云顾问（CloudAdvisor）

产品文档

目录

产品简介
产品概述

云顾问（CloudAdvisor）是一款开箱即用的云资源风险评估产品。基于 CAM 服务角色授权 机制，一键分析云资源、应用架构、业务性能及安全方面存在的风险，并根据业务实际使用情况，在线提供优化建议，助您提升系统安全性、业务稳定性以及服务可靠性。

功能介绍

云顾问（CloudAdvisor）提供了丰富的评估项目、灵活的评估配置以及系统的优化建议，助您提升业务连续性。

灵活的评估配置

- 支持自定义选择评估项目，您可以屏蔽不关注的评估项目，精简评估报告结果，聚焦核心指标；
- 支持指定可忽略的云资源，减少测试资源对评估结果的干扰，专注核心资源的运行情况。

系统的优化建议

针对每项评估结果，结合 TCloudFinanceZone 多年服务客户沉淀的最佳实践，在线提供有针对性且可操作的优化建议，您可以按照检查建议来优化服务和资源。

丰富的评估项目

从安全性、可靠、成本、服务限制、性能这几个维度覆盖多个云产品，每维度均包含多个风险评估项目。目前支持评估的云产品有：

- Elasticsearch Service
- 云服务器（CVM）
- 云数据库（MongoDB）
- 云数据库（Redis®）
- 云硬盘（CBS）
- 负载均衡（CLB）
- 云防火墙（CFW）
- 消息队列 CKafka 版
- 消息队列 Pulsar 版
- 云数据库（MariaDB）
- 容器服务（TKE）
- 私有网络（VPC）

随着产品的迭代，我们会逐步覆盖更多云产品，并提供更多的风险评估项目。

应用场景

日常运维保障

定期评估云上资源健康状况，对云架构、云资源进行风险评级，在线反馈优化操作建议，高效提升业务连续性。

资源容量规划

定期检查云资源使用水位，评估服务使用情况，按需对云资源进行扩缩容。

架构主动优化

验证架构部署的安全性、容错能力、备份能力，以确保业务持续稳定运行。用户可以按需主动触发巡检。

功能介绍

云顾问的巡检项包含五个维度：安全、可靠、性能、成本、服务限制。

安全

通过建议您启用安全功能以及检查权限，提高系统和业务的安全性。

产品	巡检项	巡检说明
云服务器（CVM）	云服务器（CVM）公网访问不受限制	检查 CVM 公网访问安全策略，若 CVM 配置了公网 IP，且安全组放通了对所有 IP 和 Port 的访问权限，会存在被恶意入侵的风险
云服务器（CVM）	云服务器（CVM）公网高危端口	检查 CVM 公网访问安全策略，若 CVM 配置了公网 IP，且放通了高危端口访问权限，会存在被恶意入侵的风险
Elasticsearch Service	ES 集群公网访问策略	检查 ES 集群的 Elasticsearch 组件公网访问策略，若未配置任何限制，则告警
Elasticsearch Service	ES 集群的 Kibana 组件公网访问策略	检查 ES 集群的 Kibana 组件公网访问策略，若未配置任何限制，则告警
云数据库（Redis®）	云数据库（Redis®）高危命令检查	检查 Redis® 实例禁用命令配置，若高危命令未禁用，容易出现应用阻塞，数据误删等风险
云防火墙（CFW）	云防火墙（CFW）资源防护检查	检查云防火墙防护策略，若资源类型为CVM/NAT/VPN/CLB的实例未开启，则提示风险

可靠

通过多方位监控，维护实例的运行稳定性。

产品	巡检项	巡检说明
云服务器（CVM）	云服务器（CVM）系统盘快照	检查CVM系统盘快照，若未创建快照，服务器或云硬盘出现问题时数据找回非常困难，易造成较大损失
云服务器（CVM）	云服务器（CVM）实例磁盘空间使用率过高	检查CVM实例磁盘使用情况，若使用率过高，则磁盘读写会受到影响
云服务器（CVM）	云服务器（CVM）实例本地盘类型检查	检查CVM实例使用本地盘的情况，若实例为非IO或大数据类型，且使用了本地盘，则磁盘数据无法通过快照备份，存在容灾风险
云服务器（CVM）	云服务器（CVM）带宽利用率过高	检查CVM实例带宽利用率情况，若带宽利用率过高，则网络性能可能会受到影响

