Elasticsearch Service

API Documentation

Product Documentation

Contents

API Documentation History

Last updated : 2020-10-16 18:27:17

Release 3

Release time: 2020-10-16 18:22:54

Release updates:

Improvement to existing documentation.

New APIs:

- RestartNodes

Modified APIs:

- CreateInstance
  - New input parameters:SceneType
- UpdateInstance
  - New input parameters:MultiZoneInfo, SceneType
- UpgradeInstance
  - New input parameters:UpgradeMode

Modified data structures:

- InstanceInfo
  - New members:SceneType

Release 2

Release time: 2020-08-13 20:05:18

Release updates:

Improvement to existing documentation.

New APIs:

- UpdatePlugins

Existing Release

Release time: 2020-07-30 19:52:43

Existing APIs/data structures are as follows:

Improvement to existing documentation.

Existing APIs:

- CreateInstance
- DeleteInstance

- DescribeInstanceLogs
- DescribeInstanceOperations
- DescribeInstances
- RestartInstance
- UpdateInstance
- UpgradeInstance
- UpgradeLicense

Existing data structures:

- CosBackup
- DictInfo
- EsAcl
- EsDictionaryInfo
- EsPublicAcl
- InstanceInfo
- InstanceLog
- KeyValue
- LocalDiskInfo
- MasterNodeInfo
- NodeInfo
- Operation
- OperationDetail
- SubTaskDetail
- TagInfo
- TaskDetail
- ZoneDetail

Introduction

Last updated：2019-09-09 21:30:36

Tencent Cloud Elasticsearch Service (ES) is a highly available and scalable cloud-managed Elasticsearch service built by Tencent Cloud based on the open-source search engine Elasticsearch. ES is deployed in VPCs and integrates components such as Elasticsearch and Kibana. It enables you to quickly deploy, easily manage, and scale your Elasticsearch clusters on demand. In addition, it simplifies complex OPS tasks such as cluster deployment and management, helping you construct various types of businesses with speed like log analysis, exception monitoring, website search and navigation, enterprise search, and BI analysis.

API Category

Last updated : 2020-10-16 18:27:16

Instance APIs

API Name	Feature
CreateInstance	Creates an ES cluster instance
DeleteInstance	Terminates an ES cluster instance
DescribeInstanceLogs	Queries ES cluster logs
DescribeInstanceOperations	Queries instance operation history
DescribeInstances	Queries ES cluster instances
RestartInstance	Restarts an ES cluster instance
RestartNodes	Restarts cluster nodes
UpdateInstance	Updates an ES cluster instance
UpdatePlugins	Changes plugin list
UpgradeInstance	Upgrades ES cluster version
UpgradeLicense	Upgrades ES X-Pack

Making API Requests
Request Structure

Last updated : 2020-02-18 19:10:14

1. Service Address

The API supports access from either a nearby region (at es.tencentcloudapi.com) or a specified region (at es.ap-guangzhou.tencentcloudapi.com for Guangzhou, for example).

We recommend using the domain name to access the nearest server. When you call an API, the request is automatically resolved to a server in the region **nearest** to the location where the API is initiated. For example, when you initiate an API request in Guangzhou, this domain name is automatically resolved to a Guangzhou server, the result is the same as that of specifying the region in the domain like "es.ap-guangzhou.tencentcloudapi.com".

**Note: For latency-sensitive businesses, we recommend that you specify the region in the domain name. **

Tencent Cloud currently supports the following regions:

Hosted region	Domain name
Local access region (recommended, only for non-financial availability zones)	es.tencentcloudapi.com
South China (Guangzhou)	es.ap-guangzhou.tencentcloudapi.com
East China (Shanghai)	es.ap-shanghai.tencentcloudapi.com
North China (Beijing)	es.ap-beijing.tencentcloudapi.com
Southwest China (Chengdu)	es.ap-chengdu.tencentcloudapi.com
Southwest China (Chongqing)	es.ap-chongqing.tencentcloudapi.com
Hong Kong, Macao, Taiwan (Hong Kong, China)	es.ap-hongkong.tencentcloudapi.com
Southeast Asia (Singapore)	es.ap-singapore.tencentcloudapi.com
Southeast Asia (Bangkok)	es.ap-bangkok.tencentcloudapi.com
South Asia (Mumbai)	es.ap-mumbai.tencentcloudapi.com
Northeast Asia (Seoul)	es.ap-seoul.tencentcloudapi.com
Northeast Asia (Tokyo)	es.ap-tokyo.tencentcloudapi.com
U.S. East Coast (Virginia)	es.na-ashburn.tencentcloudapi.com
U.S. West Coast (Silicon Valley)	es.na-siliconvalley.tencentcloudapi.com
North America (Toronto)	es.na-toronto.tencentcloudapi.com
Europe (Frankfurt)	es.eu-frankfurt.tencentcloudapi.com
Europe (Moscow)	es.eu-moscow.tencentcloudapi.com

Note: As financial availability zones and non-financial availability zones are isolated, when accessing the services in a financial availability zone (with the common parameter `Region` specifying a financial availability zone), it is

necessary to specify a domain name of the financial availability zone, preferably in the same region as specified in `Region`.

Access region for financial availability zone	Domain name for financial availability zone
East China (Shanghai Finance)	es.ap-shanghai-fsi.tencentcloudapi.com
South China (Shenzhen Finance)	es.ap-shenzhen-fsi.tencentcloudapi.com

2. Communications Protocol

All the Tencent Cloud APIs communicate via HTTPS, providing highly secure communication tunnels.

3. Request Methods

Supported HTTP request methods:

- POST (recommended)
- GET

The Content-Type types supported by POST requests:

- application/json (recommended). The TC3-HMAC-SHA256 signature algorithm must be used.
- application/x-www-form-urlencoded. The HmacSHA1 or HmacSHA256 signature algorithm must be used.
- multipart/form-data (only supported by certain APIs). You must use TC3-HMAC-SHA256 to calculate the signature.

The size of a GET request packet is up to 32 KB. The size of a POST request is up to 1 MB when the HmacSHA1 or HmacSHA256 signature algorithm is used, and up to 10 MB when TC3-HMAC-SHA256 is used.

4. Character Encoding

Only UTF-8 encoding is used.

Common Params

Last updated：2020-10-16 18:27:16

Common parameters are used for all APIs authenticating requestors. Common parameters must be included in all API requests, and they will not be described in individual API documents.

Signature Algorithm v3

When the TC3-HMAC-SHA256 algorithm is used, the common parameters should be uniformly placed in the HTTP request header, as shown below:

Parameter Name	Type	Required	Description
X-TC-Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter `Action` in the input parameters in related API documentation. For example, the API for querying the CVM instance list is `DescribeInstances`.
X-TC-Region	String	Yes	Region parameter, which is used to identify the region to which the data you want to work with belongs. For values supported for an API, see the description of common parameter `Region` in the input parameters in related API documentation. Note: This parameter is not required for some APIs (which will be indicated in related API documentation), and will not take effect even it is passed.
X-TC-Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. Note: If the difference between the UNIX timestamp and the server time is greater than 5 minutes, a signature expiration error may occur.
X-TC-Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter `Version` in the API documentation. For example, the version of CVM is 2017-03-12.
Authorization	String	Yes	The HTTP authentication request header, for example: TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 Here: - TC3-HMAC-SHA256: Signature method, currently fixed as this value; - Credential: Signature credential; AKIDEXAMPLE is the SecretId; Date is a date in UTC time, and this value must match the value of X-TC-Timestamp (a common parameter) in UTC time format; service is the name of the product/service, and is generally a domain name prefix. For example, a domain name cvm.tencentcloudapi.com refers to the CVM product and the value would be cvm; - SignedHeaders: The headers that contains the authentication information; content-type and host are the required headers; - Signature: Signature digest.
X-TC-Token	String	No	The token used for a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

Assuming you want to query the list of Cloud Virtual Machine instances in the Guangzhou region, the request structure in the form of request URL, request header and request body may be as follows:

Example of an HTTP GET request structure:

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

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

The following example shows you how to structure an HTTP POST (application/json) request:

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

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

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

Example of an HTTP POST (multipart/form-data) request structure (only supported by specific APIs):

