弹性 MapReduce

API 文档

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2020 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

版权所有：腾讯云计算（北京）有限责任公司

API 文档
更新历史

最近更新时间：2020-10-23 08:05:00

第 15 次发布

发布时间：2020-10-23 08:04:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CreateInstance
  - 新增入参：ApplicationRole

第 14 次发布

发布时间：2020-09-21 08:03:23

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- ScaleOutInstance
  - 新增入参：ClickHouseClusterName, ClickHouseClusterType, YarnNodeLabel

修改数据结构：

- PersistentVolumeContext
  - 新增成员：DiskNum

第 13 次发布

发布时间：2020-08-19 08:11:03

本次发布包含了以下内容：

改善已有的文档。

新增数据结构：

- HostVolumeContext
- PersistentVolumeContext
- PodVolume

修改数据结构：

- PodSpec
  - 新增成员：CpuType, PodVolumes
  - **修改成员：**DataVolumes

第 12 次发布

发布时间：2020-07-16 08:10:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- DescribeClusterNodes
  - 新增入参：HardwareResourceType, SearchFields
  - 新增出参：HardwareResourceTypeList
- ScaleOutInstance
  - 新增入参：HardwareResourceType, PodSpec

新增数据结构：

- PodSpec
- SearchItem

修改数据结构：

- NodeHardwareInfo
  - 新增成员：HardwareResourceType

第 11 次发布

发布时间：2020-04-23 08:09:03

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- ClusterInstancesInfo
  - 新增成员：ProductId

第 10 次发布

发布时间：2020-04-02 08:08:50

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- ClusterInstancesInfo
  - 新增成员：AliasInfo

第 9 次发布

发布时间：2020-03-16 08:09:17

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- ClusterInstancesInfo
  - 新增成员：ServiceClass
- NodeHardwareInfo
  - 新增成员：AutoFlag

- PriceResource
  - 新增成员：DiskNum, LocalDiskNum

第 8 次发布

发布时间：2020-02-28 19:34:32

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- DescribeClusterNodes

新增数据结构：

- CdbInfo
- MultiDiskMC
- NodeHardwareInfo

第 7 次发布

发布时间：2020-02-21 16:04:57

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CreateInstance
  - 新增入参：CbsEncrypt, MetaType, UnifyMetaInstanceId, MetaDBInfo
- InquiryPriceCreateInstance
  - 新增入参：MetaType, UnifyMetaInstanceId, MetaDBInfo, ProductId

新增数据结构：

- CustomMetaInfo

第 6 次发布

发布时间：2020-02-21 09:42:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CreateInstance
  - 新增入参：Tags, DisasterRecoverGroupIds
- DescribeInstances
  - 新增出参：TagKeys
- InquiryPriceScaleOutInstance
  - 新增入参：RouterCount
- ScaleOutInstance
  - 新增入参：RouterCount, SoftDeployInfo, ServiceNodeInfo, DisasterRecoverGroupIds, Tags
  - 新增出参：FlowId, BillId

新增数据结构：

- Tag

修改数据结构：

- ClusterInstancesInfo
  - 新增成员：IsWoodpeckerCluster, MetaDb, Tags, HiveMetaDb
- EmrProductConfigOutter
  - 新增成员：RouterNodeSize, SupportHA, SecurityOn, SecurityGroup, CbsEncrypt
- OutterResource
  - 新增成员：InstanceType
- PriceResource
  - 新增成员：InstanceType, Tags
- Resource
  - 新增成员：Tags, InstanceType, LocalDiskNum, DiskNum
- UpdateInstanceSettings
  - 新增成员：InstanceType

第 5 次发布

发布时间：2019-09-19 14:52:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- InquiryPriceRenewInstance
- InquiryPriceUpdateInstance

修改接口：

- CreateInstance
  - 新增入参：CheckSecurity, ExtendFsField
  - **修改入参：** ResourceSpec, ClientToken
  - **删除出参：** Result
- DescribeInstances
  - 新增入参：DisplayStrategy, ProjectId, OrderField, Asc
  - 新增出参：TotalCnt, ClusterList
  - **删除出参：** Result
- InquiryPriceCreateInstance
  - **修改入参：** ResourceSpec
  - 新增出参：OriginalCost, DiscountCost, TimeUnit, TimeSpan
  - **删除出参：** Result
- InquiryPriceScaleOutInstance
  - 新增出参：OriginalCost, DiscountCost, Unit, PriceSpec
  - **删除出参：** Result
- ScaleOutInstance
  - 新增入参：UnNecessaryNodeList
  - **修改入参：** ClientToken
  - 新增出参：InstanceId, DealNames, ClientToken
  - **删除出参：** Result
- TerminateInstance

版权所有：腾讯云计算（北京）有限责任公司

- 新增入参：ResourceIds
- **删除出参：** Result
- TerminateTasks
- **删除出参：** Result

新增数据结构：

- ClusterInstancesInfo
- EmrProductConfigOutter
- NewResourceSpec
- OutterResource
- PriceResource
- Resource
- UpdateInstanceSettings

删除数据结构：

- ClusterInfoResult
- ClusterInstanceInfo
- CreateInstanceResult
- EMRProductConfigSettings
- InquiryPriceResult
- NodeSpec
- ResourceSpec
- ScaleOutInstanceResult
- TerminateResult

修改数据结构：

- COSSettings
- **修改成员：** LogOnCosPath
- Placement
- **修改成员：** ProjectId

第 4 次发布

发布时间：2019-08-09 13:06:05

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CreateInstance
- 新增入参：RemoteLoginAtCreate
- **修改入参：** PreExecutedFileSettings
- ScaleOutInstance
- **修改入参：** PreExecutedFileSettings

修改数据结构：

- MultiDisk
- 新增成员：Count
- **修改成员：** DiskType, Volume
- PreExecuteFileSettings

- 新增成员：RunOrder, WhenRun, CosFileName, CosFileURI, CosSecretId, CosSecretKey, AppId
- **修改成员：**Path, Args, Bucket, Region, Domain

第 3 次发布

发布时间：2019-05-16 16:14:48

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- CreateInstance
  - 新增入参：NeedMasterWan

新增数据结构：

- MultiDisk

修改数据结构：

- NodeSpec
  - 新增成员：MultiDisks

第 2 次发布

发布时间：2019-04-12 12:08:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- CreateInstance
- DescribeInstances
- InquiryPriceScaleOutInstance
- ScaleOutInstance
- TerminateInstance
- TerminateTasks

新增数据结构：

- COSSettings
- ClusterInfoResult
- ClusterInstanceInfo
- CreateInstanceResult
- EMRProductConfigSettings
- LoginSettings
- PreExecuteFileSettings
- ScaleOutInstanceResult
- TerminateResult

第 1 次发布

发布时间：2019-03-28 20:22:49

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- InquiryPriceCreateInstance

新增数据结构：

- InquiryPriceResult
- NodeSpec
- Placement
- ResourceSpec
- VPCSettings

简介

最近更新时间：2019-07-24 15:18:41

弹性 MapReduce API 升级到 3.0 版本。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。

弹性 MapReduce（EMR）是腾讯云提供的云上 Hadoop 托管服务，为用户提供便捷的 Hadoop 集群部署、软件安装、配置修改、监控告警、弹性伸缩等功能，为企业及用户提供安全稳定的大数据处理解决方案。

API 概览

最近更新时间：2020-11-12 08:05:10

信息查询相关接口

接口名称	接口功能
DescribeClusterNodes	查询硬件节点信息
DescribeInstances	查询EMR实例
InquiryPriceCreateInstance	创建实例询价
InquiryPriceRenewInstance	续费询价
InquiryPriceScaleOutInstance	扩容询价
InquiryPriceUpdateInstance	变配询价

集群生命周期相关接口

接口名称	接口功能
CreateInstance	创建EMR实例
TerminateInstance	销毁EMR实例

扩缩容相关接口

接口名称	接口功能
ScaleOutInstance	实例扩容
TerminateTasks	缩容Task节点

调用方式
请求结构

最近更新时间：2020-09-28 08:04:52

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 emr.tencentcloudapi.com ，也支持指定地域域名访问，例如广州地域的域名为 emr.ap-guangzhou.tencentcloudapi.com 。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到**最近的**某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 emr.ap-guangzhou.tencentcloudapi.com 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	emr.tencentcloudapi.com
华南地区(广州)	emr.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	emr.ap-shanghai.tencentcloudapi.com
华北地区(北京)	emr.ap-beijing.tencentcloudapi.com
西南地区(成都)	emr.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	emr.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	emr.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	emr.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	emr.ap-bangkok.tencentcloudapi.com
亚太南部(孟买)	emr.ap-mumbai.tencentcloudapi.com
亚太东北(首尔)	emr.ap-seoul.tencentcloudapi.com
亚太东北(东京)	emr.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	emr.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	emr.na-siliconvalley.tencentcloudapi.com
北美地区(多伦多)	emr.na-toronto.tencentcloudapi.com
欧洲地区(法兰克福)	emr.eu-frankfurt.tencentcloudapi.com
欧洲地区(莫斯科)	emr.eu-moscow.tencentcloudapi.com

注意：由于金融区和非金融区是隔离不互通的，因此当访问金融区服务时（公共参数 Region 为金融区地域），需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致。

金融区接入地域	金融区域名

金融区接入地域	金融区域名
华东地区(上海金融)	emr.ap-shanghai-fsi.tencentcloudapi.com
华南地区(深圳金融)	emr.ap-shenzhen-fsi.tencentcloudapi.com

2. 通信协议

腾讯云 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），必须使用签名方法 v3（TC3-HMAC-SHA256）。
- application/x-www-form-urlencoded，必须使用签名方法 v1（HmacSHA1 或 HmacSHA256）。
- multipart/form-data（仅部分接口支持），必须使用签名方法 v3（TC3-HMAC-SHA256）。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1（HmacSHA1、HmacSHA256）时不得超过1MB。POST 请求使用签名方法 v3（TC3-HMAC-SHA256）时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2020-10-23 08:05:00

公共参数是用于标识用户和接口签名的参数，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

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 文档。

注意：接口文档中的示例由于目的是展示接口参数用法，简化起见，使用的是签名方法 v1 GET 请求，如果依旧想使用签名方法 v1 请参考下文章节。

使用签名方法 v3 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	-	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，通常为域名前缀，例如域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 emr； - 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，计算过程详见 文档。
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。

假设用户想要查询广州地域的云服务器实例列表，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKID********EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
```

```
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST（application/json）请求结构示例：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID********EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST（multipart/form-data）请求结构示例（仅特定的接口支持）：

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID********EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

签名方法 v1

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见文档

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	-	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 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 。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID********EXAMPLE

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded

Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID********EXAMPLE
```

地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

地域	取值
华北地区(北京)	ap-beijing

地域	取值
西南地区(成都)	ap-chengdu
西南地区(重庆)	ap-chongqing
华南地区(广州)	ap-guangzhou
港澳台地区(中国香港)	ap-hongkong
亚太南部(孟买)	ap-mumbai
华东地区(南京)	ap-nanjing
华东地区(上海)	ap-shanghai
华东地区(上海金融)	ap-shanghai-fsi
华南地区(深圳金融)	ap-shenzhen-fsi
亚太东南(新加坡)	ap-singapore
欧洲地区(莫斯科)	eu-moscow
美国西部(硅谷)	na-siliconvalley

版权所有：腾讯云计算（北京）有限责任公司

签名方法 v3

最近更新时间：2020-09-08 08:04:42

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 json 格式，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 API Explorer 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 Python、Java、PHP、Go、NodeJS、.NET、C++。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共请求参数中指定该签名结果并以指定的方式和格式发送请求。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- **用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄漏，请立刻禁用该安全凭证。**

申请安全凭证的具体步骤如下：

1. 登录 腾讯云管理中心控制台。
2. 前往 云API密钥 的控制台页面。
3. 在 云API密钥 页面，单击【新建密钥】即可以创建一对密钥。

签名过程

云 API 支持 GET 和 POST 请求。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 格式绝大多数接口均支持，multipart 格式只有特定接口支持，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。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

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。我们选择该接口是因为：

1. 云服务器默认已开通，该接口很常用；
2. 该接口是只读的，不会改变现有资源的状态；
3. 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。

假设用户的 SecretId 和 SecretKey 分别是：`AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******` 和 `Gu5t9xGARNpq86cd98joQYCN3*******`。用户想查看广州区云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******/2019-02-25/cvm/tc3_request, Sig
nedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
```

```
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST ）。此示例取值为 POST。
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，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode，字符集 UTF8，推荐使用编程语言标准库，所有特殊字符均需编码，大写形式。
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 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
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 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 content-type;host
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com

content-type;host
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

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
```

