凭据管理系统（SSM）

产品文档

腾讯云TCE

文档目录

- 产品介绍
  - 产品介绍
  - 产品功能
  - 应用场景
- 快速入门
- 操作指南
  - 创建凭据
  - 编辑凭据
  - 凭据多版本管理
  - 删除凭据
  - 日志审计
  - 访问控制
    - 概述
    - 子账号管理
    - 创建访问控制策略
- 最佳实践
  - 凭据托管和使用
  - 如何轮换托管凭据
  - 调用示例
- 常见问题
- 词汇表
- API文档
  - 凭据管理服务（ssm）
    - 版本（2019-09-23）
      - API概览
      - 调用方式
        - 接口签名v1
        - 接口签名v3
        - 请求结构
        - 返回结果
        - 公共参数
      - 数据安全相关接口
        - 创建凭据
        - 删除凭据信息
        - 删除指定版本的凭据
        - 获取凭据详细信息
        - 停用凭据
        - 启用凭据
        - 获取控制台展示region列表
        - 获取凭据明文
        - 获取用户服务开通状态
        - 获取指定凭据下的版本列表信息。
        - 获取凭据的详细信息列表

增加新版本凭据
恢复计划删除中的凭据
更新凭据描述信息
更新凭据内容
数据结构
错误码

产品介绍
产品介绍

最近更新时间: 2024-06-12 15:06:00

什么是凭据管理系统

凭据管理系统（Secrets Manager，SSM）为用户提供凭据的创建、检索、更新、删除等全生命周期的管理服务，结合资源级角色授权及全面细致的审计管控，轻松实现对敏感凭据的统一管理。用户或应用程序可通过调用凭据管理系统 API，规避敏感配置、敏感凭据硬编码等风险问题，同时可有效避免敏感信息泄密以及权限失控带来的业务风险。

产品优势

企业级凭据管理

凭据管理系统专注于解决用户敏感凭据管理问题，有效避免程序硬编码导致的明文泄密，以及权限失控带来的业务风险。

全生命周期管理

凭据管理系统可以为用户提供凭据的创建、检索、更新、删除、权限管控等全生命周期的管理服务，结合资源级角色授权及全面细致的审计管控，轻松实现对敏感凭据的统一管理。

安全可靠

凭据管理系统架构采用集群化部署方式，通过分布式数据库存储系统实现数据存储与容灾备份。业务侧用户也可多地域创建同样的凭据，实现业务侧的跨区域容灾。

加密存储

凭据通过腾讯云金融专区密钥管理系统进行加密存储，基于第三方认证的硬件安全模块（HSM）来生成和保护加密密钥。检索凭据时，通过 TLS 安全传输到服务器本地。

按需付费

用户在使用凭据管理系统时，仅按实际使用量收费。用户可按照在凭据管理系统中管理的凭据数量和 API 调用次数进行付费，无最低费用和设置费用。

产品功能

最近更新时间: 2024-06-12 15:06:00

针对敏感配置、敏感凭据硬编码带来的泄露风险问题，所有的凭据由密钥管理系统（KMS）进行加密保护，并且提供简单易用的 API 和 SDK，能够降低用户的使用成本和管理成本。 用户可通过凭据管理系统轻松实现对数据库凭证、API 密钥和其他密钥、敏感配置的集中检索、管理以及加密存储，有效避免程序硬编码导致的明文泄密，以及权限失控带来的业务风险。

安全可控的凭据检索

从应用程序的源代码中删除硬编码凭据，将代码中的硬编码凭据替换为对凭据管理系统 API 调用，以便用编程的方式动态检索及管理凭据。

凭据加密存储与传输

凭据管理系统使用密钥管理系统（KMS）安全保护的主密钥（CMK）作为加密密钥，并对所管理的凭据内容进行加密存储，使用凭据时，将通过 TLS 安全传输到服务器本地。

应用层凭据轮换

用户可通过凭据管理系统按周期更新敏感凭据内容，依赖该凭据的所有应用点将自动完成同步，安全实现凭据轮换管理，同时确保依赖该凭据业务的连续性。

存储多类型凭据

通过 Name-Value 的方式存储多种类型数据，Value 部分最大支持4096字节，例如数据库凭据、账号密码、IP 端口等。

资源级访问授权

凭据管理系统与腾讯云金融专区访问管理（CAM）集成，通过身份管理和策略管理方式确保只有授权用户可以访问或修改凭据，并可以将这些策略附加到用户或角色，指定这些用户或角色可以访问哪些凭据。

精细化监管审计

凭据管理系统与腾讯云金融专区审计（CloudAudit）结合，支持对用户的腾讯云金融专区账号进行监管、合规性检查、操作审核和风险审核等，同时可记录凭据管理操作和凭据使用情况。

高可用容灾备份

凭据管理系统架构采用集群化部署方式，通过分布式数据库存储系统实现数据固化与容灾备份，实现业务侧的跨区域容灾。

安全合规性说明

凭据管理系统与密钥管理系统（KMS）相关联，密钥管理系统底层使用经过第三方认证的硬件安全模块（HSM）来生成和保护密钥，符合监管和合规要求。

应用场景

最近更新时间: 2024-06-12 15:06:00

凭据集中管控

- **场景痛点**：为保障业务开发敏捷性，系统中往往存在大量的敏感信息，例如账户信息、Tokens、证书、SSH 密钥、API 密钥等，因此需要对敏感凭据进行统一的存储、检索、使用等全生命周期管控。
- **场景举例**：多应用敏感配置信息凭证加密存储、查询管理等生命周期管理。
- **面临挑战**：敏感凭据硬编码、权限管理混乱、凭据托管管理难。
- **解决方案**：业务开发者可通过凭据管理系统控制台、SDK 或命令行界面创建、使用、存储敏感配置信息的凭据。通过凭据管理系统与访问控制 CAM、云审计 CloudAudit 产品的结合，业务管理者可实现对企业凭据全生命周期的统一管理。

