

命令行工具 操作指南 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

操作指南

- 安装 TCCLI

- 配置 TCCLI

 - TCCLI 配置方法

 - 使用 HTTPS 代理

 - 使用命令行自动补全功能

- 使用 TCCLI

 - TCCLI 使用方法

 - 获取帮助信息

 - 多版本接口访问

 - 指定最近接入点

 - 结果返回过滤

 - 结果轮询

- 使用高级功能

操作指南

安装 TCCLI

最近更新时间：2023-10-07 17:12:40

本文介绍如何安装 TCCLI。

前提条件

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。

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。如需查看 TCCLI 最新版本相关信息，请前往 [github 仓库](#)。

操作步骤

您可以通过以下三种方式进行安装：

使用 pip 工具进行安装（推荐）

使用源码进行安装

使用 Homebrew 进行安装

1. 安装命令在 Windows、Mac OS 以及 Linux 上一致，请打开您使用系统的命令行窗口。
2. 在命令行窗口中，执行以下命令进行 TCCLI 安装。



```
sudo pip install tccli
```

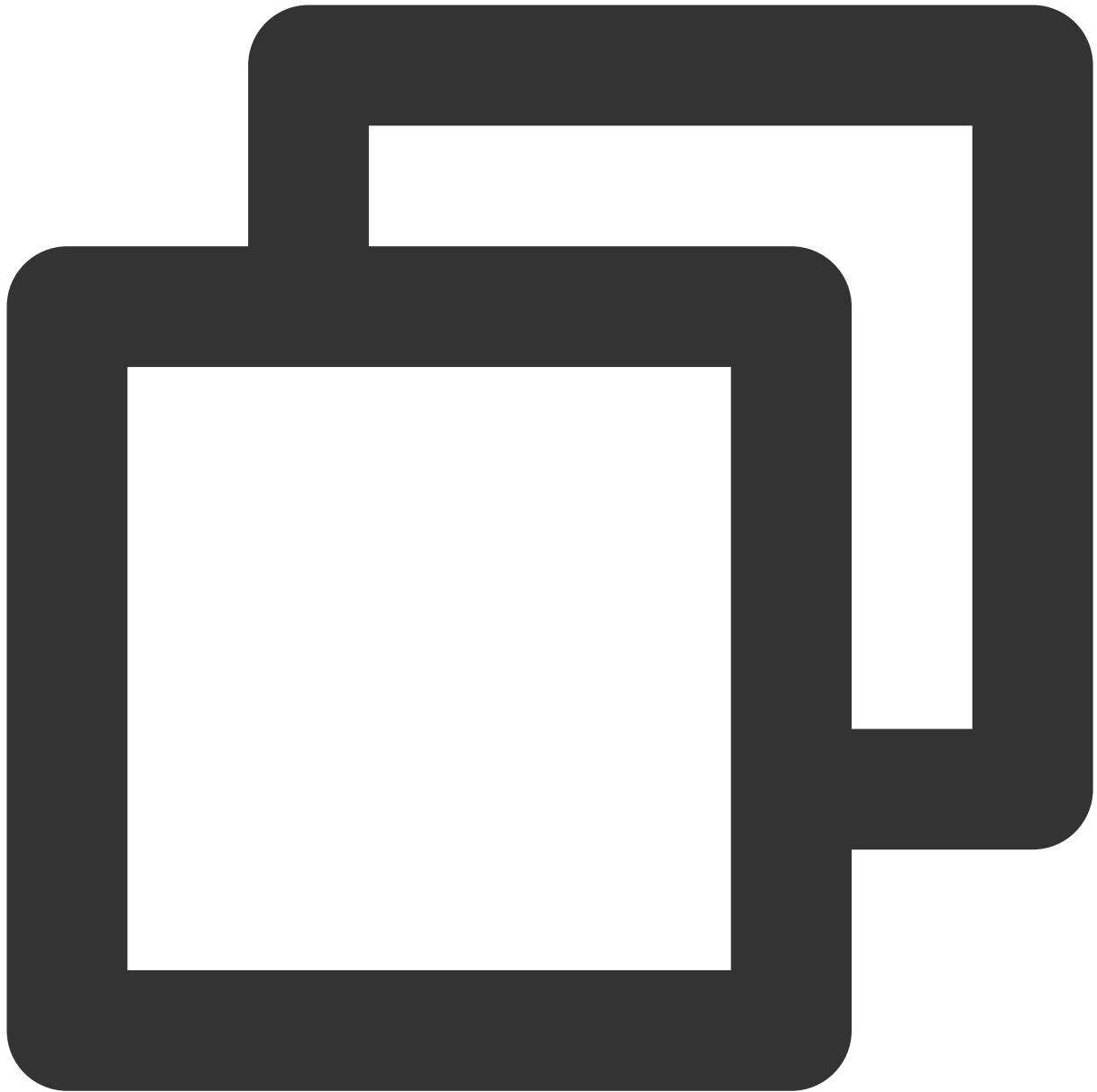
注意：

如果您是从3.0.252.3以下版本进行升级，需要执行以下命令：



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

3. 安装完成之后，执行以下命令，检测是否安装成功。



```
tccli --version
```

依次执行以下命令，下载 [tencentcloud-cli](#) 仓库中的 TCCLI 项目，并执行 `setup.py` 脚本安装。



```
git clone https://github.com/TencentCloud/tencentcloud-cli.git
```




```
cd tencentcloud-cli
```



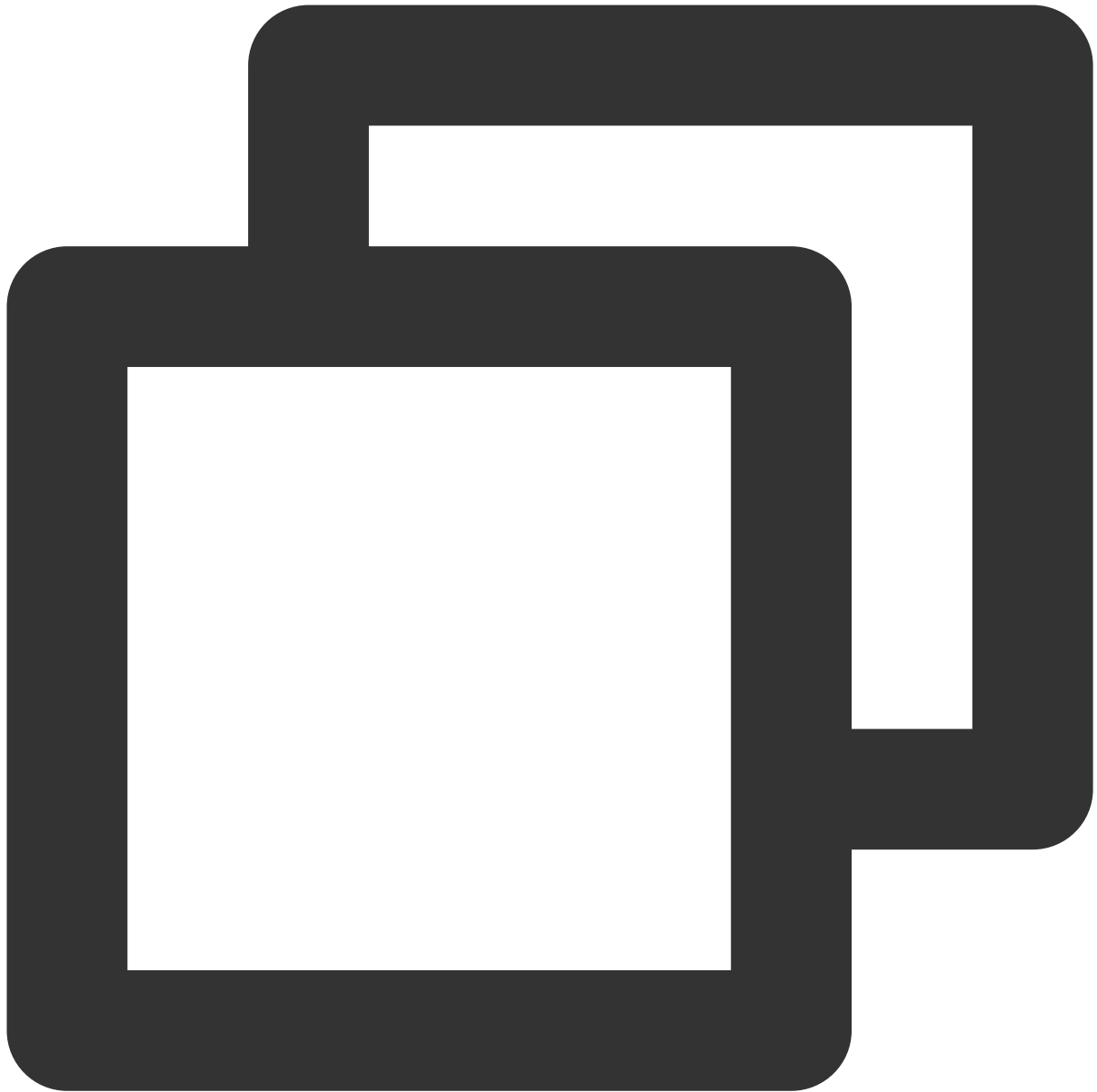
```
python setup.py install
```

注意：

该方法只适用于 Mac OS 系统。您需要先安装 [homebrew](#)，安装方法请参见 [homebrew 官网](#)。
依次执行以下命令，安装 TCCLI。

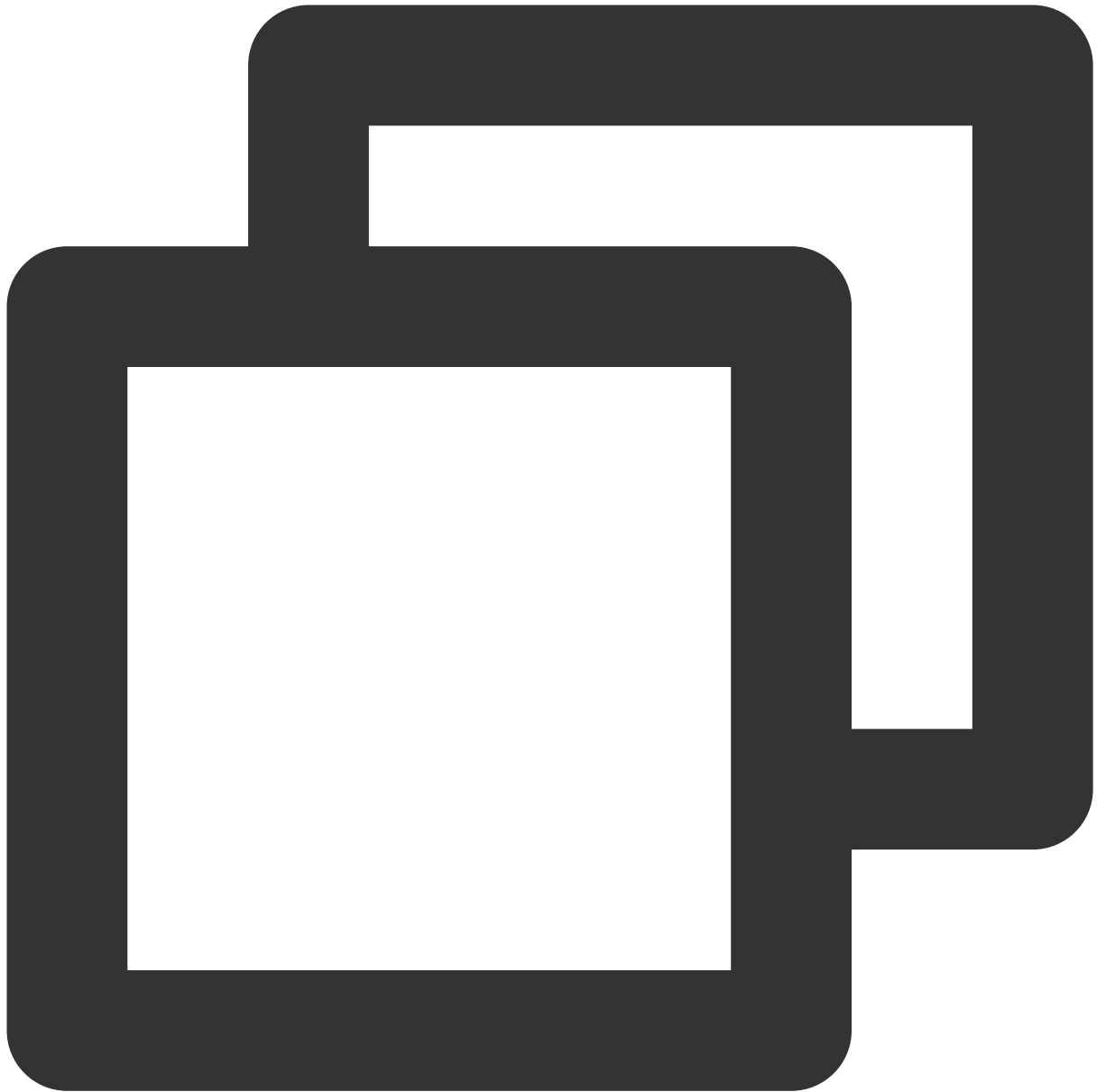


```
brew tap tencentcloud/tccli
```



```
brew install tccli
```

您可以执行以下命令，更新 TCCLI。



```
brew upgrade
```

配置 TCCLI

TCCLI 配置方法

最近更新时间：2023-02-21 10:41:35

本文指导您如何配置初始化 TCCLI，包括如何使用交互模式、命令行模式进行初始化，以及如何如何进行账户切换。

前提条件

已安装 TCCLI，详情请参见 [安装 TCCLI](#)。

操作步骤

使用 TCCLI 前您需要进行一些初始化配置，使其完成使用云 API 的必要前提条件。

说明：

下文中 `secretId`、`secretKey`、地域等信息仅作为示例，请以实际情况为准。

交互模式

您可以通过执行 `tccli configure` 命令进入交互模式快速配置。



```
$ tccli configure
TencentCloud API secretId [*afcQ]:AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
TencentCloud API secretKey [*ArFd]:Gu5t9xGARNpq86cd98joQYCN3*****
region: ap-guangzhou
output [json]:
```

secretId : 云 API 密钥 SecretId, 前往 [API 密钥管理](#) 获取。

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

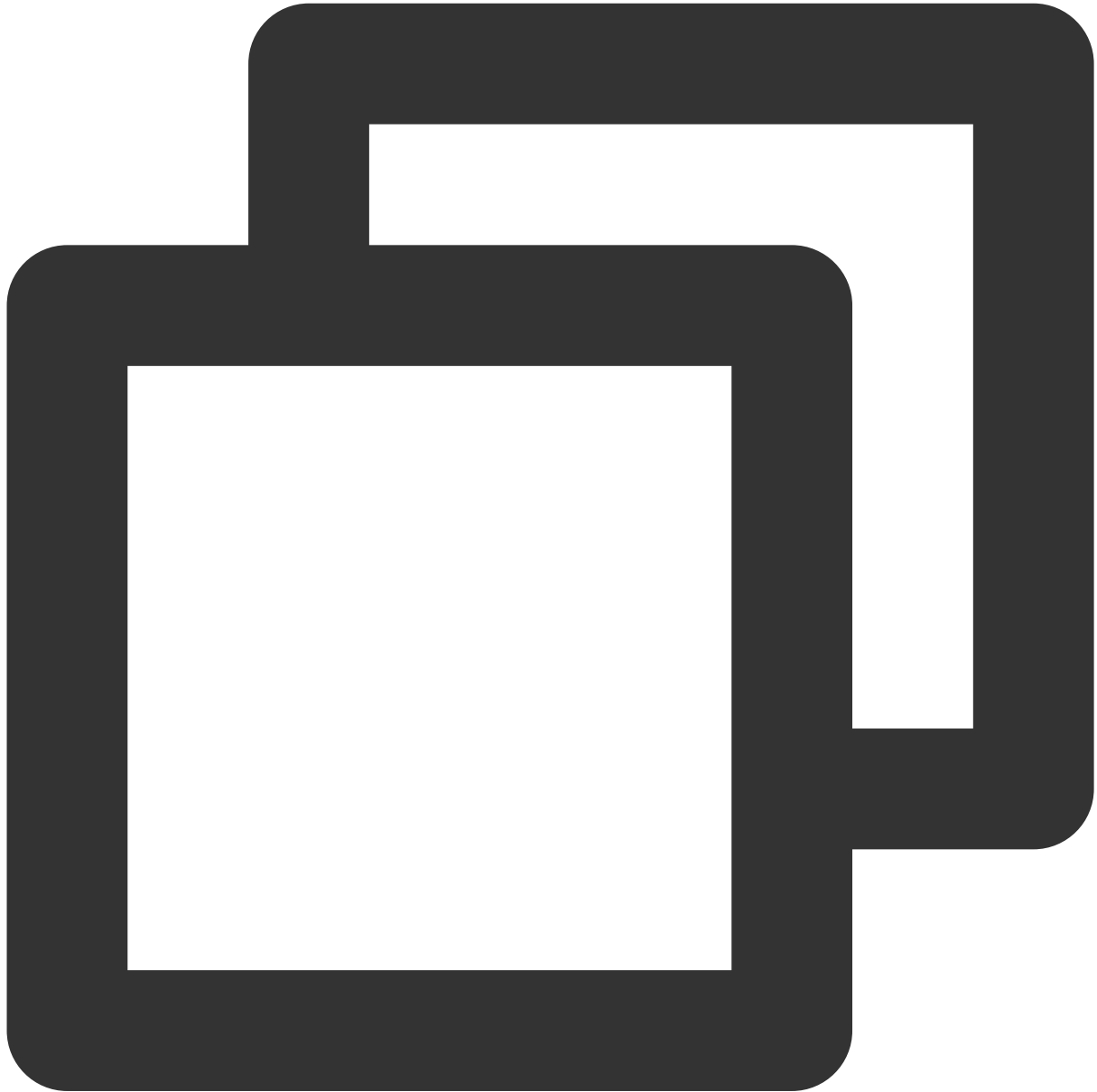
region : 云产品地域, 请前往对应云产品的 [API 文档](#) 获取可用的 region。例如云服务器的 [地域列表](#)。

output : 可选参数, 请求回包输出格式, 支持 [json table text] 三种格式, 默认为 json。

更多信息请执行 `tccli configure help` 查看。

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 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
$ 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 = Gu5t9xGARNpq86cd98joQYCN3*****
```



