Bare Metal Server

API Reference

Issue 03

Date 2023-05-23

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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.

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 Bare Metal Server (BMS). A BMS is a physical server that is dedicated for you on the cloud. It provides the excellent computing performance and data security needed for core databases, key application systems, high-performance computing (HPC), and Big Data services. With the high scalability offered by cloud resources, you can apply for and use BMSs flexibly.

This document describes how to use application programming interfaces (APIs) to perform operations on BMSs, such as creating a BMS, attaching disks to a BMS, and start or stop a BMS. For details about all supported operations, see **API Overview**.

If you plan to access BMSs through an API, ensure that you are familiar with BMS concepts. For details, see **Service Overview**.

1.2 API Calling

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

1.3 Constraints

- The number of BMSs that you can create is determined by your quota. To view or increase the quota, see section "Adjusting Resource Quotas" in *Bare Metal Server User Guide*.
- For more constraints, see API description.

1.4 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. 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.

- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in different enterprise projects are logically isolated.

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

1.5 Selecting an API Type

BMS APIs are classified as follows:

1. APIs for Huawei Cloud services with customized specifications (BMS APIs)
2. Native OpenStack APIs that comply with OpenStack community specifications

The two types of APIs offer similar functions but are used in different application scenarios. Native OpenStack APIs are used to meet open-source ecosystem requirements, while the APIs for the BMS service with customized specifications are developed based on Native OpenStack APIs with enhanced functions, such as creation of yearly/monthly BMSs and enterprise project management.

Currently, BMS APIs are v1 version, and the native OpenStack APIs are v2.1 version. To use native OpenStack APIs, you must be familiar with basic concepts and knowledge of OpenStack. For details, visit **https://www.openstack.org/**.

2 API Overview

API Introduction

Native OpenStack APIs and BMS APIs are provided.

A combination of the two types of APIs allows you to use all functions provided by the BMS service. For example, you can create a BMS using a native OpenStack API or a BMS API.

Table 2-1 API description

Type	Function	Description
BMS APIs	**API version information query**	Query the API version of the BMS service.
	Lifecycle management	Create yearly/monthly BMSs and query BMS details.
	Status management	Change the BMS name, reinstall the BMS OS, start, restart, and stop BMSs.
	Flavor management	Query BMS flavor details and extended flavor information, such as the flavor ID, flavor name, number of CPUs, and boot device.
	NIC management	Query the BMS NIC information, such as the IP address and MAC address of the NIC.
	EVS disk management	Attach EVS disks to or detach EVS disks from a BMS, and query EVS disks attached to a BMS.

Type	Function	Description
	Metadata management	The BMS metadata includes BMS basic information on the cloud platform, such as the BMS ID, hostname, and network information. You can update the BMS metadata.
	Tenant quota management	Query the quotas of all resources of a specified tenant, including used quotas.
	Password management	Query whether one-click password reset is supported. If yes, you can reset the BMS password. Obtain and delete the password of a Windows BMS.
	Task status query	After a task, such as creating a BMS or attaching disks, is delivered, **job_id** is returned, based on which you can query the execution status of the task.
Native OpenStack v2.1 APIs	Lifecycle management	Query BMS details, query BMSs, and query details about BMSs.
	Status management	Start, restart, and stop BMSs.
	Metadata management	The BMS metadata includes BMS basic information on the cloud platform, such as the BMS ID, hostname, and network information. You can query, update, and delete BMS metadata.
	IP address query	Query the private IP address of a BMS, including the IP address version (IPv4 or IPv6) and the specific IP address.
	BMS flavor query	• Querying BMS Flavors: Query all BMS flavors or query specified BMS flavors by filters. • Querying BMS Flavor Details: Query details about a BMS based on the flavor ID, such as the flavor name, number of CPUs, and memory. • Querying Parameter extra_specs of a BMS Flavor: Parameter **extra_specs** specifies the key-value pair of a BMS flavor. If you want to check whether a BMS flavor supports quick provisioning, call this API.
	BMS NIC query	Query all NICs of a BMS or query details about a NIC based on the NIC ID, such as the IP address and MAC address of the NIC.

Type	Function	Description
	EVS disk management	Query all EVS disks attached to a BMS or query details about an EVS disk attached to a BMS based on the disk ID, such as the mount directory.
	SSH key management	Query SSH key pairs or details about an SSH key pair, and create and delete SSH key pairs.
	1D tag management	Add, delete, modify, and query 1D tags of BMSs.

NOTE

- When using BMS APIs, you must use the endpoints of the BMS service.
- When using native OpenStack APIs, you must use the endpoints registered by the ECS service.
- In the current version, long HTTP connections cannot be used to call native OpenStack APIs.

BMS API Constraints

Table 2-2 BMS API constraints

Type	API	URI	Constraint
API version query	Querying API Versions	GET /	2000 times per minute
	Querying an API Version	GET /{api_version}	2000 times per minute
Lifecycle management	Creating a BMS	POST /v1/{project_id}/baremetalservers	50 times per minute
	Querying Details About a BMS	GET /v1/{project_id}/baremetalservers/detail	500 times per minute
	Querying Details About BMSs	GET /v1/{project_id}/baremetalservers/{server_id}	1000 times per minute
Status management	Changing the Name of a BMS	PUT /v1/{project_id}/baremetalservers/{server_id}	100 times per minute
	Reinstalling the BMS OS	POST /v1/{project_id}/baremetalservers/{server_id}/reinstallos	50 times per minute

Type	API	URI	Constraint
	Starting a BMS	POST /v1/{project_id}/baremetalservers/action	50 times per minute
	Restarting a BMS	POST /v1/{project_id}/baremetalservers/action	50 times per minute
	Stopping a BMS	POST /v1/{project_id}/baremetalservers/action	50 times per minute
Flavor management	Querying Flavor Details and Extended Flavor Information	GET /v1/{project_id}/baremetalservers/flavors	500 times per minute
NIC management	Querying BMS NICs	GET /v1/{project_id}/baremetalservers/{server_id}/os-interface	500 times per minute
EVS disk management	Attaching an EVS Disk to a BMS	POST /v1/{project_id}/baremetalservers/{server_id}/attachvolume	100 times per minute
	Detaching an EVS Disk from a BMS	DELETE /v1/{project_id}/baremetalservers/{server_id}/detachvolume/{attachment_id}	100 times per minute
	Querying Disks Attached to a BMS	GET /v1/{project_id}/baremetalservers/{server_id}/os-volume_attachments	500 times per minute
Metadata management	Updating the Metadata of a BMS	POST /v1/{project_id}/baremetalservers/{server_id}/metadata	100 times per minute
Tenant quota management	Querying Tenant Quotas	GET /v1/{project_id}/baremetalservers/limits	500 times per minute
Password management	Querying Whether One-Click Password Reset Is Supported	GET /v1/{project_id}/baremetalservers/{server_id}/os-resetpwd-flag	500 times per minute

Type	API	URI	Constraint
	Resetting the BMS Password with a Few Clicks	-	-
	Obtaining the Password of a Windows BMS	GET /v1/{project_id}/baremetalservers/{server_id}/os-server-password	50 times per minute
	Deleting the Password of a Windows BMS	DELETE /v1/{project_id}/baremetalservers/{server_id}/os-server-password	50 times per minute
Task management	**Querying the Task Status**	GET /v1/{project_id}/jobs/{jobId}	2000 times per minute

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 **Dublin** is **iam.myhuaweicloud.eu**.
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 **Dublin** region, obtain the endpoint of IAM (**iam.myhuaweicloud.eu**) 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.myhuaweicloud.eu/v3/auth/tokens
```

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://iam.myhuaweicloud.eu/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: 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.myhuaweicloud.eu/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 (for example, JSON or XML) 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*, *$ADMIN_PASS* (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.myhuaweicloud.eu/v3/auth/tokens
Content-Type: application/json
```



```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "username",
                    "password": "$ADMIN_PASS",     //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": "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.

BMS 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://iam.myhuaweicloud.eu/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

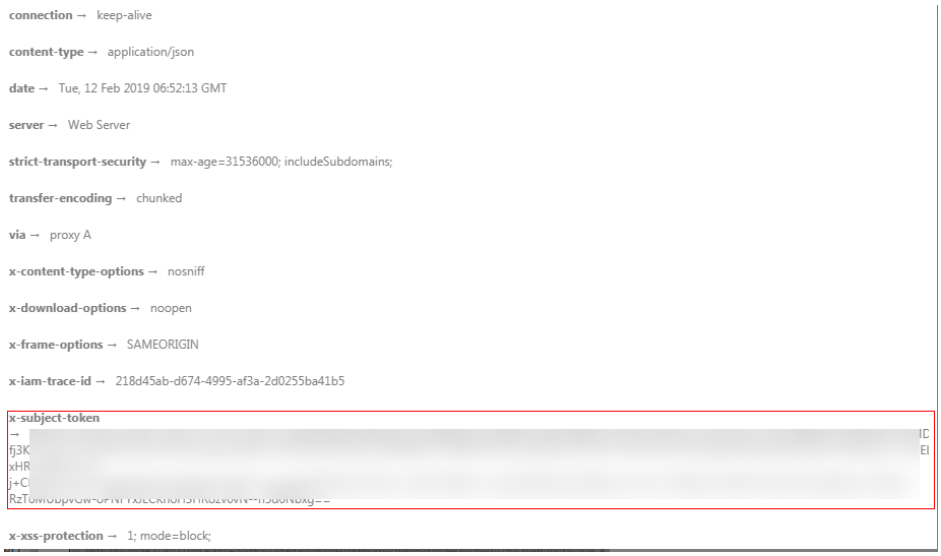
Similar to a request, a response also has a header, for example, **Content-Type**.

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.

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 3-1 Header fields of the response to the request for obtaining a user token



(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 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 BMS APIs

4.1 API Use Instructions

- For details about network service APIs, see **Virtual Private Cloud API Reference**.

4.2 API Version Query

4.2.1 Querying API Versions

Function

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

URI

GET /

Request Parameters

None

Example Request

Querying all available API versions
```
GET https://{BMS Endpoint}/
```

Response Parameters

Parameter	Type	Description
versions	Array of objects	Specifies API versions of the BMS service. For details, see **Table 4-1**.

Table 4-1 versions field data structure description

Parameter	Type	Description
id	String	Specifies the API version ID.
links	Array of objects	Specifies the API URL. For details, see **Table 4-2**.
min_version	String	Specifies the earliest micro API version that is supported.
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 release date of an API version. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2018-09-30T00:00:00Z.
version	String	Specifies the latest micro API version that is supported.

Table 4-2 links field data structure description

Parameter	Type	Description
href	String	Specifies the API URL.
rel	String	Specifies the API URL dependency. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.

Example Response

```
{
    "versions": [
        {
            "id": "v1",
            "links": [
                {
                    "href": "http://bms.xxx.com/v1/",
                    "rel": "self"
                }
            ],
            "min_version": "",
            "status": "CURRENT",
            "updated": "2018-09-30T00:00:00Z",
```

```
        "version": ""
      }
    ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.2.2 Querying an API Version

Function

This API is used to query a specified API version of the BMS service.

URI

GET /{api_version}

Table 4-3 lists the parameters.

Table 4-3 Parameter description

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

Request Parameters

None

Example Request

Querying the API V1

```
GET https://{BMS Endpoint}/v1
```


Response Parameters

Parameter	Type	Description
version	Object	Specifies a specified API version of the BMS service. For details, see Table 4-4.

Table 4-4 version field data structure description

Parameter	Type	Description
id	String	Specifies the API version ID.
links	Array of objects	Specifies the API URL. For details, see Table 4-5.
min_version	String	Specifies the earliest micro API version that is supported.
status	String	Specifies the API version status. • **CURRENT**: indicates a primary version. • **SUPPORTED**: indicates an earlier version that is still supported. • **DEPRECATED**: indicates a deprecated version that may be deleted later.
updated	String	Specifies the release date of an API version. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2018-09-30T00:00:00Z.
version	String	Specifies the latest micro API version that is supported.

Table 4-5 links field data structure description

Parameter	Type	Description
href	String	Specifies the API URL.
rel	String	Specifies the API URL dependency. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.

Example Response

```
{
    "version": {
        "id": "v1",
        "links": [
            {
                "href": "http://bms.xxx.com/v1/",
                "rel": "self"
            }
        ],
        "min_version": "",
        "status": "CURRENT",
        "updated": "2018-09-30T00:00:00Z",
        "version": ""
    }
}
```

Returned Values

Normal values

Return Value	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.3 BMS Lifecycle Management

4.3.1 Creating BMSs

Function

This API is used to create one or more BMSs.

Background for Setting the Login Authentication Mode

Logging in to a BMS can be authenticated using either a key pair or password. For security purposes, you are advised to use key pair authentication.

- Key pair

  A key pair is used for BMS login authentication.