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

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

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

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

10
--58731222010402--
```

Signature Algorithm v1

To adopt the HmacSHA1 and HmacSHA256 signature methods, common parameters must be put into the request string, as shown below:

Parameter Name	Type	Required	Description
Action	String	Yes	The name of the API for the desired operation. For the specific value, see the description of common parameter `Action` in the input parameters in related API

			documentation. For example, the API for querying the CVM instance list is `DescribeInstances`.
Region	String	Yes	Region parameter, which is used to identify the region to which the data you want to work with belongs. For values supported for an API, see the description of common parameter `Region` in the input parameters in related API documentation. Note: This parameter is not required for some APIs (which will be indicated in related API documentation), and will not take effect even if it is passed.
Timestamp	Integer	Yes	The current UNIX timestamp that records the time when the API request was initiated, for example, 1529223702. If the difference between the value and the current system time is too large, a signature expiration error may occur.
Nonce	Integer	Yes	A random positive integer used along with `Timestamp` to prevent replay attacks.
SecretId	String	Yes	The identifying SecretId obtained on the Cloud API Key page. A SecretId corresponds to a unique SecretKey which is used to generate the request signature (Signature).
Signature	String	Yes	Request signature used to verify the validity of this request. This is calculated based on the actual input parameters. For more information about how this is calculated, see the API authentication documentation.
Version	String	Yes	API version of the action. For the valid values, see the description of the common input parameter `Version` in the API documentation. For example, the version of CVM is 2017-03-12.
SignatureMethod	String	No	Signature method. Currently, only HmacSHA256 and HmacSHA1 are supported. The HmacSHA256 algorithm is used to verify the signature only when this parameter is specified as HmacSHA256. In other cases, the signature is verified with HmacSHA1.
Token	String	No	The token used for a temporary certificate. It must be used with a temporary key. You can obtain the temporary key and token by calling a CAM API. No token is required for a long-term key.

Assuming you want to query the list of Cloud Virtual Machine instances in the Guangzhou region, the request structure in the form of request URL, request header and request body may be as follows:

Example of an HTTP GET request structure:

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

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

Example of an HTTP POST request structure:

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

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

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

Region List

The supported Region field values for all APIs in this product are listed as below. For any API that does not support any of the following regions, this field will be described additionally in the relevant API document.

Region	Value
North China (Beijing)	ap-beijing
Southwest China (Chengdu)	ap-chengdu
Southwest China (Chongqing)	ap-chongqing
South China (Guangzhou)	ap-guangzhou
Hong Kong/Macao/Taiwan (Hong Kong, China)	ap-hongkong
South Asia Pacific (Mumbai)	ap-mumbai
East China (Nanjing)	ap-nanjing
Northeast Asia Pacific (Seoul)	ap-seoul
East China (Shanghai)	ap-shanghai
East China (Shanghai Finance)	ap-shanghai-fsi
South China (Shenzhen Finance)	ap-shenzhen-fsi
Southeast Asia Pacific (Singapore)	ap-singapore
Northeast Asia Pacific (Tokyo)	ap-tokyo
Europe (Frankfurt)	eu-frankfurt
Europe (Moscow)	eu-moscow
Eastern U.S. (Virginia)	na-ashburn
Western U.S. (Silicon Valley)	na-siliconvalley
North America (Toronto)	na-toronto

Signature v3

Last updated：2020-10-16 18:27:16

TencentCloud API authenticates every single request, i.e., the request must be signed using the security credentials in the designated steps. Each request has to contain the signature information (Signature) in the common request parameters and be sent in the specified way and format.

Applying for Security Credentials

The security credential used in this document is a key, which includes a SecretId and a SecretKey. Each user can have up to two pairs of keys.

- SecretId: Used to identify the API caller, which is just like a username.
- SecretKey: Used to authenticate the API caller, which is just like a password.
- **You must keep your security credentials private and avoid disclosure; otherwise, your assets may be compromised. If they are disclosed, please disable them as soon as possible.**

You can apply for the security credentials through the following steps:

1. Log in to the Tencent Cloud Console.
2. Go to the TencentCloud API Key console page.
3. On the TencentCloud API Key page, click **Create** to create a SecretId/SecretKey pair.

Using the Resources for Developers

TencentCloud API comes with SDKs for seven commonly used programming languages, including Python, Java, PHP, Go, NodeJS and .NET. In addition, it provides API Explorer which enables online call, signature verification, and SDK code generation. If you have any troubles calculating a signature, consult these resources.

TC3-HMAC-SHA256 Signature Algorithm

Compatible with the previous HmacSHA1 and HmacSHA256 signature algorithms, the TC3-HMAC-SHA256 signature algorithm is more secure and supports larger requests and JSON format with better performance. We recommend using TC3-HMAC-SHA256 to calculate the signature.

TencentCloud API supports both GET and POST requests. For the GET method, only the Content-Type: application/x-www-form-urlencoded protocol format is supported. For the POST method, two protocol formats, Content-Type: application/json and Content-Type: multipart/form-data, are supported. The JSON format is supported by default for all business APIs, and the multipart format is supported only for specific business APIs. In this case, the API cannot be called in JSON format. See the specific business API documentation for more information. The POST method is recommended, as there is no difference in the results of both the methods, but the GET method only supports request packets up to 32 KB.

The following uses querying the list of CVM instances in the Guangzhou region as an example to describe the steps of signature splicing. We chose this API because:

1. CVM is activated by default, and this API is often used;
2. It is read-only and does not change the status of existing resources;
3. It covers many types of parameters, which allows it to be used to demonstrate how to use arrays containing data structures.

In the example, we try to choose common parameters and API parameters that are prone to mistakes. When you actually call an API, please use parameters based on the actual conditions. The parameters vary by API. Do not copy the parameters and values in this example.

Assuming that your SecretId and SecretKey are `AKIDz8krbsJ5yKBZQpn74WFkmLPx3******` and `Gu5t9xGARNpq86cd98joQYCN3******`, respectively, if you want to view the status of the instance in the Guangzhou region whose CVM instance name is "unnamed" and have only one data entry returned, then the request may be:

```
curl -X POST https://cvm.tencentcloudapi.com ¥
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3******/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff" ¥
-H "Content-Type: application/json; charset=utf-8" ¥
-H "Host: cvm.tencentcloudapi.com" ¥
-H "X-TC-Action: DescribeInstances" ¥
-H "X-TC-Timestamp: 1551113065" ¥
-H "X-TC-Version: 2017-03-12" ¥
-H "X-TC-Region: ap-guangzhou" ¥
-d '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
```

The signature calculation process is explained in detail below.

1. Concatenating the CanonicalRequest String

Concatenate the canonical request string (CanonicalRequest) in the following pseudocode format:

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

Field Name	Explanation
HTTPRequestMethod	HTTP request method (GET or POST). This example uses `POST`.
CanonicalURI	URI parameter. Slash ("/") is used for API 3.0.
CanonicalQueryString	The query string in the URL of the originating HTTP request. This is always an empty string "" for POST requests, and is the string after the question mark (?) for GET requests. For example: Limit=10&Offset=0。 Note: `CanonicalQueryString` must be URL-encoded, referencing RFC3986, the UTF8 character set. We recommend using the programming language library. All special characters must be encoded and capitalized.
CanonicalHeaders	Header information for signature calculation, including at least two headers of `host` and `content-type`. Custom headers can be added to participate in the signature process to improve the uniqueness and security of the request. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase with the leading and trailing spaces removed, so they are concatenated in the format of key:value\n format; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase). The calculation result in this example is `content-type:application/json; charset=utf-8¥nhost:cvm.tencentcloudapi.com¥n`. Note: `content-type` must match the actually sent content. In some programming languages, a charset value would be added even if it is not specified. In this case, the request sent is different from the one signed, and the sever will return an error indicating that signature verification failed.

SignedHeaders	Header information for signature calculation, indicating which headers of the request participate in the signature process (they must each individually correspond to the headers in CanonicalHeaders). `Content-type` and `host` are required headers. Concatenation rules: 1. Both the key and value of the header should be converted to lowercase; 2. If there are multiple headers, they should be sorted in ASCII ascending order by the header keys (lowercase) and separated by semicolons (;). The value in this example is `content-type;host`
HashedRequestPayload	Hash value of the request payload (i.e., the body, such as `{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}` in this example). The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(RequestPayload))) by SHA256 hashing the payload of the HTTP request, performing hexadecimal encoding, and finally converting the encoded string to lowercase letters. For GET requests, `RequestPayload` is always an empty string. The calculation result in this example is `99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907`.

According to the rules above, the `CanonicalRequest` string obtained in the example is as follows:

```
POST
/

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

content-type;host
99d58dfbc6745f6747f36bfca17dee5e6881dc0428a0a36f96199342bc5b4907
```

2. Concatenating the String to Be Signed

The string to sign is concatenated as follows:

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

Field Name	Explanation
Algorithm	Signature algorithm, which is currently always `TC3-HMAC-SHA256`.
RequestTimestamp	Request timestamp, i.e., the value of the common parameter `X-TC-Timestamp` in the request header, which is the UNIX timestamp of the current time in seconds, such as `1551113065` in this example.
CredentialScope	Scope of the credential in the format of `Date/service/tc3_request`, including the date, requested service and termination string (tc3_request). `Date` **is a date in UTC time, whose value should match the UTC date converted by the common parameter** `X-TC-Timestamp`**;** `service` is the product name, which should match the domain name of the product called. The calculation result in this example is `2019-02-25/cvm/tc3_request`.
HashedCanonicalRequest	Hash value of the CanonicalRequest string concatenated in the steps above. The pseudocode for calculation is Lowercase(HexEncode(Hash.SHA256(CanonicalRequest))). The calculation result in this example is `2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a`.

Note:

1. Date has to be calculated from the timestamp "X-TC-Timestamp" and the time zone is UTC+0. If you add the system's local time zone information (such as UTC+8), calls can succeed both day and night but will definitely fail at 00:00. For example, if the timestamp is 1551113065 and the time in UTC+8 is 2019-02-26 00:44:25, the UTC+0 date in the calculated Date value should be 2019-02-25 instead of 2019-02-26.
2. Timestamp must be the same as your current system time, and your system time and standard time must be synced; if the difference between Timestamp and your current system time is larger than five minutes, the request will fail. If your system time is out of sync with the standard time for a while, the request will fail and return a signature expiration error.

According to the preceding rules, the string to be signed obtained in the example is as follows:

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
2815843035062fffda5fd6f2a44ea8a34818b0dc46f024b8b3786976a3adda7a
```

3. Calculating the Signature

1) Calculate the derived signature key with the following pseudocode:

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

Field Name	Explanation
SecretKey	The original SecretKey, i.e., `Gu5t9xGARNpq86cd98joQYCN3*******` .
Date	The Date field information in `Credential` , such as `2019-02-25` in this example.
Service	Value in the Service field in `Credential` , such as `cvm` in this example.

2) Calculate the signature with the following pseudocode:

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

4. Concatenating the Authorization

The Authorization is concatenated as follows:

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

Field Name	Explanation
Algorithm	Signature algorithm, which is always `TC3-HMAC-SHA256` .
SecretId	The SecretId in the key pair, i.e., `AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******` .
CredentialScope	Credential scope (see above). The calculation result in this example is `2019-02-25/cvm/tc3_request` .

SignedHeaders	Header information for signature calculation (see above), such as `content-type;host` in this example.
Signature	Signature value. The calculation result in this example is `c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff`.

According to the rules above, the value obtained in the example is:

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
```

The following example shows a finished authorization header:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=c492e8e41437e97a620b728c301bb8d17e7dc0c17eeabce80c20cd70fc3a78ff
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}
```

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Tencent Cloud SDK 3.0 which encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- NodeJS
- .NET

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGel%3D&Timestamp=1465185768&Version=2017-03-12.

