

Tencent Cloud Command Line Interface Operation Guide Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

- Installing TCCLI

- Configuring TCCLI

 - TCCLI Configuration Method

 - Using an HTTPS Proxy

 - Using the Autocomplete Feature

- Using TCCLI

 - TCCLI Usage

 - Obtaining Help Information

 - Multi-version API Access

 - Specifying the Nearest Endpoint

 - Filtering Return Results

 - Result Polling

- Using Advanced Features

Operation Guide

Installing TCCLI

Last updated : 2023-05-30 10:39:35

This document describes how to install Tencent Cloud Command Line Interface (TCCLI).

Prerequisites

You have installed the Python environment and pip tool.

Note:

Python 2.7 or later is required. For more information, visit the [Python website](#) and [pip website](#).

TCCLI depends on the TencentCloudApi Python SDK. If the version of the SDK is earlier than that of TCCLI, the SDK is automatically upgraded during TCCLI installation. Check the latest TCCLI version at [GitHub](#).

Directions

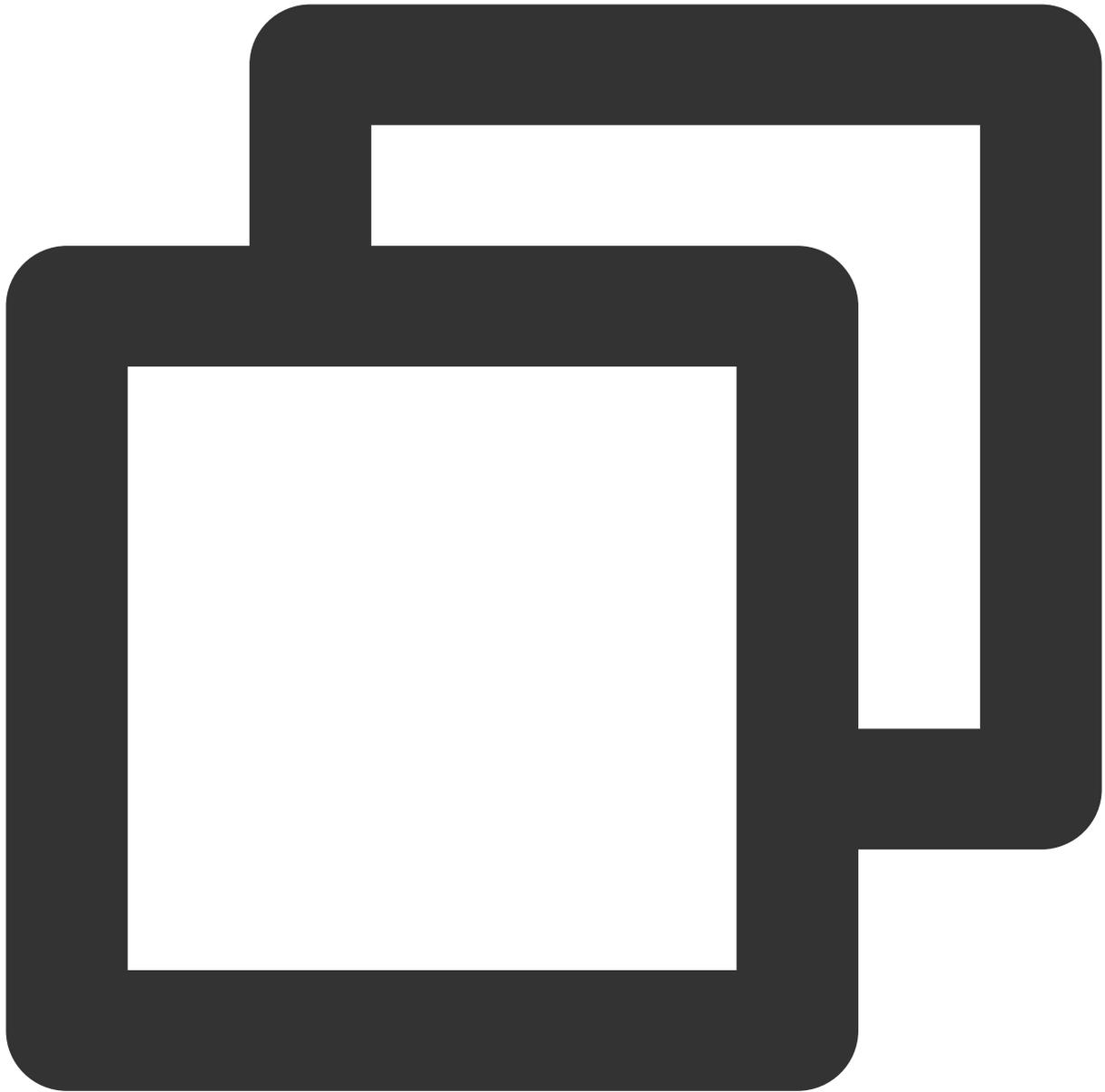
You can install TCCLI in the following ways:

Using pip (recommended)

Using source code

Using Homebrew

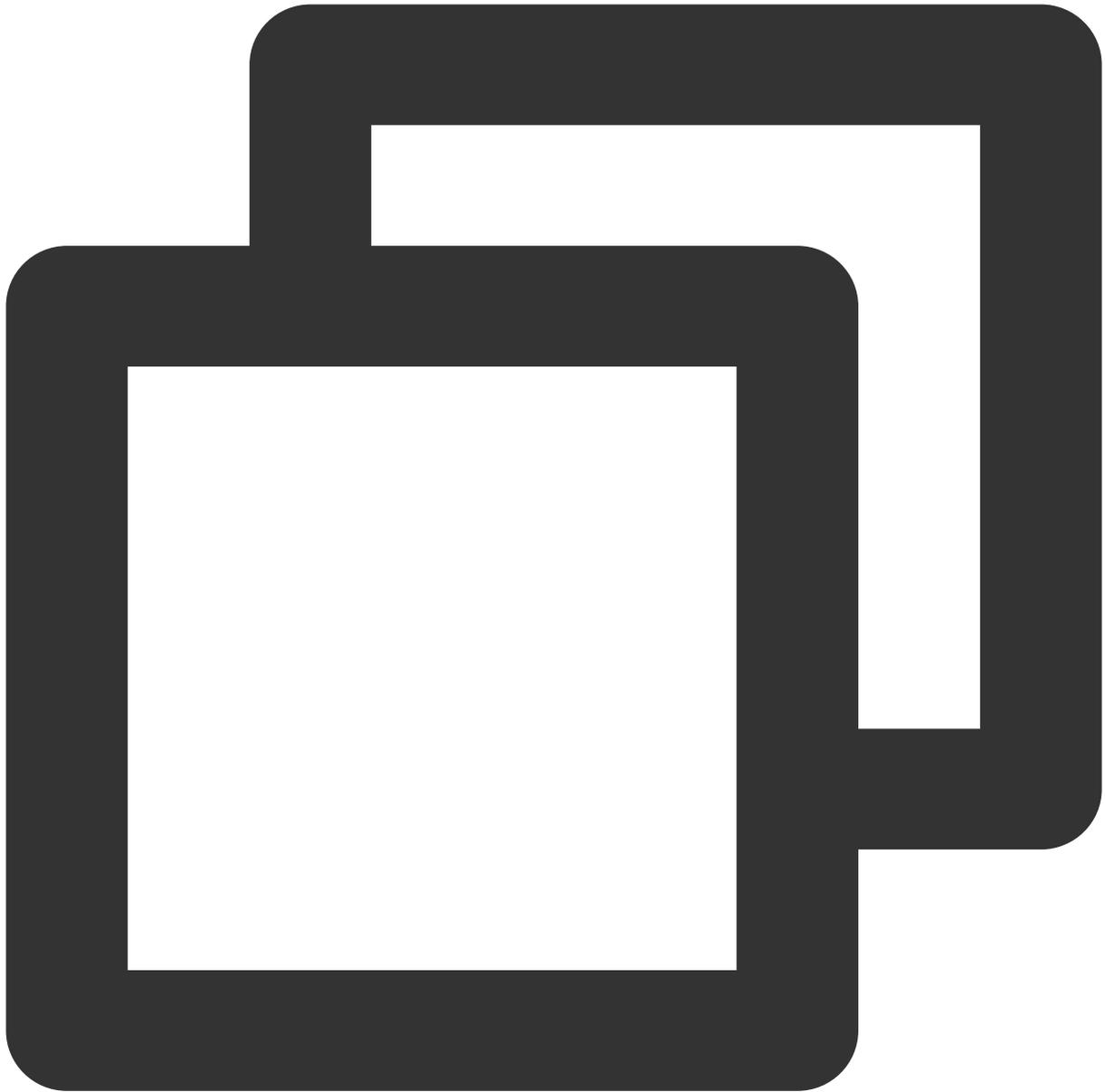
1. The installation command is the same on Windows, macOS, and Linux. Open the command line window on your operating system.
2. Run the following command to install TCCLI:



```
sudo pip install tccli
```

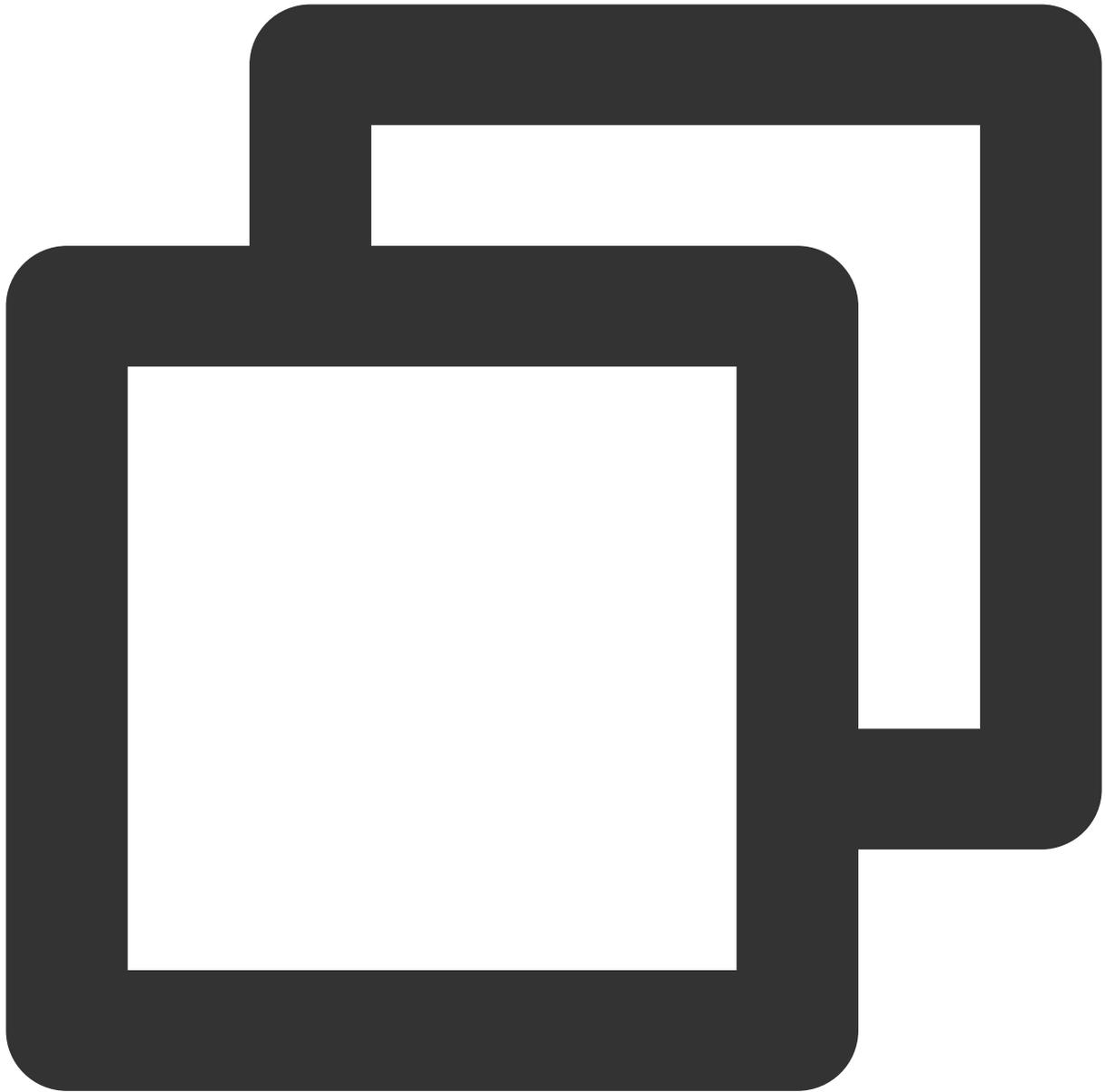
Note:

If you upgrade TCCLI from a version earlier than 3.0.252.3, run the following commands:



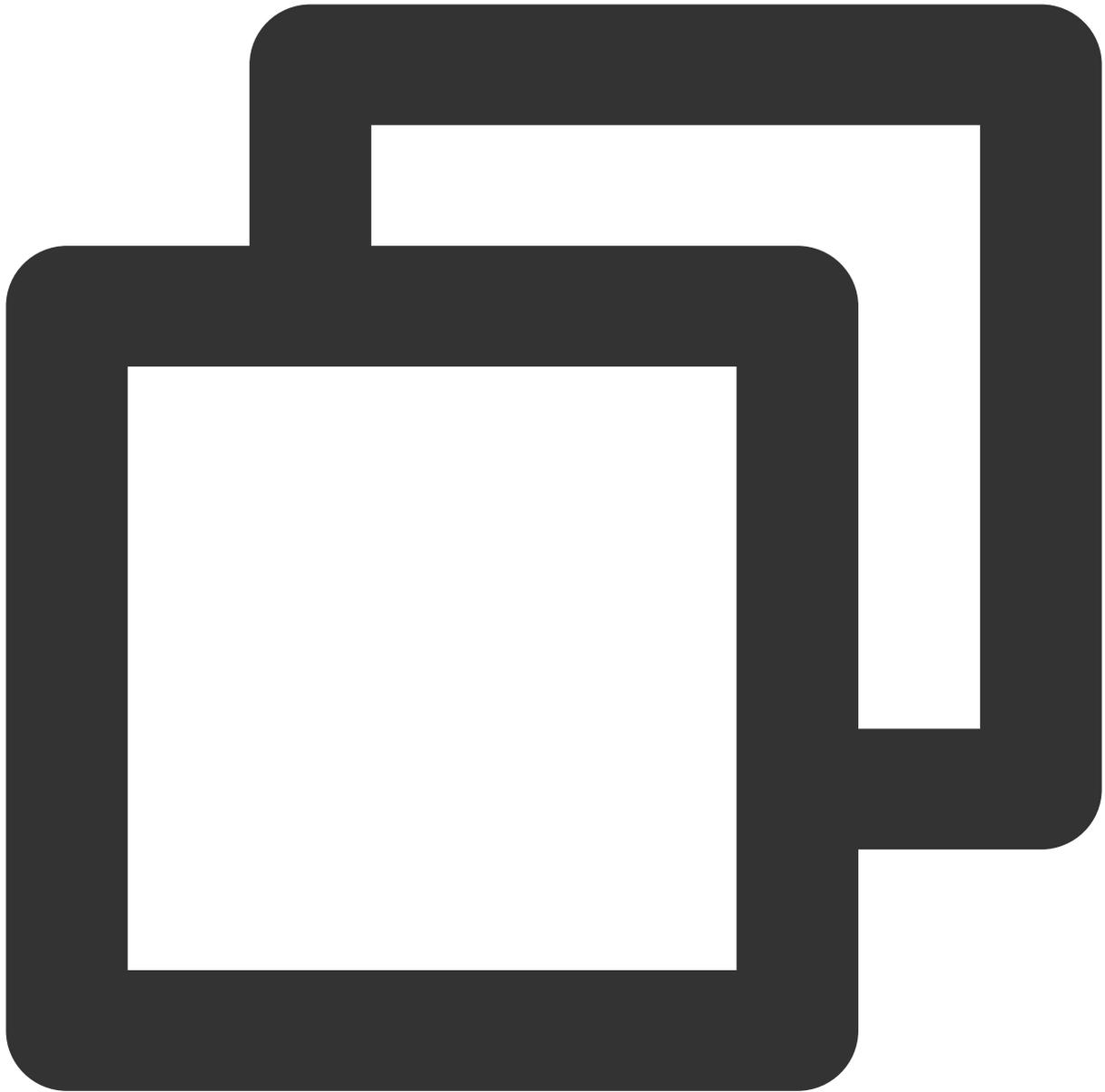
```
sudo pip uninstall tccli jmespath  
sudo pip install tccli
```

3. After the installation is completed, run the following command to check whether the installation is successful:

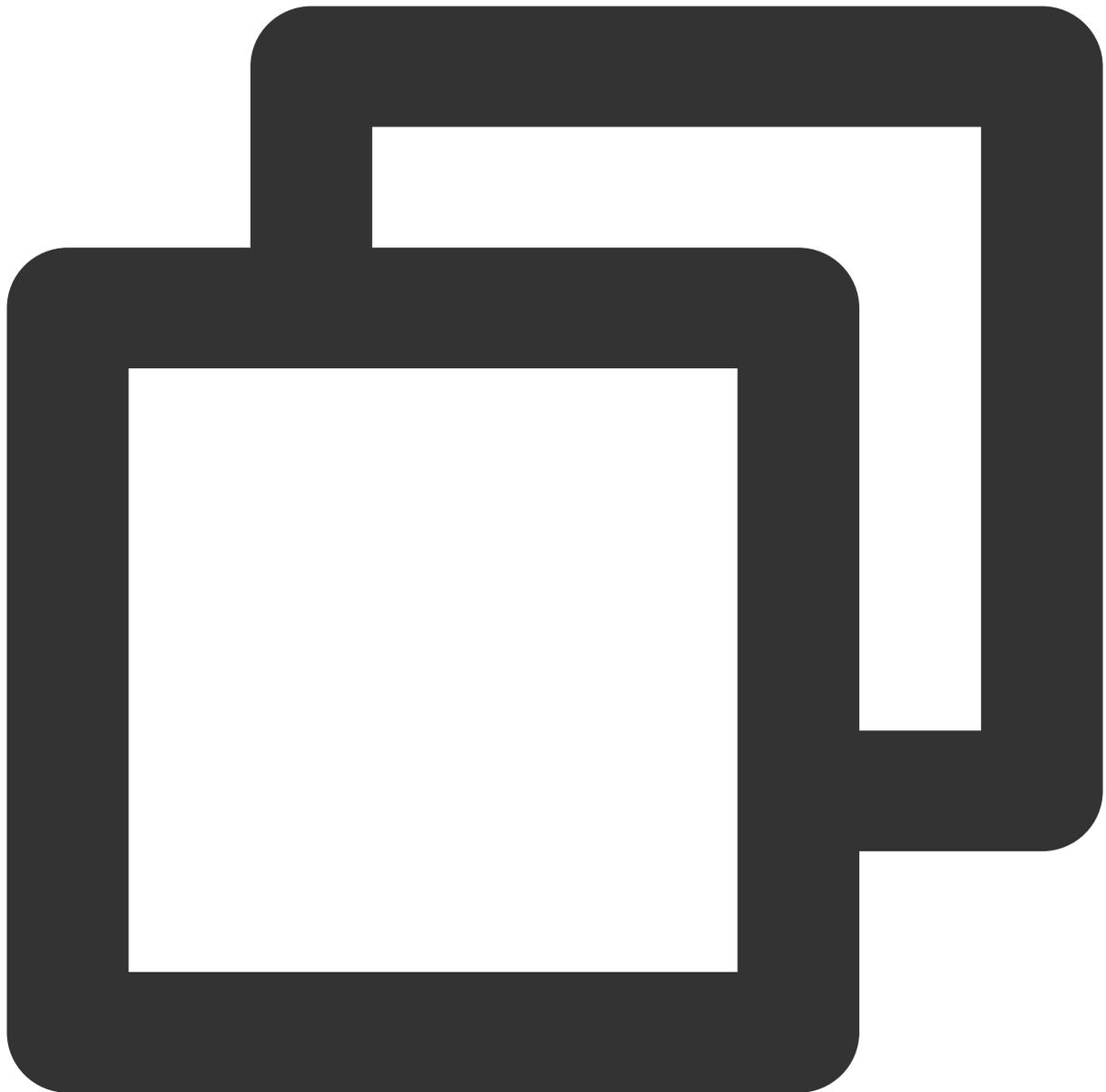


```
tccli --version
```

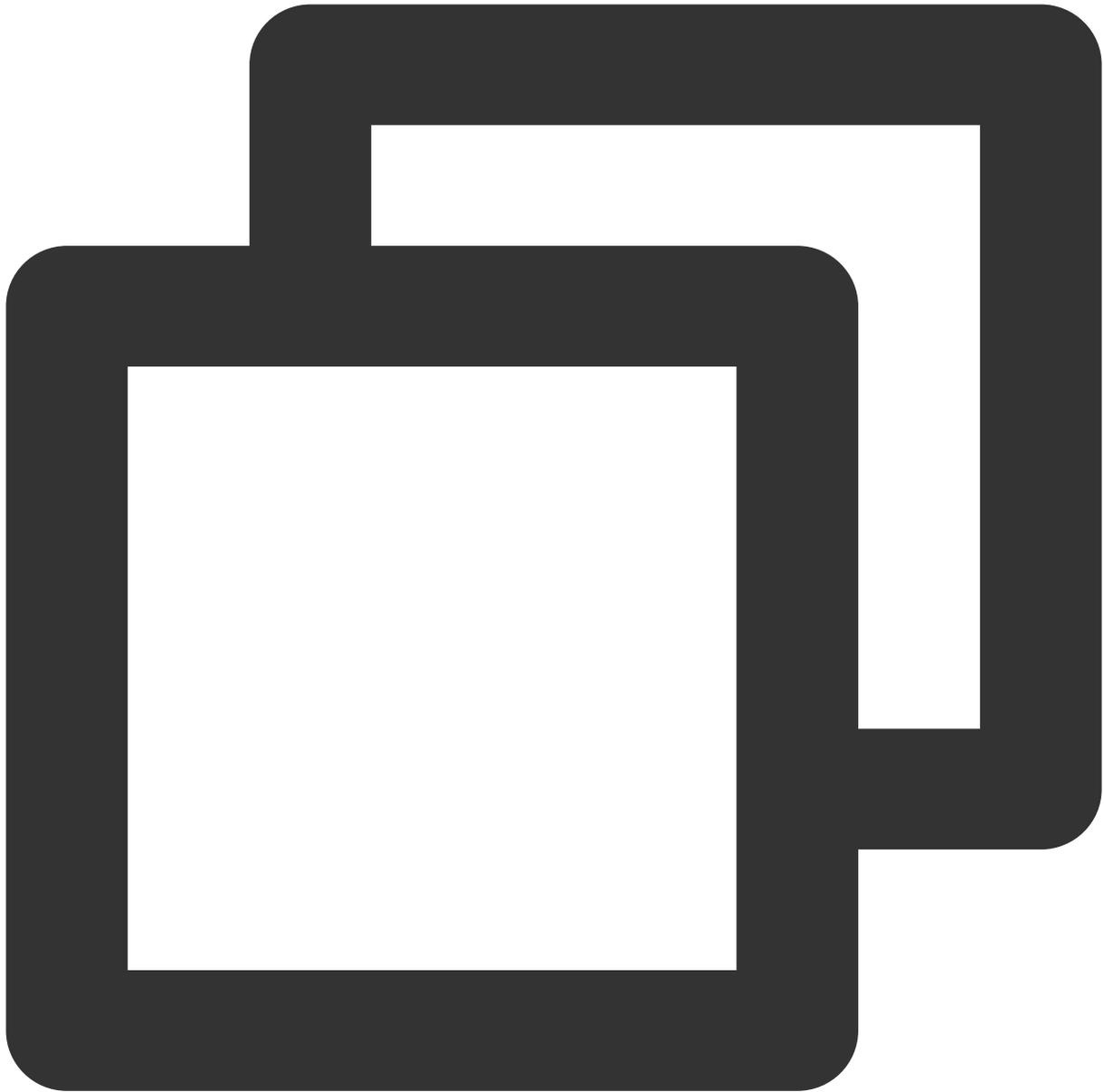
Run the following commands in sequence to download the TCCLI project from [tencentcloud-cli](#) and to install TCCLI by using the `setup.py` script.