  Method of calling APIs: Use the **key_name** field to specify the key file used for logging in to the BMS. For details about how to use the **key_name** field, see **Table 4-7**.

- Password

  If you choose the initial password for authentication in a BMS, you can log in to a BMS using the username and its initial password. The initial password of user **root** is used for authentication in Linux.

  Methods of calling APIs:

  – Method 1 (recommended): Use the **adminPass** field to specify the initial login password of the specified administrator account. For details about how to use the **adminPass** field, see **Table 4-7**.

  NOTE

  For Linux BMSs with Cloud-Init, if field **user_data** is specified, field **adminPass** is invalid.

  – Method 2:

  ▪ For Linux BMSs with Cloud-Init, use the **user_data** field to inject data. For details, see **Table 4-7**.

NOTE

Public images contain Cloud-Init or Cloudbase-Init by default. For private images, you need to check whether Cloud-Init or Cloudbase-Init is installed.

Constraints

- File injection is not supported.
- Currently, only yearly/monthly BMSs can be created.

URI

POST /v1/{project_id}/baremetalservers

Table 4-6 lists the parameters.

Table 4-6 Parameter description

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

Request Parameters

Parameter	Mandatory	Type	Description
server	Yes	Object	Specifies BMS information. For details, see **Table 4-7**.

Table 4-7 server field data structure description

Parameter	Mandatory	Type	Description
imageRef	Yes	String	Specifies the image ID or image resource URL used for creating the BMS. The ID is in the format of a Universally Unique Identifier (UUID). You can obtain the image ID from the IMS console or by following the instructions in **Querying Images** in *Image Management Service API Reference*. When using an API to query images, you can add the field **?virtual_env_type=Ironic** to filter BMS images.
flavorRef	Yes	String	Specifies the flavor ID of the BMS. The format is **physical.***x.x*. You can obtain the flavor ID from the BMS console or using the **Querying Flavor Details and Extended Flavor Information** API. **NOTE** For details about the images supported by different BMS flavors, see **OSs Supported by Different Types of BMSs**. Only DeC tenants can apply for BMSs using the physical.*x.x*.hba flavor, and only DESS disks can be attached to BMSs using this flavor.
name	Yes	String	Specifies the BMS name. Value range: • The value can contain a maximum of 63 characters consisting of letters (case-insensitive), digits, underscores (_), hyphens (-), and periods (.). • If more than one BMS is to be created, tags similar to **-0000** will be automatically added to the end of the BMS names during the creation. In this case, the BMS name contains 1 to 58 characters.
metadata	Yes	Object	Specifies the BMS metadata. The maximum size for both the metadata **key** and **value** is 255 characters. For details, see **Table 4-8**.

Parameter	Mandatory	Type	Description

user_data	No	String	Specifies the user data to be injected during the BMS creation. Text can be injected.
			Constraints:
			• The content to be injected must be encoded with base64. The maximum size of the content to be injected (before encoding) is 32 KB.
			• If **key_name** is not specified, the password of user **root** for logging in to the BMS will be injected by default.
			• This parameter is mandatory when you create a Linux BMS using password authentication. Its value is the initial user **root** password.
			Password complexity requirements:
			• Contains 8 to 26 characters.
			• Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters !@$%^-_=+[{}]:,./?
			An example is as follows:
			`#!/bin/bash`
			`echo 'root:$6$V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig' | chpasswd -e`
			where, **6V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig** is the ciphertext password, which can be generated as follows:
			1. Generate an encrypted salt value.
			`[root@test linux]# python -c "import crypt, getpass, pwd;print crypt.mksalt()"`
			`$6$V6azyeLwcD3CHlpY`
			2. Generate a ciphertext password based on the salt value.
			`[root@test linux]# python -c "import crypt, getpass, pwd;print crypt.crypt('Cloud.1234','\$6\$V6azyeLwcD3CHlpY')"`
			`$6$V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig`
			NOTE
			Data injection is not supported for BMSs that use a Linux image and the password login mode. For details about how to inject user data, see **Injecting User Data into BMSs**.

Parameter	Mandatory	Type	Description
adminPass	No	String	Specifies the initial login password of the administrator account for logging in to a BMS using password authentication. The Linux administrator is **root**. Password complexity requirements: • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters !@$%^-_=+[{}]:,./? • (Linux OSs) Cannot contain the username or the username in reverse.
key_name	No	String	Specifies the name of a key pair. This is an extended attribute. To log in to a BMS using an SSH key pair, set the value to the name of an existing private key. You can create a key pair using the **Creating or Importing an SSH Key Pair (Native OpenStack API)** API, or query existing key pairs using the **Querying SSH Key Pairs (Native OpenStack API)** API. Constraints: • If both **key_name** and **user_data** are specified, **user_data** only injects user data. • When a Windows image is used to create a BMS, this parameter cannot be left blank because the Windows image supports only login using a key. • To log in to a Windows BMS, decrypt the key pair to obtain a password and use it to remotely log in to the BMS. For details, see **Login Using an MSTSC Password**.
security_groups	No	Array of objects	Specifies security groups of the BMS. For details, see **Table 4-9**.
nics	Yes	Array of objects	Specifies NICs of the BMS. For details, see **Table 4-10**. Constraints: A maximum of two NICs can be attached to a BMS. The first will be used as the primary NIC. If multiple NICs are specified, ensure that all NICs belong to the same VPC.
availability_zone	Yes	String	Specifies the name of the AZ where the BMS is located.

Parameter	Mandatory	Type	Description
vpcid	Yes	String	Specifies the ID of the VPC to which the BMS belongs. The value is in UUID format. You can obtain the VPC ID from the network console or by following the instructions in **Querying VPC Details** of *Virtual Private Cloud API Reference*.
publicip	No	Object	Specifies the EIP information of the BMS. Possible values include: • **Do not use** (This parameter is unavailable.) • **Automatically assign**: Assign a new EIP. • **Specify**: Specify an EIP that has been created. For details, see **Table 4-11**.
count	No	Integer	Specifies the number of BMSs to be created. Constraints: • If this parameter is not specified, the default value is **1**. • If the quota is sufficient, the maximum value is **100**.
root_volume	No	Object	Specifies system disk configurations of the BMS. For details, see **Table 4-12**. Constraints: If the flavor supports quick provisioning, this parameter is mandatory. Otherwise, this parameter is not required. For how to check whether a flavor supports quick provisioning, see **Querying Details About extra_specs Parameters of a BMS Flavor (Native OpenStack API)**.
data_volumes	No	Array of objects	Specifies data disk configurations of the BMS. Each data structure represents a data disk to be created. For details, see **Table 4-13**. Constraints: A maximum of 60 EVS disks (including the system disk and data disks) can be attached to a BMS.
extendparam	Yes	Object	Specifies the supplementary for creating the BMS. For details, see **Table 4-14**.

Parameter	Mandatory	Type	Description
schedulerHints	No	Object	Specifies scheduling information of the BMS. This parameter is mandatory for creating a BMS in a DeC. For details, see **Table 4-15**.
server_tags	No	List <resource_tag>	Specifies tags of the BMS. For details, see **Table 4-16**. **NOTE** A maximum of 10 tags can be added to a BMS. **__type_baremetal** is an internal tag of the system. Therefore, you can add a maximum of nine tags.

Table 4-8 metadata field data structure description

Parameter	Mandatory	Type	Description
op_svc_userid	Yes	String	Specifies the user ID. You can obtain the user ID from **My Credential** on the management console.
BYOL	No	String	Specifies whether a license is provided. The value can be **true** or **false**.
admin_pass	No	String	Specifies the initial password of the administrator account for logging in to a BMS. For Linux, the administrator account is **root**. Password complexity requirements: • Contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters !@$%^-_=+[{}]:,./? • Cannot contain the username, the username in reverse, or more than two consecutive characters in the username.

Parameter	Mandatory	Type	Description
agency_name	No	String	Specifies the IAM agency name. An agency provides a temporary security credential for accessing a BMS. The agency is created by the tenant administrator on the IAM console. **NOTE** To obtain and update an agency, perform the following steps: 1. You can obtain a valid agency name by using the **Listing Agencies** API provided by IAM. 2. Use the **Updating the Metadata of a BMS** API to change the value of field **agency_name** in the metadata to the new agency name.

Table 4-9 security_groups field data structure description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the security group ID, which takes effect for all NICs configured for the BMS. • If this parameter is not specified, the default security group will be bound to the BMS. • If this parameter is required (in UUID format), use the ID of an existing security group. For details about how to obtain existing security groups, see **Querying Security Groups** in *Virtual Private Cloud API Reference*.

Table 4-10 nics field data structure description

Parameter	Mandatory	Type	Description
subnet_id	Yes	String	Specifies the subnet information of a BMS NIC. The value must be the ID of the subnet (**network_id**) created in the VPC specified by **vpcid** and in the format of UUID. You can obtain the subnet ID (**network_id**) from the VPC console or by following the instructions in **Querying Subnets** in *Virtual Private Cloud API Reference*.

Parameter	Mandatory	Type	Description
ip_address	No	String	Specifies the IPv4 address of a BMS NIC. Constraints: • If this parameter is left blank or set to "", an unused IP address in the subnet of this network is automatically assigned as the IP address of the NIC. • If this parameter is specified, its value must be an unused IP address in the network segment of the subnet. • The IP address cannot be specified when you create BMSs in a batch.

Table 4-11 publicip field data structure description

Parameter	Mandatory	Type	Description
id	No	String	Specifies the ID of an existing EIP assigned to the BMS. The value is in UUID format. You can obtain the EIP ID from the network console or by following the instructions in **Querying EIPs** in *Elastic IP API Reference*. Constraints: • Only EIPs in the **DOWN** state can be assigned. • Existing EIPs cannot be used for creating BMSs in a batch. That is, this parameter is invalid in such a case.
eip	No	Object	Specifies the configuration for creating an EIP that will be automatically assigned to the BMS. For details, see **Table 4-17**.

NOTE

You can configure either but not both of **id** and **eip** in the **publicip** field.

Table 4-12 root_volume field data structure description

Parameter	Mandatory	Type	Description
volumetype	Yes	String	Specifies the BMS system disk type. The disk type must match the available disk type. ● GPSSD: general-purpose SSD disk type ● SAS: high I/O disk type ● SSD: ultra-high I/O disk type
size	Yes	Integer	Specifies the system disk size (GB). The value ranges from **40** to **1024**. Constraints: The system disk size must be greater than or equal to the minimum system disk size of the image (**min_disk** attribute).
cluster_id	No	String	Specifies the ID of the storage pool to which the BMS system disk belongs. **NOTE** ● This parameter is required when Dedicated Distributed Storage Service (DSS) is used.
cluster_type	No	String	Specifies the storage type of the BMS system disk. An example value is **DSS**, which indicates the DSS service. **NOTE** ● This parameter is required when DSS is used.

Table 4-13 data_volumes field data structure description

Parameter	Mandatory	Type	Description
volumetype	Yes	String	Specifies the BMS data disk type. The disk type must match the available disk type. ● GPSSD: general-purpose SSD disk type ● SAS: high I/O disk type ● SSD: ultra-high I/O disk type
size	Yes	Integer	Specifies the data disk size (GB). The value ranges from **10** to **32768**.
shareable	No	Boolean	Specifies whether the disk is shareable. ● **true**: shared EVS disk ● **false**: common EVS disk The default value is **false**.

Parameter	Mandatory	Type	Description
cluster_id	No	String	Specifies the ID of the storage pool to which BMS data disks belong. **NOTE** • This parameter is required when Dedicated Distributed Storage Service (DSS) is used.
cluster_type	No	String	Specifies the storage type of BMS data disks. An example value is **DSS**, which indicates the DSS service. **NOTE** • This parameter is required when DSS is used.

Table 4-14 extendparam field data structure description

Parameter	Mandatory	Type	Description
chargingMode	No	String	Specifies the billing mode. Value range: **prePaid** (default): yearly/monthly billing
regionID	No	String	Specifies the ID of the region where the BMS resides.
periodType	No	String	Specifies the subscription period. Value range: • **month**: indicates that the subscription unit is month. • **year**: indicates that the subscription unit is year. **NOTE** This parameter is valid and mandatory if **chargingMode** is set to **prePaid**.
periodNum	No	Integer	Specifies the number of subscription periods. Value range: • If **periodType** is **month**, the value ranges from **1** to **9**. • If **periodType** is **year**, the value is **1**. **NOTE** This parameter is valid and mandatory if **chargingMode** is set to **prePaid**.

Parameter	Mandatory	Type	Description
isAutoRenew	No	String	Specifies whether auto-renewal is enabled. • **true**: indicates that auto-renewal is enabled. • **false**: indicates that auto-renewal is disabled. **NOTE** This parameter is valid when **chargingMode** is set to **prePaid**. If this parameter is not specified or is left blank, auto-renewal is disabled.
isAutoPay	No	String	Specifies whether the order is automatically or manually paid. • **true**: The order will be automatically paid. • **false**: You must manually pay the order. **NOTE** This parameter is valid when **chargingMode** is set to **prePaid**. If this parameter is not specified or is left blank, manual payment is required.

