Dedicated Host

API Reference

Issue 01

Date 2025-08-21

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Contents

1 Before You Start 1
2 API Overview 4
3 Calling APIs 6
3.1 Making an API Request 6
3.2 Authentication 10
3.3 Response 12
4 API 14
4.1 Querying DeHs 14
4.2 Querying Details About a DeH 17
4.3 Querying ECSs on a DeH 19
4.4 Modifying DeH Attributes 23
4.5 Querying Available DeH Types 25
4.6 API Version Query 26
4.6.1 Querying API Versions 26
4.6.2 Querying an API Version 28
4.7 DeH Tag Management 30
4.7.1 Adding Tags to a DeH in Batches 30
4.7.2 Deleting Tags from a DeH in Batches 32
4.7.3 Querying Tags of a DeH 34
4.7.4 Querying DeHs by Tag 36
4.8 Quota Configuration 42
4.8.1 Querying the DeH Quota of a Tenant 42
5 Public Parameters 45
5.1 Object Models 45
5.2 Status Codes 48
5.3 Obtaining a Project ID 49

1 Before You Start

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 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**.

API Calling

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

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**.

Concepts

- Account

  An account is created upon successful signing up. 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

  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 into universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides specific services for specific tenants.

  For details, see **Region and AZ**.

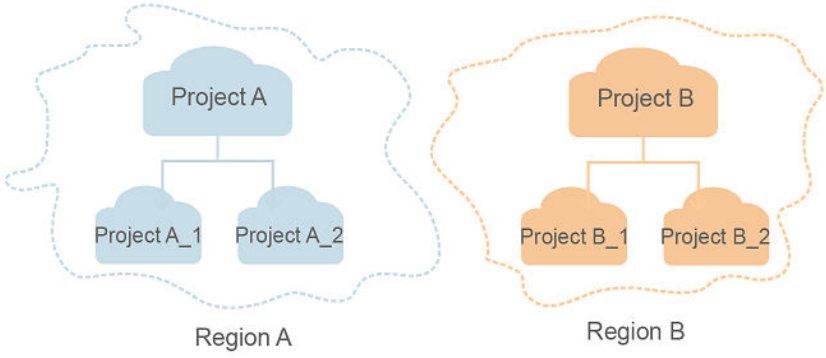
- AZ

  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

- Enterprise Project

  Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

For details about enterprise projects and about how to obtain enterprise project IDs, see ***Enterprise Management User Guide***.

2 API Overview

You can use almost all DeH functions by making calls to the APIs provided by the DeH service, for example, querying the DeH list and viewing DeH details. However, no APIs are provided for purchasing and deleting DeHs currently. You can create or delete DeHs on the management console only.

Table 2-1 APIs

API	Description
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.
Querying Available DeH Types	Query available DeH types in an AZ.
Querying API Versions	Query all available API versions and the versions of specified APIs.
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

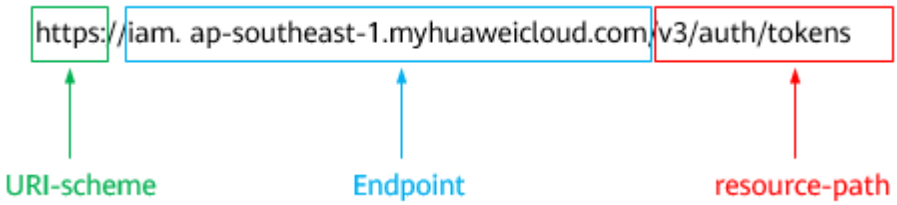
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from **Regions and Endpoints**. For example, the endpoint of IAM in region **CN-Hong Kong** is **iam.ap-southeast-1.myhuaweicloud.com**.
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**.

Parameter	Description
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.

For example, to obtain an IAM token in the **CN-Hong Kong** region, obtain the endpoint of IAM (iam.ap-southeast-1.myhuaweicloud.com) for this region and the resource-path (/v3/auth/tokens) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



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.