云硬盘（CBS）	云硬盘（CBS）存储容量	检查云硬盘（CBS）的存储容量使用情况，若已使用容量占总容量比率过高，会导致云硬盘读写受到影响
云硬盘（CBS）	云硬盘（CBS）未创建快照	检查CBS是否有创建快照或定期快照策略，若都没有，服务器或云硬盘出现问题时数据找回非常困难，易造成较大损失
ElasticsearchService	Elasticsearch集群自动快照备份	检查Elasticsearch集群自动快照备份配置，若未配置，则提示风险
负载均衡（CLB）	负载均衡（CLB）健康检查配置	检查CLB是否配置健康检查，若未配置健康检查，CLB将向所有后端服务器转发流量（包括异常的后端服务器）
负载均衡（CLB）	负载均衡（CLB）转发规则配置	检查CLB监听器配置，若未配置转发规则，则无法正常使用CLB功能，产生额外成本
负载均衡（CLB）	负载均衡（CLB）健康检查存在跳变情况	检查CLB监听器的健康检查是否有跳变情况，即是否存在服务器端口状态异常
负载均衡（CLB）	负载均衡（CLB）实例类型	检查CLB实例类型为传统型还是应用型，应用型功能更加丰富，如每个四层监听器可以配置不同的后端服务、支持七层监听器、支持CLS日志、SNI、绑定弹性网卡等多种特性
负载均衡（CLB）	负载均衡（CLB）及其绑定的CVM跨区	检查CLB及其绑定的CVM实例是否在同一个可用区，如果不是，跨区转发可能影响服务可靠性，如降低部分转发请求的速度
负载均衡（CLB）	负载均衡（CLB）后端服务单点	检查CLB监听器或转发规则绑定的如CVM、EVM等类型的后端服务实例，如果只有一个，存在单点隐患
负载均衡（CLB）	负载均衡（CLB）转发规则绑定CVM多个端口	检查CLB同一转发规则是否绑定同一台CVM的多个端口，如果是的话，随着业务量的增长，进程间的资源争抢会增加排障难度，同时多个端口可能会降低系统对流量波峰的抵御能力
负载均衡（CLB）	负载均衡（CLB）下的CVM跨子网	检查CLB同一监听器或转发规则绑定的多个CVM实例是否跨VPC子网，如果是的话，在异常发生情况不利于快速排障
负载均衡（CLB）	负载均衡（CLB）下的CVM权重	检查CLB同一监听器或转发规则关联的CVM权重，如果出现相同配置不同权重，或相同权重不同配置的情况，则可能在业务高峰时暴露性能短板的风险，影响业务稳定
容器服务（TKE）	容器服务（TKE）集群节点跨可用区	集群节点是否都在单一可用区，单一可用区不可用时影响业务，集群无法调度到其他可用区
云数据库（MongoDB）	云数据库（MongoDB）oplog	检查MongoDBoplog保存时间，若保存时间过短，会导致回档失败或影响问题排查

	保存时间	
云数据库（MongoDB）	云数据库（MongoDB）备份是否成功	检查MongoDB备份是否成功，如果备份任务失败，可能导致无法恢复数据
云数据库（MongoDB）	云数据库（MongoDB）使用基础网络	检查MongoDB是否使用基础网络
云数据库（MariaDB）	云数据库（MariaDB）主从延迟	当主从延迟持续过大时，主从数据一致性将得不到保障，此时如果实例发生了HA主从切换，极端情况下数据可能出现丢失
云数据库（Redis®）	云数据库（Redis®）跨可用区部署	检查Redis®实例是否跨可用区部署，如果实例未跨可用区部署，当实例出现可用区级别的灾难故障时，可能造成实例无法访问风险
云数据库（Redis®）	云数据库（Redis®）使用基础网络	检查Redis®是否使用基础网络
消息队列 Pulsar 版	消息队列 Pulsar 版集群健康状态检查	非健康状态下，集群使用可能面临一定风险
消息队列 Pulsar 版	消息队列 Pulsar 版备份消费者检查	检查是否只有一个消费者，如果采用单个消费者消费，单点挂了会影响消费业务
消息队列 Pulsar 版	消息队列 Pulsar 版死信队列检查	如果没有死信队列，消费者可能无法处理一些特殊情况的消息
消息队列 CKafka 版	消息队列 CKafka 版跨可用区部署	如果没有跨可用区部署，单可用区集群出现严重故障的情况下，可能会导致CKafka集群不可用
私有网络（VPC）	私有网络（VPC）网络属性	检查VPC的网络属性，若网络属性为基础网络类型，则云服务器的内网IP地址都由云统一分配，无法自定义网段划分、IP地址，有一定的使用限制
私有网络（VPC）	私有网络（VPC）子网规划	检查子网网段与VPC网段是否一致，如果完全一致，导致不能规划更多子网使用，不利于跨区拓展等长期规划实施

性能

根据监控实例运行中的资源使用情况和最佳实践，为您提供改善性能的建议。

产品	巡检项	巡检说明
云硬盘（CBS）	云硬盘（CBS）IO高负载	检查云硬盘（CBS）IO负载情况，若IO负载过高，则发出警告

产品	巡检项	巡检说明
云硬盘（CBS）	云硬盘（CBS）IOPS超限	检查CBS的IOPS峰值是否达到该类型CBS的配置上限，若已达到会有受到限流的风险
云硬盘（CBS）	云硬盘（CBS）吞吐量超限	检查CBS的吞吐量峰值是否达到该类型CBS的配置上限，若已达到会有受到限流的风险
云数据库（Redis®）	云数据库（Redis®）CPU使用率	检查Redis®实例CPU使用率，CPU使用率长期过高可能导致请求延迟上升，阻塞等现象
云数据库（MongoDB）	云数据库（MongoDB）Cache脏数据	检查MongoDBCache脏数据情况，若Cache脏数据百分比大于20%，用户线程将参与刷盘，阻塞业务
云数据库（MongoDB）	云数据库（MongoDB）CPU使用率	检查MongoDB实例CPU使用率情况，若使用率过高，可能会出现业务请求延迟增加、等待等风险
云数据库（MariaDB）	云数据库（MariaDB）CPU使用率	当CPU使用率较高时，说明当前实例处理繁忙，容易导致查询变慢、堵塞的问题