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。
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。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031。

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：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031
```

3. 计算签名

1）计算派生签名密钥，伪代码如下：

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3*******"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate 、SecretService 和 SecretSigning 是二进制的数据，可能包含不可打印字符，此处不展示中间结果。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，此处的伪代码密钥参数在后，请以实际编程语言为准。通常标准库函数会提供二进制格式的计算值，也即此处使用的，也会提供打印友好的十六进制格式的计算值，将在下面计算签名结果时使用。

字段名称	解释
SecretKey	原始的 SecretKey，即 Gu5t9xGARNpq86cd98joQYCN3*******。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2）计算签名，伪代码如下：

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

此示例计算结果是 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c 。

4. 拼接 Authorization

按如下格式拼接 Authorization：

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

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host。
Signature	签名值。此示例计算结果是 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******/2019-02-25/cvm/tc3_request, SignedH
eaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}
```

签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 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
- NodeJS
- .NET
- C++

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。

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
private final static Charset UTF8 = StandardCharsets.UTF_8;
private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";
private final static String CT_JSON = "application/json; charset=utf-8";

public static byte[] hmac256(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(UTF8));
}

public static String sha256Hex(String s) throws Exception {
MessageDigest md = MessageDigest.getInstance("SHA-256");
byte[] d = md.digest(s.getBytes(UTF8));
return DatatypeConverter.printHexBinary(d).toLowerCase();
}

public static void main(String[] args) throws Exception {
String service = "cvm";
String host = "cvm.tencentcloudapi.com";
String region = "ap-guangzhou";
String action = "DescribeInstances";
String version = "2017-03-12";
String algorithm = "TC3-HMAC-SHA256";
String timestamp = "1551113065";
//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 = "POST";
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n";
String signedHeaders = "content-type;host";

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name
\"}]}";
String hashedRequestPayload = sha256Hex(payload);
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 = sha256Hex(canonicalRequest);
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ************ 步骤 3：计算签名 ************
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(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("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\"")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\"")
.append(" -H \"X-TC-Action: ").append(action).append("\"")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\"")
.append(" -H \"X-TC-Version: ").append(version).append("\"")
.append(" -H \"X-TC-Region: ").append(region).append("\"")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
```

```
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": [u"未命名"]}]}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%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)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + host + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'
```

```
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'")
```

Golang

```
package main

import (
"crypto/hmac"
"crypto/sha256"
"encoding/hex"
"fmt"
"time"
)

func sha256hex(s string) string {
b := sha256.Sum256([]byte(s))
return hex.EncodeToString(b[:])
}

func hmacsha256(s, key string) string {
hashed := hmac.New(sha256.New, []byte(key))
hashed.Write([]byte(s))
return string(hashed.Sum(nil))
}

func main() {
secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secretKey := "Gu5t9xGARNpq86cd98joQYCN3*******"
host := "cvm.tencentcloudapi.com"
algorithm := "TC3-HMAC-SHA256"
service := "cvm"
version := "2017-03-12"
action := "DescribeInstances"
region := "ap-guangzhou"
//var timestamp int64 = time.Now().Unix()
var timestamp int64 = 1551113065

// step 1: build canonical request string
httpRequestMethod := "POST"
canonicalURI := "/"
canonicalQueryString := ""
canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
signedHeaders := "content-type;host"
payload := `{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}`
hashedRequestPayload := sha256hex(payload)
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
httpRequestMethod,
canonicalURI,
```

```
canonicalQueryString,
canonicalHeaders,
signedHeaders,
hashedRequestPayload)
fmt.Println(canonicalRequest)

// step 2: build string to sign
date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
hashedCanonicalRequest := sha256hex(canonicalRequest)
string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
algorithm,
timestamp,
credentialScope,
hashedCanonicalRequest)
fmt.Println(string2sign)

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*******";
$host = "cvm.tencentcloudapi.com";
```

```
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n"."host:".$host."\n";
$signedHeaders = "content-type;host";
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/".$credentialScope
.", SignedHeaders=content-type;host, Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"'
.' -H "Content-Type: application/json; charset=utf-8"'
.' -H "Host: '.$host.'"'
```

```
.' -H "X-TC-Action: '.$action.'"'
.' -H "X-TC-Timestamp: '.$timestamp.'"'
.' -H "X-TC-Version: '.$version.'"'
.' -H "X-TC-Region: '.$region.'"'
." -d '".$payload."'";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
require 'json'
require 'time'
require 'openssl'

# 密钥参数
secret_id = 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*******'

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ************ 步骤 1：拼接规范请求串 ************
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\n"
signed_headers = 'content-type;host'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in *******, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
http_request_method,
canonical_uri,
canonical_querystring,
canonical_headers,
signed_headers,
hashed_request_payload,
].join("\n")
```

```
puts canonical_request

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
algorithm,
timestamp.to_s,
credential_scope,
hashed_request_payload,
].join("\n")
puts string_to_sign

# ************* 步骤 3：计算签名 *************
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ************* 步骤 4：拼接 Authorization *************
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signat
ure=#{signature}"
puts authorization

puts 'curl -X POST ' + endpoint \
+ ' -H "Authorization: ' + authorization + '"' \
+ ' -H "Content-Type: application/json; charset=utf-8"' \
+ ' -H "Host: ' + host + '"' \
+ ' -H "X-TC-Action: ' + action + '"' \
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '"' \
+ ' -H "X-TC-Version: ' + version + '"' \
+ ' -H "X-TC-Region: ' + region + '"' \
+ " -d '" + payload + "'"
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Security.Cryptography;
using System.Text;

public class Application
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
```

```
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
return builder.ToString();
}
}
public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
{
string datestr = date.ToString("yyyy-MM-dd");
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 10
00;
// ************* 步骤 1：拼接规范请求串 *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n" + "host:" + endpoint + "\n";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("-----------------------------------");

// ************* 步骤 2：拼接待签名字符串 *************
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + requestTimestamp.ToString() + "\n" + credentialScope + "\n" + hashedCanon
icalRequest;
Console.WriteLine(stringToSign);
Console.WriteLine("-----------------------------------");
```

```
// ************* 步骤 3：计算签名 *************
byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
Console.WriteLine(signature);
Console.WriteLine("----------------------------------");

// ************* 步骤 4：拼接 Authorization *************
string authorization = algorithm + " "
+ "Credential=" + secretid + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", "
+ "Signature=" + signature;
Console.WriteLine(authorization);
Console.WriteLine("----------------------------------");

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}
public static void Main(string[] args)
{
// 密钥参数
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

string service = "cvm";
string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";

// 此处由于示例规范的原因，采用时间戳2019-02-26 00:44:25，此参数作为示例，如果在项目中，您应当使用：
// DateTime date = DateTime.UtcNow;
// 注意时区，建议此时间统一采用UTC时间戳，否则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instan
ce-name\"}]}";

Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

Console.WriteLine("POST https://cvm.tencentcloudapi.com");
```

```
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine();
Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
const hmac = crypto.createHmac('sha256', secret)
return hmac.update(message).digest(encoding)
}
function getHash(message, encoding = 'hex') {
const hash = crypto.createHash('sha256')
return hash.update(message).digest(encoding)
}
function getDate(timestamp) {
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}
function main(){
// 密钥参数
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******"

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
//时间处理, 获取世界时间日期
const date = getDate(timestamp)

// ************* 步骤 1：拼接规范请求串 *************
const signedHeaders = "content-type;host"

const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name
\"}]}"

const hashedRequestPayload = getHash(payload);
```

```
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + endpoint + "\n"

const canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload
console.log(canonicalRequest)
console.log("-----------------------------")

// ************* 步骤 2：拼接待签名字符串 *************
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\n" +
timestamp + "\n" +
credentialScope + "\n" +
hashedCanonicalRequest
console.log(stringToSign)
console.log("-----------------------------")

// ************* 步骤 3：计算签名 *************
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)
console.log("-----------------------------")

// ************* 步骤 4：拼接 Authorization *************
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)
console.log("-----------------------------")

const Call_Information = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'"
console.log(Call_Information)
```

```
}
main()
```

C++

```
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
string utcDate;
char buff[20] = {0};
// time_t timenow;
struct tm sttime;
sttime = *gmtime(&timestamp);
strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
utcDate = string(buff);
return utcDate;
}
string int2str(int64_t n)
{
std::stringstream ss;
ss << n;
return ss.str();
}
string sha256Hex(const string &str)
{
char buf[3];
unsigned char hash[SHA256_DIGEST_LENGTH];
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
snprintf(buf, sizeof(buf), "%02x", hash[i]);
NewString = NewString + buf;
}
return NewString;
}
string HmacSha256(const string &key, const string &input)
```

```
{
unsigned char hash[32];

HMAC_CTX *h;
#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )&input[0], input.length());
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#if OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif

std::stringstream ss;
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
ss << hash[i];
}

return (ss.str());
}
string HexEncode(const string &input)
{
static const char* const lut = "0123456789abcdef";
size_t len = input.length();

string output;
output.reserve(2 * len);
for (size_t i = 0; i < len; ++i)
{
const unsigned char c = input[i];
output.push_back(lut[c >> 4]);
output.push_back(lut[c & 15]);
}
return output;
}

int main()
{
// 密钥参数
```

```
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

string service = "cvm";
string host = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

// ************* 步骤 1：拼接规范请求串 *************
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string canonicalHeaders = "content-type:application/json; charset=utf-8\nhost:" + host + "\n";
string signedHeaders = "content-type;host";
string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name
\"}]}";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
cout << canonicalRequest << endl;
cout << "-----------------------" << endl;

// ************* 步骤 2：拼接待签名字符串 *************
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalReques
t;
cout << stringToSign << endl;
cout << "-----------------------" << endl;

// ************* 步骤 3：计算签名 ***************
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;
cout << "-----------------------" << endl;

