

云 API

快速入门

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

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。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。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模式的承诺或保证。

文档目录

快速入门

入门必读

使用 API Explorer

使用 TCCLI

使用 API Inspector

快速入门

入门必读

最近更新时间：2023-03-07 18:16:40

通过云 API，您只需少量代码即可快速操作云产品。为方便您快速轻松调用云 API 来管理腾讯云资源，云 API 提供了 API Explorer、腾讯云命令行工具 TCCLI、SDK 及 API Inspector 工具。

使用工具

本文以云服务器 CVM 的 [查看实例列表 \(DescribeInstances\)](#) 接口为例，引导您如何快速使用 API Explorer、TCCLI 及 SDK 工具完成接口的调用，适用于初次接触腾讯云产品的开发者用户。同时云 API 提供 API Inspector 工具，方便用户查看腾讯云提供的所有 API 服务，以及控制台操作对应的 API 调用信息。

工具	适用场景	详细介绍
API Explorer	腾讯云 API Explorer 是一款自动化工具，适用于交互式操作界面的场景。无需多余的配置、操作便捷，但不适用多次连续使用。	使用 API Explorer
TCCLI	TCCLI 是管理腾讯云资源的统一工具，适用于命令行工具的场景。通过腾讯云命令行工具，您可以快速轻松的调用腾讯云 API 来管理您的腾讯云资源。	使用 TCCLI
SDK	腾讯云开发者工具套件（SDK）3.0，适用于 SDK 编码等场景。SDK 3.0 是云 API 3.0 平台的配套工具。目前已经支持 CVM、VPC、CBS 等产品，后续所有的云服务产品都会陆续接入。	SDK 简介
API Inspector	腾讯云 API Inspector 支持录制控制台每一步操作关联的 API 调用信息、并自动生成各语言版本的 API 代码。	使用 API Inspector

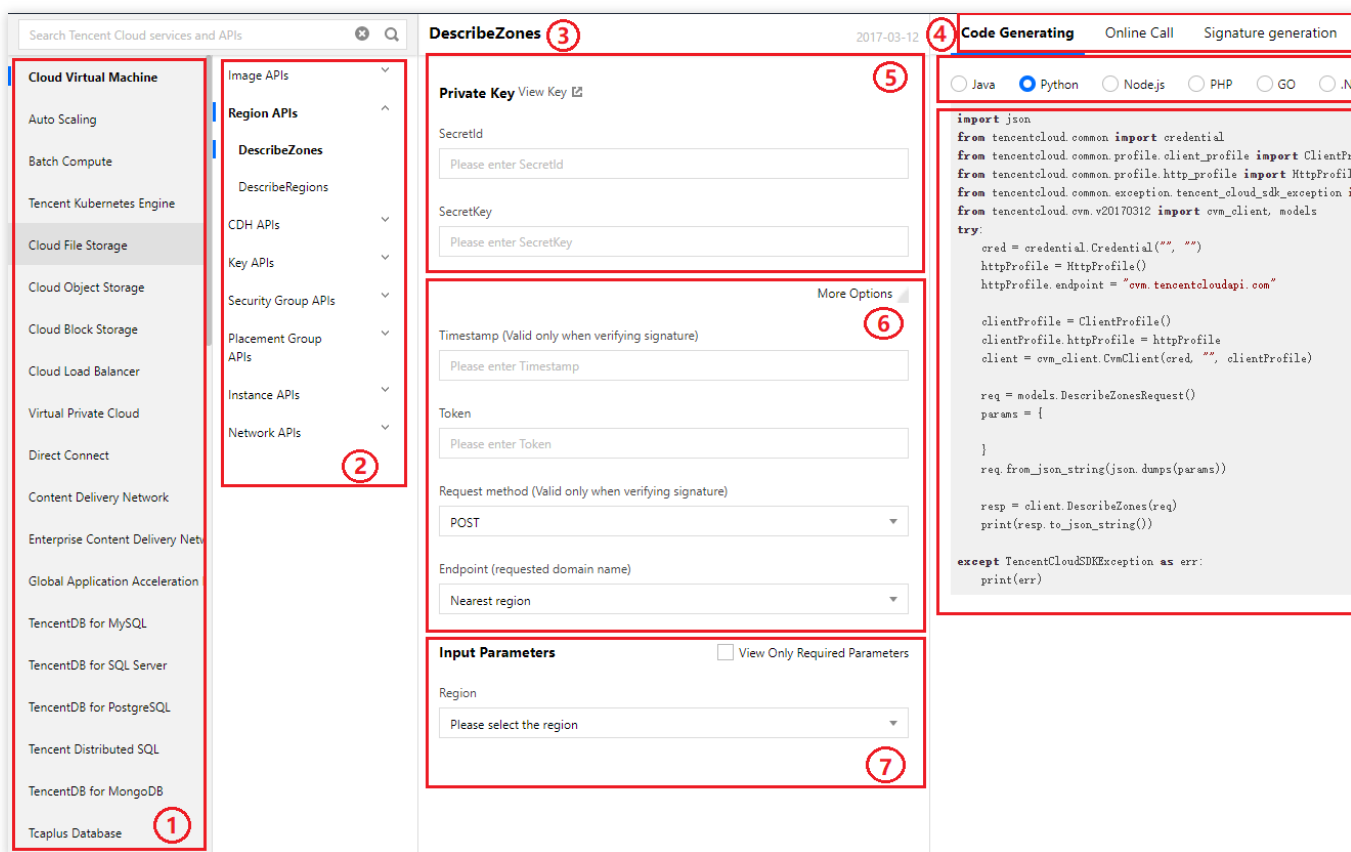
使用 API Explorer

最近更新时间：2023-03-07 18:16:40

腾讯云 API Explorer 工具是一款自动化工具，目前已支持云服务器 CVM、私有网络 VPC、云硬盘 CBS 等 腾讯云产品 的 API 接口调用自动化。可实现自动生成 Java、Python、Node.js、PHP、GO 及 .NET 语言的 SDK 代码、在线调用、发送真实请求及签名串自动生成等功能，降低了 SDK 的使用难度。

API Explorer 工具详细介绍

本文将 API Explorer 工具整体页面从左至右顺序依次详细介绍。如下图所示：



1. 产品区域：目前已支持的所有产品。
2. 产品接口区域：当前产品下已支持的所有功能接口。
3. 接口名：所选择的接口名。
4. 产品版本：各产品版本有一定区别，详情请参见各接口文档。图中为云服务器接口版本 2017-03-12。
5. 密钥对（SecretId、SecretKey）：输入账户的安全凭证，可单击【查看密钥】进行获取。

注意：

密钥代表您的账号身份和所拥有的权限，等同于您的登录密码，切勿泄露他人。

6. **签名所需参数**：单击**更多选项**，可查看用于签名串生成、签名串验证等功能所需的参数。各产品接口所需参数可能有一定区别，详情请参见各接口文档。云服务器签名所需参数如下：

Timestamp（仅在验证签名串生成时有效）：当前 UNIX 时间戳，精确到秒，用于记录发起 API 请求的时间。

Timestamp 必须为当前系统时间，且需确保系统时间与标准时间同步，若相差超过5分钟则会发生请求失败。若长时间不和标准时间同步，则可能导致运行一段时间后，请求失败并返回签名过期错误。

Token（仅在验证签名串生成时有效）：用于认证用户身份。各产品对该参数要求可能有一定区别，如有需要会注明获取方式，详情请参见各接口文档。

请求方式（仅在验证签名串生成时有效）：默认为 POST 请求，请结合接口文档进行选择。

Endpoint（仅在验证签名串生成时有效）：选择接入地域，默认就近接入。

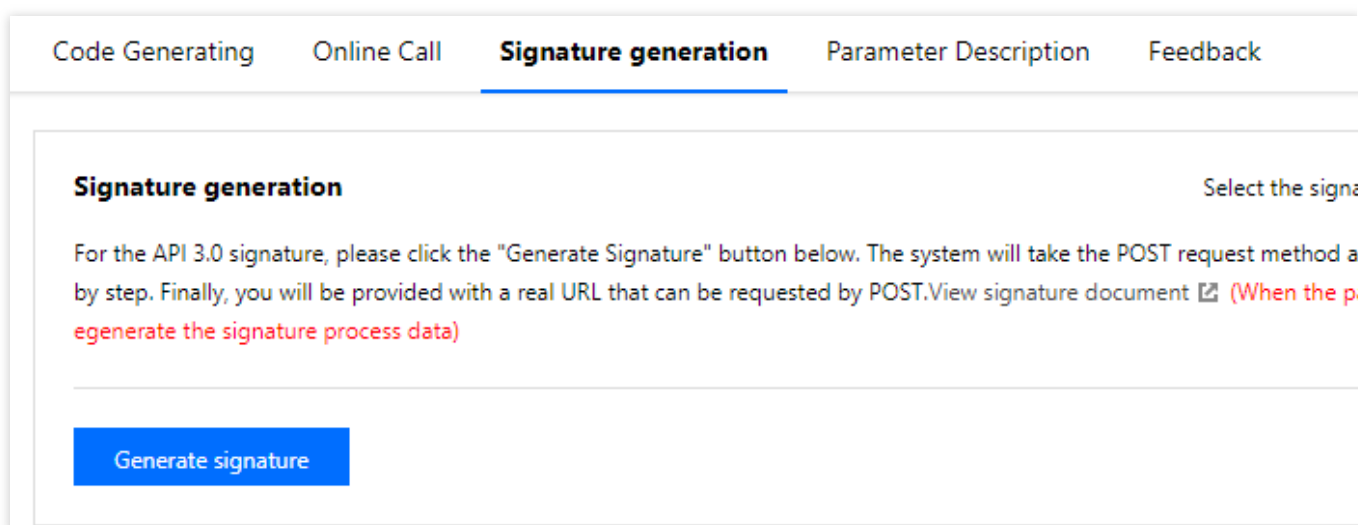
7. **接口所需参数**：仅展示接口所需参数，您可勾选“只看必填参数”进行筛选。具体参数说明可通过选择右侧的【参数说明】进行查看。

8. **功能区**：

代码生成：可通过此功能自动生成多语言版本代码，降低使用难度。

在线调用：在填写参数后，选择**发送请求**，系统会发送您在左侧填写的参数到对应的接口。该操作等同于真实操作，同时系统会展示请求结果、响应头等相关信息。

签名串生成：可通过此功能自动生成签名串，默认使用 API 3.0 v3 版本，您可按需选择其他版本。如下图所示：



9. **子功能区**：可切换语言版本，生成对应代码。

10. **响应区**：此处展示生成代码、请求结果等响应信息。

11. **SDK 使用说明**：如需了解 SDK 更多信息，例如所需环境及调用示例，可查阅相应的 SDK 使用说明。

调用接口

本文以 [查询可用区列表](#) 接口为例，使用 API Explorer 工具进行调用：

1. 获取个人密钥（SecretId、SecretKey），并填入对应位置。
2. 填写所需参数，可选择右侧功能区中的**参数说明**查看接口具体参数信息。

3. 选择右侧功能区中的**在线调用** > **发送请求**，即可在响应区查看请求结果。

生成签名串

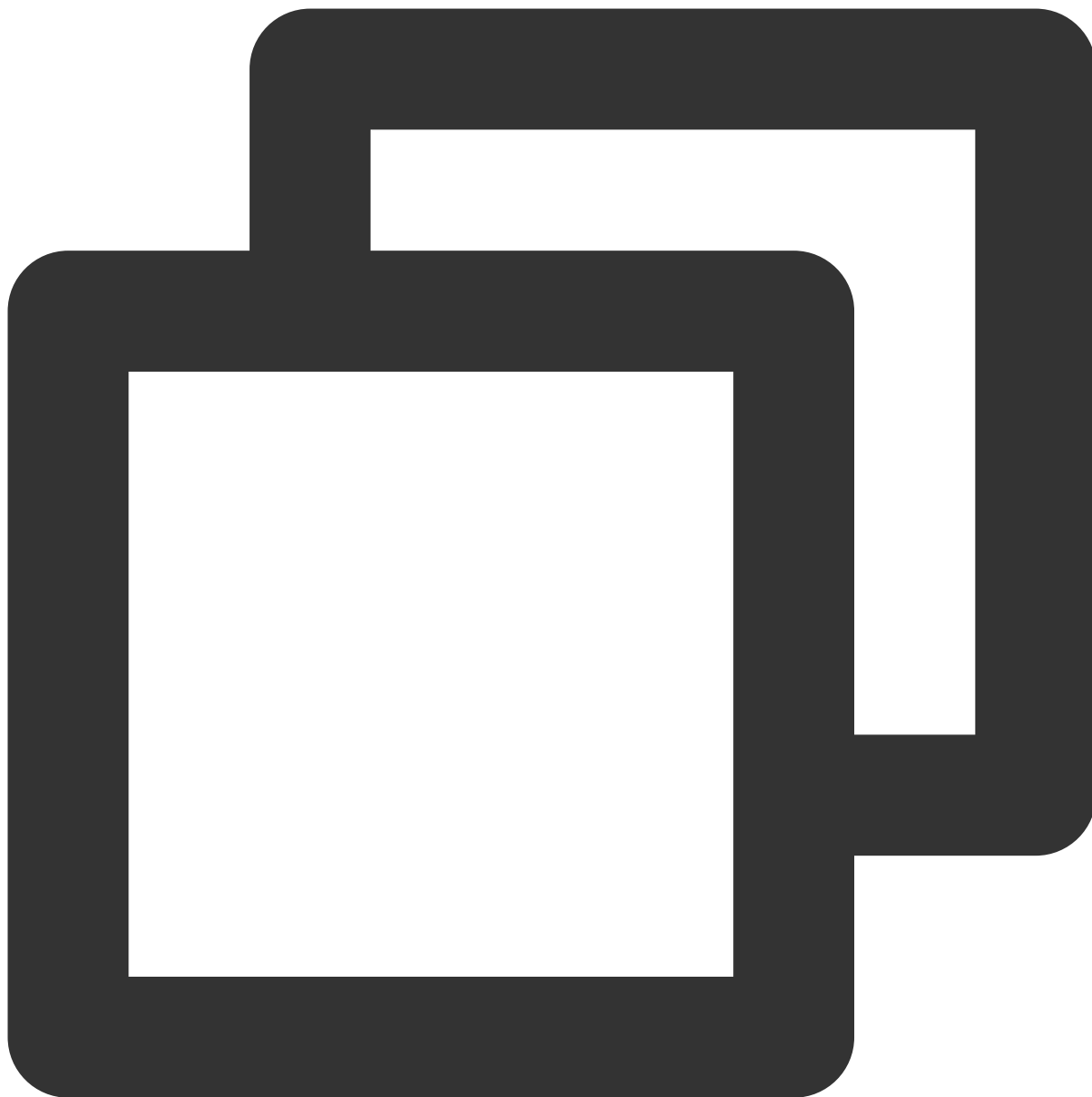
本文以 [查询可用区列表](#) 接口为例，使用 API Explorer 工具进行签名串生成：