敏感凭据检索管理

- **场景痛点**：当用户访问应用程序或服务时，需创建身份验证的数字证书，例如密码、令牌、证书、SSH 密钥或 API 密钥等各种类型机密信息，通常直接使用明文方式嵌入在应用程序的配置文件中，安全性较低。通过凭据管理系统可有效避免敏感凭据硬编码等风险问题。
- **场景举例**：数据库凭据、API 密钥、账号密码等。
- **面临挑战**：敏感凭据信息泄露。
- **解决方案**：用户可以将代码中的硬编码凭证（包括密码）替换为对凭据管理系统 API 的调用，以便用编程的方式动态查询凭据，由于该凭据中不包含敏感信息，所以可以保证密钥不被泄露。

凭据轮换

- **技术痛点**：为提升系统安全性，需要对敏感凭据进行定期更新，通过凭据管理系统可以实现更新。
- **场景举例**：应用层凭据轮换。
- **面临挑战**：凭据轮换时要求对目标凭据具备依赖性的应用或配置同步更新，多应用系统凭据更新容易遗漏，可能带来应用中断风险。

- **解决方案**：在凭据管理系统中通过控制台新增凭据版本或通过调用 API 更新目标凭据内容，用户可自主选择全量或者灰度轮换凭据，实现对依赖目标凭据的所有应用点的同步更新。

快速入门

最近更新时间: 2024-06-12 15:06:00

用户可以通过凭据管理系统（SSM）实现对数据库凭证、API 密钥和其他密钥、敏感配置等各类型凭据的集中检索、管理以及加密存储，可以有效避免程序硬编码导致的明文泄密以及权限失控带来的业务风险。

第1步：注册账号

注册腾讯云金融专区账号，并完成实名认证。

第2步：立即购买

进入 凭据管理系统购买页 ，阅读并勾选相关费用说明，单击【立即开通】，即可开通凭据管理系统。

第3步：控制台操作

服务开通后，通过凭据管理系统控制台、SDK、命令行界面进行创建、存储、删除等凭据生命周期管理。

- 凭据管理系统将借助密钥管理系统实现对敏感凭据的加密存储，因此在使用凭据管理系统前，请确认已开通密钥管理系统。
- 为保障凭据管理系统功能的正常使用，请开通密钥管理系统对凭据管理系统服务的角色授权。您可前往 访问管理 进行授权设置。

操作指南
创建凭据

最近更新时间: 2024-06-12 15:06:00

操作场景

您可以在凭据管理系统控制台中创建凭据，凭据创建成功后，您可以对凭据进行启用、禁用、修改、计划删除等操作。

操作步骤

1. 登录 凭据管理系统控制台，在左侧导航栏中，单击【凭据列表】。
2. 在凭据列表左上角选择需要创建凭据的地区，单击【新建】。
3. 在弹出的配置框中，输入相应信息，信息输入完成后，单击【确定】，返回凭据列表，新创建的凭据会出现在凭据列表首位。

字段说明：

- 凭据名称：名称长度1 - 128字节，使用字母、数字、连接符（ - ）、下划线（ _ ）的组合，首字符必须为字母或者数字。
- 凭据版本：必填。
- 凭据内容：必填。
- 描述信息：非必填。
- 选择加密密钥：
  - 使用凭据管理系统在密钥管理系统（KMS）中默认创建的主密钥（CKM）进行加密。
  - 使用自定义加密密钥。

若使用凭据管理系统表明您已开启 密钥管理系统，您可以通过以下两种方案创建加密密钥：

- 选择在密钥管理系统控制台中默认创建的云产品主密钥作为加密密钥，并通过信封加密方案进行加密存储。
- 选择在密钥管理系统控制台中创建一个用户密钥，将该密钥作为自定义的加密密钥对凭据进行加密存储。

