

Cloud Bastion Host (CBH)

API Reference

Issue	01
Date	2025-12-10



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

Contents

1 Before You Start.....

2 Calling APIs.....

2.1 Making an API Request.....

2.2 Authentication.....

2.3 Response.....

3 API Description.....

3.1 Querying CBH Details.....

3.1.1 Obtaining the CBH Instance List.....

A Appendixes.....

A.1 Status Codes.....

A.2 Error Codes.....

A.3 Obtaining a Project ID.....

1

3

3

5

7

9

9

9

17

17

18

18

1 Before You Start

Overview

Cloud Bastion Host (CBH) is a unified security management and control platform. It provides account, authorization, authentication, and audit management services.

API Calling

CBH supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For more information, see [Calling APIs](#).

Endpoints

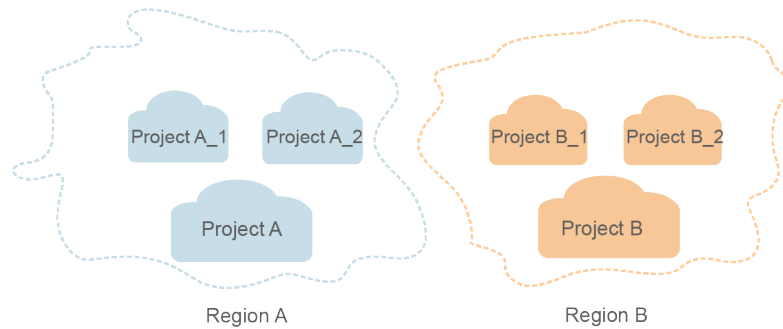
An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see [Regions and Endpoints](#).

Basic Concepts

- **Region**
Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common users. A dedicated region provides services of the same type only or for specific users.
- **Availability Zone (AZ)**
An AZ comprises one or multiple physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Compute, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build highly available systems across AZs.
- **Project**
A project corresponds to a region. Projects group and isolate resources (including compute, storage, and network resources) across physical regions. Users can be granted permissions in a default project to access all resources in

the region associated with the project. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

Figure 1-1 Project isolation model



API Versions

CBH provides V1 and V2 APIs for you.

APIs of V2 are recommended. APIs of V1 are no longer maintained.

2 Calling APIs

2.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

- **URI-scheme:**
Protocol used to transmit requests. All APIs use HTTPS.
- **Endpoint:**
Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from [Regions and Endpoints](#).
- **resource-path:**
Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.
- **query-string:**
Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "Parameter name=Parameter value". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

- **GET**: requests the server to return specified resources.
- **PUT**: requests the server to update specified resources.
- **POST**: requests the server to add resources or perform special operations.
- **DELETE**: requests the server to delete specified resources, for example, an object.
- **HEAD**: same as GET except that the server must return only the response header.
- **PATCH**: requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to obtain a user token, the request method is POST. The request is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. The user token is a response to the API used to obtain a user token. This API is the only one that does not require authentication.

NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see [AK/SK-based Authentication](#).

The API used to obtain a user token does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
```

Request Body

The body of a request is often sent in a structured format as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Set **username** to the name of a user, **domainname** to the name of the account that the user belongs to, ********* to the user's login password, and **xxxxxxxxxxxxxxxxxxxx** to the project name. You can learn more information about projects from [Regions and Endpoints](#).

NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see "Obtaining a User Token".

```
POST https://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

2.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. This method is recommended because it provides higher security than token-based authentication.

Token-based Authentication

NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API.

The token can be obtained by calling the required API. For more information, see "Obtaining a User Token". A project-level token is required for calling this API, that is, **auth.scope** must be set to **project** in the request body. Example:

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    }
  },
  "scope": {
    "project": {
      "name": "xxxxxxx"
    }
  }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFGH....**, add **X-Auth-Token: ABCDEFGH....** to a request as follows:

```
GET https://{{endpoint}}/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFGH....
```

AK/SK-based Authentication

NOTE

AK/SK-based authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token-based authentication is recommended.