1. 获取个人密钥（SecretId、SecretKey），并填入对应位置。
2. （可选）填写签名需参数，您可按需进行填写。若不填写，系统在生成签名串时会自动输入签名所需参数。
3. 填写接口所属参数，可选择右侧功能区中的**参数说明**查看接口具体参数信息。
4. 选择右侧功能区中的**签名串生成** > **生成签名**，即可在响应区查看签名的详细步骤与结果。

相关问题

使用工具如何验证 API 签名

当您遇到如下报错信息时，可使用 API Explorer 工具进行验证：



```
[TencentCloudSDKException] code:AuthFailure.SecretIdNotFound message:The SecretId i
```

1. 将参数填写在 API Explorer 工具中，Timestamp 等可变参数请与所需验证的签名（API 错误签名）使用的参数保持一致，并选择功能区的**签名串生成 > 生成签名**。
2. 在响应区获得签名步骤及结果后，即可进行前后数据对比。

签名错误

如您在签名过程中出现错误，则可能收到以下签名失败的错误代码，请对应实际情况进行处理：

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）。

联系我们

若您在使用过程中发现问题，可通过选择功能区**问题反馈**进行反馈。

您也可以单击右下角的



，在“帮助中心”查询相关信息。

使用 TCCLI

最近更新时间：2023-03-07 18:16:40

欢迎使用腾讯云命令行工具 TCCLI，TCCLI 是管理腾讯云资源的统一工具。通过腾讯云命令行工具，您可以快速轻松的调用腾讯云 API 来管理您的腾讯云资源。您还可以基于腾讯云的命令行工具来做自动化和脚本处理，能够以更多样的方式进行组合和重用。

腾讯云 TCCLI 包含基础功能和高级功能，请参见下表：

功能	描述
基础功能	配置 TCCLI helper 信息支持中文信息 支持 JSON、table 及 text 输出格式
高级功能	多版本接口访问 指定最近的接入点（Endpoint） 返回结果过滤 支持输出入参数据结构到 JSON 文件 支持从 JSON 文件读取参数调用 复杂类型点（.）连接展开方式调用

安装 TCCLI

1. 安装 Python 环境和 pip 工具，安装命令行工具前请确保您的系统已经安装了 Python 环境和 pip 工具。详情请参见 [Python SDK](#) 文档。

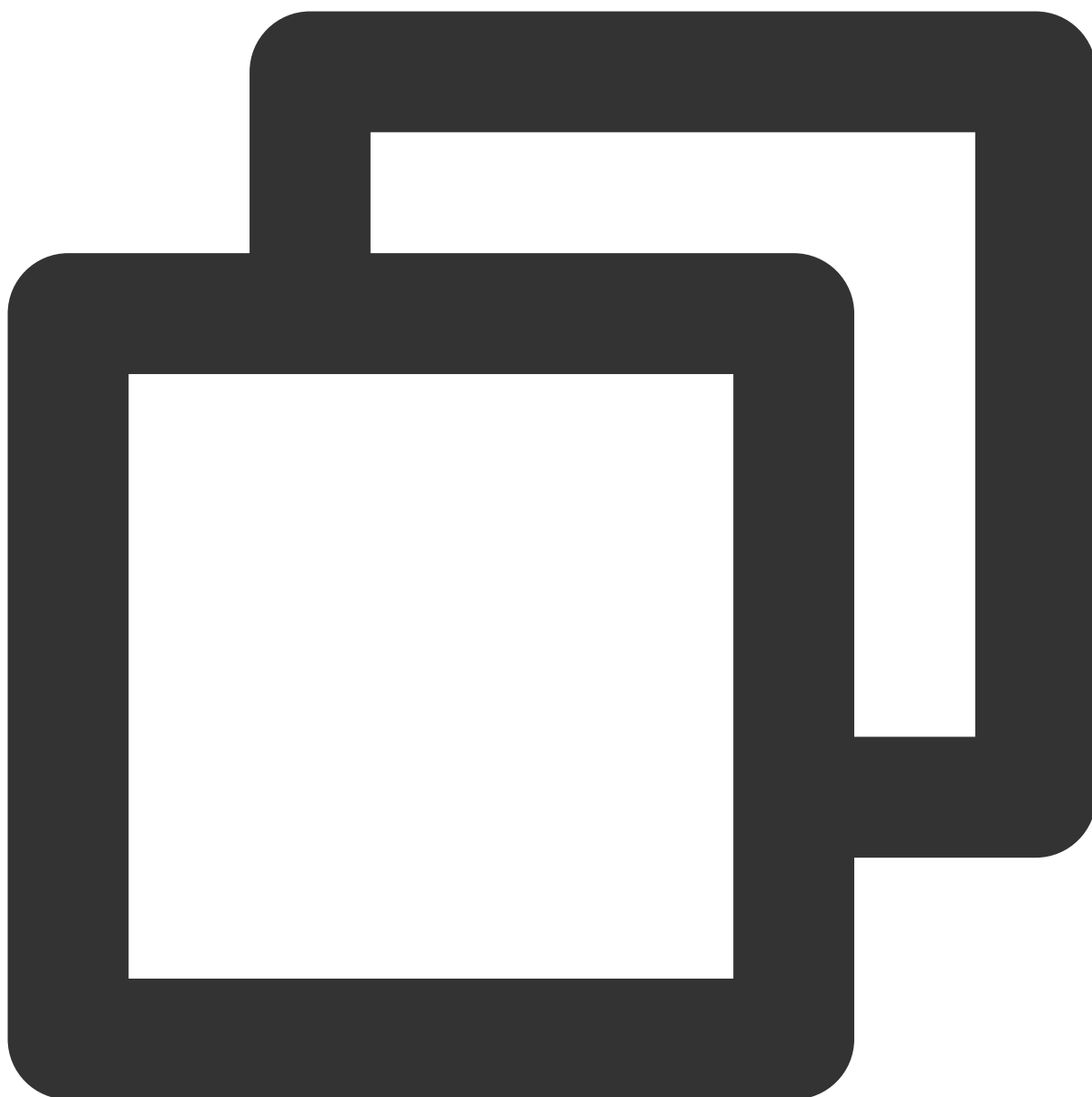
注意：

Python 版本必须为2.7及以上版本，更多内容请参考 [Python](#) 和 [pip](#) 官网文档。

TCCLI 依赖于 TencentCloudApi Python SDK，如果 TencentCloudApi Python SDK 的版本号小于要安装 TCCLI 版本号，在安装 TCCLI 时会自动升级 TencentCloudApi Python SDK。

2. Windows 系统按 **Win+R** 打开运行窗口输入 cmd 并单击**确定**，本文以 Linux 为例。

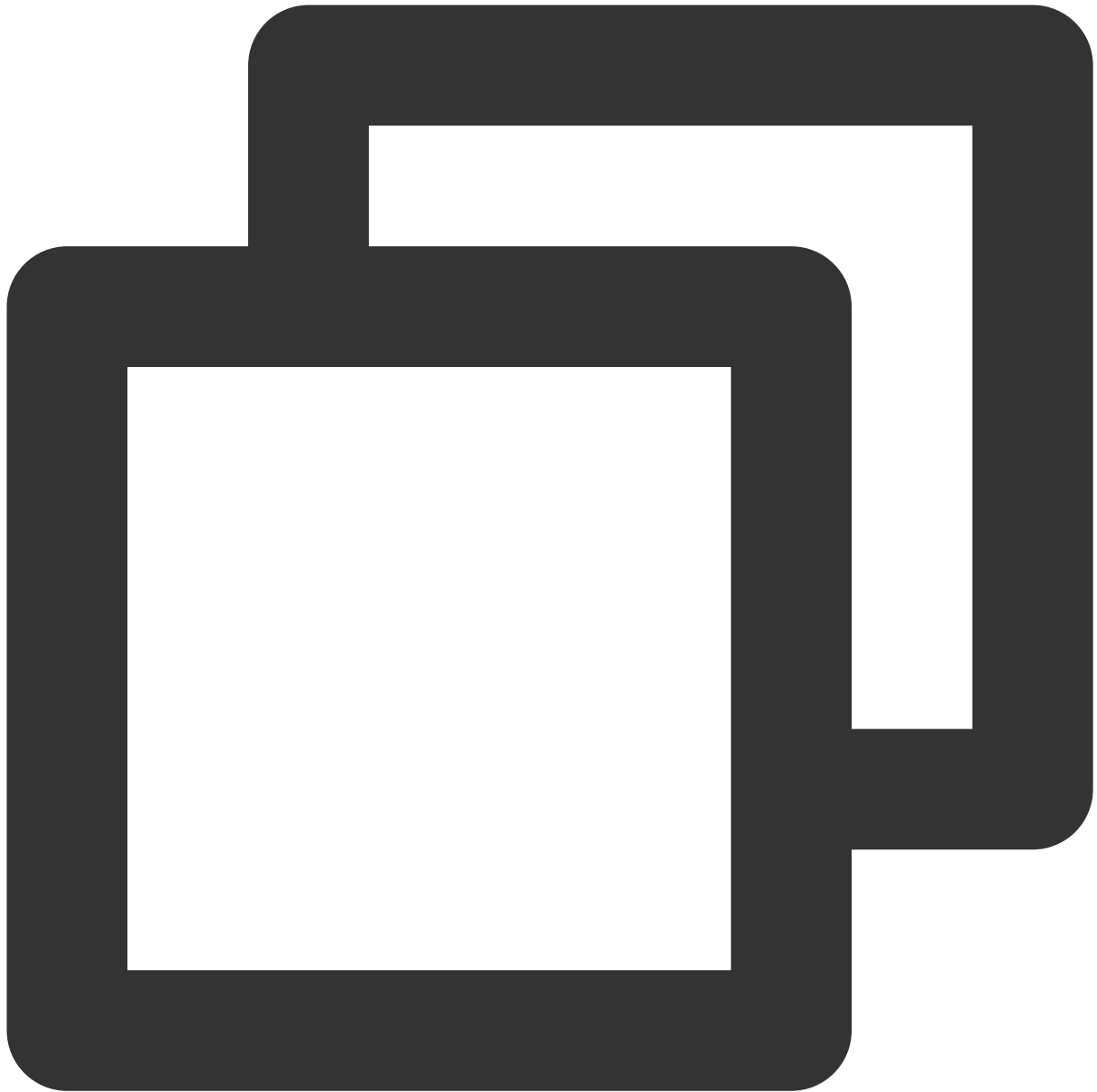
3. 在命令行窗口中，执行以下命令进行 TCCLI 安装。



```
pip install tccli
```