编辑凭据

最近更新时间: 2024-06-12 15:06:00

操作场景

您可以登录腾讯云金融专区凭据管理系统控制台查看并编辑凭据信息列表、名称、状态、所属地区等凭据详情。

编辑凭据

1. 登录 凭据管理系统控制台，在左侧导航栏中，单击【凭据列表】，在凭据列表左上方可以切换不同地区，根据需求查看并编辑其他地区的凭据列表。

2. 在页面右侧搜索框中，输入凭据全称或部分名称，查找您需要的凭据。

3. 单击凭据名称，即可查看该凭据的详细信息，同时可以对该凭据的密钥进行启用或禁用状态设置。

4. 进入该凭据的详情页，您可查看凭据的名称、状态、描述信息以及版本号等内容，同时可以对凭据内容进行更改、删除、版本管理等操作。

凭据多版本管理

最近更新时间: 2024-06-12 15:06:00

操作场景

凭据管理系统为用户提供凭据多版本管理服务。用户可以通过多版本管理特性，灰度实现应用层的凭据轮换。

操作步骤

1. 登录 凭据管理系统控制台 ，在左侧导航栏中，单击【凭据列表】，在凭据列表左上方可以切换不同地区，找到您需要添加凭据版本的凭据，单击凭据名称进入凭据详情页。

2. 在凭据管理区域，单击【添加】，进入添加凭据信息页面。

3. 在添加凭据信息页面，填写凭据版本和凭据内容，填写完成后，单击【添加】，即可完成凭据新版本添加操作。

4. 凭据添加完后，如需删除，可以在对应版本右侧操作栏，单击【删除】，在弹出的删除确认框中，确认删除即可。

> **说明：**
>
> 同一个凭据可以多版本并存，每个凭据最多同时存在10个版本。

删除凭据

最近更新时间: 2024-06-12 15:06:00

操作场景

为避免误删除操作，凭据管理系统使用计划删除机制，即对删除操作强制执行0 - 30天的等待期，及确认删除后等待0 - 30天再进行删除。

> **注意：**
>
> 凭据删除后将无法恢复，此凭据下的所有凭据内容也将无法调用。

操作步骤

1. 登录 凭据管理系统控制台，在左侧导航栏中，单击【凭据列表】，在凭据列表左上方可以切换不同地区，根据需求可以查看其他地区的凭据列表。

2. 在凭据列表中，选择需要计划删除的凭据，若凭据是启用状态，请先对凭据进行禁用操作，然后在计划删除操作栏中，单击【计划删除】。

3. 输入计划删除天数，单击【确定】，凭据将按计划删除。

> **说明：**
>
> 若选择等待期为"0"天，凭据将立即删除。

4. 在1 - 30天等待期内，您可以对计划删除的凭据进行取消删除操作。若需取消删除凭据，可以在右侧计划删除操作栏，单击【取消删除】，即可取消删除凭据。确认取消删除后，凭据密钥重置为"启用"状态，可对该凭据进行禁用、修改、删除等操作。

日志审计

最近更新时间: 2024-06-12 15:06:00

操作场景

凭据管理系统与腾讯云金融专区 云审计 CloudAudit 结合，对您的腾讯云金融专区账号进行监管、合规性检查、操作审核和风险审核服务，可记录所有凭据管理操作和凭据使用情况。

操作步骤

1. 您可以登录 云审计控制台 ，在左侧导航栏中，单击【操作记录】，最多可查看最近30天用户在腾讯云金融专区账号下的操作记录。

2. 单击目标事件前方的展开按钮，可查看事件详情。

可查看内容包括：

- **操作记录列表**：您可以查看操作记录列表，以及对应操作事件时间、用户名、事件名称、资源类型、资源名称等。
- **操作记录详情**：您可以获取单个操作记录详情，包括访问密钥、区域、错误码、事件 ID、事件名称、事件源、事件时间、请求 ID、源 IP 地址、用户名。

访问控制
概述

最近更新时间: 2024-06-12 15:06:00

如果您不需要对子账户进行凭据管理系统相关资源的访问控制，您可以跳过此章节，跳过此章节并不影响您对其他文档的理解和使用。 如果同时使用凭据管理系统、私有网络（VPC）、云服务器（CVM）、数据库等服务，且这些服务由不同人进行管理，但都共享同一个云账号密钥，将存在密钥由多人共享，泄密风险高等问题，且无法限制其它人访问权限，易产生误操作造成安全风险问题。 访问控制（CAM）用于管理腾讯云金融专区账号下资源访问权限，您可以通过 CAM 的身份管理和策略管理控制各子账号的资源操作权限。例如，您的主账号下有一个凭据，您只想让子账号 A 使用该凭据，而子账号 B 不能使用，就可以通过在 CAM 中配置策略，对子账号的权限进行控制。

CAM 基本概念

主账号通过给子账号绑定策略实现授权，策略设置可精确到多个（API、资源、用户、用户组、允许、拒绝、条件）维度。

- **账号**
- **主账号**：腾讯云金融专区资源归属及资源使用、计量、计费的基本主体，可登录腾讯云金融专区服务。
- **子账号**：由主账号创建的账号，有确定的身份 ID 和身份凭证，且能登录到腾讯云金融专区控制台。主账号可以创建多个子账号(用户)。子账号默认不拥有资源，必须由所属主账号进行授权。
- **身份凭证**：包括登录凭证和访问证书两种，登录凭证指用户登录名和密码，访问证书指云 API 密钥（SecretId 和 SecretKey）。
- **资源与权限**
- **资源**：资源是云服务中被操作的对象，如一个凭据管理系统的凭据，云服务器实例，COS 存储桶，VPC 实例等。
- **权限**：权限是指允许或拒绝某些用户执行某些操作。默认情况下，主账号拥有其名下所有资源的访问权限，而子账号没有主账号下任何资源的访问权限。
- **策略**：策略是定义和描述一条或多条权限的语法规范。主账号通过将策略关联到用户或用户组完成授权。

  如需了解更多请参见 腾讯云金融专区访问管理 CAM。

子账号管理

最近更新时间: 2024-06-12 15:06:00

概述

本文详细介绍如何创建子账号，并授予子账号管理凭据管理系统的权限。

操作步骤

1. 创建子账号。用主账号登录腾讯云金融专区 访问管理 CAM 控制台，在左侧导航中，选择【用户】>【用户列表】，在【用户列表】页面下，单击【新建用户】，即可创建子账号。

2. 创建 API 密钥。单击子账号名称，进入子账号详情页，选择【API 密钥】>【新建密钥】，即可创建 SecretId 和 SecretKey，您通过该 API 密钥访问凭据管理系统。