Method	Description
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://iam.ap-southeast-1.myhuaweicloud.com/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

Parameter	Description	Mandatory	Example Value
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
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: MIIPAgYJKoZIhvcNAQcCo...ggg1BBIINPXsidG9rZ

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://iam.ap-southeast-1.myhuaweicloud.com/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 (name of an account to which a user belongs), ******** (login password), and *xxxxxxxxxxxxxxxxx* (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://iam.ap-southeast-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": $USER_NAME, # You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "password": $PASSWORD,  # You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "domain": {
                        "name": $DOMAIN_NAME, # You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": $PROJECT_NAME, # You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
            }
        }
    }
}
```

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": $USER_NAME, # IAM user name. You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "password":$PASSWORD, # IAM user password. You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    "domain": {
                        "name": $DOMAIN_NAME, # Name of the account to which the IAM user belongs. You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
                    }
                }
            }
        },
        "scope": {
            "project": {
                "name": $PROJECT_NAME, # Project name. You are advised to store it in ciphertext in the configuration file or an environment variable and decrypt it when needed to ensure security.
            }
        }
    }
}
```

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://iam.ap-southeast-1.myhuaweicloud.com/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.
- The AK/SK can either be permanent or temporary. If it is temporary, the **X-Security-Token** field must be included in the request header. The value is the security token of the temporary AK/SK.

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 **create an IAM user**, the request is successful.

Response Header

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

Figure 3-2 shows the response header fields for the API used to **create an IAM user**. The **X-Subject-Token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-2 Header fields of the response to the request for creating an IAM user

```
"X-Frame-Options": "SAMEORIGIN",
"X-IAM-ETag-id": "2562365939-d8f6f12921974cb097338ac11fceac8a",
"Transfer-Encoding": "chunked",
"Strict-Transport-Security": "max-age=31536000; includeSubdomains;",
"Server": "api-gateway",
"X-Request-Id": "af2953f2bcc67a42325a69a19e6c32a2",
"X-Content-Type-Options": "nosniff",
"Connection": "keep-alive",
"X-Download-Options": "noopen",
"X-XSS-Protection": "1; mode=block;",
"X-IAM-Trace-Id": "token_[redacted]_null_af2953f2bcc67a42325a69a19e6c32a2",
"Date": "Tue, 21 May 2024 09:03:40 GMT",
"Content-Type": "application/json; charset=utf8"
```

(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 shows part of the response body for the API used to **create an IAM user**.

```
{
    "user": {
        "id": "c131886aec...",
        "name": "IAMUser",
        "description": "IAM User Description",
        "areacode": "",
        "phone": "",
        "email": "***@***.com",
        "status": null,
        "enabled": true,
        "pwd_status": false,
        "access_mode": "default",
        "is_domain_owner": false,
        "xuser_id": "",
        "xuser_type": "",
        "password_expires_at": null,
        "create_time": "2024-05-21T09:03:41.000000",
        "domain_id": "d78cbac1.....",
        "xdomain_id": "30086000.....",
        "xdomain_type": "",
        "default_project_id": null
    }
}
```

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 request message format is invalid.",
    "error_code": "IMG.0001"
}
```

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

4 API

4.1 Querying DeHs

Function

This API is used to query the DeH list.

URI

GET /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

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 4-2 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. For details, see **Overview**.
host_type_name	query	String	No	Specifies the name of the DeH type.
flavor	query	String	No	Specifies the flavor ID.
state	query	String	No	Specifies the DeH status. Range: • **available** • **fault** • **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.
availability_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.

Parameter	In	Type	Mandatory	Description
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 4-3 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": "d465d0ae-f859-4a83-a508-8db654c05e7e",
            "name": "DEH001",
            "auto_placement": "off",
            "availability_zone": "cn-east-3a",
            "host_properties": {
                "vcpus": 74,
                "cores": 22,
                "sockets": 2,
                "memory": 151552,
```



```
            "host_type": "c6",
            "host_type_name": "dedicated_general_purpose",
            "available_instance_capacities": [
               {
                  "flavor": "c6.large.2"
               },
               {
                  "flavor": "c6.xlarge.2"
               },
               {
                  "flavor": "c6.2xlarge.2"
               },
               {
                  "flavor": "c6.3xlarge.2"
               },
               {
                  "flavor": "c6.6xlarge.2"
               },
               {
                  "flavor": "c6.16xlarge.2"
               }
            ]
         },
         "state": "available",
         "project_id": "9c53a566cb3443ab910cf0daebca90c4",
         "available_vcpus": 20,
         "available_memory": 40960,
         "instance_total": 5,
         "allocated_at": "2016-10-10T14:35:47Z",
         "released_at": ""
      }
   ],
   "total": 1
}
```


Status Code

See **Status Codes**.