// ************* 步骤 4：拼接 Authorization *************
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;
cout << "-----------------------" << endl;

string headers = "curl -X POST https://" + host + "\n"
```

```
+ " -H \"Authorization: " + authorization + "\n"
+ " -H \"Content-Type: application/json; charset=utf-8\"" + "\n"
+ " -H \"Host: " + host + "\n"
+ " -H \"X-TC-Action: " + action + "\n"
+ " -H \"X-TC-Timestamp: " + RequestTimestamp + "\n"
+ " -H \"X-TC-Version: " + version + "\n"
+ " -H \"X-TC-Region: " + region + "\n"
+ " -d '" + payload;
cout << headers << endl;
return 0;
};
```

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不相符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2020-08-11 08:13:15

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 API Explorer 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 Python、Java、PHP、Go、NodeJS、.NET。

腾讯云 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 腾讯云管理中心控制台。
2. 前往 云 API 密钥 的控制台页面
3. 在 云 API 密钥 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：每个账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: `AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******`
- SecretKey: `Gu5t9xGARNpq86cd98joQYCN3*******`

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	`AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******`
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20

参数名称	中文	参数值
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' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******',
'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=AKI
Dz8krbsJ5yKBZQpn74WFkmLPx3*******&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.tencentcloudapi.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.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Regi
on=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&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 = 'Gu5t9xGARNpq86cd98joQYCN3*******';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&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：

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 zmmjn35mikh6pM3V7sUEuX4wyYM= ，最终得到的签名串请求参数（Signature）为：zmmjn35mikh6pM3V7sUEuX4wyYM%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：有些编程语言的网络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，推荐使用配套的腾讯云 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
- NodeJS
- .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：https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%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 TencentCloudAPIDemo {
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.tencentcloudapi.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.tencentcloudapi.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", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"); // 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), "Gu5t9xGARNpq86cd98joQYCN3*******", "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 = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

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.tencentcloudapi.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("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

import (
"bytes"
"crypto/hmac"
"crypto/sha1"
"encoding/base64"
"fmt"
"sort"
)

func main() {
secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secretKey := "Gu5t9xGARNpq86cd98joQYCN3*******"
params := map[string]string{
"Nonce": "11886",
"Timestamp": "1465185768",
"Region": "ap-guangzhou",
"SecretId": secretId,
"Version": "2017-03-12",
"Action": "DescribeInstances",
"InstanceIds.0": "ins-09dx96dg",
"Limit": "20",
"Offset": "0",
}

var buf bytes.Buffer
buf.WriteString("GET")
buf.WriteString("cvm.tencentcloudapi.com")
```

```
buf.WriteString("/")
buf.WriteString("?")

// sort keys by ascii asc order
keys := make([]string, 0, len(params))
for k, _ := range params {
keys = append(keys, k)
}
sort.Strings(keys)

for i := range keys {
k := keys[i]
buf.WriteString(k)
buf.WriteString("=")
buf.WriteString(params[k])
buf.WriteString("&")
}
buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*******";
$param["Nonce"] = 11886;//rand();
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceIds.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param as $key => $value ) {
$signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
```

```
// need to install and enable curl extension in php.ini
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
'Action' => 'DescribeInstances',
'InstanceIds.0' => 'ins-09dx96dg',
'Limit' => 20,
'Nonce' => 11886,
'Offset' => 0,
'Region' => 'ap-guangzhou',
'SecretId' => secret_id,
'Timestamp' => 1465185768, # Time.now.to_i
'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

版权所有：腾讯云计算（北京）有限责任公司

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
public static string Sign(string signKey, string secret)
{
string signRet = string.Empty;
using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
{
byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
signRet = Convert.ToBase64String(hash);
}
return signRet;
}
public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, strin
g requestHost, string requestPath)
{
string retStr = "";
retStr += requestMethod;
retStr += requestHost;
retStr += requestPath;
retStr += "?";
string v = "";
foreach (string key in requestParams.Keys)
{
v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
// 密钥参数
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25,此参数作为示例，以实际为准
// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
```

```
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);

param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
string sigOutParam = Sign(SECRET_KEY, sigInParam);
Console.WriteLine(sigOutParam);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
params['Signature'] = escape(params['Signature']);
const url_strParam = sort_params(params)
return "https://" + endpoint + "/?" + url_strParam.slice(1);
}

function formatSignString(reqMethod, endpoint, path, strParam){
let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
return strSign;
}
function sha1(secretKey, strsign){
let signMethodMap = {'HmacSHA1': "sha1"};
let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
let strParam = "";
let keys = Object.keys(params);
keys.sort();
for (let k in keys) {
//k = k.replace(/_/g, '.');
strParam += ("&" + keys[k] + "=" + params[keys[k]]);
}
return strParam
}

function main(){
```

```
// 密钥参数
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******"

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48，此参数作为示例，以实际为准
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886 // 随机正整数
//const nonce = Math.round(Math.random() * 65535)

let params = {};
params['Action'] = Action;
params['InstanceIds.0'] = 'ins-09dx96dg';
params['Limit'] = 20;
params['Offset'] = 0;
params['Nonce'] = Nonce;
params['Region'] = Region;
params['SecretId'] = SECRET_ID;
params['Timestamp'] = Timestamp;
params['Version'] = Version;

// 1. 对参数排序,并拼接请求字符串
strParam = sort_params(params)

// 2. 拼接签名原文字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)

// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
// console.log(req_url)
}
main()
```

返回结果

最近更新时间：2020-06-12 08:11:37

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是200，而不是401。

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

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

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	HTTPS 请求方法错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

版权所有：腾讯云计算（北京）有限责任公司

信息查询相关接口
创建实例询价

最近更新时间：2020-11-30 08:17:46

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

创建实例询价

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi ）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：InquiryPriceCreateInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
TimeUnit	是	String	购买实例的时间单位。取值范围： • s：表示秒。PayMode取值为0时，TimeUnit只能取值为s。 • m：表示月份。PayMode取值为1时，TimeUnit只能取值为m。
TimeSpan	是	Integer	购买实例的时长。结合TimeUnit一起使用。 • TimeUnit为s时，该参数只能填写3600，表示按量计费实例。 • TimeUnit为m时，该参数填写的数字表示包年包月实例的购买时长，如1表示购买一个月
ResourceSpec	是	NewResourceSpec	询价的节点规格。
Currency	是	String	货币种类。取值范围： • CNY：表示人民币。
PayMode	是	Integer	实例计费模式。取值范围： • 0：表示按量计费。 • 1：表示包年包月。
SupportHA	是	Integer	是否开启节点高可用。取值范围： • 0：表示不开启节点高可用。 • 1：表示开启节点高可用。

参数名称	必选	类型	描述
Software.N	是	Array of String	部署的组件列表。不同的EMR产品ID（ProductId：具体含义参考入参ProductId字段）需要选择不同的必选组件： • ProductId为1的时候，必选组件包括：hadoop-2.7.3、knox-1.2.0、zookeeper-3.4.9 • ProductId为2的时候，必选组件包括：hadoop-2.7.3、knox-1.2.0、zookeeper-3.4.9 • ProductId为4的时候，必选组件包括：hadoop-2.8.4、knox-1.2.0、zookeeper-3.4.9 • ProductId为7的时候，必选组件包括：hadoop-3.1.2、knox-1.2.0、zookeeper-3.4.9
Placement	是	Placement	实例所在的位置。通过该参数可以指定实例所属可用区，所属项目等属性。
VPCSettings	是	VPCSettings	私有网络相关信息配置。通过该参数可以指定私有网络的ID，子网ID等信息。
MetaType	否	String	hive共享元数据库类型。取值范围： • EMR_NEW_META：表示集群默认创建 • EMR_EXIT_METE：表示集群使用指定EMR-MetaDB。 • USER_CUSTOM_META：表示集群使用自定义MetaDB。
UnifyMetaInstanceId	否	String	EMR-MetaDB实例
MetaDBInfo	否	CustomMetaInfo	自定义MetaDB信息
ProductId	否	Integer	产品ID，不同产品ID表示不同的EMR产品版本。取值范围： • 1：表示EMR-V1.3.1。 • 2：表示EMR-V2.0.1。 • 4：表示EMR-V2.1.0。 • 7：表示EMR-V3.0.0。

3. 输出参数

参数名称	类型	描述
OriginalCost	Float	原价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
DiscountCost	Float	折扣价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
TimeUnit	String	购买实例的时间单位。取值范围： • s：表示秒。 • m：表示月份。 注意：此字段可能返回 null，表示取不到有效值。
TimeSpan	Integer	购买实例的时长。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建询价

输入示例

```
https://emr.tencentcloudapi.com/?Action=InquiryPriceCreateInstance
&SupportHA=0
&PayMode=0
&Placement.Zone=ap-guangzhou-3
&Placement.ProjectId=0
&Software.0=hadoop-2.7.3
&Software.1=zookeeper-3.4.9
&Software.2=hive-2.3.2
&Software.3=knox-1.2.0
&ResourceSpec.MasterResourceSpec.MemSize=16384
&ResourceSpec.MasterResourceSpec.Cpu=4
&ResourceSpec.MasterResourceSpec.DiskSize=100
&ResourceSpec.MasterResourceSpec.DiskType=CLOUD_PREMIUM
&ResourceSpec.MasterResourceSpec.Spec=CVM.S3
&ResourceSpec.MasterResourceSpec.RootSize=100
&ResourceSpec.MasterResourceSpec.StorageType=5
&ResourceSpec.CoreResourceSpec.MemSize=16384
&ResourceSpec.CoreResourceSpec.Cpu=4
&ResourceSpec.CoreResourceSpec.DiskSize=100
&ResourceSpec.CoreResourceSpec.DiskType=CLOUD_PREMIUM
&ResourceSpec.CoreResourceSpec.Spec=CVM.S3
&ResourceSpec.CoreResourceSpec.RootSize=100
&ResourceSpec.CoreResourceSpec.StorageType=5
&ResourceSpec.MasterCount=1
&ResourceSpec.CoreCount=2
&VPCSettings.VpcId=vpc-ezt5qmqz
&VPCSettings.SubnetId=subnet-jhgsahx0
&TimeSpan=3600
&TimeUnit=s
&Currency=CNY
&ProductId=2
&<公共请求参数>
```

输出示例

```
{
"Response": {
"DiscountCost": 5.53,
"OriginalCost": 7.87,
"RequestId": "863e0be5-ab86-4daa-84f2-f84953f18aec",
"TimeSpan": 3600,
"TimeUnit": "s"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation	操作失败。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InvalidParameter	参数错误。
InvalidParameter.InvalidSoftWareName	软件名无效。
InvalidParameter.InvalidZone	无效的可用区。
InvalidParameter.SoftwareNotInProduct	存在无效的产品组件。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceInsufficient.DiskInsufficient	硬盘规格不满足。
ResourceInsufficient.InstanceInsufficient	不支持或售罄的节点规格。
ResourceNotFound.SubnetNotFound	找不到对应的子网。
ResourcesSoldOut.	资源售罄。
ResourcesSoldOut.CbsSoldOut	CBS资源售罄。
ResourcesSoldOut.CvmSoldOut	云服务器已售罄。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

扩容询价

最近更新时间：2020-11-30 08:17:46

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

扩容询价. 当扩容时候，请通过该接口查询价格。

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer

点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：InquiryPriceScaleOutInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
TimeUnit	是	String	扩容的时间单位。取值范围： • s：表示秒。PayMode取值为0时，TimeUnit只能取值为s。 • m：表示月份。PayMode取值为1时，TimeUnit只能取值为m。
TimeSpan	是	Integer	扩容的时长。结合TimeUnit一起使用。 • TimeUnit为s时，该参数只能填写3600，表示按量计费实例。 • TimeUnit为m时，该参数填写的数字表示包年包月实例的购买时长，如1表示购买一个月
ZoneId	是	Integer	实例所属的可用区ID，例如100003。该参数可以通过调用 DescribeZones 的返回值中的ZoneId字段来获取。
PayMode	是	Integer	实例计费模式。取值范围： • 0：表示按量计费。 • 1：表示包年包月。
InstanceId	是	String	实例ID。
CoreCount	是	Integer	扩容的Core节点数量。
TaskCount	是	Integer	扩容的Task节点数量。
Currency	是	String	货币种类。取值范围： • CNY：表示人民币。
RouterCount	否	Integer	扩容的Router节点数量。

3. 输出参数

参数名称	类型	描述
OriginalCost	String	原价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
DiscountCost	String	折扣价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
Unit	String	扩容的时间单位。取值范围： • s：表示秒。 • m：表示月份。 注意：此字段可能返回 null，表示取不到有效值。
PriceSpec	PriceResource	询价的节点规格。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 扩容询价

输入示例

```
https://emr.tencentcloudapi.com/?Action=InquiryPriceScaleOutInstance
&TimeUnit=s
&TimeSpan=3600
&ZoneId=100003
&PayMode=0
&InstanceId=emr-3ida6zmi
&CoreCount=1
&TaskCount=0
&Currency=CNY
&RouterCount=0
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "04daa603-e1e7-4243-b25d-31e6a6736528"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.CheckQuotaErr	cvm或cbs资源不够或软件不合法。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InvalidParameter.InvaildCoreCount	core节点的数量不能超过20。
InvalidParameter.InvalidAppId	无效参数，AppId。

错误码	描述
InvalidParameter.InvalidCountNum	同一请求只能扩容Task或者Core节点。
InvalidParameter.InvalidModifySpec	变配规格无效。
InvalidParameter.InvalidPaymode	无效的付费类型。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。
ResourceNotFound.InstanceNotFound	无法找到该实例。
ResourceUnavailable.ResourceSpec_NotDefaultSpec	当前资源规格不存在默认规格。