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



```
cd tencentcloud-cli
```

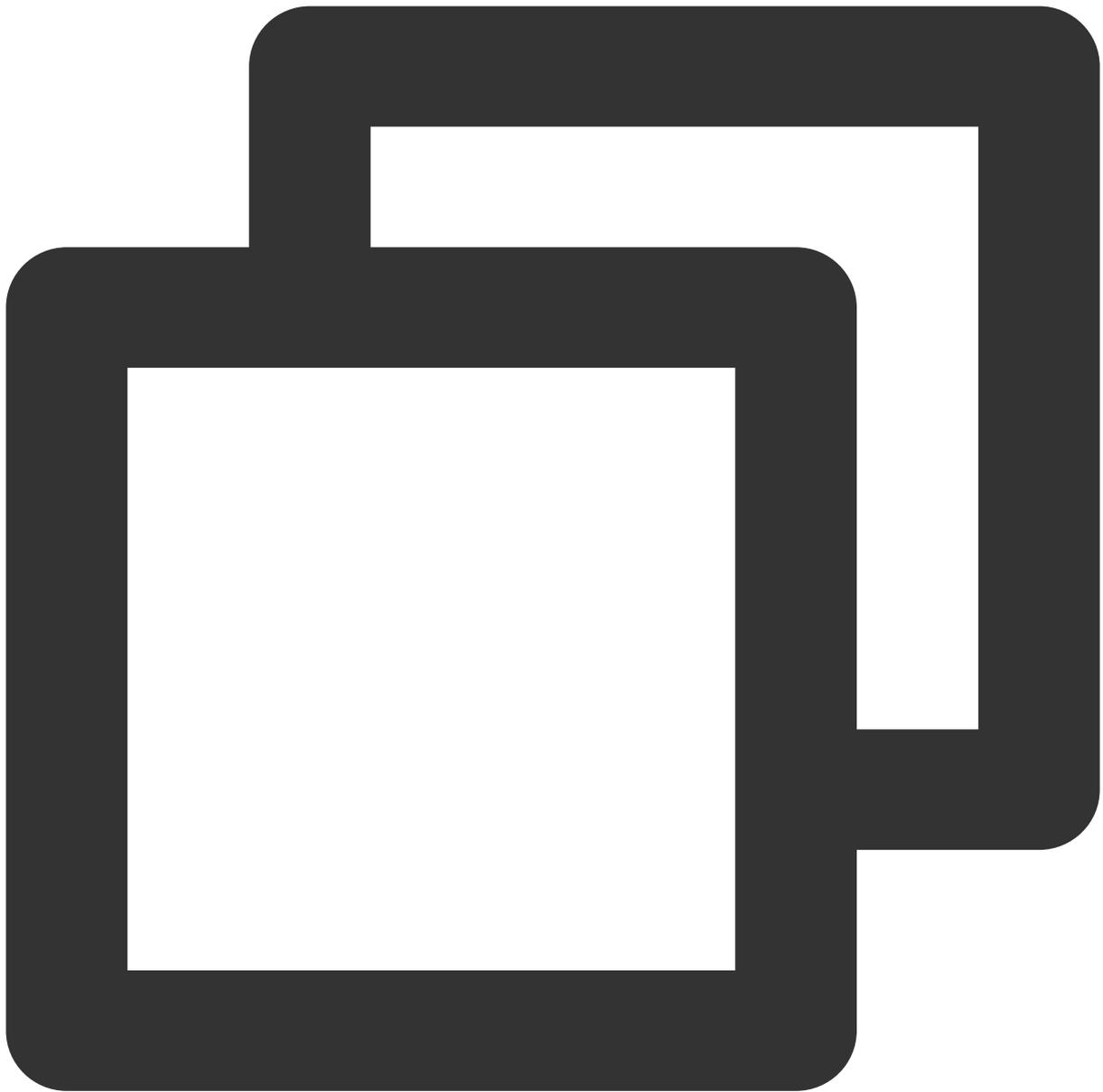


```
python setup.py install
```

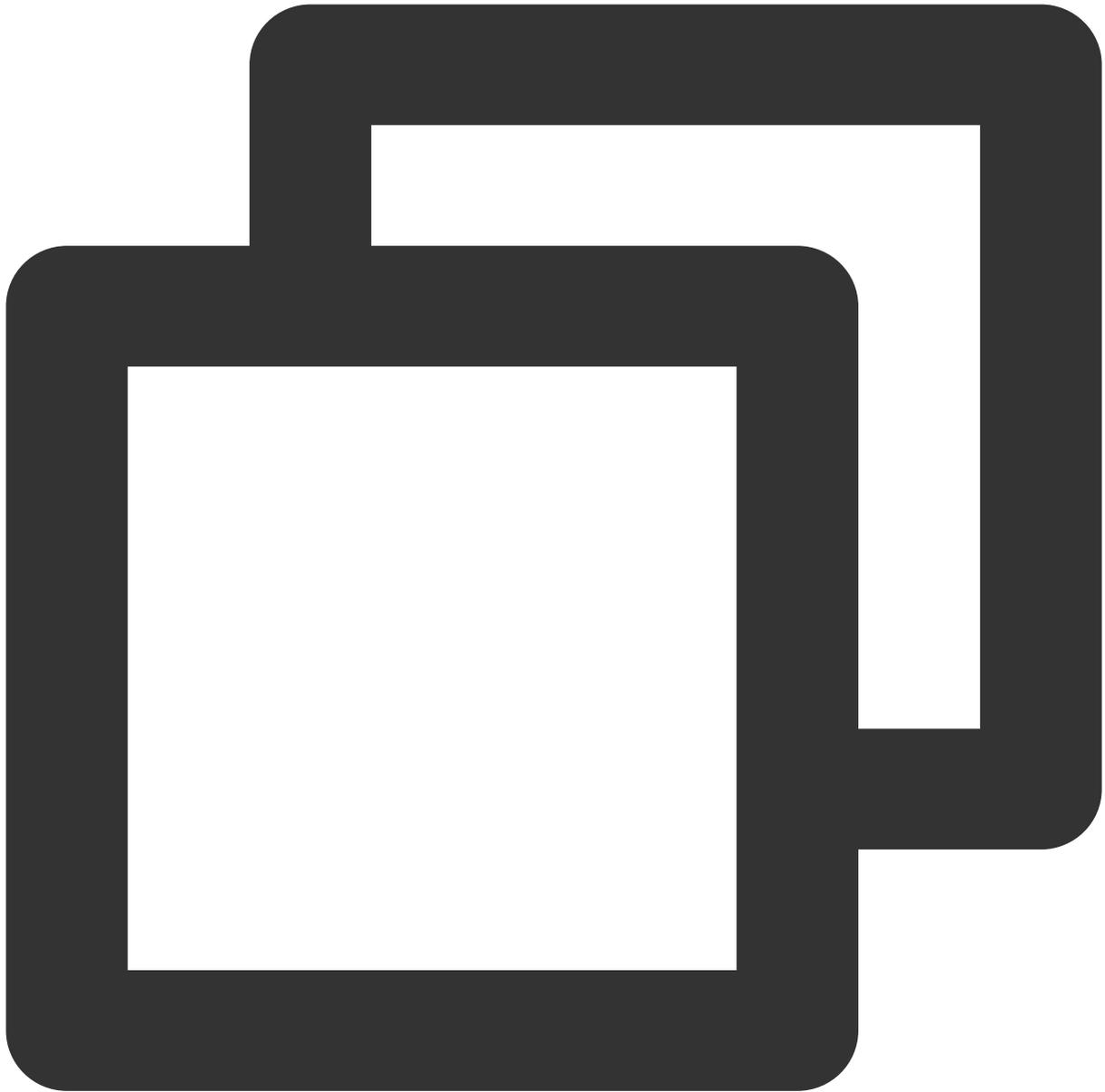
Note:

This method applies only to macOS. You need to install Homebrew first. See the installation method at the [Homebrew website](#).

Run the following commands in sequence:

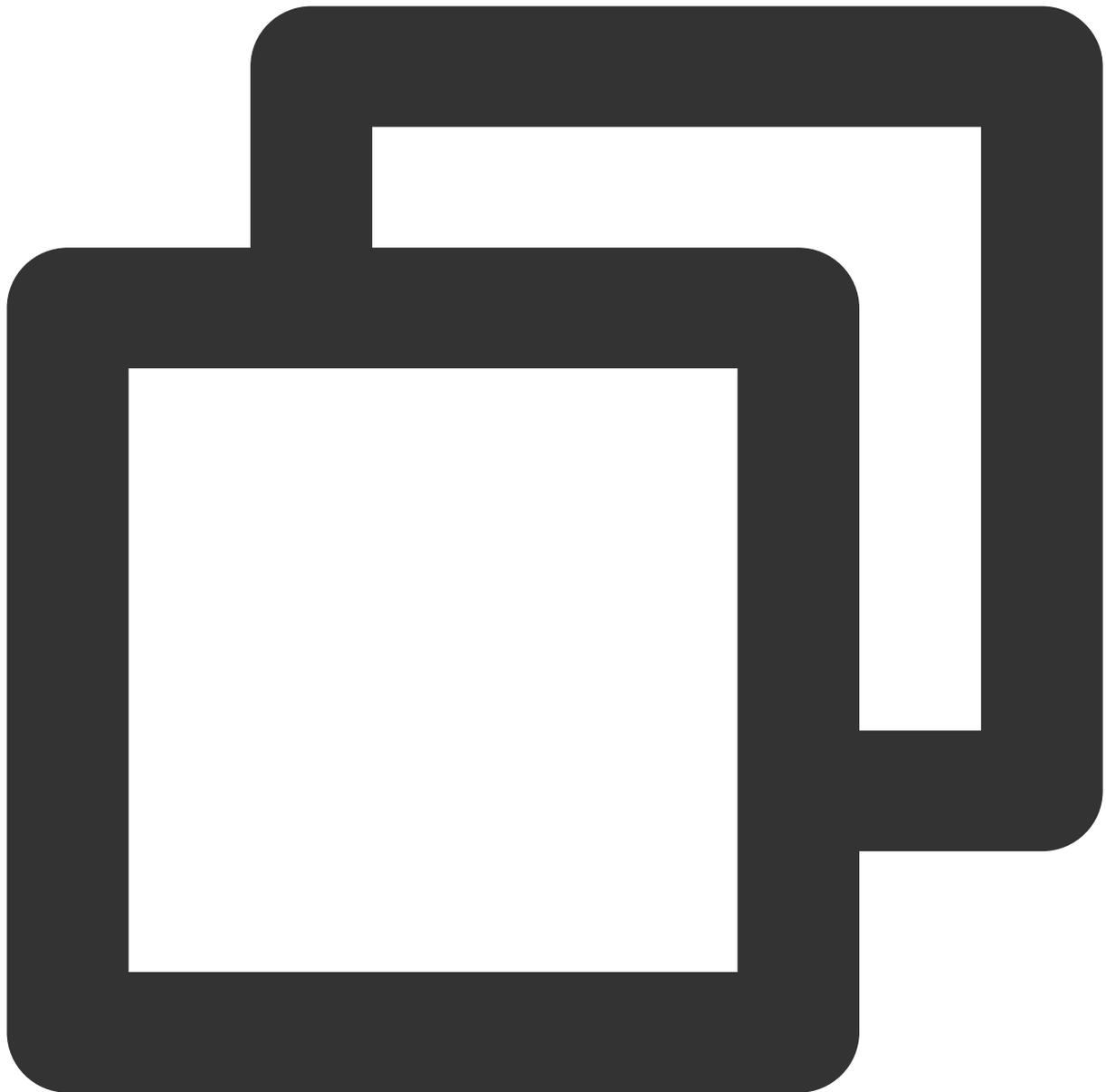


```
brew tap tencentcloud/tccli
```



```
brew install tccli
```

You can run the following command to upgrade TCCLI:



```
brew upgrade
```

Configuring TCCLI

TCCLI Configuration Method

Last updated : 2023-02-21 10:41:35

This document describes how to initialize the configuration of Tencent Cloud Command Line Interface (TCCLI) in interactive and command line modes, and how to switch between accounts.

Prerequisites

You have installed TCCLI. For more information, see [Installing TCCLI](#).

Directions

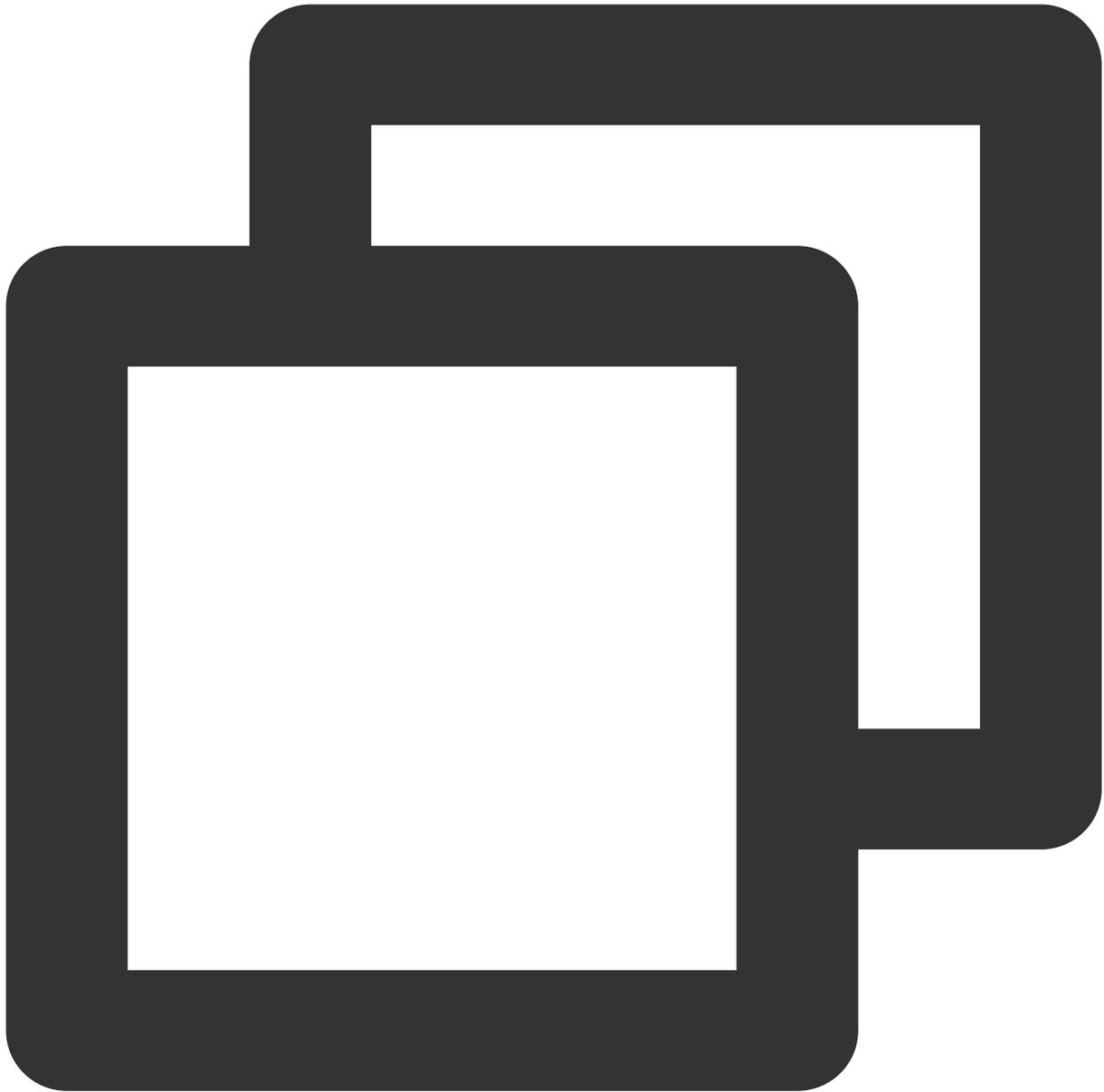
Before using TCCLI, you must initialize its configuration to ensure that the prerequisites for using TencentCloud API are met.

Note:

The secretId, secretKey, and region used in this document are only examples. Replace them with actual information.

Interactive mode

You can enter the interactive mode for quick configuration by running the `tccli configure` command.



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

secretId: the SecretId of your TencentCloud API key, which can be obtained at [Manage API Key](#).

secretKey: the SecretKey of your TencentCloud API key, which can be obtained at [Manage API Key](#).

region: the region where your Tencent Cloud product resides. Go to [APIs](#) and obtain available regions from the page of your Tencent Cloud product. For example, you can obtain [Region List](#) from the page of Cloud Virtual Machine

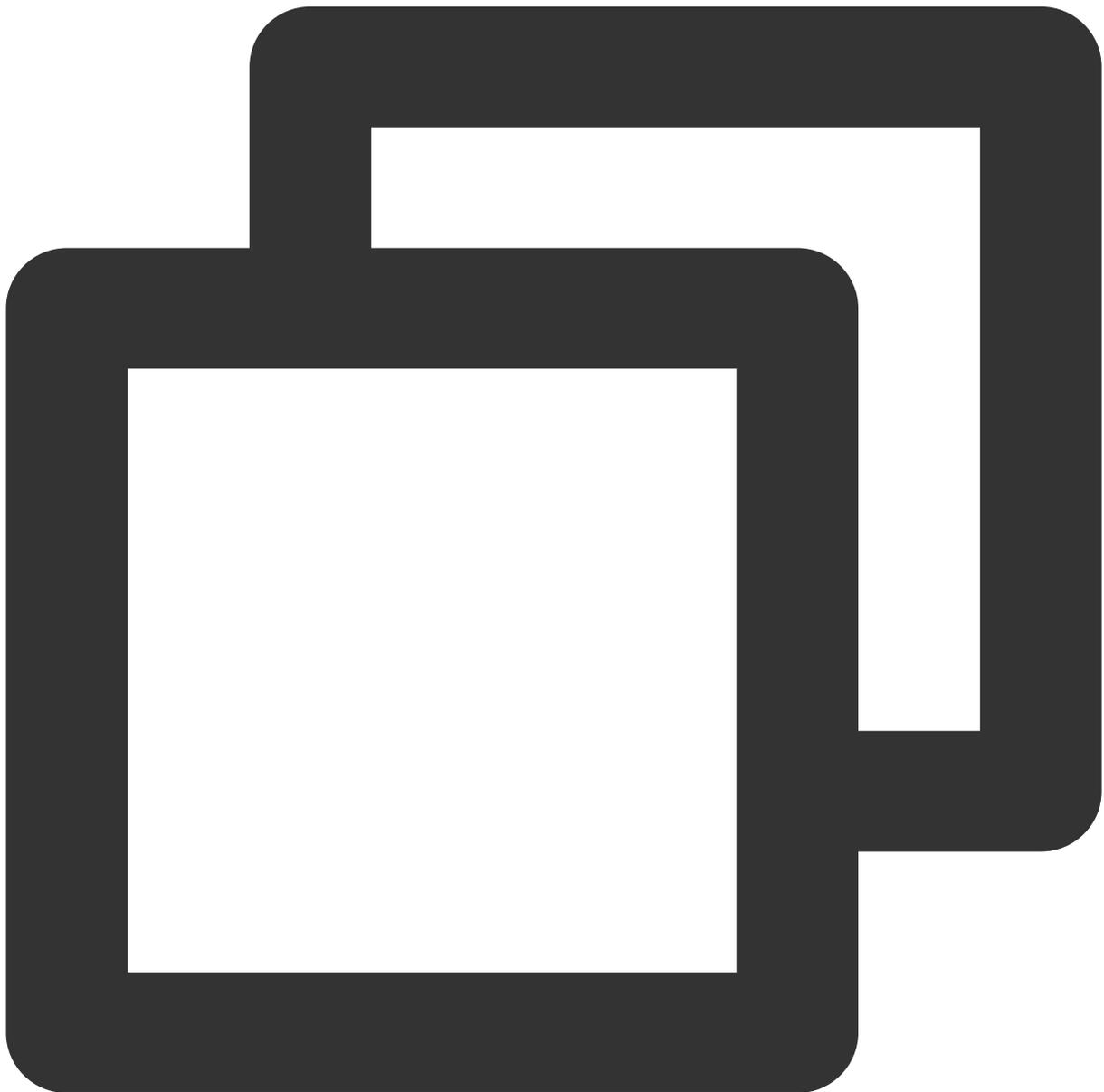
(CVM).

output: the output format of the request return packet, which is optional. Valid values: json, table, text. Default value: json.

For more information, run the `tccli configure help` command.

Command line mode

In command line mode, you can configure the information in an automated script.

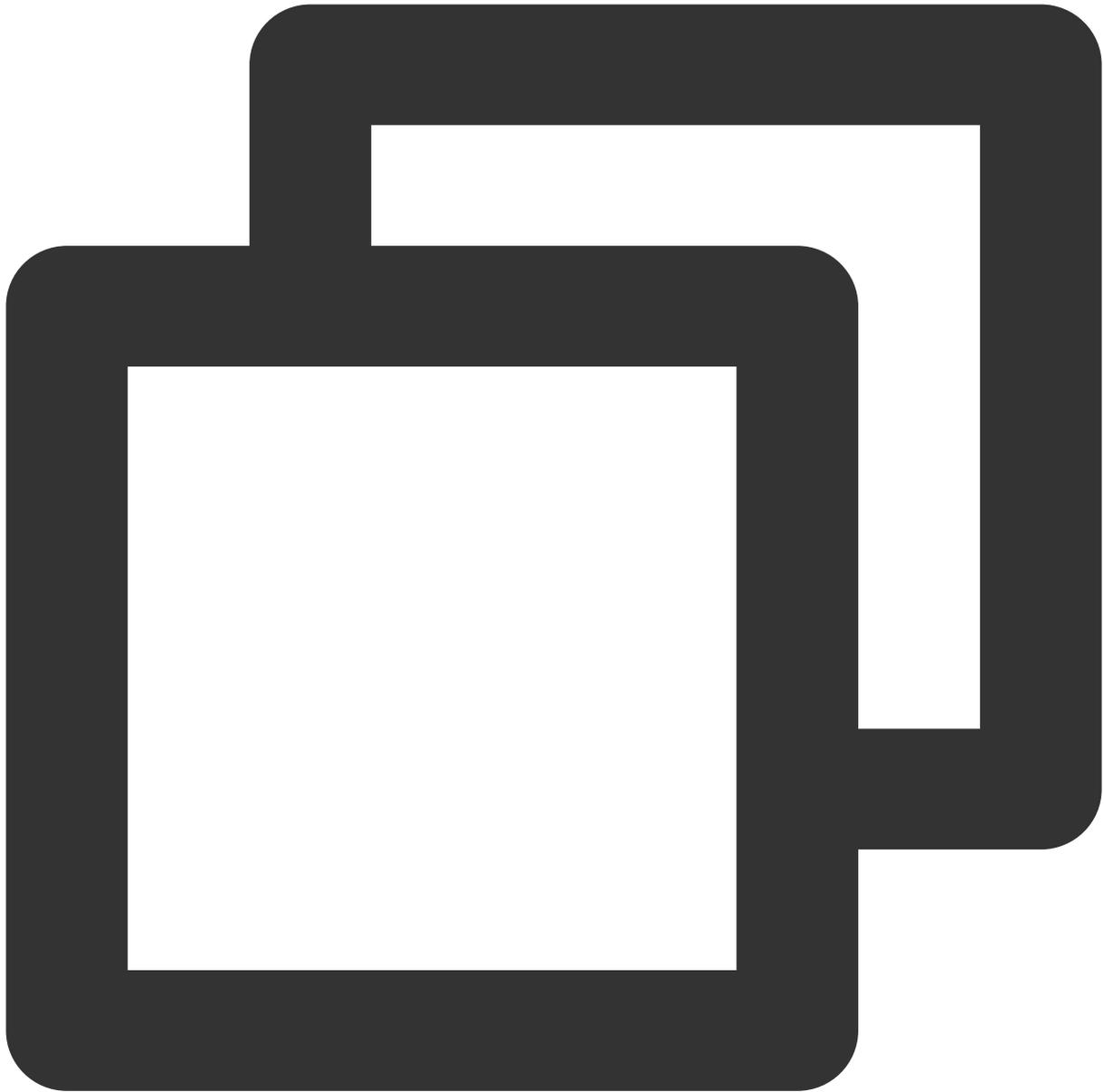


```
# The `set` subcommand is used to configure one or more items.  
$ tccli configure set secretId AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
```

```
$ tccli configure set region ap-guangzhou output json
# The `get` subcommand is used to obtain configuration information.
$ tccli configure get secretKey
secretKey = Gu5t9xGARNpq86cd98joQYCN3*****
# The `list` subcommand is used to print all configuration information.
$ tccli configure list
credential:
secretId = AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secretKey = Gu5t9xGARNpq86cd98joQYCN3*****
configure:
region = ap-guangzhou
output = json
# Specify the values of secretId and secretKey in the command line. For example, yo
$ tccli cvm DescribeInstances --secretId AKIDz8krbsJ5yKBZQpn74WFkmLPx3***** --secr
```

Supporting multiple accounts

TCCLI supports multiple accounts, making it easier for you to use multiple configurations at the same time.



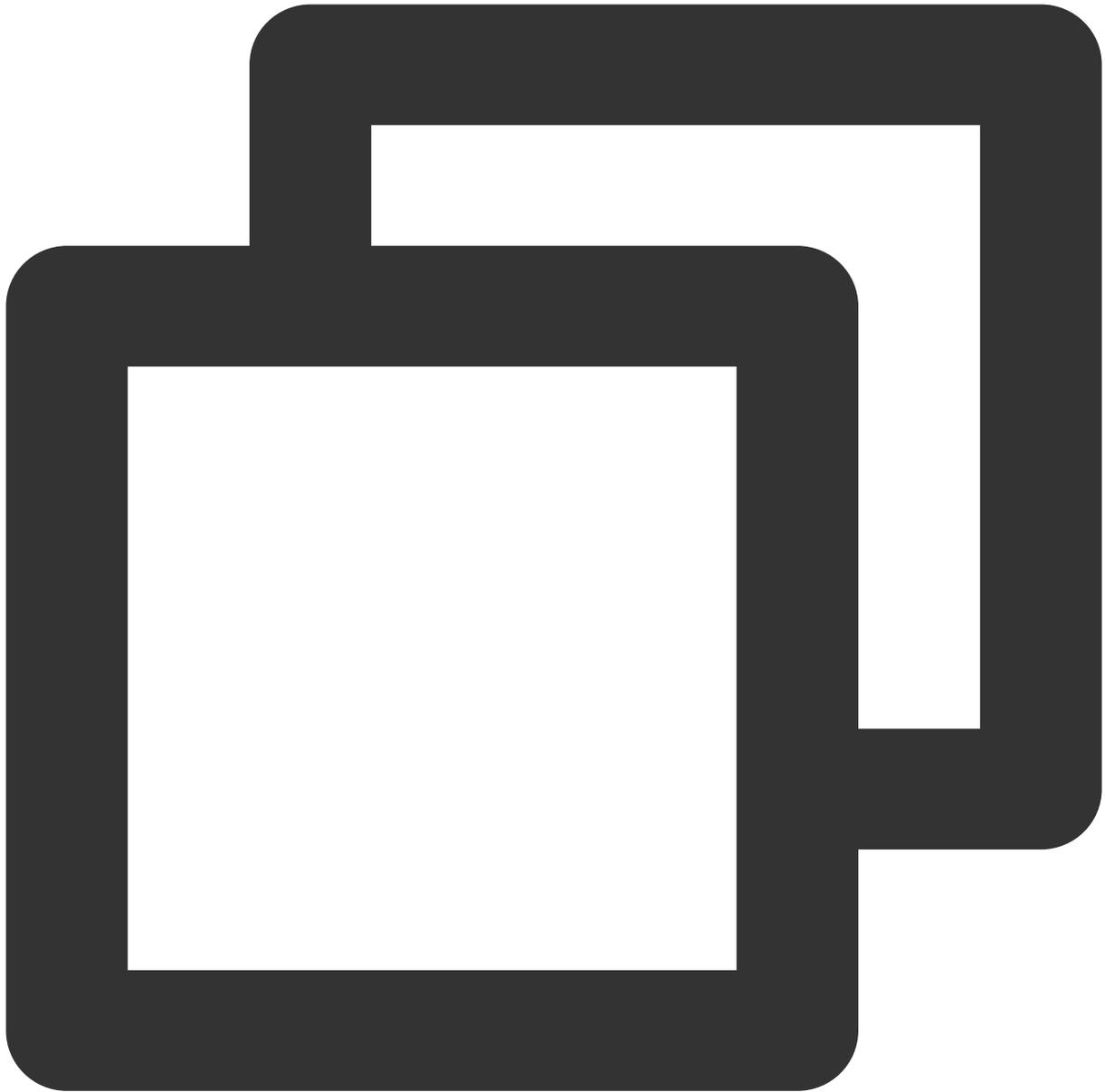
```
# Specify the account name `test` in interactive mode.
$ tccli configure --profile test
TencentCloud API secretId [*BCDP]:AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
TencentCloud API secretKey [*ArFd]:Gu5t9xGARNpq86cd98joQYCN3*****
region: ap-guangzhou
output[json]:
# Specify the account name `test` for set/get/list subcommands.
$ tccli configure set region ap-guangzhou output json --profile test
$ tccli configure get secretKey --profile test
$ tccli configure list --profile test
# Use the `remove` subcommand to delete the configuration file of the specified acc
```

```
$ tccli configure remove -profile test
# Specify an account when calling an API such as the DescribeZones API of CVM.
$ tccli cvm DescribeZones --profile test
```

TCCLI configuration file

After you run the `tccli configure` command, the `default.configure` and `default.credential` files for TCCLI are generated under the `~/.tccli` directory. Content in the two files is in JSON format.

The `default.configure` file records the version (the latest version by default) of the API to call, the endpoint (the nearest endpoint by default), the default output format, and the specified region. The `default.credential` file records your key pairs. Example:



```
# Format of the default.configure file: (The following takes CVM as an example. By
{
  ...
  "cvm": {
    "endpoint": "cvm.tencentcloudapi.com",
    "version": "2017-03-12"
  },
  ...
  "output": "json",
  "region": "ap-guanzhou",
  ...
}
```

```
}  
# Format of the default.credential file:  
{  
  "secretId": "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****",  
  "secretKey": "Gu5t9xGARNpq86cd98joQYCN3*****"  
}
```

If you specified an account name in the configuration command, the configuration files corresponding to the account name are generated. For example, after you run the `tccli configure --profile test` command, the `test.configure` and `test.credential` files are generated.