注意：

3.0.96.1以下版本升级需要执行以下代码：



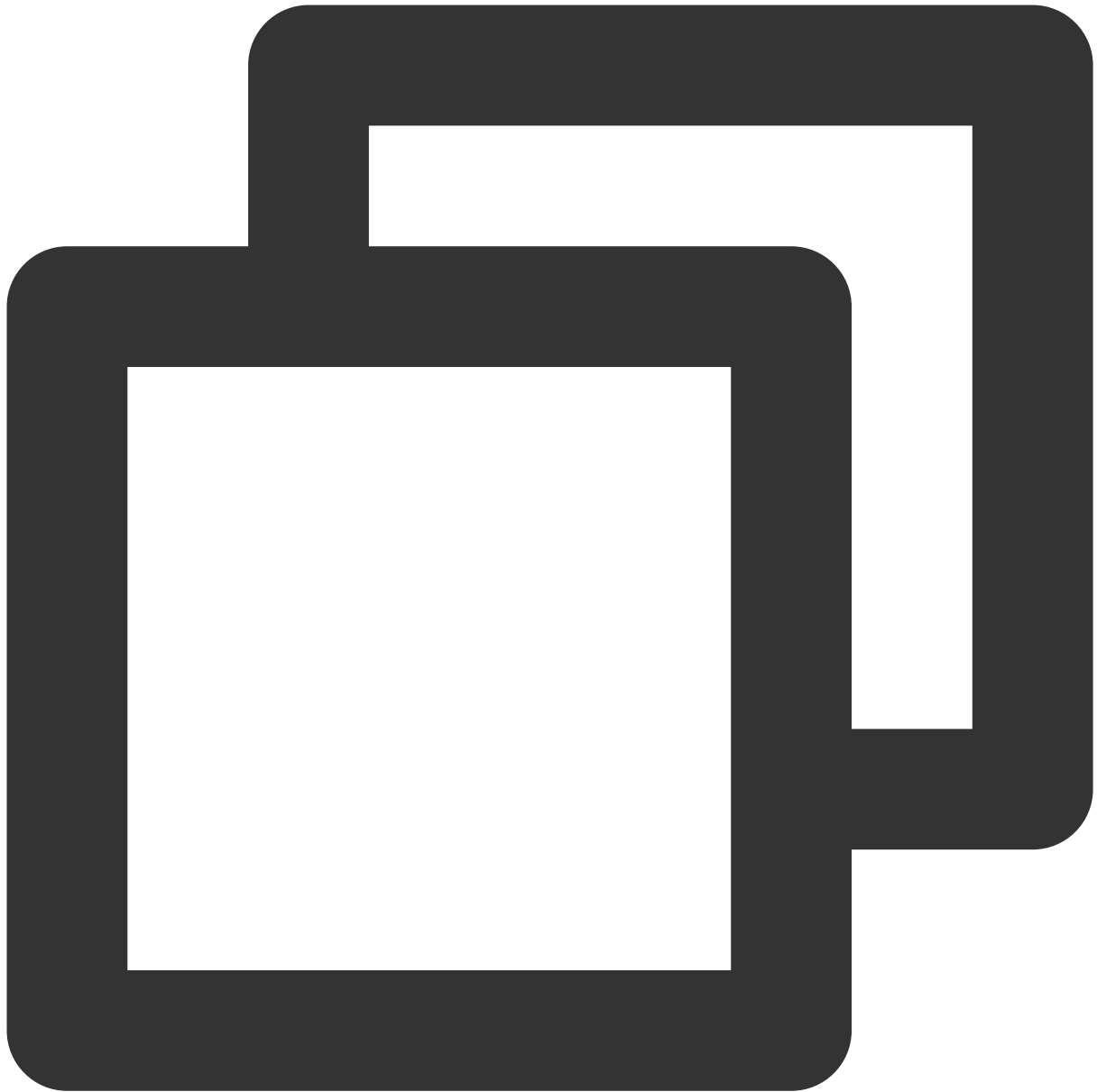
```
sudo pip uninstall tccli jmespath  
sudo pip install tccli
```

4. 执行以下命令，查看 TCCLI 是否安装成功。



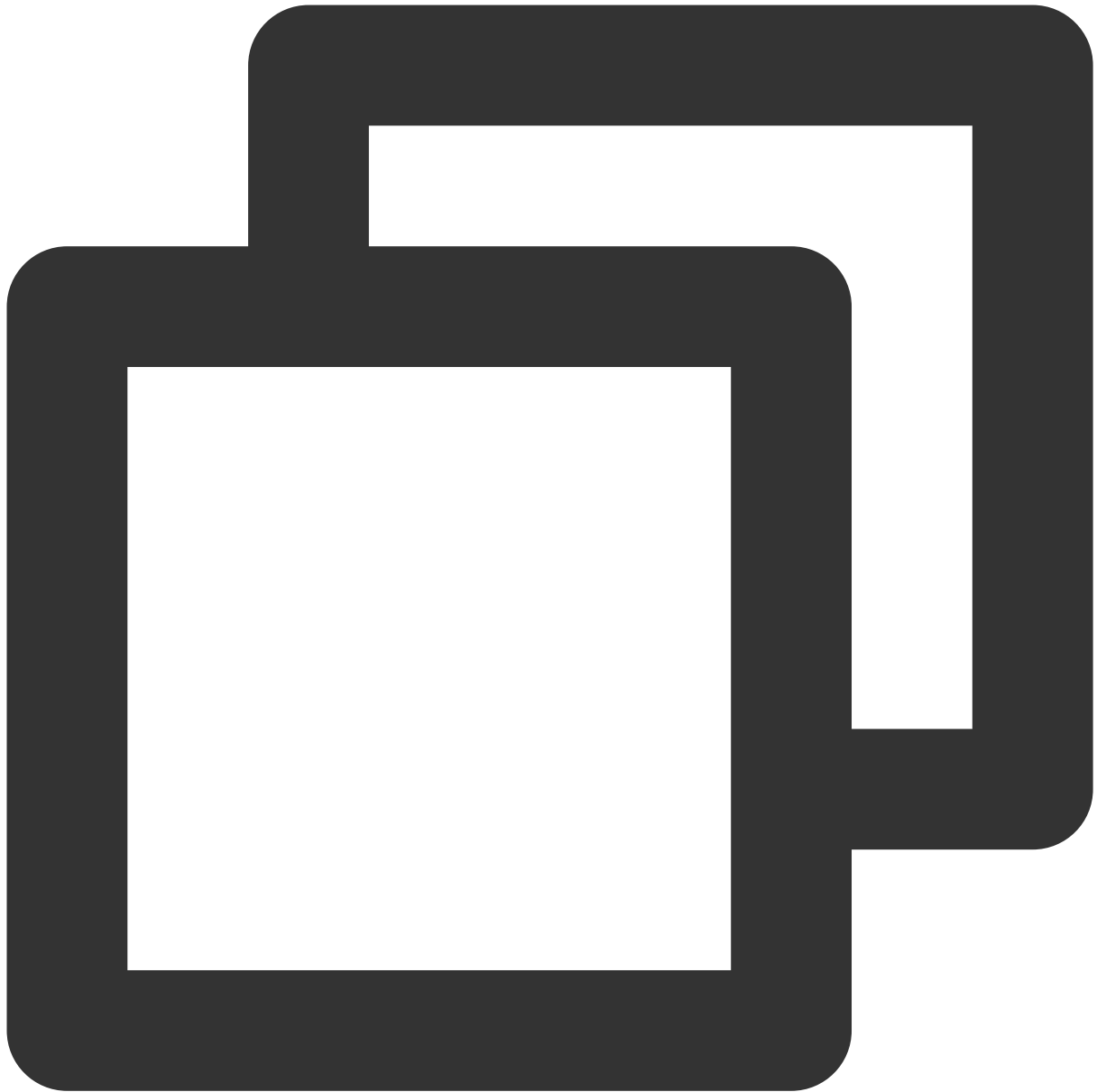
```
tccli --version
```

返回类似如下结果，则说明已成功安装 TCCLI。



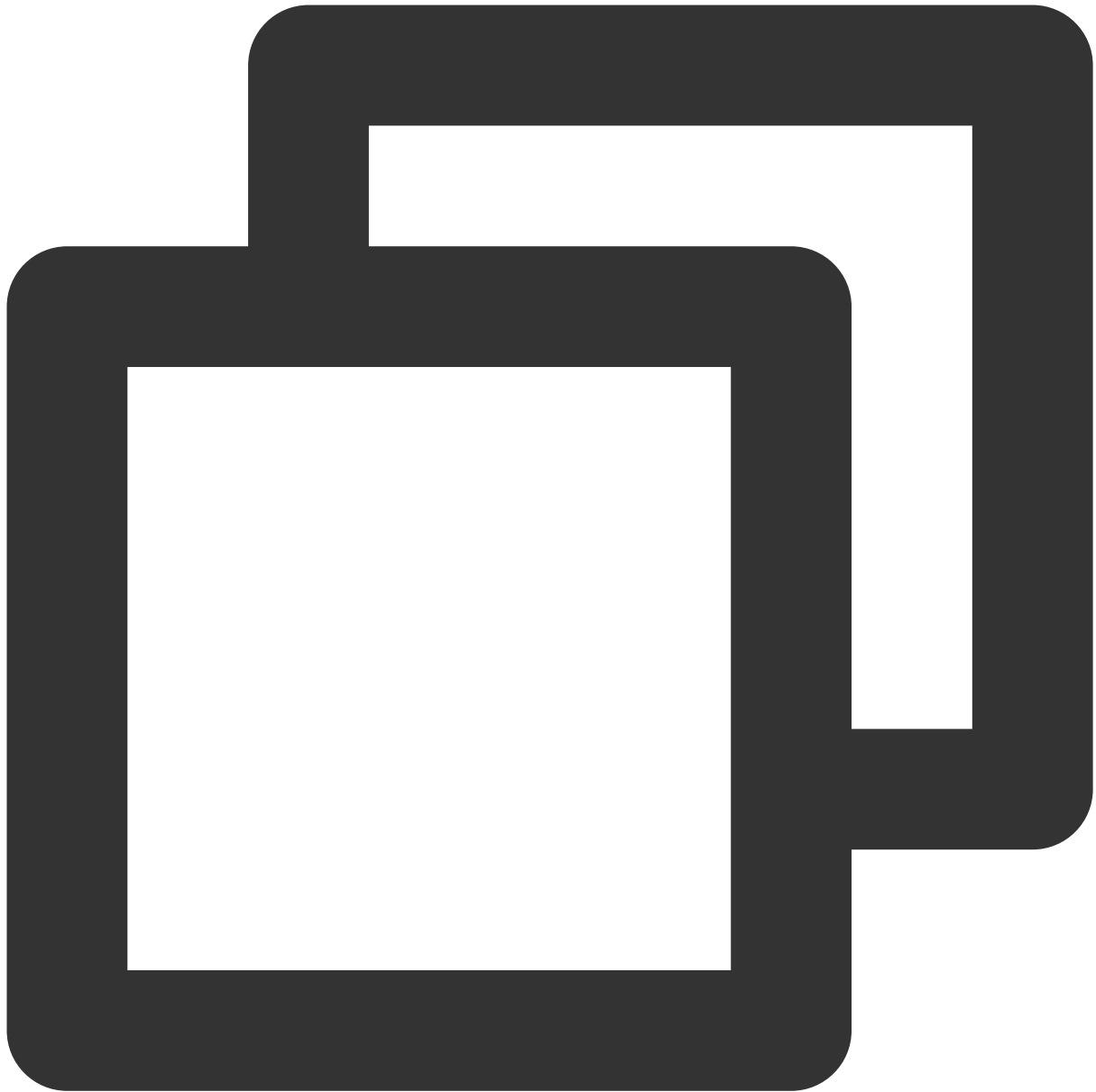
```
[root@VM_180_248_centos ~]# tccli --version  
3.0.250.1
```

5. 在 Linux 环境中，执行以下命令启动自动补全功能，支持大小写自动纠错：



```
complete -C 'tccli_completer' tccli
```

以下代码片段展示自动补全过程：



```
[root@VM_33_50_centos ~]# tccli c
cam          cbs          cdn          chdfs        ckafka       cloudhsm     cms
captcha     ccc          cds          cim          clb          cme          confi
cat         cdb          cfs          cis          clouaudit   cmq          cpdp
[root@VM_33_50_centos ~]# tccli cvm R
RebootInstances          ResetInstance          ResetInst
RenewHosts              ResetInstancesInternetMaxBandwidth  ResizeIns
RenewInstances          ResetInstancesPassword  RunInstan
[root@VM_33_50_centos ~]# tccli cvm RunInstances --
--ActionTimer          --generate-cli-skeleton  --InstanceType
--ClientToken          --HostName              --InternetAccessible
```



```
--cli-input-json          --HpcClusterId          --LoginSettings
--DataDisks               --ImageId               --output
--DisasterRecoverGroupIds --InstanceChargePrepaid --Placement
--DryRun                  --InstanceChargeType    --profile
--endpoint                --InstanceCount         --region
--EnhancedService         --InstanceMarketOptions --secretId
--filter                  --InstanceName          --secretKey
[root@VM_33_50_centos ~]# tccli cvm RunInstances --Placement
```