   > **说明：**
   >
   > 如果您不需要通过 API 管理凭据管理系统，可直接操作授权子账号。

3. 授权子账号。对于新创建的子账号，通过授权凭据管理系统策略，即可允许该子账号访问凭据管理系统。在子账号详情页，选择【权限】>【关联策略】，进入添加策略页面。

4. 添加策略。在添加策略页面，单击【从策略列表中选取策略关联】，选择合适的凭据管理系统策略，选择【下一步】>【确定】，即可授权子账号访问凭据管理系统权限。

创建访问控制策略

最近更新时间: 2024-06-12 15:06:00

可授权的资源类型

资源级权限是能够指定用户对哪些资源具有执行操作的能力。凭据管理系统部分接口支持使用资源级权限对凭据进行操作，可控制允许用户何时执行操作或是否允许用户使用的特定资源。 例如，您授权用户拥有广州地域凭据的权限，在 CAM 中可授权的资源类型为：

```
qcs::ssm:ap-guangzhou:uin/${uin}:*
qcs::ssm:ap-guangzhou::*
```

您授权接口访问某个 UIN 创建的所有凭据，则资源类型为：

```
qcs::ssm:$region:uin/$uin:secret/creatorUin/*
```

您授权接口访问某个具体的凭据，则资源类型为：

```
qcs::ssm:$region:uin/$uin:secret/creatorUin/$creatorUin/$secretName
```

其中：

- `$region` ：指代地域。
- `$uin` ：指代主账号 ID。
- `$creatorUin` ：指代创建该资源的账号 ID。
- `$secretName` ：指代需要配置的凭据名称。

资源级授权接口

如下 API 接口 DeleteSecretVersion、UpdateDescription、RestoreSecret、EnableSecret、PutSecretValue、DescribeSecret、UpdateSecret、DeleteSecret、GetSecretValue、DisableSecret、ListSecretVersionIds 资源路径为：

```
qcs::ssm:$region:uin/$uin:secret/*
qcs::ssm:$region:uin/$uin:secret/creatorUin/*
qcs::ssm:$region:uin/$uin:secret/creatorUin/$creatorUin/$secretName
```

接口级别授权列表

API 接口	描述信息
CreateSecret	创建新的凭据。

API 接口	描述信息
GetRegions	获取可用 region 列表，用于控制台展示。
GetServiceStatus	获取服务状态，用于判定服务是否开通。
ListSecrets	获取所有凭据列表信息。

最佳实践
凭据托管和使用

最近更新时间: 2024-06-12 15:06:00

应用程序或服务中，用于身份验证的各种认证信息，如口令、令牌、证书、SSH 密钥或 API 密钥等，通常情况下直接明文保存在应用程序的配置文件中，安全性较低。借助凭据管理系统将这些敏感认证信息加密存储，可有效避免敏感凭据明文编码带来的风险问题。

操作流程

以数据库用户名和口令的托管为例，介绍基本的凭据托管和使用场景。

1. DB 管理员在目标数据库配置应用系统中，访问数据库所需的用户名和口令。
2. DB 管理员在 SSM 凭据管理系统中创建一个凭据对象，用来加密存储步骤1中获取的用户名和口令。
3. 应用系统需要访问数据库时，需要向 SSM 凭据管理系统请求访问凭据。
4. SSM 凭据管理系统获取到存储的凭据密文，解密后将凭据明文通过 HTTPS 返回给应用系统。
5. 应用系统读取并解析 SSM 凭据管理系统返回的凭据明文，从而获取用户名和口令，并可使用该账号访问目标数据库。
6. DB 管理员可为凭据创建多个版本内容，也可更新凭据版本内容，实现配置同步、版本管理、凭据轮换。

应用效果

对应用系统而言，通过调用 SSM 凭据管理系统的 API 或 SDK 来获取敏感的凭据明文，可避免在程序或配置中，明文编码凭据带来的信息泄露风险，调用对比如下：

- 使用本地存储数据库连接信息，连接信息明文保存在本地配置或者代码文件中，敏感凭据易泄露。
  - 获取凭据明文示例代码：

```
func GetDBConfig() string {
dbConnStr := "user:password@tcp(127.0.0.1:3306)/test"
return dbConnStr
}
```

  - 使用凭据明文示例代码：

```
conn, err := sql.Open("mysql", GetDBConfig())
if err != nil {
// error handler
}
```

- 使用 SSM 凭据管理系统连接数据库 DB 时，代码和本地配置中无需明文存储 DB 的连接信息。
  - 获取凭据明文示例代码：

```
func GetDBConfig(secretName, version *string) string {
credential := common.NewCredential(
secretId,
secretKey,
)
```

```
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewGetSecretValueRequest()
request.SecretName = secretName
request.VersionId = version

resp, err := client.GetSecretValue(request)
if err != nil {
// error handler
}
return *resp.Response.SecretString
}
```

- 使用凭据明文示例代码：

```
secretName := "MySecret1"
version := "MyVersion1"
conn, err := sql.Open("mysql", GetDBConfig(&secretName, &version))
if err != nil {
// error handler
}
```

如何轮换托管凭据

最近更新时间: 2024-06-12 15:06:00

为提升系统安全性，要求对目标凭据具备依赖性的应用配置同步更新。当多种应用系统在本地存储凭据内容时，在凭据更新时容易遗漏，从而带来应用中断风险。使用凭据管理系统，可以实现凭据管理系统内一处凭据更新，处处生效。此外，还可以为凭据创建配置多个版本，实现凭据的灰度更新和轮换。

