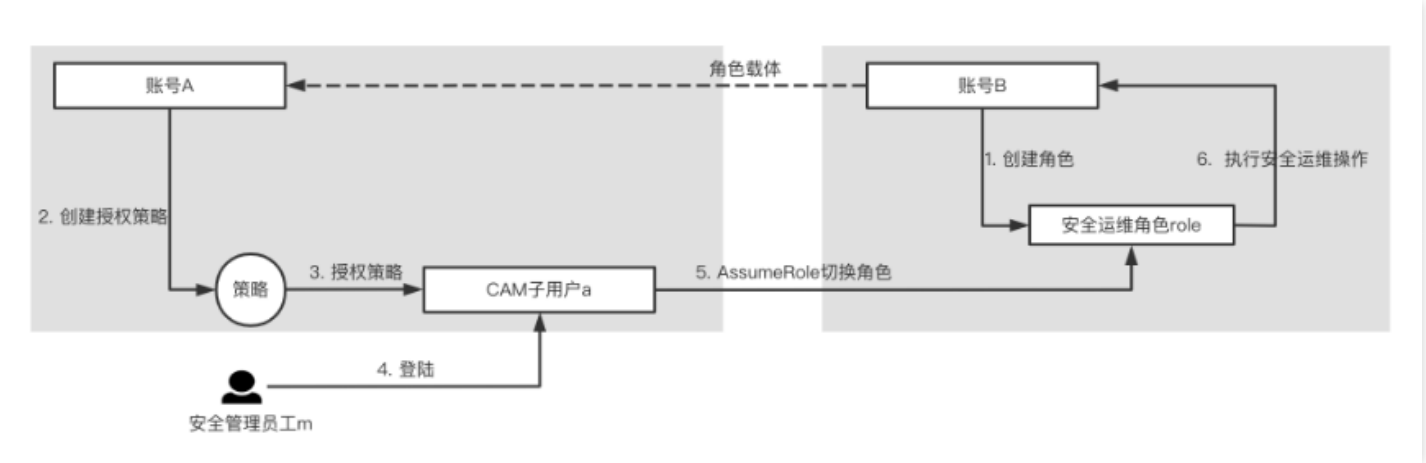
为子账号赋予扮演角色策略

作为角色载体的主账号可以允许其子账号对角色进行扮演，这里我们通过一个案例让您轻松了解如何为子账号创建并赋予扮演角色的策略。
假设如下场景，公司 A 有一个运维工程师的职位，并且希望将该职位外包给公司 B，该职位需要操作公司 A 地域所有云服务器资源。
您可以按以下步骤进行操作：

1. 在公司 A 的账号下，新建一条策略 DevOpsPolicy ，该策略需关联运维操作对应的API。
2. 公司 A 的 账号 CompanyExampleA 下， 创建一个 roleName 为 DevOpsRole 的角色，这个角色的载体设置为公司B，并为角色关联上策略 DevOpsPolicy。
3. 公司 B的管理员登录到控制台，管理员可授权某一个子用户user01扮演DevOpsRole 角色来访问公司A。具体操作如下：在角色授权菜单下，点击授权扮演，选择子用户user01。
4. B公司的子用户user01登录到控制台之后，可点击右上角的切换角色按钮，选择角色DevOpsRole，通过角色扮演的方式登录到公司A的租户下。

最佳实践

在创建角色时，可以选择以主账号作为角色载体、创建角色，并为角色绑定授权策略。 作为载体的主账号可以通过创建权限策略，将扮演角色的权限授予其 CAM 子账号，之后 CAM 子账号可以在控制台通过切换角色登录到对应的主账号控制台执行授权范围内的操作，也可以通过云 API 发起跨账号请求。



操作场景

假设企业内有账号 A 和账号 B 两个主账号，企业安全管理员工 m 在账号 A 下有 CAM 子用户 a，员工 m 希望使用该子账号能够同时运维管理账号 B 下的安全信息。这时我们可以按照以下步骤执行操作：

1、在账号 B 下创建安全运维角色 role，并将角色载体指定为主账号 A。

2、A的管理员登录平台，在访问管理的角色列表里，授权允许m扮演角色role。

3、员工 m 登录 CAM 子用户 a。

4、员工 m 在控制台选择切换角色，使用安全角色 role 登录控制台。

5、执行安全运维相关操作。

6、如果员工 m 需要同时对多个主账号执行安全运维的相关操作，则可以参照上述步骤为员工 m 授予对应主账号的安全运维权限。

访问密钥
查看当前用户访问密钥

操作场景

本文档介绍如何查看当前登录用户的 API 密钥信息。

前提条件

已登录访问管理控制台，进入 云API 密钥管理 页面。

操作步骤

主账号或子账号可以查看和复制当前账号 API 密钥的 SecretId 和 SecretKey 信息，通过 SecretId 和 SecretKey 在权限范围内使用 API、SDK 或其他开发工具管理主账号下的资源。

1. 进入 API 密钥管理页面，在密钥对列可直接获取复制 SecretId。
2. 在密钥对列，单击【显示】，完成身份验证，可以获取复制SecretKey。

> 说明：
>
> 如您的子账号需要自助管理 API 密钥，请授予您的子账号 CamFullAccess 策略 权限。

排除故障
如何根据故障反馈创建策略

如何根据故障反馈创建策略

操作场景

本文档介绍如何通过故障反馈创建策略解除故障，解除之后子账号将在新设置的权限范围内管理主账号下的资源。

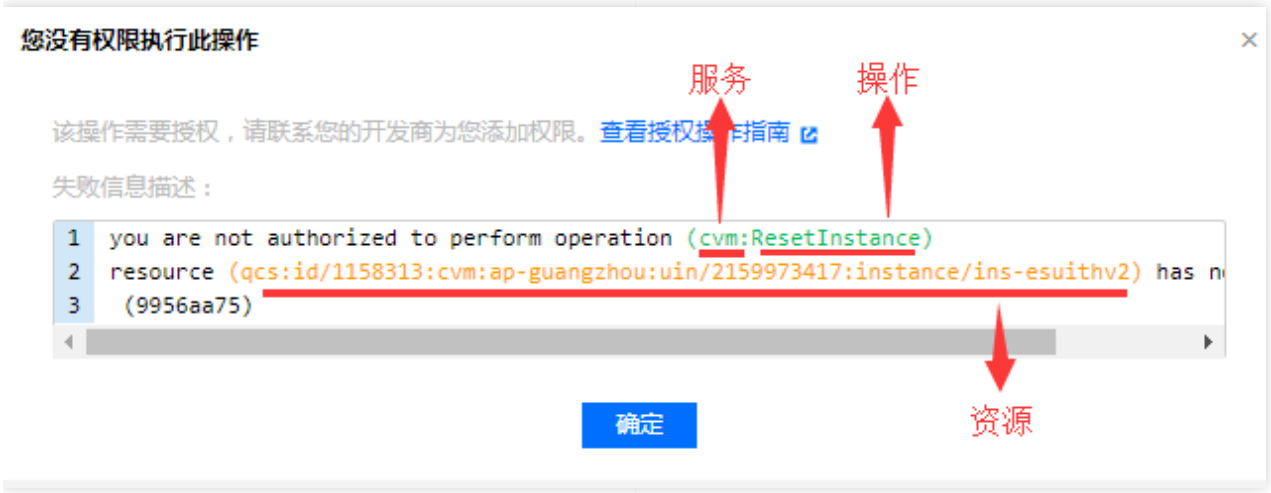
示例

当拥有 QcloudCVMReadOnlyAccess 策略的子账号尝试进行重装云服务器时将进行如下报错：

如您愿意授权子账号继续进行操作，您可以根据当前报错信息为其创建并关联一个自定义策略。

操作步骤

1. 进入 访问管理 的 策略管理菜单，单击新建自定义策略。
2. 在弹出的选择创建方式窗口中，单击【按策略生成器创建】，进入选择服务和操作页面。
3. 在选择服务和操作页面，补充以下信息，如下图所示：



- 效果（必选）：根据授权效果，选择允许还是拒绝。在本次示例中，选择「允许」。
- 服务（必选）：根据产品英文简称选择您要授权的产品。在本次示例中，对应报错信息的 operation 中的「cvm」，您将从产品列表里选择「云服务器」。
- 操作（必选）：选择您要授权的操作。在本次示例中，对应报错信息 operation 中的「ResetInstance」。
- 资源（必填）：填入您要授权的资源的资源六段式。在本次示例中，对应报错信息的「resource」，您可直接复制「qcs:id/1158313:cvm:ap-region:uin/2159973417:instance/instance/ins-esuithv2」填入。
- 条件（选填）：设置子账号上述授权的生效条件，例如指定 IP 才可访问。在本次示例中，不需要填入。

4. 单击【添加声明】>【下一步】，进入编辑策略页面。
5. 在策略编辑页面，补充策略名称、策略备注信息，确认策略内容，其中策略名称和策略内容由控制台自动生成。
    - 策略名称默认为 "policygen" ，后缀数字根据创建日期生成。您可进行自定义。
    - 策略内容与步骤 3 的服务和操作对应，您可根据实 际需求进行修改。
6. 单击【创建策略】，完成按策略生成器创建自定义策略的操作。
7. 为子账号授权，授权成功后，子账号将获得相应的权限，解除故障。

手机收不到验证信息

现象描述

在进行绑定或者修改手机号码、重置密码等操作时，手机收不到验证信息。

可能原因

导致手机收不到验证信息的主要原因包括：

- 手机号码、区号填写错误。
- 手机系统根据关键词，自动隐藏了内容。
- 手机号码自身原因导致的接收异常，如欠费、网络故障等。

处理步骤

1. 请确认手机号码是否填写正确。
    - 是，请执行下一步。
    - 否，请 修改手机号码。
2. 请核实手机是否已停机。
    - 是，请进行缴费或者更换手机号码。
    - 否，请执行下一步。
3. 请确认验证信息是否被视作垃圾短信而被拦截。
    - 是，请解除应用软件的短信拦截。
    - 否，请执行下一步。
4. 网络通讯异常可能会造成短信丢失，请确认网络通讯是否存在异常。

邮箱收不到验证信息

现象描述

在进行绑定或者修改邮箱、重置密码等操作时，邮箱收不到验证信息。

可能原因

导致邮箱收不到验证信息的主要原因包括：

- 邮箱地址填写错误。
- 邮箱系统根据关键词，自动隐藏了内容。
- 邮箱系统存在特殊限制，导致接收失败。例如，企业的自建邮箱禁止接收第三方邮件。

处理步骤

1. 请确认邮箱地址是否填写正确。
    - 是，请执行下一步。
    - 否，请 修改邮箱地址。
2. 请确认验证信息是否被视作垃圾邮件，存放在垃圾箱中。
    - 是，请将云的邮箱设置为白名单。
    - 否，请执行下一步。
3. 网络通讯异常可能会造成邮件丢失，请确认网络通讯是否存在异常。
    - 是，请重新获取或稍后再试。
    - 否，请执行下一步。
4. 请确认邮箱地址是否为企业自建邮箱，且设置了禁止接收第三方邮件。

