Dedicated Host

API Reference

Issue 03

Date 2023-12-25

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

Contents

1 Before You Start

1.1 Overview

Welcome to *Dedicated Host API Reference*. Dedicated Host (DeH) provides dedicated physical hosts for you. You can create ECSs on a DeH to enhance isolation, security, and performance of your ECSs. When you migrate services to a DeH, you can continue to use your server software licenses used before the migration. That is, you can use the Bring Your Own License (BYOL) feature on the DeH to independently manage your ECSs.

This document describes how to use application programming interfaces (APIs) to perform operations on DeHs, such as creating, querying, deleting, and modifying DeHs. For details about all supported operations, see **API Overview**.

If you plan to access DeHs through an API, ensure that you are familiar with DeH concepts. For details, see **Service Overview** in the *Dedicated Host User Guide*.

1.2 API Calling

DeH supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 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**.

1.4 Constraints

- The number of DeHs that you can create is determined by your quota. To view or increase the quota, see **Adjusting DeH Resource Quotas** in the *Dedicated Host User Guide*.
- For more constraints, see API description.

1.5 Concepts

- Account

  An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.

- User

  An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).

  API authentication requires information such as the account name, username, and password.

- Region

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

- AZ

  An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, 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 cross-AZ high-availability systems.

- Project

  A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model
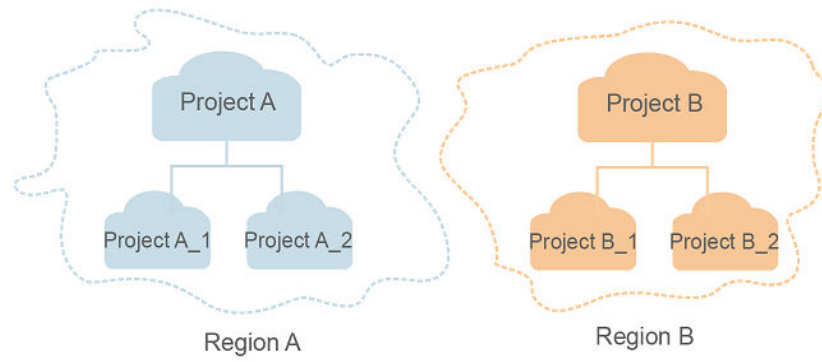
2 API Overview

You can use all DeH functions by making calls to the APIs provided by the DeH service, for example, allocating DeHs, querying the DeH list, viewing DeH details, and releasing a DeH.

Table 2-1 APIs

API	Description
Allocating DeHs	Allocate one or more DeHs and set required parameters, such as the flavor, AZ, and quantity. The number of allocatable DeHs depends on the DeH quota owned by the tenant.
Querying DeHs	Query the DeH list. You can add parameters such as **flavor**, **dedicated_host_id**, and **state** to the URI to filter the result.
Querying Details About a DeH	Query details about a DeH, such as its name, AZ, number of available vCPUs, and available memory size.
Querying ECSs on a DeH	Query information about the ECSs deployed on a DeH, such as the names, IDs, and status of the ECSs.
Modifying DeH Attributes	Change the name of a DeH and enable or disable the auto placement function. After auto placement is enabled, ECSs can be automatically scheduled to this DeH.
Releasing a DeH	Released a DeH that is no longer used.
Querying Available DeH Types	Query available DeH types in an AZ.
DeH Tag Management	Add or delete tags for DeHs and query DeHs by tag.
Quota Configuration	Query the DeH quota of a tenant.

📖 **NOTE**

For details about the APIs for creating ECSs on DeHs, see the *Elastic Cloud Server API Reference*.

3 Calling APIs

3.1 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.

Table 3-1 URI parameter description

Parameter	Description
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.

Table 3-2 HTTP methods

Method	Description
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.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of *Hostname:Port number*. If the port number is not specified, the default port is used. The default port number for **https** is **443**.	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value **application/json** is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No This field is mandatory for requests that use AK/SK authentication in the Dedicated Cloud (DeC) scenario or multi-project scenario.	e9993fc787d94b6c886cbaa340f9c0f4

Parameter	Description	Mandatory	Example Value
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of **X-Subject-Token** in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZlhvc NAQcCo...ggg1B BIINPXsidG9rZ

☐ NOTE

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in **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
```

(Optional) Request Body

This part is optional. 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. Replace *username*, *domainname*, ******** (login password), and *xxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name 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": "xxxxxxxxxxxxxxxxx"
         }
      }
   }
}
```

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.

3.2 Authentication

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

- Token authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token 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. You can obtain a token by calling the Obtaining User Token API.

DeH is a project-level service. When you call the API, set **auth.scope** in the request body to **project**.

```
{
   "auth": {
      "identity": {
         "methods": [
            "password"
         ],
         "password": {
            "user": {
               "name": "username",   // IAM user name
```

```
                  "password": "********",  // IAM user password
                  "domain": {
                     "name": "domainname"  // Name of the account to which the IAM user belongs
                  }
               }
            }
         }
      },
      "scope": {
         "project": {
            "name": "xxxxxxxx"    // Project Name
         }
      }
   }
}
```

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 **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

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

AK/SK Authentication

📖 **NOTE**

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

In AK/SK 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, which is 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 authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see **API Request Signing Guide**.