```
# list 子命令打印所有配置信息
$ tccli configure list
credential:
secretId = AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secretKey = Gu5t9xGARNpq86cd98joQYCN3*****
configure:
region = ap-guangzhou
output = json
# 在命令行中直接指定secretId和secretKey的值，如查询cvm实例信息：
$ tccli cvm DescribeInstances --secretId AKIDz8krbsJ5yKBZQpn74WFkmLPx3***** --secr
```

多账户支持

TCCLI 支持多账户，方便您多种配置同时使用。



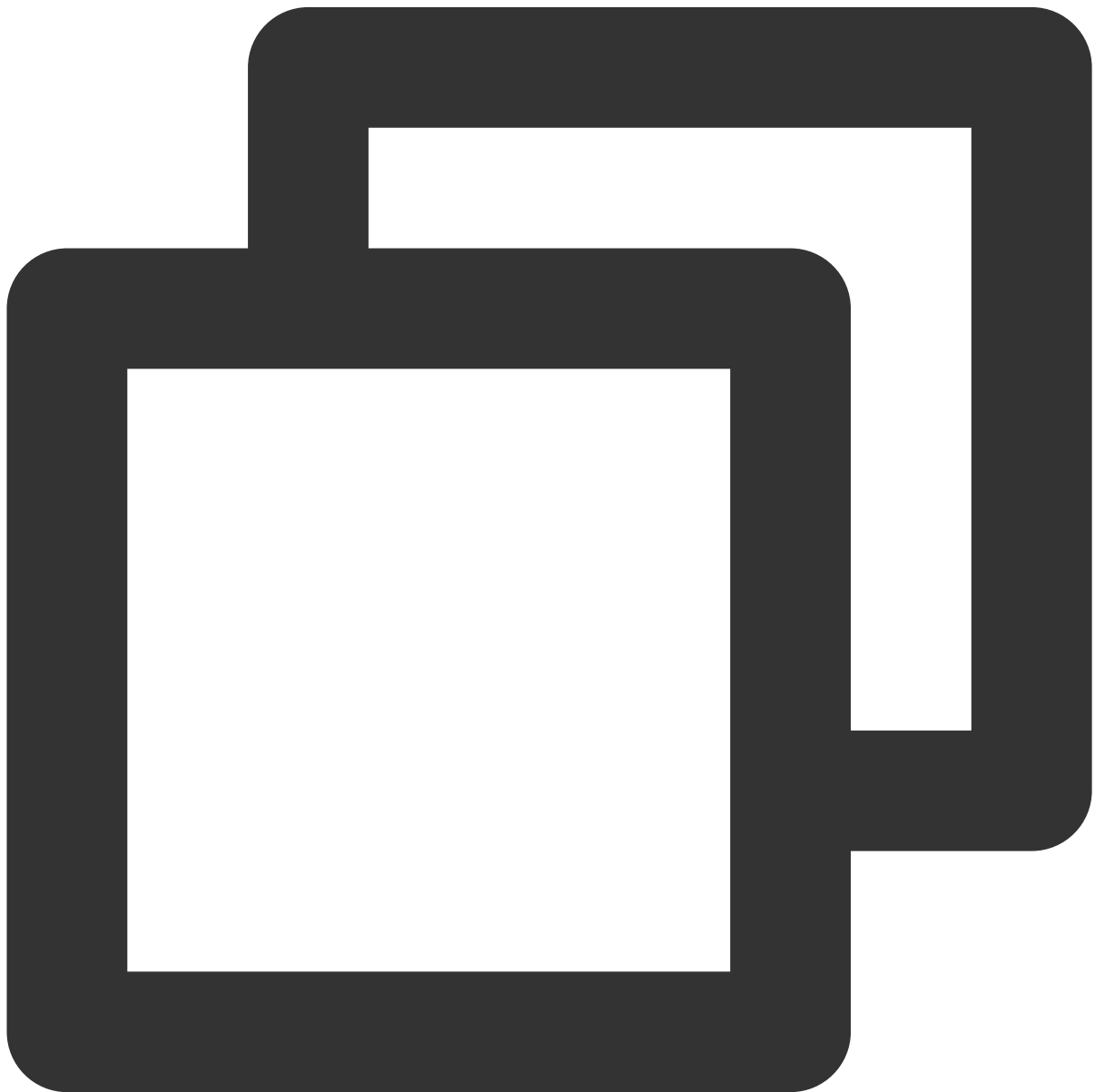
```
# 在交互模式中指定账户名 test
$ tccli configure --profile test
TencentCloud API secretId [*BCDP]:AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
TencentCloud API secretKey [*ArFd]:Gu5t9xGARNpq86cd98joQYCN3*****
region: ap-guangzhou
output[json]:
# set/get/list 子命令指定账户名 test
$ tccli configure set region ap-guangzhou output json --profile test
$ tccli configure get secretKey --profile test
$ tccli configure list --profile test
# remove 子命令删除指定账户的配置文件。当 remove 子命令不指定账户名时，会删除 default 配置文件。
```

```
$ tccli configure remove -profile test
# 在调用接口时指定账户（以 cvm DescribeZones 接口为例）
$ tccli cvm DescribeZones --profile test
```

TCCLI 配置文件

使用 `tccli configure` 命令后，TCCLI 会在 `~/.tccli` 目录下生成对应的 `default.configure` 和 `default.credential` 文件，这两个文件记录均为 JSON 格式的内容。

`default.configure` 记录对应产品调用的版本（默认最新版）和 `endpoint`（默认为最近接入点），以及默认的输出格式和指定的地域。`default.credential` 则记录的是用户的密钥信息。示例如下：



```
# default.configure文件格式, cvm产品默认调用版本 2017-03-12 的接口, 默认的请求域名为 cvm.ten
{
  ...
  "cvm": {
    "endpoint": "cvm.tencentcloudapi.com",
    "version": "2017-03-12"
  },
  ...
  "output": "json",
  "region": "ap-guanzhou",
  ...
}
# default.credential的文件格式
{
  "secretId": "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****",
  "secretKey": "Gu5t9xGARNpq86cd98joQYCN3*****"
}
```

若用户在使用配置命令时指定了账户名, 则会生成账户名对应的配置文件。例如, 用户使用了 `tccli configure --profile test` 命令, 则会生成 `test.configure` 和 `test.credential` 文件。

若需修改配置文件内容, 您可以直接编辑文件或者使用 `set` 子命令修改。例如 `tccli configure set cvm.version 2017-03-12`, 将调用的 `cvm` 的版本指定为默认 `2017-03-12` 版本。

配置环境变量

TCCLI 支持配置云 API 密钥到环境变量, 让您的信息更安全。下文以 Linux 系统配置为例, 您可以通过如下两种方式进行配置:

使用 `export` 命令 (临时性)

写入 `profile` 文件 (永久性)



```
# 设置云 API 密钥 SecretId
$ export TENCENTCLOUD_SECRET_ID=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
# 设置云 API 密钥 SecretKey
$ export TENCENTCLOUD_SECRET_KEY=Gu5t9xGARNpq86cd98joQYCN3*****
# 设置云产品地域
$ export TENCENTCLOUD_REGION=ap-guangzhou
```



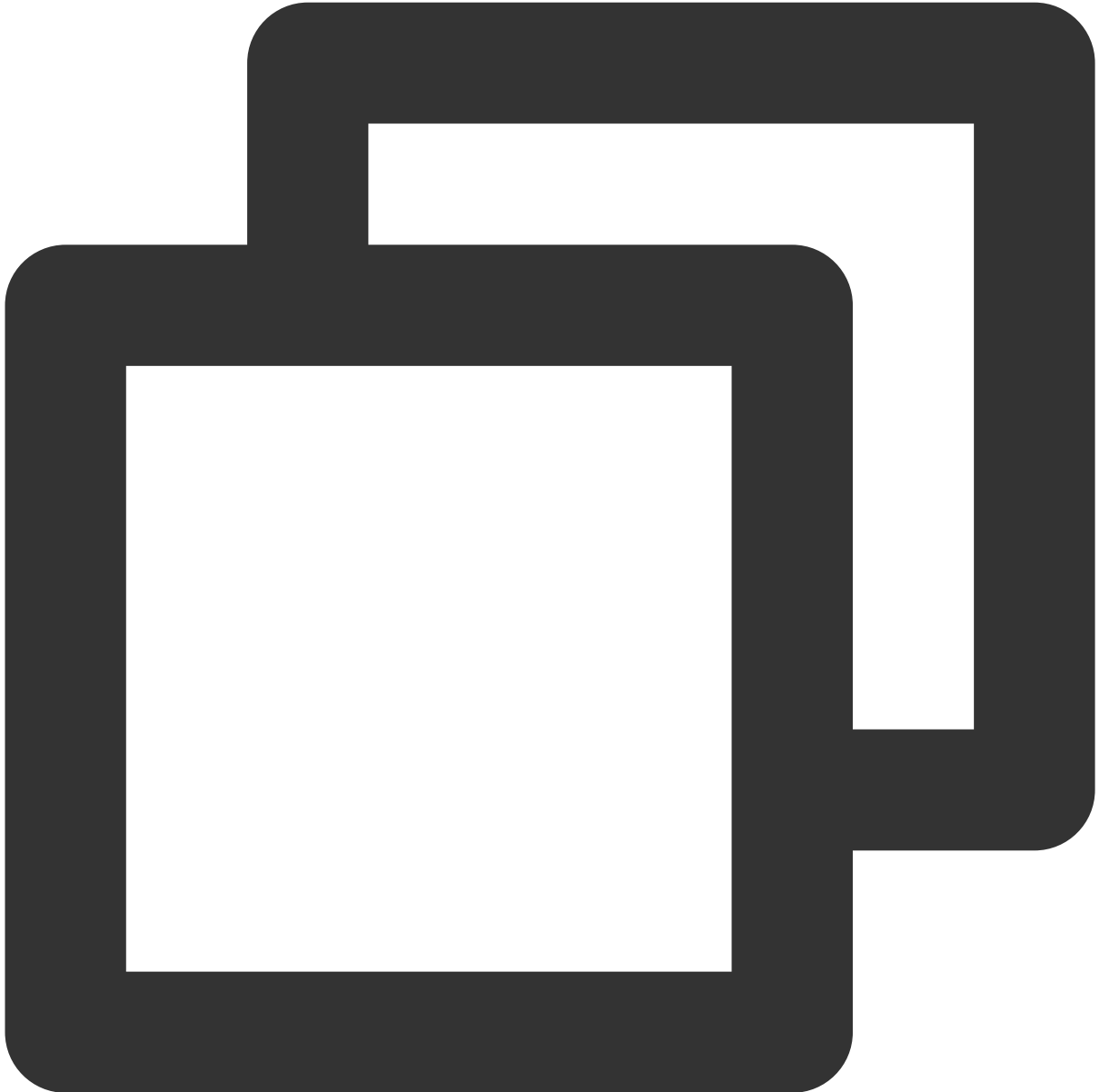
```
# 编辑 /etc/profile 文件，写入如下内容
export TENCENTCLOUD_SECRET_ID=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
export TENCENTCLOUD_SECRET_KEY=Gu5t9xGARNpq86cd98joQYCN3*****
export TENCENTCLOUD_REGION=ap-guangzhou
# 写入后需执行如下命令使环境变量生效
$ source /etc/profile
```

说明：

如果您在命令、配置文件和环境变量都指定了相同的内容，TCCLI 执行的优先级为：命令 > 配置文件 > 环境变量。

其他配置

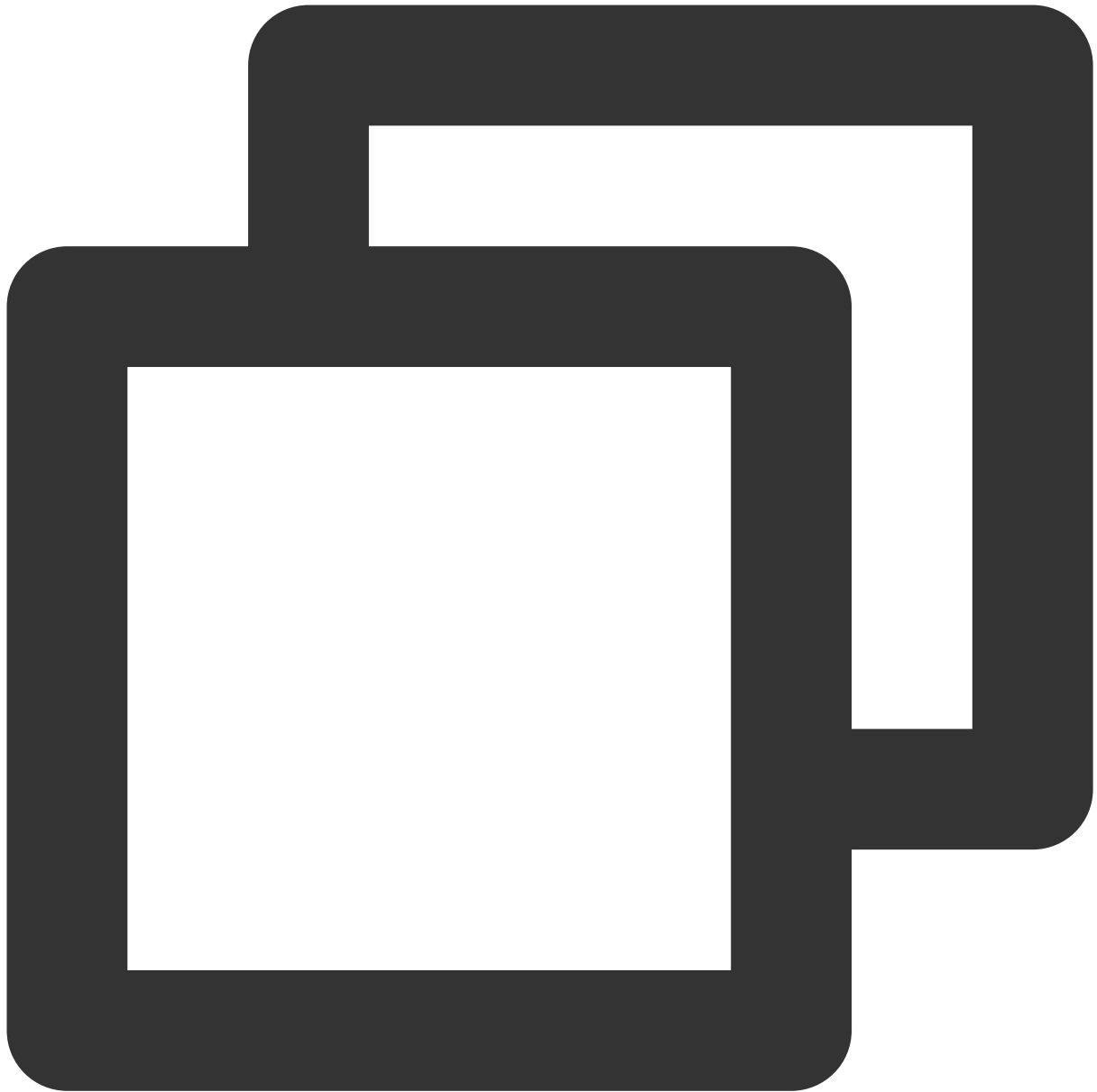
TCCLI 支持通过 CAM 角色的方式进行认证，您可以参考 [角色概述](#) 查看相关信息。



cam 角色的配置不支持交互模式，您可以使用非交互模式的方式进行配置：