In AK/SK-based authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK-based authentication, you can use an AK/SK to sign requests based on the signature algorithm or use the signing SDK to sign requests. For details about how to sign requests and use the signing SDK, see [API Signature Guide](#).

NOTICE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

2.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Status Codes](#).

For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

A response header corresponds to a request header, for example, **Content-Type**.

Figure 2-1 shows the response header for the API of obtaining a user token, in which **x-subject-token** is the desired user token. Then, you can use the token to authenticate the calling of other APIs.

Figure 2-1 Header of the response to the request for obtaining a user token

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → MIIYXQYJKoZIhvcNAQcCoIIYJCCEoCAQExDTALBgIghkgB8ZQMCAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iO3nsiZXhwaXJlc19hdCI6ijlwMTktMDItMTNUMC
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JlGzrpd18LGXK5tdfdq4lqHCYb8P4NaY0NYejcAgz/VeFYtLWT1GSO0zxKZmlQHqJ82HBqHdglZO9fuEbL5dMhdavj+33wEI
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FrucL6jaglfKkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxjECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

(Optional) Response Body

A response body is generally returned in a structured format, corresponding to the **Content-Type** in the response header, and is used to transfer content other than the response header.

The following shows part of the response body for the API to obtain a user token. For the sake of space, only part of the content is displayed here.

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "xxxxxxx",
            .....

```

If an error occurs during API calling, the system returns an error code and a message to you. The following shows the format of an error response body:

```
{
  "error": {
    "message": "The request you have made requires authentication.",
    "title": "Unauthorized"
  }
}
```

In the preceding information, **error_code** is an error code, and **error_msg** describes the error.

3 API Description

3.1 Querying CBH Details

3.1.1 Obtaining the CBH Instance List

Function

This API is used to obtain the CBH instance list of the current account.

URI

GET /v1/{project_id}/cbs/instance/list

Table 3-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see [Obtaining a Project ID].

Request Parameters

None

Response Parameters

Status code: 200

Table 3-2 Response body parameters

Parameter	Type	Description
total	Integer	The total number of CBH instances.
quotaDetail	QuotaDetail object	The CBH quota of the current user. The default value null is returned.
instance	Array of InstanceDetail objects	CBH instance list.

Table 3-3 QuotaDetail

Parameter	Type	Description
zh_cn	String	Quota description.
en_us	String	Quota description.
remaining	Integer	Remaining quota.

Table 3-4 InstanceDetail

Parameter	Type	Description
publicip	String	EIP of the CBH instance. The default value null is returned.
expTime	String	Expiration time of the CBH instance.
startTime	String	Start time of the CBH instance, in timestamp format.
endTime	String	End time of the CBH instance, in timestamp format.
releaseTime	String	Release time of the CBH instance, in timestamp format.
name	String	CBH instance name.
instanceId	String	CBH instance ID, in UUID format.
privateIp	String	Private IP address of the CBH instance.

Parameter	Type	Description
taskStatus	String	<p>Task status of the CBH instance.</p> <ul style="list-style-type: none"> • powering-on: Started. • powering-off: Stopped. • rebooting: The system is being restarted. • delete_wait: Delete • frozen: The instance is frozen. • NO_TASK: Running • unfrozen: Unfrozen • alter: Change • updating: Being updated. • configuring-ha: Configuring HA.
status	String	<p>CBH instance status.</p> <ul style="list-style-type: none"> • SHUTOFF: The instance is shut down. • ACTIVE: The instance is running. • DELETING: The instance is being deleted. • BUILD: The instance is being created. • DELETED: The instance was deleted. • ERROR: Faulty • HAWAIT: The standby instance is being created. • FROZEN: The instance has been frozen. • UPGRADING: The instance is being upgraded. • UNPAID: Pending payment. • RESIZE: The instance specifications are being changed. • DILATATION: The instance capacity is being expanded. • HA: HA is being configured.
created	String	Time when the CBH instance was created, in UTC format.
region	String	Site where the CBH instance was located.

Parameter	Type	Description
zone	String	Region where the CBH instance is located.
availability_zone_display	String	Name of the region where the CBH instance is located.
vpcId	String	ID of the VPC where the CBH instance is located.
subnetId	String	ID of the subnet where the CBH instance is located.
securityGroupId	String	ID of the security group to which the CBH instance belongs.
specification	String	CBH instance specifications.
update	String	Whether the CBH instance can be upgraded. <ul style="list-style-type: none"> • NEW: The upgrade can be performed. • OLD: The upgrade cannot be performed.
createinstanceStatus	String	Status of a CBH instance during instance creation. <ul style="list-style-type: none"> • Waiting for payment: Waiting for payment • creating-network: Creating a network. • creating-server: Creating the service. • tranfering-horizontal-network: Establishing network connections. • adding-policy-route: Adding a routing policy. • configing-dns: Configuring DNS. • starting-cbs-service: The service is running. • setting-init-conf: Initializing • buying-EIP: Buying an EIP.
failReason	String	Cause of the CBH instance creation failure.
instanceKey	String	CBH instance tag key.
orderId	String	Order ID.

Parameter	Type	Description
periodNum	String	Number of subscription periods of a CBH instance.
resourceId	String	Resource ID of the CBH instance, in UUID format.
bastion_type	String	CBH instance type. <ul style="list-style-type: none">• OEM
alterPermit	String	Whether the CBH instance can be expanded. <ul style="list-style-type: none">• 1: Enable scale-out.• 0: Disable scale-out.
publicId	String	ID of the EIP bound to the CBH instance. The value is in UUID format.
bastionVersion	String	Current version of the CBH instance.
newBastionVersion	String	The version the CBH instance can be upgraded to.
instanceStatus	String	CBH instance status. <ul style="list-style-type: none">• building: The instance is being created.• deleting: The instance is being deleted.• deleted: The instance was deleted.• unpaid: The order is unpaid.• upgrading: The instance is being upgraded.• resizing: The instance capacity is being expanded.• abnormal: The instance is abnormal.• error: The instance is faulty.• ok: The instance is normal.
instanceDescription	String	CBH instance status description.
slaveZone	String	AZ where the standby instance locates. By default, null is returned.
enterpriseProjectId	String	ID of the enterprise project to which the CBH instance belongs.

Parameter	Type	Description
instanceType	String	CBH instance type. <ul style="list-style-type: none">• null: For a single-node system, null is returned by default.• master: The type of the active instance is returned in HA mode.• slave: The type of the standby instance is returned in HA mode.
hald	String	ID of the primary/standby CBH instance. <ul style="list-style-type: none">• For a single-node CBH instance, null is returned by default.• ID of active CBH instance returns in HA mode.
slaveZoneDisplay	String	Name of the AZ where the standby CBH instance is located. For a single-node CBH instance and a standby CBH instance, null is returned. For an HA CBH instance, the name of the AZ where the standby instance resides is returned.
webPort	String	Port for accessing the web page of the CBH instance.
vip	String	Floating IP address of the CBH instance. The default value null is returned.

Status code: 400**Table 3-5** Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_description	String	Bad request.

Status code: 401

Table 3-6 Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_description	String	Failed to verify the token.

Status code: 403**Table 3-7** Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_description	String	Permissions required.

Status code: 404**Table 3-8** Response body parameters

Parameter	Type	Description
error_code	String	Error code
error_description	String	Not CBH instances found.

Example Requests

None

Example Responses

Status code: 200

The CBH instance list is queried successfully.