成本

根据运行情况，给出性价比更高的配置建议，降低用户成本花费。

产品	巡检项	巡检说明
负载均衡（CLB）	负载均衡（CLB）实例被闲置	检查CLB后端云资源绑定情况，若未绑定云资源（CVM实例、弹性网卡），则会判定为实例被闲置，产生额外成本
云硬盘（CBS）	云硬盘（CBS）未充分利用	检查CBS的挂载状态及IO读写情况，若CBS在近5天一直处于未挂载状态或近7天每天的IOPS不超过1次，则发出警报。长期闲置的云硬盘会带来不必要的开销
云数据库（MariaDB）	云数据库（MariaDB）利用率不足	检查实例是否闲置，如果业务生命周期已经稳定，长时间的闲置资源对业务成本会造成较多浪费
云数据库（MongoDB）	云数据库（MongoDB）利用率不足	检查MongoDB实例是否闲置，如果业务生命周期已经稳定，长时间的闲置资源对业务成本会造成较多浪费
云数据库（Redis®）	云数据库（Redis®）利用率不足	检查实例是否闲置，实例长时间业务请求次数小于100次，如果业务生命周期已经稳定，长时间的闲置资源对业务成本会造成较多浪费
负载均衡（CLB）	负载均衡（CLB）低利用率	检查CLB低利用率情况，如果连接数小于配额的10%，可能存在冗余成本

服务限制

通过监控可提供的服务资源的最大数量。提醒您按照建议删除资源或请求增加配额。

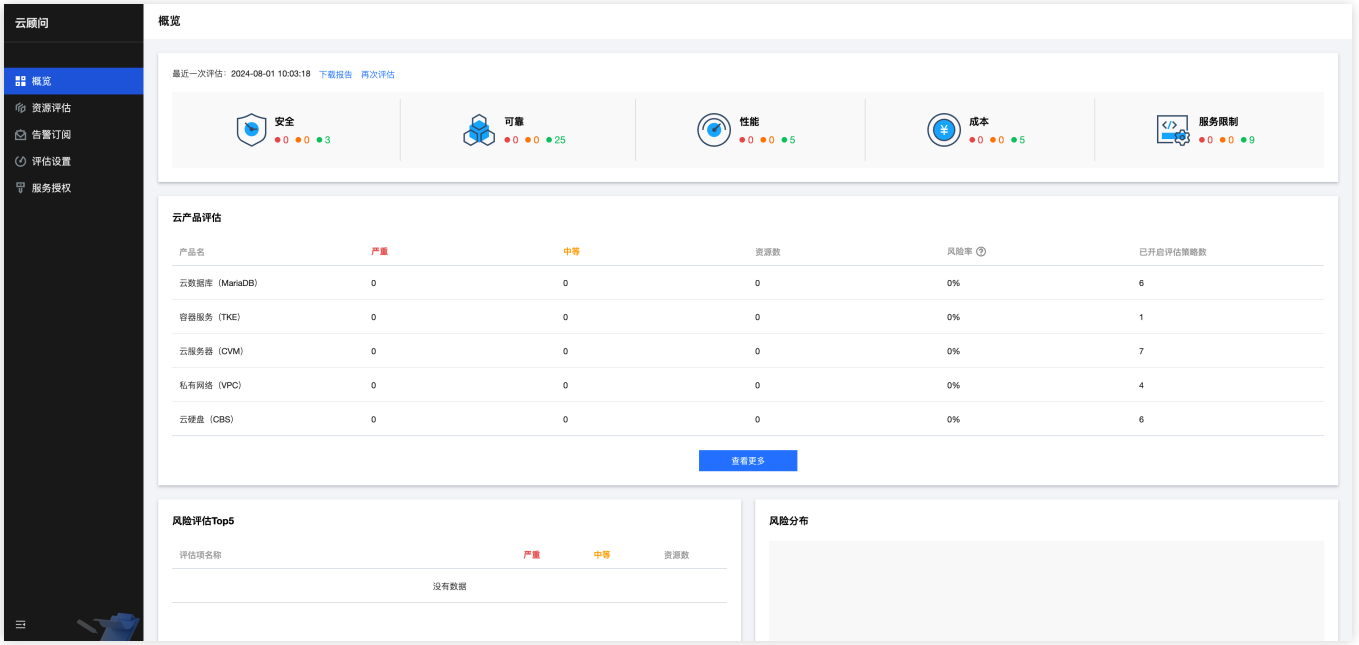
产品	巡检项	巡检说明
云数据库（MongoDB）	云数据库（MongoDB）存储容量	检查MongoDB存储容量的使用情况，当容量使用率达到100%时，将会导致写入失败
云数据库（Redis®）	云数据库（Redis®）内存接近4T上限	检查Redis®实例内存是否接近4T上限
云数据库（Redis®）	云数据库（Redis®）副本数达到上限5个	检查Redis®实例副本数是否达到上限5个
云数据库（MariaDB）	云数据库（MariaDB）活跃连接数	当活跃连接数过多时，表明实例目前已经处于较高的压力状态，容易出现请求阻塞的情况
云数据库（MariaDB）	云数据库（MariaDB）连接使用率	当连接数使用率100%的时候，新增请求将无法建立连接，访问失败
云数据库（MariaDB）	云数据库（MariaDB）数据盘使用率	当磁盘使用率达到100%时，写入将会失败
云防火墙（CFW）	云防火墙（CFW）规则配额检查	检查云防火墙规则列表配额，若配额不足，则提示风险
私有网络（VPC）	私有网络（VPC）路由表使用数量	检查VPC路由表的数量，若接近或超过上限值，容易导致无法及时建立新的路由表
私有网络（VPC）	私有网络（VPC）路由表策略用量	检查VPC路由表策略数，若接近或超过上限值，容易导致无法及时建立新的路由策略