📖 **NOTE**

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

3.3 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

Similar to a request, a response also has a header, for example, **Content-Type**.

Figure 3-1 shows the response header fields for the API used to obtain a user token. The **x-subject-token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

Figure 3-1 Header fields 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
→ MIIYXQYJKoZIhvcNAQcCoIIYTjCCGEoCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOnsiZXhwaXJlc19hdCI6IjIwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkqjACgklqO1wi4JIGzrpd18LGXK5txldfq4IqHCYb8P4NaY0NYejcAgzJVeFIYtLWT1GSO0zxKZmIQHQj82HBqHdglZO9fuEbL5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXI1jipPEGA270g1FruooL6jqgIFkNPQuFSOU8+uSsttVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==

x-xss-protection → 1; mode=block;

(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to obtain a user token.

```
{
   "token": {
      "expires_at": "2019-02-13T06:52:13.855000Z",
      "methods": [
         "password"
      ],
      "catalog": [
         {
            "endpoints": [
               {
                  "region_id": "az-01",
......
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
   "error_msg": "The format of message is error",
```

```
    "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 API

4.1 Allocating DeHs

Function

This API is used to allocate one or more DeHs and set required parameters, such as the flavor, AZ, and quantity.

Constraints

The number of allocatable DeHs depends on the DeH quota owned by the tenant.

URI

POST /v1.0/{project_id}/dedicated-hosts

Table 4-1 describes the parameters.

Table 4-1 Parameters description

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

Request

Table 4-2 Request parameters

Parameter	In	Type	Mandatory	Description
name	body	String	Yes	Specifies the DeH name.
auto_placement	body	String	No	Specifies whether to allow an ECS to be placed on any available DeH if its DeH ID is not specified during its creation. The value can be **on** or **off**. The default value is **on**.
availability_zone	body	String	Yes	Specifies the AZ to which the DeH belongs.
host_type	body	String	Yes	Specifies the DeH type.
quantity	body	Integer	Yes	Specifies the number of allocatable DeHs.
tags	body	Array of objects	No	Specifies the DeH tags. For details, see **Table 4-3**.

Table 4-3 **tag** field description

Parameter	Type	Mandatory	Description
key	String	Yes	Specifies the tag key. ● It contains a maximum of 36 Unicode characters. ● The value cannot be empty. ● It cannot contain the following ASCII characters: =*<>|/,
value	String	Yes	Specifies the tag value. ● It contains a maximum of 43 Unicode characters. ● It cannot contain the following ASCII characters: =*<>|/,

Response

Table 4-4 Response parameters

Parameter	In	Type	Description
dedicated_host_ids	body	Array of strings	Specifies a group of IDs of allocated DeHs. The tenant can create ECSs on these DeHs.

Example Request

Two s3 DeHs are allocated in **az1**, and the DeH name is **General Computing server1**.

```
POST https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts
{
    "availability_zone": "dc1.az1",
    "name": "General-Purpose server1",
    "auto_placement": "off",
    "host_type": "s3",
    "quantity": 2,
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        }
    ]
}
```

Example Response

```
{
    "dedicated_host_ids": ["xxxxxxx1","xxxxxxx2"]
}
```

Status Code

Table 4-5 Returned error codes

Error Code	Description
403 Forbidden	1. Insufficient quota. 2. The flavor is not supported.
404 FlavorNotFound	Invalid flavor.

For more status codes, see **Status Codes**.

4.2 Querying DeHs

Function

This API is used to query the DeH list.

URI

GET /v1.0/{project_id}/dedicated-hosts

Table 4-6 describes the parameters.

Table 4-6 Parameters description

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

Request

You can add parameters **host_type**, **host_type_name**, **flavor**, **dedicated_host_id**, **state**, **tenant**, **availability_zone**, **name**, **limit**, **marker**, **tags**, **instance_uuid**, or **changes-since** to the URI to filter the search result,

for example, **/v1.0/{project_id}/dedicated-hosts? host_type={host_type}&state={state}**.

Table 4-7 Request parameters

Parameter	In	Type	Mandatory	Description
dedicated _host_id	query	String	No	Specifies the DeH ID.
name	query	String	No	Specifies the DeH name.
host_type	query	String	No	Specifies the DeH type.
host_type _name	query	String	No	Specifies the name of the DeH type.
flavor	query	String	No	Specifies the flavor ID.

Paramete r	In	Type	Manda tory	Description
state	query	String	No	Specifies the DeH status. The value can be **available**, **fault**, or **released**.
tenant	query	String	No	The value can be a tenant ID or **all**. Only users with the DeH administrator permissions can specify this parameter.
availabilit y_zone	query	String	No	Specifies the AZ to which the DeH belongs.
limit	query	String	No	Specifies the number of records displayed per page.
marker	query	String	No	Specifies the ID of the last record on the previous page. If the **marker** value is invalid, status code 400 is returned.
tags	query	String	No	Specifies the DeH tags.
instance_ uuid	query	String	No	Specifies the ID of the ECS on the DeH.
changes- since	query	String	No	Filters the response by date and timestamp when the DeH status changes. To help keep track of changes, this parameter may also display recently deleted DeHs. The format of the date and timestamp is ISO 8601: CCYY-MM-DDThh:mm:ss±hh:mm If the **hh:mm** value is included, the time zone is returned as the UTC offset, for example, **2015-08-27T09:49:58-05:00**. If you omit the time zone, the UTC time zone is assumed.

Response

Table 4-8 Response parameters

Parameter	In	Type	Description
dedicated_hosts	body	Array of objects	Specifies the DeHs that meet the search criteria. For details, see **Table 5-1**.
total	body	Integer	Specifies the quantity of DeHs that meet the search criteria.

Example Request

Query the DeHs in the **available** state.

GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts?state=available

Example Response