可以使用以下两种方式进行凭据轮换：

- **方式一**：增加新的凭据版本，业务侧通过更新获取凭据的版本号实现灰度轮换。

- **方式二**：直接修改当前使用凭据的内容，业务侧下次调用接口获取凭据时，会自动更新凭据内容。

调用示例

最近更新时间: 2024-06-12 15:06:00

托管和保护凭据

- **示例**： DB 管理员创建凭据 MySecret，并指定版本 MyVersion1，将数据库连接信息交由 SSM 加密存储。未指定 KMS 密钥时，SSM 将自动创建一个默认密钥。

```
var (
secretName = "MySecret1"
version = "MyVersion1"
plainText = "user:password@tcp(127.0.0.1:3306)/test"
)

func ExampleCreateSecret() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewCreateSecretRequest()
request.SecretName = &secretName
request.VersionId = &version
request.SecretString = &plainText

resp, err := client.CreateSecret(request)
if err != nil {
// error handler
}
fmt.Println(*resp.Response.SecretName)

// create ok
}
```

查看凭据元数据信息

- **示例1**： 获取凭据列表和凭据元数据信息。

```
func ExampleListSecrets() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)
```

```
request := ssm.NewListSecretsRequest()

resp, err := client.ListSecrets(request)
if err != nil {
// error handler
}
fmt.Println(resp.Response.SecretMetadatas)
// get secrets metadata
// ...
}
```

- **示例2**：根据名称获取凭据MySecret1的版本信息

```
var (
secretName = "MySecret1"
)
func ExampleListSecretVersionIds() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewListSecretVersionIdsRequest()
request.SecretName = &secretName

resp, err := client.ListSecretVersionIds(request)
if err != nil {
// error handler
}
fmt.Println(resp.Response.Versions)

// get version list
// ...
}
```

获取 SSM 存储的明文敏感数据

- **示例**：服务调用方根据凭据名称 MySecret1 和指定版本 MyVersion1 获取 DB 连接明文信息。

```
var (
secretName = "MySecret1"
version = "MyVersion1"
)

func ExampleGetSecretValue() {
credential := common.NewCredential(
secretId,
secretKey,
```

```
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewGetSecretValueRequest()
request.VersionId = &version
request.SecretName = &secretName

resp, err := client.GetSecretValue(request)
if err != nil {
// error handler
}
fmt.Println(*resp.Response.SecretString)

// get plain text, connect db
// ...
}
```

更新凭据内容

- **示例**：根据凭据名称 MySecret1 和指定版本 MyVersion1 更新 DB 连接明文信息。

```
var (
secretName = "MySecret1"
version = "MyVersion1"
newSecretValue = "user2:password2@tcp(127.0.0.1:3306)/test"
)

func ExamplePutSecretValue() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewPutSecretValueRequest()
request.SecretName = &secretName
request.VersionId = &version
request.SecretString = &newSecretValue

resp, err := client.PutSecretValue(request)
if err != nil {
// error handler
}
fmt.Println(*resp.Response.SecretName)
// secret updated
// ...
}
```

禁用、删除和恢复凭据

- **示例1**：禁用凭据，禁用后服务无法再获取此凭据存储的所有内容。

```
var (
secretName = "MySecret1"
)
func ExampleDisableSecret() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewDisableSecretRequest()
request.SecretName = &secretName
resp, err := client.DisableSecret(request)

if err != nil {
// error handler
}
fmt.Println(*resp.Response.SecretName)
// secret disabled
// ...
}
```

- **示例2**：删除凭据，可以设置计划删除时间，在计划删除时间前，可以恢复凭据。

> !只有禁用后的凭据才能删除。

```
func ExampleDeleteSecret() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewDeleteSecretRequest()
request.SecretName = &secretName
request.RecoveryWindowInDays = &recoverWindowsInDays

resp, err := client.DeleteSecret(request)
if err != nil {
// error handler
}
fmt.Println(*resp.Response.DeleteTime)
// secret deleted
// ...
}
```

- **示例3**：恢复凭据，处于计划删除状态的凭据可以重新恢复并启。

```
func ExampleRestoreSecret() {
credential := common.NewCredential(
secretId,
secretKey,
)
cpf := profile.NewClientProfile()
cpf.HttpProfile.Endpoint = endpoint
client, _ := ssm.NewClient(credential, region, cpf)

request := ssm.NewRestoreSecretRequest()
request.SecretName = &secretName

resp, err := client.RestoreSecret(request)
if err != nil {
// error handler
}
fmt.Println(*resp.Response.SecretName)
// secret restored
// ...
}
```

常见问题

最近更新时间: 2024-06-12 15:06:00

什么是凭据管理系统？

凭据管理系统（Secrets Manager，SSM）为用户提供凭据的创建、检索、更新、删除等全生命周期的管理服务，结合资源级角色授权及全面细致的审计管控，轻松实现对敏感凭据的统一管理。

为什么要使用凭据管理系统？

凭据管理系统可以实现对数据库凭证、API 密钥和其他密钥、敏感配置等信息的集中查询、管理以及加密存储。可有效避免程序硬编码导致的明文泄密、以及权限失控带来的业务风险。

什么是凭据？

凭据是用于对应用程序进行身份验证的敏感凭证信息，如数据库凭据、账号密码、API 密钥、SSH 密钥等内容。您也可以通过 Name-Value 的方式，在凭据管理系统中存储多种类型敏感数据作为凭据内容，例如敏感地址、IP 端口等配置内容。

开通凭据管理系统前为什么需要开通密钥管理系统（KMS）？

凭据管理系统使用腾讯云金融专区密钥管理系统（KMS） 安全保护的主密钥（CKM）作为加密密钥，密钥可以是账户默认创建的主密钥，也可以是用户自定义创建的主密钥，实现对各类应用程序密钥的集中管理，因此在开通凭据管理系统前需要开通密钥管理系统（KMS）。