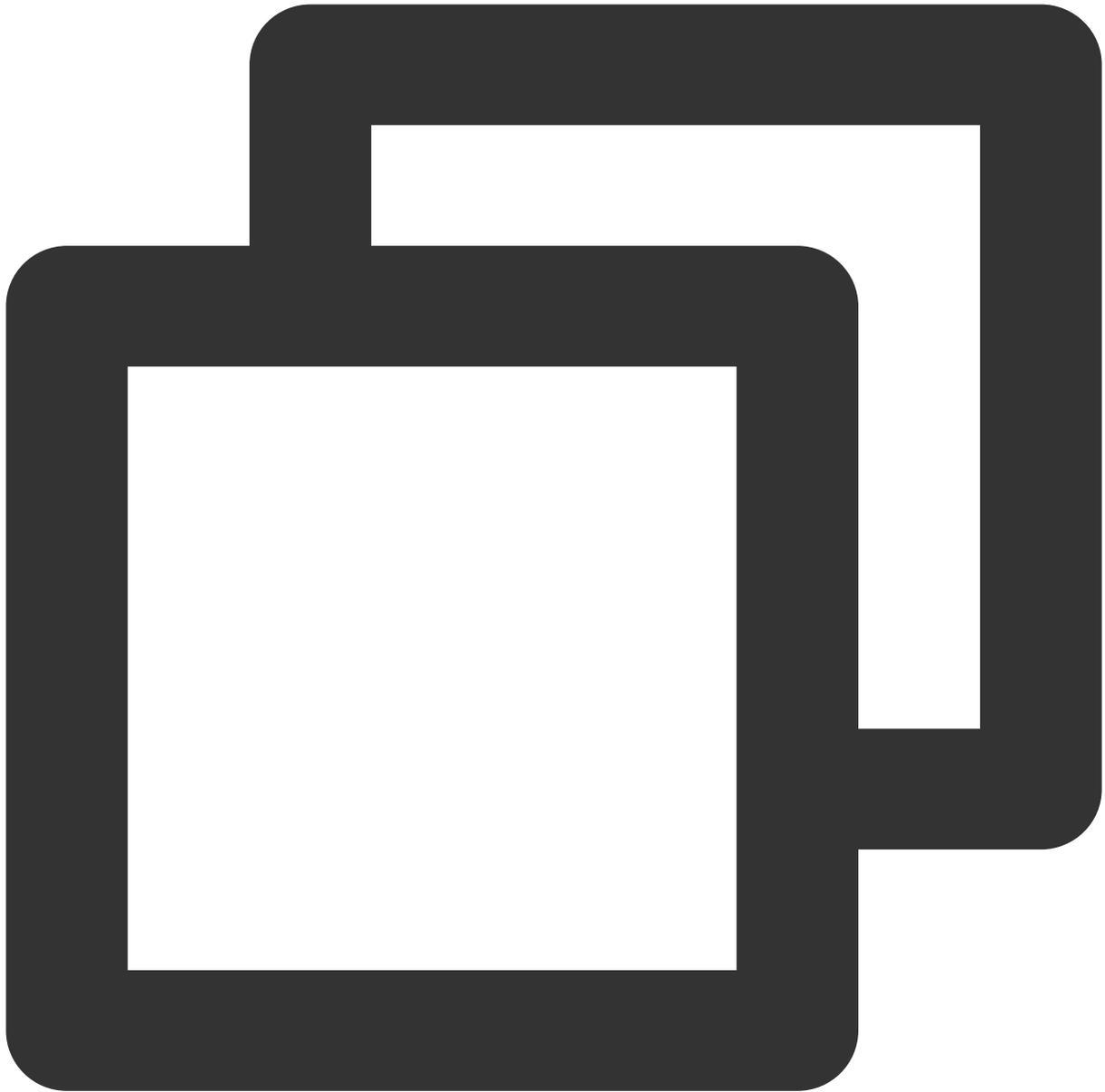
To modify content in the configuration files, you can directly edit the files or use the `set` subcommand. For example, you can run the `tccli configure set cvm.version 2017-03-12` command to set the version of CVM APIs to call to the default version 2017-03-12.

Configuring environment variables

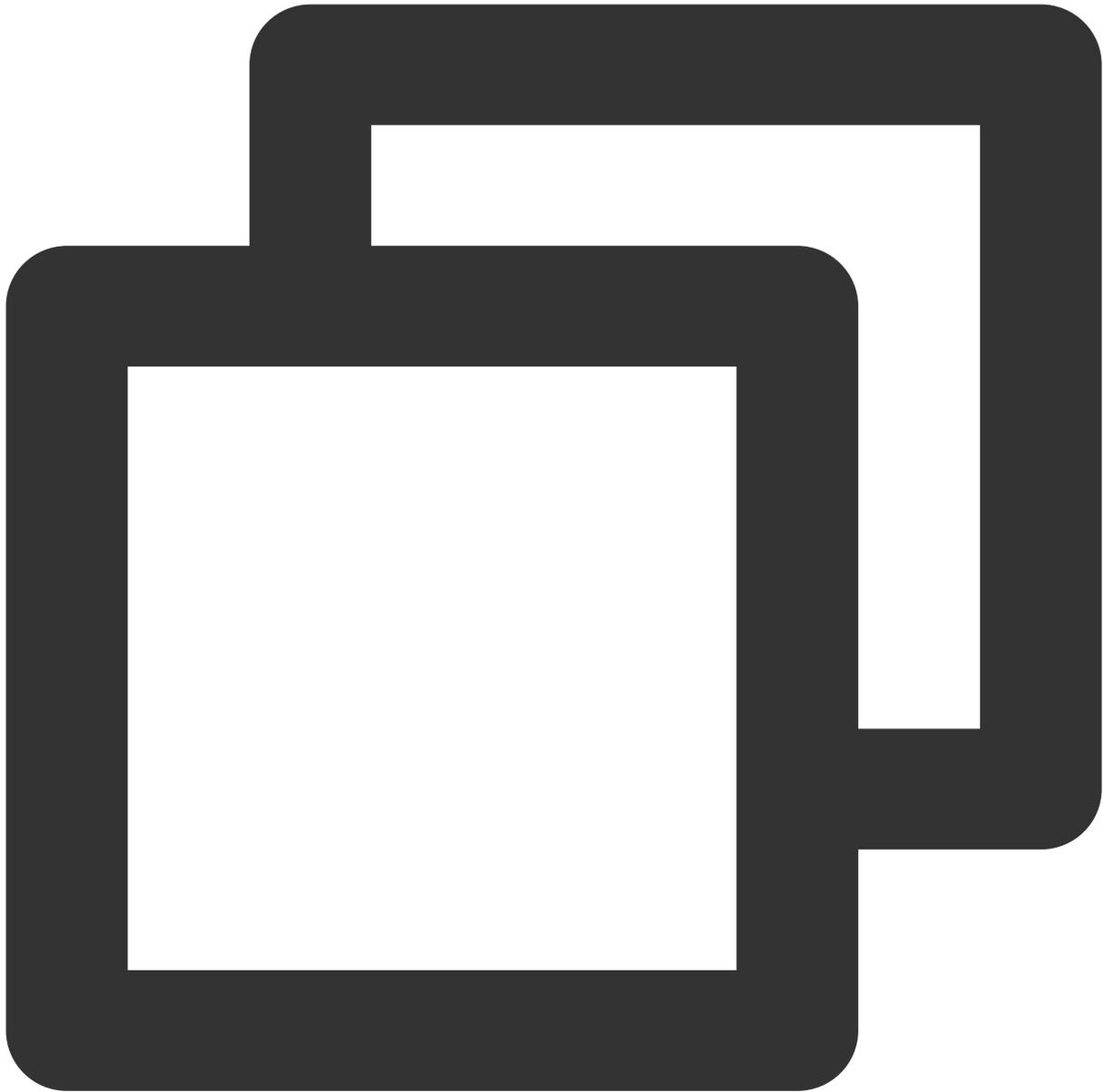
TCCLI supports configuring TencentCloud API key pairs in environment variables to ensure the security of your information. For example, in Linux, you can configure a key pair in the following ways:

Use `export` command (temporary)

Write into profile file (permanent)



```
# Set a TencentCloud API key SecretId.  
$ export TENCENTCLOUD_SECRET_ID=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****  
# Set a TencentCloud API key SecretKey.  
$ export TENCENTCLOUD_SECRET_KEY=Gu5t9xGARNpq86cd98joQYCN3*****  
# Set the region of a Tencent Cloud product.  
$ export TENCENTCLOUD_REGION=ap-guangzhou
```



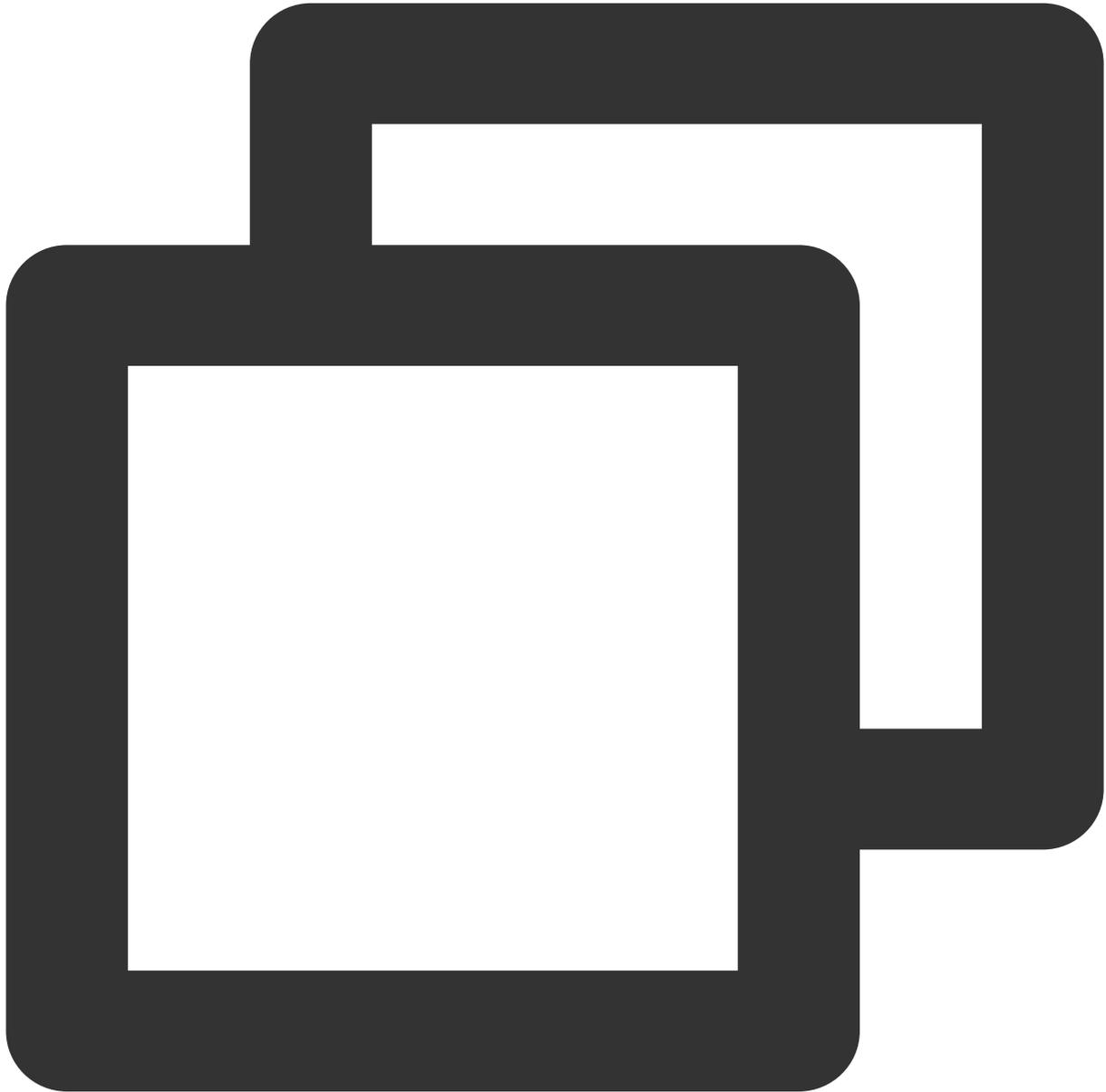
```
# Edit the /etc/profile file, and write the following content into the file:
export TENCENTCLOUD_SECRET_ID=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
export TENCENTCLOUD_SECRET_KEY=Gu5t9xGARNpq86cd98joQYCN3*****
export TENCENTCLOUD_REGION=ap-guangzhou
# Then run the following command for the environment variables to take effect:
$ source /etc/profile
```

Note:

If you specify the same information in the command, configuration file, and environment variables, the priority order in TCCLI is command > configuration file > environment variables.

Other configurations

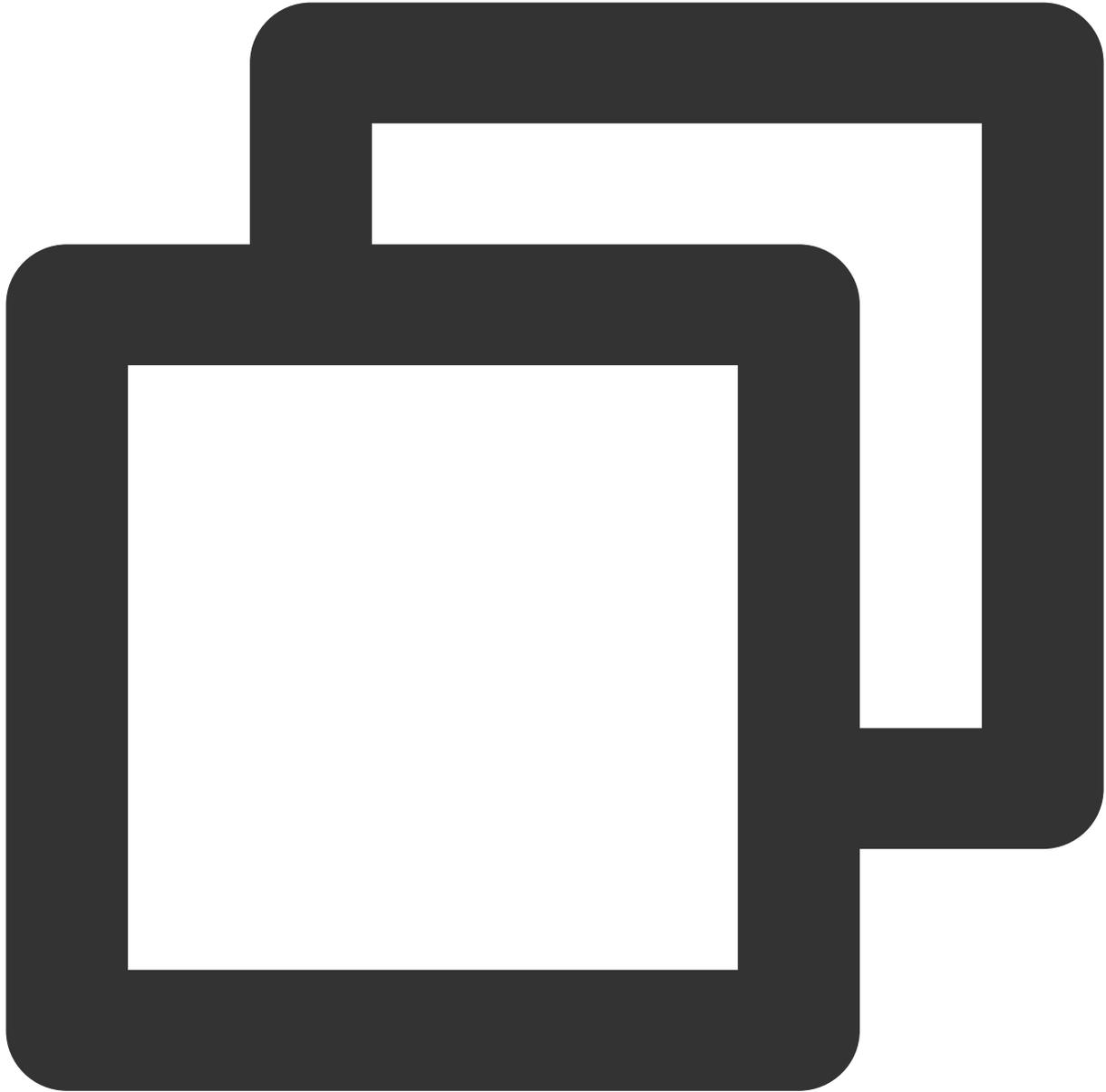
TCCLI supports authentication through Cloud Access Management (CAM) roles. For more information, see [Role Overview](#).



```
# CAM role configuration is not available in interactive mode. Use the non-interactive
$ tccli configure set role-arn qcs::cam::uin/*****/**** role-session-name ***
```

The `role-arn` and `role-session-name` fields support the `get` and `list` subcommands of `configure`. You can write values into the configuration file, or specify values in the command line, in the same

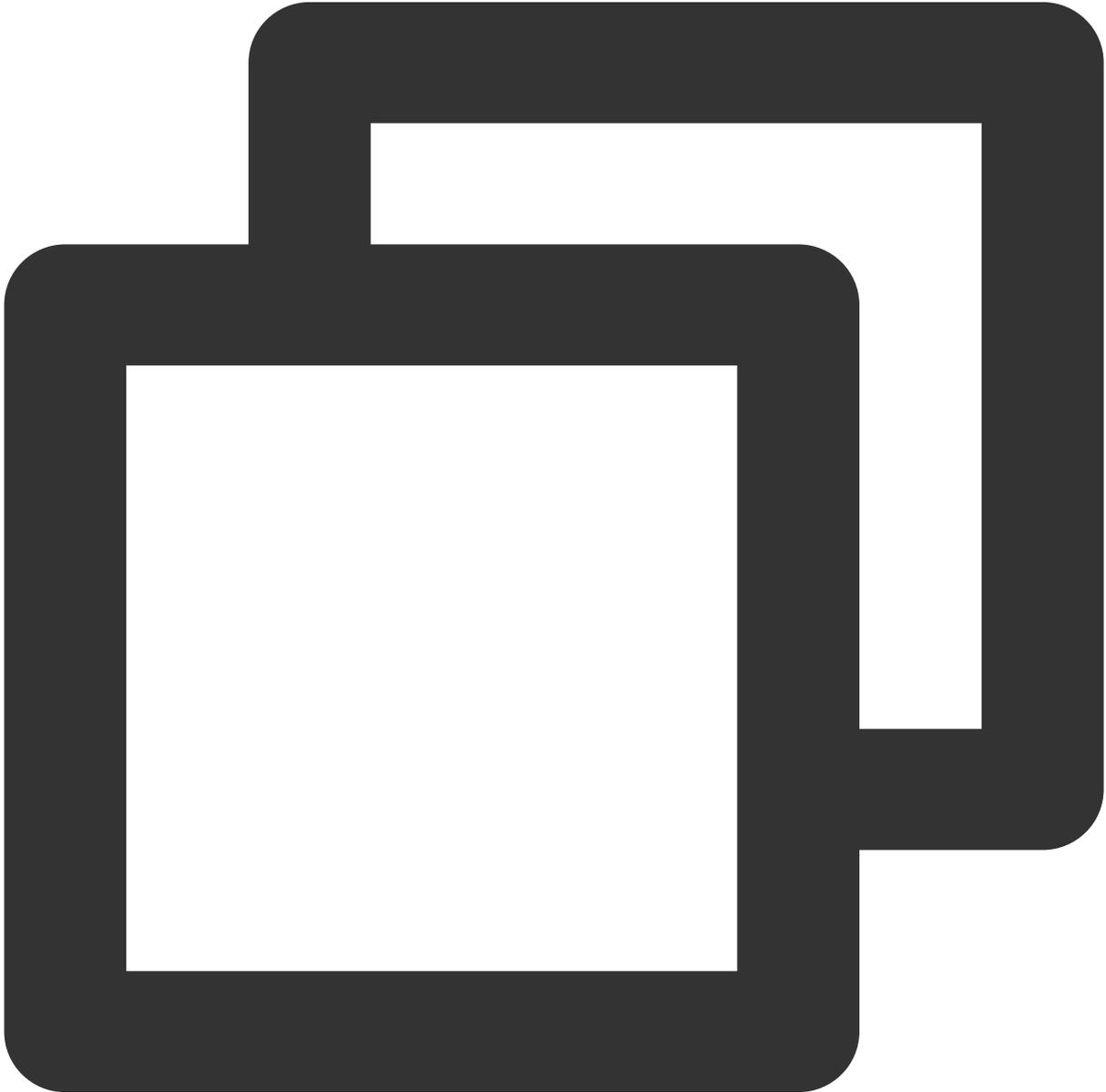
way that you configure `secretId` and `secretKey` , as shown below:



```
# Use the `get` subcommand to obtain configuration information.
$ tccli configure get role-arn
role-arn = qcs::cam::uin/*****/****
# Use the `list` subcommand to print all configuration information.
$ tccli configure list
credential:
role-arn = qcs::cam::uin/*****/****
role-session-name = ****
# Write the configuration information into environment variables.
```

```
$ export TENCENTCLOUD_ROLE_ARN=qcs::cam::uin/*****/****
$ export TENCENTCLOUD_ROLE_SESSION_NAME=****
# Specify the `role-arn` and `role-session-name` fields in the command line. The fo
$ tccli cvm DescribeZones --role-arn qcs::cam::uin/*****/**** --role-session-
```

If your instance is bound to a role, you can use the role for authentication, without providing `secretId` and `secretKey`. You can use `--use-cvm-role` to call APIs through the role.



```
# Use a role to call the DescribeZones API.
$ tccli cvm DescribeZones --use-cvm-role
```

Note:

This method applies only to instances bound to a role. For more information, see [Managing Roles](#).

Using an HTTPS Proxy

Last updated : 2023-02-21 10:41:36

This document describes how to configure and use an HTTPS proxy.

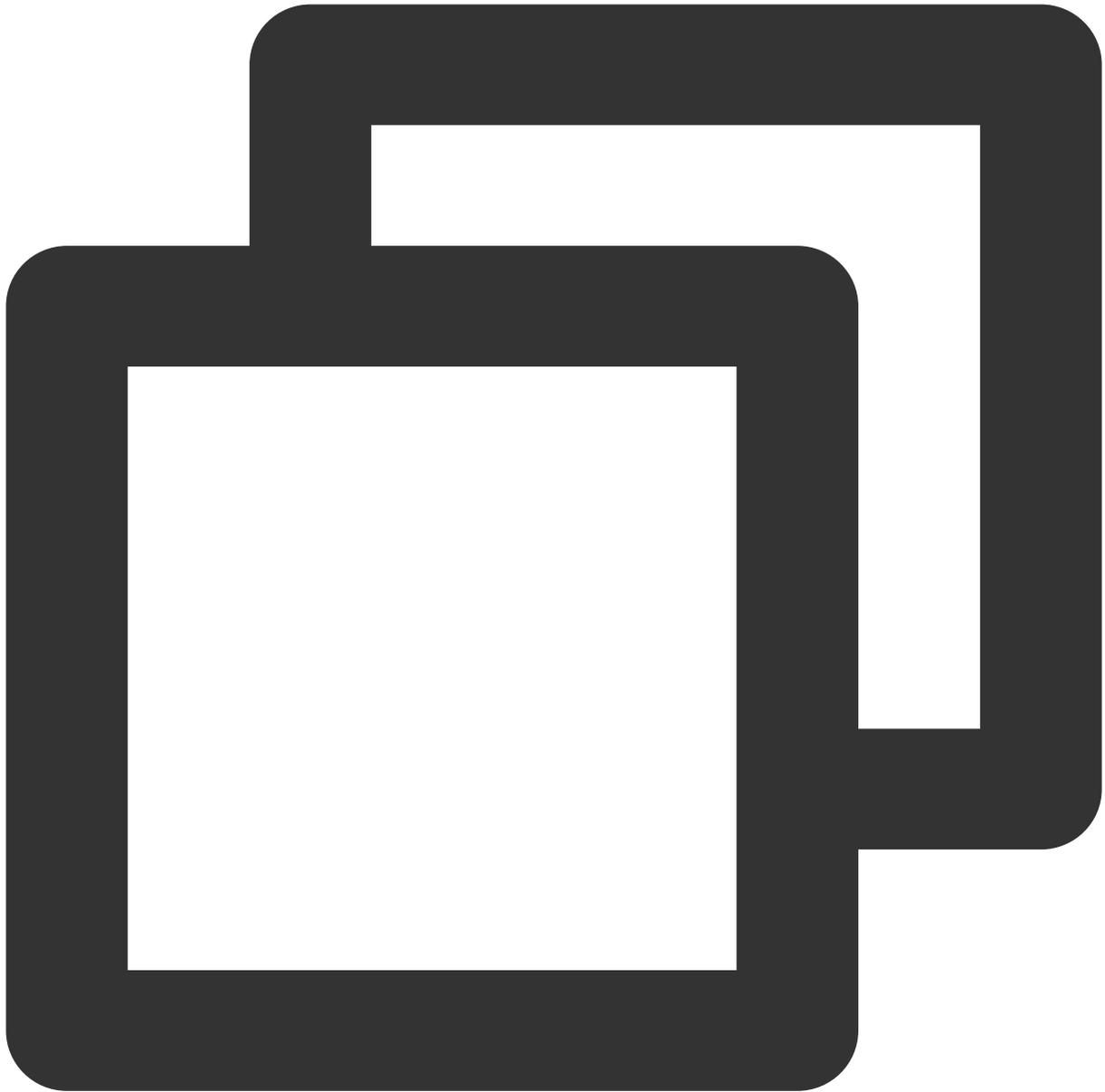
Directions

You can configure an HTTPS proxy in the following ways:

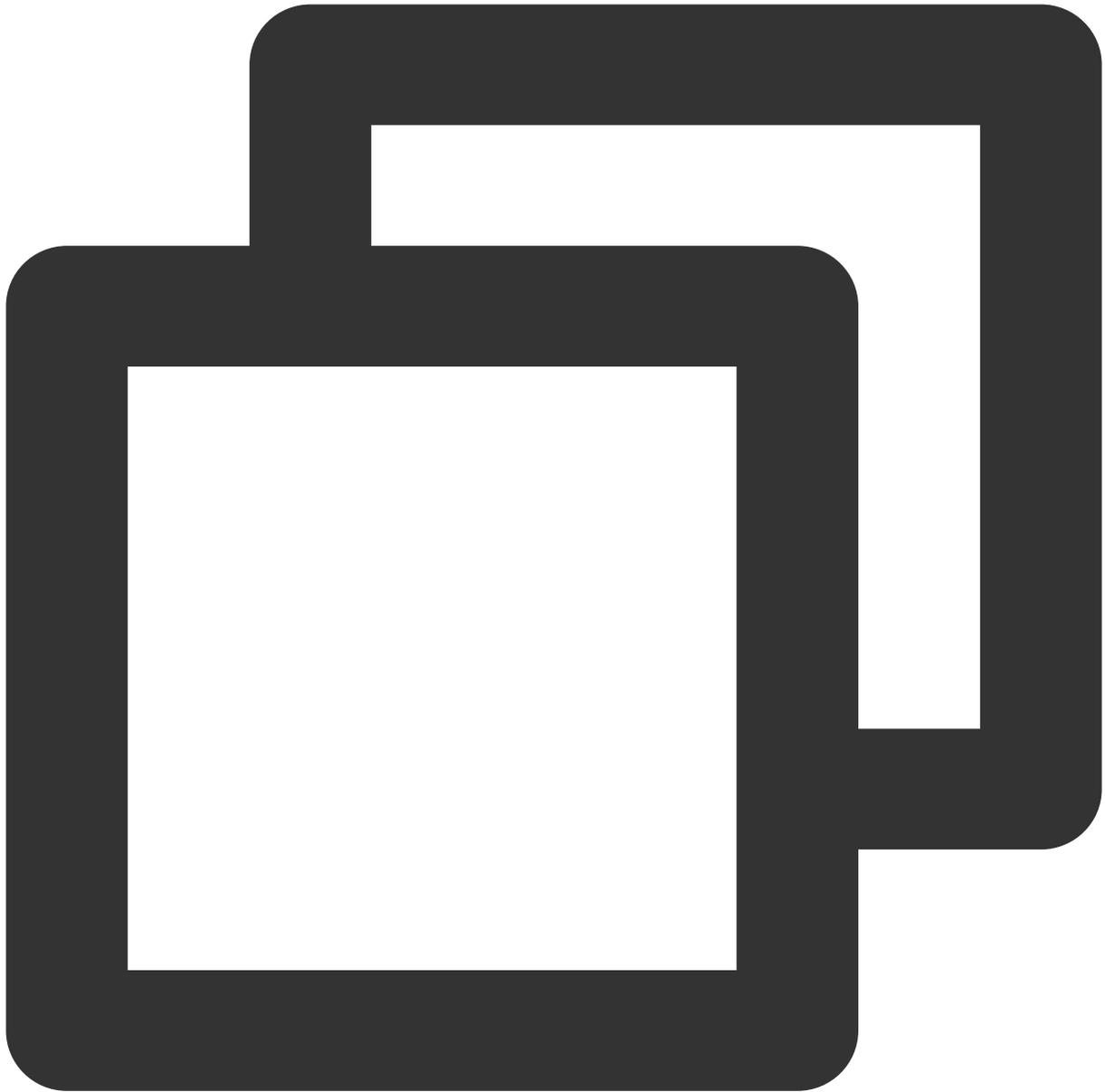
Run the following command based on the operating system that you use to configure an HTTPS proxy in an environment variable:

Linux\\Unix and macOS

Windows

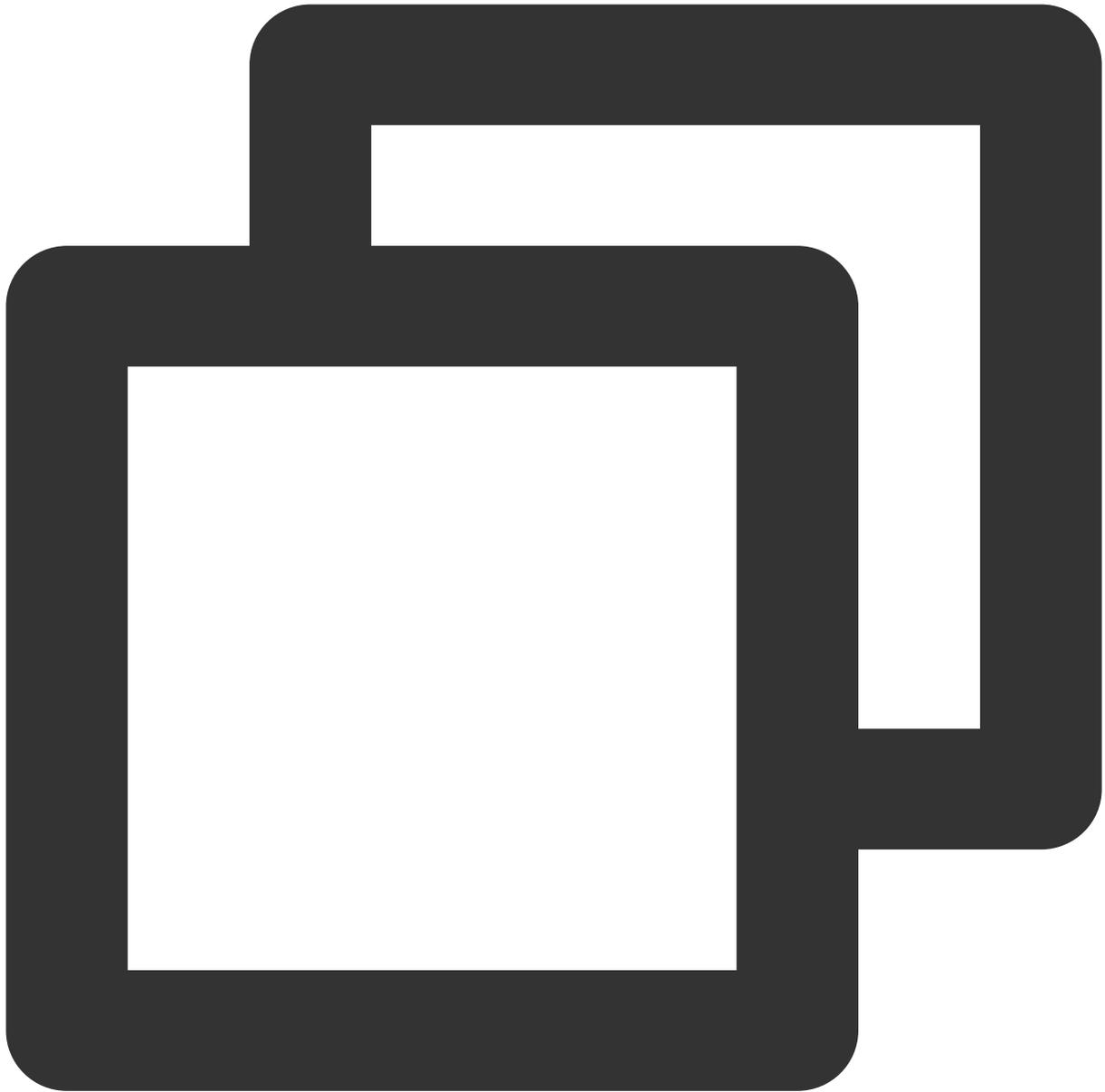


```
export https_proxy=https://192.168.1.1:1111  
export https_proxy=https://myproxy.com:1111
```



```
setx http_proxy=https://192.168.1.1:1111
set http_proxy=https://myproxy.com:1111
# setx means to set a permanent environment variable. Restart the terminal for the
```

Run the following command and use the `--https-proxy` option in the command line to configure an HTTPS proxy:



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

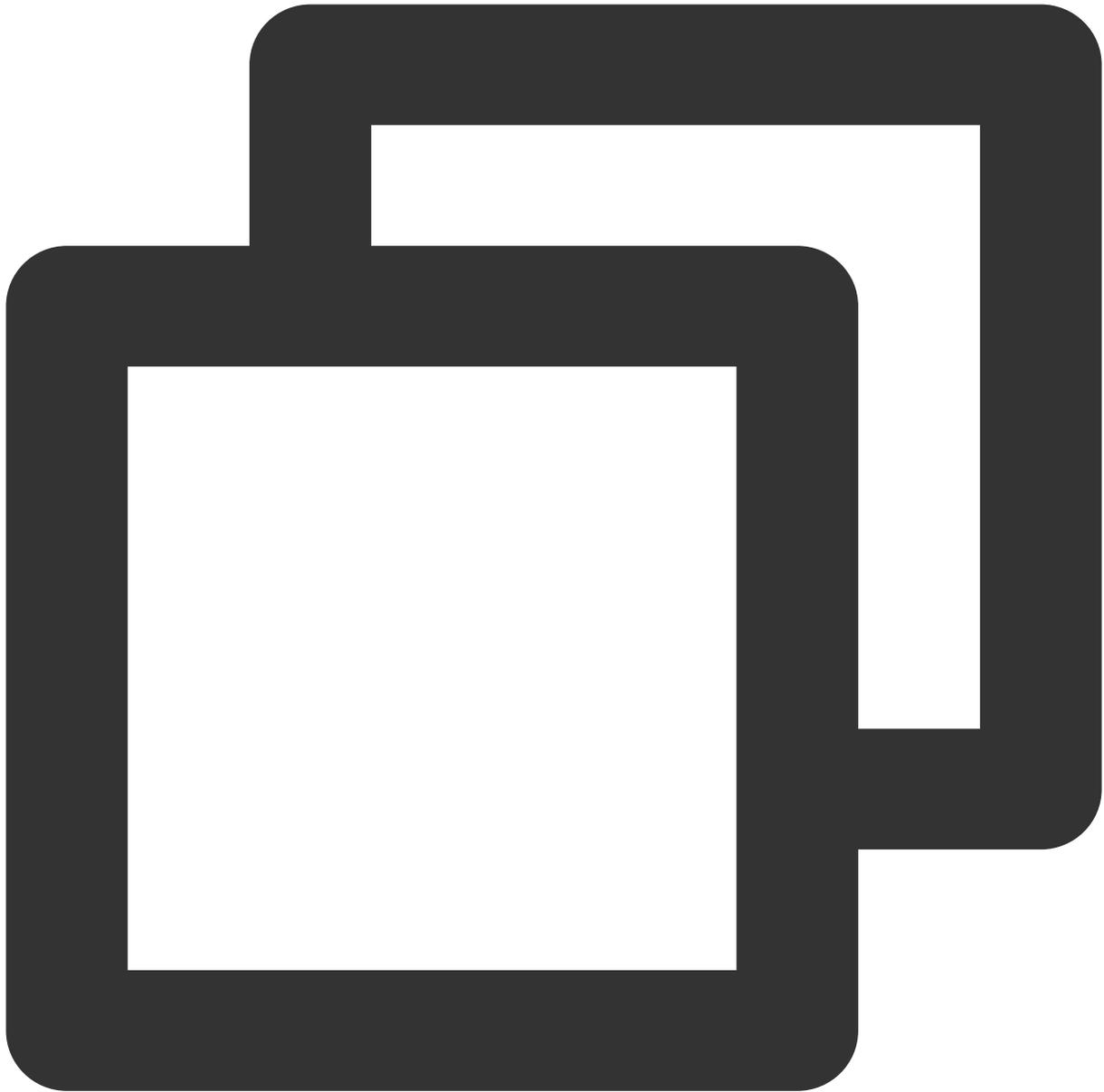
Using the Autocomplete Feature

Last updated : 2023-02-21 10:41:36

This document describes how to use the autocomplete feature of Tencent Cloud Command Line Interface (TCCLI).

Directions

In Linux, you can run the following command to enable the autocomplete feature:



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

Note:

You can also add the command to the environment variable `/etc/profile` to keep the autocomplete feature active.

Using TCCLI

TCCLI Usage

Last updated : 2023-02-21 10:41:36

Overview

This document describes how to use the basic features of TCCLI.

TCCLI integrates all Tencent Cloud products that support TencentCloud API, and allows you to configure and manage the products. For example, you can use TCCLI to create and operate a Cloud Virtual Machine (CVM) instance, create a Cloud Block Storage (CBS) disk and view its usage, and create a Virtual Private Cloud (VPC) and add resources to it. All operations that can be performed on console pages can be performed by running commands in TCCLI.

Examples:

Run the `tccli cvm DescribeInstances` command to view CVM instances under the current account.

Run the `tccli cbs DescribeDisks` command to view the list of CBS disks.

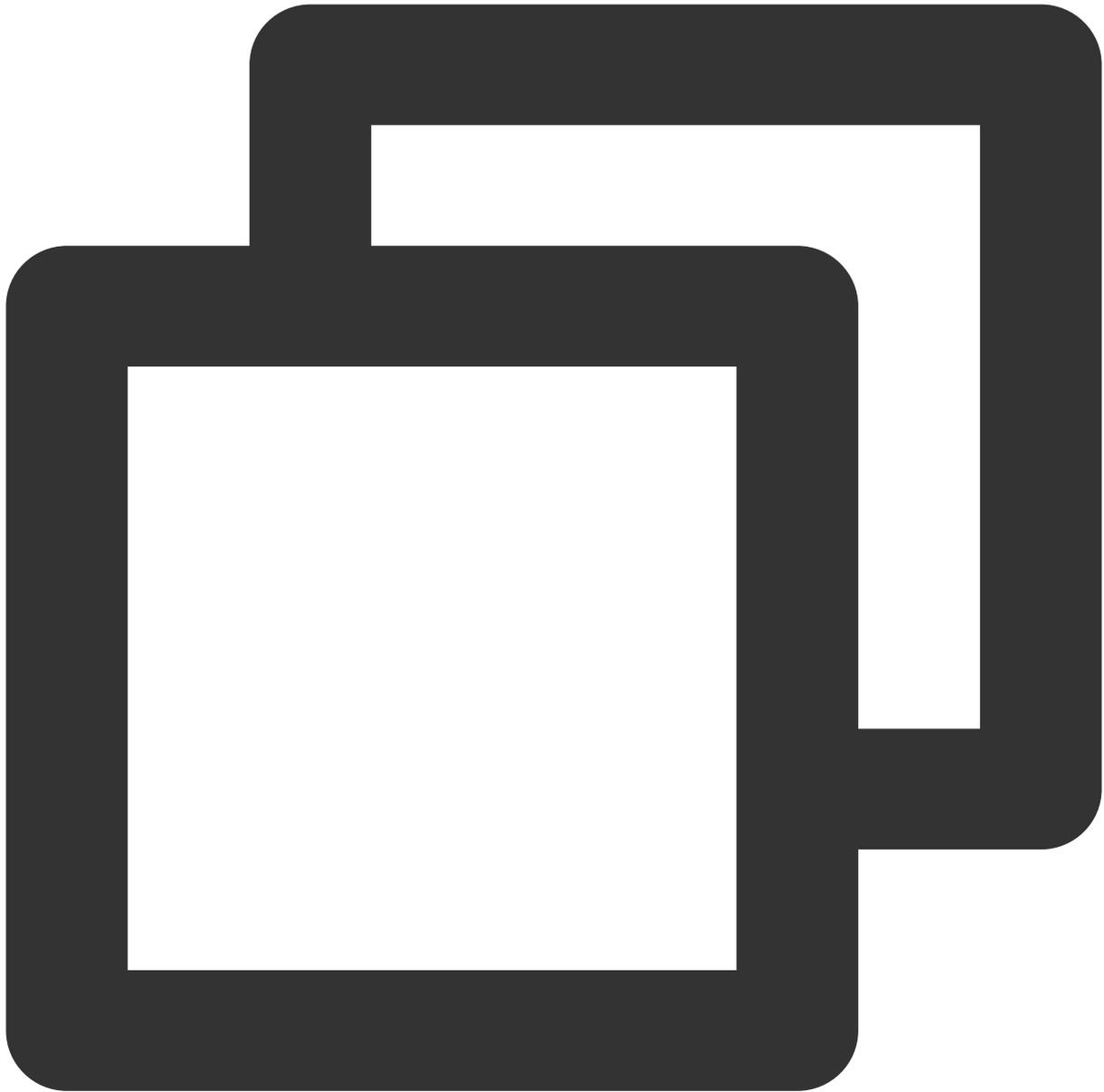
Basic Feature Usage

Note:

The following examples are based on Linux. Structures in the examples must be in standard JSON format.

Using TCCLI to create a CVM instance

Run the following command to create a CVM instance:



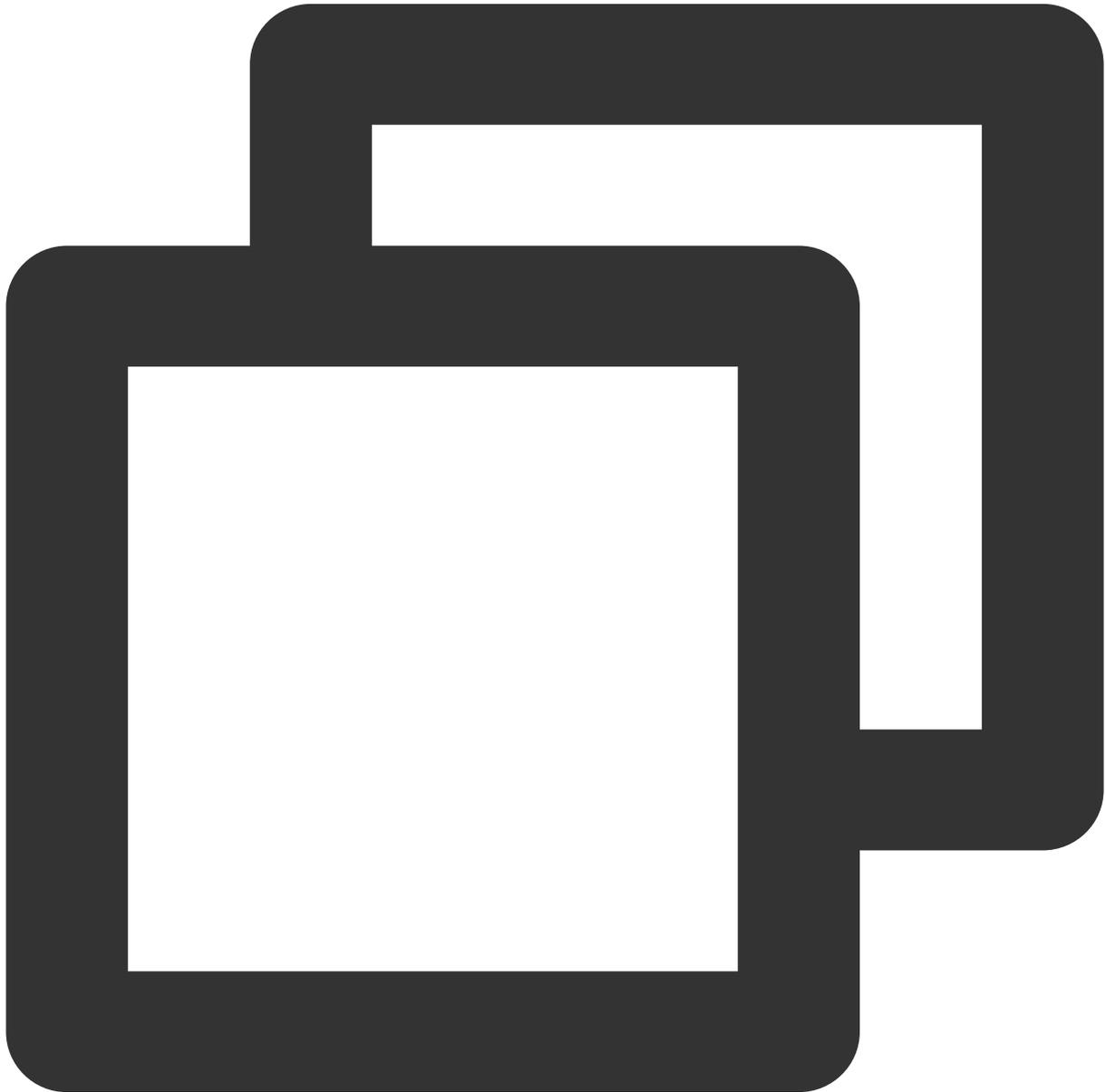
```
tccli cvm RunInstances
--InstanceChargeType POSTPAID_BY_HOUR
--InstanceChargePrepaid '{"Period":1,"RenewFlag":"DISABLE_NOTIFY_AND_MANUAL_RENEW"}'
--Placement '{"Zone":"ap-guangzhou-2"}'
--InstanceType S1.SMALL1
--ImageId img-8toqc6s3
--SystemDisk '{"DiskType":"CLOUD_BASIC", "DiskSize":50}'
--InternetAccessible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

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

Using standard input to transfer binary files

TCCLI allows you to call APIs with the content type octet-stream. When calling such APIs, you can use standard input

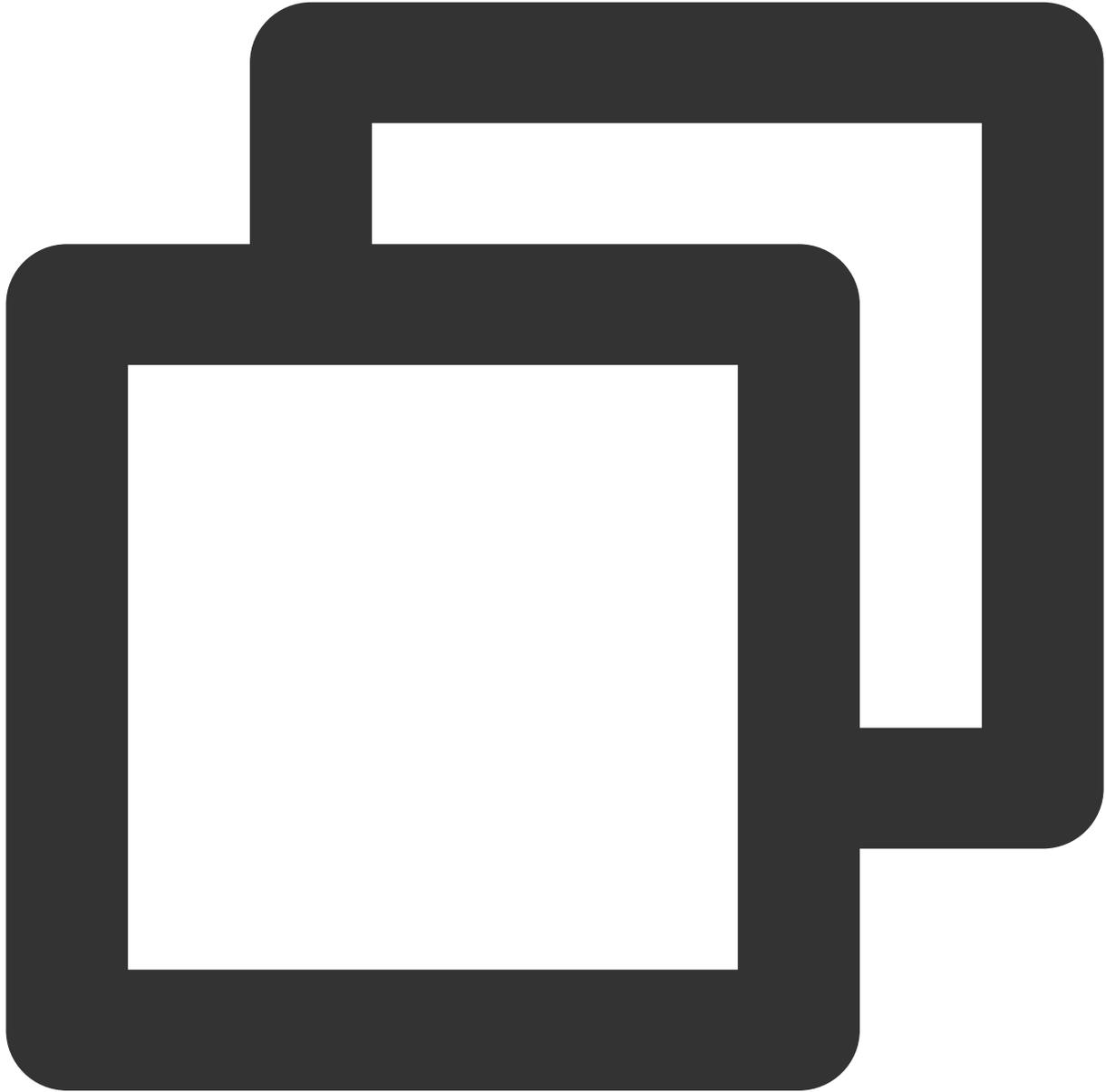
< /path/to/file to transfer binary files. Example:



```
# Taking the CLS UploadLog API as an example, you can use the following command to  
tccli cls UploadLog --TopicId xxx < /path/to/file
```

--cli-unfold-argument

If the API to call uses any structures, you can add `--cli-unfold-argument` in the command to use a dot (.) as the concatenation operator for structure input. In this case, you can use the autocomplete feature of TCCLI to make the input process easier. Example:



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

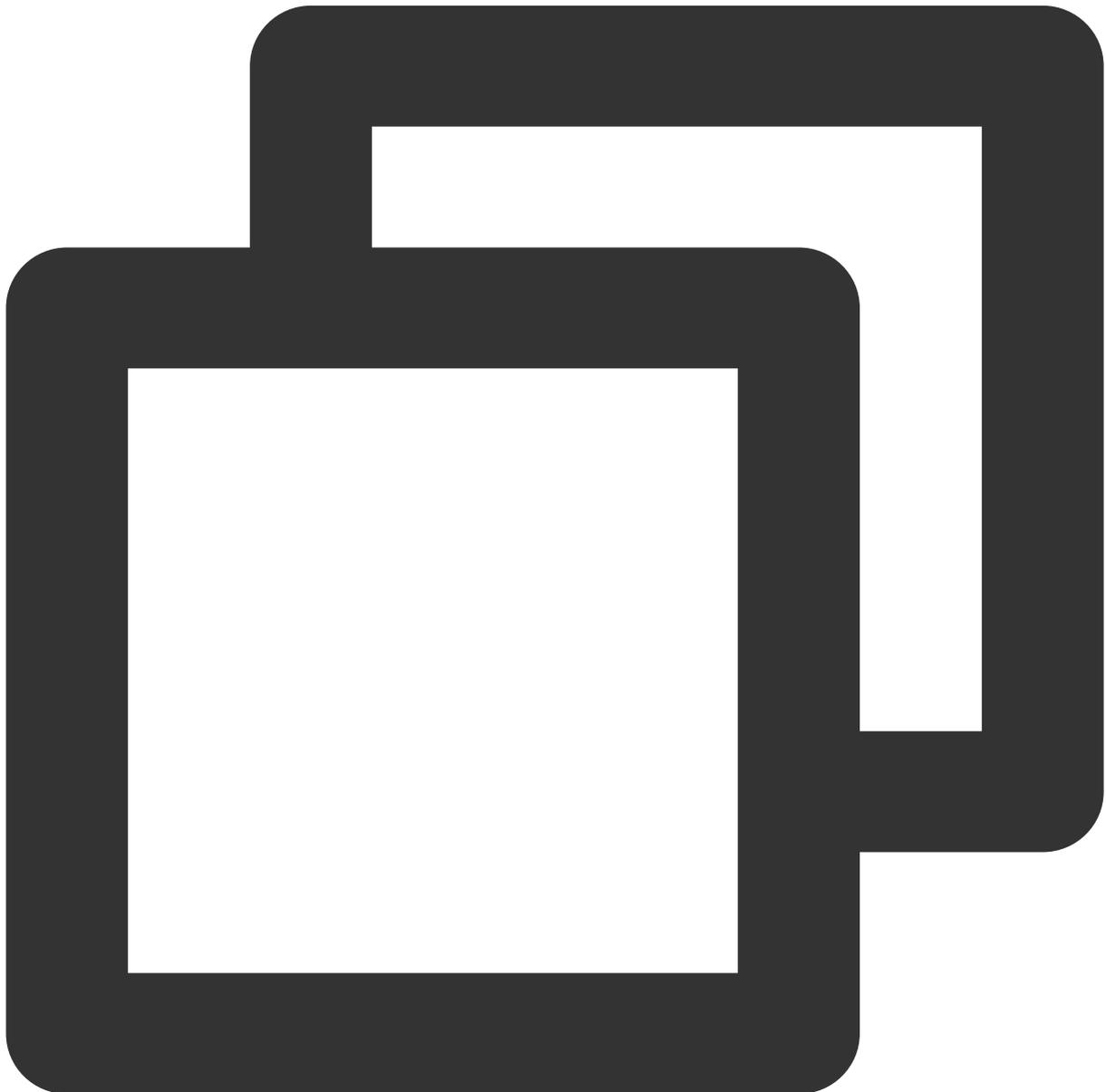
Note:

You can press the Tab key to complete `--cli-unfold-argument` . For details, see [Using the Autocomplete Feature](#).

You can specify `--cli-unfold-argument` in the command only when using TCCLI `3.0.273.1` or later.

--generate-cli-skeleton

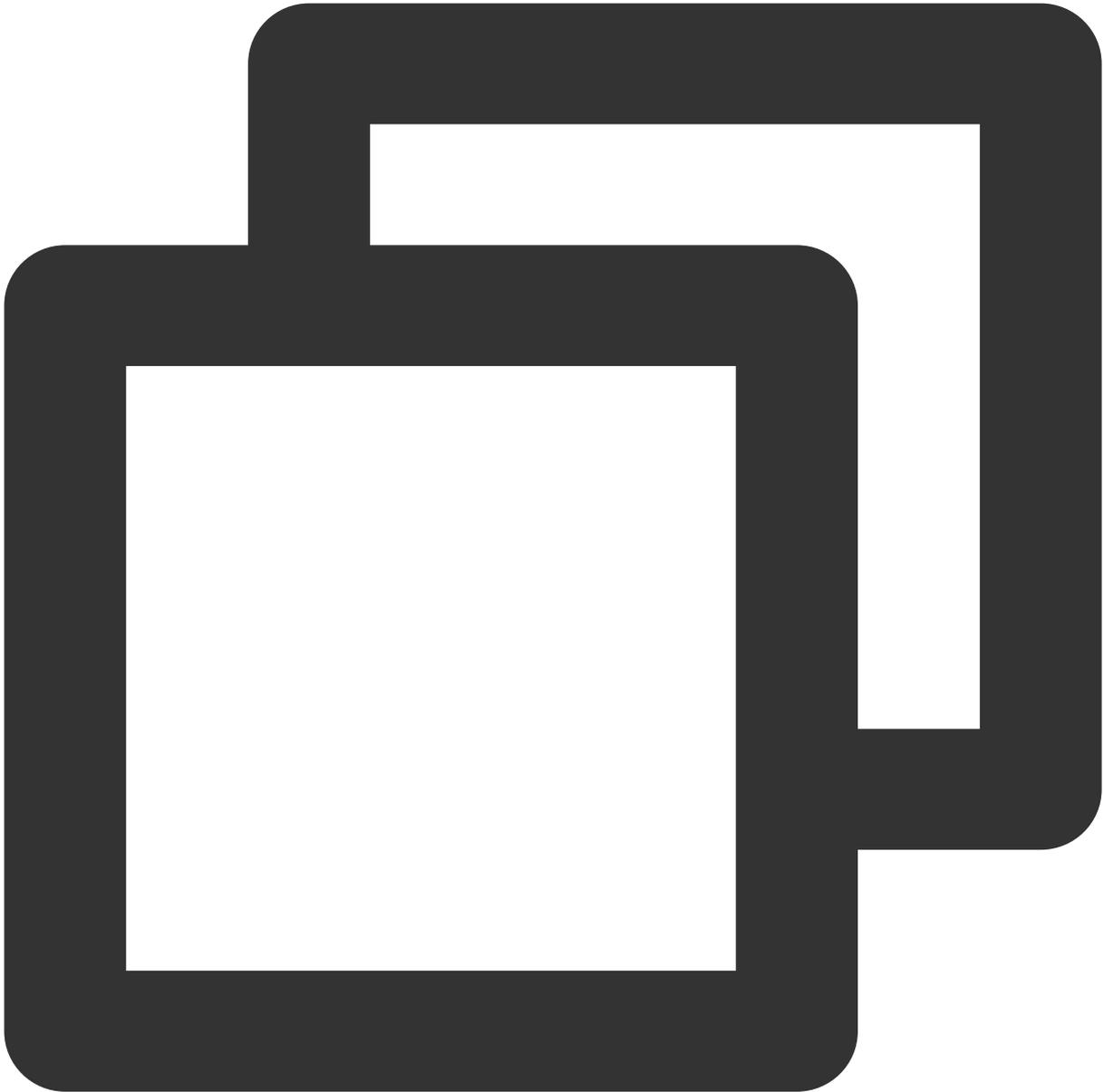
You can use `--generate-cli-skeleton` to generate a parameter skeleton in JSON format. Example:



```
# You can also input the generated JSON skeleton into a JSON file.  
# $ tccli cvm DescribeInstances --generate-cli-skeleton > /home/test.json
```

```
tccli cvm DescribeInstances --generate-cli-skeleton
```

The output is as follows:



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

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

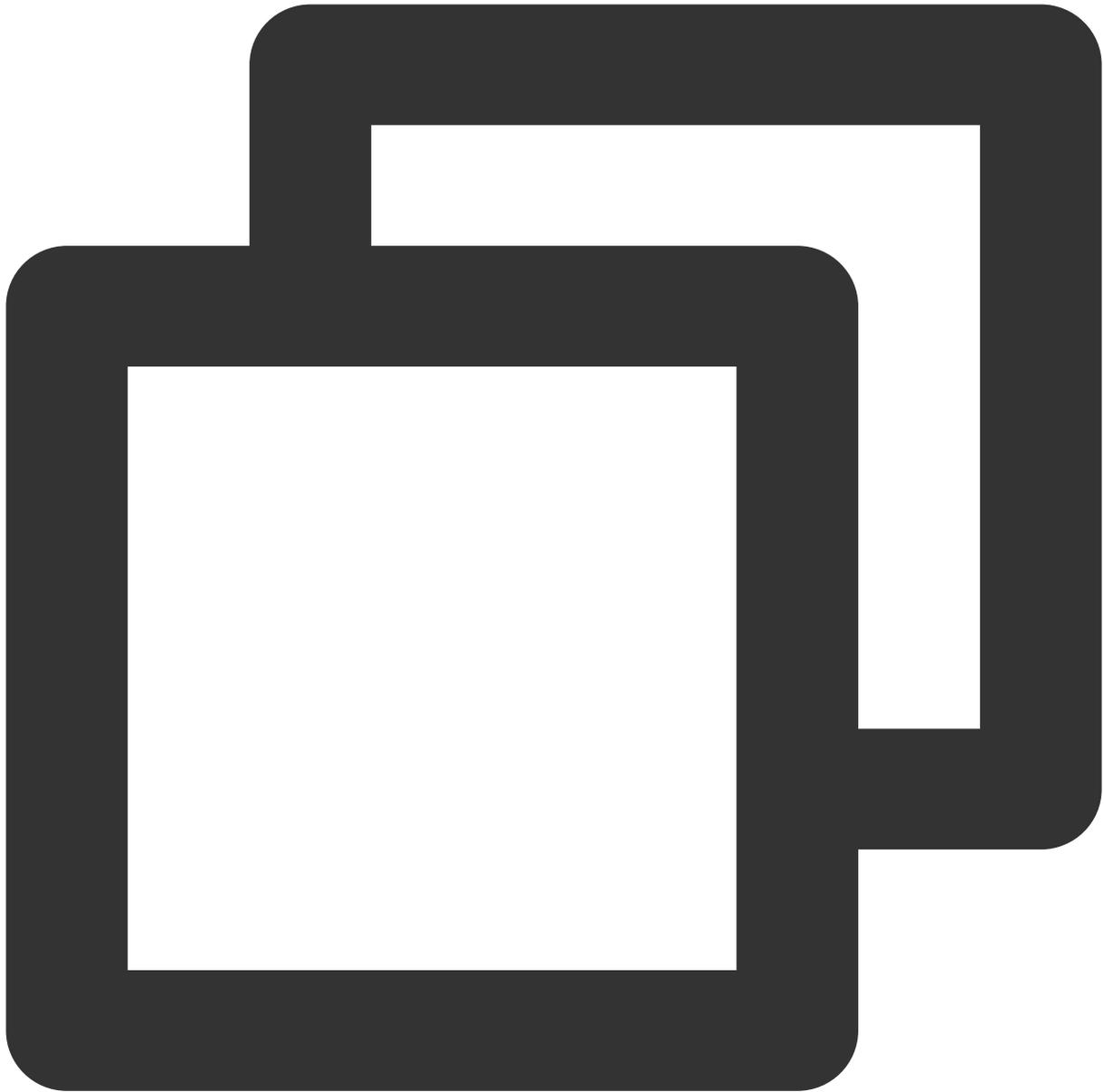
Note:

You can press the Tab key to complete `--generate-cli-skeleton` . For more information, see [Using the Autocomplete Feature](#).

You can specify `--generate-cli-skeleton` in the command only when using TCCLI `3.0.273.1` or later.

--cli-input-json

If there are too many input parameters, you can add `--cli-input-json` in the command to support input through a JSON file. (Add `file://file directory` after `--cli-input-json` .) You can use `--generate-cli-skeleton` to generate the corresponding JSON file. After specifying parameters in the JSON file, you can use the file to call an API. Example:



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

Note:

You can press the Tab key to complete `--cli-input-json` . For more information, see [Using the Autocomplete Feature](#).

You can specify `--cli-input-json` in the command only when using TCCLI `3.0.250.2` or later.

Obtaining Help Information

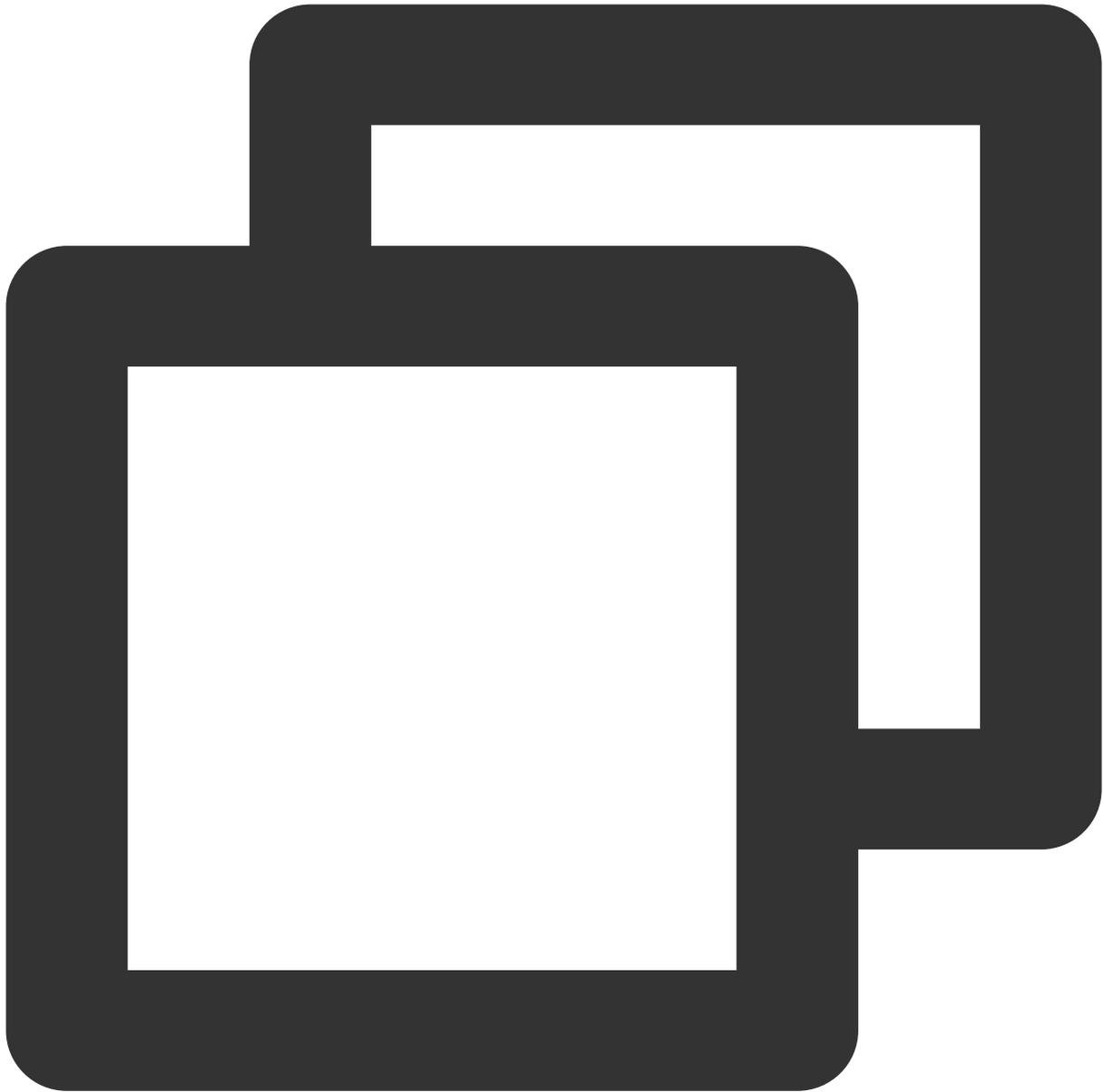
Last updated : 2023-02-21 10:41:36

This document describes how to use TCCLI to view the help information about products and their APIs.

Directions

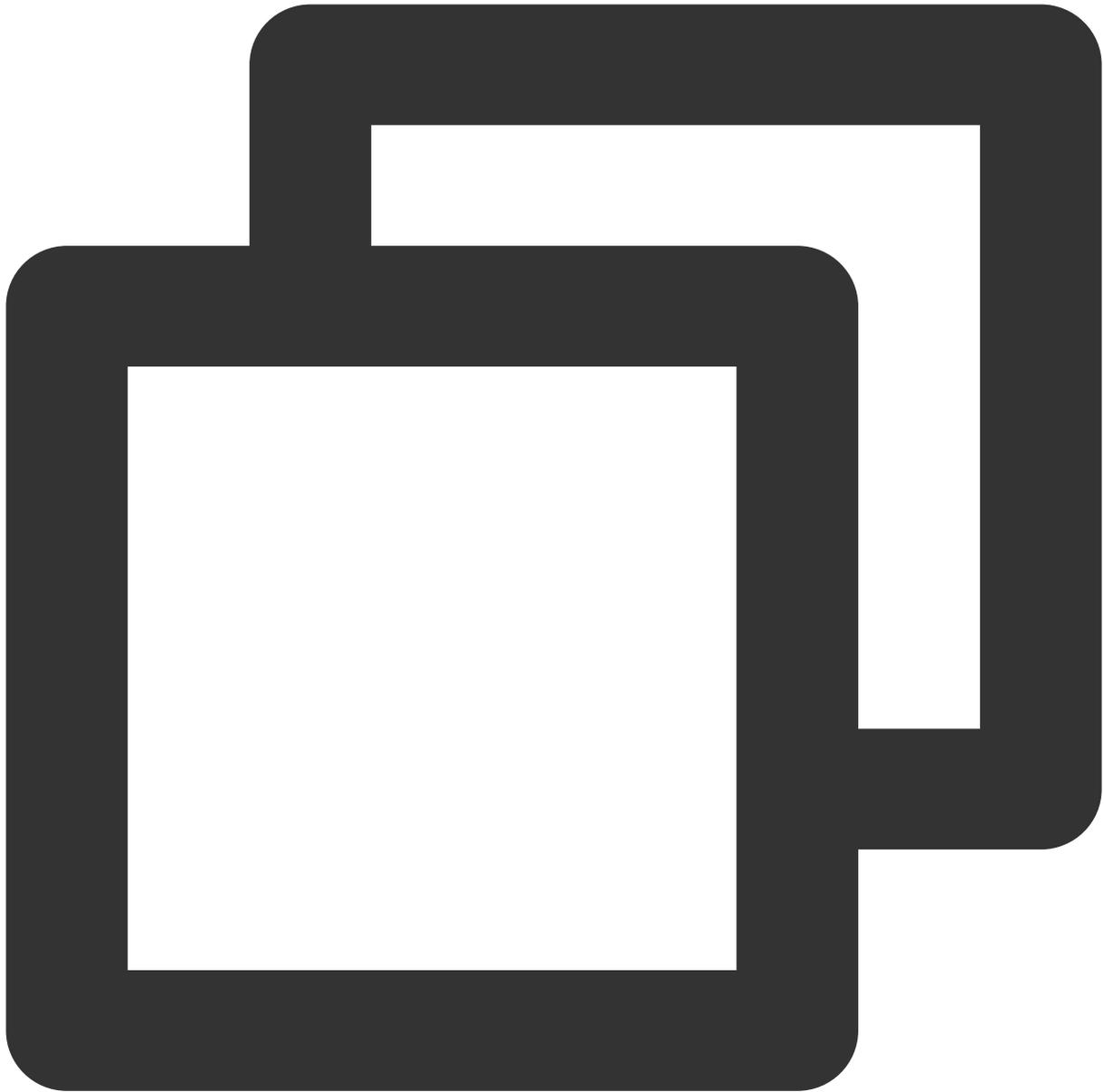
Viewing brief help information

Run the following command to view supported Tencent Cloud products. You can also view the products at [APIs](#).



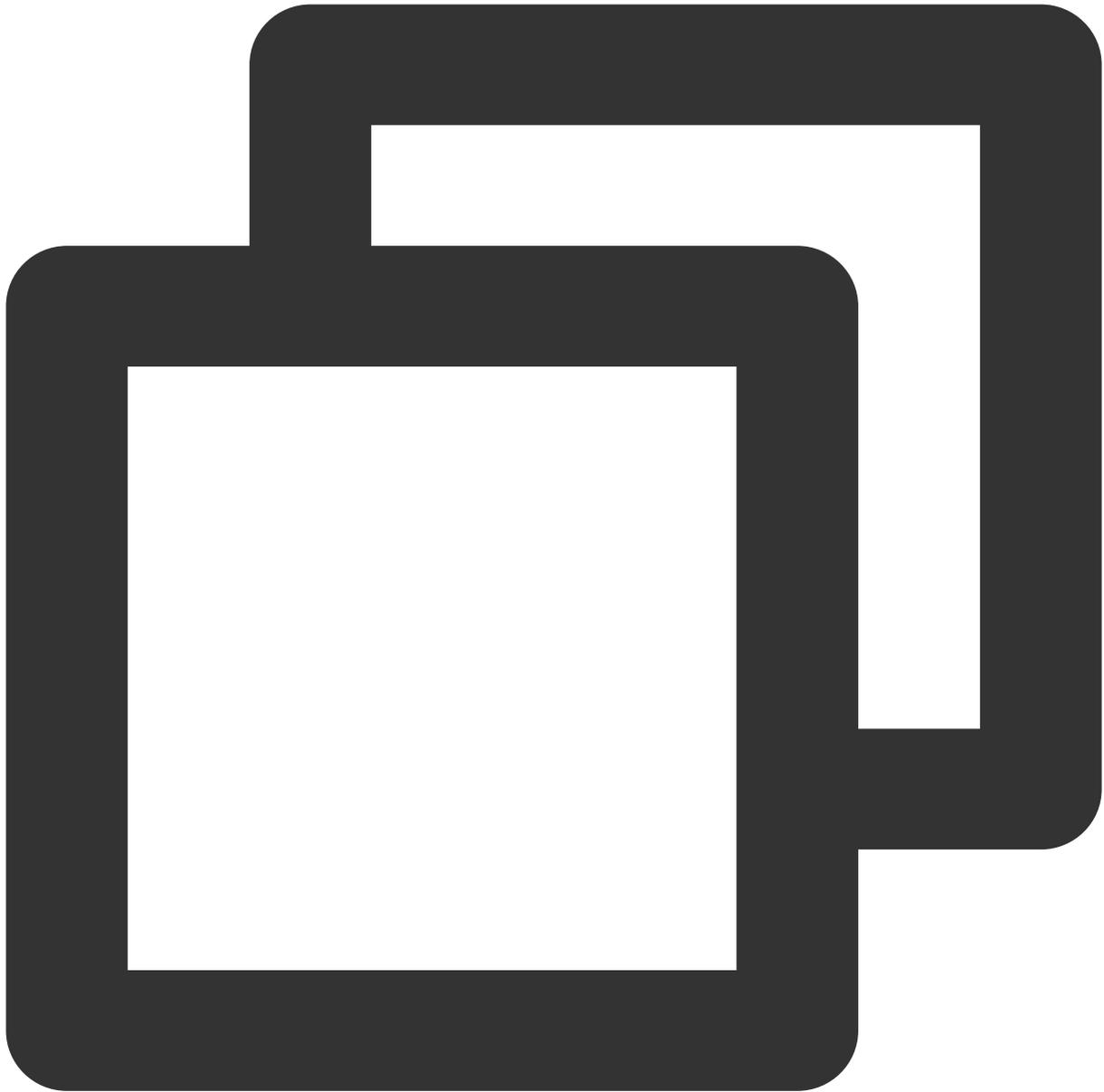
```
tccli help
```

Taking Cloud Virtual Machine (CVM) as an example, run the following command to view the CVM APIs that support TCCLI:



```
tccli cvm help
```

Taking the DescribeDisks API of Cloud Block Storage (CBS) as an example, run the following command to view the parameters of the API. For parameter descriptions and API information, see the corresponding API documentation at the Tencent Cloud website.

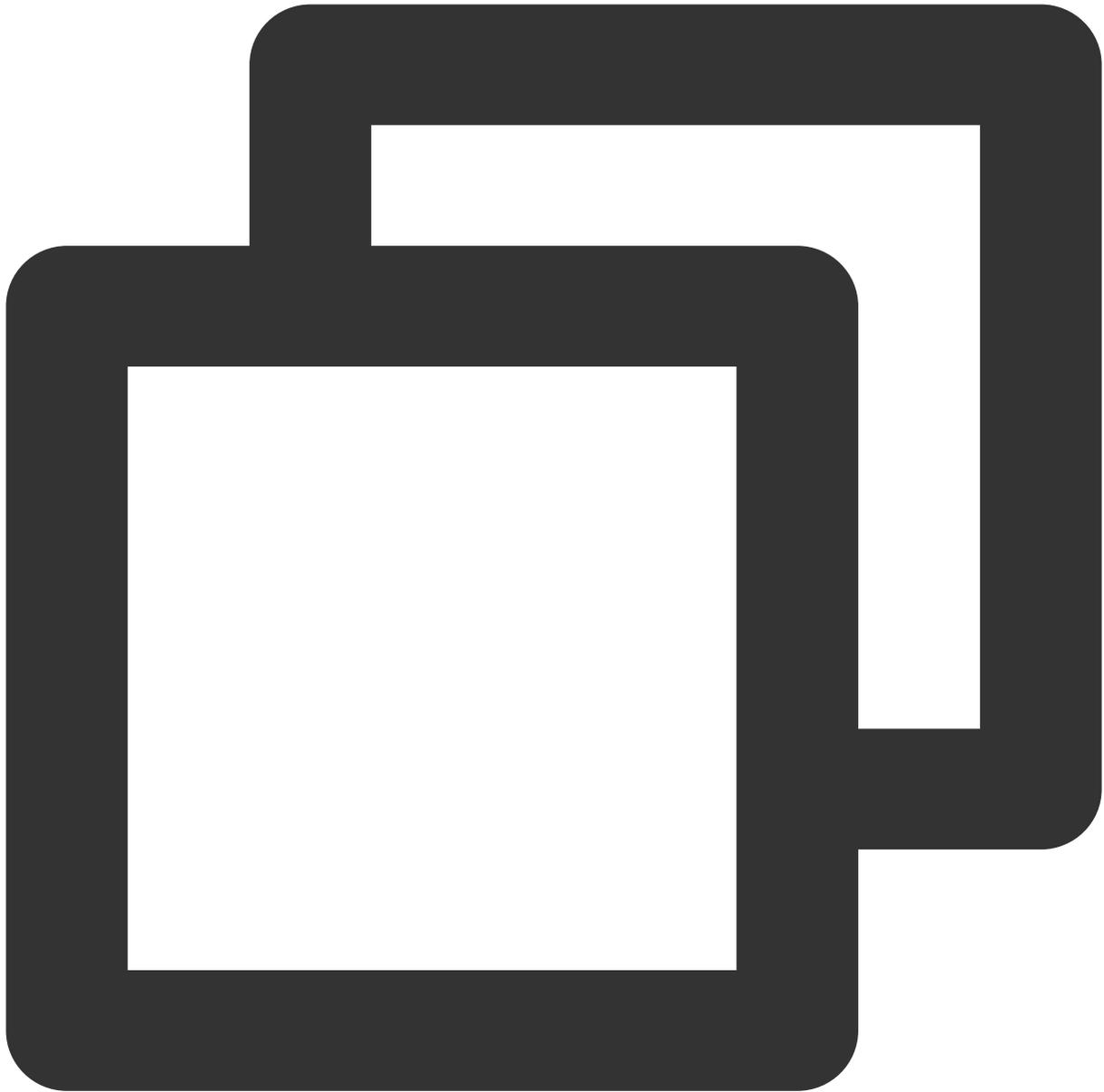