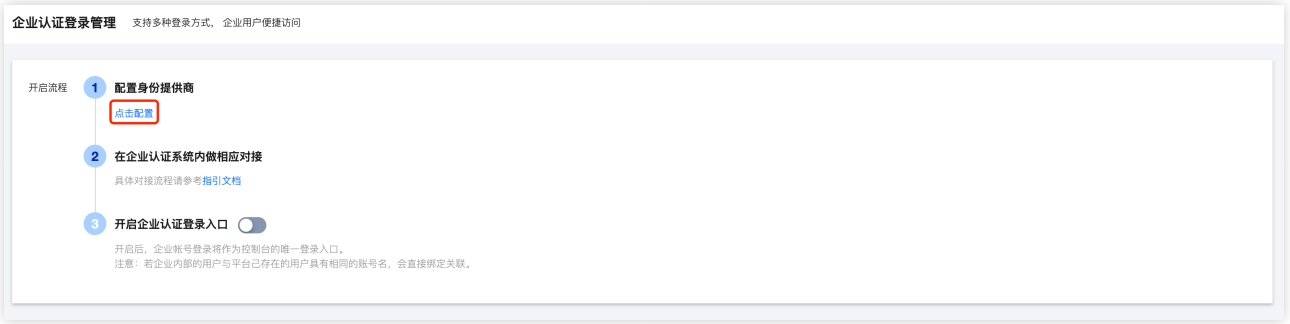
企业认证登录管理

接入准备

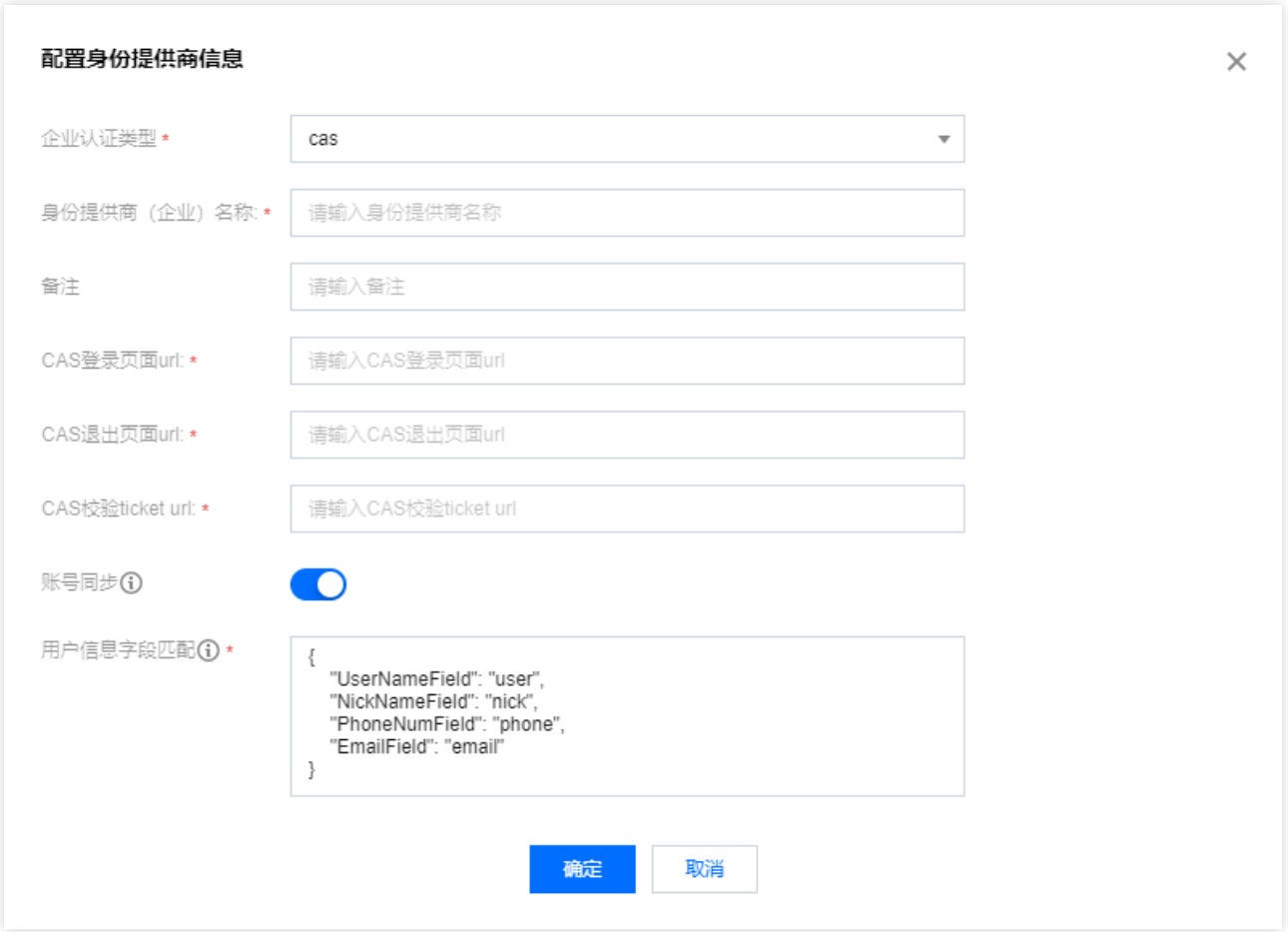
- 企业方需要依据认证类型提供相关服务器。
- 云平台的容器网络需要可访问企业服务器所在网络，如果网络未配置好，请不要开启企业账号登录，否则导致无法登录租户端 。

接入CAS

1. 登录租户端控制台 ，点击【访问管理】>【企业认证登录管理界面】 页面。



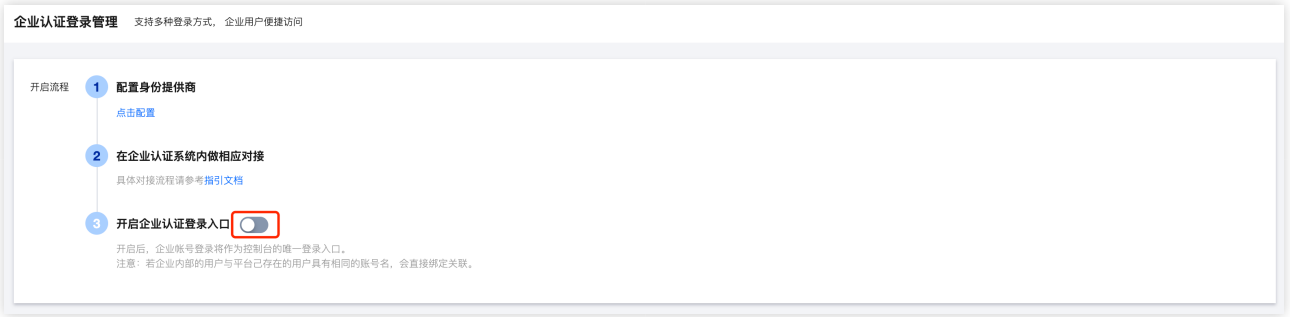
2. 配置企业用户认证CAS相关信息。



- 用户信息字段匹配的意义说明：
  CAS server端需通过CAS校验ticket url将账号server端的用户信息传给平台，用户信息包括用户名、用户昵称、手机号码和邮箱。对于同样的信息，不同server返回的字段名可能不一样，接入方可在用户信息字段匹配这里对字段名做转换，其中，json里的key表示平台读取的字段，value表示server端返回的字段名，举个例子，若server端返回的用户信息里，用户名字段命名的是user，则value里直接填value，若取的字段名是userName，则value里填写userName。
- 账号同步字段意义说明：
  server端需将用户信息传给平台，平台取server端传过来的用户名之后，会检测当前租户下，是否有同名子用户，若平台存在同名子用户，则直接以该子用户身份登录到平台，若当前租户下不存在同名子用户，则会判断账号同步开关是否开启，若账号同步开关开启，则平台会在当前租户下自动创建一个同名子用户，若账号同步开关关闭，则平台会提示，平台不存在子用户，需联系管理员创建。

3. 配置企业 CAS 登录、校验地址（以上url网络必须与TCloudFinanceZone网络可达），相关地址含义具体可以参考CAS协议说明。

4. 开启企业用户认证。



5. CAS校验 ticket url 需返回 xml cas:serviceResponse ，需包含用户名称、昵称、手机、邮箱，否则无法接入成功！其中用户名称、昵称、手机、邮箱字段名称客户可自定义，可在用户信息字段匹配里做映射（参考步骤2）。

参数名称	类型	是否必选	描述
cas:user_id	String	是	企业用户登录名称，1-50个英文字母、数字，支持_-.，不支持空格
cas:user_name	String	是	企业用户昵称
cas:email	String	是	邮箱，必须符合邮箱格式规范
cas:phone	Int	否	手机号码
cas:country_code	String	否	手机号码地区号

企业方账号server校验 serviceValidate。
响应 xml 格式示例：

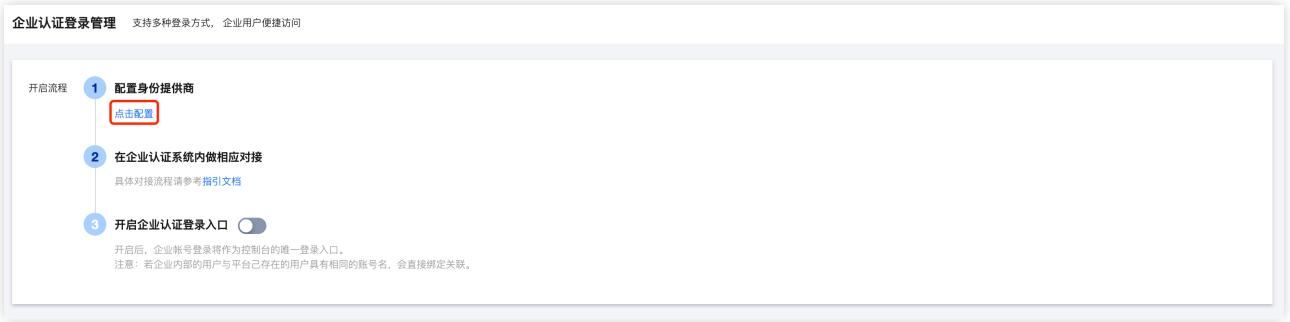
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>

```
    <cas:authenticationSuccess>
        <cas:user>chopin1</cas:user>
            <cas:attributes>
                <cas:user_id>chopin1</cas:user_id>
                <cas:user_name>chopin1</cas:user_name>
                <cas:email>chopin1@mail.com</cas:email>
                <cas:phone>13999999999</cas:phonel>
                <cas:country_code>86</cas:country_code>
            </cas:attributes>
    </cas:authenticationSuccess>
</cas:serviceResponse>
```

6. 网络层通信验证和配置：
    - 进入到TCloudFinanceZone集群master节点，进入到 ocloud-tcenter-mc-idplogin pod容器 shell 终端执行：
        a. 验证pod通往企业cas server网络是否可达，注：以下 url 需要填写企业自身 cas server validate 地址。
        b. curl -v http://cas.finance.cloud.tencent.com/cas/serviceValidte
    - 验证上述步骤网络是否可达，如果可达，以上步骤配置完成。
    - 如果不可达，验证 ocloud-tcenter-mc-idplogin pod 容器是否可以访问外网或者是否可访问企业cas server网络，网络上可达域名上不可达，则在kube-dns配置上访问的域名，如果网络不可达则联系部署交付实施配置网络与企业网络能正常连通。
7. 配置并开启之后，用户在租户端登录时，需输入主账号ID，目前企业认证登录信息是租户粒度的配置，即每个租户可配置自己的账号server，所以需先输入主账号ID，平台根据主账号ID查询对应的server信息；输入正确的主账号ID之后，平台会自动重定向到企业cas server端，若用户在cas server端是已登录状态，则直接以登录态进入控制台；若用户在cas server端是非登录态，则平台会打开cas server端登录页，用户需在cas server端的登录页输入账号密码在server端登录之后，再以登录态进入控制台。

接入OAuth

1. 进入TCloudFinanceZone租户端控制台 ，点击【访问管理】>【企业认证登录管理界面】 页面。



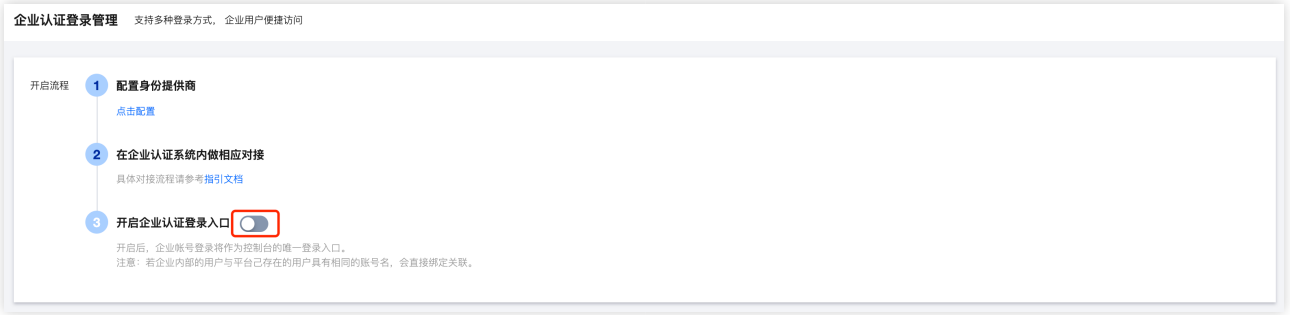
2. 配置企业用户认证OAuth相关信息。

3. 配置企业OAuth备注、ClientId、ClientSecret、Oauth验证授权信息url、获取access_token url、获取用户信息url、账号同步开关和用户信息字段匹配。

- 用户信息字段匹配的意义说明：
  Oauth server端需通过获取用户信息url将账号server端的用户信息传给平台，用户信息包括用户名、用户昵称、手机号码和邮箱。对于同样的信息，不同server返回的字段名可能不一样，接入方可在用户信息字段匹配这里对字段名做转换，其中，json里的key表示平台读取的字段，value表示server端返回的字段名，举个例子，若server端返回的用户信息里，用户名字段命名的是user，则value里直接填value，若取的字段名是userName，则value里填写userName。

- 账号同步字段意义说明：
  server端需将用户信息传给平台，平台取server端传过来的用户名之后，会检测当前租户下，是否有同名子用户，若平台存在同名子用户，则直接以该子用户身份登录到平台，若当前租户下不存在同名子用户，则会判断账号同步开关是否开启，若账号同步开关开启，则平台会在当前租户下自动创建一个同名子用户，若账号同步开关关闭，则平台会提示，平台不存在子用户，需联系管理员创建（如下图所示）。