```
$ tccli configure set role-arn qcs::cam::uin/*****/**** role-session-name ***
```

`role-arn` 和 `role-session-name` 字段支持 `configure` 的 `get` 和 `list` 操作，可以写入配置文件、直接在命令行指定，操作方式与 `secretId` 和 `secretKey` 的配置类似。如下所示：

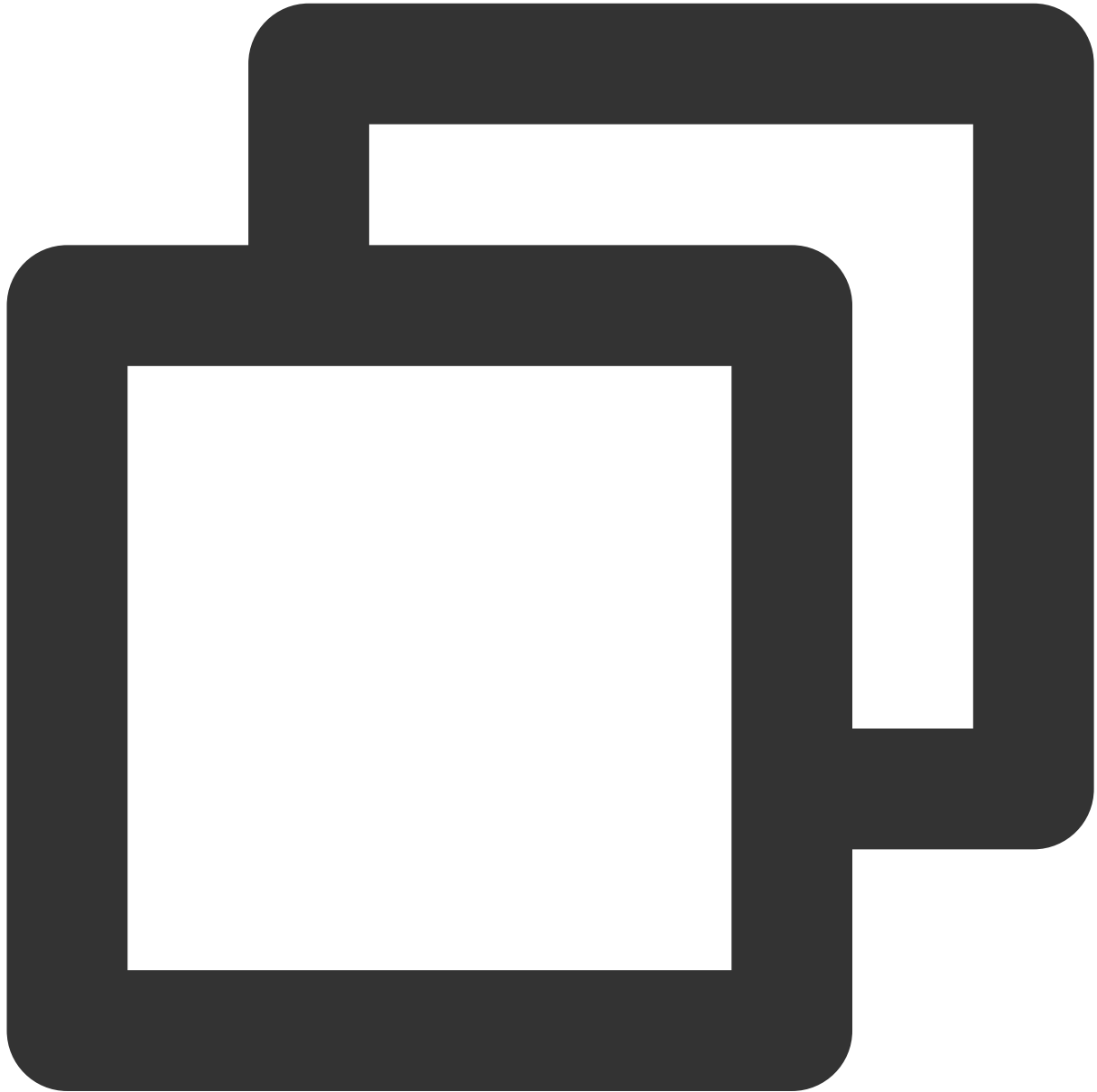


```
# get 子命令获取配置信息
$ tccli configure get role-arn
role-arn = qcs::cam::uin/*****/****
# list 子命令打印所有配置信息
$ tccli configure list
credential:
role-arn = qcs::cam::uin/*****/****
role-session-name = ****
# 将配置信息写入环境变量
$ export TENCENTCLOUD_ROLE_ARN=qcs::cam::uin/*****/****
$ export TENCENTCLOUD_ROLE_SESSION_NAME=****
```



```
# 直接在命令行中指定 role-arn 和 role-session-name 信息，如调用 DescribeZones 接口
$ tccli cvm DescribeZones --role-arn qcs::cam::uin/*****/**** --role-session-
```

如果您的实例绑定了角色，您可以直接通过实例角色的方式进行认证，无需 `secretId` 和 `secretKey` 等信息。您可以使用 `--use-cvm-role` 来使用实例角色的方式调用。



```
# 使用实例角色的方式调用 DescribeZones 的接口
$ tccli cvm DescribeZones --use-cvm-role
```

注意：

该方式仅支持在已绑定角色的实例上使用，具体方式请参见 [管理实例角色](#)。

使用 HTTPS 代理

最近更新时间：2023-02-21 10:41:36

本文介绍如何配置和使用 HTTPS 代理。

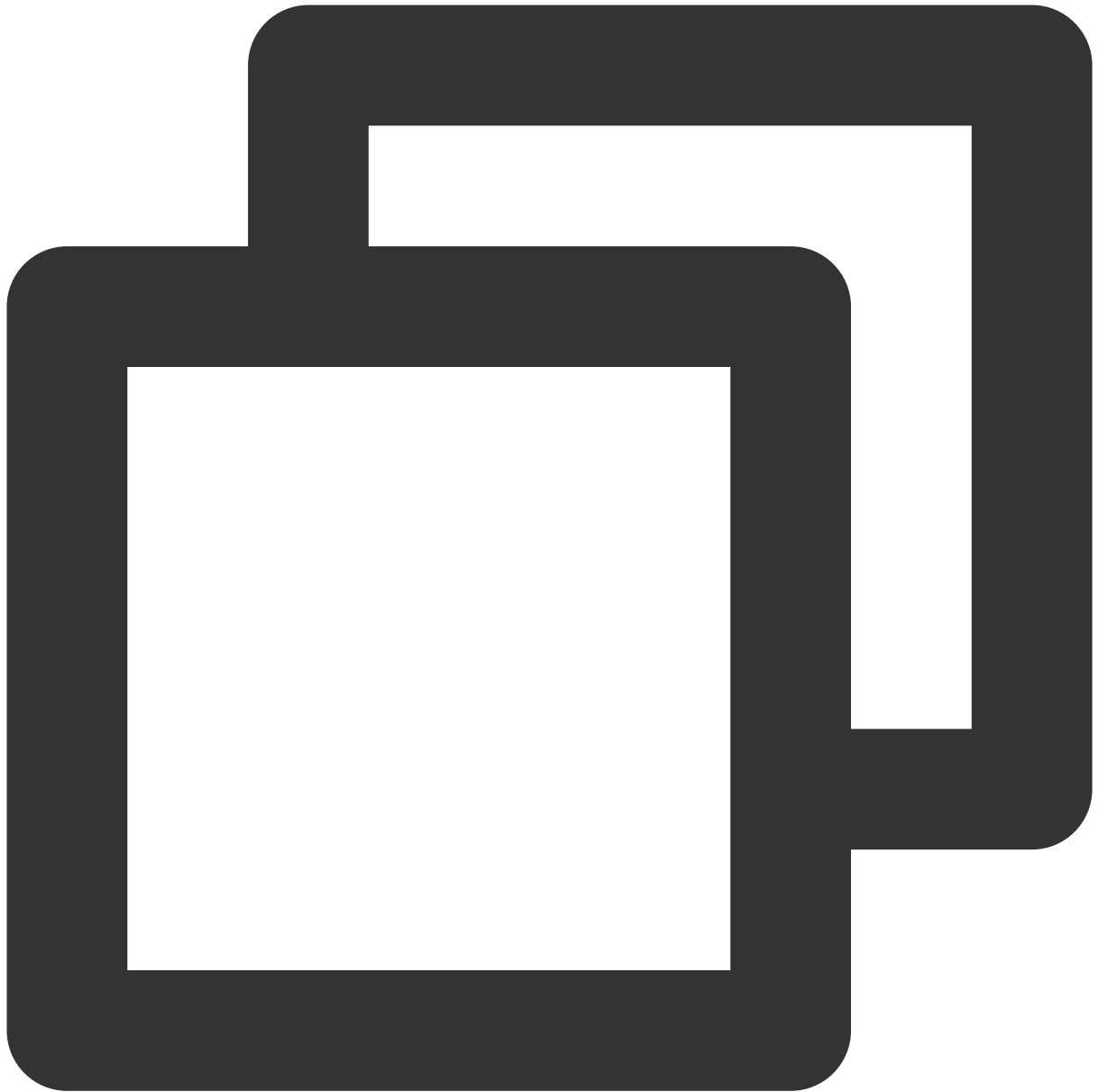
操作步骤

您可以通过以下两种方式，配置 HTTPS 代理。

对应实际使用的操作系统，执行以下命令，在环境变量中配置 HTTPS 代理。

Linux\\Unix 和 MacOS

Windows



```
export https_proxy=https://192.168.1.1:1111  
export https_proxy=https://myproxy.com:1111
```



```
setx http_proxy=https://192.168.1.1:1111
set http_proxy=https://myproxy.com:1111
# setx表示设置永久环境变量，设置后重启终端生效
```

执行以下命令，在命令行中使用 `--https-proxy` 选项设置 HTTPS 代理。



例如

```
tccli cvm DescribeRegions --https-proxy https://192.168.1.1:1111
```

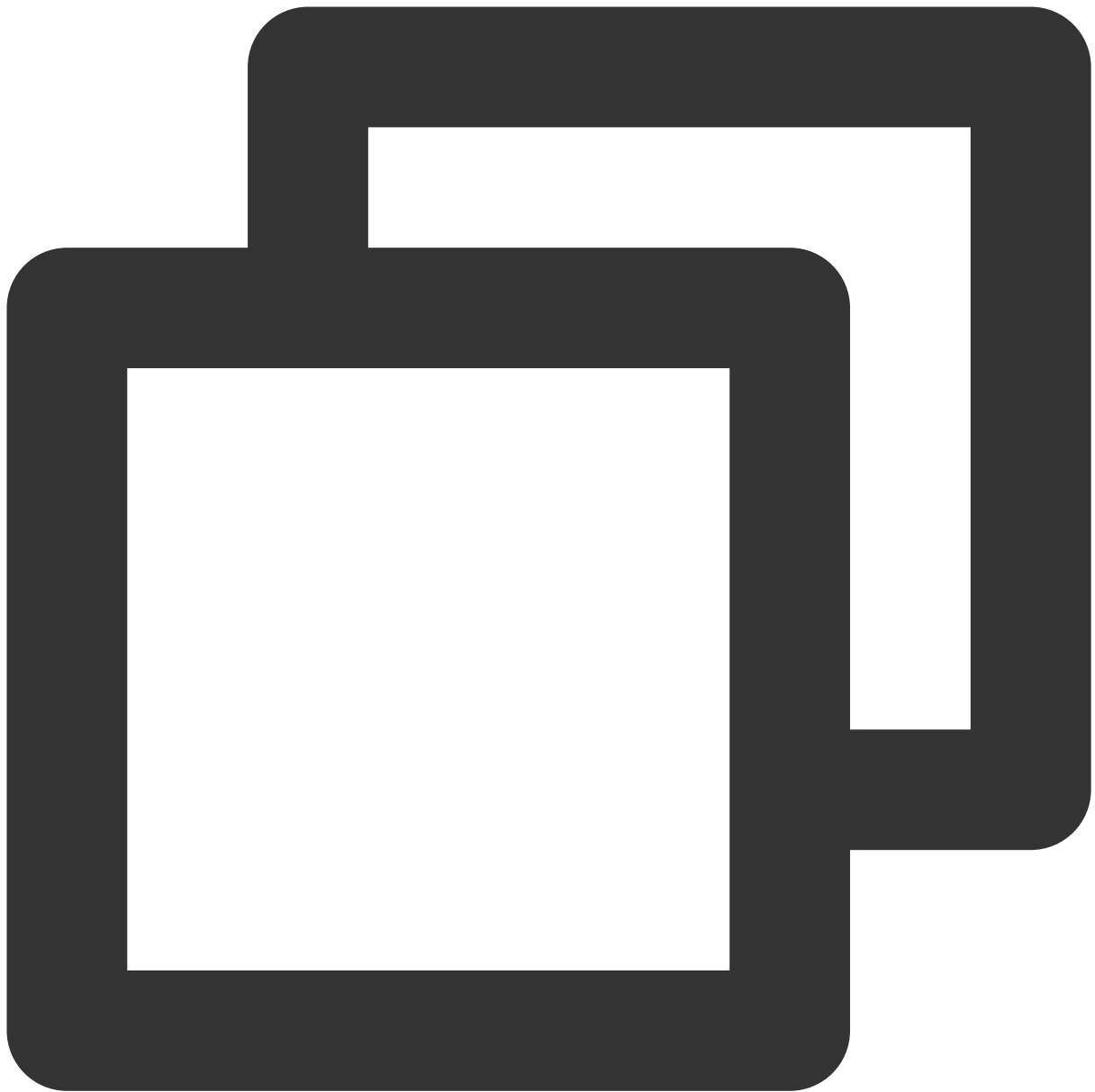
使用命令行自动补全功能

最近更新时间：2023-02-21 10:41:36

本文介绍如何使用 TCCLI 命令行自动补全功能。

操作步骤

若您使用 Linux 环境，则可执行以下命令，启动命令自动补全功能。



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

说明：

您也可以将该命令加入环境变量（ `/etc/profile` ）中，使自动补全功能一直有效。

使用 TCCLI

TCCLI 使用方法

最近更新时间：2023-02-21 10:41:36

操作场景

本文介绍如何使用命令行工具 TCCLI 的基础功能。

TCCLI 集成了腾讯云所有支持云 API 的产品，您可以在命令行下完成对腾讯云产品的配置和管理。包括使用 TCCLI 创建云服务器、操作云服务器、通过 TCCLI 创建云硬盘、查看云硬盘使用情况、通过 TCCLI 创建私有网络、往私有网络中添加资源等，所有在控制台页面能完成的操作，均能在 TCCLI 上执行命令实现。例如：

通过 `tccli cvm DescribeInstances` 命令查看当前账号有哪些云服务器。

通过 `tccli cbs DescribeDisks` 命令查看有云硬盘列表。

使用基础功能

说明：

以下以 Linux 操作系统为例，示例中非简单类型的参数，必须为标准 JSON 格式。

使用 TCCLI 创建 CVM

执行以下命令，创建一台 CVM。

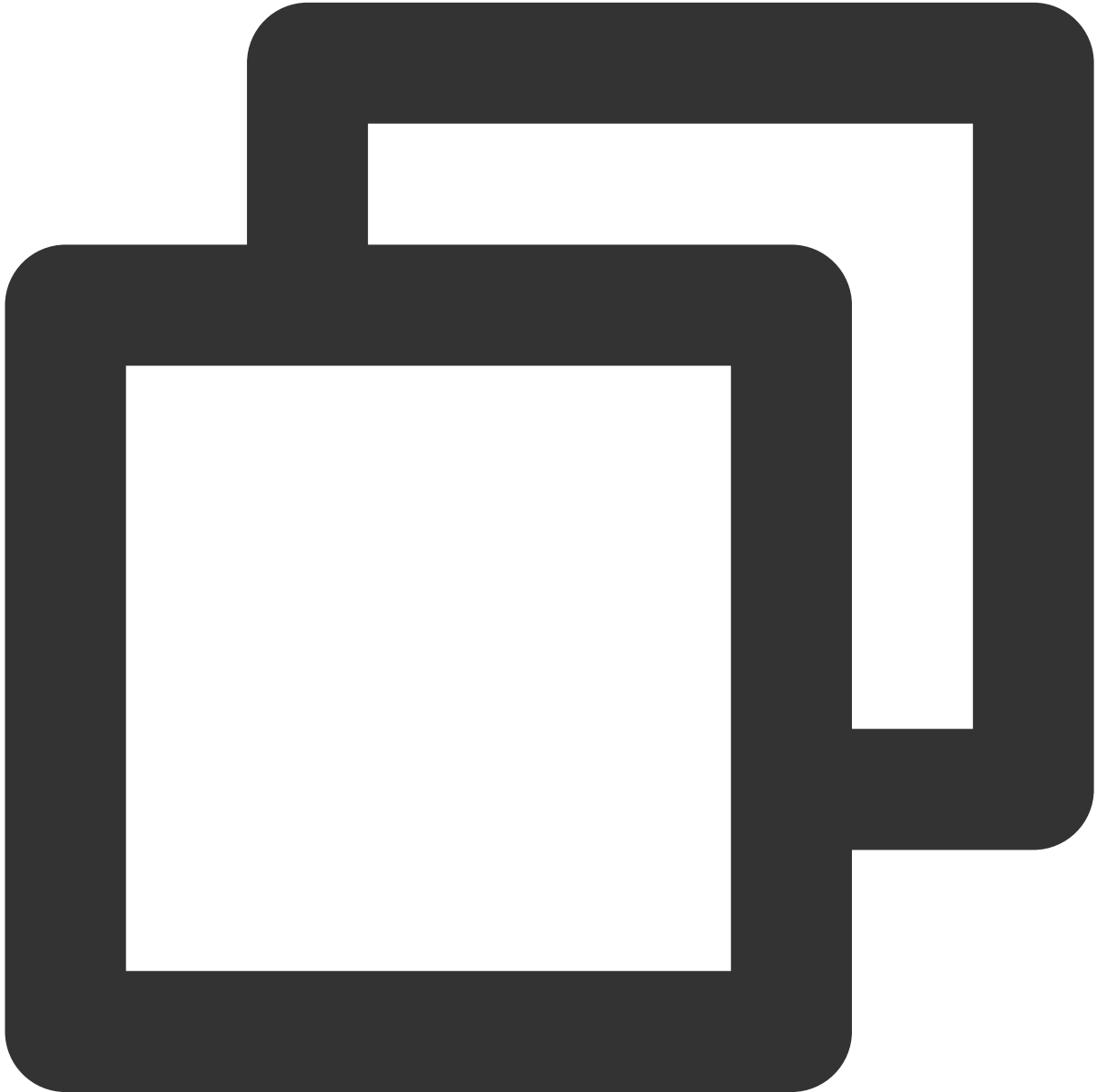