版权所有：腾讯云计算（北京）有限责任公司

查询EMR实例

最近更新时间：2020-11-30 08:17:46

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

查询EMR实例

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：DescribeInstances。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
DisplayStrategy	是	String	集群筛选策略。取值范围： • clusterList：表示查询除了已销毁集群之外的集群列表。 • monitorManage：表示查询除了已销毁、创建中以及创建失败的集群之外的集群列表。 • cloudHardwareManage/componentManage：目前这两个取值为预留取值，暂时和monitorManage表示同样的含义。
InstanceIds.N	否	Array of String	按照一个或者多个实例ID查询。实例ID形如: emr-xxxxxxxx 。(此参数的具体格式可参考API简介的 Ids.N 一节)。如果不填写实例ID，返回该APPID下所有实例列表。
Offset	否	Integer	页编号，默认值为0，表示第一页。
Limit	否	Integer	每页返回数量，默认值为10，最大值为100。
ProjectId	否	Integer	建议必填-1，表示拉取所有项目下的集群。 不填默认值为0，表示拉取默认项目下的集群。 实例所属项目ID。该参数可以通过调用 DescribeProject 的返回值中的 projectId 字段来获取。
OrderField	否	String	排序字段。取值范围： • clusterId：表示按照实例ID排序。 • addTime：表示按照实例创建时间排序。 • status：表示按照实例的状态码排序。
Asc	否	Integer	按照OrderField升序或者降序进行排序。取值范围： • 0：表示降序。 • 1：表示升序。 默认值为0。

3. 输出参数

参数名称	类型	描述
TotalCnt	Integer	符合条件的实例总数。
ClusterList	Array of ClusterInstancesInfo	EMR实例详细信息列表。 注意：此字段可能返回 null，表示取不到有效值。
TagKeys	Array of String	实例关联的标签键列表。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询实例详情

输入示例

```
https://emr.tencentcloudapi.com/?Action=DescribeInstances
&Offset=0
&Limit=10
&ProjectId=0
&OrderField=clusterid
&Asc=0
&DisplayStrategy=clusterList
&InstanceIds.0=emr-p9f700x8
&<公共请求参数>
```

输出示例

```
{
"Response": {
"ClusterList": [
{
"AddTime": "2019-09-16 16:48:01",
"AlarmInfo": "",
"AppId": 251008830,
"ChargeType": 1,
"ClusterId": "emr-p9f700x8",
"ClusterName": "beckwu_包年勿删",
"Config": {
"ChargeType": 1,
"ComNodeSize": 0,
"ComResource": {
"Cpu": 0,
"DiskSize": 0,
"DiskType": "",
"MemSize": 0,
"RootSize": 0,
"Spec": "",
"SpecName": "",
```

```
"StorageType": 0
},
"CoreNodeSize": 2,
"CoreResource": {
"Cpu": 2,
"DiskSize": 100,
"DiskType": "CLOUD_BASIC",
"MemSize": 8192,
"RootSize": 0,
"Spec": "CVM.S2",
"SpecName": "EMR标准型S2",
"StorageType": 2
},
"MasterNodeSize": 1,
"MasterResource": {
"Cpu": 2,
"DiskSize": 100,
"DiskType": "CLOUD_BASIC",
"MemSize": 8192,
"RootSize": 0,
"Spec": "CVM.S2",
"SpecName": "EMR标准型S2",
"StorageType": 2
},
"OnCos": false,
"SoftInfo": [
"zookeeper-3.4.9",
"hadoop-3.1.2",
"knox-1.2.0",
"sys-1.0"
],
"TaskNodeSize": 1,
"TaskResource": {
"Cpu": 2,
"DiskSize": 100,
"DiskType": "CLOUD_BASIC",
"MemSize": 8192,
"RootSize": 0,
"Spec": "CVM.S2",
"SpecName": "EMR标准型S2",
"StorageType": 2
}
},
"EmrVersion": "EMR-V3.0.0",
"Ftitle": "集群运行中",
"Id": 19541,
"IsTradeCluster": 0,
"MasterIp": "--",
"ProjectId": 0,
"RegionId": 1,
"ResourceOrderId": 0,
```

```
"RunTime": "0天2小时48分钟55秒",
"Status": 2,
"SubnetId": 1230738,
"TradeVersion": 1,
"Uin": "1875765535",
"VpcId": 78518,
"ZoneId": 100002
}
],
"RequestId": "4f337873-6fea-4338-9715-24f539b60949",
"TotalCnt": 1
}
}
```


5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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	内部错误。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。

错误码	描述
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InternalError.WoodServerError	Woodpecker server调用出错。
InvalidParameter	参数错误。
InvalidParameter.DisplayStrategyNotMatch	展示策略错误。
InvalidParameter.InvalidClusterId	无效参数，ClusterId。
InvalidParameter.OrderFieldNotMatch	排序字段错误。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.HardwareInfoNotFound	无法找到硬件信息。
ResourceNotFound.InstanceNotFound	无法找到该实例。

版权所有：腾讯云计算（北京）有限责任公司

变配询价

最近更新时间：2020-11-30 08:17:45

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

变配询价

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：InquiryPriceUpdateInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
TimeUnit	是	String	变配的时间单位。取值范围： • s：表示秒。PayMode取值为0时，TimeUnit只能取值为s。 • m：表示月份。PayMode取值为1时，TimeUnit只能取值为m。
TimeSpan	是	Integer	变配的时长。结合TimeUnit一起使用。 • TimeUnit为s时，该参数只能填写3600，表示按量计费实例。 • TimeUnit为m时，该参数填写的数字表示包年包月实例的购买时长，如1表示购买一个月
UpdateSpec	是	UpdateInstanceSettings	节点变配的目标配置。
PayMode	是	Integer	实例计费模式。取值范围： • 0：表示按量计费。 • 1：表示包年包月。
Placement	是	Placement	实例所在的位置。通过该参数可以指定实例所属可用区，所属项目等属性。
Currency	否	String	货币种类。取值范围： • CNY：表示人民币。

3. 输出参数

参数名称	类型	描述
OriginalCost	Float	原价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。

参数名称	类型	描述
DiscountCost	Float	折扣价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
TimeUnit	String	变配的时间单位。取值范围： • s：表示秒。 • m：表示月份。 注意：此字段可能返回 null，表示取不到有效值。
TimeSpan	Integer	变配的时长。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 变配询价

输入示例

```
https://emr.tencentcloudapi.com/?Action=InquiryPriceUpdateInstance
&TimeUnit=s
&TimeSpan=3600
&Placement.Zone=100003
&Currency=CNY
&Placement.ProjectId=0
&UpdateSpec.Memory=16
&UpdateSpec.CPUCores=8
&UpdateSpec.ResourceId=emr-vm-a0xxx9on
&PayMode=0
&<公共请求参数>
```

输出示例

```
{
"Response": {
"DiscountCost": 2.01,
"OriginalCost": 3.04,
"RequestId": "95eb9120-0883-407c-aa5a-43b4e2c250d1",
"TimeSpan": 3600,
"TimeUnit": "s"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.CheckQuotaErr	cvm或cbs资源不够或软件不合法。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidClusterId	无效参数，ClusterId。

错误码	描述
InvalidParameter.InvalidModifySpec	变配规格无效。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。
InvalidParameter.ZoneResourceNotMatch	可用区与资源不匹配。
ResourceNotFound.InstanceNotFound	无法找到该实例。
ResourceNotFound.ResourceNotFound	资源未找到。

续费询价

最近更新时间：2020-11-30 08:17:46

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

续费询价。

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi ）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：InquiryPriceRenewInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
TimeSpan	是	Integer	实例续费的时长。需要结合TimeUnit一起使用。1表示续费1一个月
ResourceIds.N	是	Array of String	待续费节点的资源ID列表。资源ID形如：emr-vm-xxxxxxxx。有效的资源ID可通过登录控制台查询。
Placement	是	Placement	实例所在的位置。通过该参数可以指定实例所属可用区，所属项目等属性。
PayMode	是	Integer	实例计费模式。此处只支持取值为1，表示包年包月。
TimeUnit	否	String	实例续费的时间单位。取值范围： • m：表示月份。
Currency	否	String	货币种类。取值范围： • CNY：表示人民币。

3. 输出参数

参数名称	类型	描述
OriginalCost	Float	原价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。
DiscountCost	Float	折扣价，单位为元。 注意：此字段可能返回 null，表示取不到有效值。

参数名称	类型	描述
TimeUnit	String	实例续费的时间单位。取值范围： • m：表示月份。 注意：此字段可能返回 null，表示取不到有效值。
TimeSpan	Integer	实例续费的时长。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 续费询价示例

输入示例

```
https://emr.tencentcloudapi.com/?Action=InquiryPriceRenewInstance
&TimeUnit=m
&TimeSpan=1
&Currency=CNY
&PayMode=1
&ResourceIds.0=emr-vm-jv1s4zas
&Placement.ProjectId=0
&Placement.Zone=ap-guangzhou-4
&<公共请求参数>
```

输出示例

```
{
"Response": {
"DiscountCost": 596.54,
"OriginalCost": 898.9,
"RequestId": "223c838e-ce27-4adf-9a41-89661fe7ad21",
"TimeSpan": 1,
"TimeUnit": "m"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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	内部错误。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.CheckQuotaErr	cvm或cbs资源不够或软件不合法。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InvalidParameter.InvalidResourceIds	资源ID无效。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。
InvalidParameter.ProjectResourceNotMatch	项目与资源不匹配。
ResourceNotFound.InstanceNotFound	无法找到该实例。

错误码	描述
ResourceNotFound.ResourceNotFound	资源未找到。

版权所有：腾讯云计算（北京）有限责任公司

查询硬件节点信息

最近更新时间：2020-11-30 08:17:46

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

查询硬件节点信息

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：DescribeClusterNodes。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
InstanceId	是	String	集群实例ID,实例ID形如: emr-xxxxxxxx
NodeFlag	是	String	节点标识，取值为： • all：表示获取全部类型节点，cdb信息除外。 • master：表示获取master节点信息。 • core：表示获取core节点信息。 • task：表示获取task节点信息。 • common：表示获取common节点信息。 • router：表示获取router节点信息。 • db：表示获取正常状态的cdb信息。 • recyle：表示获取回收站隔离中的节点信息，包括cdb信息。 • renew：表示获取所有待续费的节点信息，包括cdb信息，自动续费节点不会返回。 注意：现在只支持以上取值，输入其他值会导致错误。
Offset	否	Integer	页编号，默认值为0，表示第一页。
Limit	否	Integer	每页返回数量，默认值为100，最大值为100。
HardwareResourceType	否	String	资源类型:支持all/host/pod，默认为all
SearchFields.N	否	Array of SearchItem	支持搜索的字段

3. 输出参数

参数名称	类型	描述
TotalCnt	Integer	查询到的节点总数
NodeList	Array of NodeHardwareInfo	节点详细信息列表 注意：此字段可能返回 null，表示取不到有效值。
TagKeys	Array of String	用户所有的标签键列表 注意：此字段可能返回 null，表示取不到有效值。
HardwareResourceTypeList	Array of String	资源类型列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取硬件节点信息

通过emr集群ID获取硬件节点信息

输入示例

```
https://emr.tencentcloudapi.com/?Action=DescribeClusterNodes
&InstanceId=emr-6deluvd4
&NodeFlag=all
&Offset=0
&Limit=10
&<公共请求参数>
```