```
tccli cbs DescribeDisks help
```

Viewing detailed help information

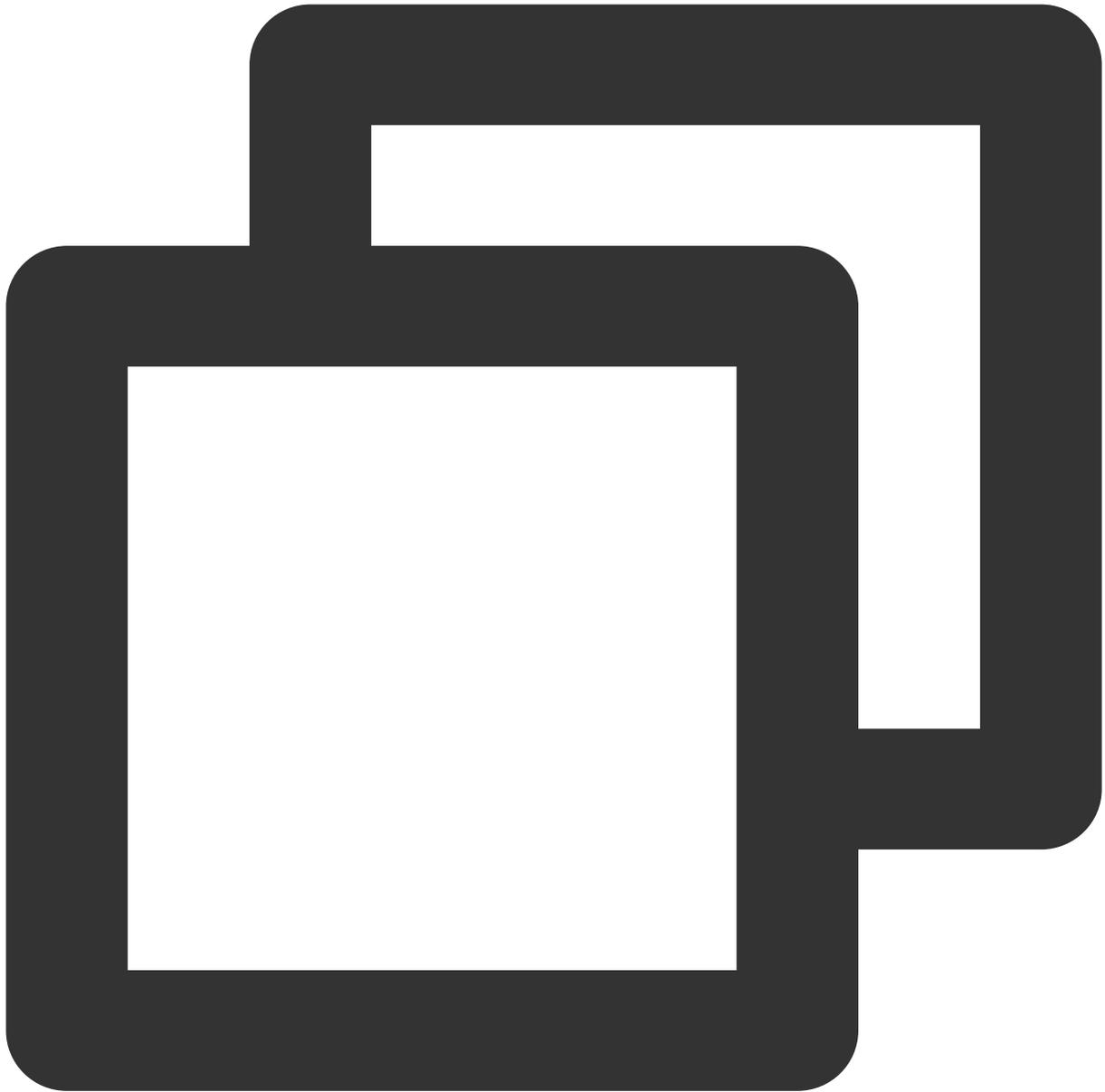
TCCLI displays brief help information by default. To view detailed information, use the `--detail` option.

Run the following command to view the detailed information about supported Tencent Cloud products:



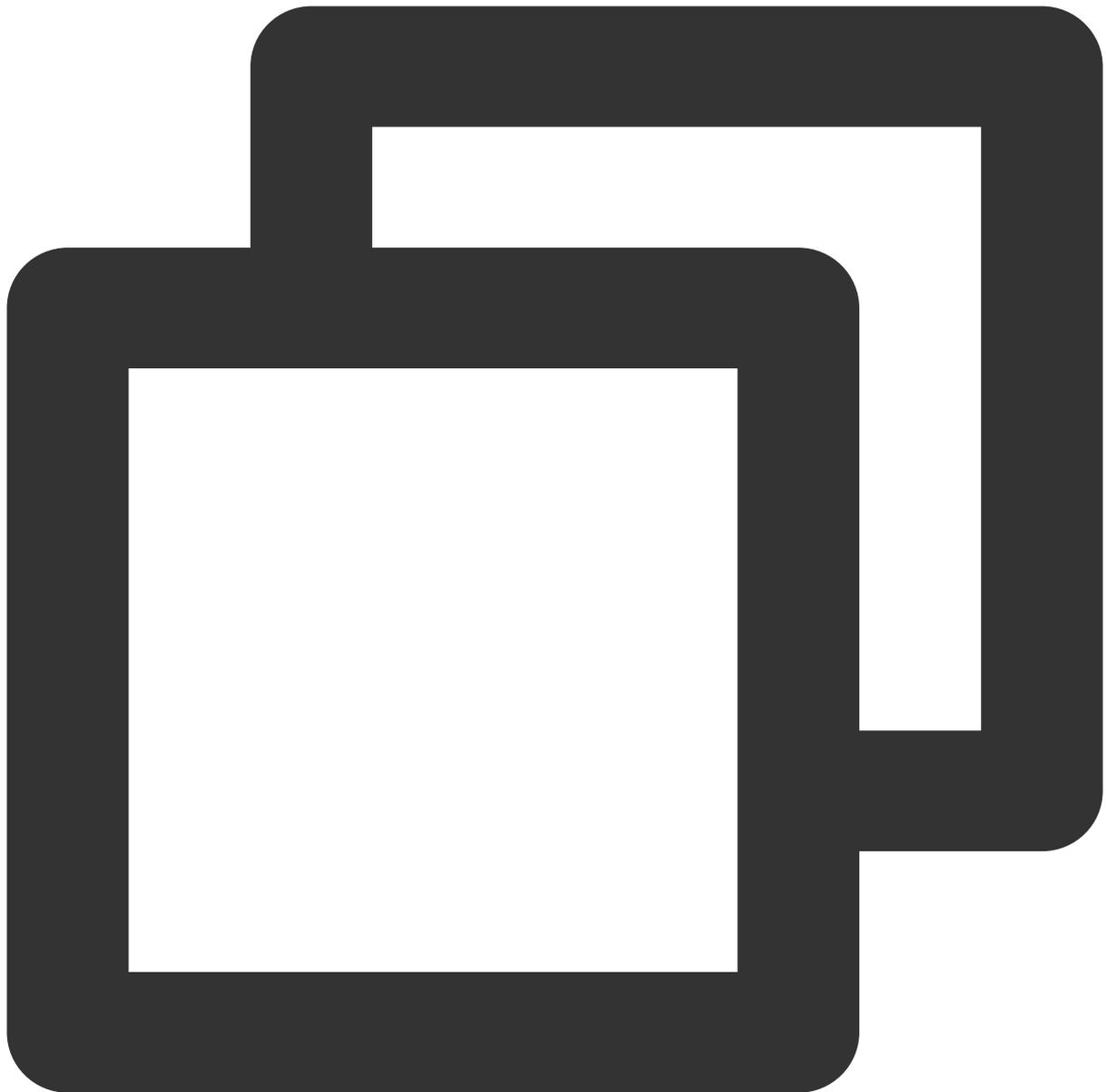
```
tccli help --detail
```

Taking CVM as an example, run the following command to view the detailed information about the CVM APIs that support TCCLI:



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

Taking the CBS DescribeDisks API as an example, run the following command to view the detailed information and use cases of the input and output parameters of the API:



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

Multi-version API Access

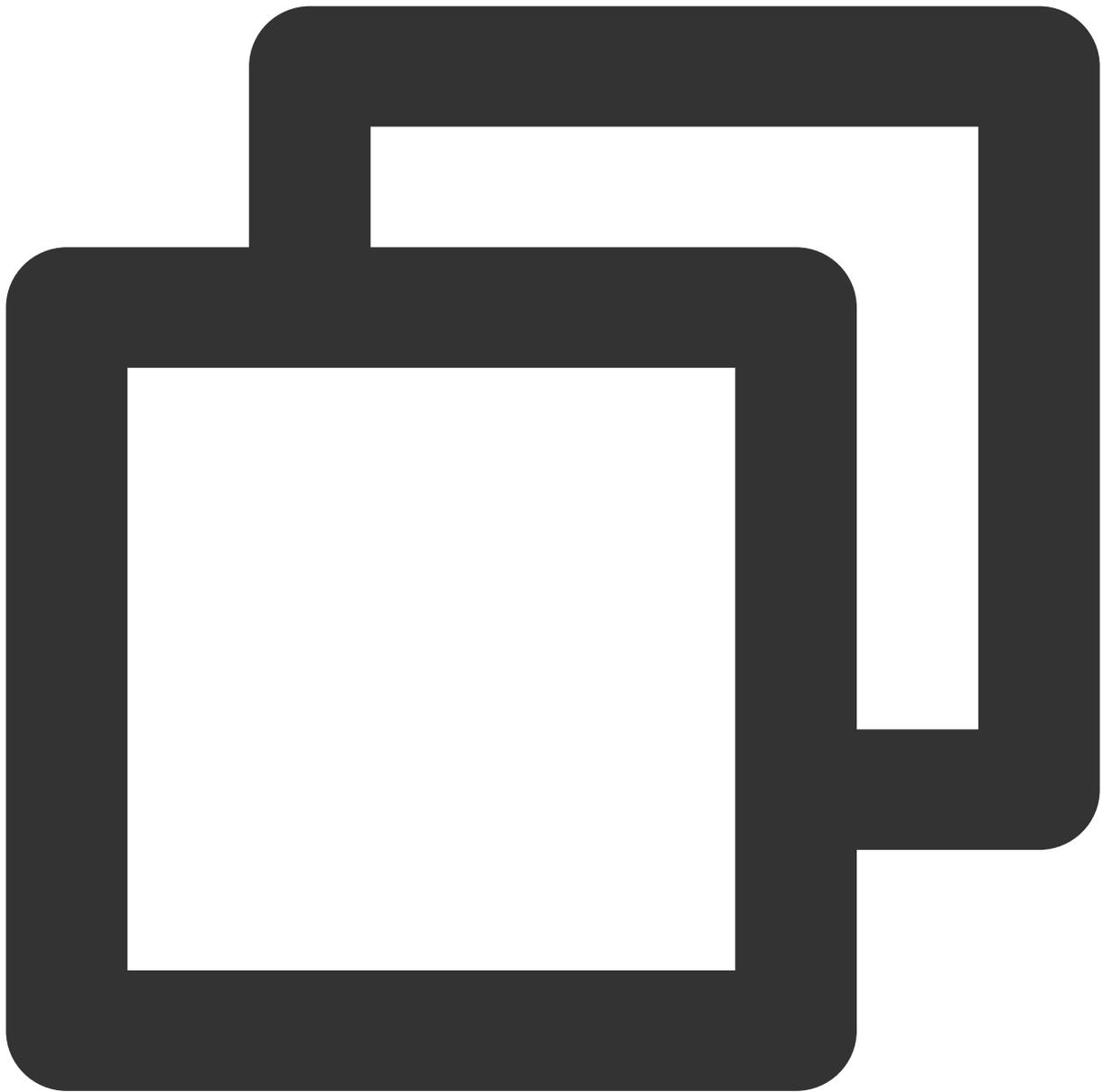
Last updated : 2023-02-21 10:41:36

By default, TCCLI accesses the latest version of APIs. However, some products have multiple versions of APIs. To access an API of a specific earlier version, refer to this document.

Directions

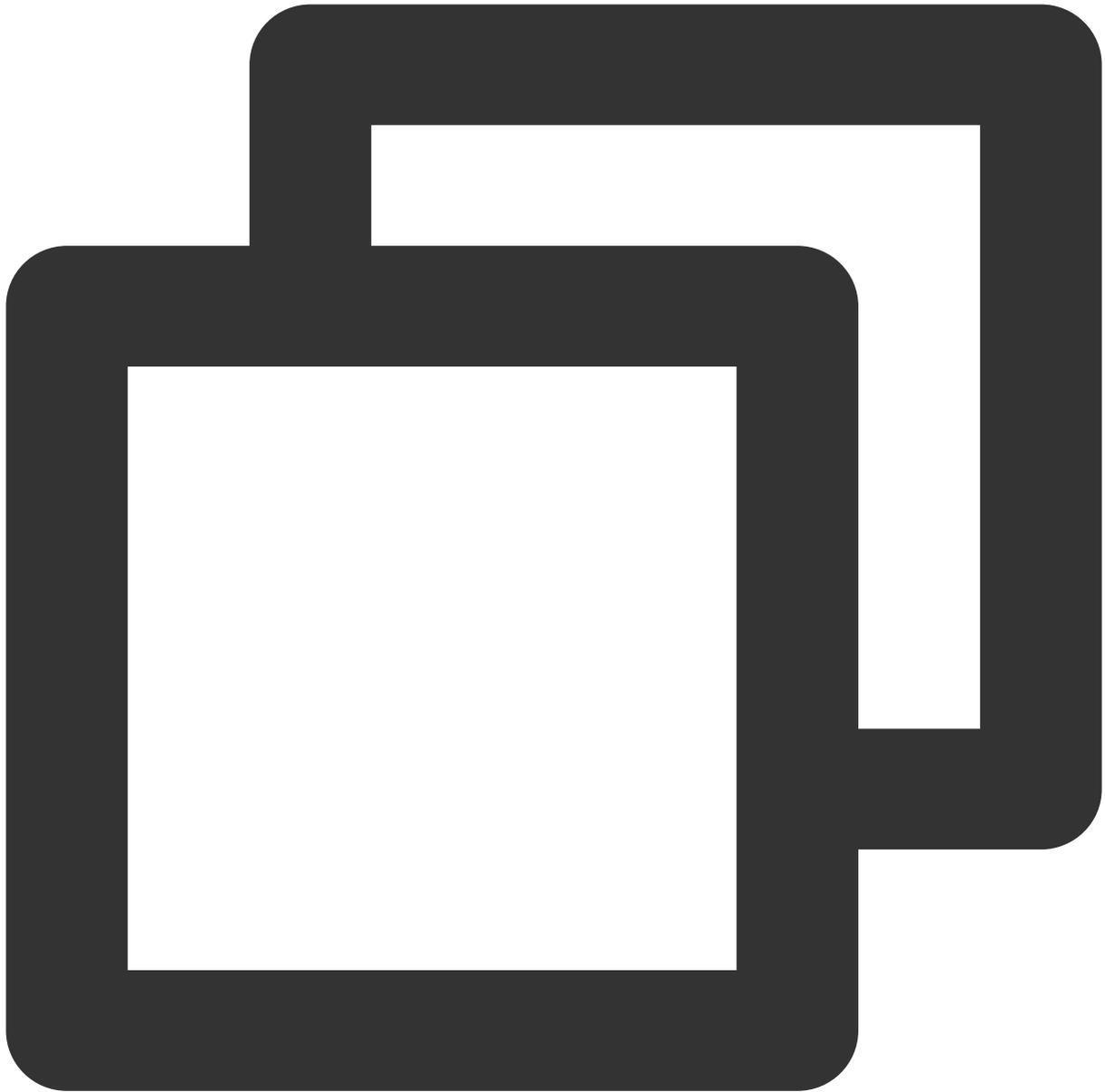
Taking Cloud Virtual Machine (CVM) as an example, run the following command to set the default API version to

```
2017-03-12 :
```



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

Run the following commands to specify the API version in real time:



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

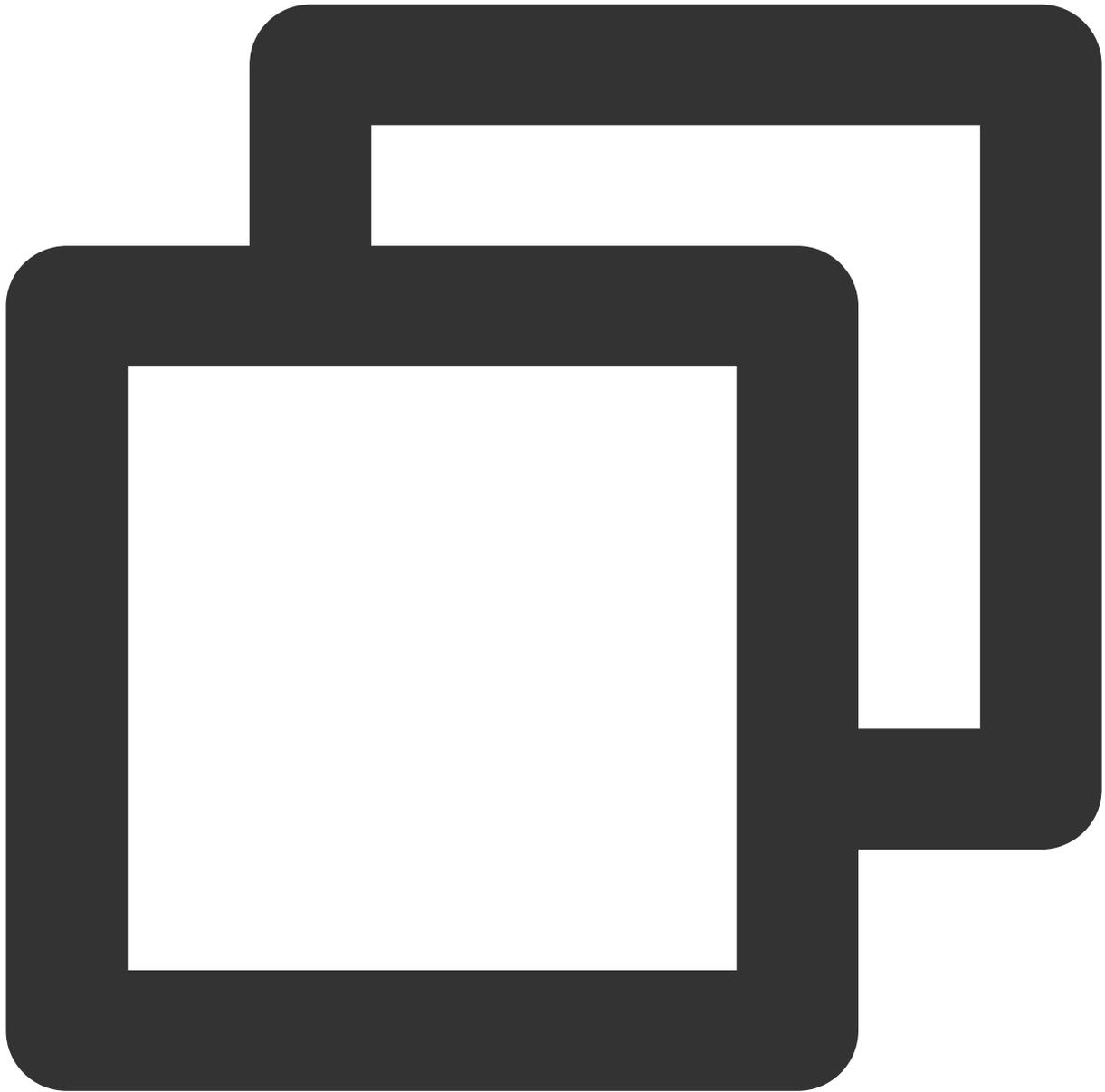
Specifying the Nearest Endpoint

Last updated : 2023-02-21 10:41:36

By default, Tencent Cloud Command Line Interface (TCCLI) sends a request to the nearest endpoint to access a service. You can also specify an endpoint for a product.

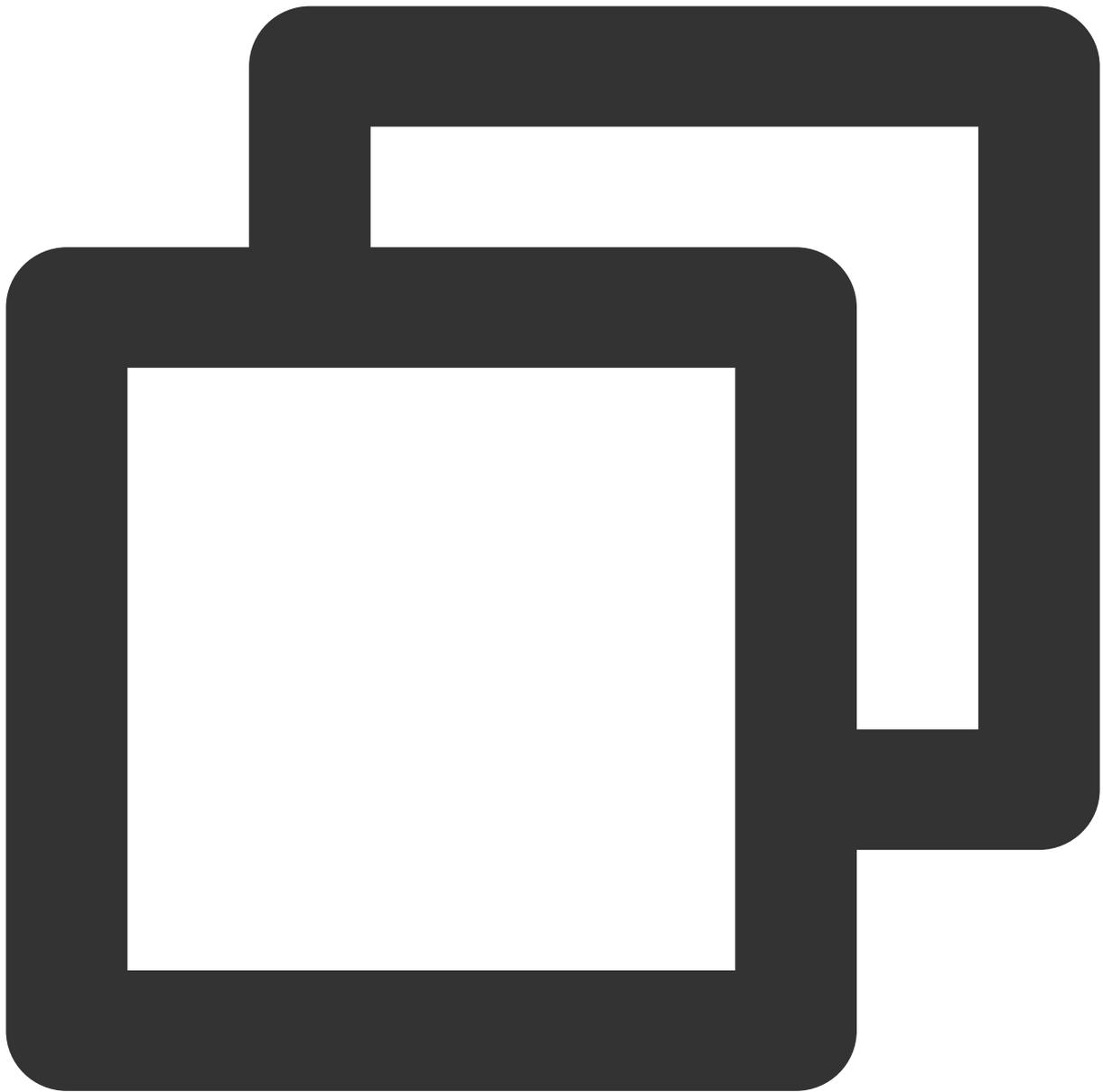
Directions

Taking Cloud Virtual Machine (CVM) as an example, run the following command to set ap-guangzhou as the default endpoint:



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

Run the following command to set ap-guangzhou as the endpoint in real time when calling the API:



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

Filtering Return Results

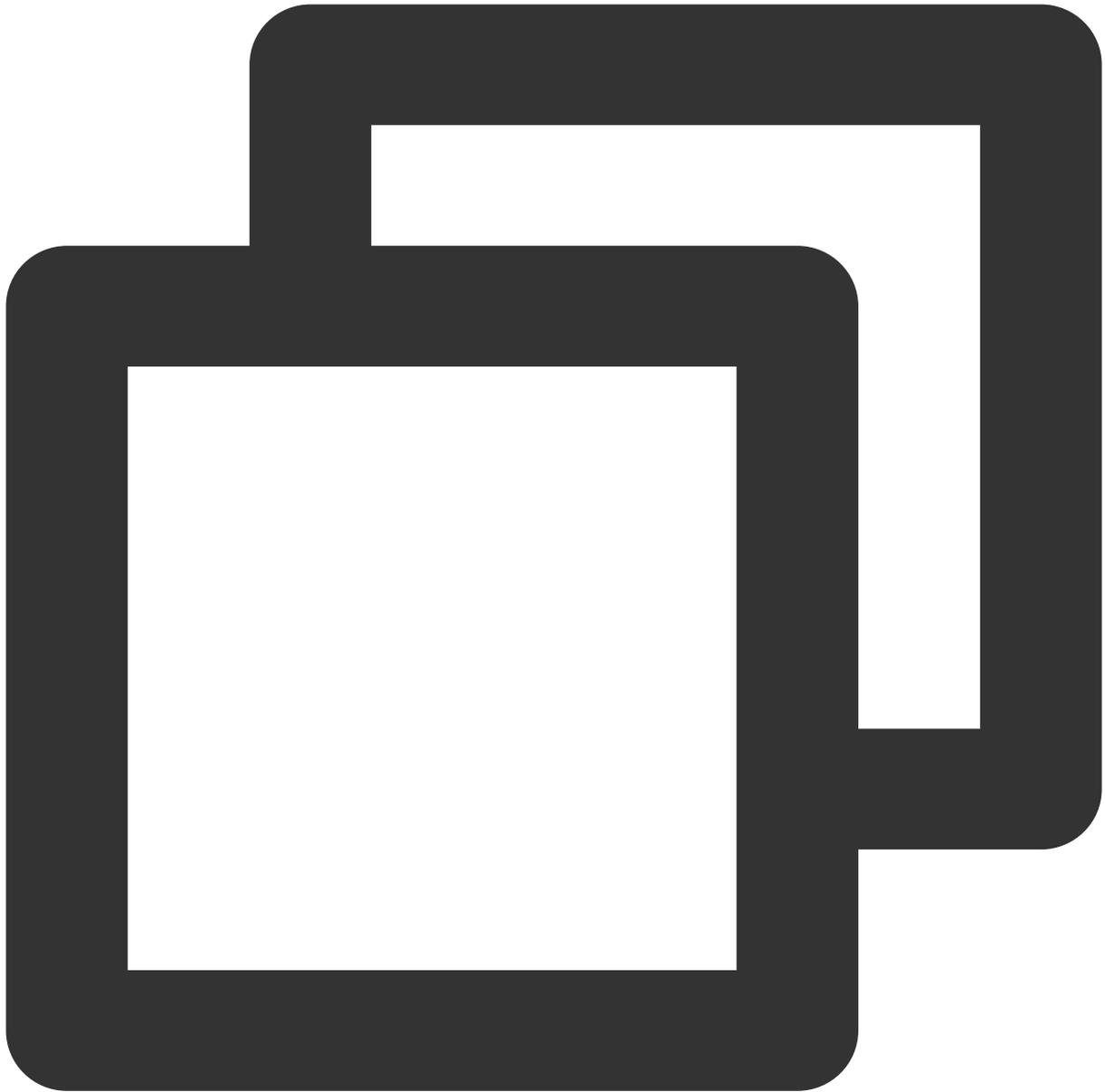
Last updated : 2023-02-21 10:41:36

Some commands return much information in the output. You can filter the information as required.

Directions

This document describes how different filtering methods are used and their corresponding outcomes. The DescribeZones API of Cloud Virtual Machine (CVM) is used as an example in this document.

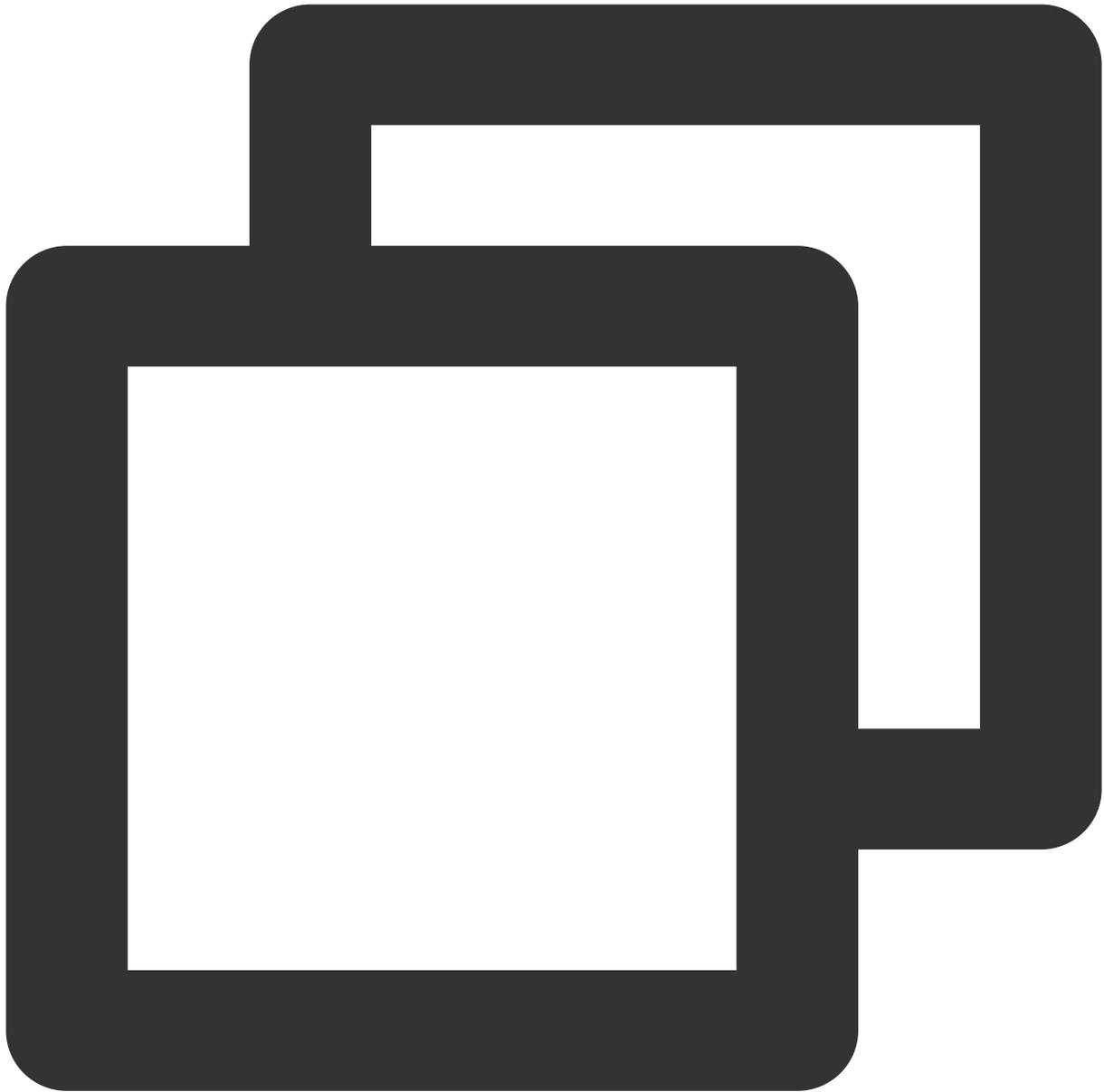
[View the unfiltered output.](#)



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