Table 4-15 schedulerHints field data structure description

Parameter	Mandatory	Type	Description
dec_baremetal	No	String	Specifies whether to create the BMS in a DeC. The value can be **share** or **dedicate**. Constraints: • If this parameter is not specified, the default value is **share**. • To create a BMS in a DeC, set this parameter to **dedicate**.

Table 4-16 server_tags field data structure description

Parameter	Mandatory	Type	Description
key	Yes	String	Specifies the tag key. • It contains a maximum of 36 Unicode characters and cannot be empty. • It cannot contain ASCII characters (0–31) or special characters =*<>\,|/ • The tag key of a BMS must be unique.

Parameter	Mandatory	Type	Description
value	No	String	Specifies the tag value. • Each value contains a maximum of 43 Unicode characters and can be an empty string. • It cannot contain ASCII characters (0–31) or special characters =*<>\,|/

Table 4-17 eip field data structure description

Parameter	Mandatory	Type	Description
iptype	Yes	String	Specifies the EIP type. Enumerated values: **5_bgp** and **5_sbgp** For details, see the **publicip** field in **Assigning an EIP** in *Elastic IP API Reference*.
bandwidth	Yes	Object	Specifies the EIP bandwidth. For details, see **Table 4-18**.
extendparam	Yes	Object	Provides additional information about the EIP. For details, see **Table 4-19**.

Table 4-18 bandwidth field data structure description

Parameter	Mandatory	Type	Description
name	No	String	Specifies the bandwidth name.
sharetype	Yes	String	Specifies the bandwidth sharing type. Value **PER** indicates dedicated bandwidth and **WHOLE** indicates shared bandwidth.
id	No	String	Specifies the shared bandwidth ID. You can specify an existing shared bandwidth when applying for an EIP with a **WHOLE** bandwidth. **NOTE** This parameter is mandatory when **sharetype** is set to **WHOLE**.

Parameter	Mandatory	Type	Description
size	Yes	Integer	• The value ranges from 5 Mbit/s to 2000 Mbit/s by default. (The specific range may vary depending on the configuration in each region. You can view the bandwidth range of each region on the management console.) • Specifies the bandwidth (Mbit/s). The minimum shared bandwidth is 5 Mbit/s by default. **NOTE** • If a decimal fraction (for example **10.2**) or a character string (for example **10**) is specified, the specified value will be automatically converted to an integer. If the bandwidth is less than 300 Mbit/s, the step is 1 Mbit/s. If the bandwidth is from 300 Mbit/s to 1000 Mbit/s, the step is 50 Mbit/s. If the bandwidth is from 1000 Mbit/s to 2000 Mbit/s, the step is 1000 Mbit/s. • This parameter is mandatory when **sharetype** is set to **PER** and is optional when **sharetype** is set to **WHOLE** with an ID specified.
chargemode	No	String	Specifies the bandwidth billing mode. The value can be **traffic** or **bandwidth**. • If this field is not specified, the BMS is billed by bandwidth. • If the field value is empty, the BMS is billed by bandwidth.

Table 4-19 extendparam field data structure description for assigning an EIP

Parameter	Mandatory	Type	Description
chargingMode	Yes	String	Specifies the billing mode of an EIP. If bandwidth is charged by **bandwidth**, both **prePaid** and **postPaid** will be available for EIP. If bandwidth is charged by **traffic**, only **postPaid** will be available for EIP. Value range: • **prePaid**: yearly/monthly billing mode. • **postPaid**: indicates the pay-per-use billing mode.

Example Request

- Creating a BMS (image ID: 766a545a-02e1-433e-b1d1-733b5dc95e94; flavor: physical.s3.large; billing: yearly/monthly; remote login: password)

```
POST https://{BMS Endpoint}/v1/{project_id}/baremetalservers
{
    "server": {
        "availability_zone": "az1",
        "name": "bms-75c7",
        "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
        "flavorRef": "physical.s3.large",
        "data_volumes": [
            {
                "volumetype": "SAS",
                "size": 10,
                "shareable": false
            }
        ],
        "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
        "nics": [
            {
                "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616",
                "ip_address": ""
            }
        ],
        "count": 1,
        "schedulerHints": {
            "dec_baremetal": "share"
        },
        "extendparam": {
            "chargingMode": "prePaid",
            "periodType": "month",
            "periodNum": 1,
            "isAutoRenew": "true",
            "isAutoPay": "true",
            "regionID": "region01"
        },
        "metadata": {
            "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81",
            "BYOL": "false",
            "admin_pass": "$ADMIN_PASS"
        },
        "publicip": {
            "id": "1b036bce-26b4-465c-acea-99fdbb0573a3"
```



```
        },
        "user_data": "$USER_DATA",
        "security_groups": [
            {
                "id": "993f2916-5bad-4f24-bc3c-db539f1a6d19"
            }
        ]
    }
}
```

- Creating a Linux BMS in AZ1 (name: bms-3b91; image ID: 766a545a-02e1-433e-b1d1-733b5dc95e94; flavor: physical.o2.large; billing: yearly/monthly; remote login: key pair)

```
{
    "server": {
        "availability_zone": "az1",
        "name": "bms-3b91",
        "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
        "flavorRef": "physical.o2.large",
        "data_volumes": [
            {
                "volumetype": "SAS",
                "size": 10,
                "shareable": true
            }
        ],
        "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
        "nics": [
            {
                "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616",
                "ip_address": ""
            },
            {
                "subnet_id": "2b8bcf33-33f2-4a63-aa9c-b1e17ff0ed7f",
                "ip_address": ""
            }
        ],
        "count": 1,
        "schedulerHints": {
            "dec_baremetal": "share"
        },
        "extendparam": {
            "chargingMode": "prePaid",
            "periodType": "month",
            "periodNum": 1,
            "isAutoRenew": "true",
            "isAutoPay": "true",
            "regionID": "region01"
        },
        "metadata": {
            "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81",
            "BYOL": "false"
        },
        "publicip": {
            "eip": {
                "iptype": "5_bpg",
                "bandwidth": {
                    "chargemode": "",
                    "name": "bms-3b91-bandwidth",
                    "size": 1,
                    "sharetype": "PER"
                },
                "extendparam": {
                  "chargingMode": "prePaid"
                }
            }
        },
        "security_groups": [
            {
```

```
            "id": "993f2916-5bad-4f24-bc3c-db539f1a6d19"
          }
        ],
        "key_name": "$key_name"
    }
}
```

The following is an example of **user_data** generated by Java:

```
String userData_org = "#!/bin/bash \r\n echo    'root:xxxxx' | chpasswd ;";
byte[] userData_byte = userData_org.getBytes();
String userData = new BASE64Encoder().encode(userData_byte);
```

- Creating a Windows BMS in AZ1 (name: bms-6a1e; image ID: fd04041a-5f35-45ae-a27c-bf30f921c6b5; billing: yearly/monthly; flavor: physical.s3.largecharged; remote login: key pair)

```
{
    "server": {
        "availability_zone": "az1",
        "name": "bms-6a1e",
        "imageRef": "fd04041a-5f35-45ae-a27c-bf30f921c6b5",
        "flavorRef": "physical.s3.large",
        "data_volumes": [
            {
                "volumetype": "SAS",
                "size": 10,
                "shareable": false
            }
        ],
        "vpcid": "0adc2847-c160-4263-be1f-e03d36e93e32",
        "nics": [
            {
                "subnet_id": "4f1cf6cc-4cc3-41d3-8192-625293dccce0",
                "ip_address": ""
            }
        ],
        "count": 1,
        "schedulerHints": {
            "dec_baremetal": "share"
        },
        "extendparam": {
            "chargingMode": "prePaid",
            "periodType": "month",
            "periodNum": 1,
            "isAutoRenew": "true",
            "isAutoPay": "true",
            "regionID": "region1"
        },
        "metadata": {
            "op_svc_userid": "8a74baed155b47e8b59fbca853d6cb48",
            "BYOL": "false"
        },
        "publicip": {
            "id": "db9c4f39-fabd-4cc2-8685-82e7715d099f"
        },
        "user_data": "$USER_DATA",
        "security_groups": [
            {
                "id": "33cf5e40-0b65-4e03-867d-4c67947c49be"
            }
        ],
        "key_name": "$key_name"
    }
}
```


Response Parameters

Table 4-20 Normal response

Parameter	Type	Description
order_id	String	Specifies the order ID returned after an order is submitted. You can query the order processing progress based on the ID. To query the order processing progress, log in to the management console and choose **Billing** > **My Orders**.
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

Table 4-21 Abnormal response

Parameter	Type	Description
error	Dictionary data structure [1]	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-22**.

Table 4-22 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "order_id": "CS2009141523OQSEQ",
    "job_id": "ff808081748b760c01748b7f80370003"
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.3.2 Querying BMS Details

Function

This API is used to query BMS details.

The information that can be queried includes the BMS billing mode and whether the BMS is frozen.

URI

GET /v1/{project_id}/baremetalservers/{server_id}

Table 4-23 lists the parameters.