```
tccli cvm RunInstances
--InstanceChargeType POSTPAID_BY_HOUR
--InstanceChargePrepaid '{"Period":1,"RenewFlag":"DISABLE_NOTIFY_AND_MANUAL_RENEW"}'
--Placement '{"Zone":"ap-guangzhou-2"}'
--InstanceType S1.SMALL1
--ImageId img-8toqc6s3
--SystemDisk '{"DiskType":"CLOUD_BASIC", "DiskSize":50}'
--InternetAccessible '{"InternetChargeType":"TRAFFIC_POSTPAID_BY_HOUR","InternetMax
--InstanceCount 1
--InstanceName TCCLI-TEST
--LoginSettings '{"Password":"TCCLI"}'
```

```
--SecurityGroupIds '["sg-0rszg2vb"]'  
--HostName TCCLI-HOST-NAME1
```

使用标准输入传输二进制文件

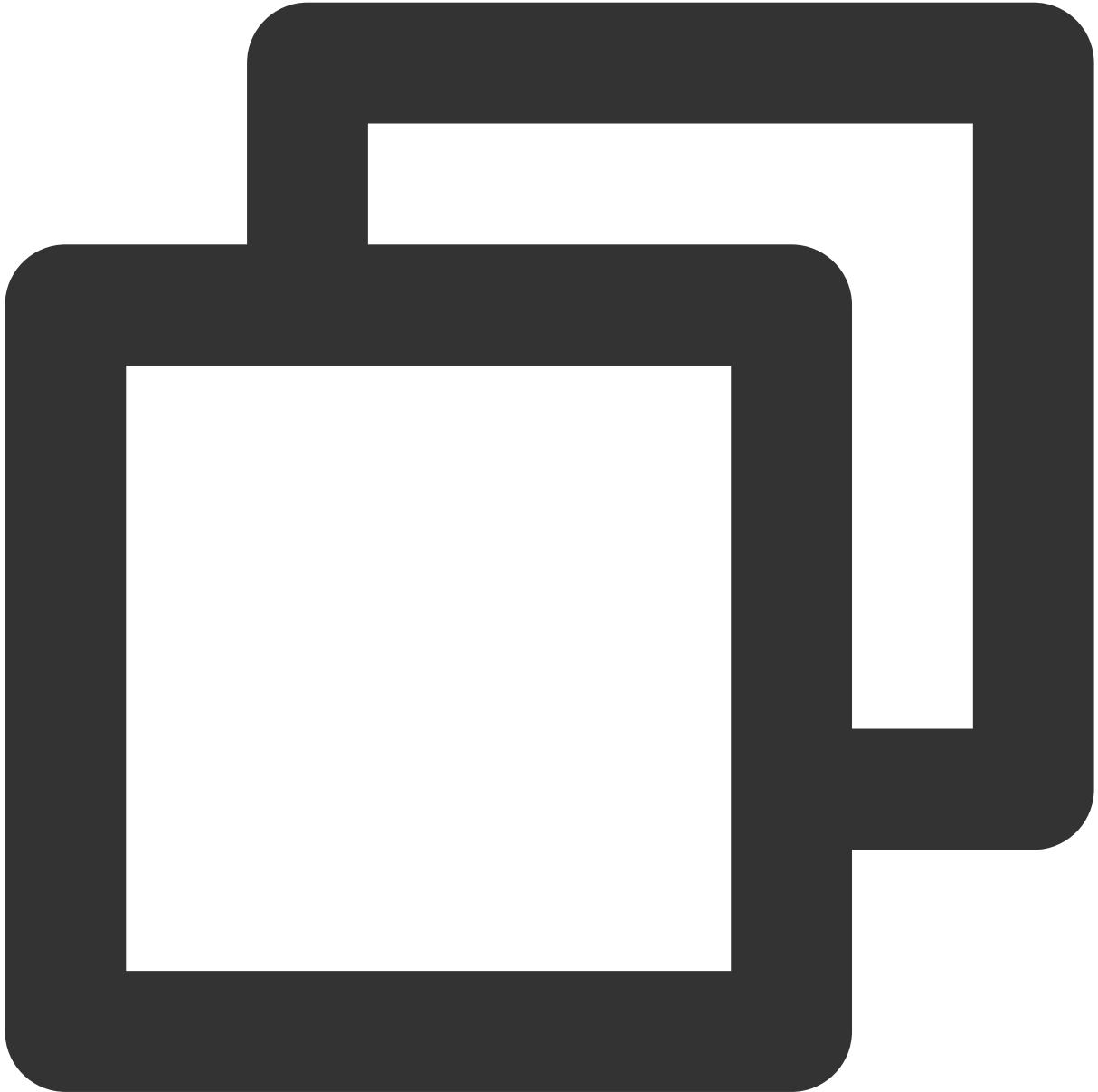
TCCLI 支持调用 `octet-stream` 类型的接口，若调用接口使用 `octet-stream` 协议，您可以使用标准输入 `<` `/path/to/file` 来传输二进制文件。示例命令如下：



```
# 以 cls 的 UploadLog 接口为例，上传日志可使用如下命令  
tccli cls UploadLog --TopicId xxx < /path/to/file
```

--cli-unfold-argument 参数

若调用接口参数为复杂类型时，可以增加 `--cli-unfold-argument` 参数，并进行参数补全，使用复杂类型点 (`.`) 展开的方式调用，降低输入难度。示例命令如下：



```
tccli cvm RunInstances --cli-unfold-argument \<\  
--Placement.Zone ap-guangzhou-3 \<\  
--ImageId img-8toqc6s3 \<\  
--DryRun True
```

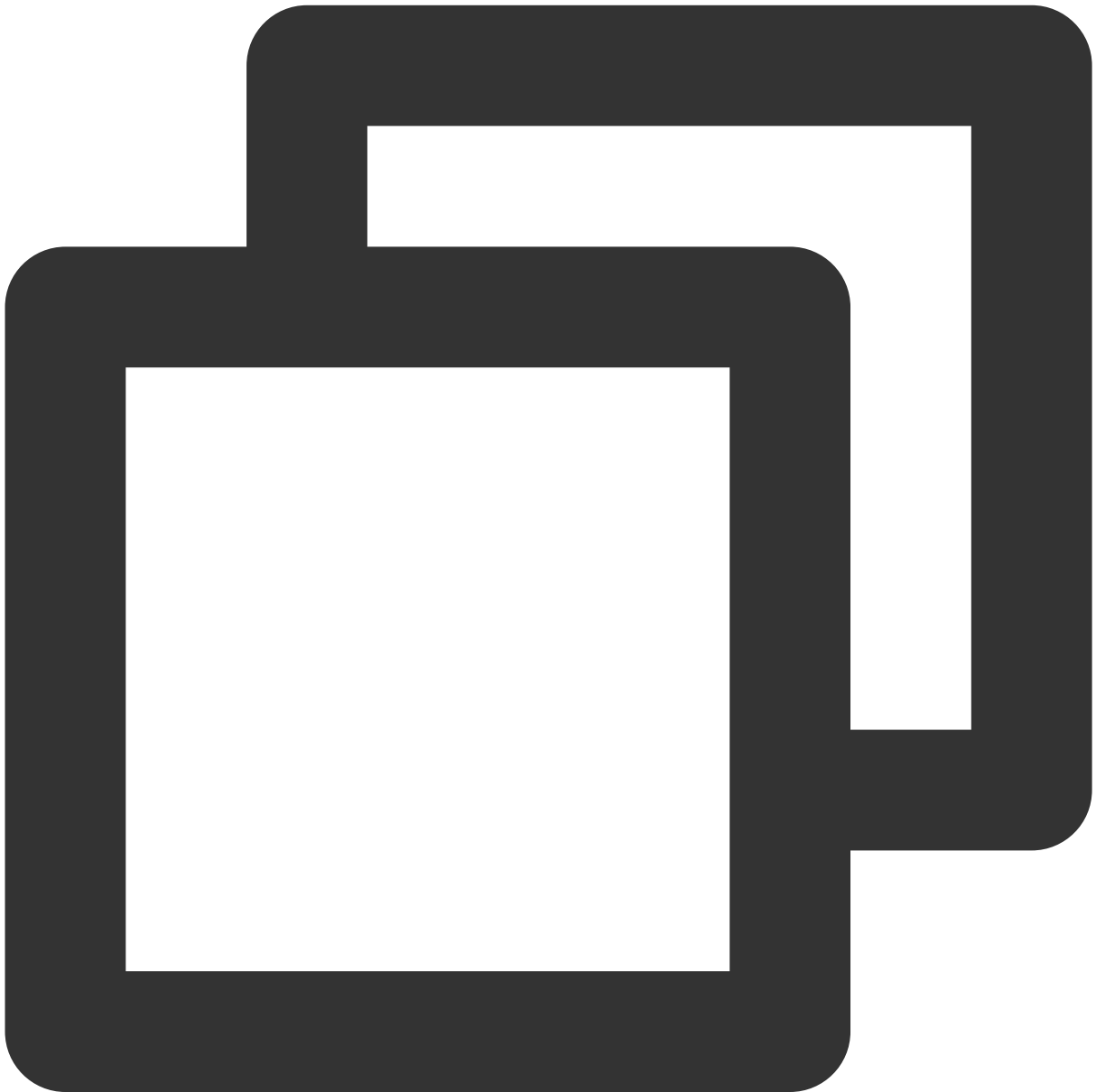
说明：

`--cli-unfold-argument` 命令可通过 **Tab** 键进行补全，详情请参见 [使用命令行自动补全功能](#)。

`--cli-unfold-argument` 命令需 `3.0.273.1` 版本及以上。

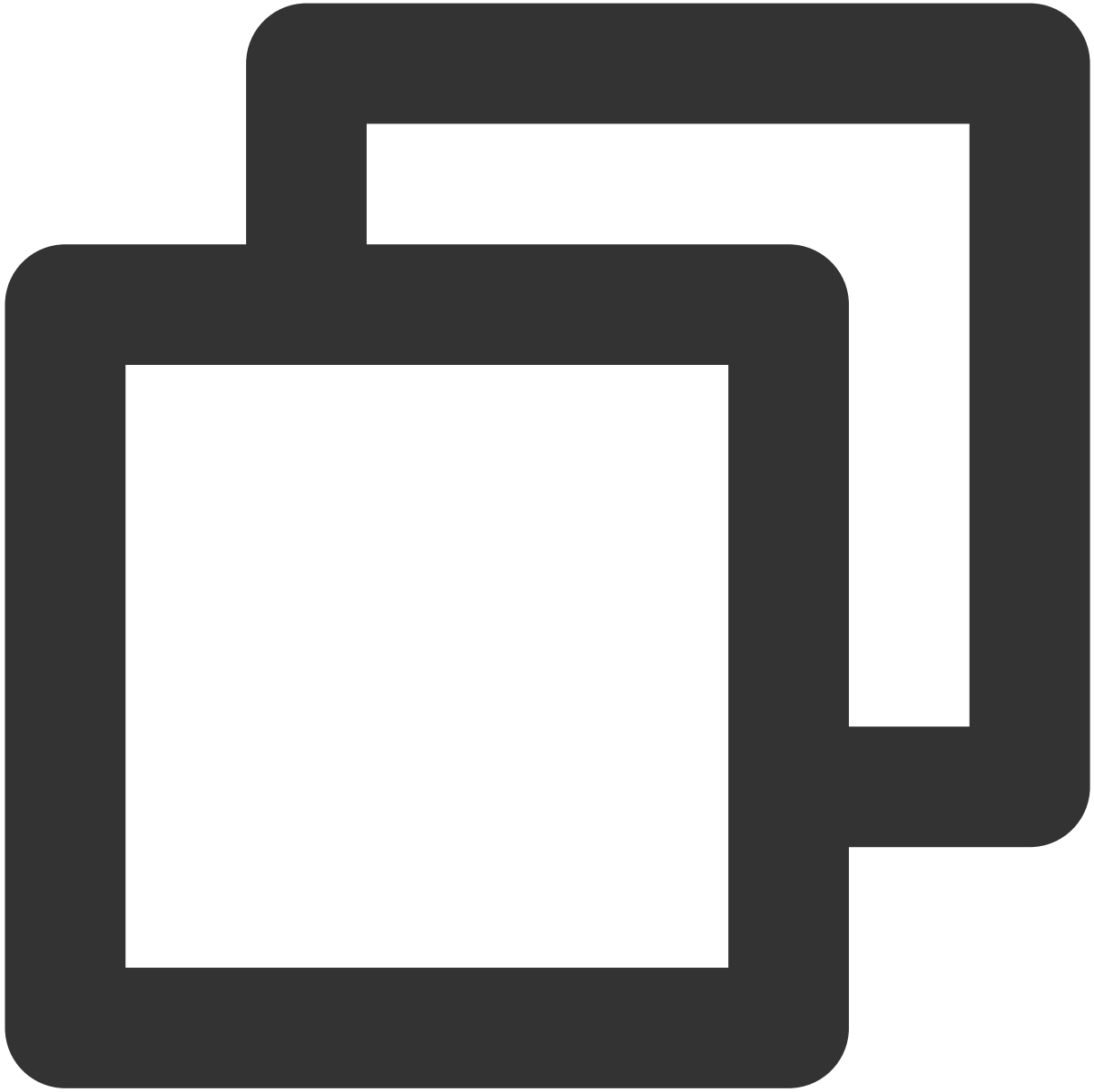
`--generate-cli-skeleton` 参数

您可增加 `--generate-cli-skeleton` 参数，输出 JSON 格式入参骨架。示例命令如下：



```
# 您还可将 json 格式入参骨架直接输入到 json 文件中
# $ tccli cvm DescribeInstances --generate-cli-skeleton > /home/test.json
tccli cvm DescribeInstances --generate-cli-skeleton
```

输出结果如下：



```
{
  "Limit": "Integer",
  "Filters": [
    {
      "Values": [
        "String"
      ],
      "Name": "String"
    }
  ]
}
```