快速入门

本文为您介绍如何在控制台界面快速使用云顾问（CloudAdvisor）。

操作步骤

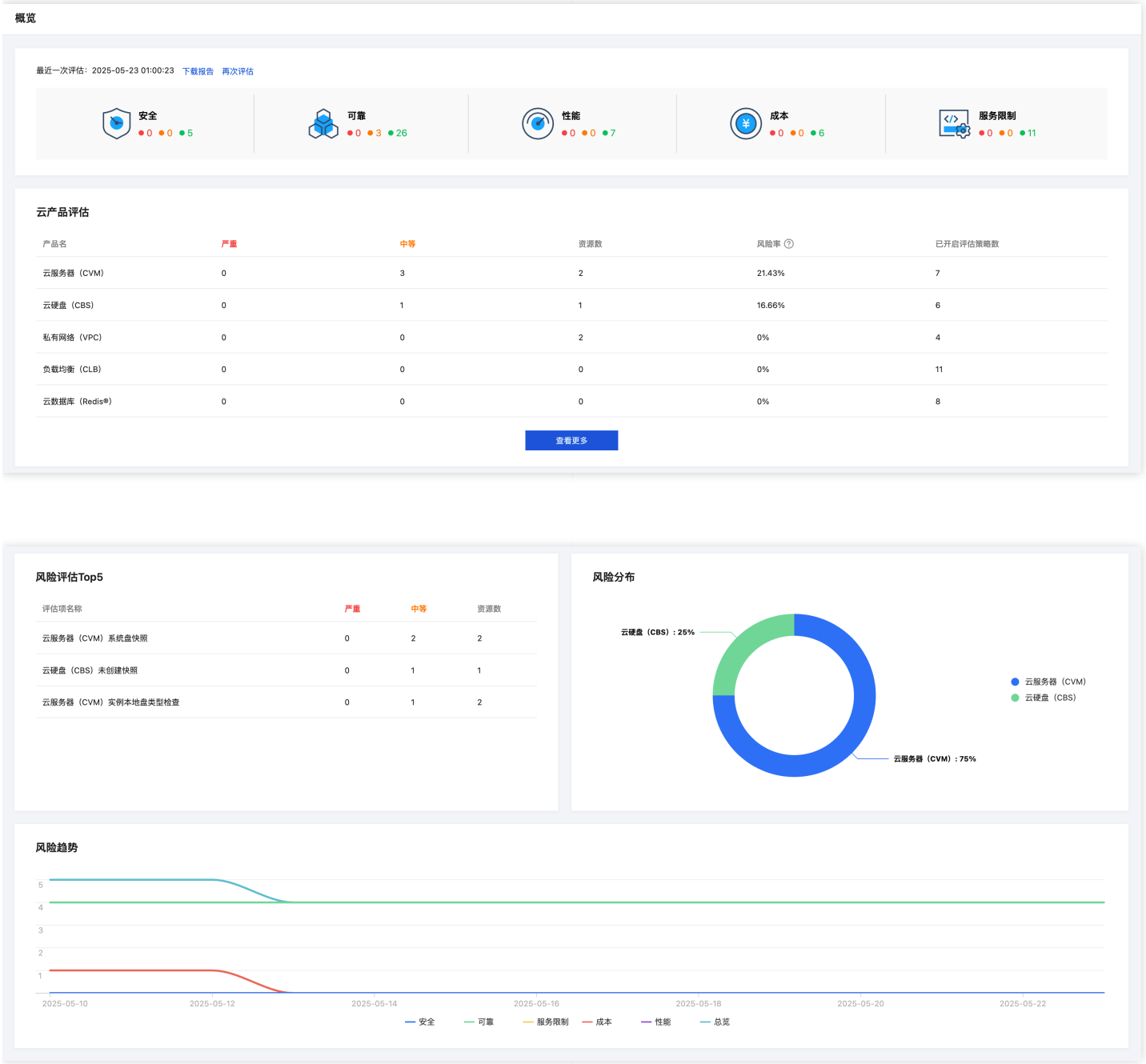
1. 登录云顾问控制台，点击云顾问默认进入概览页面。



2. 授权云顾问访问相关云产品配置权限。
   首次使用时，在服务授权页面中启用服务授权，完成授权操作。
3. 授权成功后，点击资源评估页面，单击开始评估，开始云资源评估操作。
4. 查看评估结果。
   一次评估操作预计需要花费数十秒时间，页面会动态刷新评估结果，直到评估完成。
   - 总览：统计并展现所有评估项目结果。
   - 安全：按照「安全」分类统计并展现评估项目结果。
   - 可靠：按照「可靠」分类统计并展现评估项目结果。
   - 性能：按照「性能」分类统计并展现评估项目结果。
   - 成本：按照「成本」分类统计并展现评估项目结果。
   - 服务限制：按照「服务限制」分类统计并展现评估项目结果。