配置 TCCLI

1. 在命令行中执行以下命令，进入交互模式快配置。



```
tccli configure
```

返回结果如下，请参考以下信息进行配置：



```
TencentCloud API secretId [*afcQ]:  
TencentCloud API secretKey [*ArFd]:  
region:  
output [json]:
```

secretId：云 API 密钥 SecretId，请前往 [API 密钥管理](#) 获取（一个主账号最多可以申请两个云 API 密钥）。

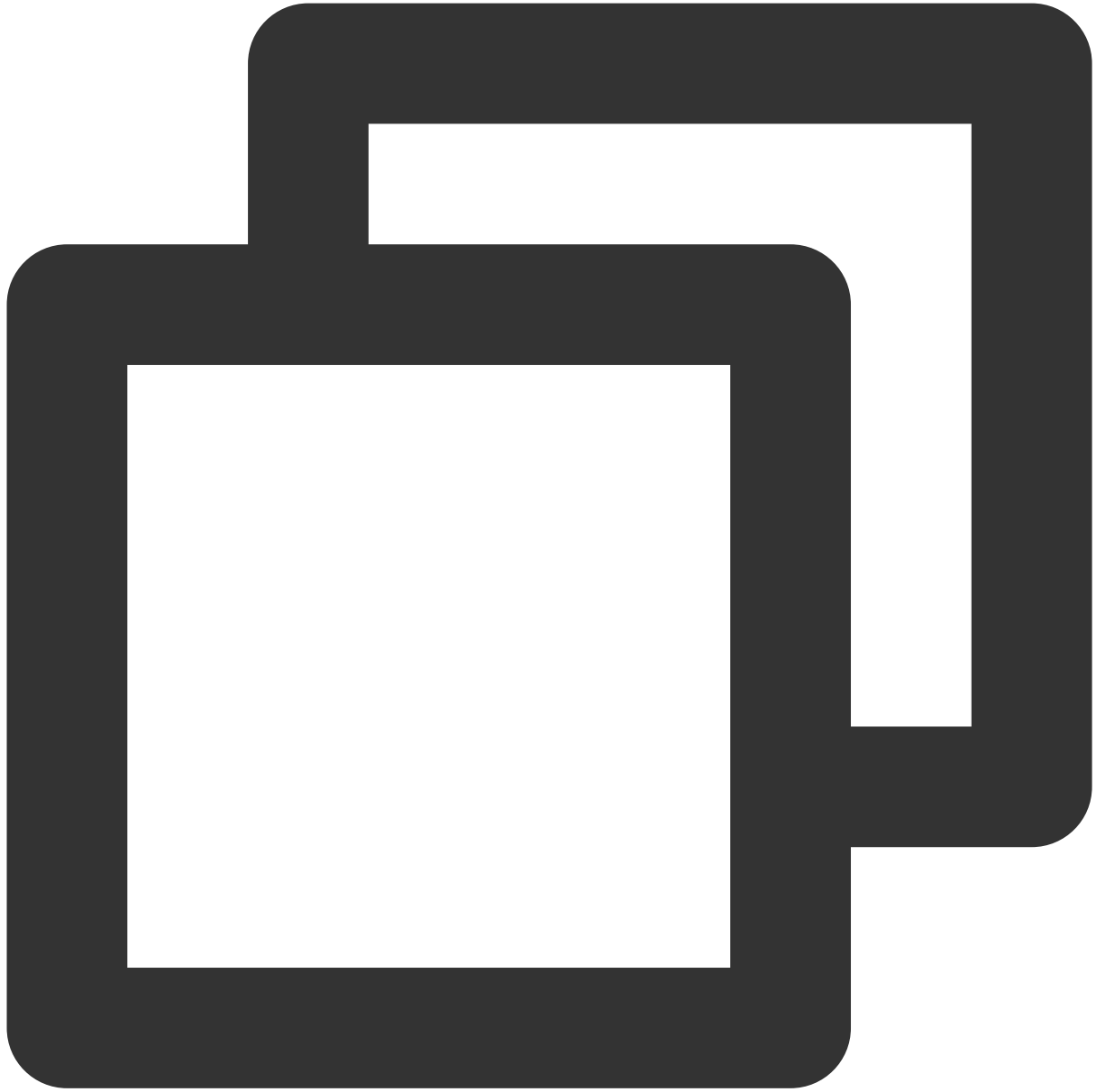
secretKey：云 API 密钥 SecretKey，请前往 [API 密钥管理](#) 获取。

region：云产品地域，请前往对应云产品的 [API 文档](#) 获取可用的 region。例如云服务器的 [地域列表](#)。

output：可选参数，请求回包输出格式，支持 JSON、table 及 text 三种格式，默认为 JSON。

更多信息请执行 `tccli configure help` 命令查看。

2. 您可执行以下命令进入命令行模式，通过命令行模式您可以在自动化脚本中配置您的信息。

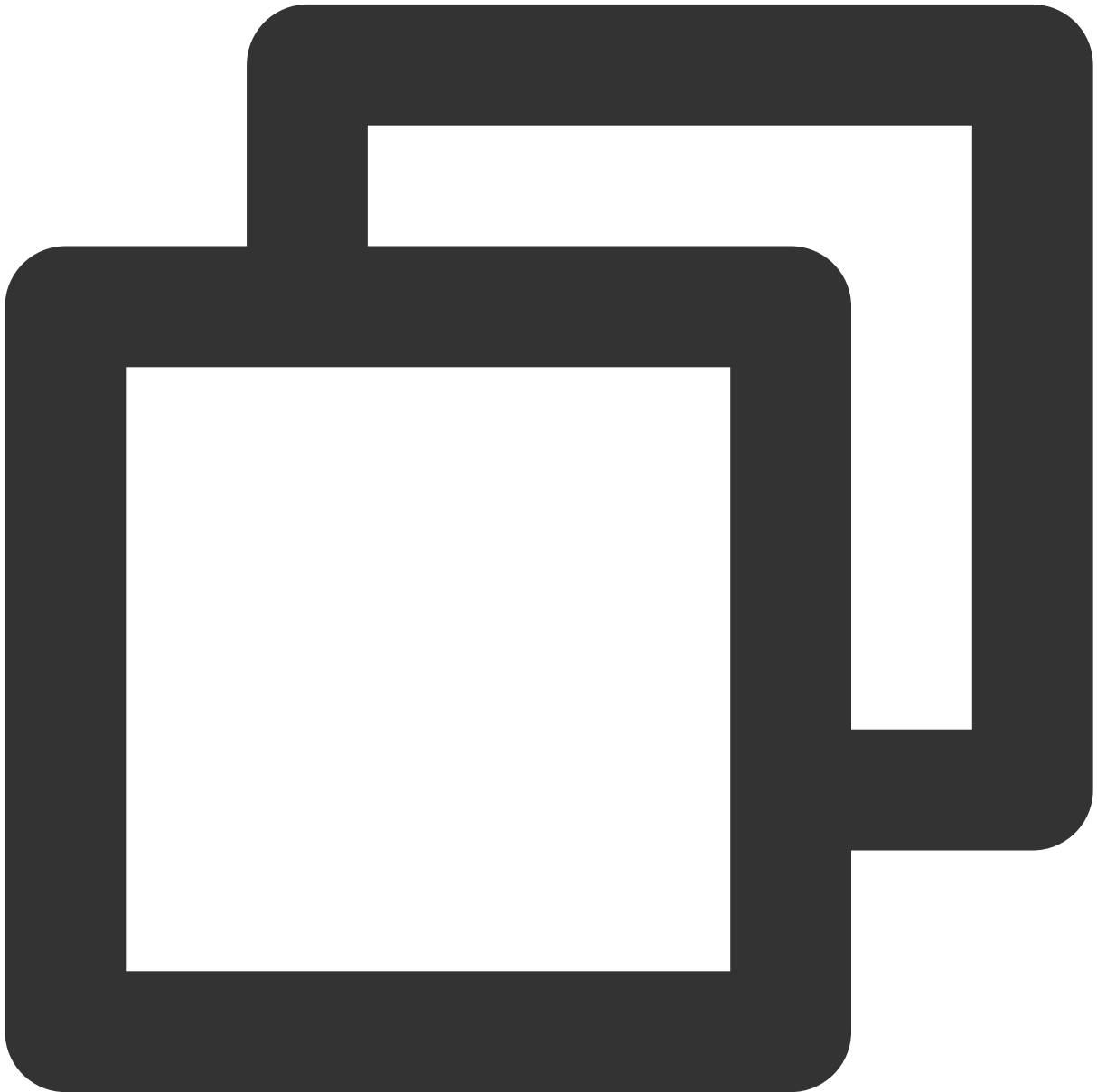


```
# set子命令可以设置某一配置，也可同时配置多个。
tccli configure set secretId AKIDwLw1234*****nR2OTI787aBCDP
tccli configure set region ap-guangzhou output json
# get子命令用于获取配置信息。
tccli configure get secretKey
secretKey = OxXj7khcV1234*****dCc1LiArFd
# list子命令打印所有配置信息。
tccli configure list
credential:
```

```
secretId = AKIDwLw1234*****nR2OTI787aBCDP
secretKey = OxXj7khcV1234*****dCc1LiArFd
configure:
region = ap-guangzhou
output = json
```

更多信息请执行 `tccli configure [list、get 或 set] help` 查看，例如 `tccli configure list help`。

3. 您可执行以下命令配置多账户支持，方便您在多种配置同时使用。



```
# 在交互模式中指定账户名 test。
```

```
$ tccli configure --profile test
TencentCloud API secretId [*BCDP]:AKIDwLw1234*****R2OTI787aBCDP
TencentCloud API secretKey [*ArFd]:OxXj7khcV1234*****dCc1LiArFd
region: ap-guangzhou
output [json]:
# set/get/list子命令指定账户名 test。此命令与上条命令作用相同
$ tccli configure set region ap-guangzhou output json secretId AKIDwLw1234*****
# 已可以修改单独一个，例如修改地域：
$ tccli configure set region ap-beijing
# 查看test用户密钥key或查看配置，使用命令如下：
$ tccli configure get secretKey --profile test
$ tccli configure list --profile test
# 在调用接口时指定账户（以 cvm DescribeZones 接口为例）。
$ tccli cvm DescribeZones --profile test
```

使用 TCCLI

基础功能

TCCLI 支持自主配置，help 信息支持中文信息且支持 JSON、table 及 text 输出格式。

注意：

请注意示例中非简单类型的参数必须为标准 JSON 格式。

TCCLI 目前支持以下三种调用方式：

JSON 字符串入参调用

JSON 文件入参调用 `--cli-input-json`

复杂类型点 (.) 连接展开形式入参调用 `--cli-unfold-argument`

JSON 字符串入参调用示例

执行以下命令，创建一台 CVM。



```
$ tccli cvm RunInstances --InstanceChargeType POSTPAID_BY_HOUR --InstanceChargePrep
```

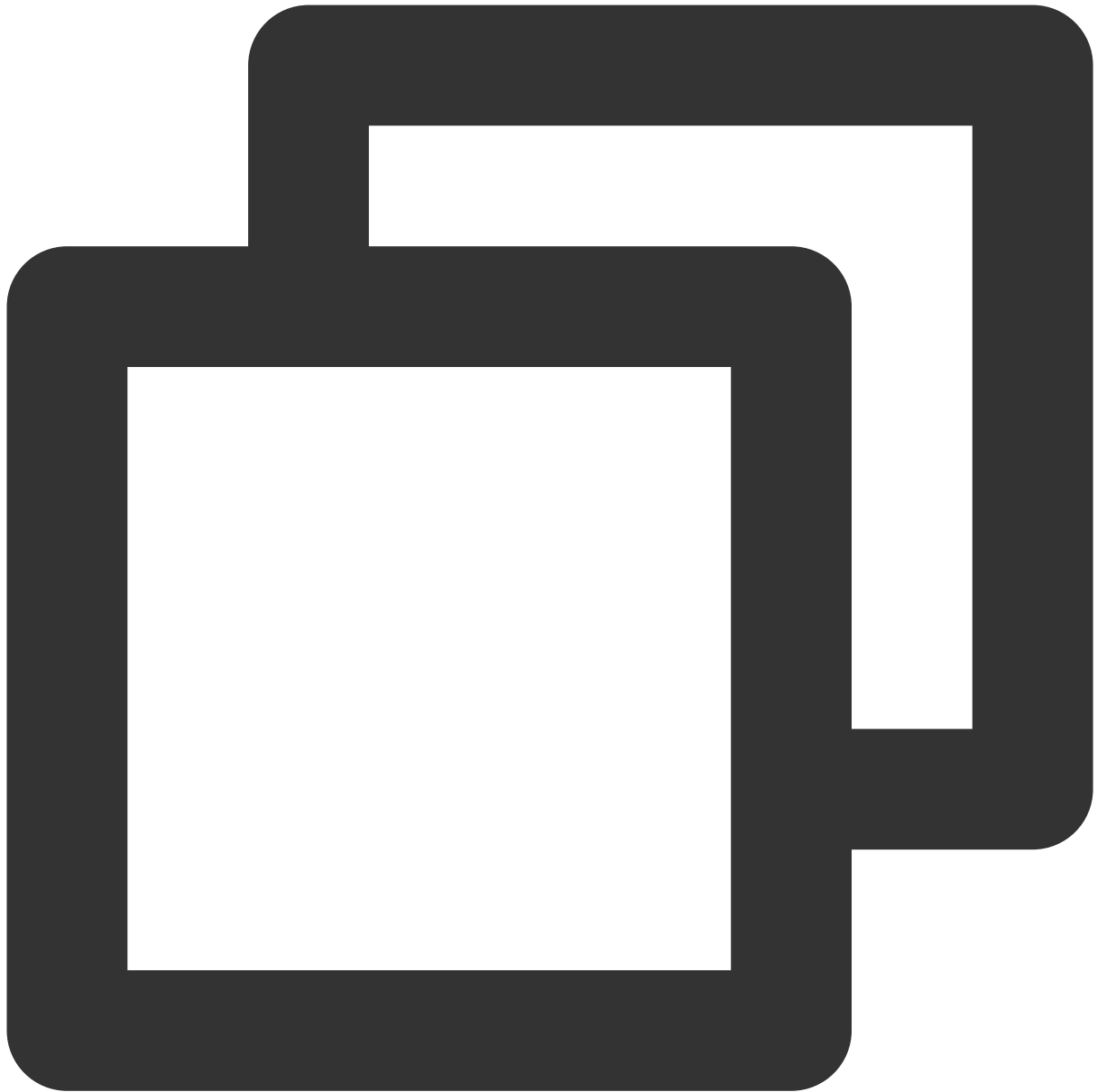
执行以下命令，获取云产品 CVM 的监控数据。



```
[root@VM_33_50_centos ~]# tccli monitor GetMonitorData --Namespace "QCE/CVM" --Peri
```