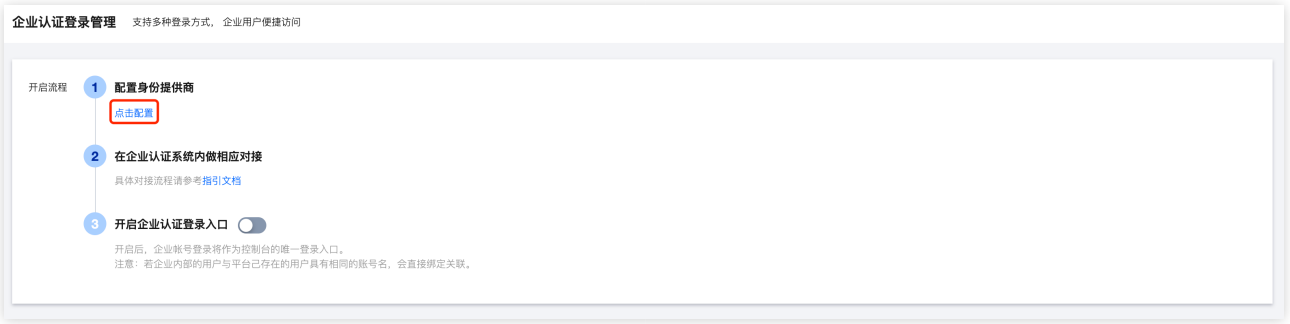
4. 开启企业用户认证。



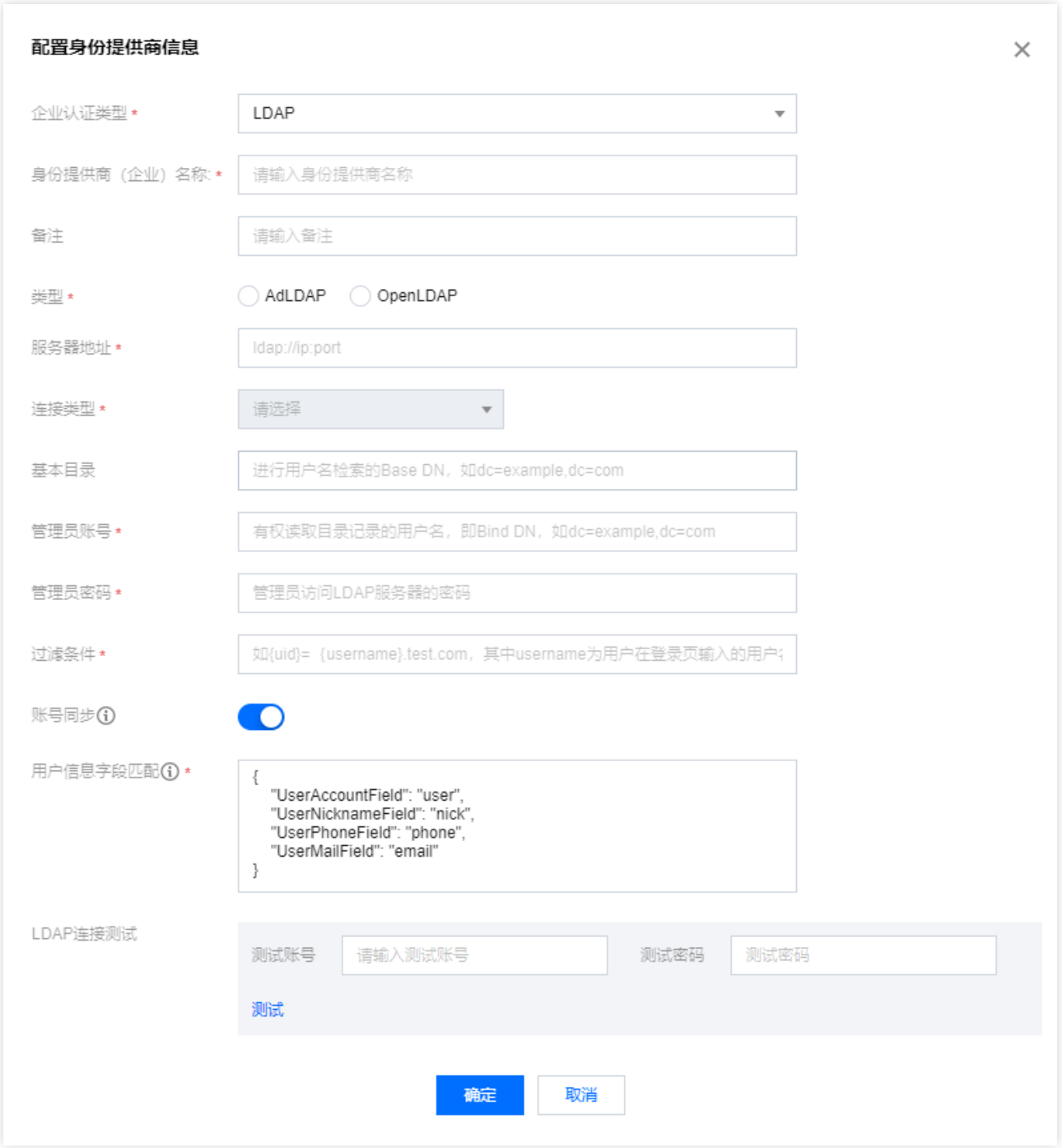
5. 网络层通信验证和配置：
   - 进入到TCloudFinanceZone集群master节点，进入到 ocloud-tcenter-mc-idplogin pod容器 shell 终端执行：
     a. 验证pod通往企业cas server网络是否可达，注：以下 url 需要填写企业自身 cas server validate 地址。
     b. curl -v http://cas.finance.cloud.tencent.com/cas/serviceValidte
   - 验证上述步骤网络是否可达，如果可达，以上步骤配置完成。
   - 如果不可达，验证 ocloud-tcenter-mc-idplogin pod 容器是否可以外网或者是否可访问企业cas server网络，网络上可达域名上不可达，则在kube-dns配置上访问的域名，如果网络不可达则联系部署交付实施配置网络与企业网络能正常连通。
6. 配置并开启之后，用户在租户端登录时，需输入主账号ID，目前企业认证登录信息是租户粒度的配置，即每个租户可配置自己的账号server，所以需先输入主账号ID，平台根据主账号ID查询对应的server信息；输入正确的主账号ID之后，平台会自动重定向到企业server端，若用户在server端是已登录状态，则直接以登录态进入控制台；若用户在server端是非登录态，则平台会打开server端登录页，用户需在server端的登录页输入账号密码在server端登录之后，再以登录态进入控制台。

接入LDAP

1. 进入TCloudFinanceZone租户端控制台 ，点击【访问管理】>【企业认证登录管理界面】 页面。



2. 配置企业用户认证LDAP相关信息。
   其中，服务器地址、连接类型、基本目录、管理员账号、管理员密码、过滤条件请参考LDAP协议说明。



- 用户信息字段匹配的意义说明：
  LDAP server端需将账号server端的用户信息传给平台，用户信息包括用户名、用户昵称、手机号码和邮箱。对于同样的信息，不同server返回的字段名可能不一样，接入方可在用户信息字段匹配这里对字段名做转换，其中，json里的key表示平台读取的字段，value表示server端返回的字段名，举个例子，若server端返回的用户信息里，用户名字段命名的是user，则value里直接填value，若取的字段名是userName，则value里填写userName。
- 账号同步字段意义说明：
  server端需将用户信息传给平台，平台取server端传过来的用户名之后，会检测当前租户下，是否有同名子用户，若平台存在同名子用户，则直接以该子用户身份登录到平台，若当前租户下不存在同名子用户，则会判断账号同步开关是否开启，若账号同步开关开启，则平台会在当前租户下自动创建一个同名子用户，若账号同步开关关闭，则平台会提示，平台不存在子用户，需联系管理员创建（如下图所示）。

- LDAP连接测试
  配置好之后，可输入server端的账号名和密码做测试，测试成功，则显示测试成功（如图所示）。

若测试失败，则会 根据失败原因显示不通失败信息。

3. 配置并开启之后，用户在租户端登录时，需输入主账号ID，目前企业认证登录信息是租户粒度的配置，即每个租户可配置自己的账号server，所以需先输入主账号ID，平台根据主账号ID查询对应的server信息。

4. 输入主账号ID之后，需用户再输入该用户在LDAP server端的账号密码，平台会用用户输入的账号密码去server端验证，验证成功之后，则以登录态进入控制台，验证失败则无法进入控制台。

企业微信账号

1. 联合账号

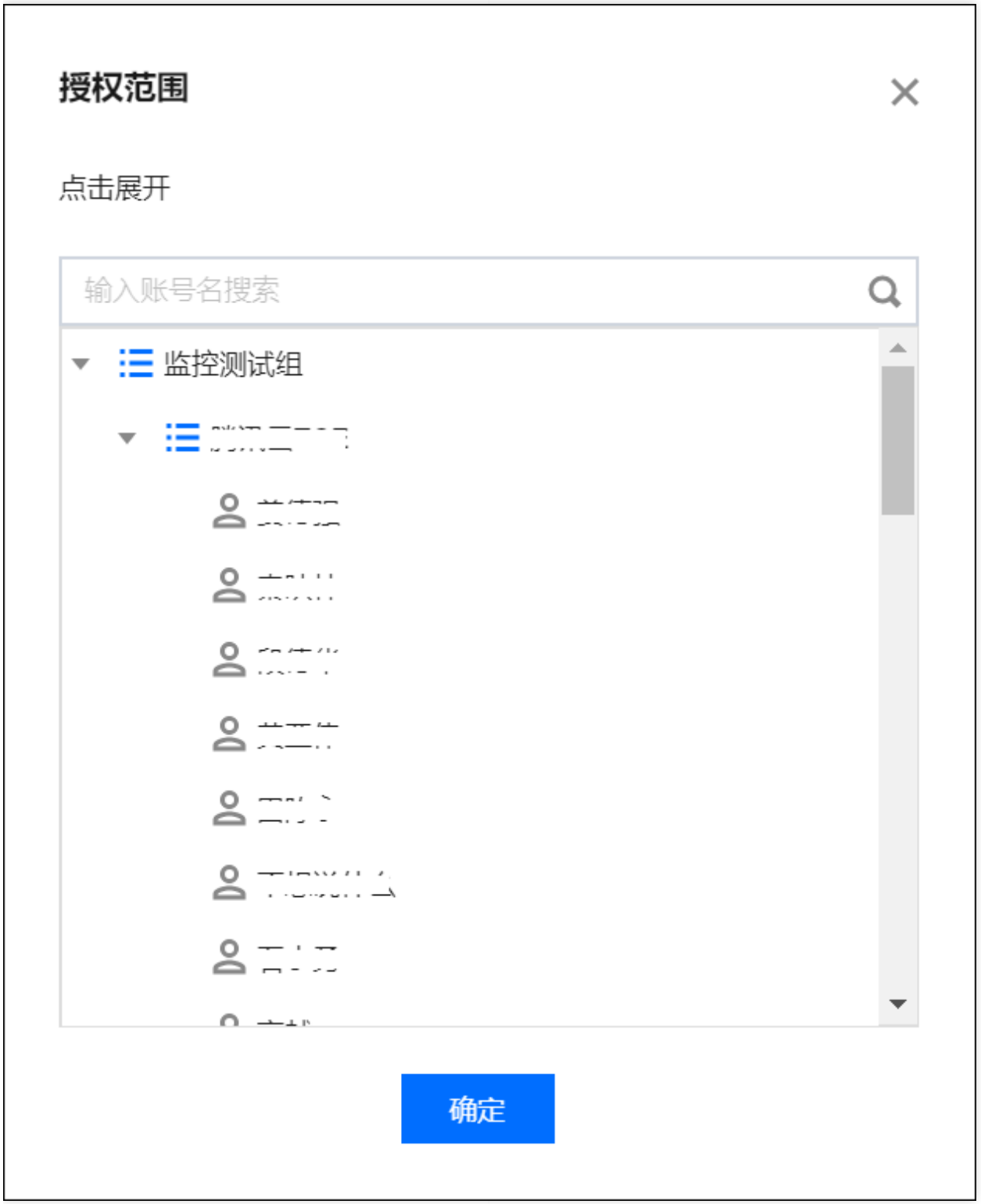
联合账号用来获取企业微信成员账号信息。

1.1 企业微信

1. 用户需要先注册企业微信，并创建企业微信的“自建应用”。
2. 获取企业微信及自建应用的相关信息：Corpid、AgentId、CorpSecret。
3. 登录云平台，进入【访问管理】>【联合账号】>【企业微信】管理页面。
4. 点击【关联企业微信账号】，将弹出“关联企业微信账号”对话框。
5. 在对话框汇总输入：AppName、Corpid、AgentId、CorpSecret

1.2 查看企业微信成员

1. 登录云平台，进入【访问管理】>【联合账号】>【企业微信】管理页面。
2. 企业微信信息列表中，点击【点击查看】，查看“自建应用”的企业微信授权成员。



2. 用户

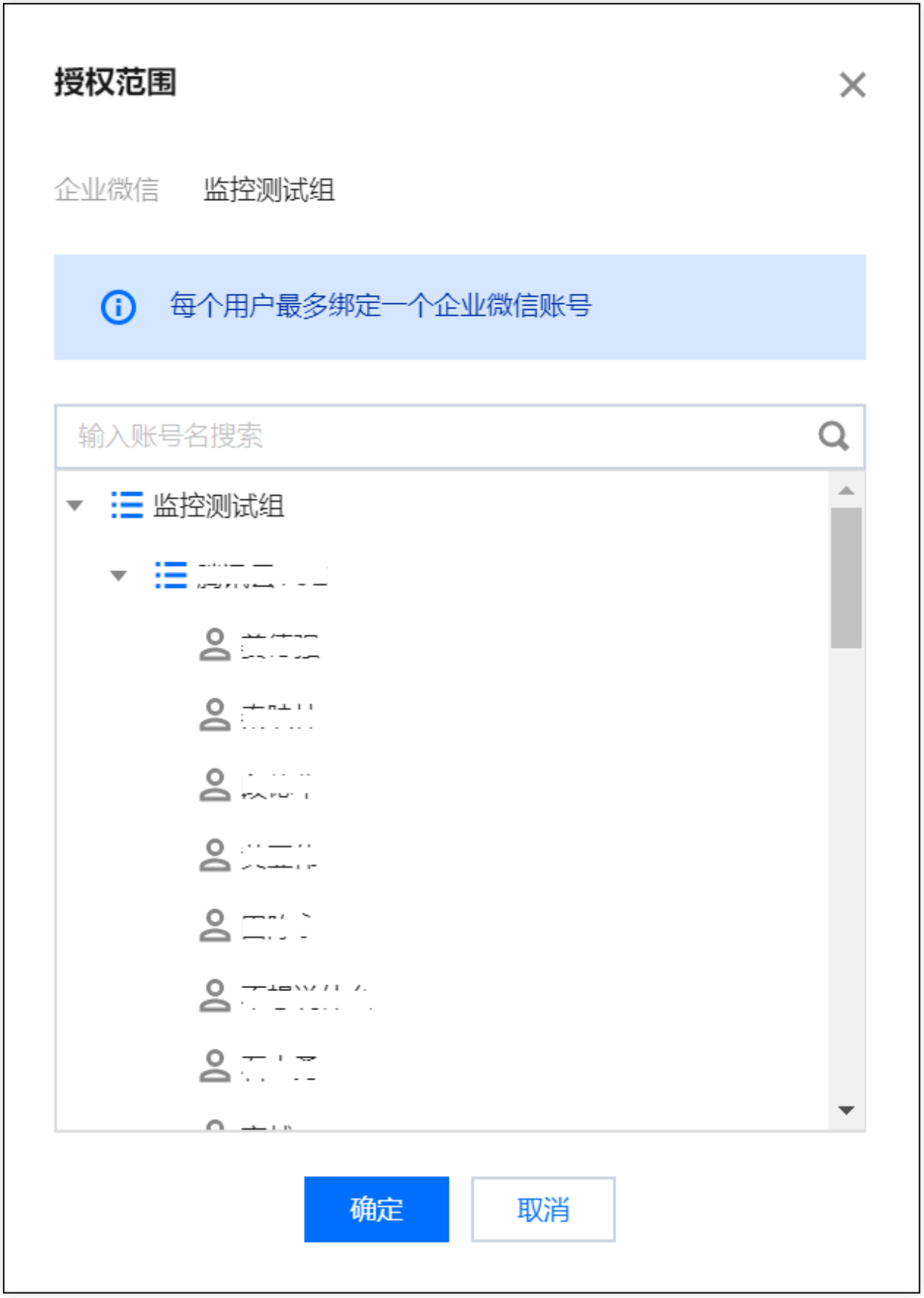
将企业微信成员的账号信息关联云平台用户，以便能够通过这些账号信息将消息发送给云平台用户的企业微信。