Table 4-23 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying details about a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/
53206ed0-56de-4d6b-b7ee-ffc62ca26f43
```

Response Parameters

Parameter	Type	Description
server	Object	Specifies BMS information. For details, see **Table 4-24**.

Table 4-24 server field data structure description

Parameter	Type	Description
id	String	Specifies the BMS ID in UUID format.
user_id	String	Specifies the ID of the user to which the BMS belongs. The value is in UUID format.
name	String	Specifies the BMS name.
created	String	Specifies the time when the BMS was created. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.
updated	String	Specifies the time when the BMS was updated. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T04:30:52Z.
tenant_id	String	Specifies the ID of the tenant owning the BMS. The value is in UUID format. This parameter specifies the same meaning as **project_id**.
hostId	String	Specifies the host ID of the BMS.
addresses	Map<String,Array of **address** objects>	Specifies the VPC used by the BMS. **key**: indicates the ID of the VPC used by the BMS. **value** indicates the VPC details.
key_name	String	Specifies the key pair that is used to authenticate the BMS.
image	Object	Specifies the BMS image. For details, see **Table 4-26**.
flavor	Object	Specifies the BMS flavor. For details, see **Table 4-28**.

Parameter	Type	Description
security_groups	Array of objects	Specifies the security groups to which the BMS belongs. For details, see **Table 4-29**.
accessIPv4	String	This is a reserved attribute.
accessIPv6	String	This is a reserved attribute.
status	String	Specifies the BMS status. Value range: • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
progress	Integer	This is a reserved attribute.
config_drive	String	Specifies whether to configure the config drive partition for the BMS. The value can be **True** or an empty string.
metadata	object	Specifies the BMS metadata. For details, see **Table 4-33**. **NOTE** Metadata includes system default fields and the fields set by users.

Parameter	Type	Description
OS-EXT-STS:task_state	String	Specifies the BMS task status. This is an extended attribute. Examples: • **rebooting**: The BMS is being restarted. • **reboot_started**: The BMS is normally restarted. • **reboot_started_hard**: The BMS is forcibly restarted. • **powering-off**: The BMS is being powered off. • **powering-on**: The BMS is being powered on. • **rebuilding**: The BMS is being rebuilt. • **scheduling**: The BMS is being scheduled. • **deleting**: The BMS is being deleted.
OS-EXT-STS:vm_state	String	Specifies the BMS status. This is an extended attribute. Examples: • **active**: The BMS is running. • **stopped**: The BMS is stopped.
OS-EXT-SRV-ATTR:host	String	Specifies the hostname of the BMS. This is an extended attribute.
OS-EXT-SRV-ATTR:instance_name	String	Specifies the BMS alias. This is an extended attribute.
OS-EXT-STS:power_state	Integer	Specifies the BMS power status. This is an extended attribute. Examples: • **0**: indicates **NO STATE**. • **1**: indicates **RUNNING**. • **4**: indicates **SHUTDOWN**.
OS-EXT-SRV-ATTR:hypervisor_hostname	String	Specifies the hostname of the hypervisor.
OS-EXT-AZ:availability_zone	String	Specifies the AZ where the BMS is located.

Parameter	Type	Description
OS-DCF:diskConfig	String	Specifies the disk configuration. This is an extended attribute. The value can be: • **MANUAL**: The API uses the partitioning scheme in the image and the file system to create a BMS. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The API uses a single partition with the same size as the disk of the target flavor to create a BMS. The API automatically adjusts the file system to adapt to the entire partition.
fault	Object	Specifies the cause of a BMS fault. For details, see **Table 4-30**.
OS-SRV-USG:launched_at	String	Specifies the time when the BMS was started. The timestamp format is ISO 8601, for example, **2019-05-22T03:23:59.000000**.
OS-SRV-USG:terminated_at	String	Specifies the time when the BMS was deleted. The timestamp format is ISO 8601, for example, **2019-05-22T04:23:59.000000**.
os-extended-volumes:volumes_attached	Array of objects	Specifies the disks attached to the BMS. For details, see **Table 4-31**.
description	String	Specifies BMS information.
host_status	String	Specifies the status of the host that accommodates the BMS. • **UP**: The nova-compute status is normal. • **UNKNOWN**: The nova-compute status is unknown. • **DOWN**: the nova-compute status is abnormal. • **MAINTENANCE**: The nova-compute is in the maintenance state. • Empty string: The BMS does not have host information.

Parameter	Type	Description
OS-EXT-SRV-ATTR:hostname	String	Specifies the hostname of the BMS.
OS-EXT-SRV-ATTR:reservation_id	String	Specifies the reserved BMS IDs in a batch creation situation. When BMSs are created in a batch, they have the same **reservation_id**.
OS-EXT-SRV-ATTR:launch_index	Integer	Specifies the BMS startup sequence in a batch creation situation.
OS-EXT-SRV-ATTR:kernel_id	String	Specifies the UUID of the kernel image when the AMI image is used. In other scenarios, leave this parameter blank.
OS-EXT-SRV-ATTR:ramdisk_id	String	Specifies the UUID of the Ramdisk image when the AMI image is used. In other scenarios, leave this parameter blank.
OS-EXT-SRV-ATTR:root_device_name	String	Specifies the device name of the BMS system disk, for example, **/dev/sda**.
OS-EXT-SRV-ATTR:user_data	String	Specifies the **user_data** specified during BMS creation. The value is encoded using Base64 or an empty string.
locked	Boolean	Specifies whether the BMS is locked. • **true**: The BMS is locked. • **false**: The BMS is not locked.
tags	Array of strings	Specifies tags of the BMS.
os:scheduler_hints	Object	Specifies scheduling information of the BMS. For details, see **Table 4-32**.
enterprise_project_id	String	Specifies the ID of the enterprise project to which the BMS belongs.
sys_tags	Array of objects	Specifies system tags of the BMS. For details, see **Table 4-34**.

Table 4-25 address field data structure description

Parameter	Type	Description
version	String	Specifies the IP address version. • **4**: indicates IPv4. • **6**: indicates IPv6.

Parameter	Type	Description
addr	String	Specifies the IP address.
OS-EXT-IPS:type	String	Specifies the IP address type. • **fixed**: indicates the private IP address. • **floating**: indicates the EIP.
OS-EXT-IPS-MAC:mac_addr	String	Specifies the MAC address.
OS-EXT-IPS:port_id	String	Specifies the port ID corresponding to the IP address.

Table 4-26 image field data structure description

Parameter	Type	Description
id	String	Specifies the image ID in UUID format.

Table 4-27 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name.
href	String	Specifies the corresponding shortcut link.

Table 4-28 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the flavor ID of the BMS.
name	String	Specifies the flavor name of the BMS.
disk	String	Specifies the system disk size in the BMS flavor. The value **0** indicates that the disk size is not limited.
vcpus	String	Specifies the number of CPU cores in the BMS flavor.

Parameter	Type	Description
ram	String	Specifies the memory size (MB) in the BMS flavor.

Table 4-29 security_groups field data structure description

Parameter	Type	Description
name	String	Specifies the security group name.
id	String	Specifies the security group ID.

Table 4-30 fault field data structure description

Parameter	Type	Description
message	String	Specifies the fault information.
code	Integer	Specifies the fault code.
details	String	Specifies the fault details.
created	String	Specifies the time when the fault occurred. The time is in ISO 8601 format.

Table 4-31 os-extended-volumes:volumes_attached field data structure description

Parameter	Type	Description
id	String	Specifies the disk ID in UUID format.
delete_on_termination	String	Specifies whether to delete the disk when deleting the BMS. • **true**: Yes • **false**: No
bootIndex	String	Specifies whether it is a boot disk. **0** specifies a boot disk, and **-1** specifies a non-boot disk.
device	String	Specifies the device name of the disk, for example, **/dev/sdb**.

Table 4-32 os:scheduler_hints field data structure description

Parameter	Type	Description
dec_baremetal	Array of strings	Specifies whether to create the BMS in a DeC. The value can be **share** or **dedicate**.

Table 4-33 metadata field data structure description

Parameter	Type	Description
chargingMode	String	Specifies the billing type of the BMS. **1**: yearly/monthly billing mode (pre-paid)
metering.order_id	String	Specifies the order ID for a yearly/monthly BMS.
metering.product_id	String	Specifies the product ID for a yearly/monthly BMS.
vpc_id	String	Specifies the ID of the VPC where the BMS is located.
metering.image_id	String	Specifies the image ID of the BMS.
metering.imagetype	String	Specifies the image type. The following types are supported: • Public image: The value is **gold**. • Private image: The value is **private**. • Shared image: The value is **shared**.
baremetalPortIDList	String	Specifies NICs of the BMS.
metering.resourcespeccode	String	Specifies the flavor code of the BMS in *{Flavor ID}.{os_type}* format, for example, **physical.o2.medium.linux**.
metering.resourcetype	String	Specifies the resource type of the BMS. The value is **__type_baremetal**.
image_name	String	Specifies the image name of the BMS.

Parameter	Type	Description
op_svc_userid	String	Specifies the user ID. You can obtain the user ID from **My Credential** on the management console.
os_type	String	Specifies the OS type. The value can be **Linux** or **Windows**.
__bms_support_evs	String	Specifies whether the BMS supports EVS disks.
os_bit	String	Specifies the number of bits in the OS: **32** or **64**.

Table 4-34 sys_tags field data structure description

Parameter	Type	Description
key	String	Specifies the key of a system tag. The value is **_sys_enterprise_project_id**, indicating an enterprise project ID.
value	String	Specifies the system tag value.

Example Response

```
{
   "server": {
      "id": "53206ed0-56de-4d6b-b7ee-ffc62ca26f43",
      "name": "bms-test",
      "addresses": {
         "5849fdf1-9d79-4589-80c2-fe557990c417": [
            {
               "version": "4",
               "addr": "192.168.1.216",
               "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:25:56:c3",
               "OS-EXT-IPS:port_id": "9e62503b-094e-4c6e-bc95-f10bbfb455d5",
               "OS-EXT-IPS:type": "fixed"
            },
            {
               "version": "4",
               "addr": "10.154.72.77",
               "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:25:56:c3",
               "OS-EXT-IPS:port_id": "9e62503b-094e-4c6e-bc95-f10bbfb455d5",
               "OS-EXT-IPS:type": "floating"
            }
         ]
       },
      "flavor": {
         "disk": "9309",
         "vcpus": "32",
         "ram": "193047",
         "id": "physical.o2.medium",
         "name": "physical.o2.medium"
      },
```



```
    "accessIPv4": "",
    "accessIPv6": "",
    "status": "ACTIVE",
    "progress": 0,
    "hostId": "cd243addb5d2c64e89218180b7a3ed95abe6882e81c337cc563137df",
    "updated": "2018-09-18T09:29:44Z",
    "created": "2018-09-18T02:43:26Z",
    "metadata": {
        "metering.order_id": "CS1809181040HCFC2",
        "metering.image_id": "8589958c-6bc7-40c6-b81b-3fe4cd2e3d85",
        "baremetalPortIDList": "[9e62503b-094e-4c6e-bc95-f10bbfb455d5]",
        "metering.resourcespeccode": "physical.o2.medium.linux",

        "image_name": "redhat_7_2",
        "op_svc_userid": "745973c535ec4d4caba86f6f9419ff6d",
        "metering.resourcetype": "__type_baremetal",

        "metering.product_id": "00301-74069-0--0",
        "os_bit": "64",
        "vpc_id": "5849fdf1-9d79-4589-80c2-fe557990c417",
        "os_type": "Linux",
        "chargingMode": "1"
    },
    "tags": [
        "__type_baremetal"
    ],
    "description": "bms-6e18",
    "locked": false,
    "image": {
        "id": "8589958c-6bc7-40c6-b81b-3fe4cd2e3d85"
    },
    "config_drive": "",
    "tenant_id": "bbf1946d374b44a0a2a95533562ba954",
    "user_id": "3fc5ab2b0c544979abcaafd86edd80e6",
    "OS-EXT-STS:power_state": 1,
    "OS-EXT-STS:vm_state": "active",
    "OS-EXT-SRV-ATTR:host": "bms.dc1",
    "OS-EXT-SRV-ATTR:instance_name": "instance-00154f94",
    "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova002@2",
    "OS-DCF:diskConfig": "MANUAL",
    "OS-EXT-AZ:availability_zone": "az-dc-1",
    "os:scheduler_hints": {},
    "OS-EXT-SRV-ATTR:root_device_name": "/dev/sda",
    "OS-EXT-SRV-ATTR:ramdisk_id": "",
    "enterprise_project_id": "2c7b6d77-9013-4d74-a221-2f612fc56372",
    "OS-EXT-SRV-ATTR:user_data": "$USER_DATA",
    "OS-SRV-USG:launched_at": "2018-09-18T02:46:07.954587",
    "OS-EXT-SRV-ATTR:kernel_id": "",
    "OS-EXT-SRV-ATTR:launch_index": 0,
    "host_status": "UP",
    "OS-EXT-SRV-ATTR:reservation_id": "r-08tuyo8v",
    "OS-EXT-SRV-ATTR:hostname": "bms-6e18",
    "sys_tags": [
        {
            "key": "_sys_enterprise_project_id",
            "value": "2c7b6d77-9013-4d74-a221-2f612fc56372"
        }
    ]
  }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.3.3 Querying Details About BMSs

Function

This API is used to query BMSs by filters and display details about the BMSs.

The information that can be queried includes the BMS billing mode and whether the BMS is frozen.

URI

GET /v1/{project_id}/baremetalservers/detail

Table 4-35 lists the parameters.

Table 4-35 Parameter description

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

Table 4-36 Parameters for querying BMS details

Parameter	Mandatory	Type	Description
flavor	No	String	Specifies the ID of the BMS flavor.
name	No	String	Specifies the BMS name.

Parameter	Mandatory	Type	Description
status	No	String	Specifies the BMS status. Value range: • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
limit	No	Integer	Specifies the number of BMSs displayed on each page. The default value is **25** and maximum value is **1000**.

Parameter	Mandatory	Type	Description
offset	No	Integer	This API is a pagination query API. **offset** indicates the page number (the start page number is **1**). The returned value contains the number of BMSs and details about the BMSs. ● If **offset** is specified: BMSs are displayed by pages based on the **limit** value. By default, the **limit** value is **1000**. The BMSs and total number of BMSs on the **offset** page are displayed. The maximum number of BMSs is the value of **limit**. If the number of BMSs is less than the value of **limit**, the actual number of BMSs is displayed. ● If **offset** is not specified: – If **limit** is specified: The BMSs and total number of BMSs are displayed. The maximum number of BMSs is the value of **limit**. If the number of BMSs is less than the value of **limit**, the actual number of BMSs is displayed. – If **limit** is not specified: 25 BMSs are displayed on each page. Details about the BMSs on the first page are displayed. If the number of BMSs is less than 25, the actual number of BMSs is displayed.
tags	No	String	Specifies the BMS tag. The value can be: ● **__type_baremetal**: internal tag of the system ● Other custom tags
reservation_id	No	String	Specifies the reserved ID, which can be used to query BMSs created in a batch.

Parameter	Mandatory	Type	Description
detail	No	String	Specifies the level for details about BMS query results. A higher level indicates more details about BMS query results. Available levels include 4, 3, 2, and 1. The default level is 4.

Table 4-37 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the flavor ID of a BMS.
name	String	Specifies the flavor name of a BMS.
disk	String	Specifies the system disk size in the BMS flavor. The value **0** indicates that the disk size is not limited.
vcpus	String	Specifies the number of CPU cores in the BMS flavor.
ram	String	Specifies the memory size (MB) in the BMS flavor.