```
    "ZoneState": "AVAILABLE",
    "ZoneId": "100002",
    "Zone": "ap-guangzhou-2",
    "ZoneName": "Guangzhou Zone 2"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100003",
    "Zone": "ap-guangzhou-3",
    "ZoneName": "Guangzhou Zone 3"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100004",
    "Zone": "ap-guangzhou-4",
    "ZoneName": "Guangzhou Zone 4"
  }
],
"RequestId": "4fd313a6-155f-4c7a-bf86-898c02fcae02"
}
```

View a specified field.



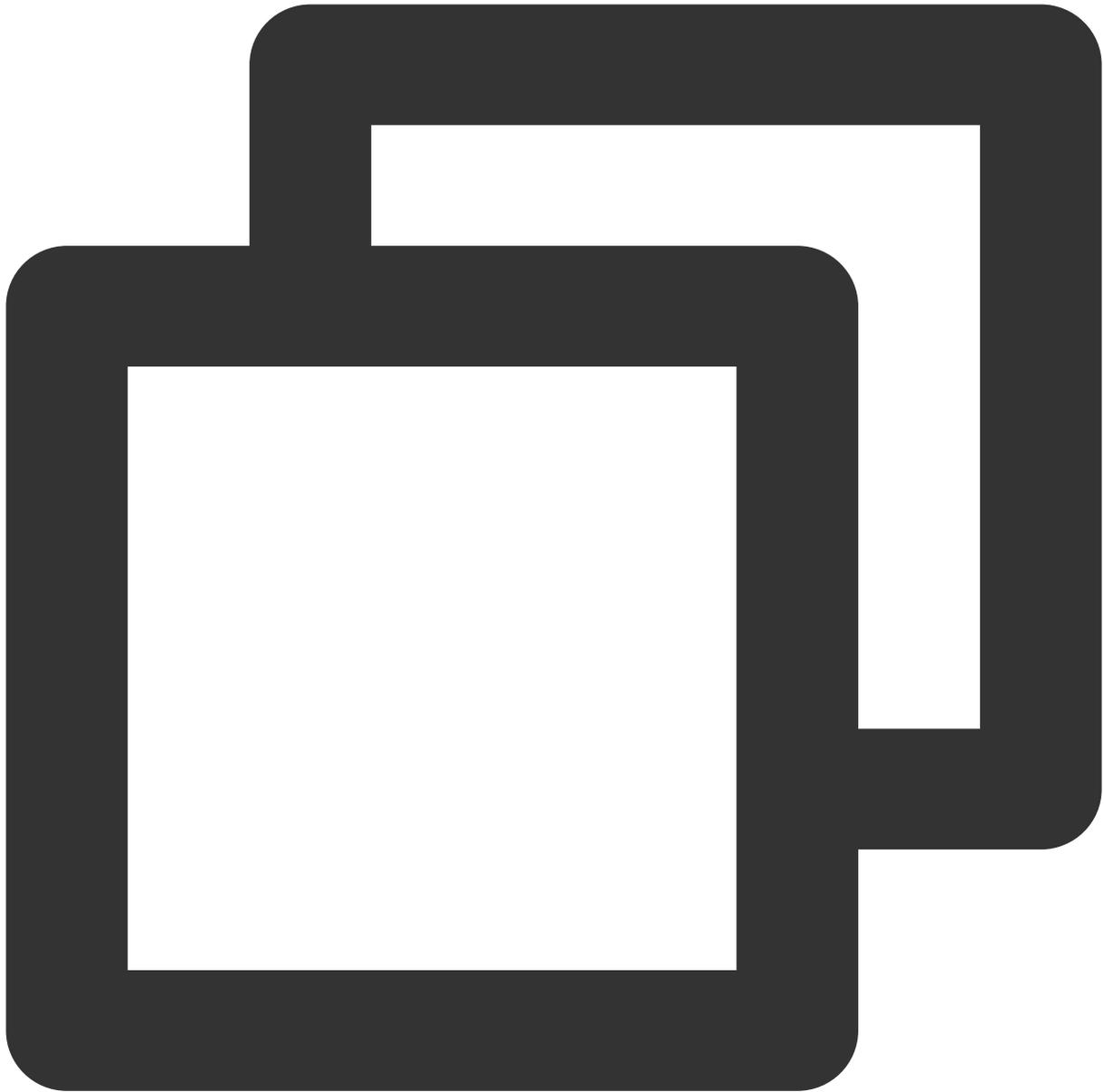
```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter TotalCount  
4
```

View the Nth sub-object of a specified object in array type.

Note:

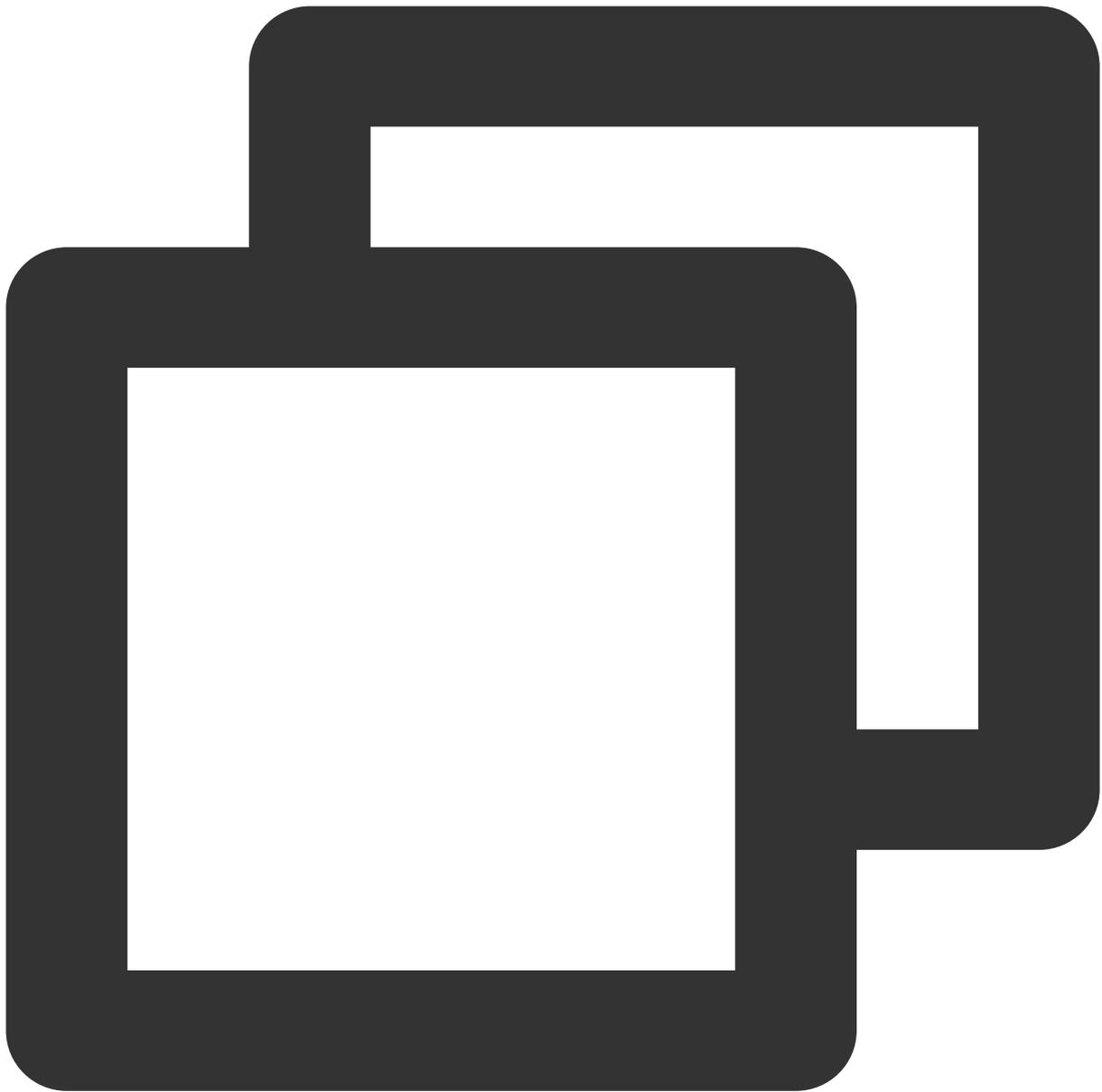
On macOS, the error `zsh: no matches found: xxx` may be returned after you run the following command.

To resolve this issue, enclose the filter in single quotes.



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

View a certain field of all the sub-objects with a certain name under the specified object in array type.

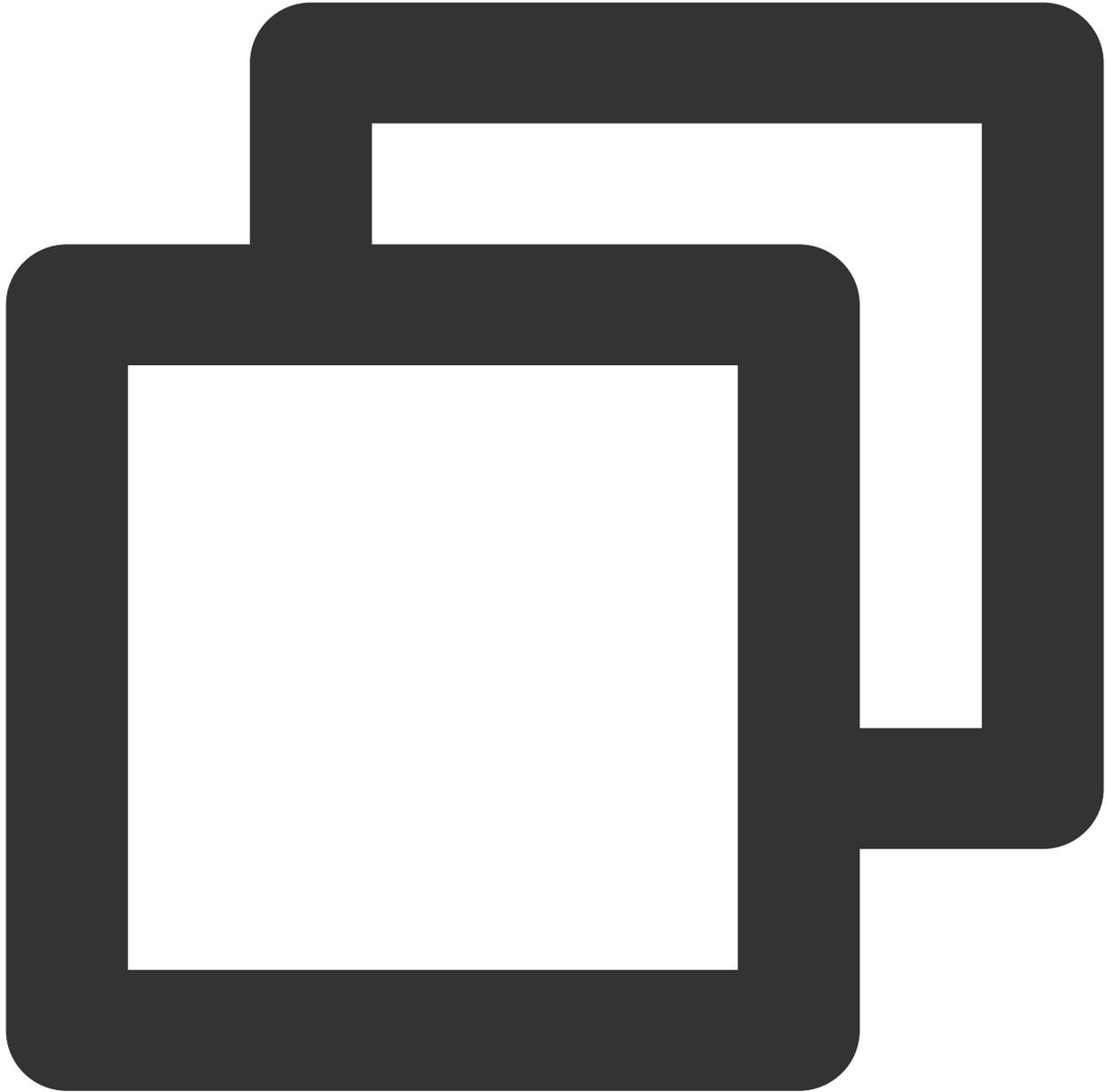


```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[*].ZoneName
[
  "Guangzhou Zone 1",
  "Guangzhou Zone 2",
  "Guangzhou Zone 3",
  "Guangzhou Zone 4"
]
```

Filter the sub-objects in an array and display them with a new name.

Note:

The filter must be enclosed in single quotes.



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter 'ZoneSet[*].{name:ZoneSet[*].Name}'
[
  {
    "name": "Guangzhou Zone 1",
    "id": "100001"
  },
  {
    "name": "Guangzhou Zone 2",
    "id": "100002"
  }
]
```

```
    },  
    {  
      "name": "Guangzhou Zone 3",  
      "id": "100003"  
    },  
    {  
      "name": "Guangzhou Zone 4",  
      "id": "100004"  
    }  
  ]
```

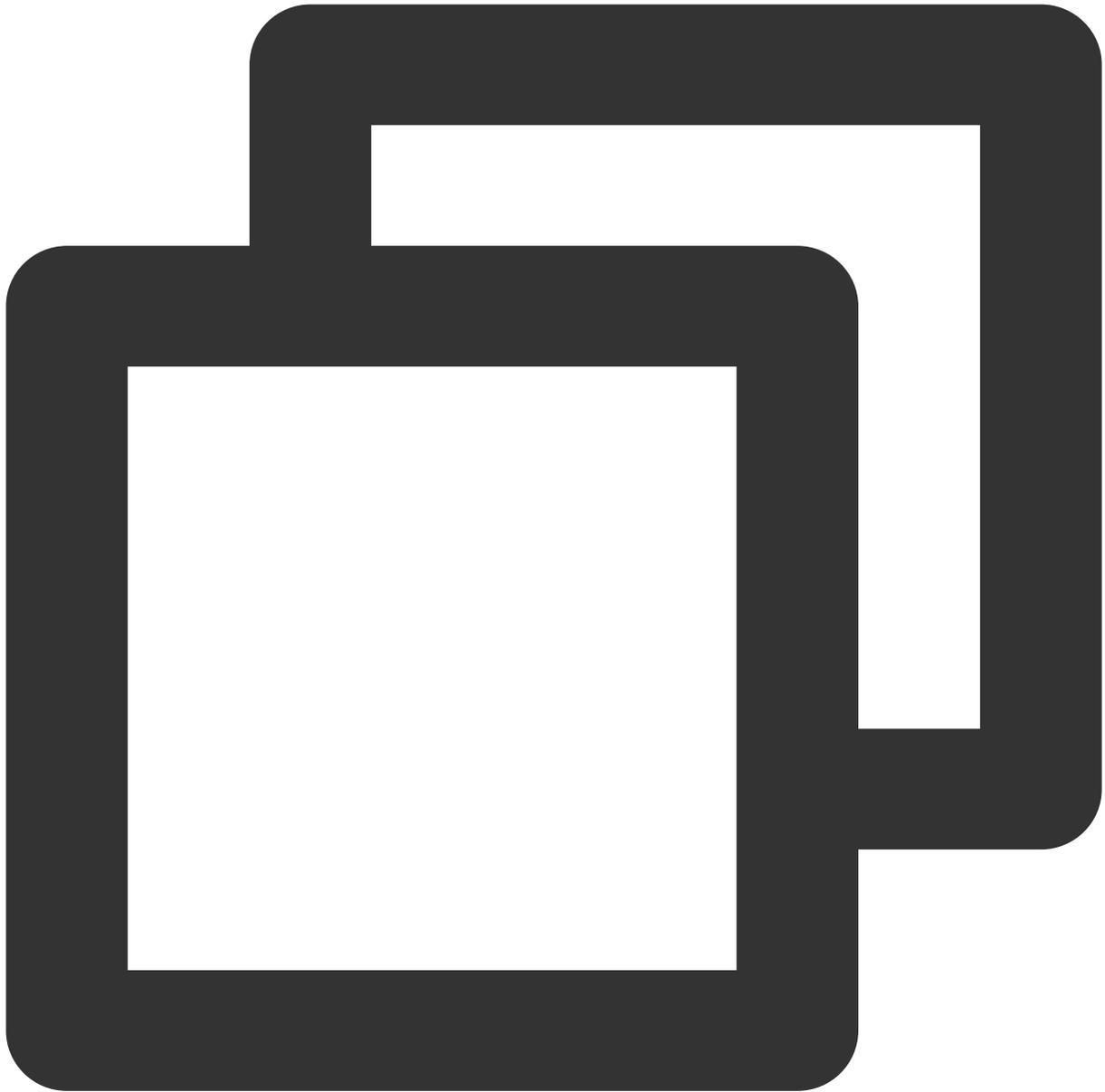
Result Polling

Last updated : 2023-02-21 10:41:36

During product use, some operations may not be instantly completed. The result polling feature allows you to periodically check if an operation is completed. For example, after an instance is activated, it may not immediately enter the `RUNNING` state. In this case, you can use polling to check its status until it enters the `RUNNING` state.

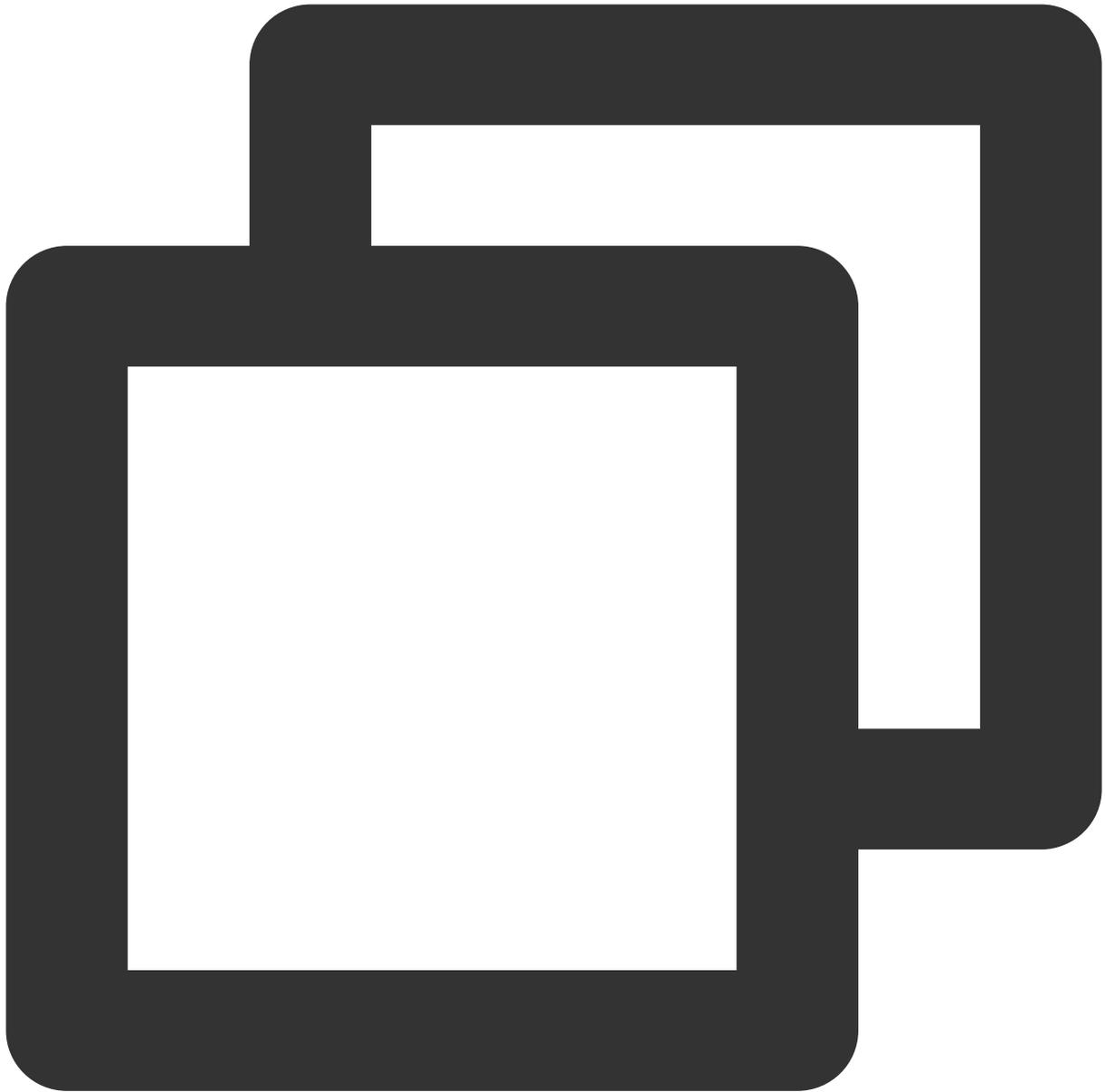
Directions

Run the following command to initiate a polling program that checks the instance status at a specific interval. The program runs until the instance enters the `RUNNING` state or until the polling times out.



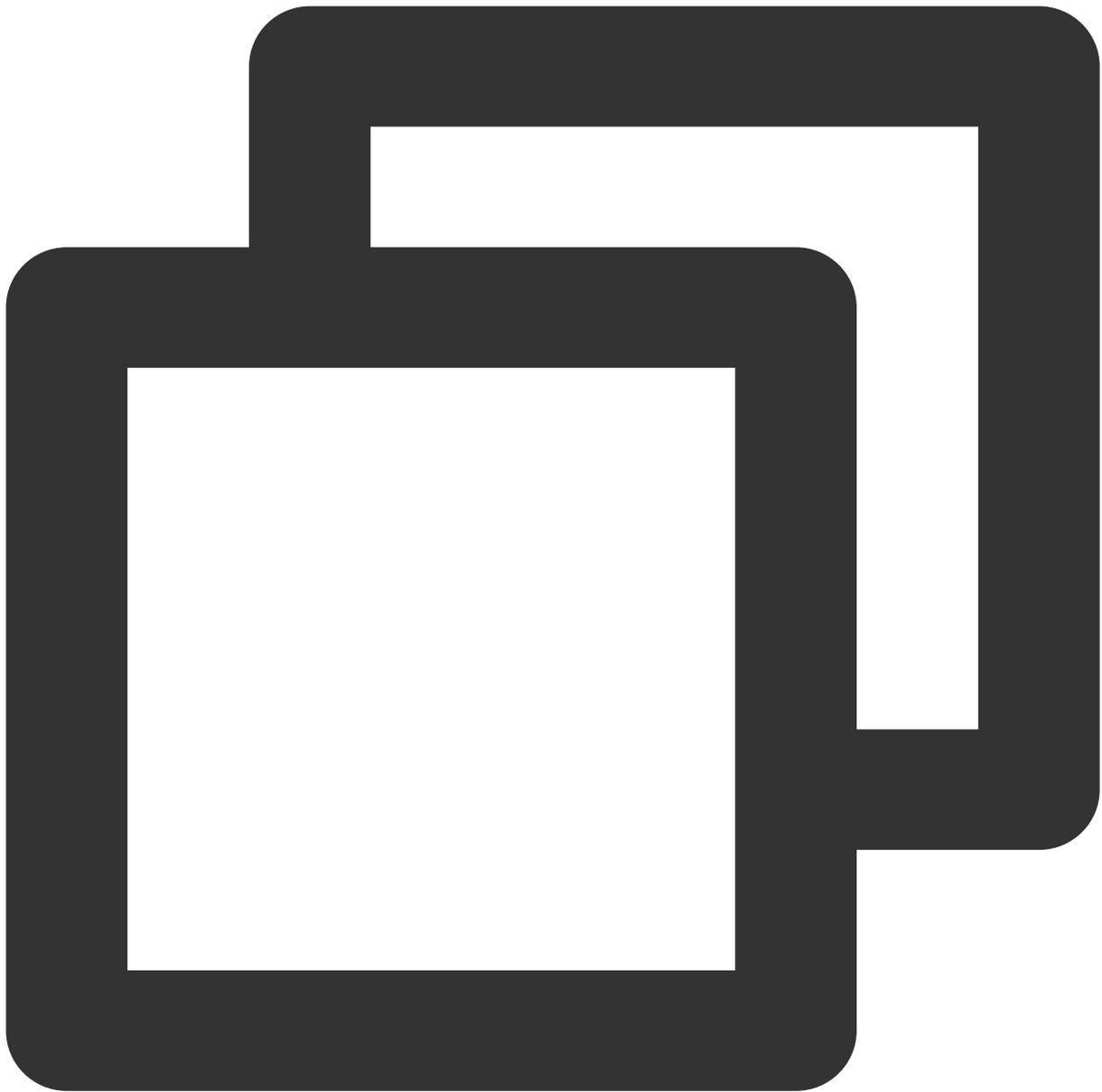
```
tccli cvm DescribeInstancesStatus --region ap-hongkong --waiter '{"expr':'Instances'
```

You can specify a timeout period and an interval for polling. Run the following command to set the timeout period to 180s and the interval to 5s.



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

You can set the values of optional subparameters in the configuration file. Add the following configuration to the `default.configure` file to specify a timeout period of 180s and an interval of 5s.



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

Parameter description

--region: Set this parameter to the region where your instance is located.

--waiter: Parameters after `waiter` must be in JSON format and enclosed in double quotation marks. The table below lists the required and optional parameters:

Parameter	Required	Description
<code>expr</code>	Yes	The field that you want to query. Use JMESPath to query the value of the specified field.
<code>to</code>	Yes	The target value of the field that you want to query.
<code>timeout</code>	No	The timeout period for polling in seconds.
<code>interval</code>	No	The polling interval in seconds.