2.1 用户账号关联企业微信

1. 登录云平台，进入【访问管理】>【用户】管理页面。

2. 在用户列表的“操作”栏点击【关联企业微信】，将弹出企业微信成员对话框。

3. 在对话框汇选择要关联的企业微信成员。



4. 点击【确定】完成配置。

最佳实践
CMQ相关案例
授权子帐号拥有消息服务的所有权限

授权子帐号拥有消息服务的所有权限

企业帐号CompanyExample下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample名下的消息队列的所有权限，无论消息队列是主题模型还是队列模型，都可以被读写。
方案A：
企业帐号CompanyExample直接将预设策略QCloudCmqQueueFullAccess和QCloudCmqTopicFullAccess授权给子账号Developer。授权方式请参考授权管理。
方案B：
step1：通过策略语法方式创建以下策略

```
{
  "version": "2.0",
  "statement":
   {
      "effect": "allow",
      "action": ["cmqtopic:*","camqueue:*"]
      "resource": "*"
   }
}
```

step2：将该策略授权给子账号。授权方式请参考授权管理。

授权子帐号拥有其创建的消息队列的所有权限

授权子帐号拥有其创建的消息队列的所有权限

企业帐号CompanyExample下有一个子账号Developer，该子账号希望其可以访问自己创建的消息队列。
方案A：
企业帐号CompanyExample直接将预设策略QCloudCmqQueueCreaterFullAccess和QCloudCmqTopicCreaterFullAccess授权给子账号Developer。授权方式请参考授权管理。
方案B：
step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement":
    [
      {
         "effect": "allow",
         "action": "cmqtopic:*",
         "resource": "qcs::cmqtopic:::topicName/uin/${uin}/*"
      },
      {
         "effect": "allow",
         "action": "cmqqueue:*",
         "resource": "qcs::cmqqueue:::queueName/uin/${uin}/*"
      }
    ]
}
```

step2：将该策略授权给子账号。授权方式请参考授权管理。

授权子帐号拥有特定的主题模型的消息队列的读权限

授权子帐号拥有特定的主题模型的消息队列的读权限

企业帐号CompanyExample（ownerUin为12345678）有一个基于主题模型的消息队列，同时他有一个子账号Developer，希望其可以访问该消息队列。
step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement":
    {
        "action": "cmqqueue:SendMessage",
        "resource":"qcs::cmqqueue:::queueName/uin/12345678/test-caten",
        "effect": "allow"
    }
}
```

step2：将该策略授权给子账号。

CVM相关案例
授权子账号拥有CVM的所有权限

授权子账号拥有CVM的所有权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的完全管理权限（创建、管理、云服务器下单支付等全部操作权限）。

方案A：

企业帐号CompanyExample直接将预设策略QcloudCVMFullAccess、QcloudCVMFinanceAccess授权给子账号Developer。

方案B：

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement":[
        {
            "effect": "allow",
            "action": "cvm:*",
            "resource": "*"
        },
        {
              "effect": "allow",
              "action": "finance:*",
              "resource": "qcs::cvm:::*"
        }
    ]
}
```

step2：将该策略授权给子账号。授权方式请参考授权管理。

授权子账号拥有CVM的只读权限

授权子账号拥有CVM的只读权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的查询CVM实例的权限，但是不具有创建、删除、开关机的权限。

方案A：

企业帐号CompanyExample直接将预设策略QcloudCVMInnerReadOnlyAccess授权给子账号Developer。

方案B：

step1：通过策略语法方式创建以下策略

```
{
  "version": "2.0",
  "statement":
   {
      "effect": "allow",
      "action": [
            "cvm:Describe*",
            "cvm:Inquiry*"
            ],
      "resource": "*"
   }
}
```

step2：将该策略授权给子账号。

授权子账号拥有CVM相关资源的只读权限

授权子账号拥有CVM相关资源的只读权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的查询 CVM 实例及相关资源（VPC 、CLB）的权限，但是不具有创建、删除、开关机的权限。

方案A：

企业帐号CompanyExample直接将预设策略QcloudCVMReadOnlyAccess授权给子账号Develope

方案B：

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:Describe*",
                "cvm:Inquiry*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "vpc:Describe*",
                "vpc:Inquiry*",
                "vpc:Get*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "action": [
                "clb:Describe*"
            ],
            "resource": "*",
            "effect": "allow"
        },
        {
            "effect": "allow",
            "action": "monitor:*",
            "resource": "*"
```

```
        }
    ]
}
```

step2：将该策略授权给子账号。

授权子账号拥有弹性云盘的操作权限

授权子账号拥有弹性云盘的操作权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的查看CVM控制台中的云硬盘信息，创建云硬盘，使用云硬盘的权限。

方案A：

企业帐号CompanyExample直接将预设策略QcloudCBSFullAccess授权给子账号Developer。

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:CreateCbsStorages",
                "cvm:AttachCbsStorages",
                "cvm:DetachCbsStorages",
                "cvm:ModifyCbsStorageAttributes",
                "cvm:DescribeCbsStorages",
                "cvm:DescribeInstancesCbsNum",
                "cvm:RenewCbsStorage",
                "cvm:ResizeCbsStorage"
                ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号。

注：如果不允许子账号修改云硬盘属性，请去掉上述策略语法的"cvm:ModifyCbsStorageAttributes"。

授权子账号拥有安全组的操作权限

授权子账号拥有安全组的操作权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的查看CVM控制台中的安全组，并且使用安全组的权限。

以下策略允许子账号在CVM 控制台中具有创建，删除安全组的权限。

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:DeleteSecurityGroup",
                "cvm:CreateSecurityGroup"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号

以下策略允许子账号在CVM 控制台中具有创建、删除修改安全组策略的权限。

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:ModifySecurityGroupPolicy",
                "cvm:CreateSecurityGroupPolicy",
                "cvm:DeleteSecurityGroupPolicy"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号。

授权子账号拥有弹性IP地址的操作权限

授权子账号拥有弹性IP地址的操作权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的查看CVM控制台中的弹性IP地址，并且使用弹性IP地址的权限。

step1：通过策略语法方式创建以下策略。

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:AllocateAddresses",
                "cvm:AssociateAddress",
                "cvm:DescribeAddresses",
                "cvm:DisassociateAddress",
                "cvm:ModifyAddressAttribute",
                "cvm:ReleaseAddresses"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号。

以下策略允许子账号查看弹性IP地址并可以将其分配给实例并与之相关联。子账号可以修改弹性IP地址的属性、取消弹性IP地址的关联或释放弹性IP地址。

step1：通过策略语法方式创建以下策略。

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "cvm:DescribeAddresses",
                "cvm:AllocateAddresses",
                "cvm:AssociateAddress"
            ],
            "resource": "*",
            "effect": "allow"
        }
```

```
    ]
}
```

step2：将该策略授权给子账号。/130622231683297280/130622242107191296)。

授权子账号拥有特定CVM的操作权限

授权子账号拥有特定CVM的操作权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的指定CVM机器（id为ins-1, 地域1）的操作权限。

step1：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "cvm:*",
            "resource": "qcs::cvm:gz::instance/ins-1",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号。

授权子账号拥有特定地域的CVM的操作权限

授权子账号拥有特定地域的CVM的操作权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的地域1所有机器的操作权限。

step1：企业帐号CompanyExample直接将预设策略QcloudCVMReadOnlyAccess授权给子账号Developer。

step2：通过策略语法方式创建以下策略

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "cvm:*",
            "resource": "qcs::cvm:gz::*",
            "effect": "allow"
        }
    ]
}
```

step2：将该策略授权给子账号。

授权子账号拥有CVM的所有权限但不包括支付权限

授权子账号拥有CVM的所有权限但不包括支付权限

企业帐号CompanyExample（ownerUin为12345678）下有一个子账号Developer，该子账号需要拥有对企业帐号CompanyExample的CVM服务的所有权限管理权限（创建、管理等全部操作），但不包括支付权限，可以下单但无法支付。

方案A：

企业帐号CompanyExample直接将预设策略QcloudCVMFullAccess授权给子账号Developer。

方案B：

step1：通过策略语法方式创建以下策略

```
{
  "version": "2.0",
  "statement":[
      {
          "effect": "allow",
          "action": "cvm:*",
          "resource": "*"
      }
  ]
}
```

step2：将该策略授权给子账号。

授予子账号不支持项目的产品的查看权限

案例场景

企业账号 CompanyExample（ownerUin为12345678）下有一个子账号 Developer，需要授权子账号在控制台查看快照、镜像、VPC、弹性公网 IP 等产品。

操作指引

授权子账号 QcloudCVMAccessForNullProject 预设策略。

API文档
访问管理（cam）
版本（2019-01-16）
API 概览

API版本

V3

其他接口

接口名称	接口功能
AttachRolePolicies	绑定多个策略到角色
AttachRolePolicy	绑定权限策略到角色
AttachRolesPolicy	绑定多个角色到策略
CreatePolicy	创建策略
CreateRole	创建角色
DeletePolicy	删除策略
DeleteRole	删除角色
DescribeRoleList	获取角色列表
DetachGroupPolicies	解除绑定多个策略到用户组
DetachGroupsPolicy	解除绑定策略到多个用户组
DetachUsersPolicy	解除绑定策略到多个用户
GetPolicy	查看策略详情
GetRole	获取角色详情
GetServiceRoleInfo	获取服务角色信息

接口名称	接口功能
ListAttachedGroupPolicies	查询用户组关联的策略列表
ListAttachedRolePolicies	获取角色绑定的策略列表
ListEntitiesForPolicy	查询策略关联的实体列表
ListPolicies	查询策略列表
UpdateAssumeRolePolicy	修改角色信任策略
UpdatePolicy	更新策略

用户相关接口

接口名称	接口功能
GetPasswordRules	获取CAM密码规则
GetSubsGroup	子账户所属用户组列表
GetUinBySecretId	根据SecretId查询Uin
UpdatePasswordRules	更新CAM密码规则

身份提供商接口

接口名称	接口功能
CreateOauthProvider	新增oauth配置
GetUserAccessToken	获取用户oauth标识
RefreshUserToken	刷新用户userAccessToken
UpdateOauthProvider	更新Oauth配置信息
VerifyUserAccessToken	验证用户userAccessToken

调用方式
接口签名v1

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（ Signature ）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

参数名称	中文	参数值

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 Action=DescribeInstances 。
注意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.finance.cloud.tencent.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 / 。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。
首先使用 HMAC-SHA1 算法对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的TCloudFinanceZone SDK 3.0 ，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java

- PHP
- Go
- Node

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK 。

最终输出的 url 可能为： https://cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url ，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp 。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.finance.cloud.tencent.com/?");
```

```
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.finance.cloud.tencent.com/?");
        // 实际请求的url中对参数顺序没有要求
        for (String k : params.keySet()) {
            // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
        params.put("Action", "DescribeInstances"); // 公共参数
        params.put("Version", "2017-03-12"); // 公共参数
        params.put("Region", "shjr"); // 公共参数
        params.put("Limit", 20); // 业务参数
        params.put("Offset", 0); // 业务参数
        params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： pip install requests 。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

接口签名v3

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
    HTTPRequestMethod + '\n' +
    CanonicalURI + '\n' +
    CanonicalQueryString + '\n' +
    CanonicalHeaders + '\n' +
    SignedHeaders + '\n' +
    HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- CanonicalURI：URI 参数，API 3.0 固定为正斜杠（/）；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1）头部 key 统一转成小写；2）多个头部 key（小写）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
    Algorithm + \n +
```