输出示例


```
{
"Response": {
"NodeList": [
{
"AppId": 251008830,
"ApplyTime": "2020-02-24 20:31:06",
"CdbIp": "",
"CdbNodeInfo": null,
"CdbPort": 0,
"ChargeType": 0,
"CpuNum": 8,
"Destroyable": 0,
"DeviceClass": "VSELF_2",
"DiskSize": "100.00 GB",
"EmrResourceId": "emr-vm-6xyf2cb2",
"ExpireTime": "0000-00-00 00:00:00",
"Flag": 1,
"FreeTime": "0000-00-00 00:00:00",
"HwDiskSize": 107374182400,
"HwDiskSizeDesc": "100.00 GB",
"HwMemSize": 17179869184,
```

```
"HwMemSizeDesc": "16GB",
"Ip": "10.0.0.76",
"IsAutoRenew": 0,
"MCMultiDisk": [
{
"Count": 1,
"Type": 5,
"Volume": 107374182400
}
],
"MemDesc": "16GB",
"MemSize": 17179869184,
"Mutable": 1,
"NameTag": "master.0",
"OrderNo": "ins-20224atg",
"RegionId": 1,
"RootSize": 0,
"SerialNo": "83d977e5-fa68-4051-875e-ad30ff42534f",
"Services": "Zookeeper,NameNode,ResourceManager,JobHistoryServer,HMaster,HbaseThrift,HiveServer2,HiveMetaStore,Hiv
eWebHcat,Spark,SparkJobHistoryServer,Presto-Coordinator,knox",
"Spec": "CVM.S2",
"StorageType": 5,
"Tags": [],
"WanIp": "--",
"ZoneId": 100002
},
{
"AppId": 251008830,
"ApplyTime": "2020-02-24 20:31:07",
"CdbIp": "",
"CdbNodeInfo": null,
"CdbPort": 0,
"ChargeType": 0,
"CpuNum": 8,
"Destroyable": 0,
"DeviceClass": "VSELF_2",
"DiskSize": "100.00 GB",
"EmrResourceId": "emr-vm-cinlo2wc",
"ExpireTime": "0000-00-00 00:00:00",
"Flag": 2,
"FreeTime": "0000-00-00 00:00:00",
"HwDiskSize": 107374182400,
"HwDiskSizeDesc": "100.00 GB",
"HwMemSize": 17179869184,
"HwMemSizeDesc": "16GB",
"Ip": "10.0.0.33",
"IsAutoRenew": 0,
"MCMultiDisk": [
{
"Count": 1,
"Type": 5,
```

```
"Volume": 107374182400
}
],
"MemDesc": "16GB",
"MemSize": 17179869184,
"Mutable": 1,
"NameTag": "core.0",
"OrderNo": "ins-20224gpk",
"RegionId": 1,
"RootSize": 0,
"SerialNo": "8ded940b-a579-4c81-be75-3aaf62137337",
"Services": "DataNode,NodeManager,RegionServer,Presto-Worker",
"Spec": "CVM.S2",
"StorageType": 5,
"Tags": [],
"WanIp": "",
"ZoneId": 100002
},
{
"AppId": 251008830,
"ApplyTime": "2020-02-24 20:31:08",
"CdbIp": "",
"CdbNodeInfo": null,
"CdbPort": 0,
"ChargeType": 0,
"CpuNum": 8,
"Destroyable": 0,
"DeviceClass": "VSELF_2",
"DiskSize": "100.00 GB",
"EmrResourceId": "emr-vm-b32qad6s",
"ExpireTime": "0000-00-00 00:00:00",
"Flag": 2,
"FreeTime": "0000-00-00 00:00:00",
"HwDiskSize": 107374182400,
"HwDiskSizeDesc": "100.00 GB",
"HwMemSize": 17179869184,
"HwMemSizeDesc": "16GB",
"Ip": "10.0.0.111",
"IsAutoRenew": 0,
"MCMultiDisk": [
{
"Count": 1,
"Type": 5,
"Volume": 107374182400
}
],
"MemDesc": "16GB",
"MemSize": 17179869184,
"Mutable": 1,
"NameTag": "core.1",
"OrderNo": "ins-202241if",
```

```
"RegionId": 1,
"RootSize": 0,
"SerialNo": "c045bcd7-571a-4c64-b0a5-9024c94d5c15",
"Services": "DataNode,NodeManager,RegionServer,Presto-Worker",
"Spec": "CVM.S2",
"StorageType": 5,
"Tags": [],
"WanIp": "",
"ZoneId": 100002
}
],
"RequestId": "bb22bafb-d2a4-4a02-879f-6ccf54a27892",
"TagKeys": [
"测试一下",
"alex_test",
"beckwuxingjia",
"ghghghg",
"tag_auth_test",
"test",
"beckwu",
"emr",
"lg",
"bk"
],
"TotalCnt": 3
}
}
```


5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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	内部错误。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InternalError.WoodServerError	Woodpecker server调用出错。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidInstanceName	无效的集群名称。
InvalidParameter.InvalidNodeType	无效的NodeType。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.InstanceNotFound	无法找到该实例。
UnsupportedOperation	操作不支持。

集群生命周期相关接口
销毁EMR实例

最近更新时间：2020-11-30 08:17:45

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

销毁EMR实例。此接口仅支持弹性MapReduce正式计费版本。

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi ）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：TerminateInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
InstanceId	是	String	实例ID。
ResourceIds.N	否	Array of String	销毁节点ID。该参数为预留参数，用户无需配置。

3. 输出参数

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

4. 示例

示例1 销毁实例

销毁整个集群

输入示例

```
https://emr.tencentcloudapi.com/?Action=TerminateInstance
&InstanceId=emr-4slr7ad7
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "4d701c1e-8507-47e1-8c69-a8f06a236f24"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidClusterId	无效参数，ClusterId。
ResourceInUse.InstanceInProcess	实例在流程中。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.InstanceNotFound	无法找到该实例。

创建EMR实例

最近更新时间：2020-11-30 08:17:45

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

创建EMR实例

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer

点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：CreateInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
ProductId	是	Integer	产品ID，不同产品ID表示不同的EMR产品版本。取值范围： • 1：表示EMR-V1.3.1。 • 2：表示EMR-V2.0.1。 • 4：表示EMR-V2.1.0。 • 7：表示EMR-V3.0.0。
VPCSettings	是	VPCSettings	私有网络相关信息配置。通过该参数可以指定私有网络的ID，子网ID等信息。
Software.N	是	Array of String	部署的组件列表。不同的EMR产品ID（ProductId：具体含义参考入参ProductId字段）需要选择不同的必选组件： • ProductId为1的时候，必选组件包括：hadoop-2.7.3、knox-1.2.0、zookeeper-3.4.9 • ProductId为2的时候，必选组件包括：hadoop-2.7.3、knox-1.2.0、zookeeper-3.4.9 • ProductId为4的时候，必选组件包括：hadoop-2.8.4、knox-1.2.0、zookeeper-3.4.9 • ProductId为7的时候，必选组件包括：hadoop-3.1.2、knox-1.2.0、zookeeper-3.4.9
ResourceSpec	是	NewResourceSpec	节点资源的规格。
SupportHA	是	Integer	是否开启节点高可用。取值范围： • 0：表示不开启节点高可用。 • 1：表示开启节点高可用。

参数名称	必选	类型	描述
InstanceName	是	String	实例名称。 • 长度限制为6-36个字符。 • 只允许包含中文、字母、数字、-、_。
PayMode	是	Integer	实例计费模式。取值范围： • 0：表示按量计费。 • 1：表示包年包月。
Placement	是	Placement	实例所在的位置。通过该参数可以指定实例所属可用区，所属项目等属性。
TimeSpan	是	Integer	购买实例的时长。结合TimeUnit一起使用。 • TimeUnit为s时，该参数只能填写3600，表示按量计费实例。 • TimeUnit为m时，该参数填写的数字表示包年包月实例的购买时长，如1表示购买一个月
TimeUnit	是	String	购买实例的时间单位。取值范围： • s：表示秒。PayMode取值为0时，TimeUnit只能取值为s。 • m：表示月份。PayMode取值为1时，TimeUnit只能取值为m。
LoginSettings	是	LoginSettings	实例登录设置。通过该参数可以设置所购买节点的登录方式密码或者密钥。 • 设置密钥时，密码仅用于组件原生WebUI快捷入口登录。 • 未设置密钥时，密码用于登录所购节点以及组件原生WebUI快捷入口登录。
COSSettings	否	COSSettings	开启COS访问需要设置的参数。
SgId	否	String	实例所属安全组的ID，形如sg-xxxxxxxx。该参数可以通过调用DescribeSecurityGroups 的返回值中的SecurityGroupId字段来获取。
PreExecutedFileSettings.N	否	Array of PreExecuteFileSettings	引导操作脚本设置。
AutoRenew	否	Integer	包年包月实例是否自动续费。取值范围： • 0：表示不自动续费。 • 1：表示自动续费。
ClientToken	否	String	客户端Token。
NeedMasterWan	否	String	是否开启集群Master节点公网。取值范围： • NEED_MASTER_WAN：表示开启集群Master节点公网。 • NOT_NEED_MASTER_WAN：表示不开启。 默认开启集群Master节点公网。
RemoteLoginAtCreate	否	Integer	是否需要开启外网远程登录，即22号端口。在SgId不为空时，该参数无效。
CheckSecurity	否	Integer	是否开启安全集群。0表示不开启，非0表示开启。
ExtendFsField	否	String	访问外部文件系统。
Tags.N	否	Array of Tag	标签描述列表。通过指定该参数可以同时绑定标签到相应的实例。
DisasterRecoverGroupIds.N	否	Array of String	分散置放群组ID列表，当前只支持指定一个。

参数名称	必选	类型	描述
CbsEncrypt	否	Integer	集群维度CBS加密盘，默认0表示不加密，1表示加密
MetaType	否	String	hive共享元数据库类型。取值范围： • EMR_NEW_META：表示集群默认创建 • EMR_EXIT_METE：表示集群使用指定EMR-MetaDB。 • USER_CUSTOM_META：表示集群使用自定义MetaDB。
UnifyMetaInstanceId	否	String	EMR-MetaDB实例
MetaDBInfo	否	CustomMetaInfo	自定义MetaDB信息
ApplicationRole	否	String	自定义应用角色。

3. 输出参数

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

4. 示例

示例1 创建实例

输入示例

```
https://emr.tencentcloudapi.com/?Action=CreateInstance
&ProductId=4
&SupportHA=0
&InstanceName=emr测试
&PayMode=0
&Placement.Zone=ap-guangzhou-3
&Placement.ProjectId=0
&AutoRenew=0
&Software.0=hadoop-2.8.4
&Software.1=zookeeper-3.4.9
&Software.2=knox-1.2.0
&ResourceSpec.MasterResourceSpec.MemSize=8192
&ResourceSpec.MasterResourceSpec.Cpu=4
&ResourceSpec.MasterResourceSpec.DiskSize=100
&ResourceSpec.MasterResourceSpec.DiskType=CLOUD_PREMIUM
&ResourceSpec.MasterResourceSpec.Spec=CVM.S2
&ResourceSpec.MasterResourceSpec.RootSize=100
&ResourceSpec.MasterResourceSpec.StorageType=5
&ResourceSpec.CoreResourceSpec.MemSize=8192
&ResourceSpec.CoreResourceSpec.Cpu=4
&ResourceSpec.CoreResourceSpec.DiskSize=100
&ResourceSpec.CoreResourceSpec.DiskType=CLOUD_PREMIUM
&ResourceSpec.CoreResourceSpec.Spec=CVM.S2
&ResourceSpec.CoreResourceSpec.RootSize=100
&ResourceSpec.CoreResourceSpec.StorageType=5
&ResourceSpec.MasterCount=1
```

```
&ResourceSpec.CoreCount=2
&VPCSettings.VpcId=vpc-ezt5qmqz
&VPCSettings.SubnetId=subnet-jhgsahx0
&LoginSettings.Password=tencent@cloud123
&TimeSpan=3600
&TimeUnit=s
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "d830face-6587-4263-8ab0-56bda265787d"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述
FailedOperation	操作失败。
FailedOperation.DuplicateOrderNotAllowed	重复的订单，请检查emr控制台。
InternalError	内部错误。

错误码	描述
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.CheckQuotaErr	cvm或cbs资源不够或软件不合法。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InvalidParameter	参数错误。
InvalidParameter.IncorrectCommonCount	Common节点数量无效。
InvalidParameter.IncorrectMasterCount	Master节点数量无效。
InvalidParameter.InvaildCoreCount	core节点的数量不能超过20。
InvalidParameter.InvalidAutoRenew	无效的自动续费标识。
InvalidParameter.InvalidClientToken	无效的ClientToken。
InvalidParameter.InvalidComponent	无效的组件。
InvalidParameter.InvalidCoreCount	Core节点数量无效。
InvalidParameter.InvalidInstanceName	无效的集群名称。
InvalidParameter.InvalidLoginSetting	无效的登录设置。
InvalidParameter.InvalidPassword	无效密码。
InvalidParameter.InvalidPaymode	无效的付费类型。
InvalidParameter.InvalidPreExecutedFile	无效的引导操作脚本。
InvalidParameter.InvalidProductId	无效的产品ID。
InvalidParameter.InvalidProjectId	无效的项目ID。