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

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

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

public static byte[] hmac256(byte[] key, String msg) throws Exception {
Mac mac = Mac.getInstance("HmacSHA256");
SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
mac.init(secretKeySpec);
return mac.doFinal(msg.getBytes(UTF8));
}

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

public static void main(String[] args) throws Exception {
String service = "cvm";
String host = "cvm.tencentcloudapi.com";
String region = "ap-guangzhou";
String action = "DescribeInstances";
String version = "2017-03-12";
String algorithm = "TC3-HMAC-SHA256";
String timestamp = "1551113065";
//String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
// Pay attention to the time zone; otherwise, errors may occur
sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

// ************ Step 1: Concatenate the CanonicalRequest string ************
String httpRequestMethod = "POST";
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8¥n" + "host:" + host + "¥n";
String signedHeaders = "content-type;host";

String payload = "{¥"Limit¥": 1, ¥"Filters¥": [{¥"Values¥": [¥"unnamed¥"], ¥"Name¥": ¥"instance-name¥"}]}";
String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "¥n" + canonicalUri + "¥n" + canonicalQueryString + "¥n"
+ canonicalHeaders + "¥n" + signedHeaders + "¥n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ************ Step 2: Concatenate the string to sign ************
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = sha256Hex(canonicalRequest);
```

```
String stringToSign = algorithm + "¥n" + timestamp + "¥n" + credentialScope + "¥n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ************ Step 3: Calculate the signature ************
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ************ Step 4: Concatenate the Authorization ************
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

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

Python

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

# Key Parameters
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*******"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": ["unnamed"]}]}

# ************ Step 1: Concatenate the CanonicalRequest string ************
```

```
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s¥nhost:%s¥n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "¥n" +
canonical_uri + "¥n" +
canonical_querystring + "¥n" +
canonical_headers + "¥n" +
signed_headers + "¥n" +
hashed_request_payload)
print(canonical_request)

# ************* Step 2: Concatenate the string to sign *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "¥n" +
str(timestamp) + "¥n" +
credential_scope + "¥n" +
hashed_canonical_request)
print(string_to_sign)


# ************* Step 3: Calculate the Signature *************
# Function for computing signature digest
def sign(key, msg):
return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* Step 4: Concatenate the Authorization *************
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

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

Golang

```
package main

import (
"crypto/hmac"
"crypto/sha256"
"encoding/hex"
```

```
"fmt"
"time"
)

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

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

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

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

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

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

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

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

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*******";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

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

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

// step 3: sign string
```

```
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

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

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"'
.' -H "Content-Type: application/json; charset=utf-8"'
.' -H "Host: '.$host.'"'
.' -H "X-TC-Action: '.$action.'"'
.' -H "X-TC-Timestamp: '.$timestamp.'"'
.' -H "X-TC-Version: '.$version.'"'
.' -H "X-TC-Region: '.$region.'"'
." -d '".$payload."'";
echo $curl.PHP_EOL;
```

Ruby

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

# Key Parameters
secret_id = 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*******'

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

# ************* Step 1: Concatenate the CanonicalRequest string *************
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8¥nhost:#{host}¥n"
signed_headers = 'content-type;host'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['unnamed'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["unnamed"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
http_request_method,
canonical_uri,
```

```
canonical_querystring,
canonical_headers,
signed_headers,
hashed_request_payload,
].join("¥n")

puts canonical_request

# ************* Step 2: Concatenate the string to sign *************
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
algorithm,
timestamp.to_s,
credential_scope,
hashed_request_payload,
].join("¥n")
puts string_to_sign

# ************* Step 3: Calculate the Signature *************
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
puts signature

# ************* Step 4: Concatenate the Authorization *************
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signatur
e}"
puts authorization

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

DotNet

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

public class Application
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
```

```
return builder.ToString();
}
}
public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

public static Dictionary<String, String> BuildHeaders(string secretid,
string secretkey, string service, string endpoint, string region,
string action, string version, DateTime date, string requestPayload)
{
string datestr = date.ToString("yyyy-MM-dd");
DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc);
long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
// ************* Step 1: Concatenate the CanonicalRequest string *************
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8¥n" + "host:" + endpoint + "¥n";
string signedHeaders = "content-type;host";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "¥n"
+ canonicalUri + "¥n"
+ canonicalQueryString + "¥n"
+ canonicalHeaders + "¥n"
+ signedHeaders + "¥n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);
Console.WriteLine("-----------------------------------");

// ************* Step 2: Concatenate the string to sign *************
string credentialScope = datestr + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
string stringToSign = algorithm + "¥n" + requestTimestamp.ToString() + "¥n" + credentialScope + "¥n" + hashedCanonicalRequest;
Console.WriteLine(stringToSign);
Console.WriteLine("-----------------------------------");

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

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

Dictionary<string, string> headers = new Dictionary<string, string>();
```

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

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

// The timestamp `2019-02-26 00:44:25` used here is only for reference. In a project, use the following parameter:
// DateTime date = DateTime.UtcNow;
// Enter the correct time zone. We recommend using UTC timestamp to avoid errors.
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{¥"Limit¥": 1, ¥"Filters¥": [{¥"Values¥": [¥"¥¥u672a¥¥u547d¥¥u540d¥"], ¥"Name¥": ¥"instance-name¥"}]}";

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

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

NodeJS

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

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

```
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3******"

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
const date = getDate(timestamp)

// ************ Step 1: Concatenate the CanonicalRequest string ************
const signedHeaders = "content-type;host"

const payload = "{¥"Limit¥": 1, ¥"Filters¥": [{¥"Values¥": [¥"unnamed¥"], ¥"Name¥": ¥"instance-name¥"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8¥n" + "host:" + endpoint + "¥n"

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

// ************ Step 2: Concatenate the string to sign ************
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "¥n" +
timestamp + "¥n" +
credentialScope + "¥n" +
hashedCanonicalRequest
console.log(stringToSign)
console.log("----------------------------")

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

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

const Call_Information = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"'
+ ' -H "Content-Type: application/json; charset=utf-8"'
```

```
+ ' -H "Host: ' + endpoint + '"'
+ ' -H "X-TC-Action: ' + action + '"'
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'
+ ' -H "X-TC-Version: ' + version + '"'
+ ' -H "X-TC-Region: ' + region + '"'
+ " -d '" + payload + "'"
console.log(Call_Information)
}
main()
```

C++

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

using namespace std;

string get_data(int64_t &timestamp)
{
string utcDate;
char buff[20] = {0};
// time_t timenow;
struct tm sttime;
sttime = *gmtime(&timestamp);
strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
utcDate = string(buff);
return utcDate;
}
string int2str(int64_t n)
{
std::stringstream ss;
ss << n;
return ss.str();
}
string sha256Hex(const string &str)
{
char buf[3];
unsigned char hash[SHA256_DIGEST_LENGTH];
SHA256_CTX sha256;
SHA256_Init(&sha256);
SHA256_Update(&sha256, str.c_str(), str.size());
SHA256_Final(hash, &sha256);
std::string NewString = "";
for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
{
snprintf(buf, sizeof(buf), "%02x", hash[i]);
NewString = NewString + buf;
}
return NewString;
}
string HmacSha256(const string &key, const string &input)
{
unsigned char hash[32];

HMAC_CTX *h;
```

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

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

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

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

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

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

int main()
{
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******";

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

// ************* Step 1: Concatenate the CanonicalRequest string *************
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string canonicalHeaders = "content-type:application/json; charset=utf-8¥nhost:" + host + "¥n";
```

```
string signedHeaders = "content-type;host";
string payload = "{¥"Limit¥": 1, ¥"Filters¥": [{¥"Values¥": [¥"unnamed¥"], ¥"Name¥": ¥"instance-name¥"}]}";
string hashedRequestPayload = sha256Hex(payload);
string canonicalRequest = httpRequestMethod + "¥n" + canonicalUri + "¥n" + canonicalQueryString + "¥n"
+ canonicalHeaders + "¥n" + signedHeaders + "¥n" + hashedRequestPayload;
cout << canonicalRequest << endl;
cout << "-----------------------" << endl;

// ************* Step 2: Concatenate the string to sign *************
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "¥n" + RequestTimestamp + "¥n" + credentialScope + "¥n" + hashedCanonicalRequest;
cout << stringToSign << endl;
cout << "-----------------------" << endl;

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

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

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

Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error Code	Description
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes.
AuthFailure.SecretIdNotFound	The key does not exist. Please go to the console to check whether it is disabled or you copied fewer or more characters.
AuthFailure.SignatureFailure	Signature error. It is possible that the signature was calculated incorrectly, the signature does not match the content actually sent, or the SecretKey is incorrect.
AuthFailure.TokenFailure	Temporary certificate token error.

AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).

Signature

Last updated：2020-08-14 09:39:08

Tencent Cloud API authenticates each access request, i.e. each request needs to include authentication information (Signature) in the common parameters to verify the identity of the requester.
The Signature is generated by the security credentials which include SecretId and SecretKey. If you don't have the security credentials yet, go to the TencentCloud API Key page to apply for them; otherwise, you cannot invoke the TencentCloud API.

1. Applying for Security Credentials

Before using the TencentCloud API for the first time, go to the TencentCloud API Key page to apply for security credentials.
Security credentials consist of SecretId and SecretKey:

- SecretId is used to identify the API requester.
- SecretKey is used to encrypt the signature string and verify it on the server.
- **You must keep your security credentials private and avoid disclosure.**

You can apply for the security credentials through the following steps:

1. Log in to the Tencent Cloud Console.
2. Go to the TencentCloud API Key page.
3. On the API Key Management page, click **Create Key** to create a SecretId/SecretKey pair.

Note: Each account can have up to two pairs of SecretId/SecretKey.

2. Generating a Signature

With the SecretId and SecretKey, a signature can be generated. The following describes how to generate a signature:

Assume that the SecretId and SecretKey are:

- SecretId: `AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******`
- SecretKey: `Gu5t9xGARNpq86cd98joQYCN3*******`

Note: This is just an example. For actual operations, please use your own SecretId and SecretKey.