```
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

> 1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
> 2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // 注意时区，否则容易出错
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* 步骤 1：拼接规范请求串 *************
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* 步骤 2：拼接待签名字符串 *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* 步骤 4：拼接 Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

```
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)

# ************* 步骤 3：计算签名 *************
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* 步骤 4：拼接 Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

TCloudFinanceZone API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data（仅部分接口支持），必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0， InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

公共参数

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

其他接口
绑定多个策略到角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

绑定多个策略到角色

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolePolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	Uint64	角色ID(与角色名称必传一项) 示例值： 4611686018427387904
RoleName	否	否	String	角色名称(与角色ID必传一项) 示例值： QCS_RoleName
PolicyId	否	否	Array of Uint64	策略ID list(与策略名 list必传一项) 示例值： [1001,1002]
PolicyName	否	否	Array of String	策略名 list(与策略ID list必传一项) 示例值： ["policygen-1","policygen-2"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.PolicyFull	用户策略数超过上限。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
InvalidParameter.PasswordLengthTooShort	密码太短。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。

绑定权限策略到角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（AttachRolePolicy）用于绑定策略到角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	否	否	Uint64	策略ID，入参PolicyId与PolicyName二选一 示例值： 1
AttachRoleId	否	否	String	角色ID，用于指定角色，入 参 AttachRoleId 与 AttachRoleName 二选一 示例值： 4611686018427397905
AttachRoleName	否	否	String	角色名称，用于指定角色，入 参 AttachRoleId 与 AttachRoleName 二选一 示例值： QCS_RoleName
PolicyName	否	否	String	策略名，入参PolicyId与PolicyName二选一 示例值： policygen-20141112201913

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。

绑定多个角色到策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

绑定多个角色到策略

默认接口请求频率限制：20次/秒。

接口更新时间：2023-02-09 15:36:51。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolesPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	Array of Uint64	角色ID(与角色名称必传一项) 示例值： [4611686018427387904,4611686018427387905]
RoleName	否	否	Array of String	角色名称(与角色ID必传一项) 示例值： ["role-1","role-2"]
PolicyId	否	否	Uint64	策略ID(与策略名必传一项) 示例值： 1000
PolicyName	否	否	String	策略名(与策略ID必传一项) 示例值： policygen-20141112201913

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.PolicyFull	用户策略数超过上限。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
InvalidParameter.PasswordLengthTooShort	密码太短。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。

创建策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（CreatePolicy）可用于创建策略。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-02 16:17:55。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreatePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyName	是	否	String	策略名 示例值： policygen-20141112201913
Description	否	否	String	策略描述 示例值： 测试描述信息
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}
CreateMode	否	否	Uint64	创建模式，1 dbsql带出来的预设策略, 2 按策略模版创建 角色策略 COS侧后台接口创建, 3 按照策略生成器创建, 4 标签相关 示例值： 1

3. 输出参数

参数名称	类型	描述
PolicyId	Uint64	策略id 示例值： 1000
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
ResourceNotFound.UserNotExist	用户不存在。
FailedOperation.PolicyFull	用户策略数超过上限。
InternalError.SystemError	内部错误。
InvalidParameter.ActionNotExist	action不存在

创建角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（CreateRole）用于创建角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleName	是	否	String	角色名称 示例值： QCS
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}
Description	否	否	String	角色描述 示例值： adesc
ConsoleLogin	否	否	Uint64	是否允许登录 1 为允许 0 为不允许 示例值： 0

参数名称	必选	允许NULL	类型	描述
SessionDuration	否	否	Uint64	申请角色临时密钥的最长有效期限制(范围：0~43200) 示例值： 3600
RoleType	否	否	String	角色类型(system

3. 输出参数

参数名称	类型	描述
RoleId	String	角色ID 示例值： 4611686018427388476
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.RoleFull	角色数量达到上限。
InvalidParameter.RoleNameError	角色名不合法。
InvalidParameter.RoleNameInUse	相同名称的角色已存在。
InvalidParameter.UserNotExist	用户对象不存在。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

删除策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

删除策略

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:48:23。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeletePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	否	否	Array of Uint64	数组，数组成员是策略 id，支持批量删除策略 示例值： [1000,1001]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
ResourceNotFound.NotFound	资源不存在。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。

删除角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（DeleteRole）用于删除指定角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427844696
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCS_RoleName

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.ParamError	非法入参。

获取角色列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（DescribeRoleList）用于获取账号下的角色列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRoleList
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Page	是	否	Uint64	页码，从1开始 示例值： 1
Rp	是	否	Uint64	每页行数，不能大于200 示例值： 5
Service	否	否	String	按角色的服务账号载体过滤 示例值： cvm
Keyword	否	否	Array of String	按角色名或角色描述过滤 示例值： ["rolename","roleid"]

3. 输出参数

参数名称	类型	描述
List	Array of	角色详情列表。

参数名称	类型	描述
	RoleInfo	示例值：查看
TotalNum	Uint64	角色总数 示例值： 14
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

解除绑定多个策略到用户组

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定多个策略到用户组

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:49:40。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachGroupPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
GroupId	是	否	Uint64	用户组ID 示例值： 1000
PolicyId	是	否	Array of Uint64	策略ID list 示例值： [1000,1001]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

解除绑定策略到多个用户组

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定策略到多个用户组

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:50:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachGroupsPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
GroupId	是	否	Array of Uint64	用户组ID list 示例值： [1000,1001]
PolicyId	是	否	Uint64	策略ID 示例值： 1000

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

解除绑定策略到多个用户

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定策略到多个用户

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:51:47。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachUsersPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TargetUin	是	否	Array of Uint64	目标用户ID list 示例值： [1000,1001]
PolicyId	是	否	Uint64	策略ID 示例值： 1000

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
FailedOperation.PolicyFull	用户策略数超过上限。
InternalError.SystemError	内部错误。

查看策略详情

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（GetPolicy）可用于查询查看策略详情。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略Id 示例值： 17698703

3. 输出参数

参数名称	类型	描述
PolicyName	String	策略名 示例值： policygen-20141112201913
Description	String	策略描述 示例值： 测试策略
Type	Uint64	1 表示自定义策略，2 表示预设策略 示例值： 1

参数名称	类型	描述
AddTime	Datetime	创建时间 示例值： 2019-04-29 21:18:28
UpdateTime	Datetime	最近更新时间 示例值： 2019-04-29 21:28:32
PolicyDocument	String	策略文档 示例值： {"version":"2.0","statement":[{"effect":"allow","action":["name\/cos:*"],"resource":["*"]}]}
PresetAlias	String	备注 示例值： 备注
IsServiceLinkedRolePolicy	Uint64	是否服务相关策略 示例值： 1
CreateMode	Uint64	1 dbsql带出来的预设策略, 2 按策略模版创建 角色策略 COS侧后台接口创建, 3 按照策略生成器创建, 4 标签相关 示例值： 1
IsCheck	Uint64	秘钥是否验证 1、已验证 0未验证 示例值： 1
PolicyId	Uint64	策略 ID 示例值： 1000
ServiceType	String	服务名称 示例值： cvm
Attachments	Uint64	关联的用户数 示例值： 1
IsAttached	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1
Deactived	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	Array of String	已下线产品列表 示例值： ["deacproduct"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

获取角色详情

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（GetRole）用于获取指定角色的详细信息。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色 ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427844696
RoleName	否	否	String	角色名，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCSRole

3. 输出参数

参数名称	类型	描述
RoleInfo	RoleInfo	角色详情 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.RoleNotExist	角色不存在。

获取服务角色信息

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

获取服务角色信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetServiceRoleInfo
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleName	否	否	String	角色名 示例值： QCS
PolicyName	否	否	Array of String	策略名列表 示例值： ["policygen-1","policygen-2"]

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名 示例值： QCS
ServiceType	String	业务类型 示例值： cvm

参数名称	类型	描述
ServiceTypeEn	String	英文业务类型 示例值： cvm
RoleDesc	String	角色描述 示例值： QCS
RoleDescEn	String	角色英文描述 示例值： QCS
PolicyName	String	预设策略名 示例值： policygen-20141112201913
Remark	String	描述 示例值： remark
EnRemark	String	英文描述 示例值： ""
PolicyList	Array of RolePolicyList	策略列表 示例值：查看
RoleDescI18n	String	支持国际化的角色描述 示例值： ""
Id	String	角色Id 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

查询用户组关联的策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

查询用户组关联的策略列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListAttachedGroupPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TargetGroupId	是	否	Uint64	用户组ID 示例值： 3349
Page	是	否	Uint64	页码，默认值是 1，从 1 开始 示例值： 1
Rp	是	否	Uint64	每页大小，默认值是 20 示例值： 10

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	策略总数 示例值： 1

参数名称	类型	描述
List	Array of AttachPolicyInfo	策略列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

获取角色绑定的策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListAttachedRolePolicies）用于获取角色绑定的策略列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListAttachedRolePolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色 ID。用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427397905
RoleName	否	否	String	角色名。用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： role_data
PolicyType	否	否	String	按策略类型过滤，User表示仅查询自定义策略，QCS表示仅查询预设策略 示例值： 1
Page	是	否	Uint64	页码，从 1 开始 示例值： 1
Rp	是	否	Uint64	每页行数，不能大于200 示例值： 10

3. 输出参数

参数名称	类型	描述
List	Array of AttachedPolicyOfRole	角色关联的策略列表 示例值：查看
TotalNum	Uint64	角色关联的策略总数 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

查询策略关联的实体列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListEntitiesForPolicy）可用于查询策略关联的实体列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListEntitiesForPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略 id 示例值： 524497
Page	否	否	Uint64	页码，默认值是 1，从 1 开始 示例值： 1
Rp	否	否	Uint64	每页大小，默认值是 20 示例值： 10
EntityFilter	否	否	String	可取值 'All'、'User'、'Group' 和 'Role'，'All' 表示获取所有实体类型，'User' 表示只获取子账号，'Group' 表示只获取用户组，'Role' 表示只获取角色，默认取 'All' 示例值： All

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	实体总数 示例值： 1
List	Array of AttachEntityOfPolicy	实体列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

查询策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListPolicies）可用于查询策略列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-09 17:18:22。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Rp	否	否	Uint64	每页数量，默认值是 20，必须大于 0 且小于或等于 200 示例值： 1
Page	否	否	Uint64	页码，默认值是 1，从 1开始，不能大于 200 示例值： 10
Scope	否	否	String	可取值 'All'、'QCS' 和 'Local'，'All' 获取所有策略，'QCS' 只获取预设策略，'Local' 只获取自定义策略，默认取 'All' 示例值： All