错误码	描述
InvalidParameter.InvalidResourceSpec	无效的资源规格。
InvalidParameter.InvalidSercurityGrpupId	无效的安全组ID。
InvalidParameter.InvalidServiceName	服务名无效。
InvalidParameter.InvalidSoftInfo	无效的SoftInfo。
InvalidParameter.InvalidSoftWareVersion	软件版本无效。
InvalidParameter.InvalidSubnetId	无效的子网ID。
InvalidParameter.InvalidSupportHA	无效的高可用参数。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。
InvalidParameter.InvalidVpcId	无效的私有网络ID。
InvalidParameter.InvalidZone	无效的可用区。
InvalidParameter.PayModeResourceNotMatch	付费模式与资源不匹配。
InvalidParameter.SoftwareNotInProduct	存在无效的产品组件。
InvalidParameter.UngrantedPolicy	策略为授权。
InvalidParameter.UngrantedRole	角色未授权。
InvalidParameter.ZoneResourceNotMatch	可用区与资源不匹配。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceInsufficient.DiskInsufficient	硬盘规格不满足。
ResourceInsufficient.InstanceInsufficient	不支持或售罄的节点规格。
ResourceNotFound.TagsNotFound	没有查找到指定标签。
ResourcesSoldOut.	资源售罄。
ResourcesSoldOut.CbsSoldOut	CBS资源售罄。
ResourcesSoldOut.CvmSoldOut	云服务器已售罄。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

扩缩容相关接口
缩容Task节点

最近更新时间：2020-11-30 08:17:45

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

缩容Task节点

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi ）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：TerminateTasks。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
InstanceId	是	String	实例ID。
ResourceIds.N	是	Array of String	待销毁节点的资源ID列表。资源ID形如：emr-vm-xxxxxxxx。有效的资源ID可通过登录控制台查询。

3. 输出参数

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

4. 示例

示例1 销毁节点

销毁TASK节点

输入示例

```
https://emr.tencentcloudapi.com/?Action=TerminateTasks
&InstanceId=emr-4slr7ad7
```

```
&ResourceIds.0=emr-vm-xxx33tg
&<公共请求参数>
```

输出示例

```
{
"Response": {
"RequestId": "4d701c1e-8507-47e1-8c69-a8f06a236f24"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 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	内部错误。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。

版权所有：腾讯云计算（北京）有限责任公司

错误码	描述
InternalError.CdbError	内部服务调用异常。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InternalError.WoodServerError	Woodpecker server调用出错。
InvalidParameter	参数错误。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidInstanceName	无效的集群名称。
InvalidParameter.InvalidResourceIds	资源ID无效。
ResourceInUse.InstanceInProcess	实例在流程中。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.InstanceNotFound	无法找到该实例。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

版权所有：腾讯云计算（北京）有限责任公司

实例扩容

最近更新时间：2020-11-30 08:17:45

1. 接口描述

接口请求域名： emr.tencentcloudapi.com 。

实例扩容

默认接口请求频率限制：20次/秒。

注意：本接口支持金融区地域。由于金融区和非金融区是隔离不互通的，因此当公共参数 Region 为金融区地域（例如 ap-shanghai-fsi）时，需要同时指定带金融区地域的域名，最好和 Region 的地域保持一致，例如：emr.ap-shanghai-fsi.tencentcloudapi.com。

推荐使用 API Explorer　点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

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	类型	描述
Action	是	String	公共参数，本接口取值：ScaleOutInstance。
Version	是	String	公共参数，本接口取值：2019-01-03。
Region	是	String	公共参数，详见产品支持的 地域列表。
TimeUnit	是	String	扩容的时间单位。取值范围： • s：表示秒。PayMode取值为0时，TimeUnit只能取值为s。 • m：表示月份。PayMode取值为1时，TimeUnit只能取值为m。
TimeSpan	是	Integer	扩容的时长。结合TimeUnit一起使用。 • TimeUnit为s时，该参数只能填写3600，表示按量计费实例。 • TimeUnit为m时，该参数填写的数字表示包年包月实例的购买时长，如1表示购买一个月
InstanceId	是	String	实例ID。
PayMode	是	Integer	实例计费模式。取值范围： • 0：表示按量计费。 • 1：表示包年包月。
ClientToken	否	String	客户端Token。
PreExecutedFileSettings.N	否	Array of PreExecuteFileSettings	引导操作脚本设置。
TaskCount	否	Integer	扩容的Task节点数量。
CoreCount	否	Integer	扩容的Core节点数量。
UnNecessaryNodeList.N	否	Array of Integer	扩容时不需要安装的进程。

版权所有：腾讯云计算（北京）有限责任公司

参数名称	必选	类型	描述
RouterCount	否	Integer	扩容的Router节点数量。
SoftDeployInfo.N	否	Array of Integer	部署的服务。 • SoftDeployInfo和ServiceNodeInfo是同组参数，和UnNecessaryNodeList参数互斥。 • 建议使用SoftDeployInfo和ServiceNodeInfo组合。
ServiceNodeInfo.N	否	Array of Integer	启动的进程。
DisasterRecoverGroupIds.N	否	Array of String	分散置放群组ID列表，当前仅支持指定一个。
Tags.N	否	Array of Tag	扩容节点绑定标签列表。
HardwareResourceType	否	String	扩容所选资源类型，可选范围为"host","pod"，host为普通的CVM资源，Pod为TKE集群提供的资源
PodSpec	否	PodSpec	使用Pod资源扩容时，指定的Pod规格以及来源等信息
ClickHouseClusterName	否	String	使用clickhouse集群扩容时，选择的机器分组名称
ClickHouseClusterType	否	String	使用clickhouse集群扩容时，选择的机器分组类型。new为新增，old为选择旧分组
YarnNodeLabel	否	String	规则扩容指定 yarn node label

3. 输出参数

参数名称	类型	描述
InstanceId	String	实例ID。
DealNames	Array of String	订单号。 注意：此字段可能返回 null，表示取不到有效值。
ClientToken	String	客户端Token。 注意：此字段可能返回 null，表示取不到有效值。
FlowId	Integer	扩容流程ID。 注意：此字段可能返回 null，表示取不到有效值。
BillId	String	大订单号。 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 集群扩容

输入示例

```
https://emr.tencentcloudapi.com/?Action=ScaleOutInstance
&TimeUnit=s
&TimeSpan=3600
&CoreCount=1
&InstanceId=emr-5n3l5c83
```

```
&PayMode=0
&<公共请求参数>
```

输出示例

```
{
"Response": {
"BillId": "",
"ClientToken": "",
"DealNames": [
"20200309357833",
"20200309357834",
"20200309357835",
"20200309357836"
],
"FlowId": 0,
"InstanceId": "emr-5n3l5c83",
"RequestId": "f0f11d21-6d0d-4f73-9177-8ae4ec456068"
}
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go
- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET
- Tencent Cloud SDK 3.0 for C++

命令行工具

- Tencent Cloud CLI 3.0

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 公共错误码。

错误码	描述

错误码	描述
FailedOperation.DuplicateOrderNotAllowed	重复的订单，请检查emr控制台。
InternalError	内部错误。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TKEError	TKE调用出错。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InternalError.WoodServerError	Woodpecker server调用出错。
InvalidParameter	参数错误。
InvalidParameter.InvaildCoreCount	core节点的数量不能超过20。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidClientToken	无效的ClientToken。
InvalidParameter.InvalidClusterId	无效参数，ClusterId。
InvalidParameter.InvalidCount	扩容数量必须大于0。
InvalidParameter.InvalidCountNum	同一请求只能扩容Task或者Core节点。
InvalidParameter.InvalidServiceNodeInfo	参数ServiceNodeInfo无效或错误。
InvalidParameter.InvalidSoftDeployInfo	参数InvalidSoftDeployInfo无效或错误。
InvalidParameter.InvalidTaskCount	task的数量不能超过20。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。

错误码	描述
InvalidParameterValue.InvalidTkeInstance	无效的Tke集群ID，或Tke集群不符合条件。
ResourceInUse.InstanceInProcess	实例在流程中。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.InstanceNotFound	无法找到该实例。
ResourceNotFound.TKEPreconditionNotFound	tke集群前置组件未部署。
ResourceNotFound.TagsNotFound	没有查找到指定标签。
ResourcesSoldOut.	资源售罄。
ResourcesSoldOut.CbsSoldOut	CBS资源售罄。
ResourcesSoldOut.CvmSoldOut	云服务器已售罄。

数据结构

最近更新时间：2020-09-21 08:03:26

COSSettings

COS 相关配置

被如下接口引用：CreateInstance。

名称	类型	必选	描述
CosSecretId	String	是	COS SecretId
CosSecretKey	String	是	COS SecrectKey
LogOnCosPath	String	否	日志存储在COS上的路径

CdbInfo

出参

被如下接口引用：DescribeClusterNodes。

名称	类型	描述
InstanceName	String	数据库实例 注意：此字段可能返回 null，表示取不到有效值。
Ip	String	数据库IP 注意：此字段可能返回 null，表示取不到有效值。
Port	Integer	数据库端口 注意：此字段可能返回 null，表示取不到有效值。
MemSize	Integer	数据库内存规格 注意：此字段可能返回 null，表示取不到有效值。
Volume	Integer	数据库磁盘规格 注意：此字段可能返回 null，表示取不到有效值。
Service	String	服务标识 注意：此字段可能返回 null，表示取不到有效值。
ExpireTime	String	过期时间 注意：此字段可能返回 null，表示取不到有效值。
ApplyTime	String	申请时间 注意：此字段可能返回 null，表示取不到有效值。
PayType	Integer	付费类型 注意：此字段可能返回 null，表示取不到有效值。
ExpireFlag	Boolean	过期标识 注意：此字段可能返回 null，表示取不到有效值。
Status	Integer	数据库状态 注意：此字段可能返回 null，表示取不到有效值。
IsAutoRenew	Integer	续费标识 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
SerialNo	String	数据库字符串 注意：此字段可能返回 null，表示取不到有效值。
ZoneId	Integer	ZoneId 注意：此字段可能返回 null，表示取不到有效值。
RegionId	Integer	RegionId 注意：此字段可能返回 null，表示取不到有效值。