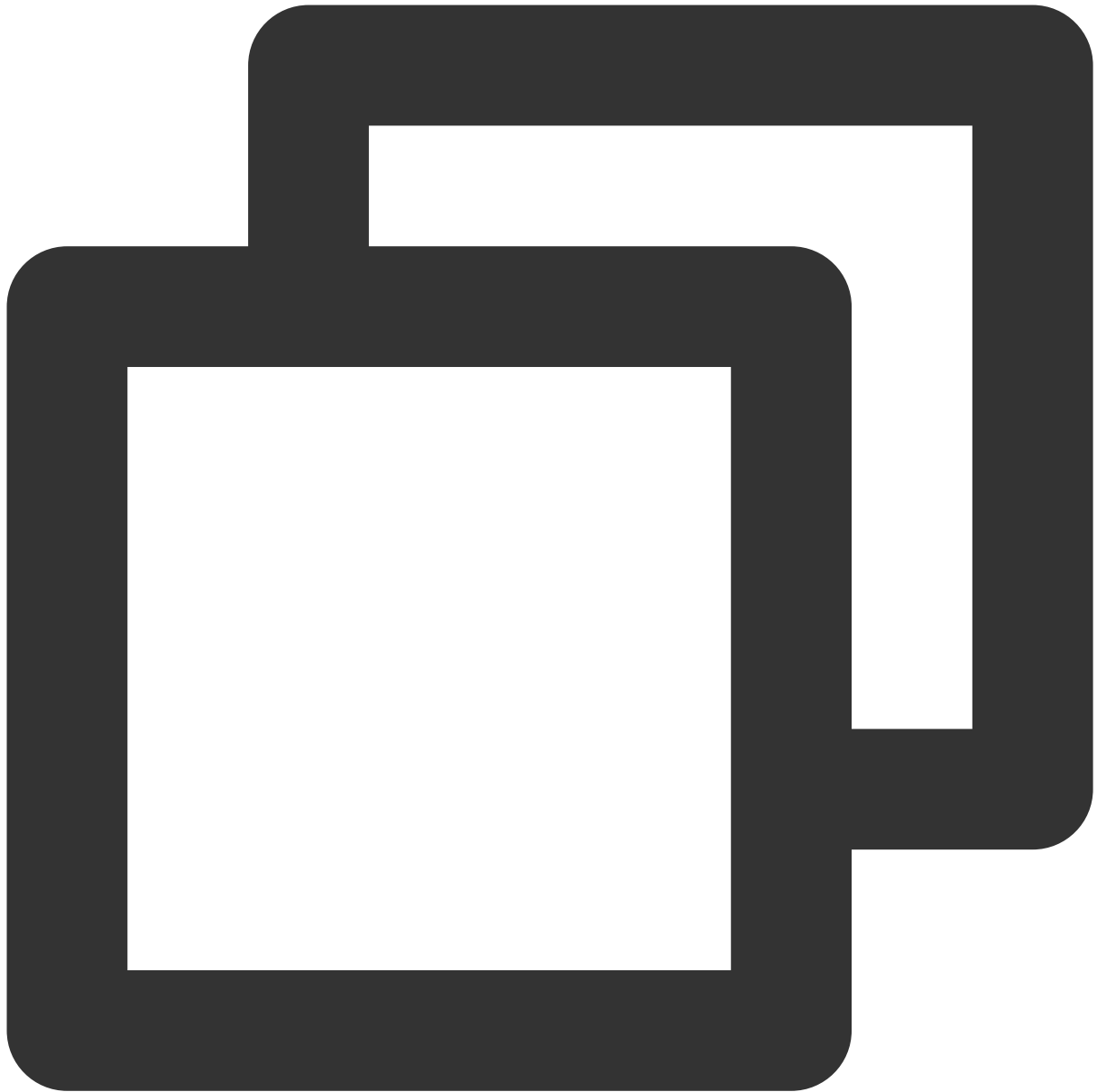
JSON 文件入参调用 (--cli-input-json) 示例

1. 执行以下命令，输出入参数据结构到 JSON 文件。



```
[root@VM_33_50_centos ~]# tccli cvm RunInstances --generate-cli-skeleton > /tmp/Ru
```

2. 将文件中参数值替换为实际使用值，并使用该 JSON 文件作为入参，格式为 `--cli-input-json` 后接 `file://+文件路径`。示例如下：



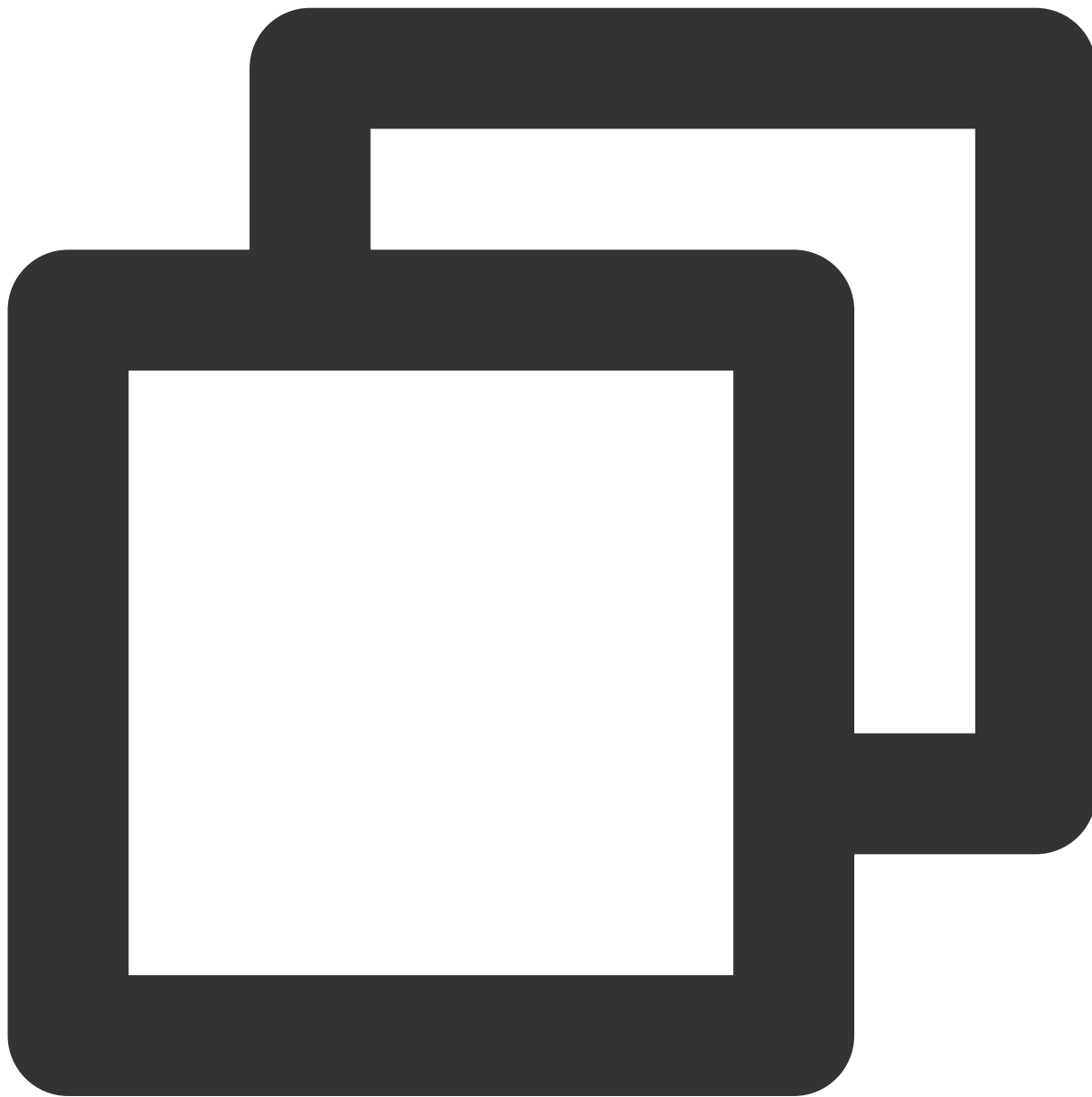
```
[root@VM_33_50_centos ~]# tccli cvm RunInstances --cli-input-json file:///tmp/RunIn
{
  "RequestId": "20e2b42d-3260-4750-9293-79116208330e",
  "InstanceIdSet": null
}
```

复杂类型点 (.) 连接展开形式入参调用 (--cli-unfold-argument) 示例

复杂类型点连接展开调用是将复杂类型按用点连接的形式展开，此方式可充分利用命令行自动补全机制来解决入参较复杂时的命令行输入困难，且易出错问题。展开方式如下：

复杂类型 `{"a":{"b": "c"}}` 展开为 `--a.b c`。复杂类型数组使用 `.0` , `.1` 表示数组的第一个、第二个元素。基本类型数组不需要使用 `.0` , `.1` , 直接将数组多个元素用空格隔开依次输入, 例如 `--Integer 10 20` , `--String str1 str2`。

执行以下命令, 创建一台 CVM。

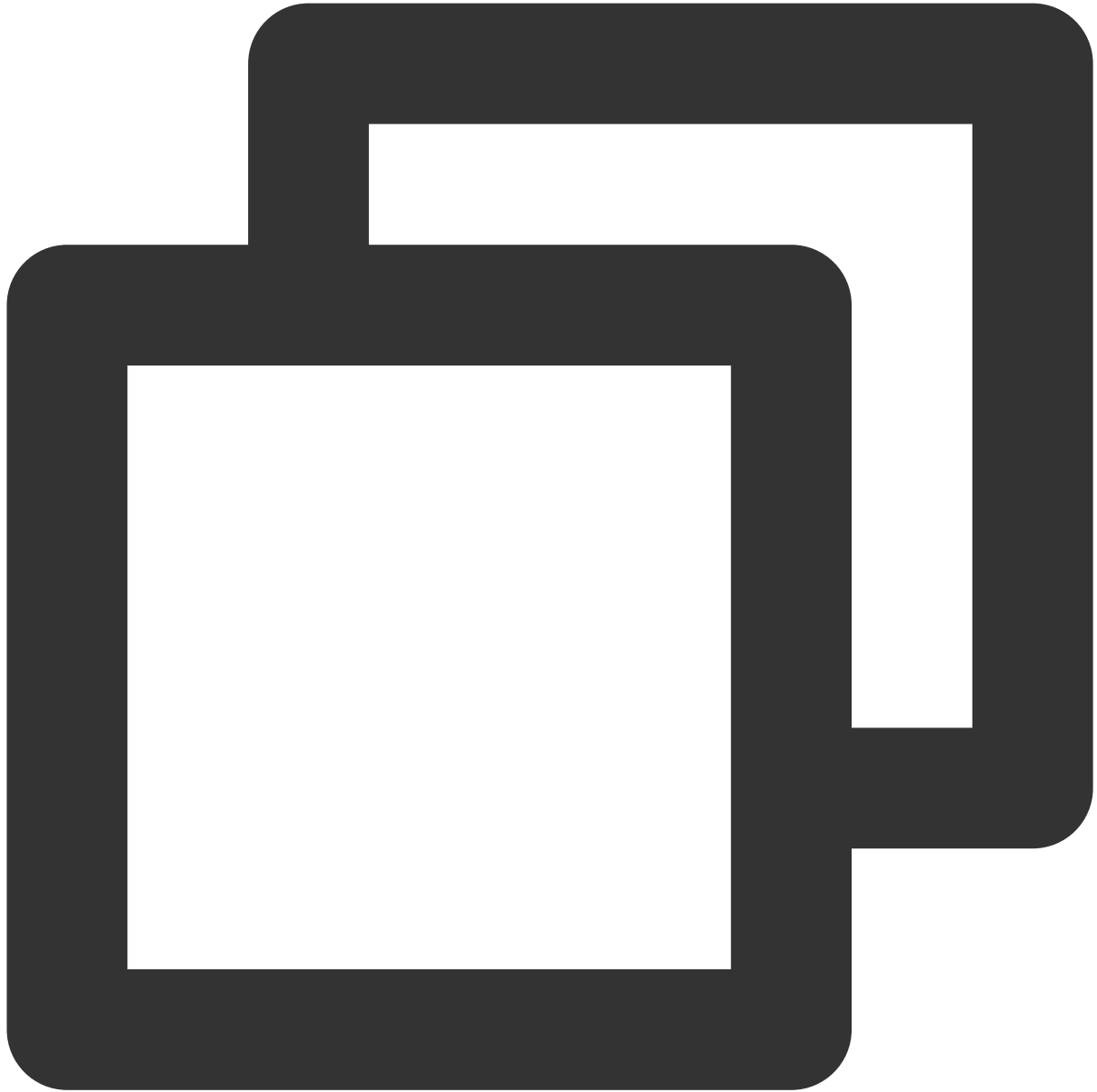


```
[root@VM_33_50_centos ~]# tccli cvm RunInstances --cli-unfold-argument --InstanceCh
```