如何接入凭据管理系统？

您的业务应用程序无论在腾讯云金融专区内还是腾讯云金融专区外，均可使用以下两种方式接入腾讯云金融专区凭据管理系统：

- 通过凭据管理系统 API 或 SDK调用凭据管理系统。
- 通过 凭据管理系统控制台 管理凭据全生命周期。

词汇表

最近更新时间: 2024-06-12 15:06:00

凭据

凭据是用于对应用程序进行身份验证的敏感凭证信息，如数据库凭据、账号密码、API 密钥、SSH 密钥等内容。您也可以通过 Name-Value 的方式，在凭据管理系统中存储多种类型敏感数据作为凭据内容，例如敏感地址、IP 端口等配置内容。

硬编码

硬编码是将数据直接嵌入到程序或其他可执行对象源代码中的软件开发实践。

密钥管理系统

密钥管理系统（Key Management Service，KMS）是一款安全管理类服务，可以让用户轻松创建和管理密钥，保护密钥的保密性、完整性和可用性，满足用户多应用多业务的密钥管理需求，符合监管和合规要求。

主密钥

在凭据管理系统中，用户主密钥（Customer Master Keys，CMK）是由腾讯云金融专区为用户保管的主密钥，这些主密钥受到经过第三方认证的硬件安全模块（HSM）的保护，可通过主密钥来加解密业务凭据。

TLS 协议

安全传输层（Transport Layer Security，TLS)，TLS 是建立在传输层 TCP 协议之上的协议，服务于应用层，实现了将应用层的报文进行加密后再交由 TCP 进行传输的功能。

API文档
凭据管理服务（ssm）
版本（2019-09-23）
API概览

最近更新时间: 2024-06-18 14:31:19

API版本

V3

数据安全相关接口

接口名称	接口功能
CreateSecret	创建凭据
DeleteSecret	删除凭据信息
DeleteSecretVersion	删除指定版本的凭据
DescribeSecret	获取凭据详细信息
DisableSecret	停用凭据
EnableSecret	启用凭据
GetRegions	获取控制台展示region列表
GetSecretValue	获取凭据明文
GetServiceStatus	获取用户服务开通状态
ListSecretVersionIds	获取指定凭据下的版本列表信息。
ListSecrets	获取凭据的详细信息列表
PutSecretValue	增加新版本凭据
RestoreSecret	恢复计划删除中的凭据
UpdateDescription	更新凭据描述信息
UpdateSecret	更新凭据内容

调用方式
接口签名v1

最近更新时间: 2024-06-18 14:31:19

tcecloud 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tcecloud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
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	ap-guangzhou

参数名称	中文	参数值
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' : 'ap-guangzhou',
'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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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

错误代码	错误描述
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tcecloud 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
- JavaScript
- .NET

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： http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
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.Integer.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", "ap-guangzhou"); // 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' : 'ap-guangzhou',
'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("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-06-18 14:31:19

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\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" + hashedCanonicalRequest;
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)).toLowerCase();
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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-06-18 14:31:19

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-06-18 14:31:19

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 correct."
},
"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 (TODO: 重复信息, 是否真的需要?)

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。
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

最近更新时间: 2024-06-18 14:31:19

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。

参数名称	类型	必选	描述
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。
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。tcecloud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。

数据安全相关接口
创建凭据

最近更新时间: 2024-06-18 14:31:19

1. 接口描述

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

创建新的凭据信息，通过KMS进行加密保护。每个Region最多可创建存储1000个凭据信息。

默认接口请求频率限制：100次/秒。

接口更新时间：2020-11-24 17:35:59。

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：CreateSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	凭据名称，同一region内不可重复，最长128字节，使用字母、数字或者 - _ 的组合，第一个字符必须为字母或者数字。
VersionId	是	否	String	凭据版本，查询凭据信息时需要根据SecretName 和 VersionId进行查询，最长64 字节，使用字母、数字或者 - _ . 的组合并且以字母或数字开头。
Description	否	否	String	描述信息，用于详细描述用途等，最大支持2048字节。
KmsKeyId	否	否	String	指定对凭据进行加密的KMS CMK。如果为空则表示使用Secrets Manager为您默认创建的CMK进行加密。您也可以指定在同region 下自行创建的KMS CMK进行加密。
SecretBinary	否	否	String	二进制凭据信息base64编码后的明文。SecretBinary 和 SecretString 必须且只能设置一个，最大支持4096字节。
SecretString	否	否	String	文本类型凭据信息明文（不需要进行base64编码）。SecretBinary 和 SecretString 必须且只能设置一个，，最大支持4096字节。
Tags	否	否	Array of Tag	标签列表

3. 输出参数

参数名称	类型	描述
SecretName	String	新创建的凭据名称。
VersionId	String	新创建的凭据版本。
TagCode	Uint64	标签操作的返回码. 0: 成功；1: 内部错误；2: 业务处理错误
TagMsg	String	标签操作的返回信息
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
LimitExceeded	
InvalidParameterValue	
InternalError	
ResourceInUse.SecretExists	
UnauthorizedOperation	
FailedOperation.AccessKmsError	
ResourceUnavailable.NotPurchased	

删除凭据信息

最近更新时间: 2024-06-18 14:31:19

1. 接口描述

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

删除指定的凭据信息，可以通过RecoveryWindowInDays参数设置立即删除或者计划删除。对于计划删除的凭据，在删除日期到达之前状态为 PendingDelete，并可以通过RestoreSecret 进行恢复，超出指定删除日期之后会被彻底删除。您必须先通过 DisableSecret 停用凭据后才可以进行（计划）删除操作。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:20:46。