Take the Cloud Virtual Machine's request to view the instance list (DescribeInstances) as an example. When you invoke this API, the request parameters may be as follows:

Parameter name	Description	Parameter value
Action	Method name	DescribeInstances
SecretId	Key ID	`AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******`
Timestamp	Current timestamp	1465185768
Nonce	Random positive integer	11886
Region	Region where the instance is located	ap-guangzhou
InstanceIds.0	ID of the instance to query	ins-09dx96dg
Offset	Offset	0

Parameter name	Description	Parameter value
Limit	Allowed maximum output	20
Version	API version number	2017-03-12

2.1. Sorting Parameters

First, sort all the request parameters in an ascending lexicographical order (ASCII code) by their names. Notes: (1) Parameters are sorted by their names instead of their values; (2) The parameters are sorted based on ASCII code, not in an alphabetical order or by values. For example, InstanceIds.2 should be arranged after InstanceIds.12. You can complete the sorting process using a sorting function in a programming language, such as the ksort function in PHP. The parameters in the example are sorted as follows:

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******',
'Timestamp' : 1465185768,
'Version': '2017-03-12',
}
```

When developing in another programming language, you can sort these sample parameters and it will work as long as you obtain the same results.

2.2. Concatenating a Request String

This step generates a request string.
Format the request parameters sorted in the previous step into the form of "parameter name"="parameter value". For example, for the Action parameter, its parameter name is "Action" and its parameter value is "DescribeInstances", so it will become Action=DescribeInstances after formatted.
Note: The "parameter value" is the original value but not the value after URL encoding.

Then, concatenate the formatted parameters with "&". The resulting request string is as follows:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQ
pn74WFkmLPx3*******&Timestamp=1465185768&Version=2017-03-12
```

2.3. Concatenating the Signature Original String

This step generates a signature original string.
The signature original string consists of the following parameters:

1. HTTP method: POST and GET modes are supported, and GET is used here for the request. Please note that the method name should be in all capital letters.
2. Request server: the domain name of the request to view the list of instances (DescribeInstances) is cvm.tencentcloudapi.com. The actual request domain name varies by the module to which the API belongs. For more information, see the instructions of the specific API.
3. Request path: The request path in the current version of TencentCloud API is fixed to /.
4. Request string: the request string generated in the previous step.

The concatenation rule of the signature original string is: Request method + request host + request path + ? + request string

The concatenation result of the example is:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Timestamp=1465185768&Version=2017-03-12
```

2.4. Generating a Signature String

This step generates a signature string.
First, use the HMAC-SHA1 algorithm to sign the **signature original string** obtained in the previous step, and then encode the generated signature using Base64 to obtain the final signature.

The specific code is as follows with the PHP language being used as an example:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3*******';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

The final signature is:

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

When developing in another programming language, you can sign and verify the original in the example above and it works as long as you get the same results.

3. Encoding a Signature String

The generated signature string cannot be directly used as a request parameter and must be URL encoded.

For example, if the signature string generated in the previous step is zmmjn35mikh6pM3V7sUEuX4wyYM=, the final signature string request parameter (Signature) is zmmjn35mikh6pM3V7sUEuX4wyYM%3D, which will be used to generate the final request URL.

Note: If your request method is GET, or the request method is POST and the Content-Type is application/x-www-form-urlencoded, then all the request parameter values need to be URL encoded (except the parameter key and the symbol of =) when sending the request. Non-ASCII characters need to be encoded with UTF-8 before URL encoding.

Note: The network libraries of some programming languages automatically URL encode all parameters, in which case there is no need to URL encode the signature string; otherwise, two rounds of URL encoding will cause the signature to fail.

Note: Other parameter values also need to be encoded using RFC 3986. Use %XY in percent-encoding for special characters such as Chinese characters, where "X" and "Y" are hexadecimal characters (0-9 and uppercase A-F), and using lowercase will cause an error.

4. Signature Failure

The following situational error codes for signature failure may occur. Please resolve the errors accordingly.

Error code	Error description
AuthFailure.SignatureExpire	The signature is expired
AuthFailure.SecretIdNotFound	The key does not exist
AuthFailure.SignatureFailure	Signature error

Error code	Error description
AuthFailure.TokenFailure	Token error
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type)

5. Signature Demo

When calling API 3.0, you are recommended to use the corresponding Tencent Cloud SDK 3.0 which encapsulates the signature process, enabling you to focus on only the specific APIs provided by the product when developing. See SDK Center for more information. Currently, the following programming languages are supported:

- Python
- Java
- PHP
- Go
- NodeJS
- .NET

To further explain the signing process, we will use a programming language to implement the process described above. The request domain name, API and parameter values in the sample are used here. This goal of this example is only to provide additional clarification for the signature process, please see the SDK for actual usage.

The final output URL might be: `https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%3D&Timestamp=1465185768&Version=2017-03-12` .

Note: The key in the example is fictitious, and the timestamp is not the current time of the system, so if this URL is opened in the browser or called using commands such as curl, an authentication error will be returned: Signature expired. In order to get a URL that can work properly, you need to replace the SecretId and SecretKey in the example with your real credentials and use the current time of the system as the Timestamp.

Note: In the example below, even if you use the same programming language, the order of the parameters in the URL may be different for each execution. However, the order does not matter, as long as all the parameters are included in the URL and the signature is calculated correctly.

Note: The following code is only applicable to API 3.0. It cannot be directly used in other signature processes. Even with an older API, signature calculation errors may occur due to the differences in details. Please refer to the corresponding documentation.

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
private final static String CHARSET = "UTF-8";

public static String sign(String s, String key, String method) throws Exception {
Mac mac = Mac.getInstance(method);
SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
```

```
mac.init(secretKeySpec);
byte[] hash = mac.doFinal(s.getBytes(CHARSET));
return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
// When signing, the parameters need to be sorted in lexicographical order. TreeMap is used here to guarantee the correct order.
for (String k : params.keySet()) {
s2s.append(k).append("=").append(params.get(k).toString()).append("&");
}
return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
// There is no requirement for the order of the parameters in the actual request URL.
for (String k : params.keySet()) {
// The request string needs to be URL encoded. As the Key is all in English letters, only the value is URL encoded here.
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap enables automatic sorting
// A random number should be used when actually calling, for example: params.put("Nonce", new Random().nextInt(java.lang.Intege
r.MAX_VALUE));
params.put("Nonce", 11886); // Common parameter
// The current time of the system should be used when actually calling, for example: params.put("Timestamp", System.currentTimeM
illis() / 1000);
params.put("Timestamp", 1465185768); // Common parameter
params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"); // Common parameter
params.put("Action", "DescribeInstances"); // Common parameter
params.put("Version", "2017-03-12"); // Common parameter
params.put("Region", "ap-guangzhou"); // Common parameter
params.put("Limit", 20); // Business parameter
params.put("Offset", 0); // Business parameter
params.put("InstanceIds.0", "ins-09dx96dg"); // Business parameter
params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3*******", "HmacSHA1")); // Common parameter
System.out.println(getUrl(params));
}
}
```

Python

Note: If running in a Python 2 environment, the following requests dependency package must be installed first: `pip install requests` .

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import time

import requests

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

def get_string_to_sign(method, endpoint, params):
```

```
s = method + endpoint + "/?"
query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
return s + query_str

def sign_str(key, s, method):
hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
return base64.b64encode(hmac_str)

if __name__ == '__main__':
endpoint = "cvm.tencentcloudapi.com"
data = {
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# An actual invocation would occur here, which may incur fees after success
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

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

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

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

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

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

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

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

PHP

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

ksort($param);

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

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

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

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

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

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

DotNet

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

public class Application {
public static string Sign(string signKey, string secret)
{
string signRet = string.Empty;
using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
{
byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
signRet = Convert.ToBase64String(hash);
}
return signRet;
}
public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost,
string requestPath)
```

```
{
string retStr = "";
retStr += requestMethod;
retStr += requestHost;
retStr += requestPath;
retStr += "?";
string v = "";
foreach (string key in requestParams.Keys)
{
v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

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

string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768;
// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
param.Add("Offset", "0");
param.Add("InstanceIds.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

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

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


Console.WriteLine("GET https://cvm.tencentcloudapi.com");
foreach (KeyValuePair<string, string> kv in headers)
{
Console.WriteLine(kv.Key + ": " + kv.Value);
}
Console.WriteLine("Signature" + ": " + WebUtility.UrlEncode(sigOutParam));
Console.WriteLine();

string result = "https://cvm.tencentcloudapi.com/?";
foreach (KeyValuePair<string, string> kv in headers)
{
result += WebUtility.UrlEncode(kv.Key) + "=" + WebUtility.UrlEncode(kv.Value) + "&";
}
result += WebUtility.UrlEncode("Signature") + "=" + WebUtility.UrlEncode(sigOutParam);
Console.WriteLine("GET " + result);
```

```
}
}
```

NodeJS

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

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

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

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

function main(){
const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*******"
const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*******"

const endpoint = "cvm.tencentcloudapi.com"
const Region = "ap-guangzhou"
const Version = "2017-03-12"
const Action = "DescribeInstances"
const Timestamp = 1465185768
// const Timestamp = Math.round(Date.now() / 1000)
const Nonce = 11886
//const nonce = Math.round(Math.random() * 65535)

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

strParam = sort_params(params)

const reqMethod = "GET";
```

```
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
console.log(strSign)
console.log("-----------------------")

params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])
console.log("-----------------------")

const req_url = get_req_url(params, endpoint)
console.log(params['Signature'])
console.log("-----------------------")
console.log(req_url)
}
main()
```

Responses

Last updated：2020-02-18 19:10:15

Response for Successful Requests

For example, when calling CAM API (version: 2017-03-12) to view the status of instances (DescribeInstancesStatus), if the request has succeeded, you may see the response as shown below:

```
{
"Response": {
"TotalCount": 0,
"InstanceStatusSet": [],
"RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
}
}
```

- The API will return `Response`, which contains `RequestId`, as long as it processes the request. It does not matter if the request is successful or not.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.
- Except for the fixed fields, all fields are action-specified. For the definitions of action-specified fields, see the corresponding API documentation. In this example, `TotalCount` and `InstanceStatusSet` are the fields specified by the API `DescribeInstancesStatus`. 0 `TotalCount` means that the requester owns 0 CVM instance so the `InstanceStatusSet` is empty.