更多使用示例

您还可通过以下命令, 进一步使用 TCCLI :

执行 `tccli help` 命令，查看支持的产品，支持中文。



```
[root@VM_33_50_centos ~]# tccli help
NAME
  tccli
DESCRIPTION
  tccli (Tencent Cloud Command Line Interface) is a tool to manage your Tencent Clo
CONFIGURE
  Before using tccli, you should use the command(tccli configure) to configure your
  formation, please enter tccli configure help
USAGE
```

```
tccli [options] <service> [options] <action> [options] [options and parameters]
OPTIONS
  help
  show the tccli help info
  --version
  show the version of tccli
AVAILABLE SERVICES
  af
  介绍如何使用API对借贷反欺诈进行操作，包括借贷反欺诈等。
  afc
  介绍如何使用API对定制建模进行操作，包括定制建模等。
  ame
  介绍如何使用API对正版曲库直通车进行操作，包括素材获取、数据上报等。
  .....
```

执行 `tccli cvm help` 命令，查看产品支持的接口。本文以 CVM 为例。



```
[root@VM_33_50_centos ~]# tccli cvm help
NAME
    cvm
AVAILABLE VERSIONS
    2017-03-12
    默认只展示最新版本信息，查看其它版本帮助信息加 --version xxxx-xx-xx
DESCRIPTION
    cvm-2017-03-12
    介绍如何使用API对云服务器进行操作，包括使用并管理实例、镜像、密钥等资源。
USAGE
    tccli cvm <action> [--param...]
```

OPTIONS

```
help
```

```
show the tccli cvm help info
```

AVAILABLE ACTIONS

```
AllocateHosts
```

```
创建CDH实例
```

```
AssociateInstancesKeyPairs
```

```
绑定密钥对
```

```
AssociateSecurityGroups
```

```
绑定安全组
```

```
.....
```

执行 `tccli cbs DescribeDisks help` 命令，查看接口支持的参数。本文以 CBS 的 `DescribeDisks` 接口为例。



```
[root@VM_33_50_centos ~]# tccli cbs DescribeDisks help
```

```
NAME
```

```
DescribeDisks
```

```
DESCRIPTION
```

```
cbs-2017-03-12-DescribeDisks
```

本接口（DescribeDisks）用于查询云硬盘列表。

* 可以根据云硬盘ID、云硬盘类型或者云硬盘状态等信息来查询云硬盘的详细信息，不同条件之间为与 (AND) 关系，见过滤器`Filter`。

* 如果参数为空，返回当前用户一定数量（`Limit`所指定的数量，默认为20）的云硬盘列表。

```
USAGE
```

```
tccli cbs DescribeDisks [--param...]
```


OPTIONS

help

show the tccli cbs DescribeDisks help info

--region

identify the region to which the instance you want to work with belongs.

--timeout

specify a request timeout

--secretKey

specify a SecretKey

.....

AVAILABLE PARAMS

--Limit (Integer | Optional)

返回数量，默认为20，最大值为100。关于`Limit`的更进一步介绍请参考 API [简介] (<https://www.tencentcloud.com/>)

--OrderField (String | Optional)

云盘列表排序的依据字段。取值范围：

CREATE_TIME：依据云盘的创建时间排序
DEADLI

默认按云盘创建时间排序。

--Offset (Integer | Optional)

偏移量，默认为0。关于`Offset`的更进一步介绍请参考API [简介] (<https://www.tencentcloud.com/>)

.....

输出格式支持 JSON、table 及 text 格式。

JSON 格式：



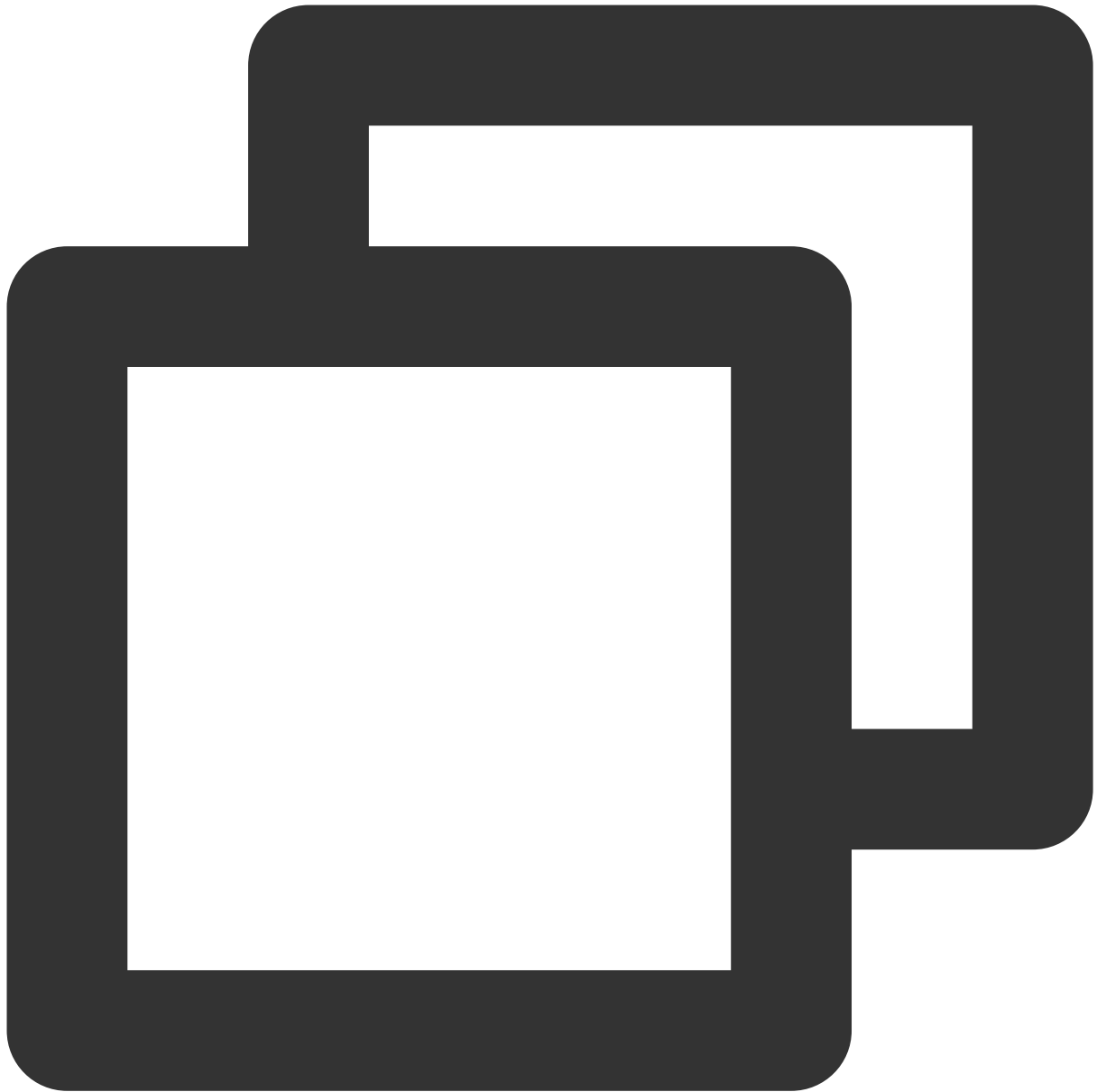
```
[root@VM_33_50_centos ~]# tccli cvm DescribeRegions
{
  "TotalCount": 20,
  "RegionSet": [
    {
      "RegionState": "AVAILABLE",
      "Region": "ap-beijing",
      "RegionName": "华北地区(北京)"
    },
    {
      "RegionState": "AVAILABLE",
```

```

    "Region": "ap-chengdu",
    "RegionName": "西南地区(成都)"
  },
  {
    "RegionState": "AVAILABLE",
    "Region": "ap-guangzhou",
    "RegionName": "华南地区(广州)"
  },
  {
    "RegionState": "AVAILABLE",
    "Region": "ap-hongkong",
    "RegionName": "港澳台地区(中国香港)"
  },
  {
    "RegionState": "AVAILABLE",
    "Region": "ap-singapore",
    "RegionName": "东南亚地区(新加坡)"
  },
  {
    "RegionState": "AVAILABLE",
    "Region": "ap-tokyo",
    "RegionName": "亚太地区(东京)"
  },
  {
    "RegionState": "AVAILABLE",
    "Region": "eu-frankfurt",
    "RegionName": "欧洲地区(法兰克福)"
  },
  .....
],
"RequestId": "e5125cf1-****-****-****-316f18eed021"
}

```

table 格式：



```
[root@VM_33_50_centos ~]# tccli cvm DescribeRegions --output table
--
|          action          |
+-----+-----+
|      RequestId      | TotalCount |
+-----+-----+
| 1af5f2a0-****-****-****-462f0271a69f | 20         |
+-----+-----+
||          RegionSet          ||
|+-----+-----+-----+
||      Region      | RegionName | RegionState ||
```

ap-bangkok	亚太地区 (曼谷)	AVAILABLE
ap-beijing	华北地区 (北京)	AVAILABLE
ap-chengdu	西南地区 (成都)	AVAILABLE
ap-chongqing	西南地区 (重庆)	AVAILABLE
ap-guangzhou	华南地区 (广州)	AVAILABLE
ap-guangzhou-open	华南地区 (广州Open)	AVAILABLE
ap-hongkong	港澳台地区 (中国香港)	AVAILABLE
ap-mumbai	亚太地区 (孟买)	AVAILABLE
ap-nanjing	华东地区 (南京)	AVAILABLE
ap-seoul	亚太地区 (首尔)	AVAILABLE
ap-shanghai	华东地区 (上海)	AVAILABLE
ap-singapore	东南亚地区 (新加坡)	AVAILABLE
ap-tokyo	亚太地区 (东京)	AVAILABLE
eu-frankfurt	欧洲地区 (法兰克福)	AVAILABLE
eu-moscow	欧洲地区 (莫斯科)	AVAILABLE
na-ashburn	美国东部 (弗吉尼亚)	AVAILABLE
na-siliconvalley	美国西部 (硅谷)	AVAILABLE
na-toronto	北美地区 (多伦多)	AVAILABLE

text 格式：



```
[root@VM_33_50_centos ~]# tccli cvm DescribeRegions --output text
70bbd02f-****-****-****-afc5c34018ae    20
REGIONSET      ap-bangkok      亚太地区(曼谷)    AVAILABLE
REGIONSET      ap-beijing       华北地区(北京)   AVAILABLE
REGIONSET      ap-chengdu       西南地区(成都)   AVAILABLE
REGIONSET      ap-chongqing     西南地区(重庆)   AVAILABLE
REGIONSET      ap-guangzhou     华南地区(广州)   AVAILABLE
REGIONSET      ap-guangzhou-open 华南地区(广州Open) AVAILABLE
REGIONSET      ap-hongkong      港澳台地区(中国香港) AVAILABLE
REGIONSET      ap-mumbai        亚太地区(孟买)   AVAILABLE
REGIONSET      ap-nanjing       华东地区(南京)   AVAILABLE
```

REGIONSET	ap-seoul	亚太地区(首尔)	AVAILABLE
REGIONSET	ap-shanghai	华东地区(上海)	AVAILABLE
REGIONSET	ap-singapore	东南亚地区(新加坡)	AVAILABLE
REGIONSET	ap-tokyo	亚太地区(东京)	AVAILABLE
REGIONSET	eu-frankfurt	欧洲地区(法兰克福)	AVAILABLE
REGIONSET	eu-moscow	欧洲地区(莫斯科)	AVAILABLE
REGIONSET	na-ashburn	美国东部(弗吉尼亚)	AVAILABLE
REGIONSET	na-siliconvalley	美国西部(硅谷)	AVAILABLE
REGIONSET	na-toronto	北美地区(多伦多)	AVAILABLE

高级功能

该步骤以 CVM 为例，详细介绍了如何使用 TCCLI 高级功能，包括多版本接口访问、指定最近的接入点、返回结果过滤、输出入参数数据结构到 JSON 文件以及从 JSON 文件读取参数等。

多版本接口访问

某些产品可能存在多个版本的接口，TCCLI 默认访问最新版本的接口。如果您想访问特定旧版本的接口，可以通过以下方式实现：

方式1：设置 CVM 产品默认使用版本：2017-03-12



```
tccli configure set cvm.version 2017-03-12
```