操作指南
概览

云顾问概览包含风险概览、云产品评估、风险评估Top5、风险分布、风险趋势等模块，您可以通过概览页快速获取风险情况。



- 风险概览
  按照五种类别展示整体的风险情况。
- 云产品评估
  展示最近一次评估中云产品的巡检结果统计，帮您了解各云产品的风险情况、资源数、风险率、已开启评估策

略数等。

- 风险评估Top5

  展示最近一次评估中Top5的风险评估项和涉及的资源数。
- 风险分布

  展示最近一次评估中风险在不同类别的分布情况，以及不同云产品的风险情况。
- 风险趋势

  展示最近14天的整体风险趋势图。

资源评估

操作场景

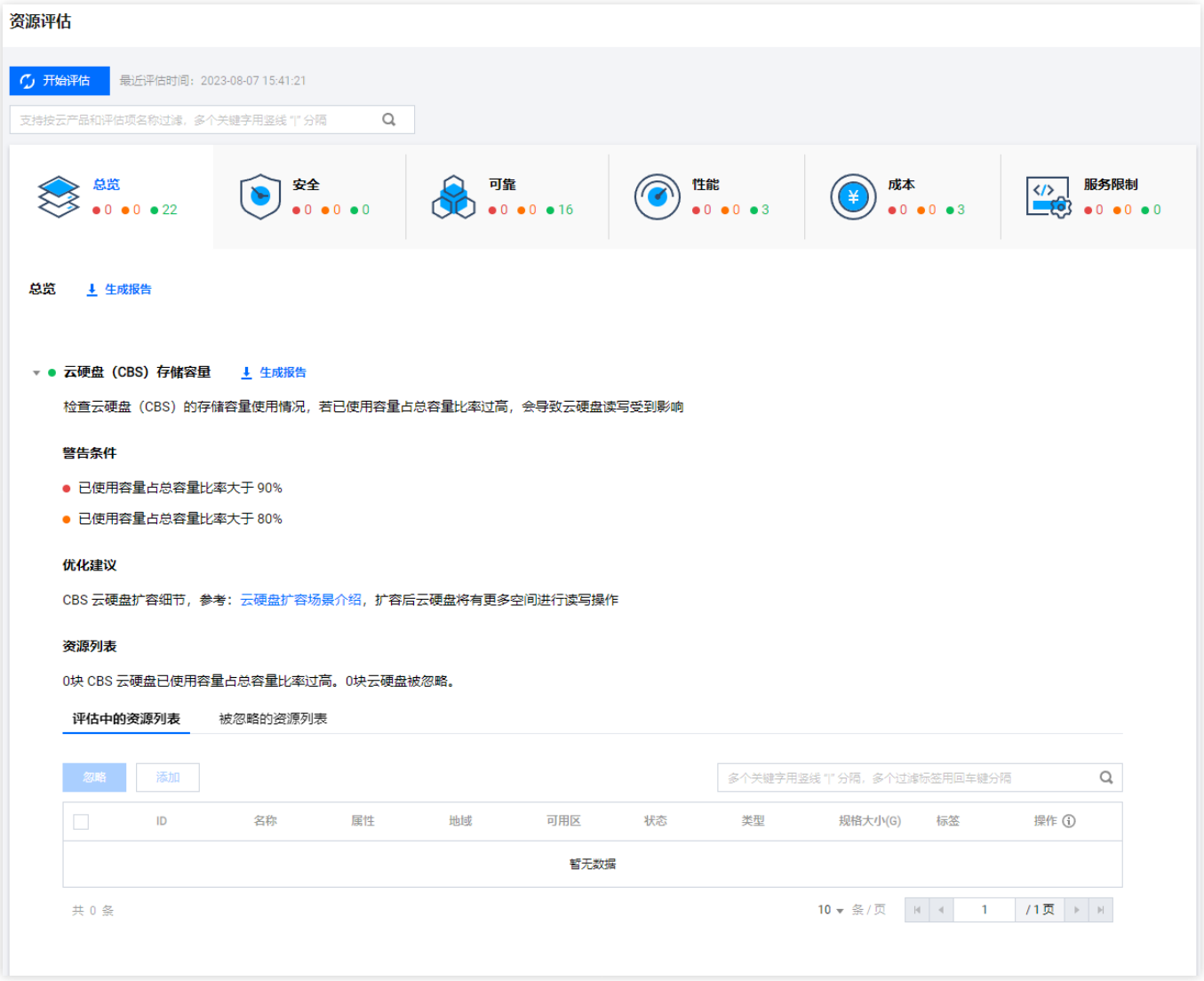
本文介绍如何通过云顾问控制台生成资源评估报告，您可参考本文按需获取报告文件。

前提条件

已授权云顾问访问相关云产品配置权限。

操作步骤

1. 登录云顾问控制台。
2. 在左侧导航栏中，单击资源评估，进入资源评估页面。
3. 单击开始评估，系统开始执行云资源评估操作。
4. 评估完成后，单击某个风险类型模块，可查看对应的风险项，展开风险项，可查看风险项详情，包括警告条件、优化建议、资源列表等信息。