Request Parameters

None

Example Request

Querying BMSs bound to all enterprise projects of the user

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/detail?offset=1&limit=2&enterprise_project_id=all_granted_eps
```

Response Parameters

Parameter	Type	Description
servers	List data structure	Specifies details about BMSs. For details, see **Querying BMS Details**. The returned details vary depending on the query level.
count	Integer	Specifies the number of BMSs that match the filters.

Example Response


```
{
   "count": 2,
   "servers": [
      {
         "id": "b544be62-1b3b-4982-ad98-572b002ac23b",
         "name": "bms-test1",
         "addresses": {
            "5849fdf1-9d79-4589-80c2-fe557990c417": [
               {
                  "version": "4",
                  "addr": "192.168.1.63",
                  "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:3a:8d:f1",
                  "OS-EXT-IPS:port_id": "c79d2813-94c9-4135-973e-cbf5d23e78e6",
                  "OS-EXT-IPS:type": "fixed"
               }
            ]
         },
         "flavor": {
            "disk": "9309",
            "vcpus": "32",
            "ram": "193047",
            "id": "physical.s1.medium.ondemand",
            "name": "physical.s1.medium.ondemand"
         },
         "accessIPv4": "",
         "accessIPv6": "",
         "status": "ACTIVE",
         "progress": 0,
         "hostId": "cd243addb5d2c64e89218180b7a3ed95abe6882e81c337cc563137df",
         "updated": "2018-09-10T01:20:58Z",
         "created": "2018-09-06T09:29:27Z",
         "metadata": {
            "baremetalPortIDList": "[c79d2813-94c9-4135-973e-cbf5d23e78e6]",
            "chargingMode": "0"
         },
         "tags": [
            "__type_baremetal"
         ],
         "description": "bms-test1",
         "locked": false,
         "config_drive": "",
         "tenant_id": "bbf1946d374b44a0a2a95533562ba954",
         "user_id": "0c50494c5816425eb05c40b5e81ab65a",
         "key_name": "$key_name",
         "OS-EXT-STS:power_state": 1,
         "OS-EXT-STS:vm_state": "active",
         "OS-EXT-SRV-ATTR:host": "bms.dc1",
         "OS-EXT-SRV-ATTR:instance_name": "instance-0014bdc2",
         "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova002@2",
         "OS-DCF:diskConfig": "MANUAL",
         "OS-EXT-AZ:availability_zone": "az-dc-1",
         "os:scheduler_hints": {},
         "OS-EXT-SRV-ATTR:root_device_name": "/dev/vda",
         "OS-EXT-SRV-ATTR:ramdisk_id": "",
         "enterprise_project_id": "0",
         "OS-SRV-USG:launched_at": "2018-09-06T09:30:36.000000",
         "OS-EXT-SRV-ATTR:kernel_id": "",
         "OS-EXT-SRV-ATTR:launch_index": 0,
         "host_status": "UP",
         "OS-EXT-SRV-ATTR:reservation_id": "r-qjad3fv0",
         "OS-EXT-SRV-ATTR:hostname": "bms-test1",
         "sys_tags": [
            {
               "key": "_sys_enterprise_project_id",
               "value": "0"
            }
         ]
      },
```



```
{
    "id": "a1541cfc-8ac3-43e9-a70d-b8d4b395b256",
    "name": "bms_test2",
    "addresses": {
        "5849fdf1-9d79-4589-80c2-fe557990c417": [
            {
                "version": "4",
                "addr": "192.168.1.50",
                "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:8f:38:2c",
                "OS-EXT-IPS:port_id": "460aa585-9f83-4719-9527-fc39ebaca9aa",
                "OS-EXT-IPS:type": "fixed"
            }
        ]
    },
    "flavor": {
        "disk": "9309",
        "vcpus": "32",
        "ram": "193047",
        "id": "physical.s1.medium",
        "name": "physical.s1.medium"
    },
    "accessIPv4": "",
    "accessIPv6": "",
    "status": "SHUTOFF",
    "hostId": "cd243addb5d2c64e89218180b7a3ed95abe6882e81c337cc563137df",
    "updated": "2018-09-06T10:00:25Z",
    "created": "2018-08-30T12:40:47Z",
    "metadata": {
        "metering.order_id": "CS18083020422CNV9",
        "baremetalPortIDList": "[460aa585-9f83-4719-9527-fc39ebaca9aa]",
        "metering.product_id": "00301-167001-0--0",
        "chargingMode": "1"
    },
    "tags": [
        "__type_baremetal",
        "_sys_enterprise_project_id=9dd1131d-71fd-40fe-8f14-3fe6b6b5ef8b",
        "key1=value1",
        "three=3",
        "two=2"
    ],
    "description": "bms_test2",
    "locked": false,
    "config_drive": "",
    "tenant_id": "bbf1946d374b44a0a2a95533562ba954",
    "user_id": "3fc5ab2b0c544979abcaafd86edd80e6",
    "key_name": "$key_name",
    "OS-EXT-STS:power_state": 4,
    "OS-EXT-STS:vm_state": "stopped",
    "OS-EXT-SRV-ATTR:host": "bms.dc1",
    "OS-EXT-SRV-ATTR:instance_name": "instance-0014581b",
    "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova002@2",
    "OS-DCF:diskConfig": "MANUAL",
    "OS-EXT-AZ:availability_zone": "az-dc-1",
    "os:scheduler_hints": {},
    "OS-EXT-SRV-ATTR:root_device_name": "/dev/vda",
    "OS-EXT-SRV-ATTR:ramdisk_id": "",
    "enterprise_project_id": "0",
    "OS-SRV-USG:launched_at": "2018-08-30T12:42:10.000000",
    "OS-EXT-SRV-ATTR:kernel_id": "",
    "OS-EXT-SRV-ATTR:launch_index": 0,
    "host_status": "UP",
    "OS-EXT-SRV-ATTR:reservation_id": "r-i5w3yc9a",
    "OS-EXT-SRV-ATTR:hostname": "bms-test2",
    "sys_tags": [
        {
            "key": "_sys_enterprise_project_id",
            "value": "0"
        }
    ]
```



```
    }
  ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.4 BMS Status Management

4.4.1 Changing the BMS Name

Function

This API is used to change the name of a BMS.

Constraints

The new name must meet the following requirements:

Contain 1 to 63 characters, consisting of letters (case-insensitive), digits, underscores (_), hyphens (-), and periods (.).

NOTE

This API can only be used to change the BMS name. The hostname will not be changed. The new name takes effect after the BMS is restarted.

URI

PUT /v1/{project_id}/baremetalservers/{server_id}

Table 4-38 lists the parameters.

Table 4-38 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
server	Yes	Object	Specifies the BMS data structure. For details, see **Table 4-39**.

Table 4-39 server field data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the new BMS name.

Example Request

Changing the name of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) to **new-server-test**

```
PUT https://{BMS Endpoint}/v1/c685484a8cc2416b97260938705deb65/baremetalservers/
95bf2490-5428-432c-ad9b-5e3406f869dd
{
  "server":
    {
      "name": "new-server-test"
    }
}
```

Response Parameters

Parameter	Type	Description
server	Object	Specifies BMS information. For details, see **Table 4-40**.

Table 4-40 server field data structure description

Parameter	Type	Description
name	String	Specifies the BMS name.
id	String	Specifies the unique ID of the BMS.
status	String	Specifies the BMS status. • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
created	String	Specifies the time when the BMS was created. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.
updated	String	Specifies the time when the BMS was last updated. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T04:30:52Z.
flavor	Object	Specifies the BMS flavor. For details, see **Table 4-43**.
image	Object	Specifies the BMS image. For details, see **Table 4-44**.
tenant_id	String	Specifies the ID of the tenant owning the BMS. The value is in UUID format. This parameter specifies the same meaning as **project_id**.
key_name	String	Specifies the SSH key name.
user_id	String	Specifies the ID of the user to which the BMS belongs.
metadata	Object	Specifies the BMS metadata. The structure is Map<String, Object>. For details, see **Table 4-46**.
hostId	String	Specifies the host ID of the BMS.

Parameter	Type	Description
addresses	Object	Specifies BMS network addresses. The structure is Map<String, Object>. For details, see **Table 4-47**.
security_groups	Array of objects	Specifies the security groups to which the BMS belongs. For details, see **Table 4-49**.
links	Array of objects	Specifies shortcut links of the BMS. For details, see **Table 4-45**.
OS-DCF:diskConfig	String	Specifies the disk configuration method. This is an extended attribute. The value can be: • **MANUAL**: The API uses the partitioning scheme in the image and the file system to create a BMS. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The API uses a single partition with the same size as the disk of the target flavor to create a BMS. The API automatically adjusts the file system to adapt to the entire partition.
OS-EXT-AZ:availability_zone	String	Specifies the AZ where the BMS is located.
OS-EXT-SRV-ATTR:host	String	Specifies the hostname of the BMS. This is an extended attribute.
OS-EXT-SRV-ATTR:hypervisor_hostname	String	Specifies the hostname of the hypervisor.
OS-EXT-SRV-ATTR:instance_name	String	Specifies the BMS alias. This is an extended attribute.
OS-EXT-STS:power_state	Integer	Specifies the BMS power status. This is an extended attribute. Examples: • **0**: indicates **NO STATE**. • **1**: indicates **RUNNING**. • **4**: indicates **SHUTDOWN**.

Parameter	Type	Description
OS-EXT-STS:task_state	String	Specifies the BMS task status. This is an extended attribute. Examples: • **rebooting**: The BMS is being restarted. • **reboot_started**: The BMS is normally restarted. • **reboot_started_hard**: The BMS is forcibly restarted. • **powering-off**: The BMS is being powered off. • **powering-on**: The BMS is being powered on. • **rebuilding**: The BMS is being rebuilt. • **scheduling**: The BMS is being scheduled. • **deleting**: The BMS is being deleted.
OS-EXT-STS:vm_state	String	Specifies the BMS status. This is an extended attribute. Examples: • **active**: The BMS is running. • **stopped**: The BMS is stopped.
OS-SRV-USG:launched_at	String	Specifies the time when the BMS was started. This is an extended attribute. The timestamp format is ISO 8601, for example, **2019-05-25T03:40:25.000000**.
OS-SRV-USG:terminated_at	String	Specifies the time when the BMS was stopped. This is an extended attribute. The timestamp format is ISO 8601, for example, **2019-06-25T03:40:25.000000**.
os-extended-volumes:volumes_attached	Array of objects	Specifies the EVS disks attached to the BMS. For details, see **Table 4-50**.
accessIPv4	String	This is a reserved attribute.
accessIPv6	String	This is a reserved attribute.
fault	Object	Specifies the cause of a BMS fault. For details, see **Table 4-51**.
config_drive	String	This is a reserved attribute.
progress	Integer	This is a reserved attribute.
description	String	Specifies BMS information.

Parameter	Type	Description
host_status	String	Specifies the status of the host that accommodates the BMS. • **UP**: The nova-compute status is normal. • **UNKNOWN**: The nova-compute status is unknown. • **DOWN**: the nova-compute status is abnormal. • **MAINTENANCE**: The nova-compute is in the maintenance state. • Empty string: The BMS does not have host information.
OS-EXT-SRV-ATTR:hostname	String	Specifies the hostname of the BMS.
OS-EXT-SRV-ATTR:reservation_id	String	Specifies the reserved BMS IDs in a batch creation situation. When BMSs are created in a batch, they have the same **reservation_id**.
OS-EXT-SRV-ATTR:launch_index	Integer	Specifies the BMS startup sequence in a batch creation situation.
OS-EXT-SRV-ATTR:kernel_id	String	Specifies the UUID of the kernel image when the AMI image is used. In other scenarios, leave this parameter blank.
OS-EXT-SRV-ATTR:ramdisk_id	String	Specifies the UUID of the Ramdisk image when the AMI image is used. In other scenarios, leave this parameter blank.
OS-EXT-SRV-ATTR:root_device_name	String	Specifies the device name of the BMS system disk, for example, **/dev/sdb**.
OS-EXT-SRV-ATTR:user_data	String	Specifies the **user_data** specified during BMS creation. The value is encoded using Base64 or an empty string.
locked	Boolean	Specifies whether the BMS is locked. • **true**: The BMS is locked. • **false**: The BMS is not locked.
tags	Array of strings	Specifies tags of the BMS.
sys_tags	Array of objects	Specifies system tags of the BMS. For details, see **Table 4-41**.

Parameter	Type	Description
enterprise_project_id	String	Specifies the enterprise project ID.
os:scheduler_hints	Object	Specifies the BMS scheduling information. For details, see **Table 4-42**.

Table 4-41 sys_tags field data structure description

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

Table 4-42 os:scheduler_hints field data structure description

Parameter	Type	Description
group	Array of strings	Specifies the BMS group ID in UUID format.

Table 4-43 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the flavor ID.
links	Array of objects	Specifies shortcut links of the flavor. For details, see **Table 4-45**.

Table 4-44 image field data structure description

Parameter	Type	Description
id	String	Specifies the image ID of the BMS.
links	Array of objects	Specifies shortcut links of the BMS image. For details, see **Table 4-45**.

Table 4-45 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Table 4-46 metadata field data structure description

Parameter	Type	Description
key	String	Specifies the key and value pair of the metadata. Each key or value contains a maximum of 255 characters.

Table 4-47 addresses field data structure description

Parameter	Type	Description
vpc_id	Array of objects	Specifies the VPC used by the BMS. • **key**: indicates the ID of the VPC used by the BMS. • **value**: indicates the VPC details. For details, see **Table 4-48**.