Keyword	否	否	String	按策略名匹配 示例值： name
TargetUin	否	否	Uint64	按Uin匹配 示例值： 10093
TargetGroupId	否	否	Uint64	按组Id匹配 示例值： 10093

参数名称	必选	允许NULL	类型	描述
TargetRoleId	否	否	Uint64	按角色Id匹配 示例值： 10093
ServiceType	否	否	String	按产品Id匹配，如cvm 示例值： cvm
FlagUin	否	否	Uint64	按Uin标记关联 示例值： 10093
FlagGroupId	否	否	Uint64	按GroupId标记关联 示例值： 10093
FlagRoleId	否	否	Uint64	按角色Id标记关联 示例值： 10093
Order	否	否	String	策略排序参数 示例值： desc
ProjectVisible	否	否	Uint64	项目可见性 示例值： 1
Client	否	否	String	客户端 示例值： console

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	策略总数 示例值： 239
List	Array of StrategyInfo	策略数组，数组每个成员包括 policyId、policyName、addTime、type、description、 createMode 字段。其中： policyId：策略 id policyName：策略名 addTime：策略创建时间 type：1 表示自定义策略，2 表示预设策略 description：策略描述 createMode：1 表示按业务权限创建的策略，其他值表示可以查看策略语法和通过策略语法更新策略 Attachments: 关联的用户数 ServiceType: 策略关联的产品 IsAttached: 当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略

参数名称	类型	描述
		示例值：查看
ServiceTypeList	Array of String	服务列表 示例值： ["cvm"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.UinError	Uin字段不合法。
InternalError.SystemError	内部错误。
InvalidParameter.ScopeError	Scope字段不合法。
InvalidParameter.KeywordError	Keyword字段不合法。
InvalidParameter.GroupIdError	GroupId字段不合法。
InvalidParameter.ServiceTypeError	ServiceType字段不合法。

修改角色信任策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（UpdateAssumeRolePolicy）用于修改角色信任策略的策略文档。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateAssumeRolePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427731422
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCS_RoleName
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InternalError.SystemError	内部错误。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.ParamError	非法入参。

更新策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（UpdatePolicy ）可用于更新策略。
如果已存在策略版本，本接口会直接更新策略的默认版本，不会创建新版本，如果不存在任何策略版本，则直接创建一个默认版本。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdatePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略ID 示例值： 17698703
PolicyName	否	否	String	策略名 示例值： QCS_PolicyName
Description	否	否	String	策略描述 示例值： Policy_Description
PolicyDocument	否	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyNameError	PolicyName字段不合法。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyDocumentError	PolicyDocument字段不合法。
InvalidParameter.ActionError	策略文档的Action字段不合法。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
InvalidParameter.NotSupportProduct	CAM不支持策略文档中所指定的资源类型。
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument字段超过长度限制。
ResourceNotFound.UserNotExist	用户不存在。
InternalError.SystemError	内部错误。
InvalidParameter.StatementError	策略文档的Statement字段不合法。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InvalidParameter.EffectError	策略文档的Effect字段不合法。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.ResourceError	策略文档的Resource字段不合法。
InvalidParameter.UserNotExist	用户对象不存在。
ResourceNotFound.GroupNotExist	用户组不存在。
ResourceNotFound.NotFound	资源不存在。

错误码	描述
FailedOperation.PolicyNameInUse	PolicyName字段指定的策略名已存在。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
UnauthorizedOperation	未授权操作
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。

用户相关接口
获取CAM密码规则

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口{GetPasswordRules}用于获取用户的密码设置规则

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-23 16:09:45。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetPasswordRules
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
Rules	PasswordRules	密码规则列表 示例值：查看
UpdateTime	String	更新时间 示例值： 2019-04-29 21:18:28
Modifier	String	更新用户 示例值： 130000000001

参数名称	类型	描述
BlackList	String	黑名单字符串列表json string 示例值： aaa
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

子账户所属用户组列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

查询子账户所属用户组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-23 15:09:13。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetSubsGroup
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Uid	是	否	Uint64	接收者用户id 示例值： 1258042
Rp	是	否	Int64	单页数量 示例值： 20
Page	是	否	Int64	分页数 示例值： 1

3. 输出参数

参数名称	类型	描述
TotalNum	String	总体数量 示例值： 1

参数名称	类型	描述
GroupInfo	Array of GroupInfo	用户信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.SystemError	内部错误。

根据SecretId查询Uin

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

根据SecretId查询Uin

默认接口请求频率限制：20次/秒。

接口更新时间：2025-03-17 18:04:03。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetUinBySecretId
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ApiSecretId	是	否	String	密钥ID 示例值： AK****

3. 输出参数

参数名称	类型	描述
Uin	Uint64	用户ID 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。

更新CAM密码规则

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

更新用户密码设置规则

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:06:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdatePasswordRules
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Rules	是	否	PasswordRules	密码设置规则 示例值：查看
BlackList	否	否	String	黑名单字符串列表json string 示例值： aaa

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

身份提供商接口
新增oauth配置

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

新增oauth配置

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-12 19:35:47。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateOauthProvider
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Name	是	否	String	身份提供商（企业）名称。示例值: oauthserver 示例值： name
Desc	否	否	String	备注。示例值: mark 示例值： desc
ClientId	是	否	String	注册应用的id。示例值: c1aa4fbce389206a7061 示例值： c1aa4fbce389206a7061
ClientSecret	是	否	String	注册应用的密钥。示例值: 837c669d84f2fe64b11a8853d521a17970df55ca 示例值：944218ad060c8f2483d4d5bbfa64b6256b78aa79
AuthorizeUri	是	否	String	oauth验证授权信息url。示例值: http://test.com/oauth/authorize 示例值： https://github.com/login/oauth/authorize

参数名称	必选	允许NULL	类型	描述
AccessTokenUri	是	否	String	获取access_token url。示例值: http://test.com/oauth/token 示例值： https://github.com/login/oauth/access_token
GetUserInfoUri	是	否	String	获取用户信息url。示例值: http://test.com/oauth/userinfo 示例值： https://api.github.com/user
UserNameField	是	否	String	登录账号对应字段名称。示例值: username 示例值： user
NickNameField	否	否	String	昵称对应字段名称。示例值: nickname 示例值： nick
PhoneNumField	是	否	String	手机号对应字段名称。示例值: phone 示例值： phone
EmailField	是	否	String	邮箱对应字段名称。示例值: email 示例值： email
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户。传0代表不同步, 传1代表同步。示例值: 0 示例值： 0

3. 输出参数

参数名称	类型	描述
Id	Int64	id 示例值： 1
Name	String	名称 示例值： name
SAMLProviderArn	String	SAMLProviderArn 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.IdentityExist	身份认证失败。

获取用户oauth标识

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

获取用户第三方开放平台的access token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetUserAccessToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserAuthCode	是	否	String	auth code授权码 示例值： ""
OpenAccessToken	否	否	String	第三方access token，复杂授权使用。 示例值： d1e459383977a892fa7b6c549d2f76f8

3. 输出参数

参数名称	类型	描述
AppId	String	app id 示例值： 1255000002
UserOpenId	String	第三方openId 示例值： e5fcfd7cf67d62c85d38b10171641c57

参数名称	类型	描述
UserUnionId	String	第三方unionId 示例值： a64e5abe67a8712c348c547a2760f9ce
UserAccessToken	String	第三方access token 示例值： 905dd67ae2a97e9870364a87d7872da2
ExpiresAt	Int64	过期时间 示例值： 1730433390
UserRefreshToken	String	refresh token 示例值： 905dd67ae2a97e9870364a87d7872da2
Scope	String	授权范围 示例值： login
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。
FailedOperation.AuthCodeError	授权码异常。

刷新用户userAccessToken

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

刷新用户第三方access_token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RefreshUserToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserRefreshToken	是	否	String	用户刷新token 示例值： d1e459383977a892fa7b6c549d2f76f8
UserOpenId	是	否	String	用户openId 示例值： e5fcfd7cf67d62c85d38b10171641c57

3. 输出参数

参数名称	类型	描述
UserAccessToken	String	第三方access_token 示例值： d1e459383977a892fa7b6c549d2f76f8
ExpiresAt	Int64	过期时间 示例值： 1730433390

参数名称	类型	描述
AppId	String	appId 示例值： 1255000002
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.RefreshTokenError	刷新用户token异常。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

更新Oauth配置信息

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

更新Oauth配置信息

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-16 17:05:50。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateOauthProvider
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Desc	否	否	String	备注 示例值： desc
Name	是	否	String	身份提供商（企业）名称 示例值： name
Id	是	否	Int64	id 示例值： 1
OwnerUin	是	否	Int64	OwnerUin 示例值： 110000000001
ClientId	是	否	String	注册应用的id 示例值： c1aa4fbce389206a7061
ClientSecret	是	否	String	注册应用的密钥 示例值： 944218ad060c8f2483d4d5bbfa64b6256b78aa79

参数名称	必选	允许NULL	类型	描述
AuthorizeUri	是	否	String	oauth验证授权信息url 示例值： https://github.com/login/oauth/authorize
AccessTokenUri	是	否	String	获取access_token url 示例值： https://github.com/login/oauth/access_token
GetUserInfoUri	是	否	String	获取用户信息url 示例值： https://api.github.com/user
UserNameField	是	否	String	登录账号对应字段名称 示例值： user
NickNameField	否	否	String	昵称对应字段名称 示例值： nick
PhoneNumField	是	否	String	手机号对应字段名称 示例值： phone
EmailField	是	否	String	邮箱对应字段名称 示例值： email
IsSyncIdpUser	是	否	Int64	是否同步 idp 用户数据 示例值： 0

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

验证用户userAccessToken

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

验证用户第三方开放平台access_token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：VerifyUserAccessToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserAccessToken	是	否	String	access token 示例值： d1e459383977a892fa7b6c549d2f76f8
UserOpenId	是	否	String	open id 示例值： e5fcfd7cf67d62c85d38b10171641c57

3. 输出参数

参数名称	类型	描述
UserOpenId	String	第三方平台openId 示例值： e5fcfd7cf67d62c85d38b10171641c57
UserUnionId	String	第三方平台unionId 示例值： d1e459383977a892fa7b6c549d2f76f8
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.UserrAccessTokenError	用户接入token异常。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

数据结构

GroupMeta

用户组元信息

被如下接口引用：GetUserGroupList、ListAllUserGroup

名称	必选	允许NULL	类型	描述
GroupId	否	是	Int64	用户组id 示例值： 100
GroupName	否	是	String	用户组名称 示例值： group1

ApiKey

API密钥数据列表

被如下接口引用：CreateApiKey、CreateCollApiKey、QueryApiKey、QueryCollApiKey、QueryKeyBySecretId

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AKID***wdew
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 1615990212
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 2
SecretKey	否	否	String	密钥Key 示例值： ACCF***wdew
Source	否	否	Uint64	来源 示例值： 0
Remark	否	否	String	备注 示例值： remark

ListGroupUserInfo

ListGroupUserInfo

被如下接口引用：DescribeGroups、ListUsersForGroup

名称	必选	允许NULL	类型	描述
CountryCode	否	否	String	CountryCode 示例值： 86
CreateTime	否	否	String	CreateTime 示例值： 2025-01-17 14:21:07
Email	否	否	String	Email 示例值： mail@mail.com
EmailFlag	否	是	Int64	EmailFlag 示例值： 0
IsReceiverOwner	否	否	Int64	IsReceiverOwner 示例值： 0
JoinTime	否	否	String	JoinTime 示例值： 2025-01-17 14:21:07
Name	否	否	String	Name 示例值： Name
NickName	否	否	String	NickName 示例值： NickName
PhoneFlag	否	否	Int64	PhoneFlag 示例值： 0
PhoneNum	否	否	String	PhoneNum 示例值： 11111111111
SystemType	否	否	Int64	SystemType 示例值： 0
Uid	否	否	Uint64	Uid 示例值： 1258683
Uin	否	否	Uint64	Uin 示例值： 130000000640
UserType	否	否	Int64	UserType 示例值： 3

名称	必选	允许NULL	类型	描述
QywxUserId	否	否	String	QywxUserId 示例值： ""

AttachPolicyInfo

关联策略信息

被如下接口引用：ListAttachedGroupPolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略id 示例值： 1000
PolicyName	是	是	String	策略名称 示例值： policygen-20141112201913
AddTime	是	是	Datetime	创建时间 示例值： 2014-08-03 12:00:00
CreateMode	是	是	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1

ExtAttr

用户属性集合

被如下接口引用：DescribeSubReceiver

名称	必选	允许NULL	类型	描述