```
  ],
  "InstanceIds": [
    "String"
  ],
  "Offset": "Integer"
}
```

说明：

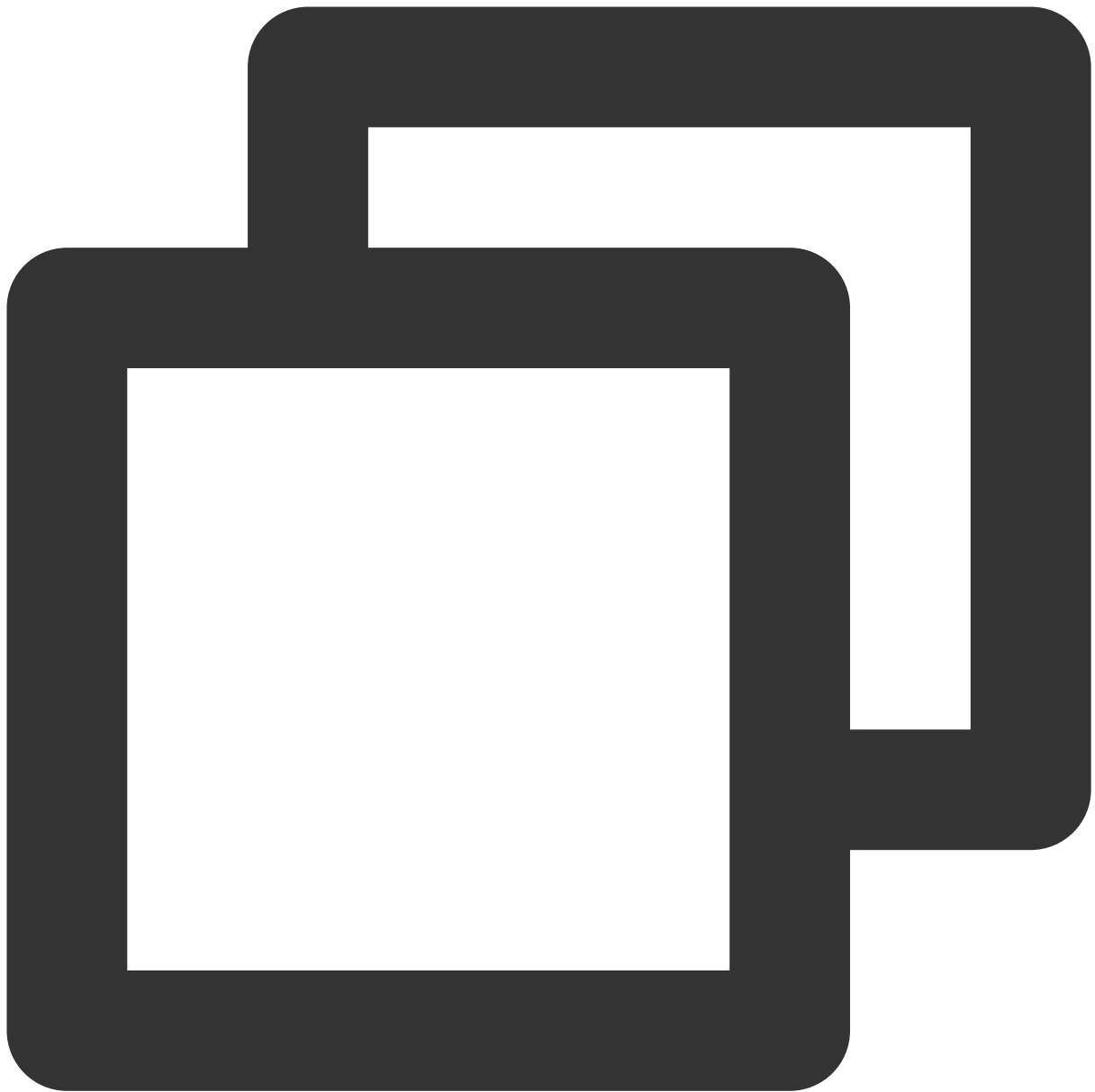
`--generate-cli-skeleton` 命令可通过 **Tab** 键进行补全，详情请参见 [使用命令行自动补全功能](#)。

`--generate-cli-skeleton` 命令需 **3.0.273.1** 版本及以上。

--cli-input-json 参数

若接口入参较多，可增加 `--cli-input-json` 参数，该参数支持 JSON 文件输入（参数后需增加

`file://+文件路径` ）。您可以使用 `--generate-cli-skeleton` 生成相应的 JSON 文件，填写参数后即可直接使用 **该 JSON 文件** 调用接口。示例命令如下：



```
tccli cvm DescribeInstances --cli-input-json file:///home/test.json
```

说明：

`--cli-input-json` 命令可通过 **Tab** 键进行补全，详情请参见 [使用命令行自动补全功能](#)。

`--cli-input-json` 命令需 **3.0.250.2** 版本及以上。

获取帮助信息

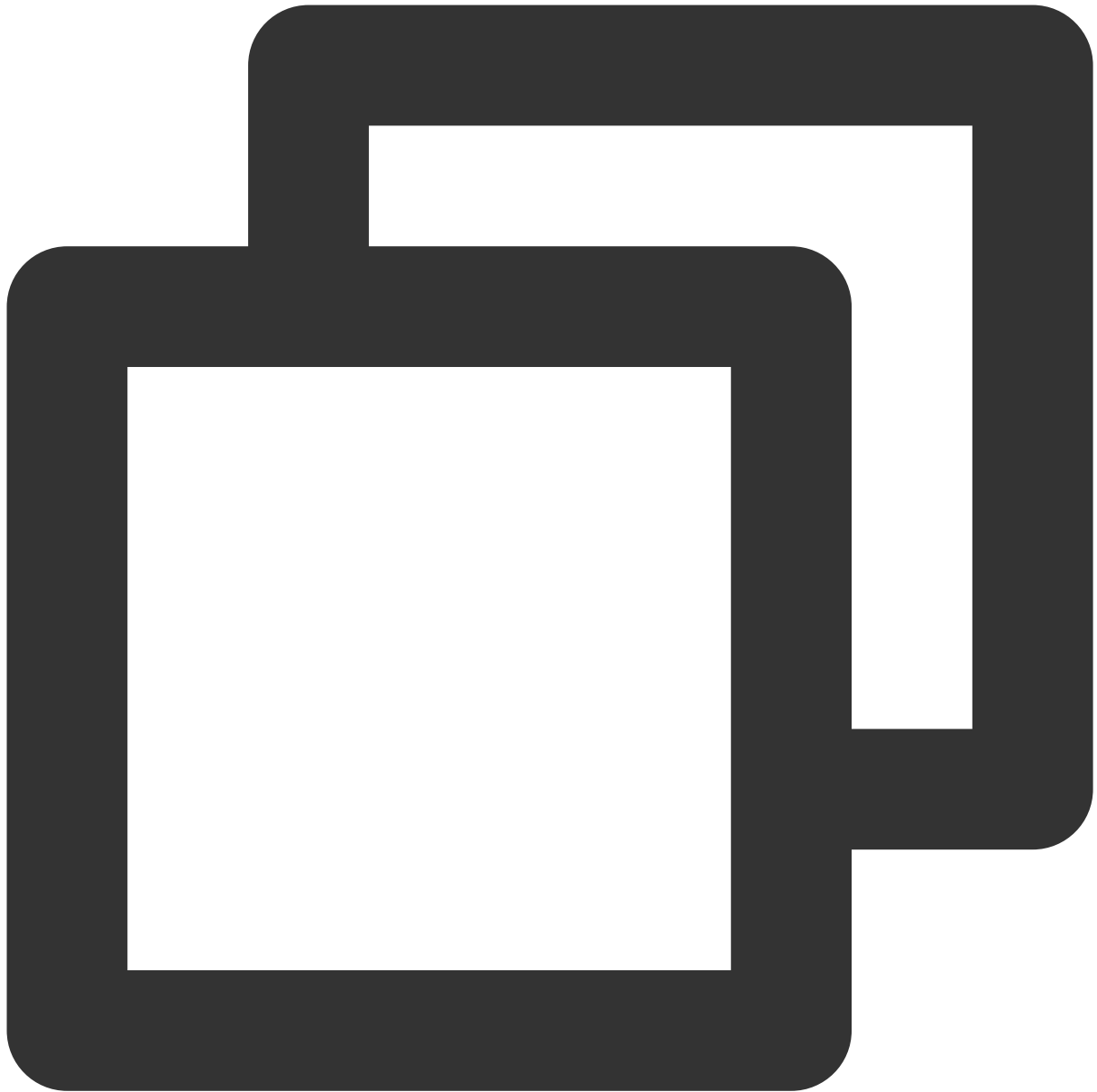
最近更新时间：2023-02-21 10:41:36

本文介绍如何使用 TCCLI 查看产品接口的帮助信息。

操作步骤

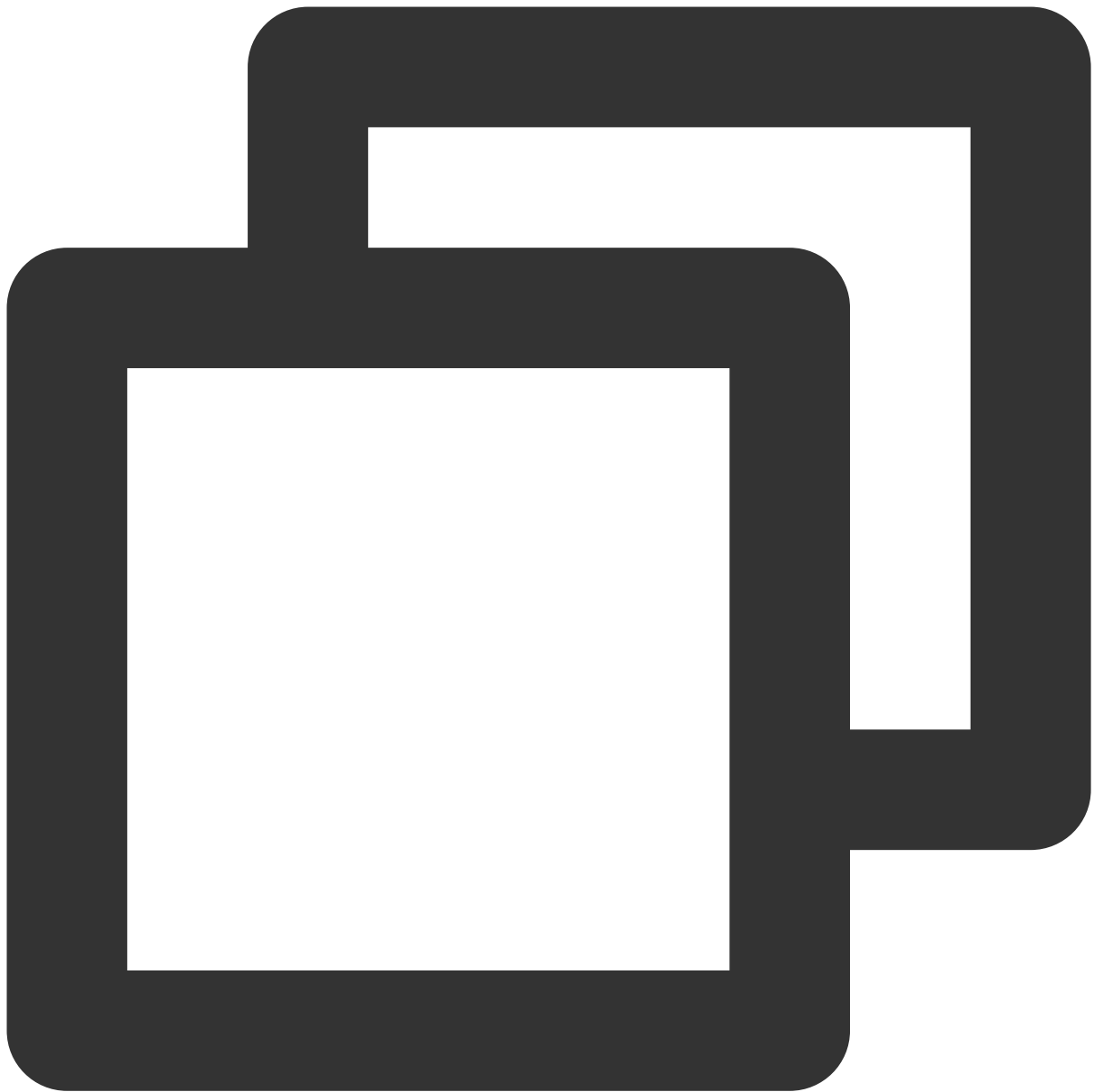
查看帮助信息

执行以下命令，查看支持的产品。您也可以在 [API中心](#) 的文档中查看。



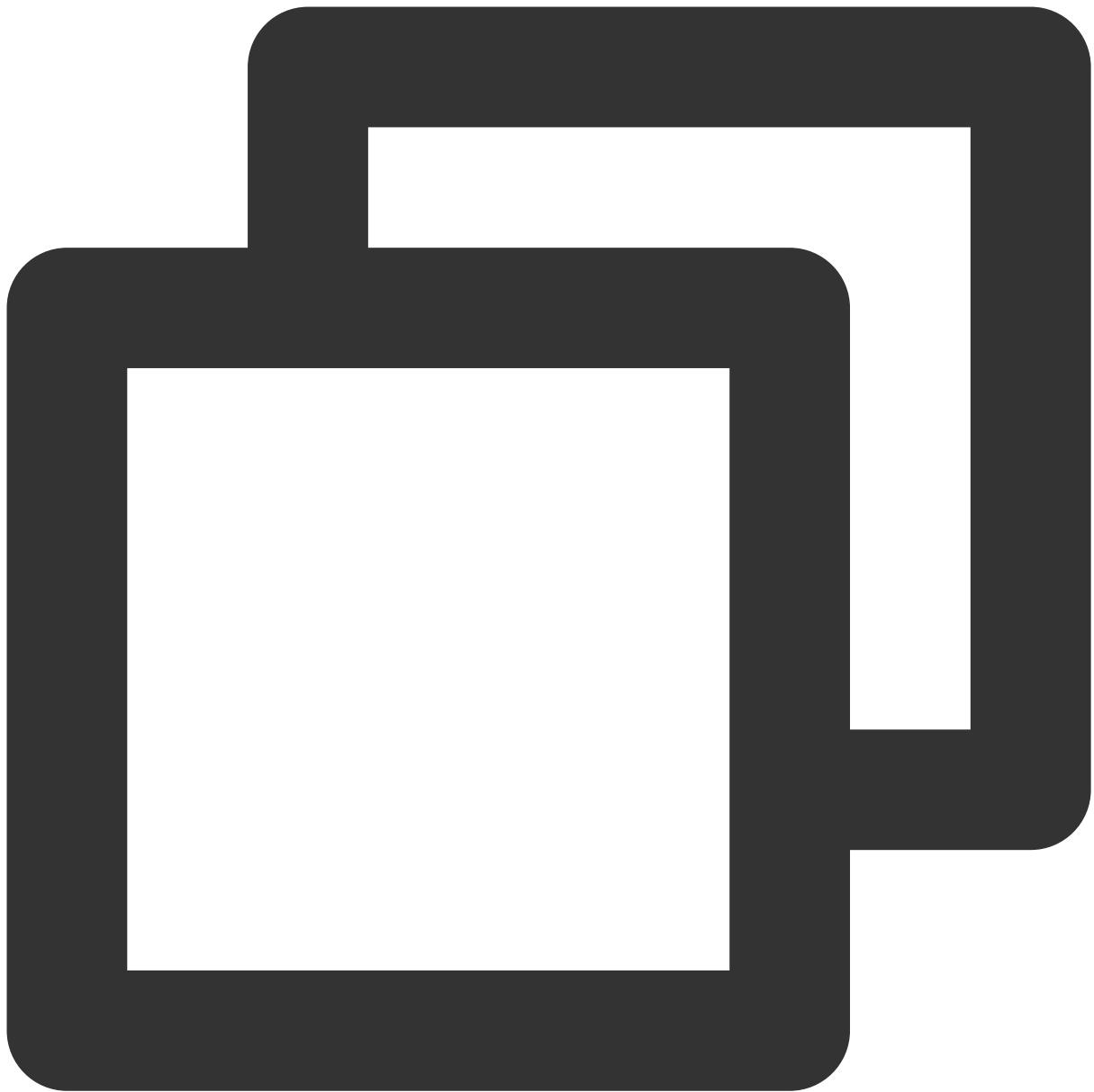
```
tccli help
```

以云服务器 CVM 为例，执行以下命令，查看 CVM 支持的接口。



```
tccli cvm help
```

以云硬盘 CBS 中的 DescribeDisks 接口为例，执行以下命令，查看接口支持的参数。具体的参数说明和 API 的相关信息可以在腾讯云官网查看对应的 API 文档。