- 警告条件：触发该风险项的告警规则。
- 优化建议：建议用户可以采取的措施，来规避该风险项。
- 资源列表：本次评估中，存在中高风险项的资源列表。

5. 单击生成报告，可针对某个风险类型模块或某个评估项生成相应的报告。
6. 报告生成后，单击下载报告，按需选择下载报告类型后，单击确定即可获取报告。

告警订阅

操作场景

云顾问支持用户通过告警订阅功能，将评估结果发送至指定邮箱。本文介绍如何通过云顾问控制台进行订阅。

操作步骤

1. 登录云顾问控制台。
2. 在左侧导航栏中，单击告警订阅，进入告警订阅页面。
3. 开启订阅，按需选择发送频率、发送时间，添加接收邮箱后，单击应用修改。
    - 每天发送：

    - 每周发送：


告警订阅
订阅
开启后，系统会定期将最新的云顾问评估结果发送到指定邮箱中。
发送频率
每天
每周
发送星期
星期五
发送时间
17:00
接收人邮箱
尚未设置邮箱
去添加
应用修改

评估设置

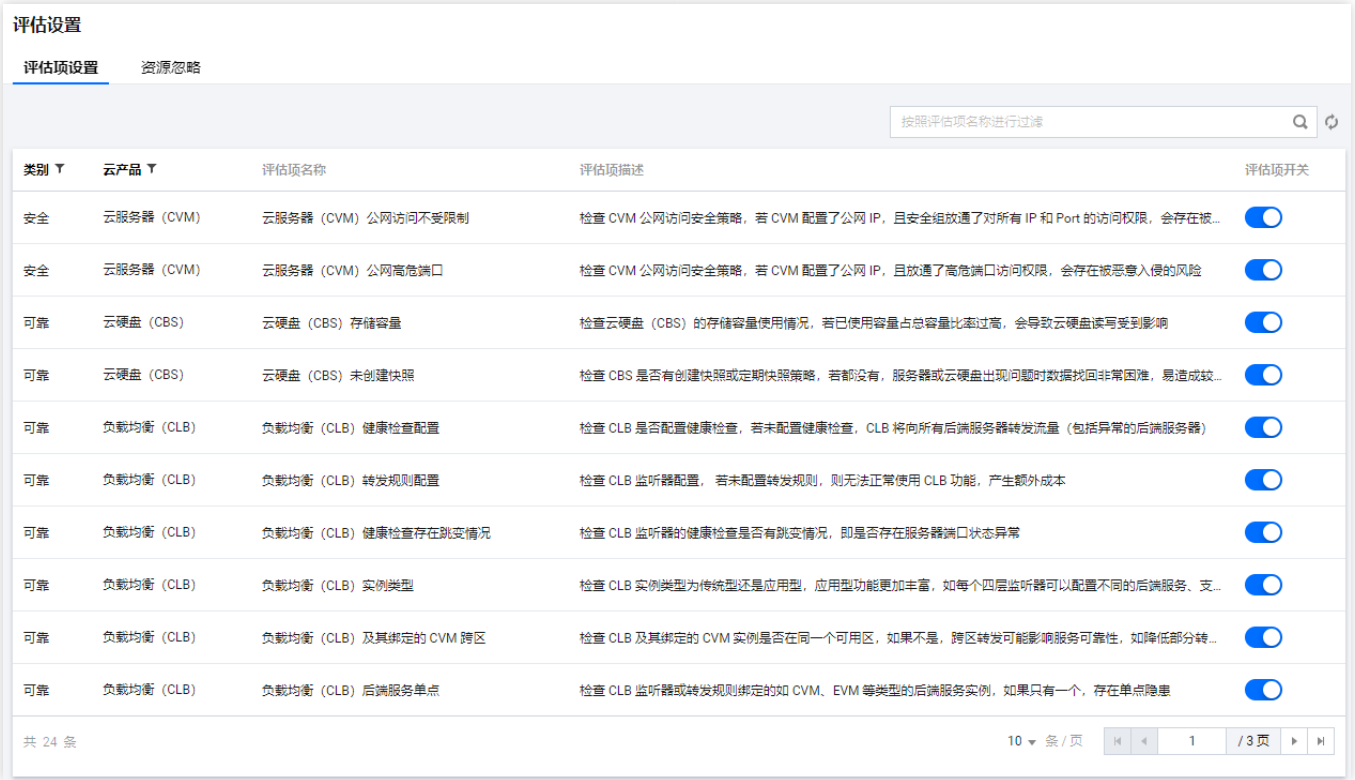
操作场景

- 云顾问支持用户自行开启或关闭评估项。若关闭某个评估项，则下次巡检将不会包含该项。
- 云顾问支持设置忽略以下维度的资源，使其不参与巡检。
  - 已配置特定标签键和标签值的资源
  - 实例
    在资源设置为“忽略”后，您可按需将资源重新添加至巡检中。

操作步骤

开启或关闭评估项

1. 登录云顾问控制台。
2. 在左侧导航栏中，单击评估设置，进入评估设置页面。
3. 在评估项设置页签中，可设置开启或关闭某个评估项。
   支持按类别、云产品筛选评估项。