Table 4-48 address field data structure description

Parameter	Type	Description
addr	String	Specifies the IP address.
version	Integer	Specifies the type of the IP address. The value 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.
OS-EXT-IPS:port_id	String	Specifies the port ID corresponding to the IP address.

Table 4-49 security_groups field data structure description

Parameter	Type	Description
name	String	• If no security group is specified during BMS creation, the **default** value is used. • When creating a BMS, you need to specify the ID of an existing security group (in UUID format).

Table 4-50 os-extended-volumes:volumes_attached field data structure description

Parameter	Type	Description
id	String	Specifies the EVS disk ID.
delete_on_termination	Boolean	Specifies whether to delete the disk when deleting the BMS. • **true**: Yes • **false**: No

Table 4-51 fault field data structure description

Parameter	Type	Description
message	String	Specifies the fault information.
code	Integer	Specifies the fault code.
details	String	Specifies the fault details.
created	String	Specifies the time when a fault occurred. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.

Example Response


```
{
    "server": {
        "tenant_id": "c685484a8cc2416b97260938705deb65",
        "addresses": {
            "08a7715f-7de6-4ff9-a343-95ba4209f24a": [
                {
                    "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:0e:c4:77",
                    "OS-EXT-IPS:type": "fixed",
                    "OS-EXT-IPS:port_id": "0c111191-9785-49e6-8b42-66ef2802bd8f",
                    "addr": "192.168.0.107",
                    "version": 4
                }
            ]
        },
        "metadata": {
            "op_svc_userid": "1311c433dd9b408886f57d695c229cbe"
        },
        "OS-EXT-STS:task_state": null,
        "OS-DCF:diskConfig": "MANUAL",
        "OS-EXT-AZ:availability_zone": "az-dc-1",
        "links": [
            {
                "rel": "self",
                "href": "https://bms.az0.dc1.domainname.com/v2.1/c685484a8cc2416b97260938705deb65/
servers/95bf2490-5428-432c-ad9b-5e3406f869dd"
            },
            {
                "rel": "bookmark",
                "href": "https://bms.az0.dc1.domainname.com/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
            }
        ],
        "OS-EXT-STS:power_state": 1,
        "id": "95bf2490-5428-432c-ad9b-5e3406f869dd",
        "os-extended-volumes:volumes_attached": [
            {
                "id": "dfa375b5-9856-44ad-a937-a4802b6434c3"
            },
            {
                "id": "bb9f1b27-843b-4561-b62e-ca18eeaec417"
            },
            {
                "id": "86e801c3-acc6-465d-890c-d43ba493f553"
            },
            {
                "id": "0994d3ac-3c6a-495c-a439-c597a4f08fa6"
            }
        ],
        "OS-EXT-SRV-ATTR:host": "bms.az-dc-1",
        "image": {
            "links": [
                {
                    "rel": "bookmark",
                    "href": "https://bms.az0.dc1.domainname.com/c685484a8cc2416b97260938705deb65/images/
1a6635d8-afea-4f2b-abb6-27a202bad319"
                }
            ],
            "id": "1a6635d8-afea-4f2b-abb6-27a202bad319"
        },
        "OS-SRV-USG:terminated_at": null,
        "accessIPv4": "",
        "accessIPv6": "",
        "created": "2017-05-24T06:14:05Z",
        "hostId": "e9c3ee0fcc58ab6085cf30df70b5544eab958858fb50d925f023e53e",
        "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova004@2",
        "key_name": "$key_name",
        "flavor": {
            "links": [
```



```
            {
                "rel": "bookmark",
                "href": "https://bms.az0.dc1.domainname.com/c685484a8cc2416b97260938705deb65/flavors/
physical.83.medium"
            }
        ],
        "id": "physical.83.medium"
    },
    "security_groups": [
        {
            "name": "0011b620-4982-42e4-ad12-47c95ca495c4"
        }
    ],
    "config_drive": "",
    "OS-EXT-STS:vm_state": "active",
    "OS-EXT-SRV-ATTR:instance_name": "instance-0000ebd3",
    "user_id": "1311c433dd9b408886f57d695c229cbe",
    "name": "bms-83",
    "progress": 0,
    "OS-SRV-USG:launched_at": "2017-05-25T03:40:25.066078",
    "updated": "2017-05-25T03:40:25Z",
    "status": "ACTIVE"
  }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.4.2 Reinstalling the BMS OS

Function

This API is used to reinstall the BMS OS. The OS of BMSs supporting quick provisioning can be reinstalled using the original image without any change to the data disks. Password and key pair injection are supported during OS reinstallation.

NOTE

To check whether a BMS is quickly provisioned, use the **Querying Flavor Details and Extended Flavor Information** API.

This is an asynchronous API. Calling the API successfully indicates that the task is delivered successfully. To check whether the task is successful, use the **Querying Task Statuses** API.

Constraints

- For BMSs created from private images, ensure that Cloud-Init (for Linux) or Cloudbase-Init (for Windows) has been installed for the image. If an image

without Cloud-Init or Cloudbase-Init is used, this API cannot inject a key pair or password. Cloud-Init or Cloudbase-Init has been installed for public images by default.

- You are not allowed to perform other operations when reinstalling the OS. Otherwise, reinstalling the OS will fail.
- You can reinstall the OS only on a BMS that is stopped or for which OS reinstallation has failed.
- Windows images do not support password injection. An encrypted password will be used as the value of **user_data** for Linux images.

URI

POST /v1/{project_id}/baremetalservers/{server_id}/reinstallos

Table 4-52 lists the parameters.

Table 4-52 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
os-reinstall	Yes	Object	Specifies the operation of reinstalling the BMS OS. For details, see **Table 4-53**.

Table 4-53 os-reinstall field data structure description

Parameter	Mandatory	Type	Description
adminpass	No	String	Specifies the initial login password of the BMS administrator account. The Linux administrator is **root**, and the Windows administrator is **Administrator**. Recommended password complexity requirements are as follows: • The password contains 8 to 26 characters. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters !@$%^-_=+[{}]:,./? • The password cannot contain the username or the username in reverse. **NOTE** • For Windows BMSs, the password cannot contain more than two consecutive characters in the username. • For Linux BMSs, **user_data** can be used to inject a password. In this case, **adminpass** is invalid. • Either **adminpass** or **keyname** can be set. • If both **adminpass** and **keyname** are empty, **user_data** in metadata must be set.
keyname	No	String	Specifies the key pair name. You can create a key pair using the **Creating or Importing an SSH Key Pair (Native OpenStack API)** API, or query existing key pairs using the **Querying SSH Key Pairs (Native OpenStack API)** API.
userid	No	String	Specifies the user ID. You can obtain the user ID from **My Credential** on the management console.
metadata	No	Object	Specifies the BMS metadata. For details, see **Table 4-54**.

Table 4-54 metadata field data structure description

Parameter	Mandatory	Type	Description

user_data	No	String	Specifies the Linux image root password injected during the BMS OS reinstallation. It is a user-defined initial password.
			Note: The password change script must be encoded using Base64.
			Recommended password complexity requirements are as follows:
			• Contains 8 to 26 characters.
			• Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters !@$%^-_=+[{}]:,./?
			An example is as follows:
			`#!/bin/bash`
			`echo 'root:$6$V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig/ GpOkLcOhab9smJoLKYm/ Tf9Hcwa6DpiPDhdHfGEAPajFmLZa0YDd910' | chpasswd -e`
			• This script must be encoded using Base64.
			where, **6V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig** is the ciphertext password, which can be generated as follows:
			1. Generate an encrypted salt value.
			`[root@test linux]# python -c "import crypt, getpass, pwd;print crypt.mksalt()" $6$V6azyeLwcD3CHlpY`
			2. Generate a ciphertext password based on the salt value.
			`[root@test linux]# python -c "import crypt, getpass, pwd;print crypt.crypt('Cloud.1234','\$6\ $V6azyeLwcD3CHlpY')" $6$V6azyeLwcD3CHlpY $BN3VVq18fmCkj66B4zdHLWevqcxlig/ GpOkLcOhab9smJoLKYm/ Tf9Hcwa6DpiPDhdHfGEAPajFmL- Za0YDd910`
			3. After the ciphertext is generated, the password change script must be encoded using Base64.

Example Request

Reinstalling the OS of a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b; key name: KeyPair-350b; user ID: 7e25b1da389f4697a79df3a0e5bd494e)

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/reinstallos
{
    "os-reinstall": {
        "keyname": "$key_name",
        "userid": "7e25b1da389f4697a79df3a0e5bd494e",
        "metadata": {
            "user_data": "$USER_DATA"
        }
    }
}
```

Response Parameters

Table 4-55 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

Table 4-56 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-57**.

Table 4-57 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.4.3 Starting BMSs

Function

This API is used to start BMSs of specified IDs. You can start a maximum of 1000 BMSs at a time.

NOTE

This is an asynchronous API. Calling the API successfully indicates that the task is delivered successfully. To check whether the task is successful, use the **Querying Task Statuses** API.

URI

POST /v1/{project_id}/baremetalservers/action

Table 4-58 lists the parameters.

Table 4-58 Parameter description

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

Request Parameters

NOTE

In the request, the command to start BMSs must be sent with the parameter **os-start**. For details, see the example request.

Parameter	Mandatory	Type	Description
os-start	Yes	Object	Specifies the operation to start BMSs. For details, see **Table 4-59**.

Table 4-59 os-start field data structure description

Parameter	Mandatory	Type	Description
servers	Yes	Array of objects	Specifies BMS IDs. For details, see **Table 4-60**.

Table 4-60 servers field data structure description

Parameter	Mandatory	Type	Description
id	Yes	String	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Example Request

Starting two BMSs (IDs: 616fb98f-46ca-475e-917e-2563e5a8cd19 and 726fb98f-46ca-475e-917e-2563e5a8cd20)

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/action
{
    "os-start": {
        "servers": [
            {
                "id": "616fb98f-46ca-475e-917e-2563e5a8cd19"
            },
            {
                "id": "726fb98f-46ca-475e-917e-2563e5a8cd20"
            }
        ]
    }
}
```

Response Parameter

- Normal response

Table 4-61 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

- Abnormal response

Table 4-62 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-63**.

Table 4-63 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.4.4 Restarting BMSs

Function

This API is used to restart BMSs of specified IDs. You can restart a maximum of 1000 BMSs at a time.

NOTE

This is an asynchronous API. Calling the API successfully indicates that the task is delivered successfully. To check whether the task is successful, use the **Querying Task Statuses** API.

URI

POST /v1/{project_id}/baremetalservers/action

Table 4-64 lists the parameters.

Table 4-64 Parameter description

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

Request Parameters

NOTE

In the request, the command to restart BMSs must be sent with the parameter **reboot**. For details, see the example request.

Parameter	Mandatory	Type	Description
reboot	Yes	Object	Specifies the operation to restart BMSs. For details, see **Table 4-65**.

Table 4-65 reboot field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the BMS reboot type. • **SOFT**: soft restart (invalid) • **HARD**: hard restart (default)
servers	Yes	Array of objects	Specifies BMS IDs. For details, see **Table 4-66**.

Table 4-66 servers field data structure description

Parameter	Mandatory	Type	Description
id	Yes	String	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Example Request

Restarting two BMSs (IDs: are 616fb98f-46ca-475e-917e-2563e5a8cd19 and 726fb98f-46ca-475e-917e-2563e5a8cd20)

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/action
{
   "reboot": {
      "type": "HARD",
      "servers": [
         {
            "id": "616fb98f-46ca-475e-917e-2563e5a8cd19"
         },
         {
            "id": "726fb98f-46ca-475e-917e-2563e5a8cd20"
         }
      ]
   }
}
```

Response Parameters

- Normal response

Table 4-67 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

- Abnormal response

Table 4-68 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-69**.

Table 4-69 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.4.5 Stopping BMSs

Function

This API is used to stop BMSs of specified IDs. You can stop a maximum of 1000 BMSs at a time.

NOTE

This is an asynchronous API. Calling the API successfully indicates that the task is delivered successfully. To check whether the task is successful, use the **Querying Task Statuses** API.

URI

POST /v1/{project_id}/baremetalservers/action

Table 4-70 lists the parameters.

Table 4-70 Parameter description

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

Request Parameters

NOTE

In the request, the command to stop BMSs must be sent with the parameter **os-stop**. For details, see the example request.

Parameter	Mandatory	Type	Description
os-stop	Yes	Object	Specifies the operation to stop BMSs. For details, see **Table 4-71**.

Table 4-71 os-stop field data structure description

Parameter	Mandatory	Type	Description
servers	Yes	Array of objects	Specifies BMS IDs. For details, see **Table 4-72**.