```
tccli cbs DescribeDisks help
```

查看详细帮助信息

TCCLI 默认显示简化版帮助信息，如需查看详细信息，可以使用 `--detail` 选项。
执行以下命令，查看支持的产品的详细信息。



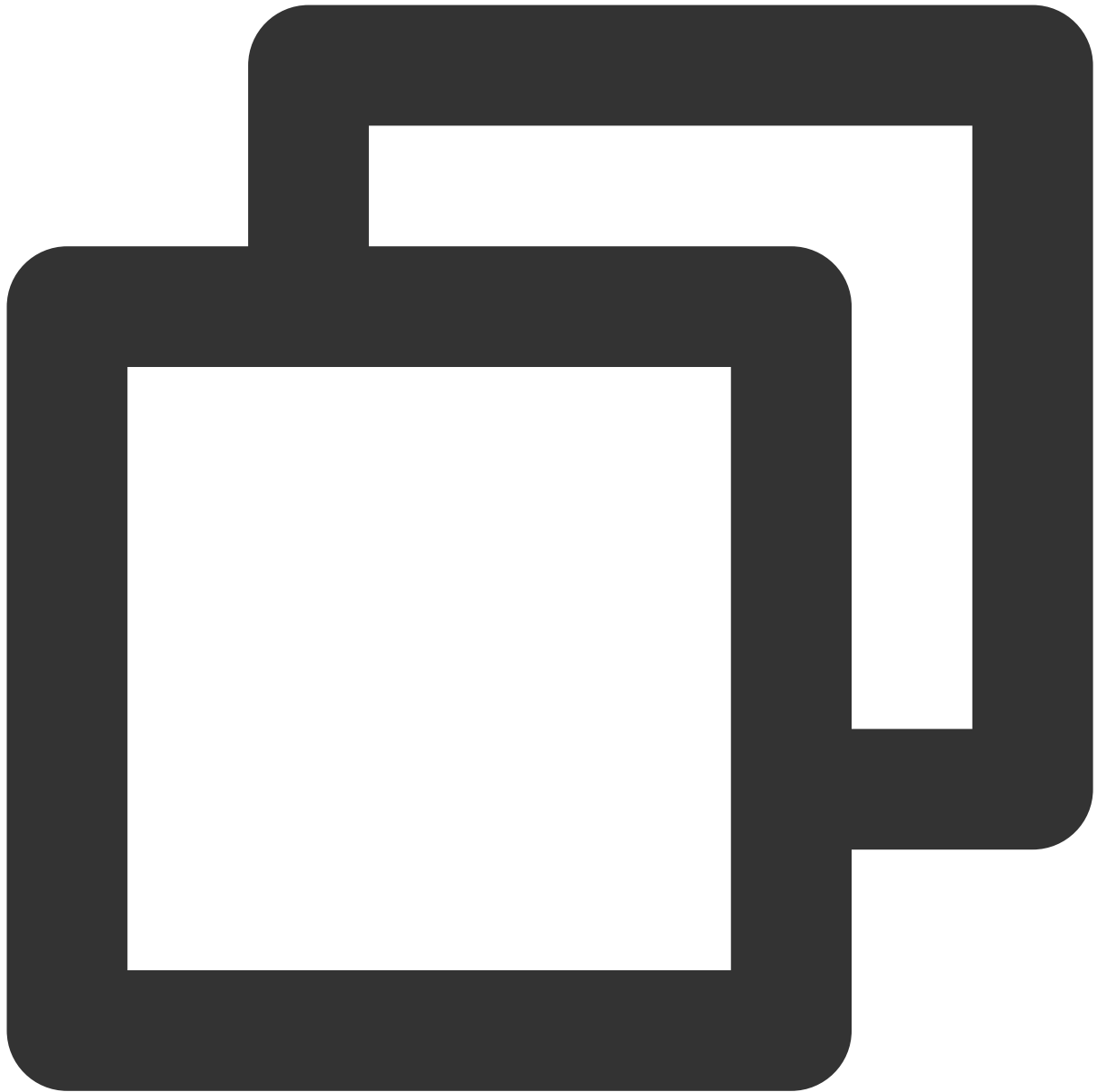
```
tccli help --detail
```

以云服务器 CVM 为例，执行以下命令，查看 CVM 支持的接口的详细信息。



```
tccli cvm help --detail
```

以 CBS 中的 DescribeDisks 接口为例，执行以下命令，查看接口的入参、出参的详细信息及使用示例。



```
tccli cbs DescribeDisks help --detail
```

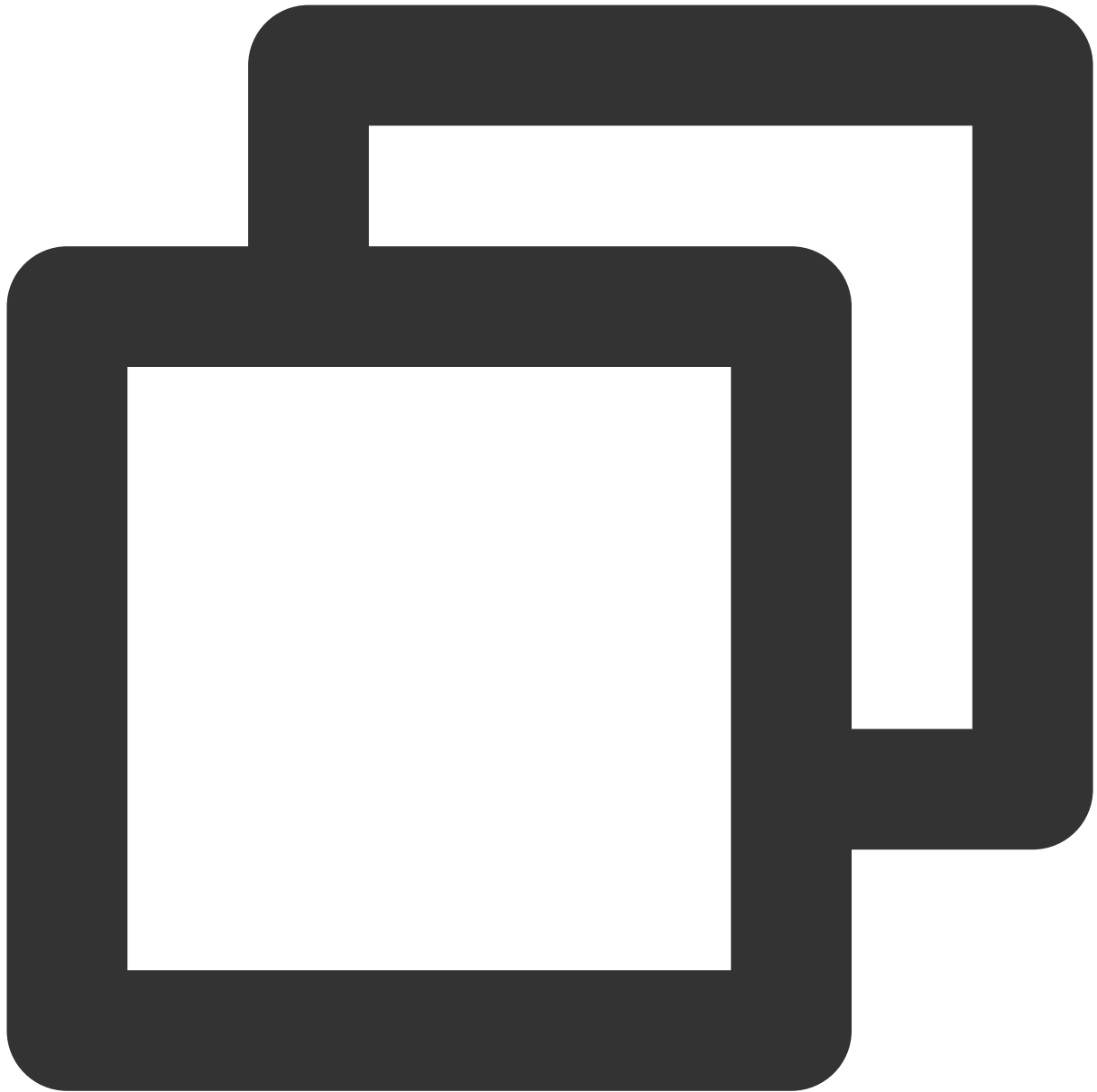
多版本接口访问

最近更新时间：2023-02-21 10:41:36

TCCLI 默认访问产品最新版本的接口，而某些产品可能存在多个版本的接口，若您需访问特定旧版本的接口，可参考本文进行实现。

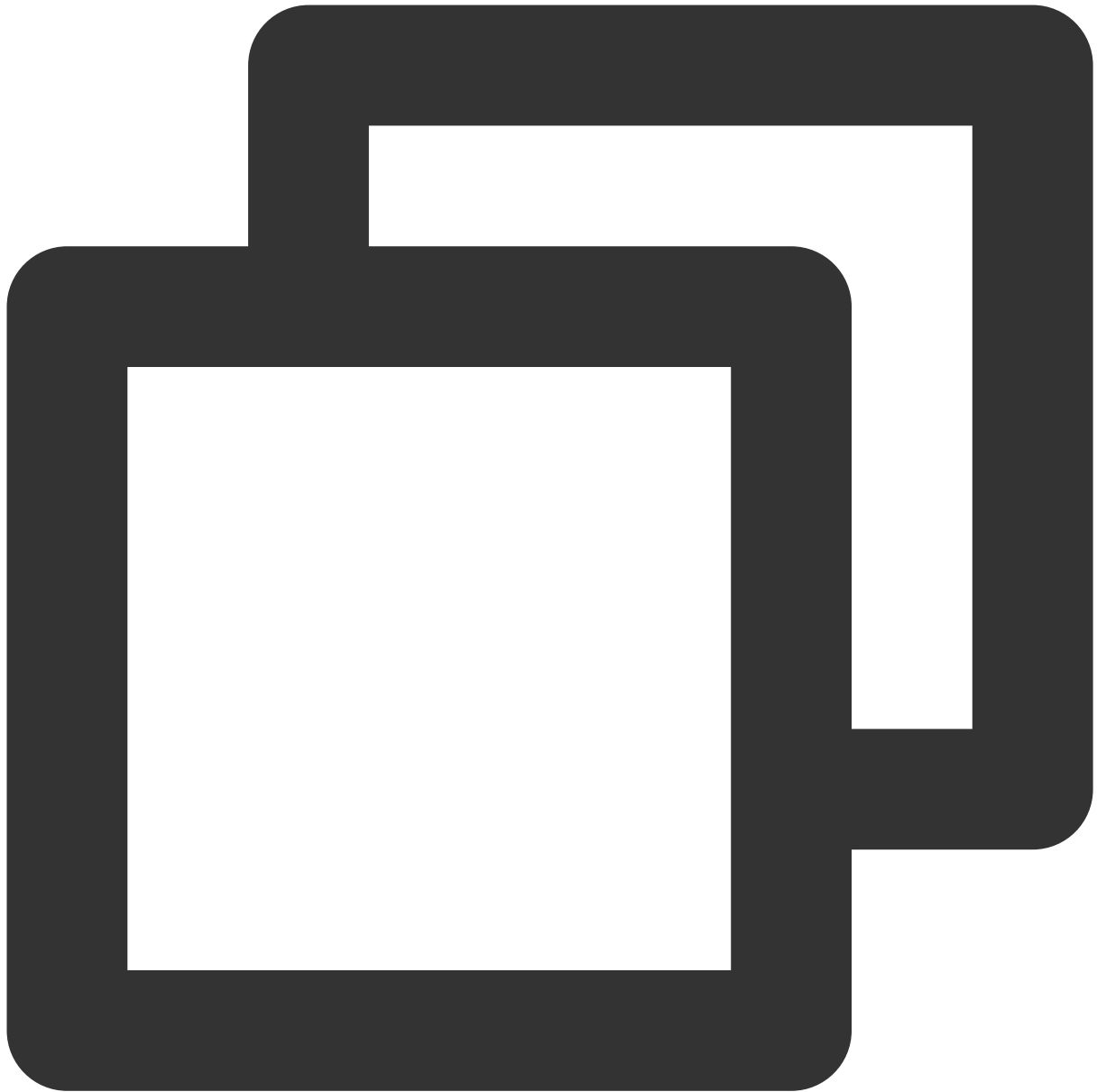
操作步骤

以云服务器 CVM 为例，执行以下命令，设置默认使用版本为 `2017-03-12`。



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

执行以下命令，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
```