NeedResetToken	是	否	Int64	需要重置mfa的token 示例值： 0
NeedResetStoken	是	否	Int64	需要重置mfa的stoken 示例值： 0

AccountDetail

账号详情

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
ActionFlag	否	否	ActionLoginFlag	敏感操作标识 示例值：查看
ConsoleLogin	否	否	String	是否允许控制台登录, 传0不可登陆控制台, 传1可以登陆控制台。示例值: 1 示例值： 1
LoginFlag	否	否	ActionLoginFlag	登录保护 示例值：查看
NeedResetPassword	否	否	String	是否需要重置密码, 传0不需要重置密码, 传1需要重置密码。示例值: 0 示例值： 0
Password	否	否	String	用户密码。示例值: password 示例值： password
UseApi	否	否	String	使用Api, 传0不使用Api, 传1使用Api。 示例值: 1 示例值： 1
TokenType	否	否	Int64	分配设备类型,传0不分配设备,传2分配设备。示例值: 0 示例值： 0

StrategyInfo

策略信息

被如下接口引用：ListPolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略ID。 示例值： 1
PolicyName	是	否	String	策略名称。 示例值： AdministratorAccess
AddTime	是	是	Datetime	策略创建时间。 示例值： 2016-06-02 19:40:09

名称	必选	允许NULL	类型	描述
Type	是	否	Uint64	策略类型。1 表示自定义策略，2 表示预设策略。 示例值： 1
Description	是	是	String	策略描述。 示例值： 该策略允许您管理账户内所有用户及其权限、财务相关的信息、云服务资产。
CreateMode	是	否	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1
Attachments	否	否	Uint64	关联的用户数 示例值： 1
ServiceType	是	是	String	策略关联的产品 示例值： cam
IsAttached	否	是	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1
Deactived	否	是	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	否	是	Array of String	已下线产品列表 示例值： ["deacproduct"]
IsCheck	是	是	Uint64	是否进行安全性校验，传1代表进行安全校验，传0代表不校验 示例值： 1
PolicyDocument	否	否	String	策略语法 示例值： {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
UpdateTime	否	否	String	修改时间 示例值： 2016-06-02 19:40:09

SubAccountFilter

带过滤条件的子帐号信息

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	子用户Uid 示例值： 1258042
Uin	是	是	Uint64	用户Uin 示例值： 130000000001
Name	是	是	String	用户Name 示例值： username
Remark	是	是	String	备注 示例值： remark
CanLogin	是	是	Uint64	是都允许登录 示例值： 1
PhoneNum	是	是	String	电话号码 示例值： 11111111111
CountryCode	是	是	String	区号 示例值： 86
PhoneFlag	是	是	Int64	电话号码是否验证 示例值： 1
Email	是	是	String	邮箱 示例值： mail@mail.com
EmailFlag	是	是	Int64	邮箱是否验证 示例值： 1
UserType	是	是	Int64	用户类型 示例值： 0
CreateTime	是	是	String	创建时间 示例值： 2019-04-29 21:18:28

名称	必选	允许NULL	类型	描述
IsReceiverOwner	是	是	Int64	是否消息接收人 示例值： 1
SystemType	是	是	String	类型 示例值： subaccount
NeedResetPassword	是	是	Int64	是否需要重置密码 示例值： 0
ConsoleLogin	是	是	Int64	是否允许控制台登录 示例值： 1
WxzsStatus	是	是	Int64	微信公众号关注状态 示例值： 0
PermType	是	是	Array of String	权限类型 示例值： ["0"]
NickName	是	是	String	昵称 示例值： nick
QywxUserId	否	否	String	企业微信用户id 示例值： qywxuserid
UserAttributeAndValues	否	是	Array of AccountAttributeAndValue	扩展属性 示例值：查看
Status	否	否	Int64	状态 示例值： 0
LoginStatus	否	否	Int64	登陆状态 示例值： 0
VerifyPhone	否	否	String	VerifyPhone 示例值： 11111111111
VerifyEmail	否	否	String	VerifyEmail 示例值： mail@mail.com
VerifyCountryCode	否	否	String	VerifyCountryCode 示例值： 86

UserData

用户信息数据

被如下接口引用：GetAllSubUser

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	子用户id 示例值： 1258042
Uin	是	否	Uint64	账号唯一序列号 示例值： 110000000001
Name	是	否	String	用户名称 示例值： name
PhoneNum	是	否	String	电话号码 示例值： 11111111111
CountryCode	是	否	String	区号 示例值： 86
PhoneFlag	是	否	Int64	电话认证标志 示例值： 1
Email	是	否	String	邮箱地址 示例值： mail@mail.com
EmailFlag	是	否	Int64	邮箱是否认证 示例值： 1
UserType	是	否	Int64	用户类型 示例值： 0
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
WechatFlag	是	否	Int64	微信标识 示例值： 0
SystemType	是	否	String	账号系统类型 示例值： Subaccount
IsReceiverOwner	是	否	Int64	是否为主账号 示例值： 1
PermType	否	否	Array of String	PermType 示例值： ["0"]

AttachedStrategyInfo

策略信息

被如下接口引用：DescribeAttachedEntityPolicies

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID。 示例值： 1000
PolicyName	否	否	String	策略名称。 示例值： policygen-20141112201913
AddTime	否	是	Datetime	策略创建时间。 示例值： 2014-08-03 12:00:00
CreateMode	否	是	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1
Description	否	是	String	策略描述。 示例值： adesc

RolePolicyList

角色策略列表

被如下接口引用：GetServiceRoleInfo

名称	必选	允许NULL	类型	描述
IsHidden	否	否	Uint64	状态 示例值： 1
PolicyId	否	否	String	策略Id 示例值： 100
PolicyName	否	否	String	策略名 示例值： policygen-20141112201913

SubAccountInfo

子账户用户信息

被如下接口引用：UpdateSubAccount

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	能否登陆，0-否，1-可 示例值： 1
ConsoleLogin	否	否	String	是否是控制台登陆，1-是 示例值： 1
CountryCode	否	否	String	国家编码 示例值： 86
Name	否	否	String	用户名 示例值： 名称
NeedResetPassword	否	否	String	是否需要重置密码，1-是 示例值： 0
PhoneNum	否	否	String	手机号 示例值： 11111111111
Remark	否	否	String	备注 示例值： 备注
SystemType	否	否	String	账户类型 示例值： subaccount
Uid	否	否	String	接收者用户ID 示例值： 1280504
Uin	否	否	String	账户唯一id 示例值： 110000000001
Password	否	否	String	密码 示例值： password
WxzsStatus	否	否	Int64	微信消息状态 示例值： 0
UserType	否	否	Int64	用户类型 示例值： 0
Email	否	否	String	联系邮箱 示例值： mail@mail.com
Account	否	否	String	用户名 示例值： username
Lang	否	否	String	语言 示例值： en-US

名称	必选	允许NULL	类型	描述
NickName	否	否	String	昵称 示例值： 昵称

UserLists

GetUserListByUinList

被如下接口引用：GetUserListByUinList

名称	必选	允许NULL	类型	描述
CountryCode	否	否	String	CountryCode 示例值： 86
IsAuthed	否	否	Int64	IsAuthed 示例值： 0
UserCamUid	否	否	Int64	UserCamUid 示例值： 1280504
UserCellphone	否	否	String	UserCellphone 示例值： 11111111111
UserEmail	否	否	String	UserEmail 示例值： mail@mail.com
UserId	否	否	String	UserId 示例值： 1280504
UserUin	否	否	Uint64	UserUin 示例值： 110000000001
UserName	否	否	String	UserName 示例值： username
UserIsLocked	否	否	Int64	UserIsLocked 示例值： 1

OwnerInfo

主账号信息

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Uin	是	是	Uint64	主帐号Uin 示例值： 130000000001
UserName	是	是	String	用户名 示例值： username
CheckStatus	是	否	Uint64	校验状态 示例值： 0

AttachedStrategyInfoPack

策略信息包

被如下接口引用：DescribeAttachedEntityPolicies

名称	必选	允许NULL	类型	描述
List	否	否	Array of AttachedStrategyInfo	策略数组，数组每个成员包括 policyId、policyName、addTime、type、description、 createMode 字段。其中： 示例值：查看
TotalNum	否	否	Uint64	策略数 示例值： 100
Id	否	否	String	入参Type=1时表示uin，2时表示groupId 示例值： 100

GroupUidUinInfo

用户组和用户信息

被如下接口引用：DeleteSubAccount

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	子用户Uid 示例值： 1258042
Uin	是	否	Uint64	子用户Uin 示例值： 130000000001

名称	必选	允许NULL	类型	描述
GroupId	是	否	Int64	用户组ID 如果没有任何组传递-1,传入指定组id表示将用户从组删除 示例值： 15094

ServiceApiListInfo

服务的API信息

被如下接口引用：GetServiceApiList

名称	必选	允许NULL	类型	描述
Name	是	否	String	API名称 示例值： CreateInstance
IsNeedObject	是	是	String	是否需要关联对象 示例值： ["1"]
Desc	是	是	String	描述 示例值： 创建实例
ReadWriteDetail	是	是	Uint64	接口类别：0.读取，1.写入，2.标记，3.列表 示例值： 1
InterfaceLevel	是	是	Uint64	授权粒度：0.接口级，1.资源级 示例值： 1
ResourceExample	是	是	String	资源六段式范例 示例值： [qcs::cvm:region:uin/110000001207:volume/*]

AttachedPolicyOfRole

角色关联的策略信息

被如下接口引用：ListAttachedRolePolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略ID 示例值： 1000

名称	必选	允许NULL	类型	描述
PolicyName	是	否	String	策略名称 示例值： policygen-20141112201913
AddTime	是	否	String	绑定时间 示例值： 2014-08-03 12:00:00
PolicyType	是	是	String	策略类型，User表示自定义策略，QCS表示预设策略 示例值： 按策略类型过滤，User表示仅查询自定义策略，QCS表示仅查询预设策略
CreateMode	是	否	Uint64	策略创建方式，1表示按产品功能或项目权限创建，其他表示按策略语法创建 示例值： 1

ActionLoginFlag

登录操作敏感标识

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
Phone	否	否	String	电话, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Stoken	否	否	String	软Token, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Token	否	否	String	硬Token, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Ukey	否	否	String	ukey, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0

ServiceApiInfo

服务及其API信息

被如下接口引用：GetServiceApiList

名称	必选	允许NULL	类型	描述
Name	是	否	String	服务名称 示例值： 用户与权限
ServiceType	是	否	String	服务ID 示例值： cam
ArnDocument	是	是	String	服务介绍文档链接 示例值： https://www.example.com/document/product/378/8965
ApiList	否	是	Array of ServiceApiListInfo	API信息列表 示例值：查看
ConditionKeyList	否	是	Array of String	条件规则列表 示例值： ["qcs:ip"]

DescribeGroupsInfo

DescribeGroupsInfo

被如下接口引用：DescribeGroups

名称	必选	允许NULL	类型	描述
Channel	否	否	Int64	Channel 示例值： 3
CreateTime	否	否	String	CreateTime 示例值： 2025-02-21 11:30:44
GroupId	否	否	Int64	GroupId 示例值： 100
GroupName	否	否	String	GroupName 示例值： name
GroupNum	否	否	Int64	GroupNum 示例值： 3
GroupType	否	否	Int64	GroupType 示例值： 0
Remark	否	否	String	Remark 示例值： Remark

名称	必选	允许NULL	类型	描述
UserInfo	否	否	Array of ListGroupUserInfo	UserInfo 示例值：查看

UserList

子账号列表

被如下接口引用：ListUsersForPolicy

名称	必选	允许NULL	类型	描述
Name	是	否	String	子账号名称 示例值： admin
SubAccountUin	是	否	String	子账号uin 示例值： 1000

CasProviderItem

cas server 配置信息

被如下接口引用：DescribeCasProvider

名称	必选	允许NULL	类型	描述
Id	是	否	Int64	provider id 示例值： 1
CreateUin	是	否	Uint64	创建账户uin 示例值： 110000000001
OwnerUin	是	否	Uint64	主账户uin 示例值： 110000000001
Name	是	否	String	名称 示例值： name
Desc	是	否	String	描述 示例值： desc
ProviderType	是	否	Int64	provider类型 示例值： 8

名称	必选	允许NULL	类型	描述
Status	是	否	Int64	状态 示例值： 0
ModifyTime	是	否	String	修改时间 示例值： 2019-04-29 21:18:28
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
SAMLMetadata	是	否	String	SAML元数据 示例值： ""
SAMLEntityId	是	否	String	SAML实例id 示例值： ""
SAMLSingleSignOn	是	否	String	SAML登陆跳转 示例值： ""
SAMLSingleLogout	是	否	String	SAML登出跳转 示例值： ""
SAMLKeys	是	否	String	SAML关键字 示例值： ""
Cas	是	否	String	Cas 示例值： ""
CasRoot	是	否	String	cas根地址 示例值： https://cas.yfm13.fsphere.cn/
CasLoginUrl	是	否	String	cas登陆url 示例值： https://cas.yfm13.fsphere.cn/cas/login
CasValidateUrl	是	否	String	cas校验url 示例值： https://cas.yfm13.fsphere.cn/cas/serviceValidate
CasLogoutUrl	是	否	String	cas登出url 示例值： https://cas.yfm13.fsphere.cn/cas/logout
Oauth	是	否	String	oauth配置 示例值： ""

GroupUserInfo

用户组下用户信息

被如下接口引用：GetGroupsSubAccount

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	接收者用户id 示例值： 1258042
Uin	是	是	Uint64	账户唯一id 示例值： 130000000001
Name	是	是	String	用户名 示例值： username
PhoneNum	是	是	String	手机号 示例值： 11111111111
CountryCode	是	是	String	国家编码 示例值： 86
PhoneFlag	是	是	String	手机号标识 示例值： 1
Email	是	是	String	邮箱 示例值： mail@mail.com
EmailFlag	是	是	String	邮箱标识 示例值： 1
UserType	是	是	Int64	用户类型 示例值： 0
CreateTime	是	是	String	创建时间 示例值： 2019-04-29 21:18:28
IsReceiverOwner	是	是	String	是否是主账户 示例值： 1
SystemType	是	是	String	账户类型 示例值： subaccount
NickName	是	是	String	昵称 示例值： nick

UserGroup

用户及用户组信息

被如下接口引用：GetUserGroupList、ListAllUserGroup

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	接收者用户id 示例值： 1258042
Uin	是	是	Uint64	账户唯一id 示例值： 110000000001
Name	是	是	String	用户名 示例值： name
IsReceiverOwner	是	是	Int64	是否是主账户 示例值： 1
Group	是	是	Array of GroupMeta	用户组信息 示例值：查看

AddSubAccountDetail

新增用户详情

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
Name	否	否	String	用户名 示例值： name
Token	否	否	String	token 示例值： ""
Password	否	否	String	密码 示例值： password
SecretId	否	否	String	secretid 示例值： ""
SecretKey	否	否	String	secretKey 示例值： ""
NickName	否	否	String	昵称 示例值： nickname
Uin	否	否	Uint64	uin 示例值： 110000000001

名称	必选	允许NULL	类型	描述
Nickname	否	否	String	Nickname 示例值： nickname

Filter

过滤条件组合

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Keywords	否	否	Array of FilterItem	过滤条件组合 示例值：查看
Operator	是	否	String	操作符 示例值： ""

ListOpenPlatform

第三方平台详细信息

被如下接口引用：ListOpenPlatforms

名称	必选	允许NULL	类型	描述
OpenId	是	否	Int64	openid 示例值： 1
AppId	是	否	String	app id 示例值： 1255000002
OpenName	是	否	String	app name 示例值： ""
OpenLogo	是	否	String	open logo 示例值： ""
OpenHome	是	否	String	第三方平台主页 示例值： ""
OpenType	是	否	Int64	授权类型 示例值： 0

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	申请账号 示例值： 110000000001
Status	是	否	Int64	状态 示例值： 0
Domain	是	否	String	第三方平台域名 示例值： ""
State	是	否	Int64	冻结状态，0-非冻结，1-冻结 示例值： 0
Modifier	是	否	String	修改人 示例值： 110000000001
ModifyTime	是	否	String	更新时间 示例值： 2019-04-29 21:18:28
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
Memo	是	否	String	备注 示例值： ""

ApiKeyDetail

持久密钥详情

被如下接口引用：GetApiKey

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AK**********
SecretKey	否	否	String	密钥Key 示例值： TR**********
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 2014-08-03 12:00:00
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 1

名称	必选	允许NULL	类型	描述
Source	否	否	Uint64	来源，默认0 示例值： 1
Remark	否	是	String	描述 示例值： aremark

ResourceTypeItem

