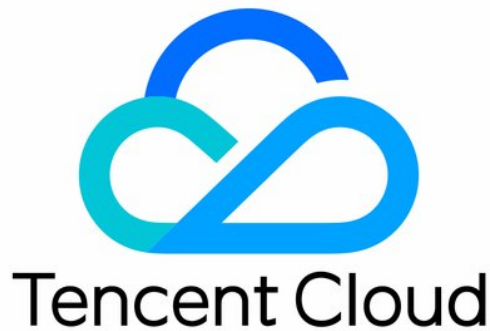


Tencent Cloud CLI

Operation Guide

Product Documentation



Copyright Notice

©2013-2019 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

Using TCCLI

Using Advanced Features

Operation Guide

Installing TCCLI

Last updated : 2020-03-11 19:59:13

Installing TCCLI

Operation Scenarios

This document describes how to install TCCLI International Version.

Prerequisites

Before installing CLI, make sure that your system has the Python environment and pip tool installed.

Note:

Python must be version 2.7 or higher. For more information, please see [Python's official website](#) and [pip's official website](#).

Notes

- TCCLI is dependent on the [TencentCloudApi Python SDK](#). If the version number of the TencentCloudApi Python SDK is lower than that of TCCLI to be installed, the TencentCloudApi Python SDK will be automatically upgraded when TCCLI is installed.
- If TCCLI Chinese Version has been installed, it will be automatically overwritten by the International Version.

Directions

1. Install TCCLI by running the following command:

```
pip install tccli-intl-en
```

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

```
tccli version
```

Note:

If your environment is Linux, you can enable autocomplete feature by running the following command:

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

Please note that a certificate issue may occur in Mac OS

Note :

While installing Python 3.6 or later versions in Mac OS, you may get an error saying

```
Error: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate in certificate chain (_ssl.c:1056).
```

This is because that in Mac OS, Python does not use the system default certificate nor provide a certificate.

HTTPS requests require a certificate provided by `certifi`, which cannot be specified by SDK. Therefore you need to install a certificate using the command `/Applications/Python 3.6/InstallCertificates.command` ."

Configuring TCCLI

Last updated : 2020-03-04 15:23:01

Operation Scenarios

This document describes how to configure TCCLI International Version.

Prerequisites

TCCLI has been installed.

Directions

To use TCCLI, you need to complete its initial configuration to make it meet the prerequisites of using TencentCloud API.

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

```
$ tccli configure
TencentCloud API secretId [*afcQ]:AKIDwLw1234MMfPRle2g9nR2OTI787aBCDP
TencentCloud API secretKey [*ArFd]:OxXj7khcV1234dQSSYNABcdCc1LiArFd
region: ap-guangzhou
output[json]:
```

- **secretId**: TencentCloud API key's SecretId.
- **secretKey**: TencentCloud API key's SecretKey.
- **region**: Tencent Cloud product region. Please go to the corresponding product page to get the information on available regions.
- **output**: Output format of the request return packet, which is optional. Valid values: json, table, text. Default value: json.

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

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

```
# The `set` subcommand can set certain configuration items or multiple configurations simultaneously.
```

```
tccli configure set secretId AKIDwLw1234MMfPRle2g9nR2OTI787aBCDP
tccli configure set region ap-guangzhou output json
# The `get` subcommand is used to obtain configuration information.
tccli configure get secretKey
secretKey = OxXj7khcV1234dQSSYNABcdCc1LiArFd
# The `list` subcommand prints out all configuration information.
tccli configure list
credential:
secretId = AKIDwLw1234MMfPRle2g9nR2OTI787aBCDP
secretKey = OxXj7khcV1234dQSSYNABcdCc1LiArFd
configure:
region = ap-guangzhou
output = json
```

For more information, please run the following command:

```
tccli configure [list get set] help
```

3. 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]:AKIDwLw1234MMfPRle2g9nR2OTI787aBCDP
TencentCloud API secretKey [*ArFd]:OxXj7khcV1234dQSSYNABcdCc1LiArFd
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
Specify an account when calling an API (e.g., cvm DescribeZones API).
tccli cvm DescribeZones --profile test
```

Using TCCLI

Last updated : 2020-03-04 15:22:50

Operation Scenarios

TCCLI integrates all the products on Tencent Cloud international website that support TencentCloud API and allows for configuration and management of such products. For example, you can use TCCLI to create and manage a CVM instance, create a CBS disk and view its usage, and create a VPC and add resources to it. All operations that can be done on console pages can be performed by running commands in TCCLI.

- Run the `tccli cvm DescribeInstances` command to see what CVM instances are under the current account.
- Run the `tccli cbs DescribeDisks` command to view the list of CBS disks.

Operation Examples

Note:

Please note that the non-simple parameters in the demo must be in standard json format.

Take creating a CVM instance as an example:

Mac and Linux:

```
tccli cvm RunInstances --InstanceChargeType POSTPAID_BY_HOUR --Placement '{"Zone": "ap-guangzhou-2"}' --InstanceType S1.SMALL1 --ImageId img-8toqc6s3 --SystemDisk '{"DiskType": "CLOUD_BASIC", "DiskSize": 50}' --InternetAccessible '{"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 10, "PublicIpAssigned": true}' --InstanceCount 1 --InstanceName TCCLI-TEST --LoginSettings '{"Password": "PleaseChange1t@"}' --HostName TCCLI-HOST-NAME1
```

Windows :

```
tccli cvm RunInstances --InstanceChargeType POSTPAID_BY_HOUR --Placement '{"Zone": "ap-guangzhou-2"}' --InstanceType S1.SMALL1 --ImageId img-8toqc6s3 --SystemDisk '{"DiskType": "CLOUD_BASIC", "DiskSize": 50}' --InternetAccessible '{"InternetChargeType": "TRAFFIC_POSTPAID_BY_HOUR", "InternetMaxBandwidthOut": 10, "PublicIpAssigned": true}' --InstanceCount 1 --InstanceName TCCLI-TEST --LoginSettings '{"Password": "PleaseChange1t@"}' --HostName TCCLI-HOST-NAME1
```


Note:

For details of more functions, you can view the supported products by running the `tccli help` command, view the supported APIs by running the `tccli cvm help` command (with CVM as an example), and view the parameters supported by APIs by running the `tccli cbs DescribeDisks help` command (with the DescribeDisks API of CBS as an example).

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:

Note:

The content that defines the filtering behavior needs to be marked with single quotes.

```
[root@VM_180_248_centos ~]# tccli cvm DescribeZones --filter 'ZoneSet[*].{name:
ZoneName, id:ZoneId}'
[
{
"name": "Guangzhou Zone 1",
"id": "100001"
},
{
"name": "Guangzhou Zone 2",
"id": "100002"
},
{
"name": "Guangzhou Zone 3",
"id": "100003"
},
{
"name": "Guangzhou Zone 4",
"id": "100004"
}
]
```