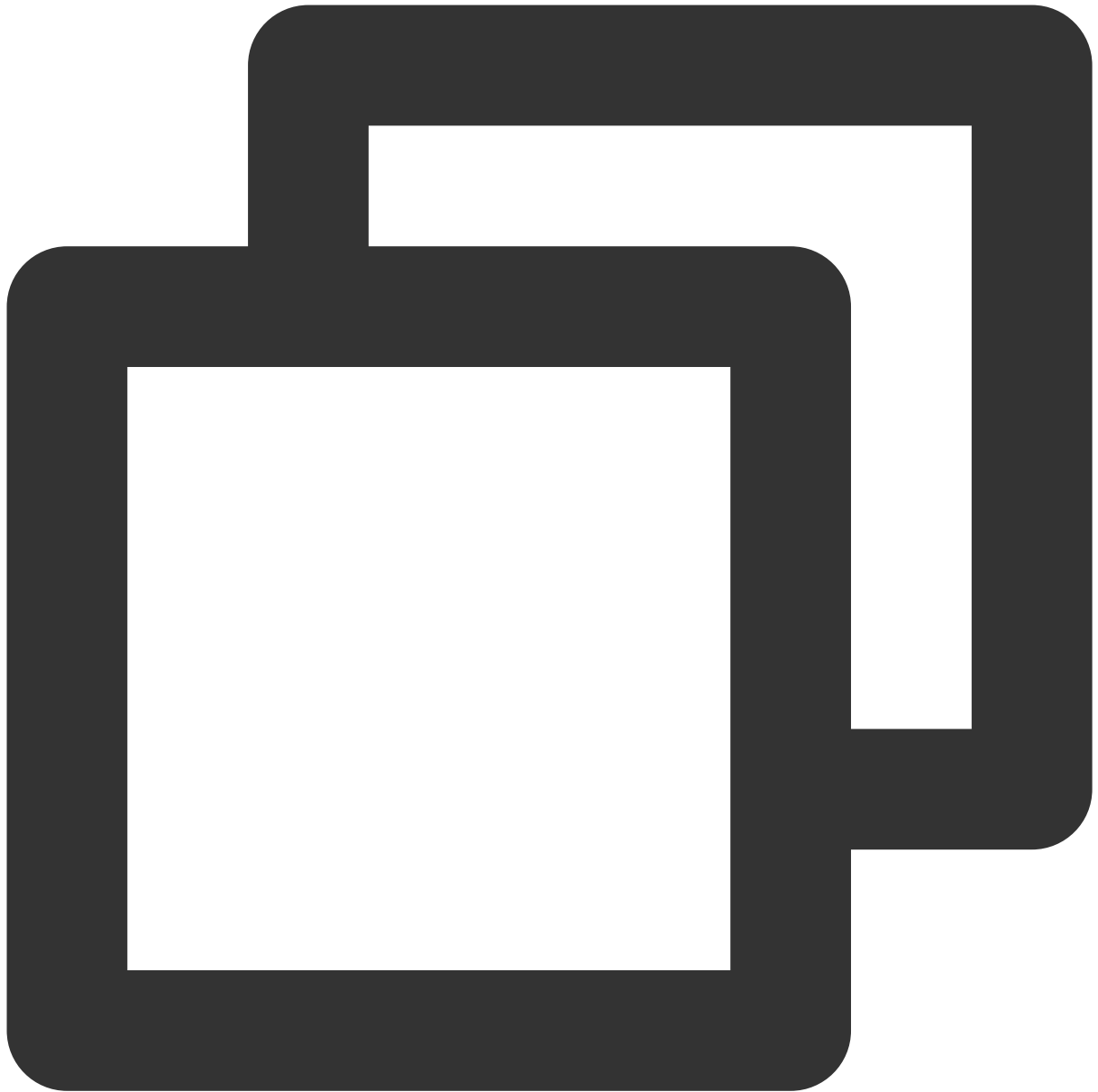
指定最近接入点

最近更新时间：2023-02-21 10:41:36

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 Endpoint。

操作步骤

以云服务器为例，执行以下命令，设置默认 Endpoint 为 ap-guangzhou。



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

执行以下命令，调用时实时指定 Endpoint 为 ap-guangzhou。



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

结果返回过滤

最近更新时间：2023-02-21 10:41:36

某些命令的返回结果较多，若您需查看某些特定内容，可以对返回结果进行过滤。

操作步骤

本文以云服务器 CVM DescribeZones 接口的返回为例，展示了不同过滤方式，及其返回结果：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": "4fd313a6-155f-4c7a-bf86-898c02fcae02"
}
```

只看某个字段。

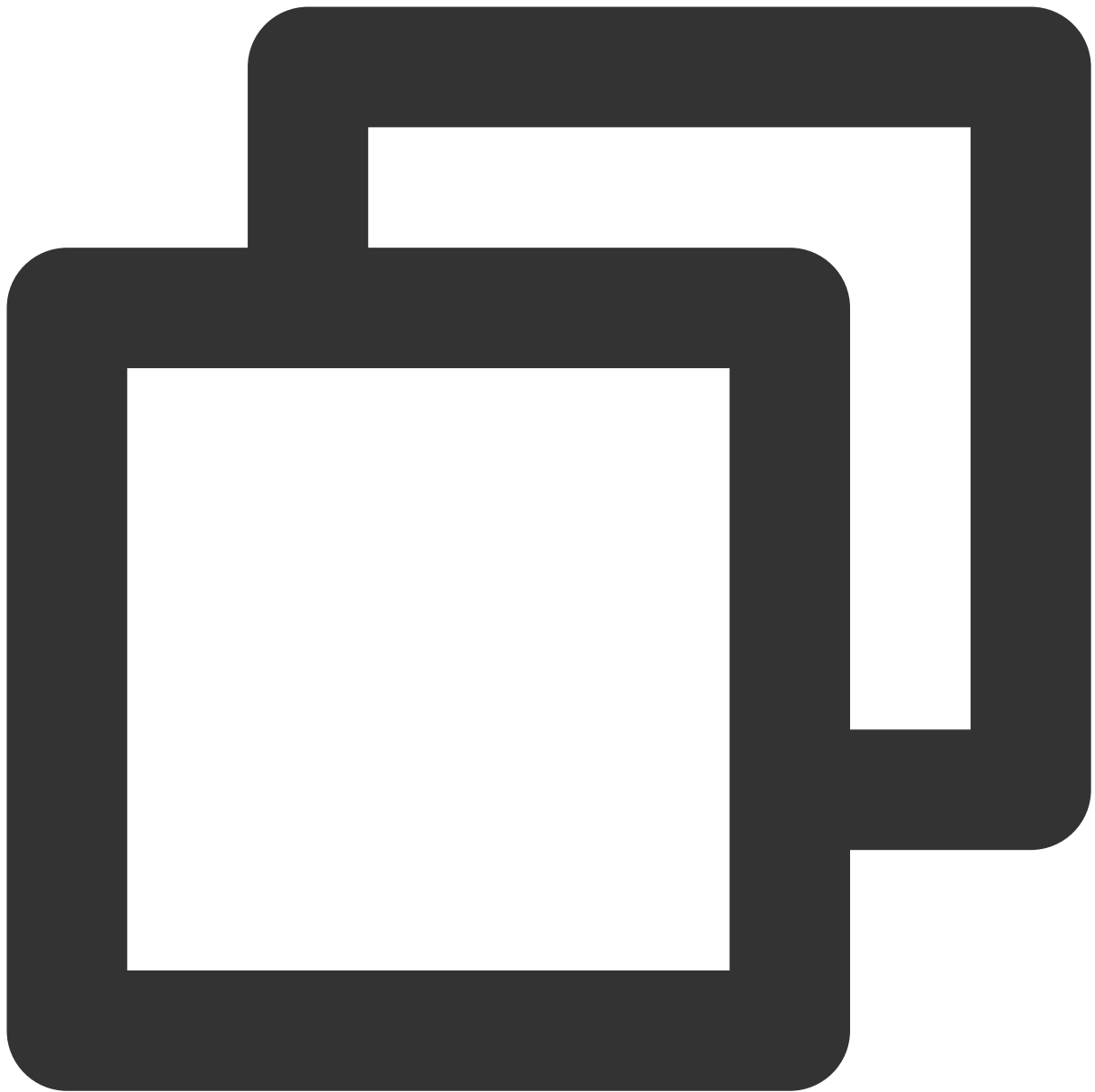


```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter TotalCount  
4
```

指定某个数组类型对象的第 N 个子对象的信息。

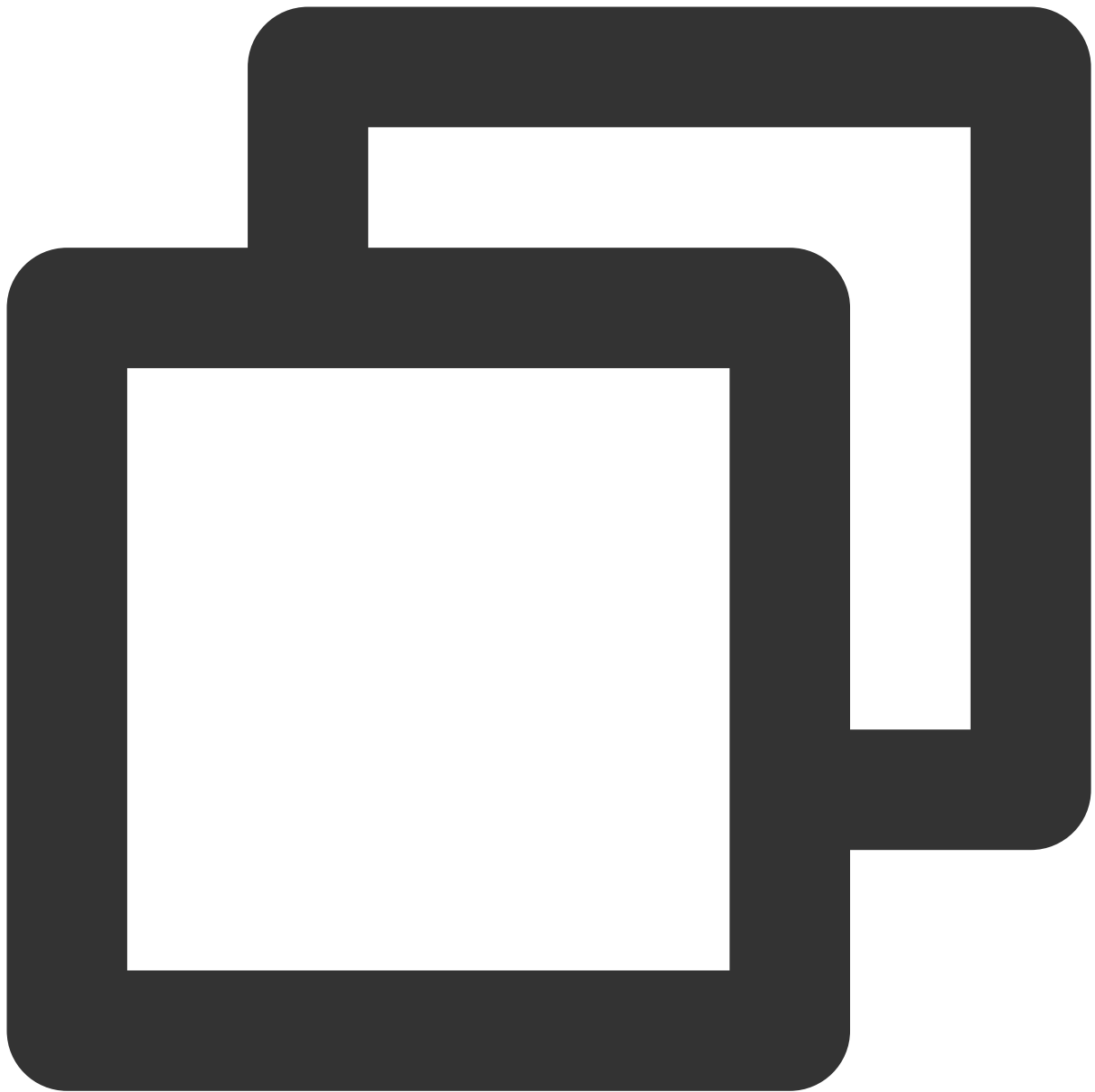
注意：

MacOS 系统使用如下命令可能会报错 `zsh: no matches found: xxx`，这时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[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。

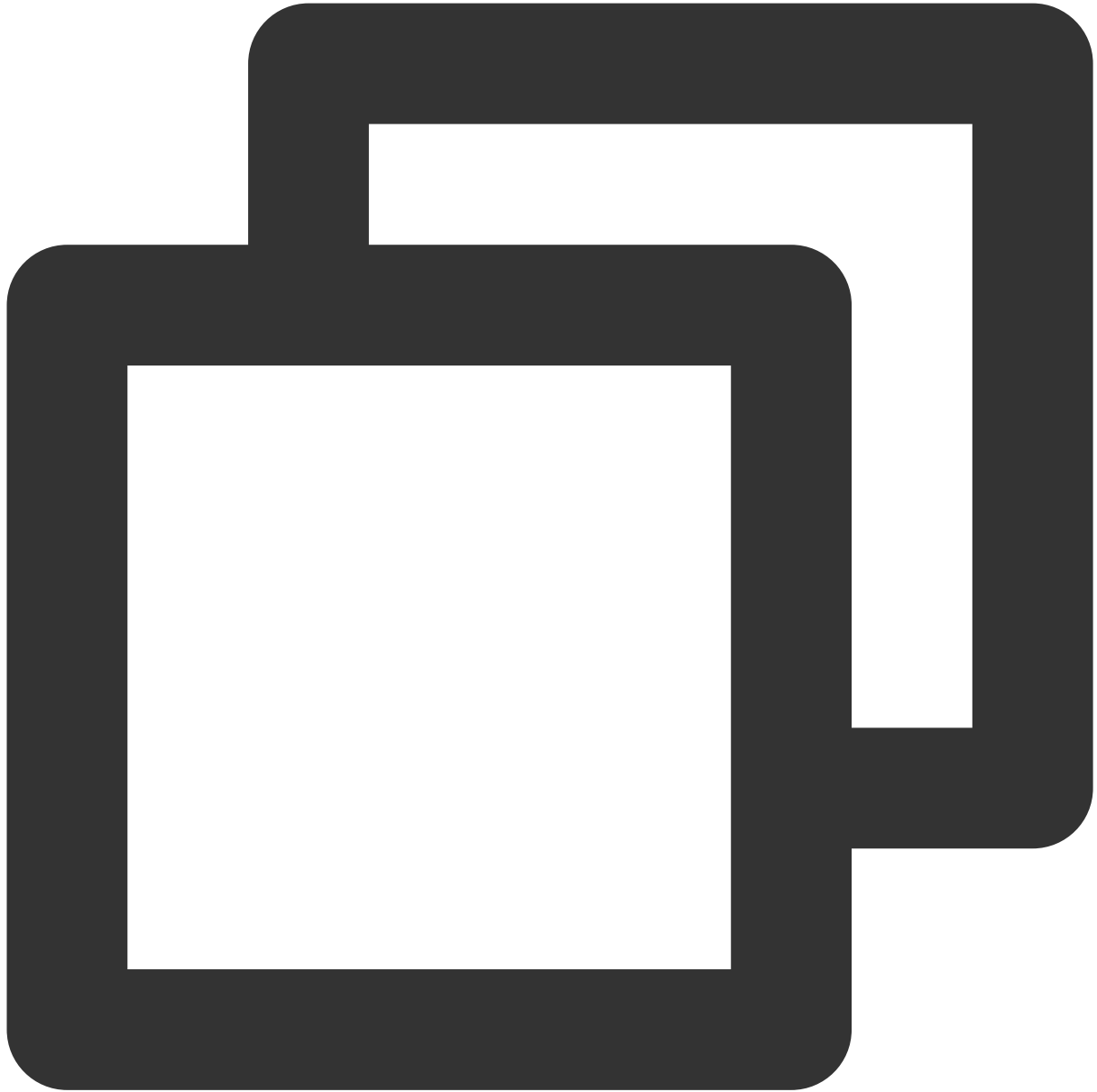


```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[*].ZoneName  
[  
  "广州一区",  
  "广州二区",  
  "广州三区",  
  "广州四区"  
]
```

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:Zon
[
  {
    "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"  
    }  
  ]
```

结果轮询

最近更新时间：2023-02-21 10:41:36

在使用产品的过程中，有些操作并不能即时完成，您可以使用结果轮询功能来不断查询操作是否完成。例如，在开启一台实例后，实例并不能立即进入 `RUNNING` 状态，则可使用结果轮询功能对实例状态轮询，直到出现 `RUNNING` 状态为止。