```
{
    "dedicated_hosts": [
        {
            "dedicated_host_id": "ab910cf0daebca90c4001",
            "name": "high performance servers1",
            "auto_placement": "off",
            "availability_zone": "az1",
            "host_properties": {
                "vcpus": 36,
                "cores": 12,
                "sockets": 2,
                "memory": 1073741824,
                "host_type": "h1",
                "host_type_name": "High performance",
                "available_instance_capacities": [
                    {
                        "flavor": "h1.large"
                    },
                    {
                        "flavor": "h1.2large"
                    },
                    {
                        "flavor": "h1.4large"
                    },
                    {
                        "flavor": "h1.8large"
                    }
                ]
            },
            "state": "available",
            "project_id": "9c53a566cb3443ab910cf0daebca90c4",
            "available_vcpus": 20,
            "available_memory": 1073201821,
            "instance_total": 2,
            "allocated_at": "2016-10-10T14:35:47Z",
            "released_at": ""
        },
        {
            "dedicated_host_id": "ab910cf0daebca90c4002",
```

```
            "name": "high performance servers2",
            "auto_placement": "off",
            "availability_zone": "az1",
            "host_properties": {
               "vcpus": 36,
               "cores": 12,
               "sockets": 2,
               "host_type": "h1",
               "host_type_name": "High performance",
               "memory": 1073741824,
               "available_instance_capacities": [
                  {
                     "flavor": "h1.large"
                  },
                  {
                     "flavor": "h1.2large"
                  },
                  {
                     "flavor": "h1.4large"
                  },
                  {
                     "flavor": "h1.8large"
                  }
               ]
            },
            "state": "available",
            "project_id": "9c53a566cb3443ab910cf0daebca90c4",
            "available_vcpus": 20,
            "available_memory": 1073101821,
            "instance_total": 3,
            "allocated_at": "2016-10-10T14:35:47Z",
            "released_at": ""
            },
            ],
      "total": 25
}
```

Status Code

See **Status Codes**.

4.3 Querying Details About a DeH

Function

This API is used to query details about a DeH.

URI

GET /v1.0/{project_id}/dedicated-hosts/{dedicated_host_id}

Table 4-9 describes the parameters.

Table 4-9 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host_id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

None

Response

Table 4-10 Response parameters

Parameter	In	Type	Description
dedicated_host	body	Object	Specifies the DeH object. For details, see **Table 5-1**.

Example Request

Query details about the DeH **ab910cf0daebca90c4001**.

GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts/ab910cf0daebca90c4001

Example Response

```
{
   "dedicated_host": {
      "dedicated_host_id": "ab910cf0daebca90c4001",
      "name": "win_2008 servers",
      "auto_placement": "off",
      "availability_zone": "az1",
      "host_properties": {
         "vcpus": 36,
         "cores": 12,
         "sockets": 2,
         "memory": 1073741824,
         "host_type": "h1",
         "host_type_name": "High performance",
         "available_instance_capacities": [
            {
```