Using Advanced Features

Last updated : 2020-03-04 15:22:38

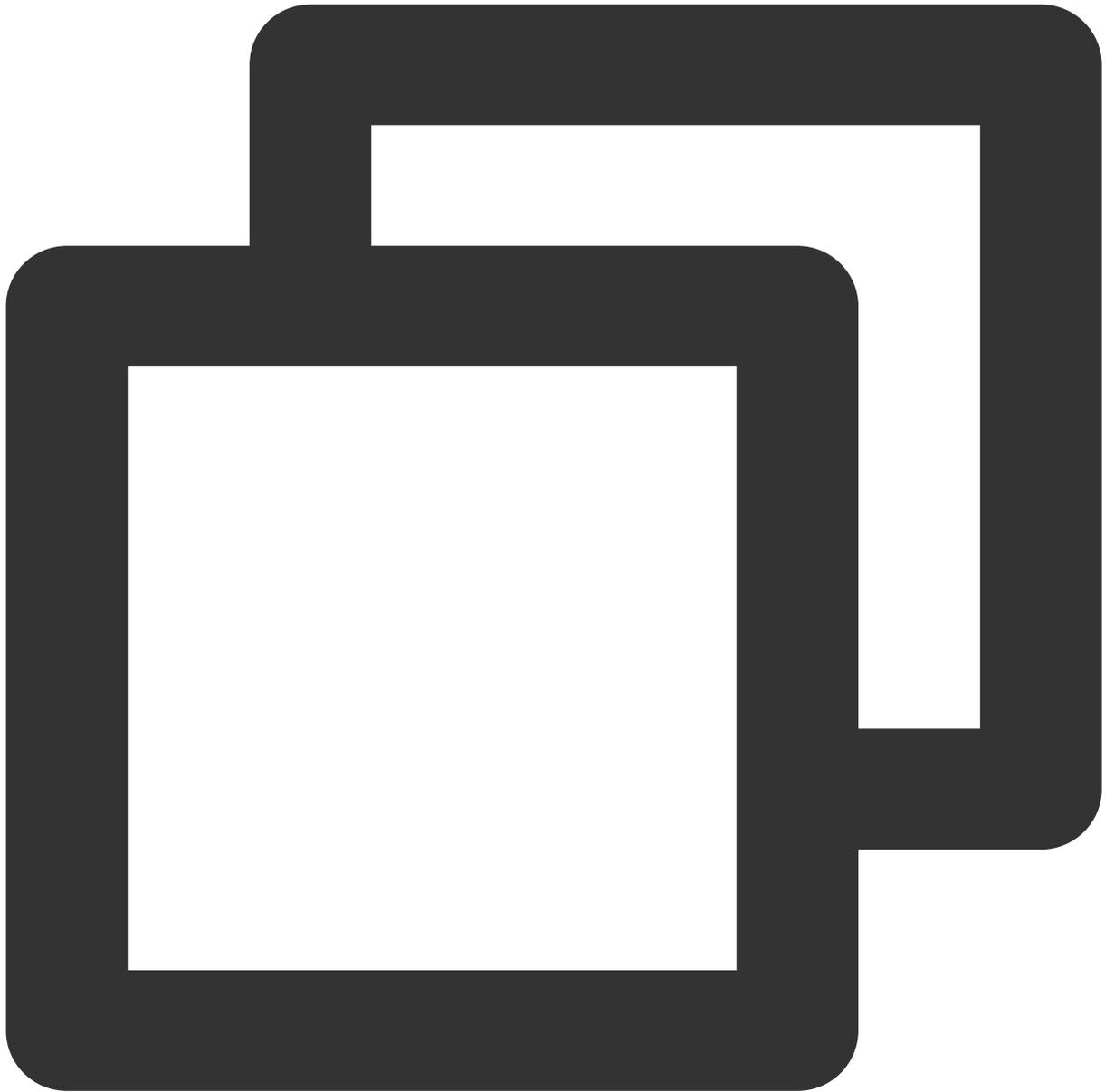
Operation Scenarios

This document describes how to use the advanced features of TCCLI International Version, including multi-version API access, nearest access point specifying, and return result filtering.

Directions

Multi-version API access

Some products may have multiple versions of APIs, and TCCLI accesses the latest version by default. If you want to access a specific legacy version, you can do so in the following way (with CVM as an example).

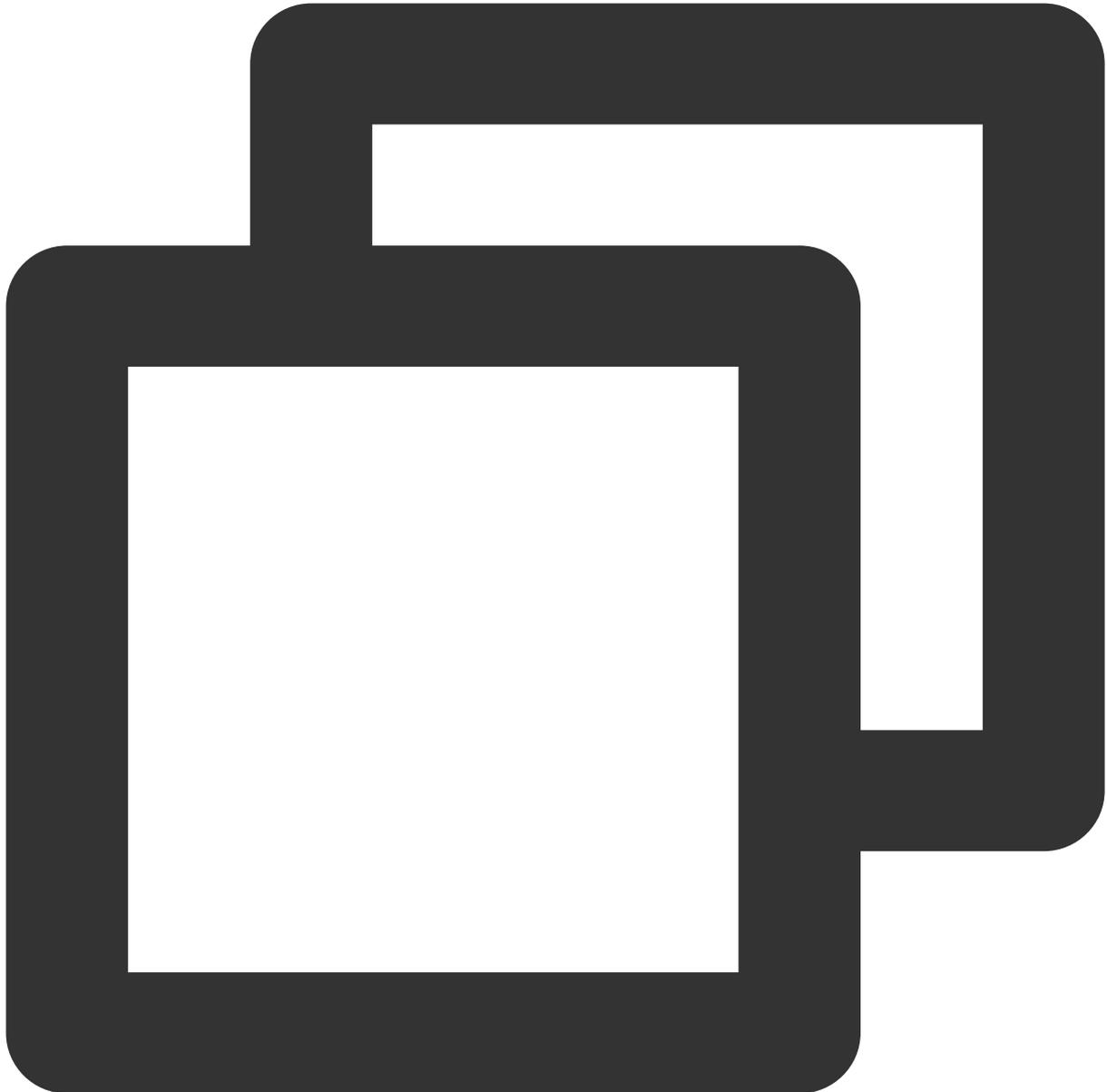


```
# Set the default version of the CVM product: 2017-03-12
tccli configure set cvm.version 2017-03-12
```

```
# Specify the version number in real time when using
tccli cvm help --version 2017-03-12
tccli cvm DescribeZones help --version 2017-03-12
tccli cvm DescribeZones --version 2017-03-12
```

Specify the nearest access point (Endpoint)

By default, TCCLI requests the nearest endpoint for accessing a service. You can also specify your own endpoint for a product (with CVM as an example).

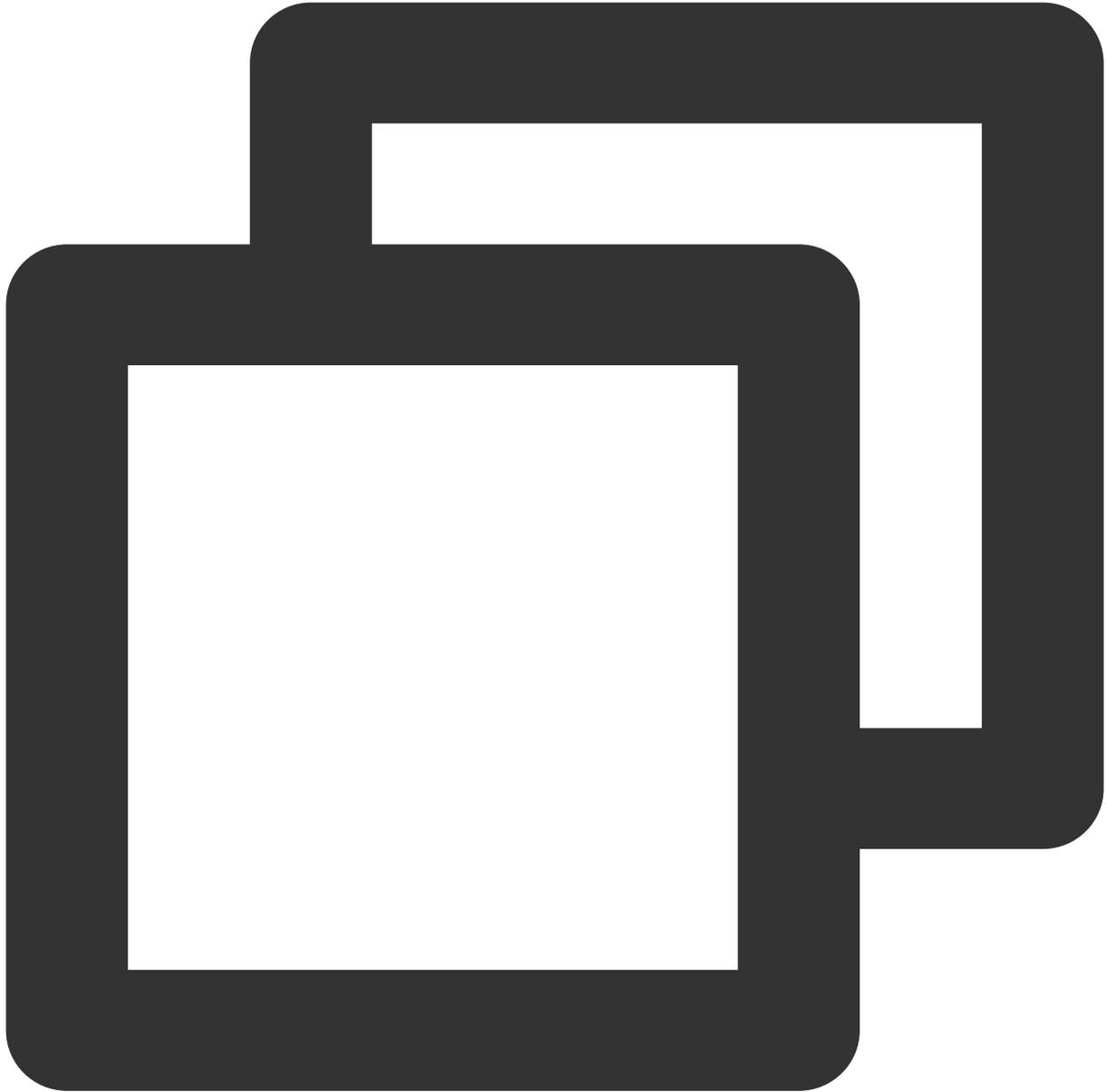


```
# Set the default endpoint of the CVM product
tccli configure set cvm.endpoint cvm.ap-guangzhou.tencentcloudapi.com

# Specify in real time when calling
tccli cvm DescribeZones --endpoint cvm.ap-guangzhou.tencentcloudapi.com
```

Filtering return results

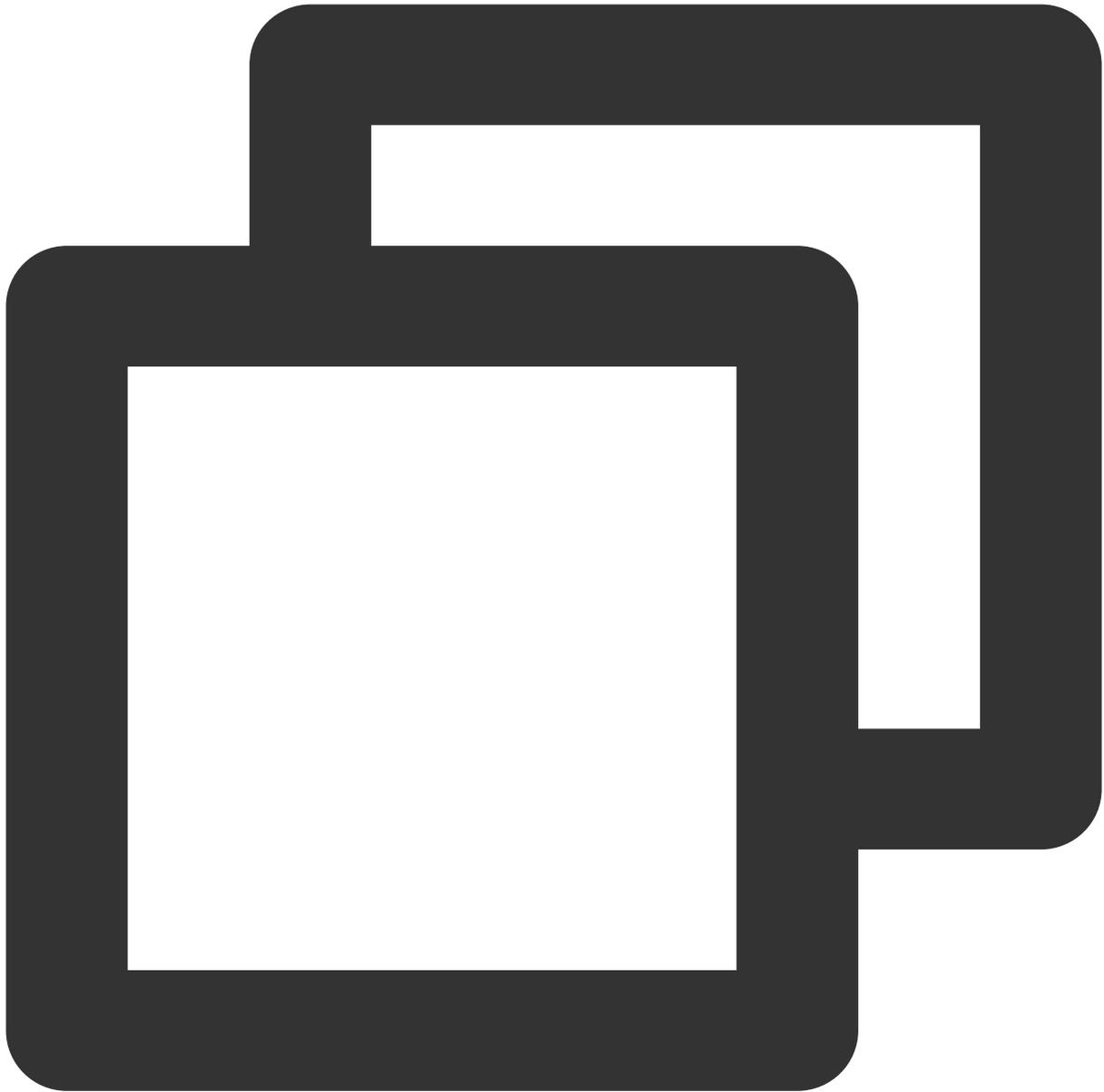
1. Output without any filtering (with the return result of CVM DescribeZones API as an example):



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

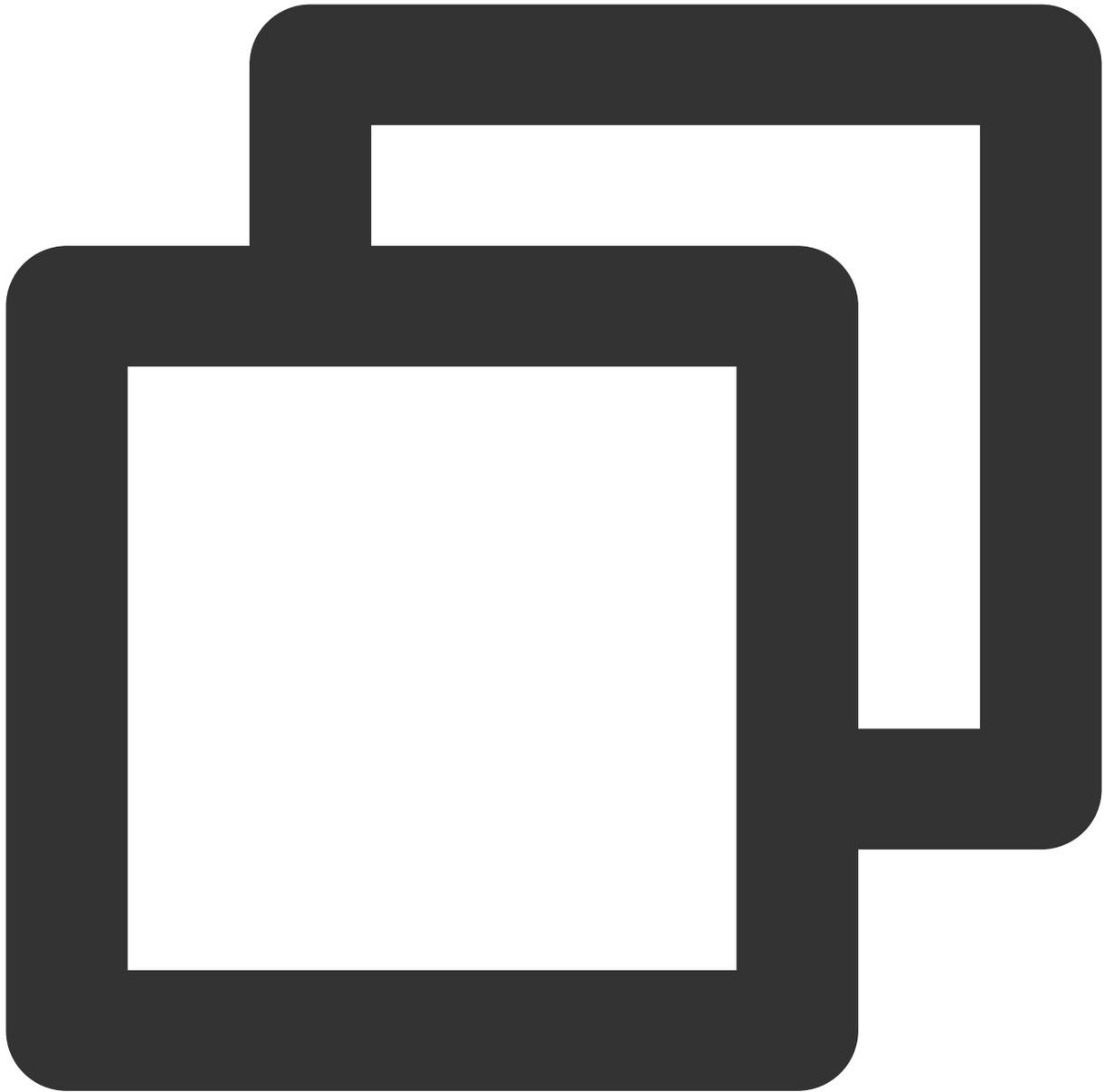
```
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100002",
    "Zone": "ap-guangzhou-2",
    "ZoneName": "Guangzhou Zone 2"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100003",
    "Zone": "ap-guangzhou-3",
    "ZoneName": "Guangzhou Zone 3"
  },
  {
    "ZoneState": "AVAILABLE",
    "ZoneId": "100004",
    "Zone": "ap-guangzhou-4",
    "ZoneName": "Guangzhou Zone 4"
  }
],
"RequestId": "4fd313a6-155f-4c7a-bf86-898c02fcae02"
}
```

2. View a specified field:



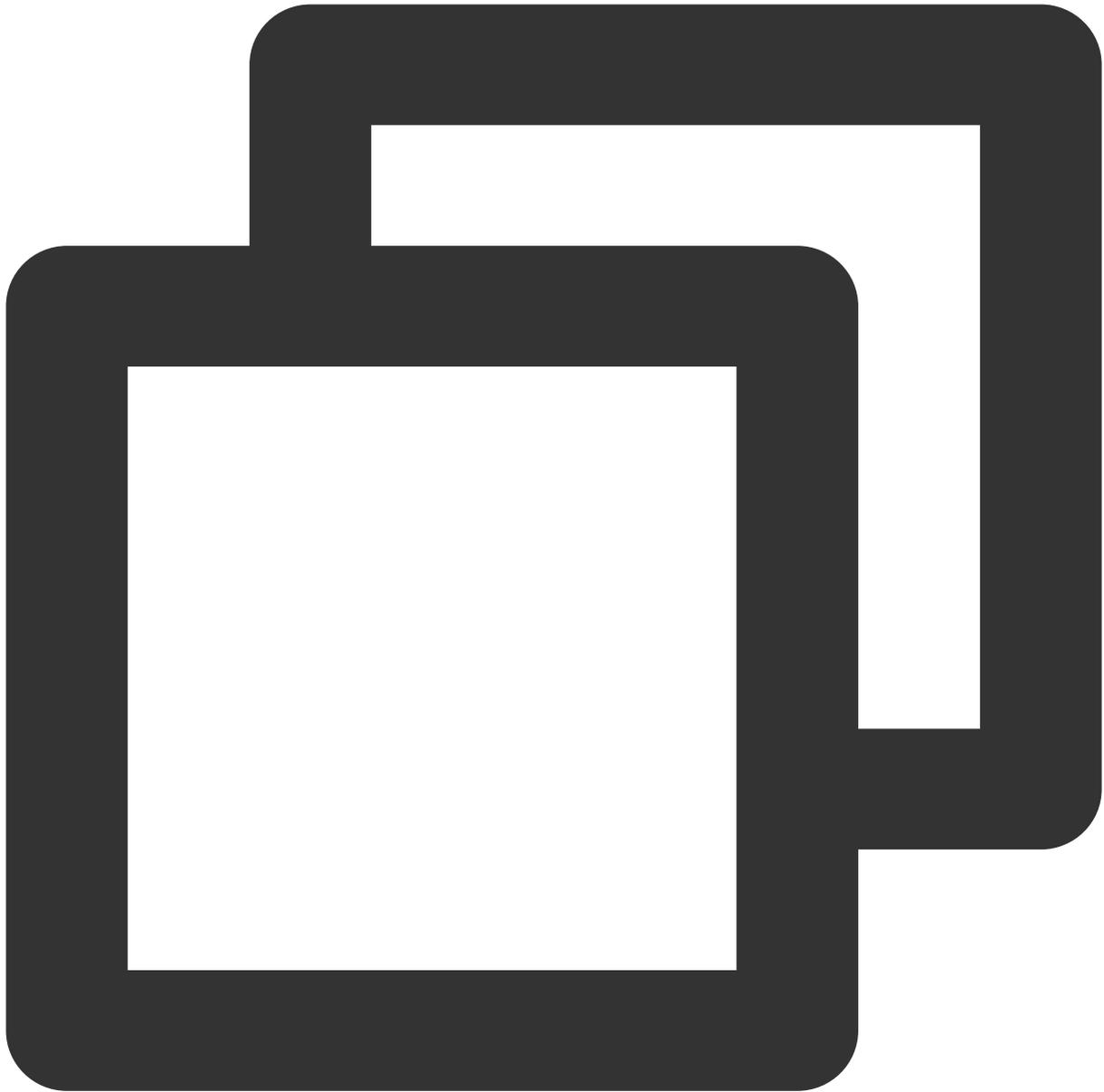
```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter TotalCount  
4
```

3. View the information of the Nth sub-object of a specified object in array type.



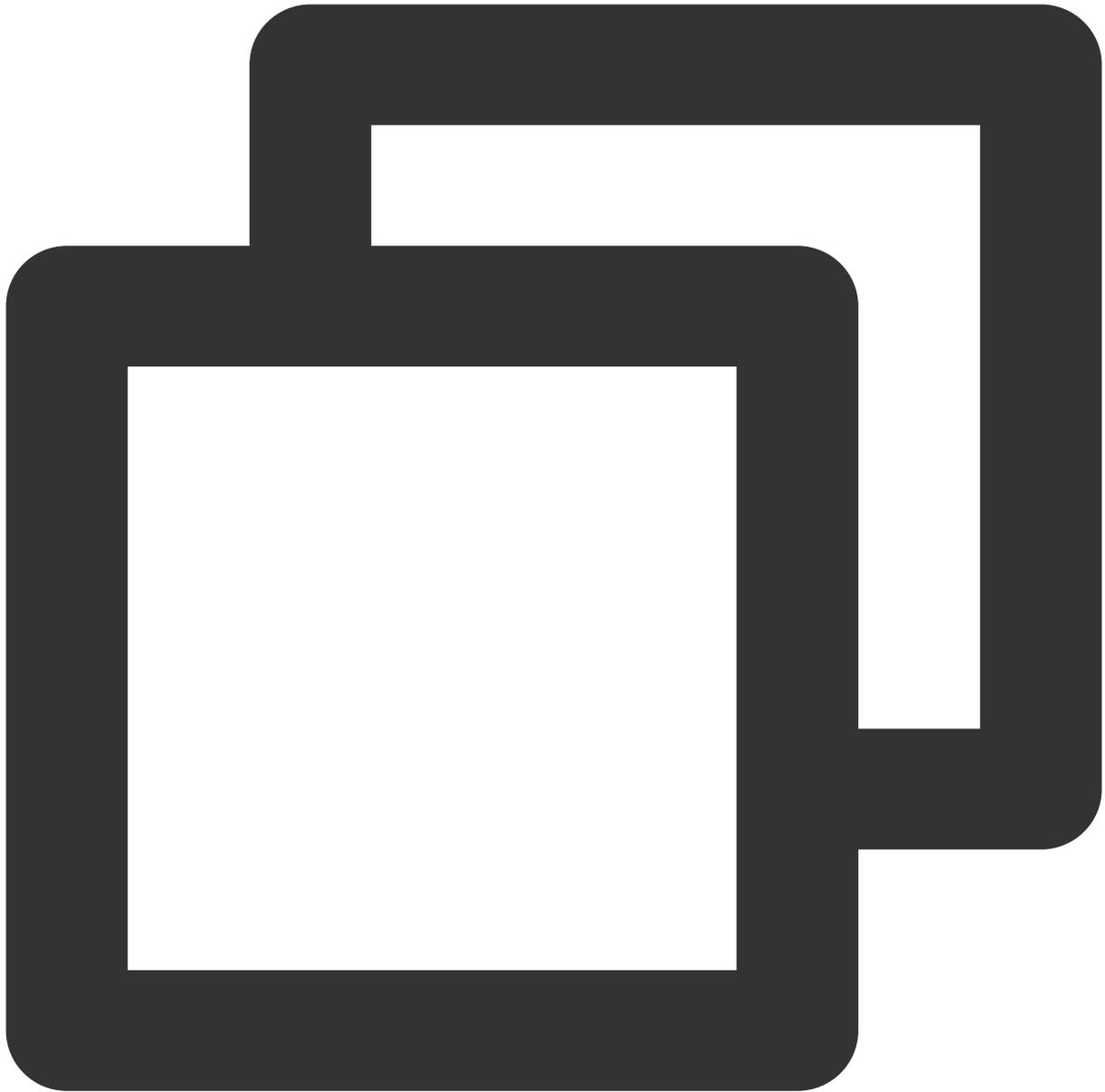
```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[0]
{
  "ZoneState": "AVAILABLE",
  "ZoneId": "100001",
  "Zone": "ap-guangzhou-1",
  "ZoneName": "Guangzhou Zone 1"
}
```

4. View a certain field of all the sub-objects with a certain name under the specified object in array type.



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter ZoneSet[*].ZoneName
[
  "Guangzhou Zone 1",
  "Guangzhou Zone 2",
  "Guangzhou Zone 3",
  "Guangzhou Zone 4"
]
```

5. Filter the sub-objects in the array and then present them under a new name:



```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter 'ZoneSet[*].{name:Zo
[
  {
    "name": "Guangzhou Zone 1",
    "id": "100001"
  },
  {
    "name": "Guangzhou Zone 2",
    "id": "100002"
  },
  {
```

```
        "name": "Guangzhou Zone 3",  
        "id": "100003"  
    },  
    {  
        "name": "Guangzhou Zone 4",  
        "id": "100004"  
    }  
]
```