Response for Failed Requests

If the request has failed, you may see the response as shown below:

```
{
"Response": {
"Error": {
"Code": "AuthFailure.SignatureFailure",
"Message": "The provided credentials could not be validated. Please ensure your signature is correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- The presence of the `Error` field indicates that the request has failed. A response for a failed request will include `Error`, `Code` and `Message` fields.
- `Code` is the code of the error that helps you identify the cause and solution. There are two types of error codes so you may find the code in either common error codes or API-specified error codes.
- Message explains the cause of the error. Note that the returned messages are subject to service updates. The information the messages provide may not be up-to-date and should not be the only source of reference.
- RequestId is the unique ID of an API request. Contact us with this ID when an exception occurs.

Common Error Codes

If there is an Error field in the response, it means that the API call failed. The Code field in Error indicates the error code. The following table lists the common error codes that all actions can return.

Error Code	Description
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.MFAFailure	MFA failed.
AuthFailure.SecretIdNotFound	The key does not exist.
AuthFailure.SignatureExpire	Signature expired.
AuthFailure.SignatureFailure	Signature error.
AuthFailure.TokenFailure	Token error.
AuthFailure.UnauthorizedOperation	The request does not have CAM authorization.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
InternalError	Internal error.
InvalidAction	The API does not exist.
InvalidParameter	Incorrect parameter.
InvalidParameterValue	Invalid parameter value.
LimitExceeded	Quota limit exceeded.
MissingParameter	A parameter is missing.
NoSuchVersion	The API version does not exist.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceNotFound	The resource does not exist.
ResourceUnavailable	Resource is unavailable.
UnauthorizedOperation	Unauthorized operation.
UnknownParameter	Unknown parameter.
UnsupportedOperation	Unsupported operation.
UnsupportedProtocol	HTTPS request method error. Only GET and POST requests are supported.
UnsupportedRegion	API does not support the requested region.

Instance APIs
CreateInstance

Last updated : 2020-10-16 18:27:17

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to create an ES cluster instance with the specified specification.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer

Try it

API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: CreateInstance.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
Zone	Yes	String	Availability Zone
EsVersion	Yes	String	Instance version ("5.6.4", "6.4.3", "6.8.2", or "7.5.1")
VpcId	Yes	String	VPC ID
SubnetId	Yes	String	Subnet ID
Password	Yes	String	Access password, which must contain 8 to 16 characters, and include at least two of the following three types of characters: [a-z,A-Z], [0-9] and [-!@#$%&^*+=_:;,.?]
InstanceName	No	String	Instance name, which can contain 1 to 50 English letters, Chinese characters, digits, dashes (-), or underscores (_)
NodeNum	No	Integer	This parameter has been disused. Please use `NodeInfoList` Number of nodes (2-50)
ChargeType	No	String	Billing mode • POSTPAID_BY_HOUR: Pay-as-you-go hourly Default value: POSTPAID_BY_HOUR

ChargePeriod	No	Integer	This parameter is not used on the global website
RenewFlag	No	String	This parameter is not used on the global website
NodeType	No	String	This parameter has been disused. Please use `NodeInfoList` Node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
DiskType	No	String	This parameter has been disused. Please use `NodeInfoList` Node storage type • CLOUD_SSD: SSD cloud storage • CLOUD_PREMIUM: premium cloud storage Default value: CLOUD_SSD
DiskSize	No	Integer	This parameter has been disused. Please use `NodeInfoList` Node disk size in GB
TimeUnit	No	String	This parameter is not used on the global website
AutoVoucher	No	Integer	Whether to automatically use vouchers • 0: No • 1: Yes Default value: 0
VoucherIds.N	No	Array of String	List of voucher IDs (only one voucher can be specified at a time currently)
EnableDedicatedMaster	No	Boolean	This parameter has been disused. Please use `NodeInfoList` Whether to create a dedicated primary node • true: yes • false: no Default value: false
MasterNodeNum	No	Integer	This parameter has been disused. Please use `NodeInfoList` Number of dedicated primary nodes (only 3 and 5 are supported. This value must be passed in if `EnableDedicatedMaster` is `true`)
MasterNodeType	No	String	This parameter has been disused. Please use `NodeInfoList` Dedicated primary node type, which must be passed in if `EnableDedicatedMaster` is `true` • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
MasterNodeDiskSize	No	Integer	This parameter has been disused. Please use `NodeInfoList` Dedicated primary node disk size in GB, which is optional. If passed in, it can only be 50 and cannot be customized currently
ClusterNameInConf	No	String	ClusterName in the cluster configuration file, which is the instance ID by default and currently cannot be customized
DeployMode	No	Integer	Cluster deployment mode

			• 0: single-AZ deployment • 1: multi-AZ deployment Default value: 0
MultiZoneInfo.N	No	Array of ZoneDetail	Details of AZs in multi-AZ deployment mode (which is required when DeployMode is 1)
LicenseType	No	String	License type • oss: Open Source Edition • basic: Basic Edition • platinum: Platinum Edition Default value: Platinum
NodeInfoList.N	No	Array of NodeInfo	Node information list, which is used to describe the specification information of various types of nodes in the cluster, such as node type, node quantity, node specification, disk type, and disk size
TagList.N	No	Array of TagInfo	Node tag information list
BasicSecurityType	No	Integer	Whether to enable X-Pack security authentication in Basic Edition 6.8 (and above) • 1: disabled • 2: enabled
SceneType	No	Integer	Scenario template type. 0: not enabled; 1: general; 2: log; 3: search

3. Output Parameters

Parameter Name	Type	Description
InstanceId	String	Instance ID
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Creating an ES Cluster Instance

This example shows you how to create an ES cluster instance based on the input parameters

Input Example

```
https://es.tencentcloudapi.com/?Action=CreateInstance?
InstanceName=es_test
&EsVersion=6.4.3
&ChargeType=POSTPAID_BY_HOUR
&VpcId=vpc-xxxxxx
&SubnetId=subnet-xxxxxx
&Zone=ap-guangzhou-3
&Password=xxxxxx
&NodeInfoList.0.Type=hotData
&NodeInfoList.0.NodeNum=2
&NodeInfoList.0.NodeType=ES.S1.SMALL2
&NodeInfoList.0.DiskType=CLOUD_SSD
&NodeInfoList.0.DiskSize=100
```

```
&NodeInfoList.1.Type=dedicatedMaster
&NodeInfoList.1.NodeNum=3
&NodeInfoList.1.NodeType=ES.S1.SMALL2
&<common request parameters>
```

Output Example

```
{
"Response": {
"InstanceId": "es-xxxxxx",
"RequestId": "d7b76d5e-ad7d-4abd-b3b2-43b96dxxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.ClusterResourceLimitError	An error occurred with the cluster resource quota limit.
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.

ResourceInsufficient.Balance	Insufficient account balance.
ResourceInsufficient.Subnet	Insufficient number of remaining subnet IPs.

DeleteInstance

Last updated : 2020-08-14 09:39:08

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to terminate a cluster instance.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: DeleteInstance.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Deleting an ES cluster instance

This example shows you how to delete a cluster instance with the specified ID.

Input Example

```
https://es.tencentcloudapi.com/?Action=DeleteInstance
&InstanceId=es-xxxxxxxx
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "6eb5c3b2-0bba-4f73-bafb-bd21esxxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
UnsupportedOperation	Unsupported operation.

DescribeInstanceLogs

Last updated : 2020-07-31 10:14:10

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to query the eligible ES cluster logs in the current region.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer

Try it

API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: DescribeInstanceLogs.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Cluster instance ID
LogType	No	Integer	Log type. Default value: 1 • 1: primary log • 2: search slow log • 3: index slow log • 4: GC log
SearchKey	No	String	Search keyword, which supports LUCENE syntax, such as `level:WARN`, `ip:1.1.1.1`, and `message:test-index`
StartTime	No	String	Log start time in the format of YYYY-MM-DD HH:MM:SS, such as 2019-01-22 20:15:53
EndTime	No	String	Log end time in the format of YYYY-MM-DD HH:MM:SS, such as 2019-01-22 20:15:53
Offset	No	Integer	Pagination start value. Default value: 0
Limit	No	Integer	Number of entries per page. Default value: 100. Maximum value: 100
OrderByType	No	Integer	Time sorting order. Default value: 0 • 0: descending • 1: ascending

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Number of returned logs
InstanceLogList	Array of InstanceLog	Log details list
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying ES cluster logs

This example shows you how to query the latest master logs of a cluster.

Input Example

```
https://es.tencentcloudapi.com/?Action=DescribeInstanceLogs
&InstanceId=es-f5mwm28u
&<Common request parameters>
```

Output Example


```
{
"Response": {
"TotalCount": 71633,
"InstanceLogList": [
{
"Time": "2019-01-22T10:45:36.220+08:00",
"Ip": "10.0.128.65",
"Level": "INFO",
"Message": "[o.e.p.o.OPackActionFilter] [1547723102001286009] forbidden request: { ID:cdc62072721547678872c0448c1ecaf9, TYP:MainRequest, USR:null, BRS:false, ACT:cluster:monitor/main, OA:10.0.128.43, IDX:, MET:GET, PTH:/, CNT:<OMITTED, LENGTH=0>, HDR:content-length, EFF:0 } Reason: null"
},
{
"Time": "2019-01-22T10:45:35.730+08:00",
"Ip": "10.0.128.65",
"Level": "INFO",
"Message": "[o.e.p.o.OPackActionFilter] [1547723102001286009] forbidden request: { ID:1a8a5b7ea41a485ebdd769586c1dcdf6, TYP:MainRequest, USR:null, BRS:false, ACT:cluster:monitor/main, OA:10.0.128.73, IDX:, MET:GET, PTH:/, CNT:<OMITTED, LENGTH=0>, HDR:content-length, EFF:0 } Reason: null"
}
],
"RequestId": "783d9290-dc60-4862-9340-10b632605374"
}
}
```


5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.

DescribeInstanceOperations

Last updated : 2020-07-31 10:14:10

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to query the operation history of an instance by specified criteria.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: DescribeInstanceOperations.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Cluster instance ID
StartTime	Yes	String	Start time, such as "2019-03-07 16:30:39"
EndTime	Yes	String	End time, such as "2019-03-30 20:18:03"
Offset	Yes	Integer	Pagination start value
Limit	Yes	Integer	Number of entries per page

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Total number of operation records
Operations	Array of Operation	Operation history

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying the operation history of ES cluster instance

Input Example