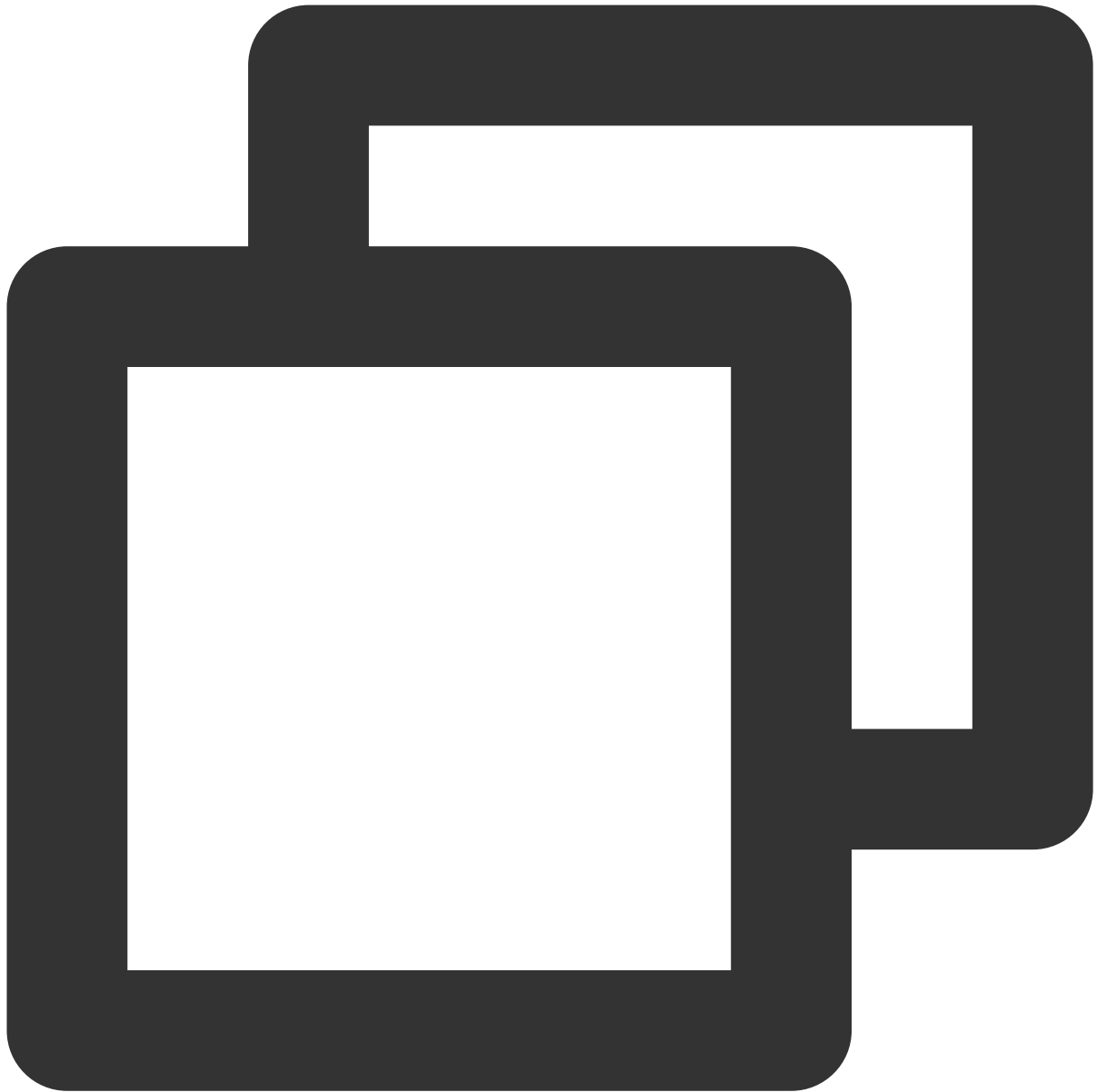
操作步骤

执行以下命令，程序将按照一定时间间隔对实例的状态进行轮询，直到实例的状态为 `RUNNING` 或者超时为止。



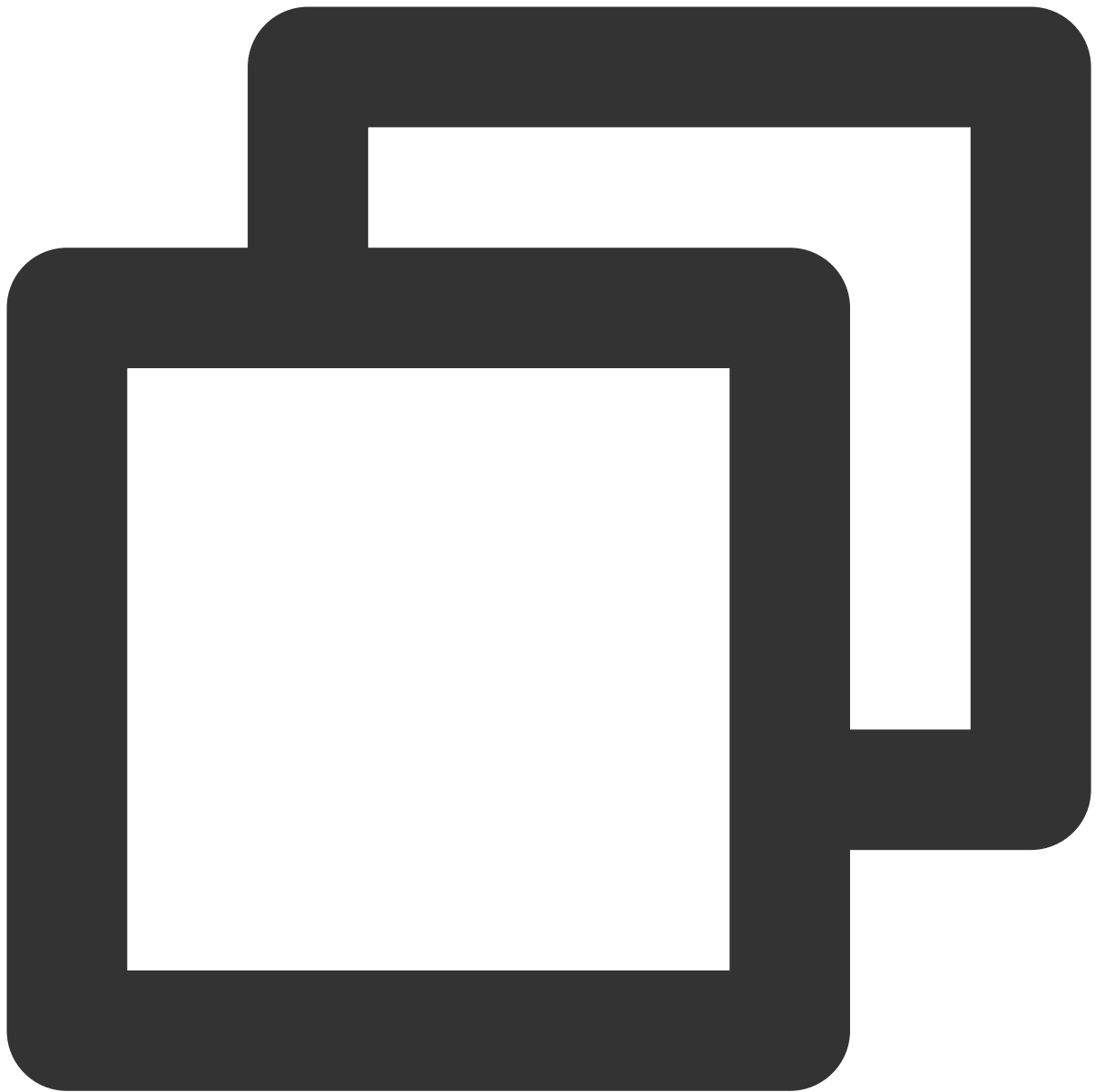
```
tccli cvm DescribeInstancesStatus --region ap-hongkong --waiter '{"expr':'Instances'
```

您可自定义超时时间和睡眠时间，执行以下命令，设定超时时间为180秒，睡眠时间为5秒。



```
tccli cvm DescribeInstancesStatus --region ap-hongkong --waiter '{"expr':'Instances'
```

您可在配置文件中设置可选子参数的值。在 `default.configure` 文件中添加如下参数，设置系统超时时间为 180s，睡眠时间为 5s。



```
"waiter": {  
  "interval": 5,  
  "timeout": 180  
},
```

参数说明

--region：需替换为您实例所在的地域。

--waiter：后的参数需使用双引号包裹，且参数需为 JSON 格式。其中必选及可选参数如下表：

参数	是否必选	说明
expr	是	指定被查询的字段，请使用 jmespath 查找被指定的字段的值。
to	是	被轮询的字段的目标值。
timeout	否	轮询的超时时间，单位：秒。
interval	否	进程睡眠的时间，单位：秒。

使用高级功能

最近更新时间：2019-12-23 15:47:25

操作场景

本文详细介绍如何使用 TCCLI 国际版高级功能，包括多版本接口访问、指定最近的接入点和返回结果过滤。

操作步骤

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（以 CVM 举例）

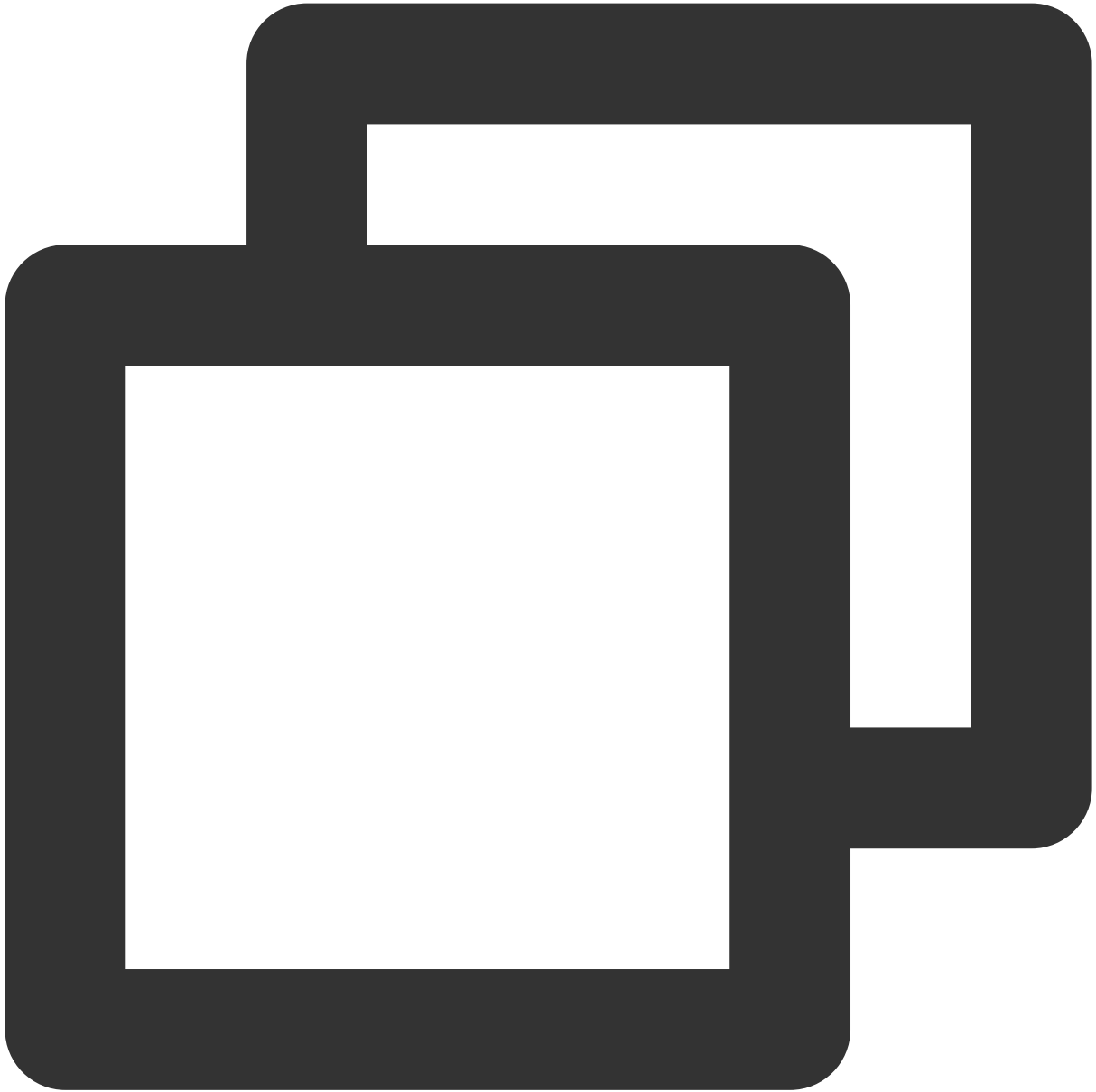


```
# 设置 cvm 产品默认使用版本:2017-03-12
tccli configure set cvm.version 2017-03-12

# 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（以 CVM 为例）



```
# 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

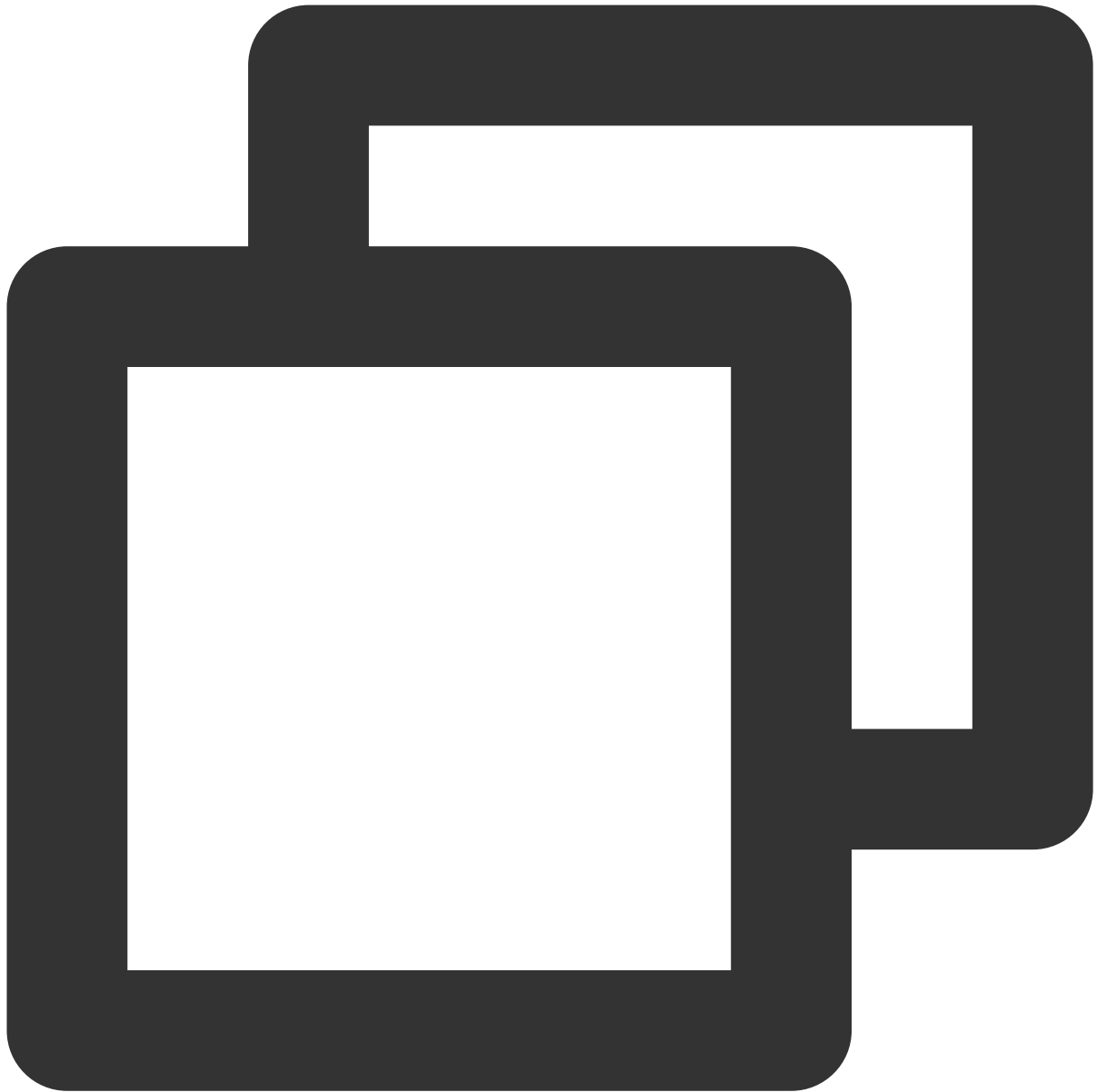
1. 不加任何过滤时的输出（以 CVM DescribeZones 接口的返回为例）：



```
[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": "4fd313a6-155f-4c7a-bf86-898c02fcae02"
}
```

2. 只看某个字段：



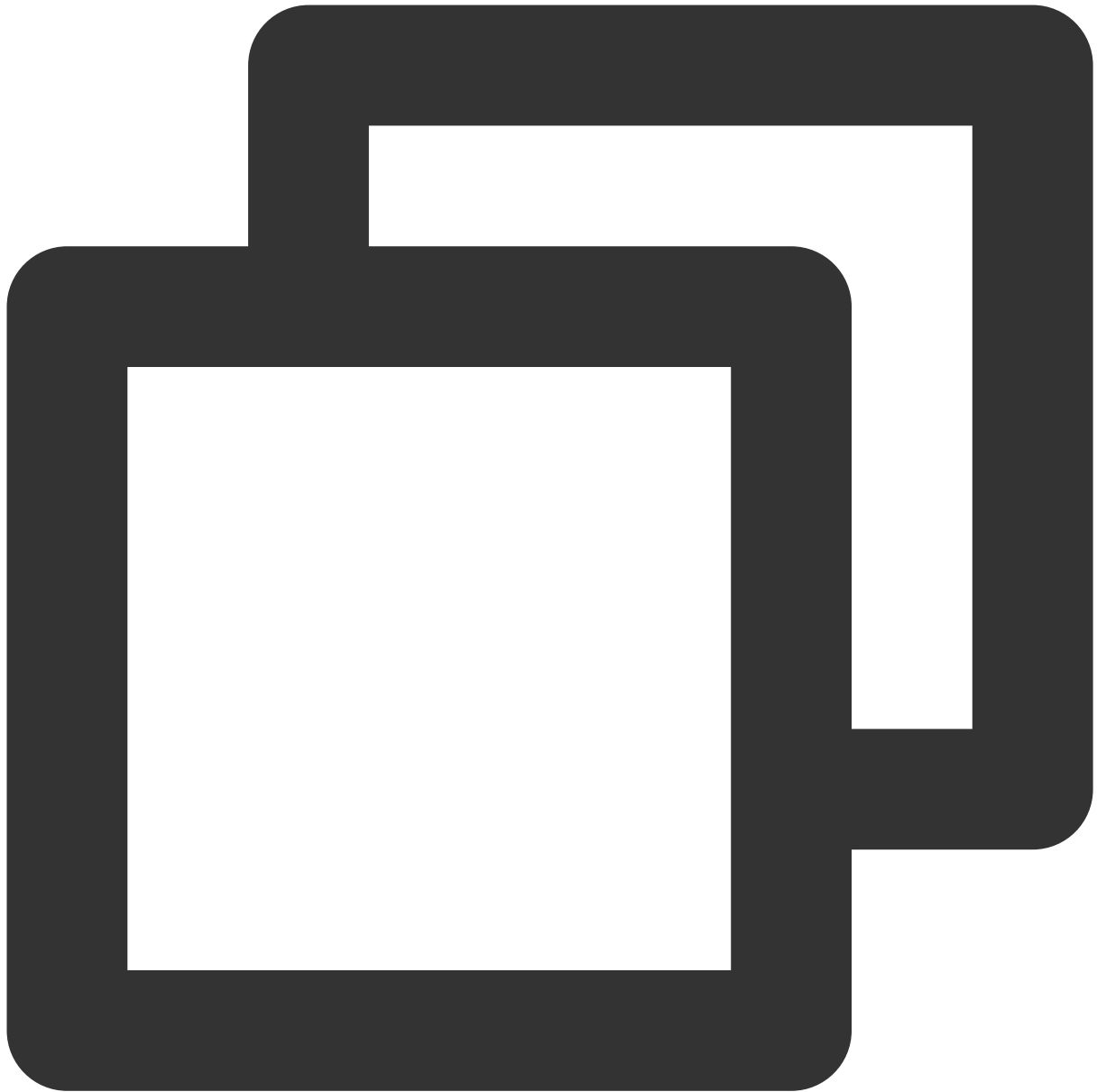
```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter TotalCount  
4
```

3. 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"
}
```

4. 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"]
```

5. 过滤数组里的子对象，同时还以新的名称展示：

注意：

这里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"  
  }  
]
```