```
                "flavor": "h1.large"
            },
            {
                "flavor": "h1.2large"
            },
            {
                "flavor": "h1.4large"
            },
            {
                "flavor": "h1.8large"
            }
        ]
    },
    "state": "available",
    "project_id": "9c53a566cb3443ab910cf0daebca90c4",
    "available_vcpus": 20,
    "available_memory": 1073201821,
    "instance_total": 2,
    "allocated_at": "2016-10-10T14:35:47Z",
    "released_at": "",
    "instance_uuids": [
        "erf5th66cb3443ab912ff0daebca3456",
        "23457h66cb3443ab912ff0daebcaer45"
    ]
  }
}
```

Status Code

See **Status Codes**.

4.4 Querying ECSs on a DeH

Function

This API is used to query information about deployed ECSs on a DeH.

URI

GET /v1.0/{project_id}/dedicated-hosts/{dedicated_host_id}/servers

Table 4-11 describes the parameters.

Table 4-11 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host _id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

Table 4-12 Request parameters

Parameter	In	Type	Mandatory	Description
limit	query	String	No	Specifies the number of records displayed per page.
marker	query	String	No	Specifies the ID of the last record on the previous page. If the **marker** value is invalid, status code 400 is returned.

Response

Table 4-13 Response parameters

Parameter	In	Type	Description
servers	body	Array of objects	Specifies the server object. For details, see **Table 4-14**.

Table 4-14 servers field description

Parameter	Type	Description
addresses	Map<String, Object>	Specifies the network addresses of an ECS. The structure is Map<String, Object>. ● The key indicates the VPC subnet ID. ● The value indicates the network attributes specified in **Table 4-15**.
created	String	Specifies the time when the ECS was created.
flavor	Object	Specifies the ECS flavor. For details, see **Table 4-16**.
id	String	Specifies the ECS ID in UUID format.
name	String	Specifies the ECS name.

Parameter	Type	Description
status	String	Specifies the ECS status. Options: **ACTIVE, BUILD, DELETED, ERROR, HARD_REBOOT, MIGRATING, PASSWORD, PAUSED, REBOOT, REBUILD, RESIZE, REVERT_RESIZE, SHUTOFF, SHELVED, SHELVED_OFFLOADED, SOFT_DELETED, SUSPENDED,** and **VERIFY_RESIZE**
tenant_id	String	Specifies the ECS tenant ID in UUID format.
updated	String	Specifies the time when the ECS was updated last time.
user_id	String	Specifies the ID of the user who has created the ECS. The value is in UUID format.
task_state	String	Specifies the ECS task status.
image	Object	Specifies the ECS image. For details, see **Table 4-17**.
metadata	Object	Specifies the ECS metadata. For details, see **Table 4-18**.

Table 4-15 Data structure of the network to which an ECS accesses

Parameter	Type	Description
addr	String	Specifies the IP address.
version	Integer	Specifies the type of an IP address. The value of this parameter can be **4** or **6**. ● **4**: The type of the IP address is IPv4. ● **6**: The type of the IP address is IPv6.
OS-EXT-IPS-MAC:mac_addr	String	Specifies the MAC address. This is an extended attribute.
OS-EXT-IPS:type	String	Specifies the IP address assignment mode. This is an extended attribute.

Table 4-16 flavor field description

Parameter	Type	Description
id	String	Specifies the flavor ID.

Table 4-17 image field description

Parameter	Type	Description
id	String	Specifies the image UUID.

Table 4-18 metadata field description

Parameter	Type	Description
os_type	String	Specifies the OS type.

Example Request

Query ECSs on the DeH **ab910cf0daebca90c4001**.

```
GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts/
ab910cf0daebca90c4001/servers
```

Example Response

```
{
  "servers": [
    {
      "addresses": {
        "68269e6e-4a27-441b-8029-35373ad50bd9": [
          {
            "addr": "192.168.0.3",
            "version": 4,
            "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:1b:35:78",
            "OS-EXT-IPS:type": "fixed"
          }
        ]
      },
      "created": "2012-09-07T16:56:37Z",
      "flavor": {
        "id": "1"
      },
      "id": "05184ba3-00ba-4fbc-b7a2-03b62b884931",
      "metadata": {
        "os_type": "Linux"
      },
      "name": "new-server-test",
      "status": "ACTIVE",
      "tenant_id": "a90b2728805d4240a72cc2eeb4e1244d",
      "updated": "2012-09-07T16:56:37Z",
      "user_id": "fake",
      "task_state": "",
      "image": {
        "id": "1ce5800a-e487-4c1b-b264-3353a39e2b4b"
```

```
      }
    }
  ]
}
```

Status Code

See **Status Codes**.

4.5 Modifying DeH Attributes

Function

This API is used to modify the **auto_placement** and **name** attributes of a DeH.

URI

PUT /v1.0/{project_id}/dedicated-hosts/{dedicated_host_id}

Table 4-19 describes the parameters.

Table 4-19 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host _id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

Table 4-20 Request parameters

Parameter	Mandatory	Type	Description
dedicated_host	Yes	Object	Specifies the dedicated host. For details, see **Table 4-21**.

Table 4-21 **dedicated_host** field data structure description

Parameter	Location	Type	Mandatory	Description
auto_placement	in	String	No	Specifies whether to allow an ECS to be placed on any available DeH if its DeH ID is not specified during its creation. The value can be **on** or **off**.
name	in	String	No	Specifies the DeH name.

Response

None

Example Request

Set **auto_placement** to **off** to not allow an ECS to be placed on any available DeH. Change the name of the DeH to **DeH_vm3**.