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：DeleteSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要删除的凭据名称。
RecoveryWindowInDays	否	否	Uint64	指定计划删除日期，单位（天），0（默认）表示立即删除， 1-30 表示预留的天数，超出该日期之后彻底删除。

3. 输出参数

参数名称	类型	描述
SecretName	String	指定删除的凭据名称。
DeleteTime	Int64	凭据删除的日期，unix时间戳。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceNotFound	
FailedOperation	
ResourceUnavailable.NotPurchased	

删除指定版本的凭据

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于直接删除指定凭据下的单个版本凭据，删除操作立即生效，对所有状态下的凭据版本都可以删除。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:20:58。

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：DeleteSecretVersion
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定凭据名称。
VersionId	是	否	String	指定该名称下需要删除的凭据的版本号。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
VersionId	String	凭据版本号。
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
InvalidParameterValue	

错误码	描述
InternalError	
UnauthorizedOperation	
ResourceNotFound	
ResourceUnavailable.NotPurchased	

获取凭据详细信息

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

获取凭据的详细属性信息。

默认接口请求频率限制：40次/秒。

接口更新时间：2019-12-26 14:21:41。

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：DescribeSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要获取凭据详细信息的凭据名称。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
Description	String	凭据描述信息。
KmsKeyId	String	用于加密的KMS CMK ID。
CreateUin	Uint64	创建者UIN。
Status	String	凭据状态：Enabled、Disabled、PendingDelete
DeleteTime	Uint64	删除日期，uinx 时间戳，非计划删除状态的凭据为0。
CreateTime	Uint64	创建日期。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceNotFound	
ResourceUnavailable.NotPurchased	

停用凭据

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

停用指定的凭据，停用后状态为 Disabled，无法通过接口获取该凭据的明文。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:22:13。

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：DisableSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定停用的凭据名称。

3. 输出参数

参数名称	类型	描述
SecretName	String	停用的凭据名称。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	

错误码	描述
ResourceNotFound	
ResourceUnavailable.NotPurchased	

启用凭据

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于开启凭据，状态为Enabled。可以通过 GetSecretValue 接口获取凭据明文。处于PendingDelete状态的凭据不能直接开启，需要通过RestoreSecret 恢复后再开启使用。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:21:25。

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：EnableSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定启用凭据的名称。

3. 输出参数

参数名称	类型	描述
SecretName	String	启用的凭据名称。
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
InvalidParameterValue	
InternalError	

错误码	描述
UnauthorizedOperation	
ResourceNotFound	
FailedOperation	
ResourceUnavailable.NotPurchased	

获取控制台展示region列表

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

获取控制台展示region列表

默认接口请求频率限制：20次/秒。

接口更新时间：2019-12-26 14:26:27。

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：GetRegions
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)

3. 输出参数

参数名称	类型	描述
Regions	String	region列表。
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	
UnauthorizedOperation	

获取凭据明文

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

获取指定凭据名称和版本的凭据明文信息，只能获取启用状态的凭据明文。

默认接口请求频率限制：300次/秒。

接口更新时间：2019-12-26 14:20:31。

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：GetSecretValue
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定凭据的名称。
VersionId	是	否	String	指定对应凭据的版本号。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据的名称。
VersionId	String	该凭据对应的版本号。
SecretBinary	String	在创建凭据(CreateSecret)时，如果指定的是二进制数据，则该字段为返回结果，并且使用base64进行编码，应用方需要进行base64解码后获取原始数据。SecretBinary和SecretString只有一个不为空。
SecretString	String	在创建凭据(CreateSecret)时，如果指定的是普通文本数据，则该字段为返回结果。SecretBinary和SecretString只有一个不为空。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceNotFound	
FailedOperation.AccessKmsError	
ResourceUnavailable.ResourceDisabled	
ResourceUnavailable.ResourcePendingDeleted	
ResourceUnavailable.NotPurchased	

获取用户服务开通状态

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用户获取用户SecretsManager服务开通状态。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:26:47。

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：GetServiceStatus
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)

3. 输出参数

参数名称	类型	描述
ServiceEnabled	Bool	true表示服务已开通，false 表示服务尚未开通。
InvalidType	Int64	服务不可用类型： 0-未购买，1-正常， 2-欠费停服， 3-资源释放。
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	
UnauthorizedOperation	

获取指定凭据下的版本列表信息。

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于获取指定凭据下的版本列表信息

默认接口请求频率限制：40次/秒。

接口更新时间：2019-12-26 14:24:23。

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：ListSecretVersionIds
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	凭据名称。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
Versions	VersionInfo	VersionId列表。
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
InvalidParameterValue	
InternalError	

错误码	描述
UnauthorizedOperation	
ResourceNotFound	
ResourceUnavailable.NotPurchased	

获取凭据的详细信息列表

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于获取所有凭据的详细列表，可以指定过滤字段、排序方式等。

默认接口请求频率限制：30次/秒。

接口更新时间：2019-12-26 14:24:40。

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：ListSecrets
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
Offset	否	否	Uint64	查询列表的起始位置，以0开始，不设置默认为0。
Limit	否	否	Uint64	单次查询返回的最大数量，0或不设置则使用默认值 20。
OrderType	否	否	Uint64	根据创建时间的排序方式，0或者不设置则使用降序排序， 1 表示升序排序。
State	否	否	Uint64	根据凭据状态进行过滤，默认为0表示查询全部，1 表示查询Enabed 凭据列表，2表示查询Disabled 凭据列表， 3 表示查询PendingDelete 凭据列表。
SearchSecretName	否	否	String	根据凭据名称进行过滤，为空表示不过滤。