方式2：在实时使用时指定版本号

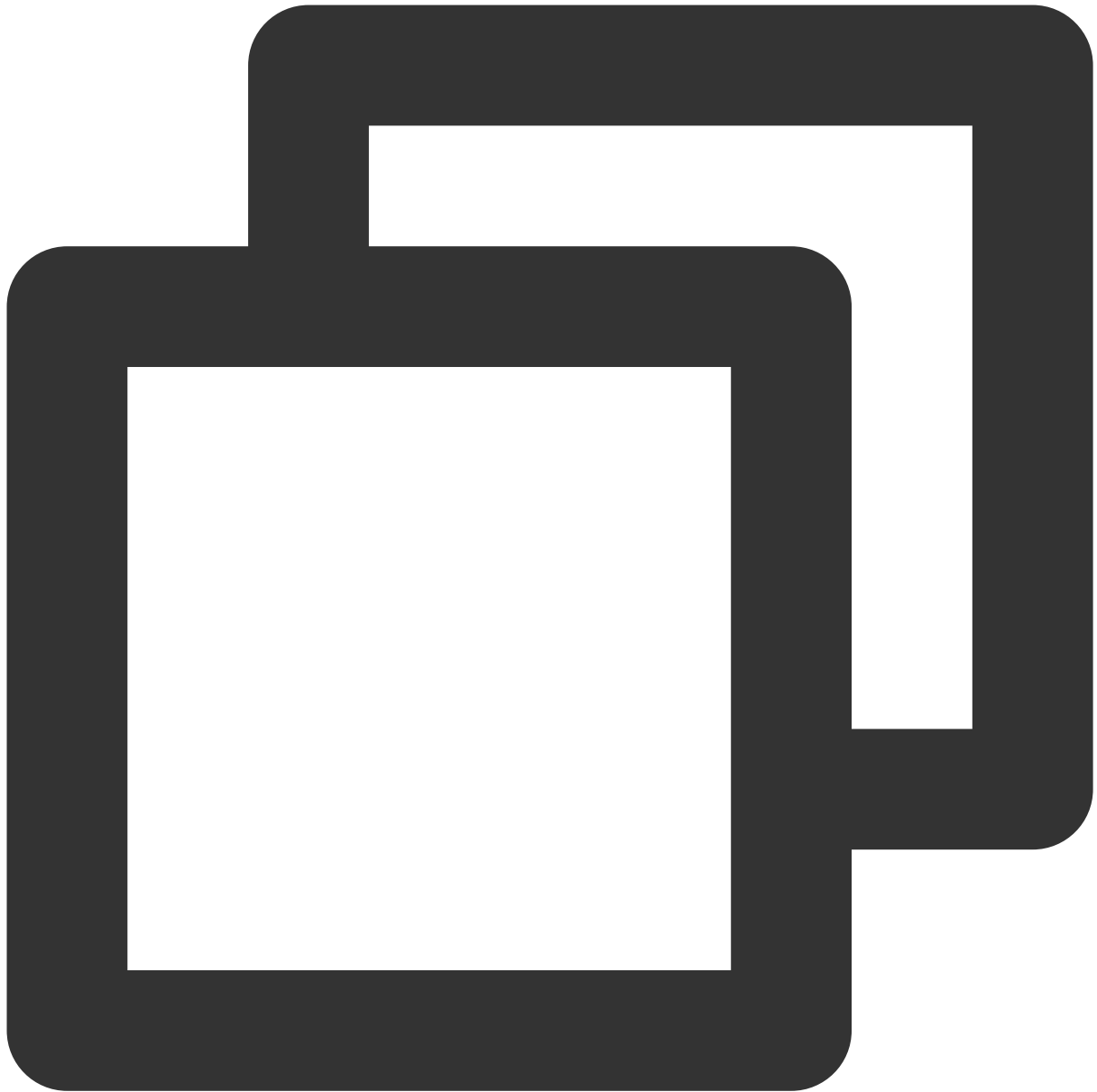


```
tccli cvm help --version 2017-03-12
tccli cvm DescribeZones help --version 2017-03-12
tccli cvm DescribeZones --version 2017-03-12
```

指定最近的接入点 (Endpoint)

TCCLI 默认会请求就近的接口点访问服务，您也可以针对某一产品指定自己的 Endpoint。

设置 CVM 产品默认 endpoint



```
tccli configure set cvm.endpoint cvm.ap-guangzhou.tencentcloudapi.com
```

调用时实时指定



```
tccli cvm DescribeZones --endpoint cvm.ap-guangzhou.tencentcloudapi.com
```

返回结果过滤

以 CVM DescribeZones 接口的返回为例，不加任何过滤时的输出。



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones
{
  "TotalCount": 4,
  "ZoneSet": [
    {
      "ZoneState": "AVAILABLE",
      "ZoneId": "100001",
      "Zone": "ap-guangzhou-1",
      "ZoneName": "广州一区"
    },
    {
```

```
    "ZoneState": "AVAILABLE",
    "ZoneId": "100002",
    "Zone": "ap-guangzhou-2",
    "ZoneName": "广州二区"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100003",
    "Zone": "ap-guangzhou-3",
    "ZoneName": "广州三区"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100004",
    "Zone": "ap-guangzhou-4",
    "ZoneName": "广州四区"
  }
],
"RequestId": "4fd313a6-****-****-****-898c02fcae02"
}
```

仅查看某个字段。



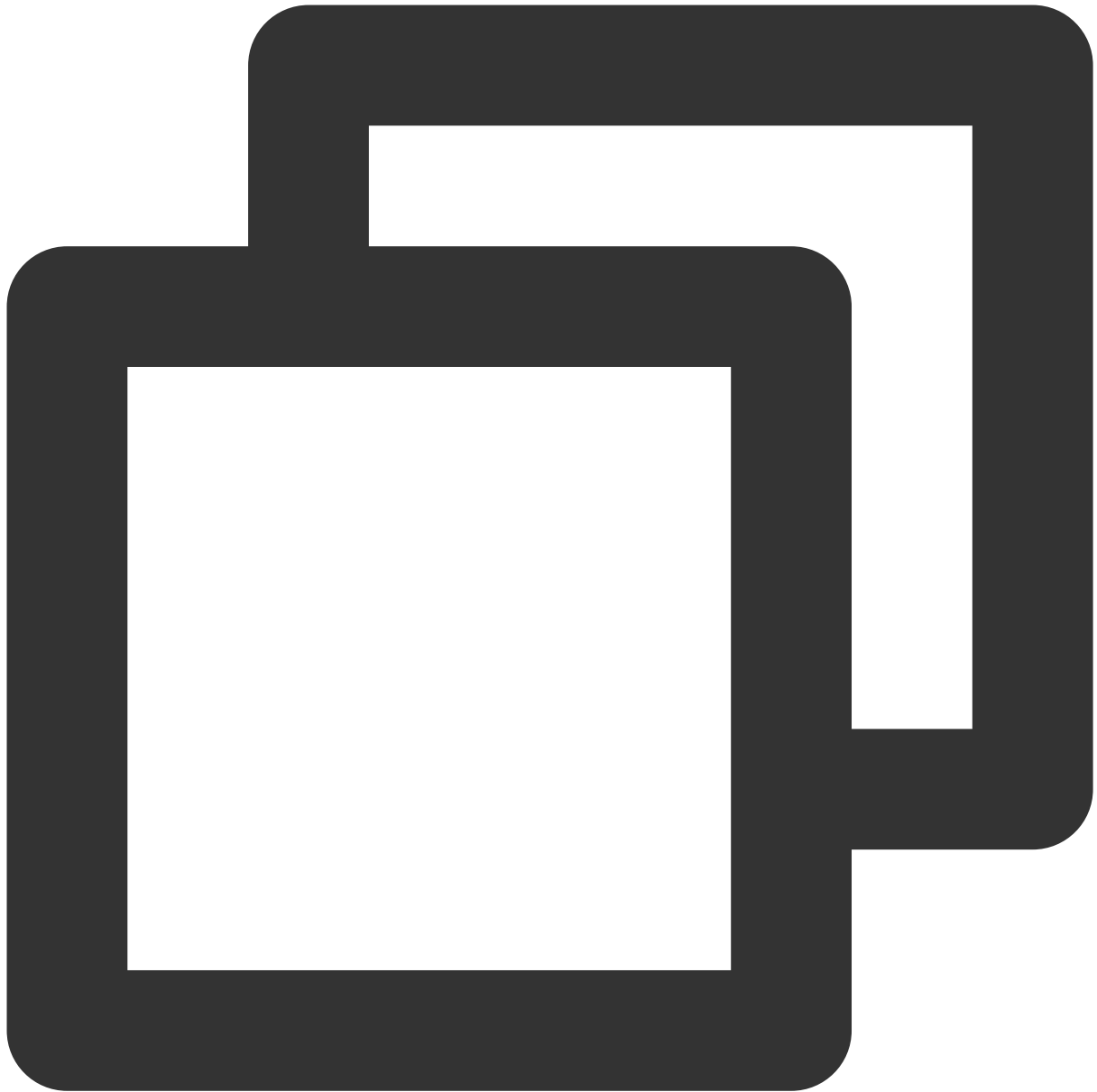
```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter TotalCount  
4
```

指定某个数组类型对象的第 N 个子对象的信息。



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[0]
{
  "ZoneState": "AVAILABLE",
  "ZoneId": "100001",
  "Zone": "ap-guangzhou-1",
  "ZoneName": "广州一区"
}
```

指定数组类型对象下所有某个名称的子对象的某个字段。

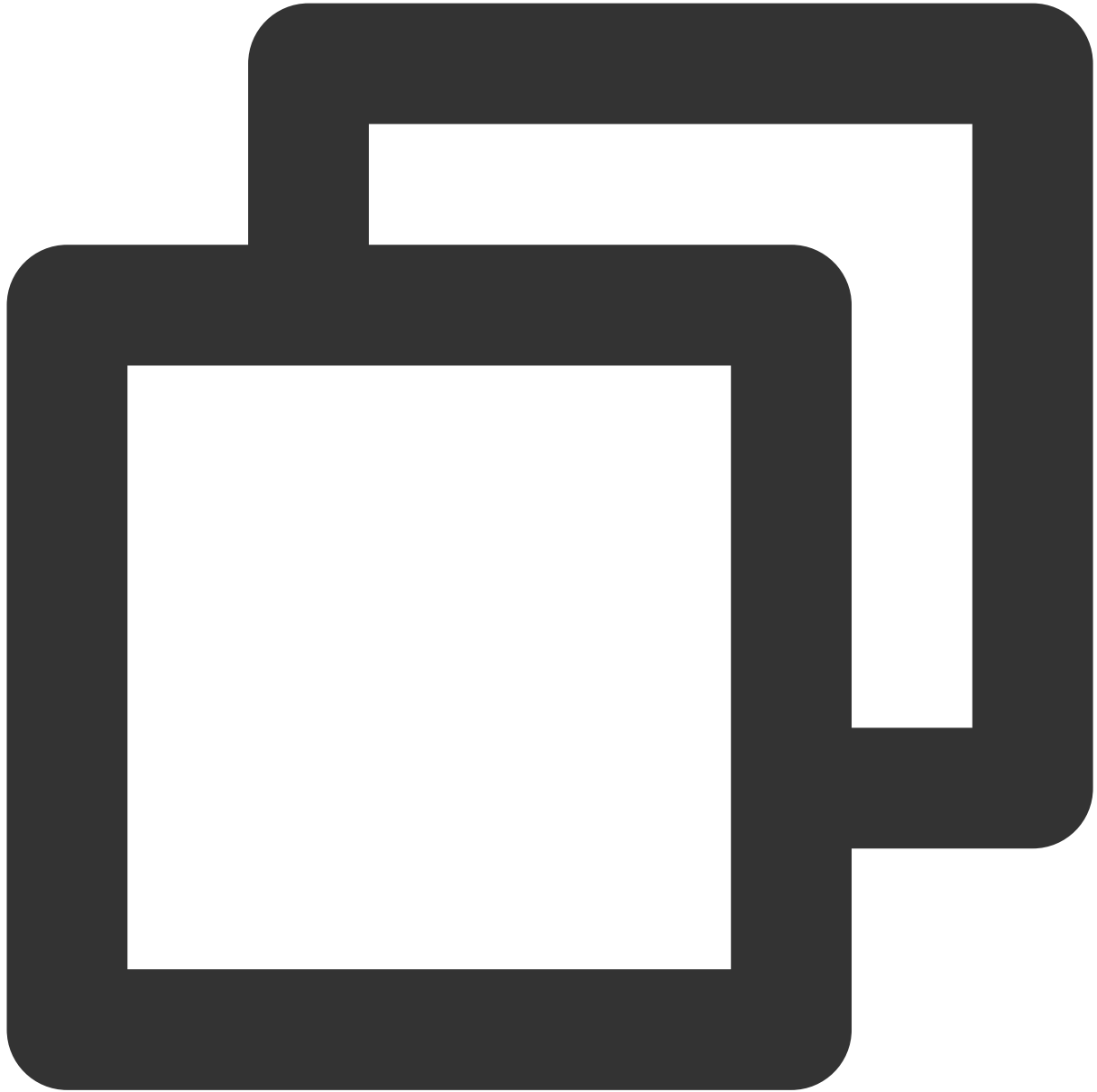


```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[*].ZoneName  
[  
  "广州一区",  
  "广州二区",  
  "广州三区",  
  "广州四区"  
]
```