```
PUT https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts/74259164-
e63a-4ad9-9c77-a1bd2c9aa187
{
    "dedicated_host": {
        "auto_placement": "off",
        "name": "DeH_vm3"
    }
}
```

Example Response

```
Http Response Code: 204
```

Status Code

See **Status Codes**.

4.6 Releasing a DeH

Function

This API is used to release a DeH.

Constraints

A DeH accommodating ECSs cannot be released.

URI

DELETE /v1.0/{project_id}/dedicated-hosts/{dedicated_host_id}

Table 4-22 describes the parameters.

Table 4-22 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host_id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

None

Response

None

Example Request

Release the DeH **74259164-e63a-4ad9-9c77-a1bd2c9aa187**.

```
DELETE https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-hosts/74259164-
e63a-4ad9-9c77-a1bd2c9aa187
```

Example Response

```
Http Response Code: 204
```

Status Code

Table 4-23 Returned error codes

Error Code	Description
409 Conflict	A DeH accommodating ECSs cannot be released.

For more status codes, see **Status Codes**.

4.7 Querying Available DeH Types

Function

This API is used to query available DeH types in an AZ.

URI

Get /v1.0/{project_id}/availability-zone/{availability_zone}/dedicated-host-types

Table 4-24 describes the parameters.

Table 4-24 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
availability_zone	String	Yes	Specifies the AZ.

Request

None

Response

Table 4-25 Response parameters

Parameter	In	Type	Description
dedicated_host_types	body	Array of objects	Specifies the available DeH types. For details, see **Table 4-26**.

Table 4-26 dedicated_host_types field description

Parameter	Type	Description
host_type	String	Specifies the DeH type.
host_type_name	String	Specifies the name of the DeH type.

Example Request

Query the DeH types available in **az1**.

```
GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/availability-zone/az1/dedicated-host-types
```

Example Response

```
{
    "dedicated_host_types": [
        {
            "host_type": "General",
            "host_type_name": "General Computing"
        },
        {
            "host_type": "m1",
            "host_type_name": "Memory-optimized"
        },
        {
            "host_type": "h2",
            "host_type_name": "High performance"
        },
        {
            "host_type": "d1",
            "host_type_name": "Disk intensive"
        }
    ]
}
```

Status Code

See **Status Codes**.

4.8 DeH Tag Management

4.8.1 Adding Tags to a DeH in Batches

Function

- This API is used to add tags to a specified DeH in batches.
- Tag Management Service (TMS) uses this API to batch add tags to a DeH.

Constraint

- A DeH allows a maximum of 10 tags.
- This API is idempotent.

  During tag creation, if a tag exists (both the key and value are the same as those of an existing tag), the tag is successfully processed by default.
- A new tag will overwrite the original one if their keys are the same and values are different.

URI

POST /v1.0/{project_id}/dedicated-host-tags/{dedicated_host_id}/tags/action

Table 4-27 describes the parameters.

Table 4-27 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host_id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

Table 4-28 Request parameters

Parameter	Type	Mandatory	Description
tags	Array of objects	Yes	Specifies the tag list. For details, see **Table 4-29**.
action	String	Yes	Specifies the operation. Only lowercase letters are supported. For example, **create** indicates the creation operation.

Table 4-29 tag field description

Parameter	Type	Mandatory	Description
key	String	Yes	Specifies the tag key. ● It contains a maximum of 36 Unicode characters. ● The value cannot be empty. ● It cannot contain the following ASCII characters: =*<>|/,

Parameter	Type	Mandatory	Description
value	String	Yes	Specifies the tag value. ● It contains a maximum of 43 Unicode characters. ● It cannot contain the following ASCII characters: =*<>|/,

Response

N/A

Example Request

Add two tags to a DeH in a batch. The keys and corresponding values for these two tags are as follows: **key1** and **value1**; **key2** and **value2**.

```
POST https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-host-tags/74259164-
e63a-4ad9-9c77-a1bd2c9aa187/tags/action
{
    "action": "create",
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        }
    ]
}
```

Example Response

N/A

Status Code

See **Status Codes**.

4.8.2 Deleting Tags from a DeH in Batches

Function

● This API is used to delete tags from a specified DeH in batches.

● Tag Management Service (TMS) uses this API to batch delete tags from a DeH.

Constraint

A DeH allows a maximum of 10 tags.

URI

POST /v1.0/{project_id}/dedicated-host-tags/{dedicated_host_id}/tags/action

Table 4-30 describes the parameters.

Table 4-30 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host_id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

Table 4-31 Request parameters

Parameter	Type	Mandatory	Description
tags	Array of objects	Yes	Specifies the tag list. For details, see **Table 4-32**.
action	String	Yes	Specifies the operation. Only lowercase letters are supported. For example, **delete** indicates the deletion operation.

Table 4-32 tag field description

Paramete r	Type	Mandator y	Description
key	String	Yes	Specifies the tag key. ● It contains a maximum of 36 Unicode characters. ● The value cannot be empty. ● It cannot contain the following ASCII characters: =*<>\\|/,
value	String	Yes	Specifies the tag value. ● It contains a maximum of 43 Unicode characters. ● It cannot contain the following ASCII characters: =*<>\\|/,

Response

N/A

Example Request

Delete two tags from a DeH in a batch. The keys and corresponding values for these two tags are as follows: **key1** and **value1**; **key2** and **value2**.