```
{
  "total": 1,
  "quotaDetail": "null",
  "instance": [ {
    "publicip": "null",
    "expTime": "14",
    "startTime": "1669911490000",
    "endTime": "1672675199000",
    "releaseTime": "1675267199000",
    "name": "CBH-8a80",
    "instanceId": "f7854c8d-9bbc-4970-8b7d-6ed8280f6bfe",
    "privateIp": "192.168.0.67",
    "status": "SHUTOFF",
    "taskStatus": "NO_TASK",
    "created": "2022-12-02 00:04:42.0",
    "region": "cn-north-4",
```

```
"zone" : "cn-north-4c",
"availability_zone_display" : "AZ 3",
"vpclId" : "279e8217-4a8a-4d83-8fd9-b8f14d47170b",
"subnetId" : "a86217e6-a96d-4803-ae5c-b57da6811249",
"securityGroupId" : "c5564ac2-bdcf-4e45-b892-92c2ae79ecd9",
"specification" : "cbh.basic.50",
"update" : "OLD",
"createInstanceStatus" : "starting-cbs-service",
"failReason" : "No FailReason Now",
"instanceKey" : "6771",
"orderId" : "CS22120200031TOLA",
"periodNum" : "1,2",
"resourceId" : "01903dfb-11fc-4ae4-8bff-bca51d11ad92",
"bastion_type" : "OEM",
"publicId" : "",
"alterPermit" : "1",
"bastionVersion" : "3.3.41.0",
"newBastionVersion" : "3.3.40.1",
"instanceStatus" : "error",
"instanceDescription" : "LoadBalance service error!,2022-12-19 17:14:30,{\"d_usage\": \"0.80\", \"m_usage\": \"30.04\", \"c_usage\": \"0.62\", \"session_count\": \"0\"}",
"webPort" : "443",
"enterpriseProjectId" : "0",
"vip" : null,
"slaveZone" : null,
"slaveZoneDisplay" : null,
"haId" : null,
"instanceType" : null
} ]
}
```

Status Codes

Status Code	Description
200	The CBH instance list is queried successfully.
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found

Error Codes

See [Error Codes](#).

A Appendixes

A.1 Status Codes

Code	Status	Description
200	OK	Request successful.
400	Bad Request	Request failed. Modify the request and then try again.
401	Unauthorized	Authentication failed. The authentication information provided by the client is incorrect or invalid.
403	Forbidden	Request rejected. Modify the request and then try again. The server has received and understood the request; yet it refused to respond, because the request is set to deny access.
404	NotFound	Cannot find the requested resource. Modify the request and then try again.
500	InternalServerError	Internal error. The server is able to receive the request but it could not understand the request.

A.2 Error Codes

Status Code	Error Code	Error Message	Description	Measure
400	CBH.10020009	Invalid request message format.	The request message format was invalid.	Invalid request message format.
400	CBH.10020010	Request message param error.	There are invalid request message body parameters.	Check your request parameters.
400	CBH.10020011	Incorrect server ID.	The service ID was incorrect.	Use a correct server ID.
401	CBH.10020100	The IAM token is invalid.	This error code is returned when the account fails to be verified through IAM.	Check the token.
403	CBH.10020002	Tenant has no authority.	The tenant does not have the required permission.	The tenant does not have the permission. Check the user permission on IAM.
500	CBH.10020000	Unknown error.	An unknown error occurred.	Contact technical support.

A.3 Obtaining a Project ID

Scenario

A project ID is required for some URLs when an API is called. Obtain the required project ID using either of the following methods:

- [Obtaining a Project ID by Calling an API](#)
- [Obtaining a Project ID from the Console](#)

Obtaining a Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is GET `https://{Endpoint}/v3/projects`. **{Endpoint}** is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [Authentication](#).

In the following example, **id** indicates the project ID.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxxxxxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following steps:

1. Log in to the management console.
2. Click the username and choose **My Credential** from the drop-down list.
On the **My Credential** page, view project IDs in the project list.