过滤数组里的子对象，同时还以新的名称展示。

注意：

需要将说明过滤行为的内容用单引号包裹起来。



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter 'ZoneSet[*].{name:Zo
[
  {
    "name": "广州一区",
    "id": "100001"
  },
  {
    "name": "广州二区",
    "id": "100002"
  }
]
```

```
    },
    {
      "name": "广州三区",
      "id": "100003"
    },
    {
      "name": "广州四区",
      "id": "100004"
    }
  ]
```

相关问题

如何购买命令行工具？

本服务免费。当您遇到问题时，请 [提交工单](#) 寻求相应的帮助。

如何实现接口鉴权？

在 API 支持的每个产品文档目录下，可选择【[调用方法](#) > [接口鉴权](#)】，结合产品的“接口鉴权”文档进行实现。例如，可前往 [CVM 接口鉴权](#) 进行查看。

使用 API Inspector

最近更新时间：2023-03-07 18:16:40

API Inspector 是云 API 新推出的一款产品功能，用户可通过此功能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。

注意：

API Inspector 仅展示公开的 [云 API 3.0](#) 接口信息。

API Inspector 功能目前处于控制台和用户双维度灰度阶段。

控制台目前仅在云服务器控制台的 [实例](#)、[专用宿主机](#)、[置放群组](#)、[弹性伸缩](#)、[SSH密钥](#) 以及 [回收站](#) 菜单中开放。

刷新浏览器当前页面，则会清空刷新前的调用记录。

功能特性

API Inspector 与 API Explorer 共同成为腾讯云 API 用户学习和调试 API 的一体化解决方案，具有以下特性：

自动录制：如果您想要了解功能背后的 API，可在控制台操作相应的功能时获得相关 API 调用信息，详情请参见 [自动录制 API 调用](#)。

一键生成：自动生成各语言的 API 代码片段参数预填充，可直接运行，详情请参见 [一键生成 API 代码](#)。

在线调试：提供 API Explorer 一键在线调试功能，可实现自动生成多语言 SDK 代码、在线调用、发送真实请求及签名串自动生成等功能。降低了 SDK 的使用难度，详情请参见 [API Explorer 在线调试](#)。

功能介绍

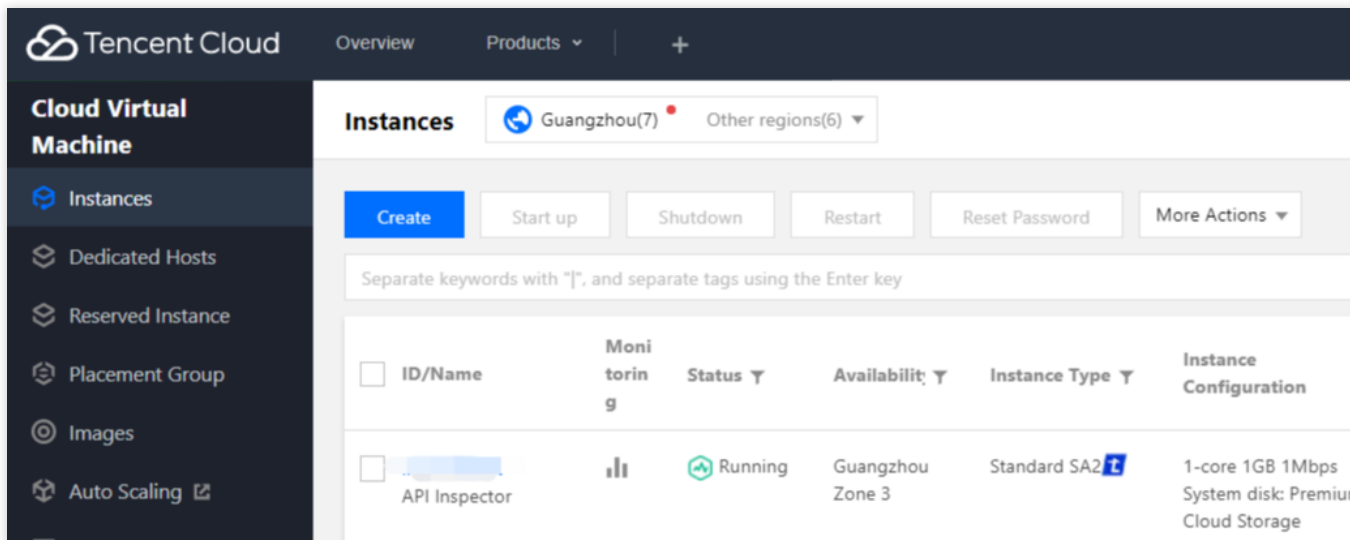
开启 API Inspector 功能

请按照以下步骤，开启 API Inspector 功能：

1. 登录 [云服务器控制台](#)。
2. 选择页面上方的



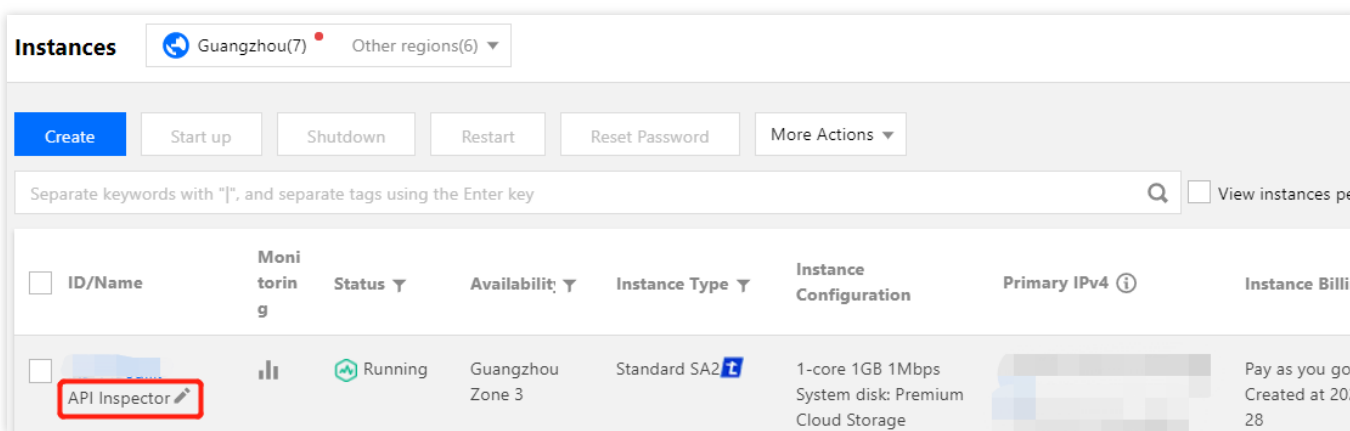
> **API**，即可开启 API Inspector 功能。如下图所示：



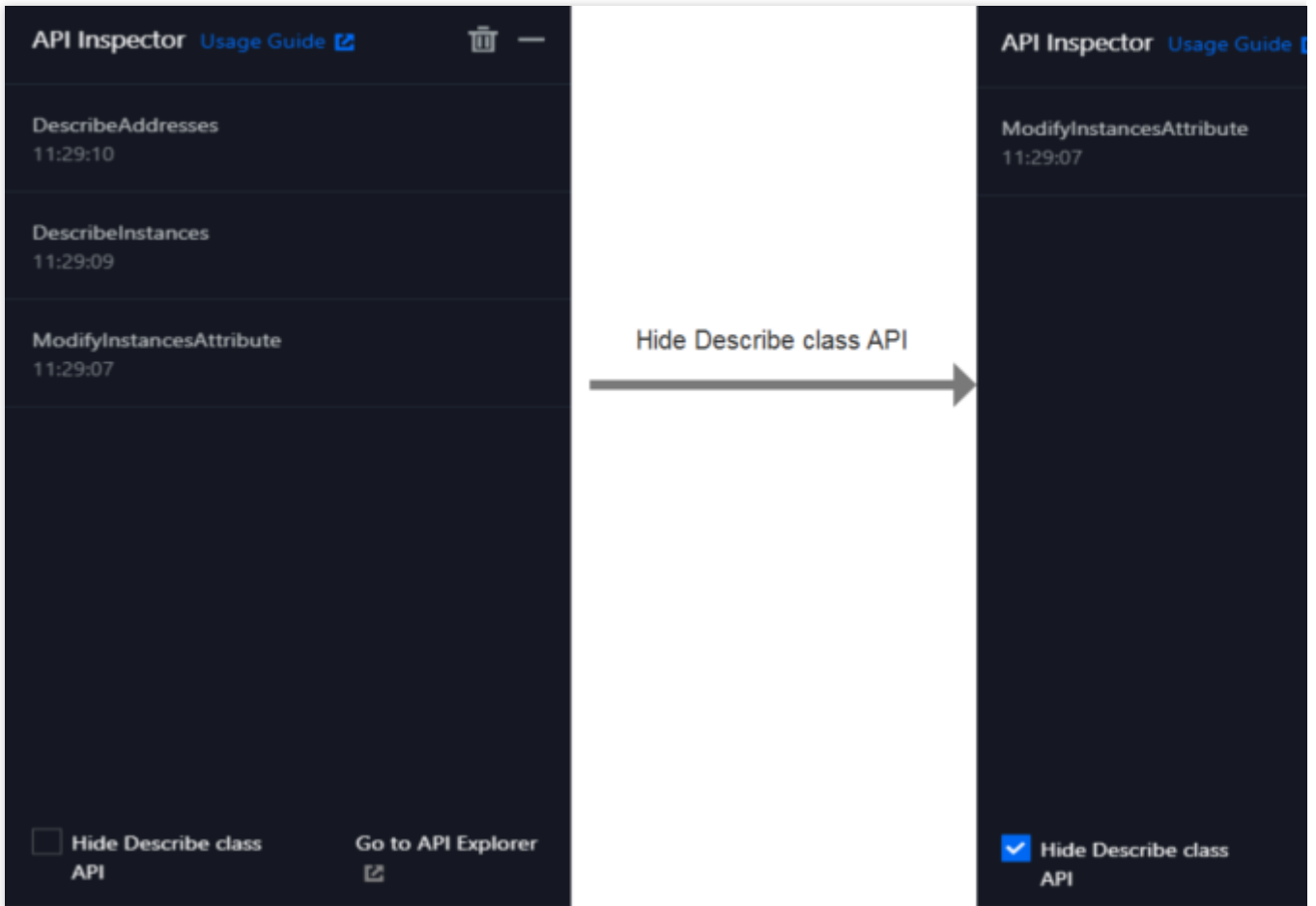
自动录制 API 调用

本文以修改实例名称为例，介绍 API Inspector 的自动录制功能：

1. 将某个实例名称修改为 API Inspector，具体操作请参见 [修改实例名称](#)。
2. 开启 API Inspector 功能，即可查看涉及改名操作的所有 API 调用。如下图所示：




您可勾选“隐藏Describe类接口”，查看功能核心接口。如下图所示：



一键生成 API 代码

当控制台操作涉及的 API 录制完成后，您可单击 API 名称，一键生成 Java、Python、Node.js、PHP、GO 及 .NET 语言的 API 代码片段及参数预填充。可选择

 复制对应格式的代码段，如下图所示：

ModifyInstancesAttribute

API Inspector [Usage](#)

product cvm

action ModifyInstancesAttribute

params

Region: ap-guangzhou

InstanceIds: ["ins-1498dilk"]

InstanceName: "API Inspector"

Python
Java
Go
Node.js
PHP
.Net

```

import json
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
from tencentcloud.cvm.v20170312 import cvm_client, models

try:
    cred = credential.Credential("", "")
    httpProfile = HttpProfile()
    httpProfile.endpoint = "cvm.tencentcloudapi.com"

    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile
            
```

Hide Describe class API




API Explorer 在线调试

您可选择

或前往 **API Explorer**，使用 API Explorer 工具直接调试对应的功能，也可选择

查看对应接口文档。如下图所示：

ModifyInstancesAttribute

API Inspector [Usage](#)

product cvm

action ModifyInstancesAttribute

params


Region: ap-guangzhou

InstanceIds: ["ins-1498dilK"]

InstanceName: "API Inspector"

ModifyInstancesAttribu

11:29:07

Python
Java
Go
Node.js
PHP
.Net


```

import json
from tencentcloud.common import credential
from tencentcloud.common.profile.client_profile import ClientProfile
from tencentcloud.common.profile.http_profile import HttpProfile
from tencentcloud.common.exception.tencent_cloud_sdk_exception import TencentCloudSDKException
from tencentcloud.cvm.v20170312 import cvm_client, models
try:
    cred = credential.Credential("", "")
    httpProfile = HttpProfile()
    httpProfile.endpoint = "cvm.tencentcloudapi.com"

    clientProfile = ClientProfile()
    clientProfile.httpProfile = httpProfile

```

Hide Describe class API