Parameter	Mandatory	Type	Description
type	No	String	Specifies the BMS stop type. • **SOFT**: soft stop (invalid) • **HARD**: hard stop (default)

Table 4-72 servers field data structure description

Parameter	Mandatory	Type	Description
id	Yes	String	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Example Request

Stopping two BMSs (IDs: are 616fb98f-46ca-475e-917e-2563e5a8cd19 and 726fb98f-46ca-475e-917e-2563e5a8cd20)

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/action
{
    "os-stop": {
        "type": "HARD",
        "servers": [
            {
                "id": "616fb98f-46ca-475e-917e-2563e5a8cd19"
            },
            {
                "id": "726fb98f-46ca-475e-917e-2563e5a8cd20"
            }
        ]
    }
}
```

Response Parameters

- Normal response

Table 4-73 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

- Abnormal response

Table 4-74 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-75**.

Table 4-75 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.5 BMS Flavor Management

4.5.1 Querying Flavor Details and Extended Flavor Information

Function

This API is used to query BMS flavor details and extended flavor information. You can call this API to query the value of parameter **baremetal:extBootType** to check whether a flavor supports quick BMS provisioning.

URI

GET /v1/{project_id}/baremetalservers/flavors?availability_zone={availability_zone}

Table 4-76 lists the parameters.

Table 4-76 Parameter description

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

Request Parameters

None

Example Request

Querying BMS flavor details and extended flavor information in the **cn-north-1a** AZ

```
GET https://{BMS Endpoint}/v1/c685484a8cc2416b97260938705deb65/baremetalservers/flavors?availability_zone=cn-north-1a
```

Response Parameters

Parameter	Type	Description
flavors	Array of objects	Specifies BMS flavors. For details, see **Table 4-77**.

Table 4-77 flavors field data structure description

Parameter	Type	Description
id	String	Specifies the ID of a BMS flavor.
name	String	Specifies the name of a BMS flavor.

Parameter	Type	Description
vcpus	String	Specifies the number of CPU cores in a BMS flavor.
ram	Integer	Specifies the memory size (MB) in a BMS flavor.
disk	String	Specifies the system disk size in a BMS flavor. The value **0** indicates that the disk size is not limited.
swap	String	This is a reserved attribute.
OS-FLV-EXT-DATA:ephemeral	Integer	This is a reserved attribute.
OS-FLV-DISABLED:disabled	Boolean	This is a reserved attribute.
rxtx_factor	Float	This is a reserved attribute.
rxtx_quota	String	This is a reserved attribute.
rxtx_cap	String	This is a reserved attribute.
os-flavor-access:is_public	Boolean	Specifies whether a flavor is public. **false** indicates a private flavor and **true** indicates a public flavor.
links	Array of objects	Specifies shortcut links of a flavor. For details, see **Table 4-78**.
os_extra_specs	Object	Specifies extended fields of a BMS flavor. For details, see **Table 4-79**.

Table 4-78 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.
type	String	Specifies the shortcut link type.

Table 4-79 os_extra_specs field data structure description

Parameter	Type	Description
resource_type	String	Specifies the resource type of a flavor.
capabilities:cpu_arch	String	Specifies the CPU architecture of the BMS. The value can be: • **x86_64** (applicable to x86 servers) • **aarch64** (applicable to ARM servers)
baremetal:disk_detail	String	Specifies physical disk specifications.
capabilities:hypervisor_type	String	Specifies a flavor of the Ironic type.
baremetal:__support_evs	String	Specifies whether a flavor supports EVS disks. • true • false If the flavor of a BMS does not contain this parameter, EVS disks cannot be attached to the BMS.
baremetal:extBootType	String	Specifies the boot source of the BMS. • **LocalDisk**: local disk • **Volume**: EVS disk (quick provisioning)
capabilities:board_type	String	Specifies the type of a BMS flavor in the format of flavor abbreviation. For example, if the flavor name is **physical.o2.medium**, the flavor type is **o2m**.
baremetal:net_num	String	Specifies the maximum number of NICs on the BMS.
baremetal:netcard_detail	String	Specifies physical NIC specifications.
baremetal:cpu_detail	String	Specifies physical CPU specifications.
baremetal:memory_detail	String	Specifies physical memory specifications.

Parameter	Type	Description
cond:operation:status	String	Specifies the status of a BMS flavor. If this parameter is not set, its default value is **normal**. • **normal**: indicates normal commercial use of the flavor. • **abandon**: indicates that the flavor has been disabled (not displayed). • **sellout**: indicates that the flavor has been sold out. • **obt**: indicates that the flavor is under OBT. • **promotion**: indicates the recommended flavor (commercial use, which is similar to **normal**).
cond:operation:az	String	Specifies the status of a BMS flavor in an AZ. This parameter takes effect AZ-wide. If an AZ is not configured in this parameter, the value of the **cond:operation:status** parameter is used by default. Its format is az (*xx*). *xx* indicates the status of a BMS flavor in an AZ, and it is mandatory. For example, a flavor is for commercial use in AZ0 and AZ3, sold out in AZ1, for OBT in AZ2, and is canceled in other AZs. Then, set parameters as follows: • **cond:operation:status**: **abandon** • **cond:operation:az**: **az0(normal), az1(sellout), az2(obt), az3(promotion)** **NOTE** Configure this parameter if the flavor status in an AZ is different from the **cond:operation:status** value.

Example Response

```
{
   "flavors": [
      {
         "id": "physical.kl1.3xlarge",
         "name": "physical.kl1.3xlarge",
         "vcpus": "24",
         "ram": 321729,
         "disk": "6707",
```

```
            "swap": "",
            "links": [
                {
                    "rel": "self",
                    "href": "https://compute.Region.dc1.domainname.com/v2/
bbf1946d374b44a0a2a95533562ba954/flavors/physical.kl1.3xlarge",
                    "type": null
                },
                {
                    "rel": "bookmark",
                    "href": "https://compute.Region.dc1.domainname.com/bbf1946d374b44a0a2a95533562ba954/
flavors/physical.kl1.3xlarge",
                    "type": null
                }
            ],
            "OS-FLV-EXT-DATA:ephemeral": 0,
            "rxtx_factor": 1,
            "OS-FLV-DISABLED:disabled": false,
            "rxtx_quota": null,
            "rxtx_cap": null,
            "os-flavor-access:is_public": false,
            "os_extra_specs": {
                "capabilities:cpu_arch": "x86_64",
                "baremetal:disk_detail": "SAS SSD:2*800G Raid 1 + NVMe SSD Card1.6T",
                "capabilities:hypervisor_type": "ironic",
                "baremetal:__support_evs": "true",
                "baremetal:extBootType": "LocalDisk",
                "capabilities:board_type": "o2m",
                "baremetal:net_num": "2",
                "baremetal:netcard_detail": "2 x 2*10GE",
                "baremetal:cpu_detail": "Intel Xeon E5-2667 V4 (2*8core* 3.2 GHz)",
                "resource_type": "ironic",
                "baremetal:memory_detail": "256GB DDR4 RAM(GB)"
            }
        }
    ]
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.6 BMS NIC Management

4.6.1 Querying NICs bound to a BMS

Function

This API is used to query information about NICs bound to a BMS, such as the IP address and MAC address of each NIC.

URI

GET /v1/{project_id}/baremetalservers/{server_id}/os-interface

Table 4-80 lists the parameters.

Table 4-80 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying information about NICs bound to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{BMS Endpoint}/v1/c685484a8cc2416b97260938705deb65/baremetalservers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-interface
```

Response Parameters

Parameter	Type	Description
interfaceAttachments	Array of objects	Specifies BMS NICs. For details, see **Table 4-81**.

Table 4-81 interfaceAttachments field data structure description

Parameter	Type	Description
port_state	String	Specifies the NIC port status. The value can be **ACTIVE**, **BUILD**, or **DOWN**.
fixed_ips	Array of objects	Specifies private IP addresses of NICs. For details, see **Table 4-82**.
net_id	String	Specifies the ID of the subnet (**network_id**) to which the NIC ports belong.
port_id	String	Specifies the ID of the NIC port.
mac_addr	String	Specifies the MAC address of the NIC.
driver_mode	String	Specifies the NIC driver type in Guest OS. The value can be **virtio** or **hinic**. The default value is **virtio**.
pci_address	String	Specifies the BDF number of the NIC in Linux Guest OS.

Table 4-82 fixed_ips field data structure description

Parameter	Type	Description
subnet_id	String	Specifies the ID of the subnet (**subnet_id**) corresponding to the private IP address of the NIC.
ip_address	String	Specifies the NIC private IP address.

Example Response

```
{
  "interfaceAttachments": [
    {
      "port_state": "ACTIVE",
      "fixed_ips": [
        {
          "subnet_id": "a5052101-11e1-4f3d-a5fa-f6ba6791219c",
          "ip_address": "192.168.1.147"
        }
      ],
      "net_id": "ad0fadbf-4bc1-472c-a030-5310e53b3818",
      "port_id": "bb585b04-f2a2-4528-9064-fd0aeb4e15a9",
      "mac_addr": "fa:16:3e:3b:58:fc"
    }
  ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.7 BMS Disk Management

4.7.1 Attaching an EVS Disk to a BMS

Function

This API is used to attach EVS disks to a BMS as data disks after the BMS is created if existing disks are insufficient or cannot meet requirements.

Constraints

- EVS disks cannot be attached to a BMS in a batch.
- A maximum of 60 EVS disks can be attached to a BMS.
- A bootable disk cannot be attached to a BMS.
- A disk cannot be attached to a BMS in the **PAUSED** state. The state is indicated by the **OS-EXT-STS:vm_state** parameter.
- Only a shared disk or a disk in the **available** state can be attached to a BMS.
- Only EVS disks whose device type is **SCSI** can be attached to a BMS.

URI

POST /v1/{project_id}/baremetalservers/{server_id}/attachvolume

Table 4-83 lists the parameters.

Table 4-83 Parameter description

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

Parameter	Mandatory	Description
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
volumeAttachment	Yes	Object	Specifies the disks to be attached. For details, see **Table 4-84**.

Table 4-84 volumeAttachment field data structure description

Parameter	Mandatory	Type	Description
volumeId	Yes	String	Specifies the ID of the disk to be attached to a BMS. You can obtain the disk ID from the EVS console or by calling the **Querying EVS Disks**.
device	No	String	Specifies the mount point, such as **/dev/sda** and **/dev/sdb**. The new disk mount point cannot be the same as an existing one. The mount point must be specified based on the sequence of existing device names. If this parameter is left blank or set to **""**, the system automatically generates a mount point.

Example Request

Attaching an EVS disk (ID: b53f23bd-ee8f-49ec-9420-d1acfeaf91d6) to a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b)

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/attachvolume
{
    "volumeAttachment": {
        "volumeId": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
        "device": ""
    }
}
```

Response Parameters

Table 4-85 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

Table 4-86 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 4-87**.

Table 4-87 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```

Returned Values

Normal values

Returned Value	Description
202	The request has been accepted, but the processing has been delayed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.7.2 Detaching an EVS Disk from a BMS

Function

This API is used to detach a disk from a BMS.

- A disk attached to **/dev/sda** functions as the system disk. You can only detach the system disk from a stopped BMS.
- Disks attached to a mount point other than **/dev/sda** function as data disks and can be detached from a running or stopped BMS.

Constraints

If a BMS is stopped, disks can be detached from it without any constraints on the OS. If a BMS is running, the constraints are as follows:

- Before detaching an EVS disk from a Linux BMS, log in to the BMS, run the **unmount** command to disassociate the disk to be detached from the file system, and ensure that no program is reading data from or writing data to the disk. Otherwise, the disk will fail to be detached.
- Before detaching an EVS disk from a Windows BMS, ensure that no program is reading data from or writing data to the disk. Otherwise, data will be lost.

URI

DELETE /v1/{project_id}/baremetalservers/{server_id}/detachvolume/{attachment_id}

Table 4-88 lists the parameters.

Table 4-88 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Parameter	Mandatory	Description
attachment_id	Yes	Specifies the IDs of the EVS disks attached to the BMS. You can obtain the IDs of EVS disks attached to a BMS using the **Querying EVS Disks Attached to a BMS** API.

Request Parameters

None

Example Request

Detaching an EVS disk (ID: 6b604cef-9bd8-4f5a-ae56-45839e6e1f0a) from a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b)

```
DELETE https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/detachvolume/6b604cef-9bd8-4f5a-ae56-45839e6e1f0a
```

Response

See **Task ID Response**.

Returned Values

Normal values

Returned Value	Description
202	The request has been accepted, but the processing has been delayed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.7.3 Querying EVS Disks Attached to a BMS

Function

This API is used to query EVS disks attached to a BMS.

URI

GET /v1/{project_id}/baremetalservers/{server_id}/os-volume_attachments

Table 4-89 lists the parameters.

Table 4-89 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying EVS disks attached to a BMS (ID: 4d8c3732-a248-40ed-bebc-539a6ffd25c0)

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/4d8c3732-a248-40ed-bebc-539a6ffd25c0/os-volume_attachments
```

Response Parameters

Parameter	Type	Description
volumeAttachments	Array of objects	Specifies disks attached to a BMS. For details, see **Table 4-90**.