```
POST https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-host-tags/74259164-
e63a-4ad9-9c77-a1bd2c9aa187/tags/action
{
    "action": "delete",
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        }
    ]
}
```

Example Response

N/A

Status Code

See **Status Codes**.

4.8.3 Querying Tags of a DeH

Function

- This API is used to query tags of a DeH.
- Tag Management Service (TMS) uses this API to query all tags of a DeH.

URI

GET /v1.0/{project_id}/dedicated-host-tags/{dedicated_host_id}/tags

Table 4-33 describes the parameters.

Table 4-33 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
dedicated_host_id	String	Yes	Specifies the DeH ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

None

Response

Table 4-34 Response parameters

Parameter	Type	Description
tags	Array of objects	Specifies the list of tags. For details, see **Table 4-35**.

Table 4-35 tag field description

Parameter	Type	Description
key	String	Specifies the tag key.
value	String	Specifies the tag value.

Example Request

Query the tags of the DeH **74259164-e63a-4ad9-9c77-a1bd2c9aa187**.

```
GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-host-tags/74259164-
e63a-4ad9-9c77-a1bd2c9aa187/tags
```

Example Response

```
{
   "tags": [
      {
         "key": "key1",
         "value": "value1"
      },
      {
         "key": "key2",
         "value": "value2"
      }
   ]
}
```

Status Code

See **Status Codes**.

4.8.4 Querying DeHs by Tag

Function

- This API is used to filter DeHs by tag and return the list of all tags of a DeH.
- Tag Management Service (TMS) uses this API to filter the DeHs.

URI

POST /v1.0/{project_id}/dedicated-host-tags/resource_instances/action

Table 4-36 describes the parameters.

Table 4-36 Parameters description

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

Request

Table 4-37 Request parameters

Parameter	Type	Mandatory	Description
tags	Array of objects	No	Displays all DeHs with specified tags. For more information, see **Table 4-38**. ● A maximum of 10 keys can be included. Each key can have a maximum of 10 values. ● The structure body must be included. ● The tag key cannot be left blank or set to an empty string. ● A key must be unique. ● Values of the same key must be unique.
not_tags	Array of objects	No	Displays the DeHs with none of specified tags. For more information, see **Table 4-38**. ● A maximum of 10 keys can be included. Each key can have a maximum of 10 values. ● The structure body must be included. ● The tag key cannot be left blank or set to an empty string. ● Keys must be unique. ● Values of the same key must be unique.

Parameter	Type	Mandatory	Description
limit	Integer	No	Limits the maximum number of queried DeHs. The value cannot be a negative number. The maximum value is 1000. ● If the **action** value is **count**, this parameter is invalid. ● If the **action** value is **filter**, the default value is **1000**.
offset	Integer	No	Specifies the index position. The query starts from the next piece of data indexed by this parameter. The value must be a non-negative number. You do not need to specify this parameter when querying resources on the first page. When you query resources on subsequent pages, set the value of **offset** to the location returned in the response body for the previous query. ● If the **action** value is **count**, this parameter is invalid. ● If the **action** value is **filter**, the default value is **0**.
action	String	Yes	Specifies the operation, which can be **filter** or **count**. ● **filter**: Filters DeHs by tag and lists DeHs that meet the search criteria. Listed DeHs are queried by page. ● **count**: Searches for DeHs by tag and returns the number of DeHs that meet the search criteria.

Paramete r	Type	Manda tory	Description
tags_any	Array of objects	No	Includes any of the specified tags. For more information, see **Table 4-38**. ● This field contains a maximum of 10 tag keys and each tag key has a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. ● Each key must be unique, and cannot contain duplicate values. ● The response returns resources containing the tags in this list. Keys in this list are in an OR relationship and values in each key-value structure are also in an OR relationship. ● If no tag filtering condition is specified, full data is returned.
not_tags_a ny	Array of objects	No	Excludes any of the specified tags. For more information, see **Table 4-38**. ● This field contains a maximum of 10 tag keys and each tag key has a maximum of 10 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. ● Each key must be unique, and cannot contain duplicate values. ● The response returns resources containing no tags in this list. Keys in this list are in an OR relationship and values in each key-value structure are also in an OR relationship. ● If no tag filtering condition is specified, full data is returned.
matches	Array of objects	No	Specifies the search field, which is used to search for DeHs by condition. Currently, only **resource_name** can be used for search. For more information, see **Table 4-39**.

Table 4-38 tag field description

Parameter	Type	Mandatory	Description
key	String	Yes	Specifies the tag key. ● It contains a maximum of 127 Unicode characters. ● This field cannot be left blank.
values	Array of strings	No	Specifies the tag values. ● Each tag contains a maximum of 10 values. ● Values of the same tag must be unique. ● Each value can contain a maximum of 255 Unicode characters. ● If this parameter is not specified, any value can be used. ● The resources containing one or more values listed in **values** will be found and displayed.

Table 4-39 match field description

Parameter	Type	Mandatory	Description
key	String	Yes	Specifies the key parameter to be matched. ● The key must be unique, and the value is used for matching. ● The **key** field is a fixed dictionary value. ● This field cannot be left blank. **NOTE** The parameter value can only be **resource_name**, which is the DeH name.

Parameter	Type	Mandatory	Description
value	String	Yes	Specifies the tag value. • Each value can contain a maximum of 255 Unicode characters. • This field cannot be left blank.

Response

Table 4-40 Response parameters

Parameter	Type	Description
resources	Array of objects	Specifies the returned DeH list. For details, see **Table 4-41**.
total_count	Integer	Specifies the total number of resources.

Table 4-41 Description of the **resource** field

Parameter	Type	Description
resource_id	String	Specifies the DeH ID.
resouce_detail	String	Specifies the DeH details. This field is used for future extension and is left empty by default.
tags	Array of objects	Specifies the tag list. For details, see **Table 4-42**.
resource_name	String	Specifies the resource name.

Table 4-42 tag field description

Parameter	Type	Description
key	String	Specifies the tag key. ● It contains a maximum of 36 Unicode characters. ● This field cannot be left blank. ● It cannot contain the following ASCII characters: =*<>\\|/,
value	String	Specifies the tag value. ● Each value contains a maximum of 43 Unicode characters. ● This field can be left blank. ● It cannot contain the following ASCII characters: =*<>\\|/,

Example Request

Filter DeHs by tag. From the first data record, query the DeH using the search field (field: **resource_name**; value: **resource1**) and the tag (key: **key1**; value: **value1**).