ClusterInstancesInfo

集群实例信息

被如下接口引用：DescribeInstances。

名称	类型	描述
Id	Integer	ID号 注意：此字段可能返回 null，表示取不到有效值。
ClusterId	String	集群ID 注意：此字段可能返回 null，表示取不到有效值。
Ftitle	String	标题 注意：此字段可能返回 null，表示取不到有效值。
ClusterName	String	集群名 注意：此字段可能返回 null，表示取不到有效值。
RegionId	Integer	地域ID 注意：此字段可能返回 null，表示取不到有效值。
ZoneId	Integer	地区ID 注意：此字段可能返回 null，表示取不到有效值。
AppId	Integer	用户APPID 注意：此字段可能返回 null，表示取不到有效值。
Uin	String	用户UIN 注意：此字段可能返回 null，表示取不到有效值。
ProjectId	Integer	项目Id 注意：此字段可能返回 null，表示取不到有效值。
VpcId	Integer	集群VPCID 注意：此字段可能返回 null，表示取不到有效值。
SubnetId	Integer	子网ID 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
Status	Integer	实例的状态码。取值范围： • 2：表示集群运行中。 • 3：表示集群创建中。 • 4：表示集群扩容中。 • 5：表示集群增加router节点中。 • 6：表示集群安装组件中。 • 7：表示集群执行命令中。 • 8：表示重启服务中。 • 9：表示进入维护中。 • 10：表示服务暂停中。 • 11：表示退出维护中。 • 12：表示退出暂停中。 • 13：表示配置下发中。 • 14：表示销毁集群中。 • 15：表示销毁core节点中。 • 16：销毁task节点中。 • 17：表示销毁router节点中。 • 18：表示更改webproxy密码中。 • 19：表示集群隔离中。 • 20：表示集群冲正中。 • 21：表示集群回收中。 • 22：表示变配等待中。 • 23：表示集群已隔离。 • 24：表示缩容节点中。 • 33：表示集群等待退费中。 • 34：表示集群已退费。 • 301：表示创建失败。 • 302：表示扩容失败。 注意：此字段可能返回 null，表示取不到有效值。
AddTime	String	添加时间 注意：此字段可能返回 null，表示取不到有效值。
RunTime	String	已经运行时间 注意：此字段可能返回 null，表示取不到有效值。
Config	EmrProductConfigOutter	集群产品配置信息 注意：此字段可能返回 null，表示取不到有效值。
MasterIp	String	主节点外网IP 注意：此字段可能返回 null，表示取不到有效值。
EmrVersion	String	EMR版本 注意：此字段可能返回 null，表示取不到有效值。
ChargeType	Integer	收费类型 注意：此字段可能返回 null，表示取不到有效值。
TradeVersion	Integer	交易版本 注意：此字段可能返回 null，表示取不到有效值。
ResourceOrderId	Integer	资源订单ID 注意：此字段可能返回 null，表示取不到有效值。
IsTradeCluster	Integer	是否计费集群 注意：此字段可能返回 null，表示取不到有效值。
AlarmInfo	String	集群错误状态告警信息 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
IsWoodpeckerCluster	Integer	是否采用新架构 注意：此字段可能返回 null，表示取不到有效值。
MetaDb	String	元数据库信息 注意：此字段可能返回 null，表示取不到有效值。
Tags	Array of Tag	标签信息 注意：此字段可能返回 null，表示取不到有效值。
HiveMetaDb	String	Hive元数据信息 注意：此字段可能返回 null，表示取不到有效值。
ServiceClass	String	集群类型:EMR,CLICKHOUSE,DRUID 注意：此字段可能返回 null，表示取不到有效值。
AliasInfo	String	集群所有节点的别名序列化 注意：此字段可能返回 null，表示取不到有效值。
ProductId	Integer	集群版本Id 注意：此字段可能返回 null，表示取不到有效值。

CustomMetaInfo

用户自建Hive-MetaDB信息

被如下接口引用：CreateInstance, InquiryPriceCreateInstance。

名称	类型	必选	描述
MetaDataJdbcUrl	String	否	自定义MetaDB的JDBC连接，请以 jdbc:mysql:// 开头
MetaDataUser	String	否	自定义MetaDB用户名
MetaDataPass	String	否	自定义MetaDB密码

EmrProductConfigOutter

EMR产品配置

被如下接口引用：DescribeInstances。

名称	类型	描述
SoftInfo	Array of String	软件信息 注意：此字段可能返回 null，表示取不到有效值。
MasterNodeSize	Integer	Master节点个数 注意：此字段可能返回 null，表示取不到有效值。
CoreNodeSize	Integer	Core节点个数 注意：此字段可能返回 null，表示取不到有效值。
TaskNodeSize	Integer	Task节点个数 注意：此字段可能返回 null，表示取不到有效值。
ComNodeSize	Integer	Common节点个数 注意：此字段可能返回 null，表示取不到有效值。
MasterResource	OutterResource	Master节点资源 注意：此字段可能返回 null，表示取不到有效值。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	描述
CoreResource	OutterResource	Core节点资源 注意：此字段可能返回 null，表示取不到有效值。
TaskResource	OutterResource	Task节点资源 注意：此字段可能返回 null，表示取不到有效值。
ComResource	OutterResource	Common节点资源 注意：此字段可能返回 null，表示取不到有效值。
OnCos	Boolean	是否使用COS 注意：此字段可能返回 null，表示取不到有效值。
ChargeType	Integer	收费类型 注意：此字段可能返回 null，表示取不到有效值。
RouterNodeSize	Integer	Router节点个数 注意：此字段可能返回 null，表示取不到有效值。
SupportHA	Boolean	是否支持HA 注意：此字段可能返回 null，表示取不到有效值。
SecurityOn	Boolean	是否支持安全模式 注意：此字段可能返回 null，表示取不到有效值。
SecurityGroup	String	安全组名称 注意：此字段可能返回 null，表示取不到有效值。
CbsEncrypt	Integer	是否开启Cbs加密 注意：此字段可能返回 null，表示取不到有效值。

HostVolumeContext

Pod HostPath挂载方式描述

被如下接口引用：ScaleOutInstance。

名称	类型	必选	描述
VolumePath	String	是	Pod挂载宿主机的目录。资源对宿主机的挂载点，指定的挂载点对应了宿主机的路径，该挂载点在Pod中作为数据存储目录使用 注意：此字段可能返回 null，表示取不到有效值。

LoginSettings

登录设置

被如下接口引用：CreateInstance。

名称	类型	必选	描述
Password	String	否	Password
PublicKeyId	String	否	Public Key

MultiDisk

多云盘参数

被如下接口引用：CreateInstance, InquiryPriceCreateInstance, InquiryPriceScaleOutInstance。

版权所有：腾讯云计算（北京）有限责任公司

名称	类型	必选	描述
DiskType	String	否	云盘类型("CLOUD_PREMIUM","CLOUD_SSD","CLOUD_BASIC")的一种
Volume	Integer	否	云盘大小
Count	Integer	否	该类型云盘个数

MultiDiskMC

多云盘参数

被如下接口引用：DescribeClusterNodes。

名称	类型	必选	描述
Count	Integer	是	该类型云盘个数 注意：此字段可能返回 null，表示取不到有效值。
Type	Integer	否	磁盘类型 注意：此字段可能返回 null，表示取不到有效值。
Volume	Integer	否	云盘大小 注意：此字段可能返回 null，表示取不到有效值。

NewResourceSpec

资源描述

被如下接口引用：CreateInstance, InquiryPriceCreateInstance。

名称	类型	必选	描述
MasterResourceSpec	Resource	否	描述Master节点资源
CoreResourceSpec	Resource	否	描述Core节点资源
TaskResourceSpec	Resource	否	描述Task节点资源
MasterCount	Integer	否	Master节点数量
CoreCount	Integer	否	Core节点数量
TaskCount	Integer	否	Task节点数量
CommonResourceSpec	Resource	否	描述Common节点资源
CommonCount	Integer	否	Common节点数量

NodeHardwareInfo

节点硬件信息

被如下接口引用：DescribeClusterNodes。

名称	类型	描述
AppId	Integer	用户APPID 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
SerialNo	String	序列号 注意：此字段可能返回 null，表示取不到有效值。
OrderNo	String	机器实例ID 注意：此字段可能返回 null，表示取不到有效值。
WanIp	String	master节点绑定外网IP 注意：此字段可能返回 null，表示取不到有效值。
Flag	Integer	节点类型。0:common节点；1:master节点；2:core节点；3:task节点 注意：此字段可能返回 null，表示取不到有效值。
Spec	String	节点规格 注意：此字段可能返回 null，表示取不到有效值。
CpuNum	Integer	节点核数 注意：此字段可能返回 null，表示取不到有效值。
MemSize	Integer	节点内存 注意：此字段可能返回 null，表示取不到有效值。
MemDesc	String	节点内存描述 注意：此字段可能返回 null，表示取不到有效值。
RegionId	Integer	节点所在region 注意：此字段可能返回 null，表示取不到有效值。
ZoneId	Integer	节点所在Zone 注意：此字段可能返回 null，表示取不到有效值。
ApplyTime	String	申请时间 注意：此字段可能返回 null，表示取不到有效值。
FreeTime	String	释放时间 注意：此字段可能返回 null，表示取不到有效值。
DiskSize	String	硬盘大小 注意：此字段可能返回 null，表示取不到有效值。
NameTag	String	节点描述 注意：此字段可能返回 null，表示取不到有效值。
Services	String	节点部署服务 注意：此字段可能返回 null，表示取不到有效值。
StorageType	Integer	磁盘类型 注意：此字段可能返回 null，表示取不到有效值。
RootSize	Integer	系统盘大小 注意：此字段可能返回 null，表示取不到有效值。
ChargeType	Integer	付费类型 注意：此字段可能返回 null，表示取不到有效值。
CdbIp	String	数据库IP 注意：此字段可能返回 null，表示取不到有效值。
CdbPort	Integer	数据库端口 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
HwDiskSize	Integer	硬盘容量 注意：此字段可能返回 null，表示取不到有效值。
HwDiskSizeDesc	String	硬盘容量描述 注意：此字段可能返回 null，表示取不到有效值。
HwMemSize	Integer	内存容量 注意：此字段可能返回 null，表示取不到有效值。
HwMemSizeDesc	String	内存容量描述 注意：此字段可能返回 null，表示取不到有效值。
ExpireTime	String	过期时间 注意：此字段可能返回 null，表示取不到有效值。
EmrResourceId	String	节点资源ID 注意：此字段可能返回 null，表示取不到有效值。
IsAutoRenew	Integer	续费标志 注意：此字段可能返回 null，表示取不到有效值。
DeviceClass	String	设备标识 注意：此字段可能返回 null，表示取不到有效值。
Mutable	Integer	支持变配 注意：此字段可能返回 null，表示取不到有效值。
MCMultiDisk	Array of MultiDiskMC	多云盘 注意：此字段可能返回 null，表示取不到有效值。
CdbNodeInfo	CdbInfo	数据库信息 注意：此字段可能返回 null，表示取不到有效值。
Ip	String	内网IP 注意：此字段可能返回 null，表示取不到有效值。
Destroyable	Integer	此节点是否可销毁，1可销毁，0不可销毁 注意：此字段可能返回 null，表示取不到有效值。
Tags	Array of Tag	节点绑定的标签 注意：此字段可能返回 null，表示取不到有效值。
AutoFlag	Integer	是否是自动扩缩容节点，0为普通节点，1为自动扩缩容节点。 注意：此字段可能返回 null，表示取不到有效值。
HardwareResourceType	String	资源类型, host/pod 注意：此字段可能返回 null，表示取不到有效值。

OutterResource

资源详情

被如下接口引用：DescribeInstances。

名称	类型	描述
Spec	String	规格 注意：此字段可能返回 null，表示取不到有效值。
SpecName	String	规格名 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
StorageType	Integer	硬盘类型 注意：此字段可能返回 null，表示取不到有效值。
DiskType	String	硬盘类型 注意：此字段可能返回 null，表示取不到有效值。
RootSize	Integer	系统盘大小 注意：此字段可能返回 null，表示取不到有效值。
MemSize	Integer	内存大小 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Integer	CPU个数 注意：此字段可能返回 null，表示取不到有效值。
DiskSize	Integer	硬盘大小 注意：此字段可能返回 null，表示取不到有效值。
InstanceType	String	规格 注意：此字段可能返回 null，表示取不到有效值。

PersistentVolumeContext

Pod PVC存储方式描述

被如下接口引用：ScaleOutInstance。

名称	类型	必选	描述
DiskSize	Integer	否	磁盘大小，单位为GB。 注意：此字段可能返回 null，表示取不到有效值。
DiskType	String	否	磁盘类型。CLOUD_PREMIUM;CLOUD_SSD 注意：此字段可能返回 null，表示取不到有效值。
DiskNum	Integer	否	磁盘数量 注意：此字段可能返回 null，表示取不到有效值。

Placement

描述集群实例位置信息

被如下接口引用：CreateInstance, InquiryPriceCreateInstance, InquiryPriceRenewInstance, InquiryPriceUpdateInstance。

名称	类型	必选	描述
ProjectId	Integer	是	实例所属项目ID。该参数可以通过调用 DescribeProject 的返回值中的 projectId 字段来获取。填0为默认项目。
Zone	String	是	实例所属的可用区，例如ap-guangzhou-1。该参数也可以通过调用 DescribeZones 的返回值中的Zone字段来获取。