```
https://es.tencentcloudapi.com/?Action=DescribeInstanceOperations
&InstanceId=es-f5mwm28u
&StartTime=2019-01-30 20:18:03
&EndTime=2019-03-30 20:18:03
&Offset=0
&Limit=30
&<Common request parameters>
```

Output Example

```
{
"Response": {
"TotalCount": 1,
"Operations": [
{
"Id": 6173,
"StartTime": "2019-03-07 16:30:39",
"Type": "CreateInstance",
"Detail": {
"OldInfo": [],
"NewInfo": []
},
"Result": "completed",
"Tasks": [
{
"Name": "prepareResource",
"Progress": 1,
"FinishTime": "2019-03-07 16:31:11",
"SubTasks": []
},
{
"Name": "deployESCluster",
"Progress": 1,
"FinishTime": "2019-03-07 16:34:32",
"SubTasks": []
},
{
"Name": "deployKibana",
"Progress": 1,
"FinishTime": "2019-03-07 16:35:13",
"SubTasks": []
},
{
"Name": "configLB",
"Progress": 1,
"FinishTime": "2019-03-07 16:35:15",
```

```
"SubTasks": []
}
],
"Progress": 1
}
],
"RequestId": "870dd618-b1ae-40cc-a5a9-22b867367ed7"
}
}
```


5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.

RestartNodes

Last updated : 2020-10-16 18:27:17

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to restart cluster nodes.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: RestartNodes.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Cluster instance ID
NodeNames.N	Yes	Array of String	Node name list
ForceRestart	No	Boolean	Whether to force restart

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Restarting cluster nodes

Input Example

```
https://es.tencentcloudapi.com/?Action=RestartNodes
&InstanceId=es-xxxxxxxx
&NodeNames.0=1592298977000 74xxxx
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.ErrorClusterState	Incorrect cluster status
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
UnsupportedOperation	Unsupported operation.

DescribeInstances

Last updated : 2020-07-31 10:14:10

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to query all eligible instances in the current region under the current account.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: DescribeInstances.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
Zone	No	String	AZ of the cluster instance. If this is not passed in, all AZs are used by default
InstanceIds.N	No	Array of String	List of cluster instance IDs
InstanceNames.N	No	Array of String	List of cluster instance names
Offset	No	Integer	Pagination start value. Default value: 0
Limit	No	Integer	Number of entries per page. Default value: 20
OrderByKey	No	Integer	Sort by field • 1: instance ID • 2: instance name • 3: AZ • 4: creation time If `orderKey` is not passed in, sort by creation time in descending order

Parameter Name	Required	Type	Description
OrderByType	No	Integer	Sorting order • 0: ascending • 1: descending If orderByKey is passed in but orderByType is not, ascending order is used by default
TagList.N	No	Array of TagInfo	Node tag information list
IpList.N	No	Array of String	VPC VIP list

3. Output Parameters

Parameter Name	Type	Description
TotalCount	Integer	Number of returned instances
InstanceList	Array of InstanceInfo	List of instance details
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Querying ES cluster instances

This example shows you how to query eligible ES cluster instances according to the given criteria and return their details.

Input Example

```
https://es.tencentcloudapi.com/?Action=DescribeInstances
&<Common request parameters>
```

Output Example


```
{
"Response": {
"TotalCount": 2,
"InstanceList": [
{
"InstanceId": "es-sample",
"InstanceName": "es-sample",
"InstanceType": 2,
"Region": "ap-guangzhou",
"Zone": "ap-guangzhou-2",
"AppId": 0,
"Uin": "xxxxxxxx",
"VpcUid": "vpc-sample",
"SubnetUid": "subnet-sample",
"Status": 1,
"ChargeType": "PREPAID",
```



```
"ChargePeriod": 1,
"RenewFlag": "RENEW_FLAG_DEFAULT",
"NodeType": "ES.S1.SMALL2",
"NodeNum": 2,
"CpuNum": 1,
"MemSize": 2,
"DiskType": "",
"DiskSize": 100,
"EsDomain": "es-sample.tencentelasticsearch.com",
"EsVip": "0.0.0.0",
"EsPort": 9200,
"KibanaUrl": "https://es-sample.kibana.tencentelasticsearch.com:5601",
"EsVersion": "5.6.4",
"EsConfig": "{}",
"EsAcl": {
"WhiteIpList": [],
"BlackIpList": []
},
"CreateTime": "2018-07-27 17:28:04",
"UpdateTime": "2018-07-30 10:22:29",
"Deadline": "2018-08-27 17:28:04"
},
{
"InstanceId": "es-sample2",
"InstanceName": "es-sample2",
"InstanceType": 2,
"Region": "ap-guangzhou",
"Zone": "ap-guangzhou-4",
"AppId": 0,
"Uin": "xxxxxx",
"VpcUid": "vpc-sample",
"SubnetUid": "subnet-sample",
"Status": 1,
"ChargeType": "PREPAID",
"ChargePeriod": 1,
"RenewFlag": "RENEW_FLAG_DEFAULT",
"NodeType": "ES.S1.MEDIUM4",
"NodeNum": 2,
"CpuNum": 2,
"MemSize": 4,
"DiskType": "",
"DiskSize": 100,
"EsDomain": "es-sample.tencentelasticsearch.com",
"EsVip": "0.0.0.0",
"EsPort": 9200,
"KibanaUrl": "https://es-sample.kibana.tencentelasticsearch.com:5601",
"EsVersion": "5.6.4",
"EsConfig": "{}",
"EsAcl": {
"WhiteIpList": [],
"BlackIpList": []
},
"CreateTime": "2018-07-26 17:47:47",
"UpdateTime": "2018-07-26 18:16:50",
"Deadline": "2018-08-26 17:47:47"
}
],
"RequestId": "5d5a201f-0a3d-485f-a82f-3c73ccxxxxxx"
}
}
```


5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.

RestartInstance

Last updated : 2020-07-31 10:14:10

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to restart an ES cluster instance (for operations such as system update).

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: RestartInstance.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID
ForceRestart	No	Boolean	Whether to force restart • true: Yes • false: No Default value: false

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Restarting an ES cluster instance

This example shows you how to perform operations such as version update.

Input Example

```
https://es.tencentcloudapi.com/?Action=RestartInstance
&InstanceId=es-xxxxxxxx
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "7f0e35a7-e03b-42cd-aa4f-0c1471xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.

UpdatePlugins

Last updated : 2020-08-14 09:39:08

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to change the list of plugins.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer

Try it

API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: UpdatePlugins.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID
InstallPluginList.N	No	Array of String	List of names of the plugins to be installed
RemovePluginList.N	No	Array of String	List of names of the plugins to be uninstalled
ForceRestart	No	Boolean	Whether to force restart

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Uninstalling plugins in batches

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdatePlugins
&InstanceId=es-xxxxxxxx
&RemovePluginList.0=analysis-qq
&RemovePluginList.1=sql
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

Example2 Installing plugins in batches

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdatePlugins
&InstanceId=es-xxxxxxxx
&InstallPluginList.0=analysis-qq
&InstallPluginList.1=sql
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.ErrorClusterState	Incorrect cluster status
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceInsufficient.Balance	Insufficient account balance.
UnsupportedOperation	Unsupported operation.

UpdateInstance

Last updated : 2020-10-16 18:27:17

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used for operations such as modifying node specification, renaming an instance, modifying configuration, resetting password, and setting Kibana blocklist/allowlist. `InstanceId` is required, while `ForceRestart` is optional. Other parameters or parameter combinations and their meanings are as follows:

- InstanceName: renames an instance (only for instance identification)
- NodeInfoList: modifies node configuration (horizontally scaling nodes, vertically scaling nodes, adding primary nodes, adding cold nodes, etc.)
- EsConfig: modifies cluster configuration
- Password: changes the password of the default user "elastic"
- EsAcl: modifies the ACL
- CosBackUp: sets auto-backup to COS for a cluster
  Only one of the parameters or parameter combinations above can be passed in at a time, while passing fewer or more ones will cause the request to fail.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: UpdateInstance.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID
InstanceName	No	String	Instance name, which can contain 1 to 50 English letters, Chinese characters, digits, dashes (-), or underscores (_)
NodeNum	No	Integer	This parameter has been disused. Please use `NodeInfoList` Number of nodes (2-50)
EsConfig	No	String	Configuration item (JSON string)

Password	No	String	Password of the default user 'elastic', which must contain 8 to 16 characters, including at least two of the following three types of characters: [a-z,A-Z], [0-9] and [-!@#$%&^*+=_:;,.?]
EsAcl	No	EsAcl	Access control list
DiskSize	No	Integer	This parameter has been disused. Please use `NodeInfoList` Disk size in GB
NodeType	No	String	This parameter has been disused. Please use `NodeInfoList` Node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
MasterNodeNum	No	Integer	This parameter has been disused. Please use `NodeInfoList` Number of dedicated primary nodes (only 3 and 5 are supported)
MasterNodeType	No	String	This parameter has been disused. Please use `NodeInfoList` Dedicated primary node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
MasterNodeDiskSize	No	Integer	This parameter has been disused. Please use `NodeInfoList` Dedicated primary node disk size in GB. This is 50 GB by default and currently cannot be customized
ForceRestart	No	Boolean	Whether to force restart during configuration update • true: Yes • false: No This needs to be set only for EsConfig. Default value: false
CosBackup	No	CosBackup	Auto-backup to COS
NodeInfoList.N	No	Array of NodeInfo	Node information list. You can pass in only the nodes to be updated and their corresponding specification information. Supported operations include: • modifying the number of nodes in the same type • modifying the specification and disk size of nodes in the same type • adding a node type (you must also specify the node type, quantity, specification, disk, etc.) The above operations can only be performed one at a time, and the disk type cannot be modified
PublicAccess	No	String	Public network access status
EsPublicAcl	No	EsPublicAcl	Public network ACL
KibanaPublicAccess	No	String	Public network access status of Kibana
KibanaPrivateAccess	No	String	Private network access status of Kibana
BasicSecurityType	No	Integer	Enables or disables user authentication for ES Basic Edition v6.8 and above

KibanaPrivatePort	No	Integer	Kibana private port
ScaleType	No	Integer	0: scaling in blue/green deployment mode without cluster restart (default); 1: scaling by unmounting disk with rolling cluster restart
MultiZoneInfo.N	No	Array of ZoneDetail	Multi-AZ deployment
SceneType	No	Integer	Scenario template type. -1: not enabled; 1: general; 2: log; 3: search

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Resetting Kibana password

This example shows you how to reset the Kibana password of a specified ES cluster instance.

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxxxx
&Password=newPwd_123
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "1b72089e-720f-4f95-a4ae-4da461xxxxxx"
}
}
```

Example2 Modifying the configuration of ES cluster instance

This example shows you how to modify the configuration of a specified ES cluster instance.