```
POST https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/dedicated-host-tags/resource_instances/
action
{
    "offset": "0",
    "limit": "100",
    "action": "filter",
    "matches": [
        {
            "key": "resource_name",
            "value": "resource1"
        }
    ],
    "tags": [
        {
            "key": "key1",
            "values": ["value1"]
        }
    ]
}
```

Example Response

Response body when **action** is set to **filter**

```
{
    "resources": [
        {
            "resource_detail": null,
            "resource_id": "cdfs_cefs_wesas_12_dsad",
            "resource_name": "resource1",
            "tags": [
                {
                    "key": "key1",
                    "value": "value1"
                }
            ]
```

```
        }
    ],
    "total_count": 1
}
```

Response body when **action** is set to **count**

```
{
    "total_count": 100
}
```

Status Code

See **Status Codes**.

4.9 Quota Configuration

4.9.1 Querying the DeH Quota of a Tenant

Function

This API is used to query the DeH quota of a tenant.

URI

GET /v1.0/{project_id}/quota-sets/{tenant_id}

Table 4-43 describes the parameters.

Table 4-43 Parameters description

Parameter	Type	Mandatory	Description
project_id	String	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
tenant_id	String	Yes	Specifies the tenant ID. You can obtain the value from the DeH console or using the API in **Querying DeHs**.

Request

You can add the **resource** parameter to the URI. For example:

/v1.0/{project_id}/quota-sets/{tenant_id}?resource={resource}

Table 4-44 Request parameters

Parameter	Location	Type	Mandatory	Description
resource	query	String	No	Specifies the resource type.

Response

Table 4-45 Response parameters

Parameter	Type	Description
quota_set	Array of objects	Specifies the quota set of a DeH. For details, see **Table 4-46**.

Table 4-46 quota_set field description

Parameter	Type	Description
resource	String	Specifies the resource type.
hard_limit	Integer	Specifies the quota limit. **-1** indicates that the resource quota is not limited.
used	Integer	Specifies the used amount of the quota.

Example Request

Query the DeH quota of the tenant whose ID is **45df5566cb3443ab910cf0daebcapoi8**.

```
GET https://{Endpoint}/v1.0/9c53a566cb3443ab910cf0daebca90c4/quota-sets/
45df5566cb3443ab910cf0daebcapoi8
```

Example Response