PodSpec

扩容容器资源时的资源描述

被如下接口引用：ScaleOutInstance。

名称	类型	必选	描述
ResourceProviderIdentifier	String	是	外部资源提供者的标识符，例如"cls-a1cd23fa"。
ResourceProviderType	String	是	外部资源提供者类型，例如"tke",当前仅支持"tke"。
NodeType	String	是	资源的用途，即节点类型，当前仅支持"TASK"。
Cpu	Integer	是	CPU核数。
Memory	Integer	是	内存大小，单位为GB。
DataVolumes	Array of String	否	资源对宿主机的挂载点，指定的挂载点对应了宿主机的路径，该挂载点在Pod中作为数据存储目录使用。弃用
CpuType	String	否	Eks集群-CPU类型，当前支持"intel"和"amd"
PodVolumes	Array of PodVolume	否	Pod节点数据目录挂载信息。

PodVolume

Pod的存储设备描述信息。

被如下接口引用：ScaleOutInstance。

名称	类型	必选	描述
VolumeType	String	是	存储类型，可为"pvc"，"hostpath"。 注意：此字段可能返回 null，表示取不到有效值。
PVCVolume	PersistentVolumeContext	否	当VolumeType为"pvc"时，该字段生效。 注意：此字段可能返回 null，表示取不到有效值。
HostVolume	HostVolumeContext	否	当VolumeType为"hostpath"时，该字段生效。 注意：此字段可能返回 null，表示取不到有效值。

PreExecuteFileSettings

预执行脚本配置

被如下接口引用：CreateInstance, ScaleOutInstance。

名称	类型	必选	描述
Path	String	否	脚本在COS上路径，已废弃
Args	Array of String	否	执行脚本参数
Bucket	String	否	COS的Bucket名称，已废弃
Region	String	否	COS的Region名称，已废弃
Domain	String	否	COS的Domain数据，已废弃
RunOrder	Integer	否	执行顺序
WhenRun	String	否	resourceAfter 或 clusterAfter
CosFileName	String	否	脚本文件名，已废弃
CosFileURI	String	否	脚本的cos地址

名称	类型	必选	描述
CosSecretId	String	否	cos的SecretId
CosSecretKey	String	否	Cos的SecretKey
AppId	String	否	cos的appid，已废弃

PriceResource

询价资源

被如下接口引用：InquiryPriceScaleOutInstance。

名称	类型	描述
Spec	String	需要的规格 注意：此字段可能返回 null，表示取不到有效值。
StorageType	Integer	硬盘类型 注意：此字段可能返回 null，表示取不到有效值。
DiskType	String	硬盘类型 注意：此字段可能返回 null，表示取不到有效值。
RootSize	Integer	系统盘大小 注意：此字段可能返回 null，表示取不到有效值。
MemSize	Integer	内存大小 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Integer	核心数量 注意：此字段可能返回 null，表示取不到有效值。
DiskSize	Integer	硬盘大小 注意：此字段可能返回 null，表示取不到有效值。
MultiDisks	Array of MultiDisk	云盘列表 注意：此字段可能返回 null，表示取不到有效值。
DiskCnt	Integer	磁盘数量 注意：此字段可能返回 null，表示取不到有效值。
InstanceType	String	规格 注意：此字段可能返回 null，表示取不到有效值。
Tags	Array of Tag	标签 注意：此字段可能返回 null，表示取不到有效值。
DiskNum	Integer	磁盘数量 注意：此字段可能返回 null，表示取不到有效值。
LocalDiskNum	Integer	本地盘的数量 注意：此字段可能返回 null，表示取不到有效值。

Resource

资源详情

被如下接口引用：CreateInstance, InquiryPriceCreateInstance。

名称	类型	必选	描述

名称	类型	必选	描述
Spec	String	是	节点规格描述 注意：此字段可能返回 null，表示取不到有效值。
StorageType	Integer	是	存储类型 注意：此字段可能返回 null，表示取不到有效值。
DiskType	String	是	磁盘类型 注意：此字段可能返回 null，表示取不到有效值。
MemSize	Integer	是	内存容量,单位为M 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Integer	是	CPU核数 注意：此字段可能返回 null，表示取不到有效值。
DiskSize	Integer	是	数据盘容量 注意：此字段可能返回 null，表示取不到有效值。
RootSize	Integer	否	系统盘容量 注意：此字段可能返回 null，表示取不到有效值。
MultiDisks	Array of MultiDisk	否	云盘列表，当数据盘为一块云盘时，直接使用DiskType和DiskSize参数，超出部分使用MultiDisks 注意：此字段可能返回 null，表示取不到有效值。
Tags	Array of Tag	否	需要绑定的标签列表 注意：此字段可能返回 null，表示取不到有效值。
InstanceType	String	否	规格类型 注意：此字段可能返回 null，表示取不到有效值。
LocalDiskNum	Integer	否	本地盘数量 注意：此字段可能返回 null，表示取不到有效值。
DiskNum	Integer	否	盘数量 注意：此字段可能返回 null，表示取不到有效值。

SearchItem

搜索字段

被如下接口引用：DescribeClusterNodes。

名称	类型	必选	描述
SearchType	String	是	支持搜索的类型
SearchValue	String	是	支持搜索的值

Tag

标签

被如下接口引用：CreateInstance, DescribeClusterNodes, DescribeInstances, InquiryPriceCreateInstance, InquiryPriceScaleOutInstance, ScaleOutInstance。

名称	类型	必选	描述
TagKey	String	否	标签键

名称	类型	必选	描述
TagValue	String	否	标签值

UpdateInstanceSettings

变配资源规格

被如下接口引用：InquiryPriceUpdateInstance。

名称	类型	必选	描述
Memory	Integer	是	内存容量，单位为G
CPUCores	Integer	是	CPU核数
ResourceId	String	是	机器资源ID（EMR测资源标识）
InstanceType	String	否	变配机器规格

VPCSettings

VPC 参数

被如下接口引用：CreateInstance, InquiryPriceCreateInstance。

名称	类型	必选	描述
VpcId	String	是	VPC ID
SubnetId	String	是	Subnet ID

错误码

最近更新时间：2020-10-27 08:04:26

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
UnsupportedOperation	操作不支持。
ResourceInUse	资源被占用。
InternalError	内部错误。
RequestLimitExceeded	请求的次数超过了频率限制。
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。
LimitExceeded	超过配额限制。
NoSuchVersion	接口版本不存在。
ResourceNotFound	资源不存在。
AuthFailure.SignatureFailure	签名错误。 签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
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。
UnsupportedRegion	接口不支持所传地域。
UnauthorizedOperation	未授权操作。
InvalidParameter	参数错误。
ResourceUnavailable	资源不可用。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.TokenFailure	token 错误。
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。
UnknownParameter	未知参数错误。
UnsupportedProtocol	HTTP(S)请求协议错误，只支持 GET 和 POST 请求。
InvalidParameterValue	参数取值错误。
InvalidAction	接口不存在。
MissingParameter	缺少参数错误。
ResourceInsufficient	资源不足。

业务错误码

错误码	说明
FailedOperation.DuplicateOrderNotAllowed	重复的订单，请检查emr控制台。
InternalError.AccountCgwError	内部服务调用异常。
InternalError.CamCgwError	内部服务调用异常。
InternalError.CamError	内部服务调用异常。
InternalError.CbsCgwError	内部服务调用异常。
InternalError.CbsError	内部服务调用异常。
InternalError.CdbCgwError	内部服务调用异常。
InternalError.CdbError	内部服务调用异常。
InternalError.CheckQuotaErr	cvm或cbs资源不够或软件不合法。
InternalError.ConfigCgwError	内部服务调用异常。
InternalError.CvmError	内部服务调用异常。
InternalError.KmsError	内部服务调用异常。
InternalError.ProjectCgwError	内部服务调用异常。
InternalError.SgError	安全组接口调用异常。
InternalError.TKEError	TKE调用出错。
InternalError.TagError	内部服务调用异常。
InternalError.TradeCgwError	内部服务调用异常。
InternalError.VpcCgwError	内部服务调用异常。
InternalError.VpcError	内部服务调用异常。
InternalError.WoodServerError	Woodpecker server调用出错。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
InvalidParameter.DisplayStrategyNotMatch	展示策略错误。
InvalidParameter.IncorrectCommonCount	Common节点数量无效。
InvalidParameter.IncorrectMasterCount	Master节点数量无效。
InvalidParameter.InvaildCoreCount	core节点的数量不能超过20。
InvalidParameter.InvalidAppId	无效参数，AppId。
InvalidParameter.InvalidAutoRenew	无效的自动续费标识。
InvalidParameter.InvalidClientToken	无效的ClientToken。
InvalidParameter.InvalidClusterId	无效参数，ClusterId。
InvalidParameter.InvalidComponent	无效的组件。
InvalidParameter.InvalidCoreCount	Core节点数量无效。
InvalidParameter.InvalidCount	扩容数量必须大于0。
InvalidParameter.InvalidCountNum	同一请求只能扩容Task或者Core节点。
InvalidParameter.InvalidInstanceName	无效的集群名称。
InvalidParameter.InvalidLoginSetting	无效的登录设置。
InvalidParameter.InvalidModifySpec	变配规格无效。
InvalidParameter.InvalidNodeType	无效的NodeType。
InvalidParameter.InvalidPassword	无效密码。
InvalidParameter.InvalidPaymode	无效的付费类型。
InvalidParameter.InvalidPreExecutedFile	无效的引导操作脚本。
InvalidParameter.InvalidProductId	无效的产品ID。
InvalidParameter.InvalidProjectId	无效的项目ID。
InvalidParameter.InvalidResourceIds	资源ID无效。
InvalidParameter.InvalidResourceSpec	无效的资源规格。
InvalidParameter.InvalidSercurityGrpupId	无效的安全组ID。
InvalidParameter.InvalidServiceName	服务名无效。
InvalidParameter.InvalidServiceNodeInfo	参数ServiceNodeInfo无效或错误。
InvalidParameter.InvalidSoftDeployInfo	参数InvalidSoftDeployInfo无效或错误。
InvalidParameter.InvalidSoftInfo	无效的SoftInfo。
InvalidParameter.InvalidSoftWareName	软件名无效。
InvalidParameter.InvalidSoftWareVersion	软件版本无效。
InvalidParameter.InvalidSubnetId	无效的子网ID。
InvalidParameter.InvalidSupportHA	无效的高可用参数。

版权所有：腾讯云计算（北京）有限责任公司

错误码	说明
InvalidParameter.InvalidTaskCount	task的数量不能超过20。
InvalidParameter.InvalidTimeSpan	无效的timespan。
InvalidParameter.InvalidTimeUnit	无效的TimeUnit。
InvalidParameter.InvalidVpcId	无效的私有网络ID。
InvalidParameter.InvalidZone	无效的可用区。
InvalidParameter.OrderFieldNotMatch	排序字段错误。
InvalidParameter.PayModeResourceNotMatch	付费模式与资源不匹配。
InvalidParameter.ProjectResourceNotMatch	项目与资源不匹配。
InvalidParameter.SoftwareNotInProduct	存在无效的产品组件。
InvalidParameter.UngrantedPolicy	策略为授权。
InvalidParameter.UngrantedRole	角色未授权。
InvalidParameter.ZoneResourceNotMatch	可用区与资源不匹配。
InvalidParameterValue.InvalidTkeInstance	无效的Tke集群ID，或Tke集群不符合条件。
ResourceInUse.InstanceInProcess	实例在流程中。
ResourceInsufficient.DiskInsufficient	硬盘规格不满足。
ResourceInsufficient.InstanceInsufficient	不支持或售罄的节点规格。
ResourceNotFound.ClusterNotFound	无法找到该实例。
ResourceNotFound.HardwareInfoNotFound	无法找到硬件信息。
ResourceNotFound.InstanceNotFound	无法找到该实例。
ResourceNotFound.ResourceNotFound	资源未找到。
ResourceNotFound.SubnetNotFound	找不到对应的子网。
ResourceNotFound.TKEPreconditionNotFound	tke集群前置组件未部署。
ResourceNotFound.TagsNotFound	没有查找到指定标签。
ResourceUnavailable.ResourceSpec_NotDefaultSpec	当前资源规格不存在默认规格。
ResourcesSoldOut.	资源售罄。
ResourcesSoldOut.CbsSoldOut	CBS资源售罄。
ResourcesSoldOut.CvmSoldOut	云服务器已售罄。