忽略或添加资源

基于标签忽略资源

1. 在评估设置页面，单击资源忽略页签。
2. 单击**+标签**，选择需忽略的标签键及标签值。
3. 单击保存，已配置该标签键及标签值的资源将不参与巡检。

基于标签添加资源

1. 在评估设置页面，单击资源忽略页签。
2. 选择需修改或删除已配置的标签键及标签值，将对应资源添加至巡检。

基于实例忽略资源

在资源评估页面，展开评估项，勾选评估中的资源列表中需忽略的实例，单击忽略，则该实例不再参与巡检。

基于实例添加资源

在资源评估页面，展开评估项，勾选被忽略的资源列表中需添加的实例，单击添加，则该实例将添加至巡检。

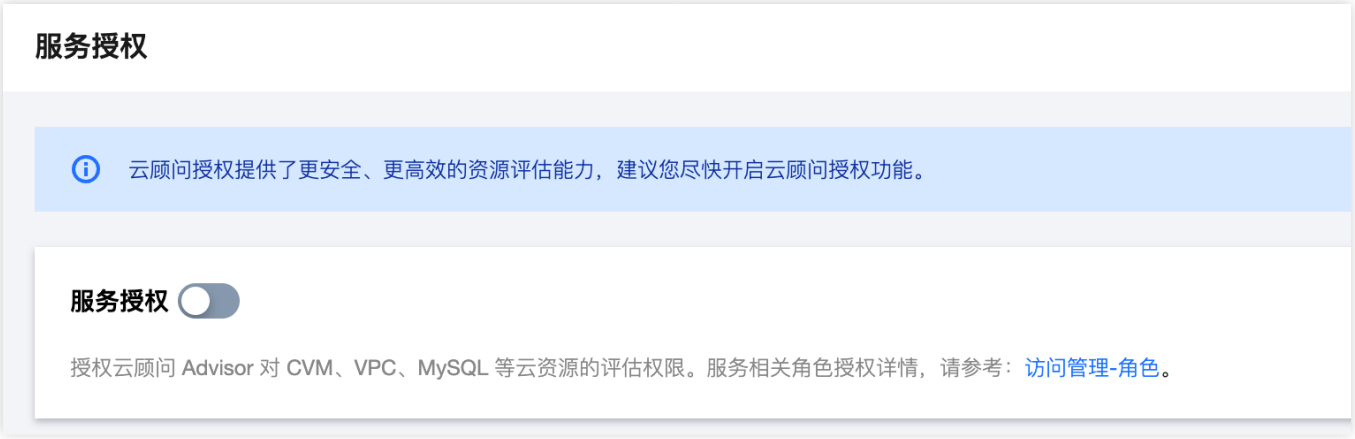
服务授权

操作场景

首次登录云顾问控制台时，需开启云顾问授权功能，对当前账号授予云顾问对云服务器（CVM）、私有网络 (VPC)、云数据库等云资源的评估权限。

操作步骤

1. 登录云顾问控制台。
2. 在左侧导航栏中，单击服务授权，进入服务授权页面。



3. 打开服务授权开关，即可完成授权，平台的访问管理>角色会出现一个专属云顾问的角色。

4. 在服务授权页面中，开启服务授权后，可开启报告解读，授权云专家架构师读取评估报告内容并提供相关优化建议。


服务授权
服务授权
授权云顾问 Advisor 对 CVM、VPC、MySQL 等云资源的评估权限。服务相关角色授权详情，请参考：访问管理-角色。
状态 授权成功
报告解读
授权云专家架构师读取评估报告内容，对报告内容进行专业的解读，并提供系统的优化建议。

常见问题

授权操作会获取用户密钥吗?

不会。云顾问是基于访问管理CAM服务角色的授权机制，通过临时密钥的方式来读取指定云资源配置，不会获取用户账户下密钥信息。

评估操作是否会影响服务性能?

不会。云顾问在进行资源评估操作时，仅仅是通过云API读取资源配置信息，不涉及业务数据流的操作，因此不会对服务性能产生影响。

是否会修改资源的配置?

不会。云顾问基于最小权限原则，仅仅读取指定资源的配置信息，基于配置信息来评估安全风险，不会修改资源的配置。

如何忽略评估项目?

您可以在评估设置 > 评估项设置页面中，在制定评估项条目中，单击关闭，停止该项目的评估。

如何忽略指定的云资源?