4.2 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 4-4 describes the parameters.

Table 4-4 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 4-5 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": "d465d0ae-f859-4a83-a508-8db654c05e7e",
      "name": "DEH001",
      "auto_placement": "off",
      "availability_zone": "cn-east-3a",
      "host_properties": {
         "vcpus": 74,
         "cores": 22,
         "sockets": 2,
         "memory": 151552,
         "host_type": "c6",
         "host_type_name": "dedicated_general_purpose",
         "available_instance_capacities": [
            {
```



```
                "flavor": "c6.large.2"
            },
            {
                "flavor": "c6.xlarge.2"
            },
            {
                "flavor": "c6.2xlarge.2"
            },
            {
                "flavor": "c6.3xlarge.2"
            },
            {
                "flavor": "c6.6xlarge.2"
            },
            {
                "flavor": "c6.16xlarge.2"
            }
        ]
    },
    "state": "available",
    "project_id": "9c53a566cb3443ab910cf0daebca90c4",
    "available_vcpus": 20,
    "available_memory": 40960,
    "instance_total": 5,
    "allocated_at": "2016-10-10T14:35:47Z",
    "released_at": "",
    "instance_uuids": [
    "erf5th66cb3443ab912ff0daebca3456",
    "23457h66cb3443ab912ff0daebcaer45"
    ]
  }
}
```


Status Code

See **Status Codes**.

4.3 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 4-6 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**.

Parameter	Type	Mandatory	Description
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 4-7 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 4-8 Response parameters

Parameter	In	Type	Description
servers	body	Array of objects	Specifies the server object. For details, see **Table 4-9**.

Table 4-9 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 4-10**.
created	String	Specifies the time when the ECS was created.
flavor	Object	Specifies the ECS flavor. For details, see **Table 4-11**.

Parameter	Type	Description
id	String	Specifies the ECS ID in UUID format.
name	String	Specifies the ECS name.
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** For details, see **ECS Statuses**.
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. For details, see the table 3 in **ECS Statuses**.
image	Object	Specifies the ECS image. For details, see **Table 4-12**.
metadata	Object	Specifies the ECS metadata. For details, see **Table 4-13**.

Table 4-10 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.

Parameter	Type	Description
OS-EXT-IPS:type	String	Specifies the IP address assignment mode. This is an extended attribute. • **fixed**: indicates a private IP address. • **floating**: indicates a floating IP address.

Table 4-11 flavor field description

Parameter	Type	Description
id	String	Specifies the flavor ID.

Table 4-12 image field description

Parameter	Type	Description
id	String	Specifies the image UUID.

Table 4-13 metadata field description

Parameter	Type	Description
os_type	String	Specifies the OS type. The value can be **Linux** or **Windows**.

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**.

4.4 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 4-14 describes the parameters.

Table 4-14 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 4-15 Request parameters

Parameter	Mandatory	Type	Description
dedicated_host	Yes	Object	Specifies the dedicated host. For details, see **Table 4-16**.

Table 4-16 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**.

4.5 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 4-17 describes the parameters.

Table 4-17 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 4-18 Response parameters

Parameter	In	Type	Description
dedicated_host_types	body	Array of objects	Specifies the available DeH types. For details, see **Table 4-19**.

Table 4-19 dedicated_host_types field description

Parameter	Type	Description
host_type	String	Specifies the DeH type. For details, see **Overview**.

Parameter	Type	Description
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": "c6",
            "host_type_name": "dedicated_general_purpose"
        },
        {
            "host_type": "m6",
            "host_type_name": "memory_optimized"
        }
    ]
}
```

Status Code

See **Status Codes**.

4.6 API Version Query

4.6.1 Querying API Versions

Function

This API is used to query all API versions available to the DeH service.

URI

GET /

Request

None

Response

Table 4-20 Response parameters

Parameter	Type	Description
versions	Array of objects	Specifies the API versions.