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	根据State和SearchSecretName 筛选的凭据总数。
SecretMetadatas	SecretMetadata	返回凭据信息列表。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceUnavailable.NotPurchased	

增加新版本凭据

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口在指定名称的凭据下增加新版本的凭据内容，一个凭据下最多可以支持10个版本。只能对处于Enabled 和 Disabled 状态的凭据添加新的版本。

默认接口请求频率限制：100次/秒。

接口更新时间：2019-12-26 14:25:17。

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：PutSecretValue
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要增加版本的凭据名称。
VersionId	是	否	String	指定新增加的版本号，最长64 字节，使用字母、数字或者 - _ . 的组合并且以字母或数字开头。
SecretBinary	否	否	String	二进制凭据信息，使用base64编码。SecretBinary 和 SecretString 必须且只能设置一个。
SecretString	否	否	String	文本类型凭据信息明文（不需要进行base64编码），SecretBinary 和 SecretString 必须且只能设置一个。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
VersionId	String	新增加的版本号。
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
LimitExceeded	
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceNotFound	
FailedOperation.AccessKmsError	
FailedOperation	
ResourceInUse.VersionIdExists	
ResourceUnavailable.NotPurchased	

恢复计划删除中的凭据

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于恢复计划删除（PendingDelete状态）中的凭据，取消计划删除。取消计划删除的凭据将处于Disabled 状态，如需恢复使用，通过EnableSecret 接口开启凭据。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:21:12。

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：RestoreSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要恢复的凭据名称。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
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
InvalidParameterValue	
InternalError	

错误码	描述
UnauthorizedOperation	
ResourceNotFound	
FailedOperation	
ResourceUnavailable.NotPurchased	

更新凭据描述信息

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于修改指定凭据的描述信息，仅能修改Enabled 和 Disabled 状态的凭据。

默认接口请求频率限制：20次/秒。

接口更新时间：2019-12-26 14:19:45。

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：UpdateDescription
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要更新描述信息的凭据名。
Description	是	否	String	新的描述信息，最大长度2048个字节。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
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	
UnauthorizedOperation	

错误码	描述
ResourceNotFound	
FailedOperation	
ResourceUnavailable.NotPurchased	

更新凭据内容

最近更新时间: 2024-06-18 14:31:20

1. 接口描述

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

该接口用于更新指定凭据名称和版本号的内容，调用该接口会对新的凭据内容加密后覆盖旧的内容。仅允许更新Enabled 和 Disabled 状态的凭据。

默认接口请求频率限制：50次/秒。

接口更新时间：2019-12-26 14:21:59。

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：UpdateSecret
Version	是	否	String	公共参数，本接口取值：2019-09-23
Region	是	否	String	公共参数，详见产品支持的**地域列表**(TODO)
SecretName	是	否	String	指定需要更新凭据内容的名称。
VersionId	是	否	String	指定需要更新凭据内容的版本号。
SecretBinary	否	否	String	新的凭据内容为二进制的场景使用该字段，并使用base64进行编码。SecretBinary 和 SecretString 只能一个不为空。
SecretString	否	否	String	新的凭据内容为文本的场景使用该字段，不需要base64编码。SecretBinary 和 SecretString 只能一个不为空。

3. 输出参数

参数名称	类型	描述
SecretName	String	凭据名称。
VersionId	String	凭据版本号。
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
InvalidParameterValue	
InternalError	
UnauthorizedOperation	
ResourceNotFound	
FailedOperation.AccessKmsError	
FailedOperation	
ResourceUnavailable.NotPurchased	

数据结构

最近更新时间: 2024-06-18 14:31:20

SecretMetadata

凭据的基础信息

被如下接口引用：ListSecrets

名称	必选	允许NULL	类型	描述
SecretName	是	否	String	凭据名称。
Description	是	否	String	凭据的描述信息。
KmsKeyId	是	否	String	用于加密凭据的KMS KeyId。
CreateUin	是	否	Uint64	创建者UIN。
Status	是	否	String	凭据状态：Enabled、Disabled、PendingDelete
DeleteTime	是	否	Uint64	凭据删除日期，对于status为PendingDelete 的有效，unix时间戳。
CreateTime	是	否	Uint64	凭据创建时间，unix时间戳。
KmsKeyType	是	否	String	用于加密凭据的KMS CMK类型，DEFAULT 表示SecretsManager 创建的默认密钥， CUSTOMER 表示用户指定的密钥。

Tag

标签键和标签值

被如下接口引用：CreateSecret

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键
TagValue	是	否	String	标签值

VersionInfo

凭据版本号列表信息

被如下接口引用：ListSecretVersionIds

名称	必选	允许NULL	类型	描述
VersionId	是	否	String	版本号。

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Uint64	创建时间，unix时间戳。

错误码

最近更新时间: 2024-06-18 14:31:20

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 correct."
},
"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。
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。

错误码	说明
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
UnauthorizedOperation	
InvalidParameterValue	
ResourceInUse.SecretExists	
ResourceUnavailable.NotPurchased	
FailedOperation	
ResourceInUse.VersionIdExists	
InternalError	
LimitExceeded	
ResourceUnavailable.ResourcePendingDeleted	
ResourceNotFound	
ResourceUnavailable.ResourceDisabled	
FailedOperation.AccessKmsError	