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxxxx
&EsConfig={"action.destructive_requires_name":"true"}
&<Common request parameters>
```

Output Example

```
{
"Response": {
```

```
"RequestId": "e7c1bb22-e5f2-42f1-8a12-a97a6dxxxxxx"
}
}
```

Example3 Vertically scaling ES cluster

This example shows you how to vertically scale the node specification (number of cores and memory size) and disk size in a cluster (only vertical scaling is supported currently).

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxxxx
&NodeType=ES.S1.MEDIUM4
&DiskSize=150
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "dd3f624d-9a72-4057-85cb-f5d32exxxxxx"
}
}
```

Example4 Setting auto-backup to COS for ES

This example shows you how to reset the Kibana password of the specified ES cluster instance.

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxxxx
&CosBackup.IsAutoBackup=true
&CosBackup.BackupTime=23:00
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "dd3f624d-9a72-4057-85cb-f5d32exxxxxx"
}
}
```

Example5 Modifying the node specification of ES cluster

This example shows you how to perform operations such as horizontally or vertically scaling the specified ES cluster instance or dedicated master node, where `NodeInfoList` contains the information of all target nodes to be passed in.

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxx
&NodeInfoList.0.Type=hotData
&NodeInfoList.0.NodeNum=2
&NodeInfoList.0.NodeType=ES.S1.SMALL2
&NodeInfoList.0.DiskType=CLOUD_SSD
```

```
&NodeInfoList.0.DiskSize=100
&NodeInfoList.1.Type=dedicatedMaster
&NodeInfoList.1.NodeNum=3
&NodeInfoList.1.NodeType=ES.S1.SMALL2
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "6001962a-17c5-4604-a0af-0d4719xxxxxx"
}
}
```

Example6 Renaming ES cluster instances

This example shows you how to rename a specified ES cluster instance.

Input Example

```
https://es.tencentcloudapi.com/?Action=UpdateInstance
&InstanceId=es-xxxxxx
&InstanceName=newName
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.ClusterResourceLimitError	An error occurred with the cluster resource quota limit.
FailedOperation.ErrorClusterState	Incorrect cluster status
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceInsufficient.Balance	Insufficient account balance.
ResourceInsufficient.Subnet	Insufficient number of remaining subnet IPs.
UnsupportedOperation	Unsupported operation.

UpgradeInstance

Last updated：2020-10-16 18:27:17

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to upgrade ES cluster version

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer

Try it

API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: UpgradeInstance.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID
EsVersion	Yes	String	Target ES version. Valid values: 6.4.3, 6.8.2, 7.5.1
CheckOnly	No	Boolean	Whether to check for upgrade only. Default value: false
LicenseType	No	String	Target X-Pack edition: • OSS: Open-source Edition • basic: Basic Edition Currently only used for v5.6.4 to v6.x upgrade. Default value: basic
BasicSecurityType	No	Integer	Whether to enable X-Pack security authentication in Basic Edition 6.8 (and above) • 1: disabled • 2: enabled
UpgradeMode	No	String	Upgrade mode. • scale: blue/green deployment • restart: rolling restart Default value: scale

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Upgrading ES Cluster Version

Input Example

```
https://es.tencentcloudapi.com/?Action=UpgradeInstance
&InstanceId=es-xxxxxxxx
&EsVersion=6.4.3
&<common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

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

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.ErrorClusterState	Incorrect cluster status
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceInsufficient.Balance	Insufficient account balance.
ResourceInsufficient.Subnet	Insufficient number of remaining subnet IPs.
UnsupportedOperation	Unsupported operation.

UpgradeLicense

Last updated : 2020-07-31 10:14:09

1. API Description

Domain name for API request: es.tencentcloudapi.com.

This API is used to upgrade ES X-Pack.

A maximum of 20 requests can be initiated per second for this API.

Note: This API supports Finance regions. If the common parameter Region is a Finance region, a domain name with the Finance region needs to be specified, for example: es.ap-shanghai-fsi.tencentcloudapi.com

We recommend you to use API Explorer
Try it
API Explorer provides a range of capabilities, including online call, signature authentication, SDK code generation, and API quick search. It enables you to view the request, response, and auto-generated examples.

2. Input Parameters

The following request parameter list only provides API request parameters and some common parameters. For the complete common parameter list, see Common Request Parameters.

Parameter Name	Required	Type	Description
Action	Yes	String	Common parameter. The value used for this API: UpgradeLicense.
Version	Yes	String	Common parameter. The value used for this API: 2018-04-16.
Region	Yes	String	Common parameter. For more information, please see the list of regions supported by the product.
InstanceId	Yes	String	Instance ID
LicenseType	Yes	String	License type • oss: Open Source Edition • basic: Basic Edition • platinum: Platinum Edition Default value: Platinum
AutoVoucher	No	Integer	Whether to automatically use vouchers • 0: No • 1: Yes Default value: 0
VoucherIds.N	No	Array of String	List of voucher IDs (only one voucher can be specified at a time currently)
BasicSecurityType	No	Integer	Whether to enable X-Pack security authentication in Basic Edition 6.8 (and above) • 1: disabled • 2: enabled

Parameter Name	Required	Type	Description
ForceRestart	No	Boolean	Whether to force restart • true: yes • false: no Default value: false

3. Output Parameters

Parameter Name	Type	Description
RequestId	String	The unique request ID, which is returned for each request. RequestId is required for locating a problem.

4. Example

Example1 Upgrading ES X-Pack

Input Example

```
https://es.tencentcloudapi.com/?Action=UpgradeLicense
&InstanceId=es-xxxxxxxx
&LicenseType=platinum
&<Common request parameters>
```

Output Example

```
{
"Response": {
"RequestId": "c96a110c-7493-452d-a99b-683d07xxxxxx"
}
}
```

5. Developer Resources

API Explorer

This tool allows online call, signature authentication, SDK code generation and quick search of APIs to greatly improve the efficiency of using TencentCloud APIs.

- API 3.0 Explorer

SDK

TencentCloud API 3.0 integrates SDKs that support various programming languages to make it easier for you to call APIs.

- Tencent Cloud SDK 3.0 for Python
- Tencent Cloud SDK 3.0 for Java
- Tencent Cloud SDK 3.0 for PHP
- Tencent Cloud SDK 3.0 for Go

- Tencent Cloud SDK 3.0 for NodeJS
- Tencent Cloud SDK 3.0 for .NET

Command Line Interface

- Tencent Cloud CLI 3.0

6. Error Code

The following only lists the error codes related to the API business logic. For other error codes, see Common Error Codes.

Error Code	Description
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
InternalError	Internal error.
InvalidParameter	Invalid parameter.
ResourceInUse	Resource is in use.
ResourceInsufficient	Insufficient resource.
ResourceInsufficient.Balance	Insufficient account balance.
UnsupportedOperation	Unsupported operation.

Data Types

Last updated : 2020-10-16 18:27:18

CosBackup

Auto-backup to COS for ES

Used by actions: DescribeInstances, UpdateInstance.

Name	Type	Required	Description
IsAutoBackup	Boolean	Yes	Whether to enable auto-backup to COS
BackupTime	String	Yes	Auto-backup time (accurate down to the hour), such as "22:00"

DictInfo

Information of the IK plugin dictionary

Used by actions: DescribeInstances.

Name	Type	Description
Key	String	Dictionary key value
Name	String	Dictionary name
Size	Integer	Dictionary size in B

EsAcl

ES cluster configuration item

Used by actions: DescribeInstances, UpdateInstance.

Name	Type	Required	Description
BlackIpList	Array of String	No	Kibana access blocklist
WhiteIpList	Array of String	No	Kibana access allowlist

EsDictionaryInfo

ES dictionary information

Used by actions: DescribeInstances.

Name	Type	Description
MainDict	Array of DictInfo	List of non-stop words
Stopwords	Array of DictInfo	List of stop words

QQDict	Array of DictInfo	QQ dictionary list
Synonym	Array of DictInfo	Synonym dictionary list
UpdateType	String	Update dictionary type

EsPublicAcl

Public network ACL information of ES

Used by actions: UpdateInstance.

Name	Type	Required	Description
BlackIpList	Array of String	No	Access blocklist
WhiteIpList	Array of String	No	Access allowlist

InstanceInfo

Instance details

Used by actions: DescribeInstances.