资源类型

被如下接口引用：GetServiceList

名称	必选	允许NULL	类型	描述
ResourceEnName	否	否	String	英文名 示例值： ""
ResourceName	否	否	String	中文名 示例值： ""
ResourceType	否	否	String	资源类型 示例值： ""
ServiceType	否	否	String	服务类型 示例值： ""

GroupInfoWithRemark

GroupInfoWithRemark

被如下接口引用：ListUsersForGroup

名称	必选	允许NULL	类型	描述
GroupId	否	否	Int64	GroupId 示例值： 100
GroupName	否	否	String	GroupName 示例值： name
Remark	否	否	String	Remark 示例值： remark

FilterItem

过滤条件数据结构

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Operator	是	否	String	操作符 示例值： ""
Attr	是	否	String	属性 示例值： ""
Value	是	否	String	匹配值 示例值： ""

StrategyInfoForAction

策略详情

被如下接口引用：ListPoliciesByAction

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID。 示例值： 1
PolicyName	否	否	String	策略名称。 示例值： AdministratorAccess
AddTime	否	否	Datetime	策略创建时间。 示例值： 2016-06-02 19:40:09
Type	否	否	Uint64	策略类型。1 表示自定义策略，2 表示预设策略。 示例值： 1
Description	否	否	String	策略描述。 示例值： 该策略允许您管理账户内所有用户及其权限、财务相关的信息、云服务资产。
CreateMode	否	否	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1

名称	必选	允许NULL	类型	描述
Attachments	否	是	Uint64	关联的用户数 示例值： 1
ServiceType	否	否	String	策略关联的产品 示例值： cam
IsAttached	否	是	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1
Deactived	否	是	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	否	是	Array of String	已下线产品列表 示例值： ["cvm"]
IsCheck	否	是	String	是否进行安全性校验，传1代表进行安全校验，传0代表不校验 示例值： 1
PolicyDocument	否	否	String	策略语法 示例值： {"version":"2.0","statement":[{"action":["cam:*","account:*"],"resource":["*"],"effect":"allow"}]}
UpdateTime	否	否	String	修改时间 示例值： 2016-06-02 19:40:09

PasswordRules

密码规则

被如下接口引用：GetPasswordRules、UpdatePasswordRules

名称	必选	允许NULL	类型	描述
MinimumLength	是	否	Int64	最小密码长度 示例值： 10
MustContain	是	否	String	最少包含 示例值： Aa!
ForcePasswordChange	是	否	Int64	密码有效期 示例值： 0

名称	必选	允许NULL	类型	描述
ReusePasswordLimit	是	否	Int64	密码重复次数 示例值： 2
RetryPasswordLimit	否	否	Int64	登陆最大密码失败次数 示例值： 2
OnlyAdminCanResetPassword	否	否	Int64	是否只有admin可以重置密码 示例值： 0
MustNotContainUsername	否	否	Int64	必须不包含用户名 示例值： 0

AccountAttributeAndValue

扩展属性值

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
AttributeName	否	否	String	属性名称 示例值： ""
Attribute	否	否	String	属性 示例值： ""
AttributeId	否	否	Int64	属性id 示例值： ""
ValueId	否	否	Int64	值id 示例值： ""
Uin	否	否	Int64	uin 示例值： 110000000001
Value	否	否	String	值 示例值： ""

GroupInfo

用户组信息

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
GroupId	否	否	Uint64	组id 示例值： 15094
GroupName	否	否	String	组名称 示例值： groupname
Channel	否	是	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件 示例值： 0
Remark	否	是	String	备注 示例值： remark
CreateTime	否	是	String	创建时间 示例值： 2019-04-29 21:18:28
UserInfo	否	是	Array of SubAccountFilter	用户组成员信息 示例值：查看
GroupType	否	是	Int64	用户组类型，0-自定义，1-预设 示例值： 0
GroupNum	否	否	Int64	组成员数 示例值： 2

GroupMember

用户组成员

被如下接口引用：AddUserToGroup、RemoveUserFromGroup、UpdateGroupMember

名称	必选	允许NULL	类型	描述
Uid	是	否	String	用户id 示例值： 100
GroupId	是	否	String	组id 示例值： 100

OwnerAccountAttribute

主账户属性

被如下接口引用：UpdateOwnerAccount

名称	必选	允许NULL	类型	描述
Remark	否	否	String	属性 示例值： remark

AttachEntityOfPolicy

策略关联的实体信息

被如下接口引用：ListEntitiesForPolicy

名称	必选	允许NULL	类型	描述
Id	是	否	String	实体ID 示例值： 1000001
Name	是	是	String	实体名称 示例值： policygen-20141112201913
Uin	是	是	Uint64	实体Uin 示例值： 3449203261
RelatedType	是	否	Uint64	关联类型。1 用户关联 ； 2 用户组关联 示例值： 1

GroupData

用户组相关信息

被如下接口引用：GetAllSubUser

名称	必选	允许NULL	类型	描述
GroupId	否	否	Int64	用户组id 示例值： 1000
GroupName	是	否	String	用户组名称 示例值： group1
GroupNum	是	否	Int64	用户组成员数量 示例值： 100

名称	必选	允许NULL	类型	描述
Channel	是	否	Int64	创建渠道 示例值： 0
GroupMem	否	否	Array of Uint64	组成员uid 示例值： [1280504,1280505]

UpdateGroupInfo

更新用户组信息

被如下接口引用：UpdateSubAccount

名称	必选	允许NULL	类型	描述
GroupId	是	否	Uint64	用户组id 示例值： 15094
Uid	是	否	Uint64	用户id 示例值： 1258042
GroupName	否	否	String	用户组名称 示例值： name
Channel	否	否	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件 示例值： 0

AttachedUserPolicy

用户关联的策略详情

被如下接口引用：ListAttachedUserAllPolicies

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID 示例值： 1000
PolicyName	否	否	String	策略名 示例值： policygen-20141112201913
Description	否	否	String	策略描述 示例值： adesc

名称	必选	允许NULL	类型	描述
AddTime	否	否	String	创建时间 示例值： 2014-08-03 12:00:00
StrategyType	否	否	Uint64	策略类型(1表示自定义策略，2表示预设策略) 示例值： 1
CreateMode	否	否	Uint64	创建模式(1表示按产品或项目权限创建的策略，其他表示策略语法创建的策略) 示例值： 1
Groups	否	是	Array of AttachedUserPolicyGroupInfo	随组关联信息 示例值：查看

AttributeInfo

属性

被如下接口引用：AddAttributeValues

名称	必选	允许NULL	类型	描述
AttributeName	否	是	String	属性名称 示例值： ""
Attribute	否	是	String	属性 示例值： ""
AttributeValue	否	是	String	属性值 示例值： ""

Receiver

消息接收人信息

被如下接口引用：DescribeSubAccountContacts

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	id 示例值： 1258042

名称	必选	允许NULL	类型	描述
Name	是	否	String	名字 示例值： username
Remark	是	否	String	备注 示例值： remark
PhoneNumber	是	否	String	手机号码 示例值： 11111111111
PhoneFlag	是	否	Int64	手机号码是否验证 示例值： 1
Email	是	否	String	邮箱 示例值： mail@mail.com
EmailFlag	是	否	Int64	邮箱是否验证 示例值： 1
IsReceiverOwner	是	否	Int64	是否主联系人 示例值： 1
WechatFlag	否	否	Int64	是否允许微信接收通知 示例值： 0
Uin	是	否	Uint64	账号uin 示例值： 130000000001

RoleInfo

角色详细信息

被如下接口引用：DescribeRoleList、GetRole

名称	必选	允许NULL	类型	描述
RoleId	是	否	String	角色ID 示例值： 12
RoleName	是	否	String	角色名称 示例值： QCS
PolicyDocument	是	否	String	角色的策略文档 示例值： {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],

名称	必选	允许NULL	类型	描述
				\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
Description	是	否	String	角色描述 示例值： Description
AddTime	是	否	String	角色的创建时间 示例值： 2020-01-01 01:01:01
UpdateTime	是	否	String	角色的最近一次时间 示例值： 2020-01-01 01:01:01
DeletionTaskId	否	是	String	兼容公有云字段，无含义 示例值： 1
ConsoleLogin	是	否	Uint64	角色是否允许登录，传1代表允许登录，传0代表不允许 示例值： 1
RoleType	否	是	String	角色类型，取user、system或service_linked 示例值： user
SessionDuration	否	是	Uint64	有效时间 示例值： 7200

ServiceItem

服务

被如下接口引用：GetServiceList

名称	必选	允许NULL	类型	描述
AddTime	否	否	Datetime	创建时间 示例值： 2014-08-03 12:00:00
ArnDocument	否	否	String	ArnDocument 示例值： https://www.example.com/document/product/378/8222965

名称	必选	允许NULL	类型	描述
ColConf	否	是	String	ColConf 示例值： ColConf
DefAddr	否	是	String	DefAddr 示例值： DefAddr
DefaultStrategyList	否	否	String	默认策略 示例值： DefaultStrategyList
IsAllowDefProj	否	否	String	IsAllowDefProj 示例值： 1
IsDisProject	否	否	String	IsDisProject 示例值： 1
IsDisZone	否	否	String	IsDisZone 示例值： 1
IsSeen	否	否	String	是否可见 示例值： 1
Online	否	否	String	Online 示例值： Online
QueryAddr	否	否	String	QueryAddr 示例值： cvm
QueryInterface	否	否	String	QueryInterface 示例值： cvm
ServiceEnName	否	否	String	服务英文名 示例值： cvm
ServiceName	否	否	String	服务名 示例值： cvm
ServiceType	否	否	String	服务类型 示例值： cvm
SynInterface	否	否	String	SynInterface 示例值： cvm
UpdateTime	否	否	Datetime	变更时间 示例值： 2014-08-03 12:00:00
Weight	否	否	String	Weight 示例值： 1

名称	必选	允许NULL	类型	描述
WhiteKey	否	否	String	WhiteKey 示例值： white1
Writter	否	否	String	创建人 示例值： admin
ResourceTypeList	否	否	Array of ResourceTypeItem	资源类型数组 示例值：查看
Type	否	否	String	类型 示例值： cvm

UserInfo

用户信息

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	子账号类型，传0不可登陆控制台，传1可以登陆控制台。 示例值： 1
CountryCode	否	否	String	区号。 示例值： 86
Detail	否	否	AccountDetail	详情 示例值：查看
Name	否	否	String	名称。 示例值： name
PhoneNum	否	否	String	电话号码。 示例值： 11111111111
SystemType	否	否	String	账号类型。 示例值： Subaccount
Email	否	否	String	安全邮箱。 示例值： mail@mail.com
NickName	否	否	String	昵称。 示例值： nickname

名称	必选	允许NULL	类型	描述
Remark	否	否	String	备注。 示例值： remark
WxzsStatus	否	否	Int64	微信登陆状态。 示例值： 0
ContactMail	否	否	String	联系邮箱。 示例值： mail@mail.com
IsReceiverOwner	否	否	Int64	是否是主账号。传1代表主账号, 传0代表子账号。 示例值： 1
IdentifyType	否	否	Int64	身份类型。 示例值： 0

ServicePermItem

接口

被如下接口引用：GetServicePermList

名称	必选	允许NULL	类型	描述
AddTime	否	否	Datetime	创建时间 示例值： 2014-08-03 12:00:00
ApiAddr	否	否	String	ApiAddr 示例值： ApiAddr
ApiZhName	否	否	String	中文描述 示例值： ADMIN
AuthFunction	否	否	String	鉴权接口 示例值： AuthFunction
CWildcardName	否	否	String	CWildcardName 示例值： CWildcardName
InterfaceEnName	否	否	String	接口名 示例值： CVM
InterfaceLevel	否	否	String	鉴权粒度，0:接口级别、1:资源级别 示例值： 1

名称	必选	允许NULL	类型	描述
IsAuthBusiness	否	否	String	鉴权方式，0:由云API转发鉴权、1:业务自行调用鉴权接口 示例值： 1
IsNeedObject	否	否	String	IsNeedObject 示例值： 1
IsSeen	否	否	Uint64	IsSeen 示例值： 1
IsSeenAtGenerator	否	否	String	策略生成器是否可见 示例值： 1
IsSpResource	否	否	String	IsSpResource 示例值： 1
IsUserSet	否	否	String	IsUserSet 示例值： 1
PermId	否	否	String	Id 示例值： 1000
ReadWriteDetail	否	否	String	接口类别 示例值： 接口类别：0.读取，1.写入，2.标记，3.列表
ResourceType	否	是	String	资源类别 示例值： cvm
UpdateTime	否	否	String	更新时间 示例值： 2014-08-03 12:00:00
Weight	否	否	String	Weight 示例值： 1
Writter	否	否	String	操作者 示例值： admin
ServiceName	否	否	String	服务名 示例值： cvm
ProductShortCode	否	否	String	ProductShortCode 示例值： p_cvm
ProductShortName	否	否	String	ProductShortName 示例值： p_cvm

名称	必选	允许NULL	类型	描述
ServiceType	否	否	String	服务类型 示例值： cvm
Interface	否	否	String	接口名 示例值： CVM
InterfaceName	否	否	String	接口名 示例值： CVM

AttachedUserPolicyGroupInfo

用户关联策略(随组管理)信息

被如下接口引用：ListAttachedUserAllPolicies

名称	必选	允许NULL	类型	描述
GroupId	否	否	Uint64	分组ID 示例值： 10012
GroupName	否	否	String	分组名称 示例值： group_one

错误码

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。 请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。 签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。
InvalidParameter.ResourceError	策略文档的Resource字段不合法。
InvalidParameter.PolicyNameError	PolicyName字段不合法。
ResourceNotFound.URLError	url解析异常。

错误码	说明
InvalidParameter.NotSupportProduct	CAM不支持策略文档中所指定的资源类型。
InvalidParameter.ScopeError	Scope字段不合法。
InvalidParameter.ServiceTypeError	ServiceType字段不合法。
InvalidParameter.RoleNameError	角色名不合法。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument字段超过长度限制。
FailedOperation.CheckPasswordError	检查密码失败
InvalidParameter.PasswordLengthTooShort	密码太短。
FailedOperation.NameAlreadyExist	用户名已存在
InvalidParameter.RoleNameInUse	相同名称的角色已存在。
InvalidParameter.CreateGroupErr	创建用户组失败
InvalidParameter.KeywordError	Keyword字段不合法。
InvalidParameter.UserNotExist	用户对象不存在。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.ParamError	非法入参。
InvalidParameter.GroupIdError	GroupId字段不合法。
FailedOperation.AccountGroupNameNotMatch	配置组名不匹配。
InvalidParameter.PolicyDocumentError	PolicyDocument字段不合法。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
FailedOperation.AuthCodeError	授权码异常。
ResourceNotFound.IdentityNotExist	身份不存在。
FailedOperation.AccountSettingConfigError	账号配置元数据异常。
FailedOperation.SubAccountHasKey	子账号存在密钥。
FailedOperation.RefreshTokenError	刷新用户token异常。

错误码	说明
ResourceUnavailable.IDPMaxLimit	超过idp最大数量限制。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
InvalidParameter.UinError	Uin字段不合法。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
FailedOperation.Accesskey	操作访问密钥错误
FailedOperation.AddIdentityError	添加认证身份失败。
InvalidParameter.ActionError	策略文档的Action字段不合法。
FailedOperation.UserrAccessTokenError	用户接入token异常。
FailedOperation.PolicyNameInUse	PolicyName字段指定的策略名已存在。
FailedOperation.UnknownAccountSettingKey	未知的账号配置项。
InternalError.SystemError	内部错误。
FailedOperation.SetLoginRuleFail	设置登录策略失败。
InvalidParameter.ActionNotExist	action不存在
InvalidParameter.RoleFull	角色数量达到上限。
ResourceNotFound.NotFound	资源不存在。
FailedOperation.SecretIdExist	SecretId已存在。
FailedOperation.NoneValue	缺少配置。
FailedOperation.InAsyncModifyError	数据修改中
FailedOperation.SkeyExpired	Skey已过期。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
ResourceNotFound.UserNotExist	用户不存在。
ResourceNotFound.RecordNotExist	记录不存在
ResourceNotFound.GroupNotExist	用户组不存在。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。

错误码	说明
FailedOperation.PolicyFull	用户策略数超过上限。
FailedOperation.AccountSettingValueCalculateError	配置值计算异常。
InvalidParameter.EffectError	策略文档的Effect字段不合法。
FailedOperation.UnknownAccountSettingGroup	未知的账号配置组。
InvalidParameter.StatementError	策略文档的Statement字段不合法。
UnauthorizedOperation	未授权操作
FailedOperation.IdentityExist	身份认证失败。