```
{
   "quota_set": [
```

```
    {
        "resource": "c1",
        "hard_limit": 5,
        "used": 2
    },
    {
        "resource": "m1",
        "hard_limit": 5,
        "used": 0
    },
    {
        "resource": "h1",
        "hard_limit": 5,
        "used": 2
    },
    {
        "resource": "d1",
        "hard_limit": 5,
        "used": 2
    }
  ]
}
```

Status Code

See **Status Codes**.

5 Public Parameters

5.1 Object Models

Objects

DeH management includes querying the DeH list, viewing DeH details, modifying DeH attributes, allocating a DeH, and releasing a DeH.

Object Models

Table 5-1 dedicated_host

Parameter	Type	CRUD	Default Value	Constraint	Description
dedicated_host_id	String	R	N/A	N/A	Specifies the DeH ID.
name	String	CUR	N/A	N/A	Specifies the DeH name. The name can contain a maximum of 255 characters and cannot start or end with spaces.
auto_placement	String	CUR	on	The value can be **on** or **off**.	Specifies whether to allow an ECS to be placed on any available DeH if its DeH ID is not specified during its creation.
availability_zone	String	CR	N/A	N/A	Specifies the AZ to which the DeH belongs.

Paramete r	Type	CRUD	Default Value	Constrain t	Description
project_id	String	CR	N/A	N/A	Specifies the tenant who owns the DeH.
host_prop erties	Dict For details, see **Table 5-2**.	R	N/A	N/A	Specifies the DeH properties.
state	String	R	N/A	The value can be **available**, **fault**, or **released**.	Specifies the DeH status.
available_ vcpus	Int	R	N/A	N/A	Specifies the number of available vCPUs for the DeH.
available_ memory	Int	R	N/A	N/A	Specifies the available memory size of the DeH.
allocated_ at	String	R	N/A	N/A	Specifies the time when the DeH is allocated.
released_a t	String	R	N/A	N/A	Specifies the time when the DeH is released.
instance_t otal	Int	R	N/A	N/A	Specifies the total number of ECSs on the DeH.
instance_u uids	List <String>	R	N/A	N/A	Specifies the UUIDs of the ECSs running on the DeH. This parameter is not displayed on the interface for **querying DeHs**.
tags	Dict(str: str)	R	N/A	N/A	Specifies the DeH tags.
sys_tags	Dict(str: str)	R	N/A	N/A	Specifies the DeH system tags.

Table 5-2 host_property

Parameter	Type	CRUD	Default Value	Constraint	Description
host_type	String	R	N/A	N/A	Specifies the DeH type.
host_type_name	String	R	N/A	N/A	Specifies the name of the DeH type.
vcpus	Int	R	N/A	N/A	Specifies the number of vCPUs on the DeH.
cores	Int	R	N/A	N/A	Specifies the number of physical cores on the DeH.
sockets	Int	R	N/A	N/A	Specifies the number of physical sockets on the DeH.
memory	Int	R	N/A	N/A	Specifies the size of physical memory on the DeH.
available_instance_capacities	List For details, see **Table 5-3**.	R	N/A	N/A	Specifies the flavors of ECSs placed on the DeH.

Table 5-3 available_instance_capacity

Paramete r	Type	CRUD	Default Value	Constraint	Description
flavor	String	R	N/A	N/A	Specifies the specifications of ECSs that can be created.

5.2 Status Codes

- Normal

Returned Value	Description
200 OK	The request has been successfully processed.
201 Created	The request has been successfully processed and a resource has been created.
202 Accepted	The request has been accepted but may not be processed immediately.
204 No Content	The request has been successfully processed, but there is no need to send any data back.

- Abnormal

Returned Value	Description
400 Bad Request	The request failed to be processed due to bad syntax.
401 Unauthorized	A username and a password are required to access a page.
403 Forbidden	The requested page cannot be accessed.
404 Not Found	The requested page cannot be found.
405 Method Not Allowed	A method in the request is not allowed.
406 Not Acceptable	The response cannot be accepted by the client.
407 Proxy Authentication Required	The client must be authorised by the proxy before the request can proceed.
408 Request Timeout	The request timed out.
409 Conflict	Failed to complete the request due to a conflict.

Returned Value	Description
500 Internal Server Error	Failed to complete the request due to a system error.
501 Not Implemented	Failed to complete the request because the server does not support all that is needed for the request to be completed.
502 Bad Gateway	Failed to complete the request because the server has received an invalid response from an upstream server.
503 Service Unavailable	Failed to complete the request because the server is unavailable.
504 Gateway Timeout	The server has waited too long for a response from an upstream server.

5.3 Obtaining a Project ID

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. The steps are as follows:

1. Obtain the token.

   For details, see **Token Authentication**.

2. Obtain a project ID.

   The API for obtaining the project ID is **GET https://iam.eu-west-0.myhuaweicloud.com/v3/projects**.

   Add **X-Auth-Token** to the request header and set its value to the token obtained in the preceding step.

   The following is an example response. The value of **id** is the project ID to be obtained.

```
{
    "links": {},
    "projects": [
        {
            "is_domain": ,
            "description": "",
            "links": {},
            "enabled": true,
            "id": "",   // Project ID
            "parent_id": "",
            "domain_id": "",
            "name": ""
        },
        ...
    ]
}
```

6 Change History

Released On	Description
2023-12-25	This issue is the third official release. It incorporates the following changes: ● Optimized the structure in **API**. ● Deleted request parameter **released_at** in **Querying DeHs**. ● Updated the value of parameter **tenant_id** in the example response of section **Querying ECSs on a DeH**. ● Corrected the value of **released_at** in the example responses of **Querying DeHs** and **Querying Details About a DeH**.
2021-03-30	This issue is the second official release. Added: **DeH Tag Management**
2020-12-02	This issue is the first official release.