在评估项详情页中，勾选需要忽略的云资源，单击忽略，系统执行下一次评估操作时，将忽略该资源。

API文档
云顾问（cloudadvisor）
版本（2020-07-21）
API 概览

API版本

V3

任务相关接口

接口名称	接口功能
DescribeTaskStrategyRisks	查询评估项风险实例列表

其他相关接口

接口名称	接口功能
DescribeStrategies	查询评估项信息

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。

任务相关接口
查询评估项风险实例列表

1. 接口描述

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

查询评估项风险实例列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-12-02 21:32:48。

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：DescribeTaskStrategyRisks
Version	是	否	String	公共参数，本接口取值：2020-07-21
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
StrategyId	是	否	Uint64	评估项ID 示例值： 1
Limit	否	否	Uint64	限制数量,默认100 示例值： 10
Offset	否	否	Uint64	偏移量,默认0 示例值： 0
TaskID	否	否	String	任务ID 示例值： e9ea4f2a-98c5-47ed-9706-a7446a06ad1a
Filters	否	否	Array of Filters	过滤条件，支持: 1. level 等级，2：中风险，3：高风险 2. fuzzy 模糊搜索，用作ID或名称过滤

参数名称	必选	允许NULL	类型	描述
				示例值：查看
IncludeAllFields	否	否	Bool	是否包含所有字段 示例值： false

3. 输出参数

参数名称	类型	描述
RiskFieldsDesc	Array of RiskFieldsDesc	示例值：查看
StrategyId	Uint64	评估项ID 示例值： 1
RiskTotalCount	Uint64	风险实例个数 示例值： 10
Risks	String	风险实例详情列表，需要json decode 示例值： 无
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	参数错误。
InternalError.System	系统错误。
InternalError.DependsDb	依赖的db出错。
InvalidParameter	参数错误。
InternalError	内部错误。
ResourceNotFound	资源不存在。
InvalidParameterValue	参数取值错误。

其他相关接口
查询评估项信息

1. 接口描述

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

用于查询评估项的信息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-12-02 21:27:41。

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：DescribeStrategies
Version	是	否	String	公共参数，本接口取值：2020-07-21
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
Strategies	Array of DescribeStrategie	评估项列表 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.System	系统错误。
InternalError.DependsDb	依赖的db出错。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

数据结构

RiskFieldsDesc

风险实例字段描述

被如下接口引用：DescribeTaskStrategyRisks

名称	必选	允许NULL	类型	描述
Field	是	否	String	字段ID 示例值： InstanceState
FieldName	是	否	String	字段名称 示例值： 状态
FieldType	是	否	String	字段类型, string: 字符串类型，例如"aa" int: 整形，例如 111 stringSlice : 字符串数组类型，例如["a", "b"] tags: 标签类型, 例如: [{"Key":"kkk","Value":"vvv"},{"Key":"kkk2","Value":"vvv2"}] 示例值： string
FieldDict	是	是	Array of KeyValue	字段值对应字典 示例值：查看

DescribeStrategie

评估项信息

被如下接口引用：DescribeStrategies

名称	必选	允许NULL	类型	描述
StrategyId	是	是	Uint64	评估项ID 示例值： 1
Name	是	是	String	评估项名称 示例值： 云数据库（MySQL）root 账号安全风险
Desc	是	是	String	评估项描述 示例值： 检查 MySQL 账号配

名称	必选	允许NULL	类型	描述
				置，若只存在 root 账号，没有其他应用账号，说明权限过大，存在误操作或恶意操作影响数据安全的风险
Ignore	是	是	String	忽略模版 示例值： 2 个实例被忽略。
Notice	是	是	String	警告模版 示例值： 2 个 MySQL 实例只设置了 root 账号。
Product	是	是	String	评估项对应产品ID 示例值： mysql
ProductDesc	是	是	String	评估项对应产品名称 示例值： 云数据库（MySQL）
Repair	是	是	String	评估项优化建议 示例值： 建议依据最小化权限原则，根据不同的业务需求创建不同的账号，并分配适当的权限，提升数据隔离性和安全性，也便于管理。请查看最佳实践
GroupId	是	是	Uint64	评估项类别ID 示例值： 1
GroupName	是	是	String	评估项类别名称 示例值： 安全
Conditions	是	是	Array of DescribeStrategiesCondition	评估项风险列表 示例值：查看

DescribeStrategiesCondition

评估项警告条件

被如下接口引用：DescribeStrategies

名称	必选	允许NULL	类型	描述
ConditionId	是	是	Uint64	警告条件ID 示例值： 1

名称	必选	允许NULL	类型	描述
Level	是	是	Uint64	警告级别，2:中风险，3:高风险 示例值： 3
LevelDesc	是	是	String	警告级别描述 示例值： 高风险
Desc	是	否	String	警告条件描述 示例值： MySQL 实例只设置了 root 账号

KeyValue

键值对

被如下接口引用：DescribeTaskStrategyRisks

名称	必选	允许NULL	类型	描述
Key	否	否	String	键名 示例值： key
Value	否	否	String	键名对应值 示例值： val

Filters

过滤条件

被如下接口引用：DescribeTaskStrategyRisks

名称	必选	允许NULL	类型	描述
Name	是	是	String	过滤名称 示例值： key
Values	是	是	Array of String	过滤值 示例值： ["val"]

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.MissAKSK	缺少平台配置。
InternalError.CallbackFail	回调出错。
ResourceNotFound.NotExistTask	任务不存在。
ResourceInUse	资源被占用。

错误码	说明
InvalidParameter	参数错误。
InternalError.DependsDb	依赖的db出错。
InternalError	内部错误。
InvalidParameter.SecretIdOrSecretKeyError	平台配置错误。
InternalError.DependsMq	依赖的mq出错。
ResourceNotFound	资源不存在。
InternalError.System	系统错误。
InvalidParameterValue	参数取值错误。
InvalidParameter.ParamError	参数错误。
InternalError.DependsApi	依赖的其他api出错。