Name	Type	Description
InstanceId	String	Instance ID
InstanceName	String	Instance name
Region	String	Region
Zone	String	Availability Zone
AppId	Integer	User ID
Uin	String	User UIN
VpcUid	String	UID of the VPC where the instance resides
SubnetUid	String	UID of the subnet where the instance resides
Status	Integer	Instance status. 0: processing; 1: normal; -1: stopped; -2: terminating; -3: terminated
ChargeType	String	Instance billing method. Valid values: POSTPAID_BY_HOUR (pay-as-you-go hourly); CDHPAID (billed based on CDH, i.e., only CDH is billed but not the instances on CDH)
ChargePeriod	Integer	This parameter is not used on the global website
RenewFlag	String	This parameter is not used on the global website
NodeType	String	Node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB

		• ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
NodeNum	Integer	Number of nodes
CpuNum	Integer	Number of CPU cores of the node
MemSize	Integer	Node memory size in GB
DiskType	String	Node disk type
DiskSize	Integer	Node disk size in GB
EsDomain	String	ES domain name
EsVip	String	ES VIP
EsPort	Integer	ES port
KibanaUrl	String	Kibana access URL
EsVersion	String	ES version number
EsConfig	String	ES configuration item
EsAcl	EsAcl	Kibana access control configuration
CreateTime	String	Instance creation time
UpdateTime	String	Last modified time of the instance
Deadline	String	This parameter is not used on the global website
InstanceType	Integer	Instance type (instance type identifier, which can be only 1 or 2 currently)
IkConfig	EsDictionaryInfo	IK analyzer configuration
MasterNodeInfo	MasterNodeInfo	Dedicated primary node configuration
CosBackup	CosBackup	Auto-backup to COS configuration
AllowCosBackup	Boolean	Whether to allow auto-backup to COS
TagList	Array of TagInfo	List of tags owned by the instance
LicenseType	String	License type • oss: Open Source Edition • basic: Basic Edition • platinum: Platinum Edition Default value: Platinum
EnableHotWarmMode	Boolean	Whether it is a hot/warm cluster • true: yes • false: no Note: this field may return null, indicating that no valid values can be obtained.
WarmNodeType	String	Warm node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB

		• ES.S1.4XLARGE64: 16-core 64 GB Note: This field may return null, indicating that no valid values can be obtained.
WarmNodeNum	Integer	Number of warm nodes Note: This field may return null, indicating that no valid values can be obtained.
WarmCpuNum	Integer	Number of warm node CPU cores Note: This field may return null, indicating that no valid values can be obtained.
WarmMemSize	Integer	Warm node memory size in GB Note: This field may return null, indicating that no valid values can be obtained.
WarmDiskType	String	Warm node disk type Note: This field may return null, indicating that no valid values can be obtained.
WarmDiskSize	Integer	Warm node disk size in GB Note: This field may return null, indicating that no valid values can be obtained.
NodeInfoList	Array of NodeInfo	Cluster node information list Note: This field may return null, indicating that no valid values can be obtained.
EsPublicUrl	String	ES public IP address Note: This field may return null, indicating that no valid values can be obtained.
MultiZoneInfo	Array of ZoneDetail	Multi-AZ network information Note: This field may return null, indicating that no valid values can be obtained.
DeployMode	Integer	Deployment mode • 0: single-AZ • 1: multi-AZ Note: This field may return null, indicating that no valid values can be obtained.
PublicAccess	String	ES public access status • OPEN: enabled • CLOSE: disabled Note: This field may return null, indicating that no valid values can be obtained.
EsPublicAcl	EsAcl	ES public access control configuration
KibanaPrivateUrl	String	Kibana private IP address Note: This field may return null, indicating that no valid values can be obtained.
KibanaPublicAccess	String	Kibana public access status • OPEN: enabled • CLOSE: disabled Note: This field may return null, indicating that no valid values can be obtained.
KibanaPrivateAccess	String	Kibana private access status • OPEN: enabled • CLOSE: disabled Note: This field may return null, indicating that no valid values can be obtained.
SecurityType	Integer	Whether to enable X-Pack security authentication in Basic Edition 6.8 (and above) • 1: disabled • 2: enabled Note: This field may return null, indicating that no valid values can be obtained.
SceneType	Integer	Scenario template type. 0: not enabled; 1: general scenario; 2: log scenario; 3: search scenario

		Note: this field may return null, indicating that no valid values can be obtained.

InstanceLog

ES cluster log details

Used by actions: DescribeInstanceLogs.

Name	Type	Description
Time	String	Log time
Level	String	Log level
Ip	String	Cluster node IP
Message	String	Log content

KeyValue

`OperationDetail` uses an array of this structure to describe the old and new configuration information

Used by actions: DescribeInstanceOperations.

Name	Type	Description
Key	String	Key
Value	String	Value

LocalDiskInfo

Local disk information of node

Used by actions: CreateInstance, DescribeInstances, UpdateInstance.

Name	Type	Description
LocalDiskType	String	Local disk type • LOCAL_SATA: big data • NVME_SSD: high IO
LocalDiskSize	Integer	Size of a single local disk
LocalDiskCount	Integer	Number of local disks

MasterNodeInfo

Information of the dedicated primary node in an instance

Used by actions: DescribeInstances.

Name	Type	Description
EnableDedicatedMaster	Boolean	Whether to enable the dedicated primary node

MasterNodeType	String	Dedicated primary node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
MasterNodeNum	Integer	Number of dedicated primary nodes
MasterNodeCpuNum	Integer	Number of CPU cores of the dedicated primary node
MasterNodeMemSize	Integer	Memory size of the dedicated primary node in GB
MasterNodeDiskSize	Integer	Disk size of the dedicated primary node in GB
MasterNodeDiskType	String	Disk type of the dedicated primary node

NodeInfo

Specification information of a node type in the cluster (such as hot data node, warm data node, or dedicated primary node), including node type, number of nodes, node specification, disk type, and disk size. If `Type` is not specified, it will be a hot data node by default; if the node is a primary node, then the `DiskType` and `DiskSize` parameters will be ignored (as a primary node has no data disks)

Used by actions: CreateInstance, DescribeInstances, UpdateInstance.

Name	Type	Required	Description
NodeNum	Integer	Yes	Number of nodes
NodeType	String	Yes	Node specification • ES.S1.SMALL2: 1-core 2 GB • ES.S1.MEDIUM4: 2-core 4 GB • ES.S1.MEDIUM8: 2-core 8 GB • ES.S1.LARGE16: 4-core 16 GB • ES.S1.2XLARGE32: 8-core 32 GB • ES.S1.4XLARGE32: 16-core 32 GB • ES.S1.4XLARGE64: 16-core 64 GB
Type	String	No	Node type • hotData: hot data node • warmData: warm data node • dedicatedMaster: dedicated primary node Default value: hotData
DiskType	String	No	Node disk type • CLOUD_SSD: SSD cloud storage • CLOUD_PREMIUM: Premium cloud disk Default value: CLOUD_SSD
DiskSize	Integer	No	Node disk size in GB
LocalDiskInfo	LocalDiskInfo	No	Local disk information Note: this field may return null, indicating that no valid values can be obtained.
DiskCount	Integer	No	Number of node disks

DiskEncrypt	Integer	No	Whether to encrypt node disk. 0: no (default); 1: yes.

Operation

ES cluster operation details

Used by actions: DescribeInstanceOperations.

Name	Type	Description
Id	Integer	Unique operation ID
StartTime	String	Operation start time
Type	String	Operation type
Detail	OperationDetail	Operation details
Result	String	Operation result
Tasks	Array of TaskDetail	Workflow task information
Progress	Float	Operation progress

OperationDetail

Operation details

Used by actions: DescribeInstanceOperations.

Name	Type	Description
OldInfo	Array of KeyValue	Original instance configuration information
NewInfo	Array of KeyValue	Updated instance configuration information

SubTaskDetail

Information of subtask in workflow task in the instance operation history (such as each check item in a upgrade check task)

Used by actions: DescribeInstanceOperations.

Name	Type	Description
Name	String	Subtask name
Result	Boolean	Subtask result
ErrMsg	String	Subtask error message
Type	String	Subtask type
Status	Integer	Subtask status. 0: processing, 1: succeeded, -1: failed
FailedIndices	Array of String	Name of the index for which the check for upgrade failed
FinishTime	String	Subtask end time

Level	Integer	Subtask level. 1: warning, 2: failed

TagInfo

Instance tag information

Used by actions: CreateInstance, DescribeInstances.

Name	Type	Description
TagKey	String	Tag key
TagValue	String	Tag value

TaskDetail

Information of workflow task in instance operation history

Used by actions: DescribeInstanceOperations.

Name	Type	Description
Name	String	Task name
Progress	Float	Task progress
FinishTime	String	Task completion time
SubTasks	Array of SubTaskDetail	Subtask

ZoneDetail

Details of AZs in multi-AZ deployment mode

Used by actions: CreateInstance, DescribeInstances, UpdateInstance.

Name	Type	Required	Description
Zone	String	Yes	AZ
SubnetId	String	Yes	Subnet ID

Error Codes

Last updated : 2020-10-16 18:27:18

Feature Description

If there is an Error field in the response, it means that the API call failed. For example:

```
{
"Response": {
"Error": {
"Code": "AuthFailure.SignatureFailure",
"Message": "The provided credentials could not be validated. Please check your signature is correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

Code in Error indicates the error code, and Message indicates the specific information of the error.

Error Code List

Common Error Codes

Error Code	Description
UnsupportedOperation	Unsupported operation.
ResourceInUse	Resource is in use.
InternalError	Internal error.
RequestLimitExceeded	The number of requests exceeds the frequency limit.
AuthFailure.SecretIdNotFound	Key does not exist. Check if the key has been deleted or disabled in the console, and if not, check if the key is correctly entered. Note that whitespaces should not exist before or after the key.
LimitExceeded	Quota limit exceeded.
NoSuchVersion	The API version does not exist.
ResourceNotFound	The resource does not exist.
AuthFailure.SignatureFailure	Invalid signature. Signature calculation error. Please ensure you've followed the signature calculation process described in the Signature API documentation.
AuthFailure.SignatureExpire	Signature expired. Timestamp and server time cannot differ by more than five minutes. Please ensure your current local time matches the standard time.
UnsupportedRegion	API does not support the requested region.
UnauthorizedOperation	Unauthorized operation.
InvalidParameter	Incorrect parameter.
ResourceUnavailable	Resource is unavailable.
AuthFailure.MFAFailure	MFA failed.

AuthFailure.UnauthorizedOperation	The request is not authorized. For more information, see the CAM documentation.
AuthFailure.InvalidSecretId	Invalid key (not a TencentCloud API key type).
AuthFailure.TokenFailure	Token error.
DryRunOperation	DryRun Operation. It means that the request would have succeeded, but the DryRun parameter was used.
FailedOperation	Operation failed.
UnknownParameter	Unknown parameter.
UnsupportedProtocol	HTTP(S) request protocol error; only GET and POST requests are supported.
InvalidParameterValue	Invalid parameter value.
InvalidAction	The API does not exist.
MissingParameter	A parameter is missing.
ResourceInsufficient	Insufficient resource.

Service Error Codes

Error Code	Description
FailedOperation.ClusterResourceLimitError	An error occurred with the cluster resource quota limit.
FailedOperation.ErrorClusterState	Incorrect cluster status
FailedOperation.NoPayment	No credit card or PayPal account is linked to the current account. Unable to make a payment.
ResourceInsufficient.Balance	Insufficient account balance.
ResourceInsufficient.Subnet	Insufficient number of remaining subnet IPs.