Table 4-21 versions field description

Parameter	Type	Description
id	String	Specifies the ID of the API version.
links	Array of objects	Specifies the URL of the API version.
min_version	String	Specifies the microversion. If the APIs of this version support micro-versions, set this parameter to the supported minimum micro-version. If the microversion is not supported, leave this parameter blank.
status	String	Specifies the API version status. • **CURRENT**: indicates a primary version. • **SUPPORTED**: indicates an earlier version which is still supported. • **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the API version update time, which must be UTC time.
version	String	If the APIs of this version support micro-versions, set this parameter to the maximum micro-version supported. If not, leave this parameter blank.

Table 4-22 links field description

Parameter	Type	Description
href	String	Specifies the URL of the API version.
rel	String	Specifies the API URL dependency.

Example Request

Query all available API versions of DeH.

```
GET /
```


Example Response

```
{
   "versions": [
      {
         "id": "v1.0",
         "links": [
            {
               "href": "https//deh.xxx.com/v1.0/",
               "rel": "self"
            }
         ],
         "min_version": "",
         "status": "SUPPORTED",
         "updated": "2016-12-01T11:33:21Z",
         "version": ""
      }
   ]
}
```

Status Code

See **Status Codes**.

4.6.2 Querying an API Version

Function

This API is used to query a specified API version.

URI

GET /{api_version}

Table 4-23 describes the parameters.

Table 4-23 Parameters description

Parameter	Mandatory	Description
api_version	Yes	Specifies the API version, for example, v1.0.

Request

None

Response

Table 4-24 Response parameters

Parameter	Type	Description
version	Object	Specifies information about a specified API version.

Table 4-25 version field description

Parameter	Type	Description
id	String	Specifies the ID of the API version.
links	Array of objects	Specifies the URL of the API version.
min_version	String	Specifies the microversion. If the APIs of this version support micro-versions, set this parameter to the supported minimum micro-version. If the microversion is not supported, leave this parameter blank.
status	String	Specifies the API version status. • **CURRENT**: indicates a primary version. • **SUPPORTED**: indicates an earlier version which is still supported. • **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the API version update time.
version	String	If the APIs of this version support micro-versions, set this parameter to the maximum micro-version supported. If not, leave this parameter blank.

Table 4-26 links field description

Parameter	Type	Description
href	String	Specifies the URL of the API version.
rel	String	Specifies the API URL dependency.

Example Request

Query the information about API v1.0 of DeH.

```
GET /v1.0
```

Example Response

```
{
    "version": {
        "id": "v1.0",
        "links": [
            {
                "href": "https//deh.xxx.com/v1.0/",
                "rel": "self"
            }
        ],
        "min_version": "",
        "status": "SUPPORTED",
```



```
        "updated": "2016-12-01T11:33:21Z",
        "version": ""
    }
}
```

Status Code

See **Status Codes**.

4.7 DeH Tag Management

4.7.1 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**.

Parameter	Type	Mandatory	Description
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: =*<>|/,
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**.

4.7.2 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

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

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**.

4.7.3 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.
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**.

4.7.4 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.

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.

Parameter	Type	Mandatory	Description
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.
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.

Parameter	Type	Mandatory	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_any	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.
resource_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**.

4.8 Quota Configuration

4.8.1 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": [
```



```
        {
            "used": 0,
            "resource": "c6",
            "hard_limit": 5
        },
        {
            "used": 0,
            "resource": "m6",
            "hard_limit": 5
        }
    ]
}
```


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.

Parameter	Type	CRUD	Default Value	Constraint	Description
project_id	String	CR	N/A	N/A	Specifies the tenant who owns the DeH.
host_properties	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** • **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_at	String	R	N/A	N/A	Specifies the time when the DeH is released.
instance_total	Int	R	N/A	N/A	Specifies the total number of ECSs on the DeH.
instance_uuids	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

Parameter	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

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- **Obtain the Project ID by Calling an API**
- **Obtain the Project ID from the Console**

Obtain the Project ID by Calling an API

You can obtain a project ID by calling the API used to **query projects based on 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**.

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

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "project_name",
            "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"
  }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list. On the **API Credentials** page, view the project ID in the project list.

Figure 5-1 Viewing the project ID