Table 4-90 volumeAttachments field data structure description

Parameter	Type	Description
device	String	Specifies the mount directory, for example, **dev/sdd**.
id	String	Specifies the ID of the attached resource.
serverId	String	Specifies the ID of the BMS to which the disk is attached.
volumeId	String	Specifies the ID of the disk attached to the BMS.

Example Response

```
{
   "volumeAttachments": [
      {
         "device": "/dev/sdd",
         "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
         "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
         "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
      },
      {
         "device": "/dev/sdc",
         "id": "a26887c6-c47b-4654-abb5-dfadf7d3f804",
         "serverId": "4d8c3732-a248-40ed-bebc-539a6ffd25c0",
         "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f804"
      }
   ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.8 BMS Metadata Management

4.8.1 Updating BMS Metadata

Function

This API is used to update BMS metadata.

- If the metadata does not contain the target field, the field is automatically added to the field.
- If the metadata contains the target field, the field value is automatically updated.
- If the field in the metadata is not requested, the field value remains unchanged.

Constraints

The BMS must be in active, stopped, or paused state. The state is indicated by the **OS-EXT-STS:vm_state** parameter.

URI

POST /v1/{project_id}/baremetalservers/{server_id}/metadata

Table 4-91 lists the parameters.

Table 4-91 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Table 4-92 Request parameters

Parameter	Mandatory	Type	Description
metadata	Yes	Object	Specifies the user-defined metadata key-value pair. If you do not specify any key-value pair, metadata will not be updated. **key**: It contains a maximum of 255 Unicode characters which can be letters, digits, hyphens (-), underscores (_), colons (:), and periods (.). **key** cannot be left blank. **value**: It contains a maximum of 255 Unicode characters.

Example Request

Updating the metadata of a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b) to customize a key-value pair

```
POST https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/metadata
{
    "metadata": {
        "key": "value"
```

```
    }
}
```

Response Parameters

Table 4-93 Parameter description

Parameter	Mandatory	Type	Description
metadata	Yes	Object	Specifies the user-defined metadata key-value pair. If you do not specify any key-value pair, metadata will not be updated. **key**: It contains a maximum of 255 Unicode characters which can be letters, digits, hyphens (-), underscores (_), colons (:), and periods (.). **key** cannot be left blank. **value**: It contains a maximum of 255 Unicode characters.

Example Response

```
{
    "metadata":{
        "key": "value"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.9 BMS Quota Management

4.9.1 Querying Tenant Quotas

Function

This API is used to query the quotas of all resources for a specified tenant, including used quotas.

URI

GET /v1/{project_id}/baremetalservers/limits

Table 4-94 lists the parameters.

Table 4-94 Parameter description

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

Request Parameters

None

Example Request

Querying the quotas of a tenant (project ID: bbf1946d374b44a0a2a95533562ba954)

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/limits
```

Response Parameters

Parameter	Type	Description
absolute	Object	Specifies tenant quotas. For details, see **Table 4-95**.

NOTE

Value **-1** indicates that the quantity is unlimited.

Table 4-95 absolute field data structure description

Parameter	Type	Description
maxTotalInstances	Integer	Specifies the maximum number of BMSs you can create.

Parameter	Type	Description
maxTotalCores	Integer	Specifies the maximum number of CPUs you can use.
maxTotalRAMSize	Integer	Specifies the maximum memory (MB) you can use.
maxTotalKeypairs	Integer	Specifies the maximum number of SSH key pairs you can use.
maxServerMeta	Integer	Specifies the maximum length of the metadata you can use.
maxPersonality	Integer	Specifies the maximum number of files that can be injected.
maxPersonalitySize	Integer	Specifies the maximum size (byte) of the file to be injected.
maxServerGroups	Integer	Specifies the maximum number of server groups.
maxServerGroupMembers	Integer	Specifies the maximum number of BMSs in a server group.
totalServerGroupsUsed	Integer	Specifies the number of used server groups.
maxSecurityGroups	Integer	Specifies the maximum number of security groups you can use. **NOTE** The quota limit complies with the VPC quota limit.
maxSecurityGroupRules	Integer	Specifies the maximum number of security group rules that you can configure in a security group. **NOTE** The quota limit complies with the VPC quota limit.
maxTotalFloatingIps	Integer	Specifies the maximum number of EIPs you can use.
maxImageMeta	Integer	Specifies the maximum length of the image metadata.
totalInstancesUsed	Integer	Specifies the number of the used BMSs.
totalCoresUsed	Integer	Specifies the number of used CPUs.
totalRAMUsed	Integer	Specifies the used memory (MB).
totalSecurityGroupsUsed	Integer	Specifies the number of used security groups.
totalFloatingIpsUsed	Integer	Specifies the number of used EIPs.

Example Response


```
{
   "absolute": {
      "maxServerMeta": 128,
      "maxPersonality": 5,
      "maxImageMeta": 128,
      "maxPersonalitySize": 10240,
      "maxSecurityGroupRules": 20,
      "maxTotalKeypairs": 100,
      "totalRAMUsed": 799836,
      "totalInstancesUsed": 21,
      "maxSecurityGroups": 10,
      "totalFloatingIpsUsed": 0,
      "maxTotalCores": -1,
      "totalSecurityGroupsUsed": 1,
      "maxTotalFloatingIps": 10,
      "maxTotalInstances": 100,
      "totalCoresUsed": 148,
      "maxTotalRAMSize": -1,
      "maxServerGroups": -1,
      "maxServerGroupMembers": -1,
      "totalServerGroupsUsed": 1
   }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.10 BMS Password Management

4.10.1 Querying Whether One-Click Password Reset Is Supported

Function

This API is used to query whether a BMS supports one-click password reset.

URI

GET /v1/{project_id}/baremetalservers/{server_id}/os-resetpwd-flag

Table 4-96 lists the parameters.

Table 4-96 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying whether a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b) supports one-click password reset

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/os-resetpwd-flag
```

Response Parameters

Parameter	Type	Description
resetpwd_flag	String	Specifies whether one-click password reset is supported. • **True**: One-click password reset is supported. • **False**: One-click password reset is not supported.

Example Response

```
{
    "resetpwd_flag": "False"
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.10.2 Obtaining the Password of a Windows BMS

Function

This API is used to query the random password of the administrator (account **Administrator** or an account configured using Cloudbase-Init) generated by the system during the installation of a Windows BMS.

If the BMS is created from a private image, ensure that Cloudbase-Init has been installed. Cloudbase-Init is installed on public images by default.

URI

GET /v1/{project_id}/baremetalservers/{server_id}/os-server-password

Table 4-97 lists the parameters.

Table 4-97 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Obtaining the password of a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b)

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/os-server-password
```

Response Parameters

Parameter	Type	Description
password	String	Specifies the password in ciphertext.

Example Response

```
{
    "password": "UHC9+YW1xDC1Yu8Mg9n+tnOp7euEO/cW//9KgdJKWhr5w=="
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.10.3 Deleting the Password of a Windows BMS

Function

This API is used to delete the random password generated during initial Windows BMS installation. After the password is deleted, you can still use your password to log in to your BMS. However, you cannot use the Get Password function to recover the BMS initial password.

If the BMS is created from a private image, ensure that Cloudbase-Init has been installed. Cloudbase-Init is installed on public images by default.

URI

DELETE /v1/{project_id}/baremetalservers/{server_id}/os-server-password

Table 4-98 lists the parameters.

Table 4-98 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Deleting the password of a BMS (ID: cf2a8b97-b5c6-47ef-9714-eb27adf26e5b)

```
DELETE https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/baremetalservers/cf2a8b97-b5c6-47ef-9714-eb27adf26e5b/os-server-password
```

Response

N/A

Returned Values

Normal values

Returned Value	Description
202	The request has been accepted, but the processing has been delayed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

4.11 Task Management

4.11.1 Querying Task Statuses

Function

This API is used to query the execution status of a task.

After a task, such as creating a BMS or attaching disks, is delivered, **job_id** is returned, based on which you can query the execution status of the task.

URI

GET /v1/{project_id}/jobs/{job_id}

Table 4-99 lists the parameters.

Table 4-99 Parameter description

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

Request Parameters

None

Example Request

Querying the status of a task (ID: 2c9eb2c5544cbf6101544f0635672b60)

```
GET https://{BMS Endpoint}/v1/bbf1946d374b44a0a2a95533562ba954/jobs/
2c9eb2c5544cbf6101544f0635672b60
```

Response Parameters

Parameter	Type	Description
status	String	Specifies the task status. • **SUCCESS**: The task is successful. • **RUNNING**: The task is running. • **FAIL**: The task failed. • **INIT**: The task is being initialized.

Parameter	Type	Description
entities	Object	Specifies the task object. For details, see **Table 4-100**. The displayed information varies depending on the task type. For operations related to BMSs, **server_id** is displayed. For operations related to NICs, **nic_id** is displayed. If subtasks exist, details about the subtasks are also displayed.
job_id	String	Specifies the task ID.
job_type	String	Specifies the task type. The value can be: • **baremetalBatchCreate**: creating BMSs in a batch • **baremetalBatchOperate**: changing the power status of BMSs in a batch. • **baremetalVolumeBootReinstallOs**: reinstalling the OS of a BMS which is provisioned quickly • **baremetalReinstallOs**: reinstalling the OS of a BMS with local disks • **baremetalAttachVolume**: attaching a disk • **baremetalDetachVolume**: detaching a disk
begin_time	String	Specifies the time when the job started. The timestamp format is ISO 8601, for example, **2019-04-25T20:04:47.591Z**.
end_time	String	Specifies the time when the job ended. The timestamp format is ISO 8601, for example, **2019-04-26T20:04:47.591Z**.
error_code	String	Specifies the error code returned upon a task execution failure.
fail_reason	String	Specifies the cause of a task execution failure.
message	String	Specifies the message returned when an error occurs.
code	String	Specifies the error code returned when an error occurs. For details about error codes, see section **Status Codes**.

Table 4-100 entities field data structure description

Parameter	Type	Description
sub_jobs_total	Integer	Specifies the number of subtasks. If the task has no subtasks, the value of this parameter is **0**.
sub_jobs	Array of objects	Specifies the execution information about a subtask. If the task has no subtasks, the value of this parameter is left blank. For details, see **Table 4-101**.

Table 4-101 sub_jobs field data structure description

Parameter	Type	Description
status	String	Specifies the task status. • **SUCCESS**: The task is successful. • **RUNNING**: The task is running. • **FAIL**: The task failed. • **INIT**: The task is being initialized.
entities	Object	Specifies the task object. The displayed information varies depending on the task type. For operations related to BMSs, **server_id** is displayed. For operations related to NICs, **nic_id** is displayed. For details, see **Table 4-102**.
job_id	String	Specifies the task ID.
job_type	String	Specifies the task type. The value can be: • **baremetalSingleCreate**: creating a single BMS • **baremetalSingleOperate**: changing the power status of a single BMS
begin_time	String	Specifies the time when the job started. The timestamp format is ISO 8601, for example, **2019-04-25T20:04:47.591Z**.
end_time	String	Specifies the time when the job ended. The timestamp format is ISO 8601, for example, **2019-04-26T20:04:47.591Z**.

Parameter	Type	Description
error_code	String	Specifies the error code returned upon a task execution failure.
fail_reason	String	Specifies the cause of a task execution failure.
message	String	Specifies the message returned when an error occurs.
code	String	Specifies the error code returned when an error occurs. For details about error codes, see section **Status Codes**.

Table 4-102 entities field data structure description

Parameter	Type	Description
server_id	String	If the task is a BMS-related operation, **server_id** is displayed.
nic_id	String	If the task is a NIC-related operation, the value is **nic_id**.

Example Response

```
{
    "status": "SUCCESS",
    "entities": {
        "sub_jobs_total": 1,
        "sub_jobs": [
            {
                "status": "SUCCESS",
                "entities": {
                    "server_id": "bae51750-0089-41a1-9b18-5c777978ff6d"
                },
                "job_id": "2c9eb2c5544cbf6101544f0635672b60",
                "job_type": "baremetalSingleCreate",
                "begin_time": "2019-04-25T20:04:47.591Z",
                "end_time": "2019-04-25T20:08:21.328Z",
                "error_code": null,
                "fail_reason": null
            }
        ]
    },
    "job_id": "2c9eb2c5544cbf6101544f0602af2b4f",
    "job_type": "baremetalBatchCreate",
    "begin_time": "2019-04-25T20:04:34.604Z",
    "end_time": "2019-04-25T20:08:41.593Z",
    "error_code": null,
    "fail_reason": null
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5 Native OpenStack Nova V2.1 APIs

5.1 API Use Instructions

- For details about network service APIs, see **Virtual Private Cloud API Reference**.
- For the native OpenStack APIs not provided in this document (such as the API used to attach EVS disks to the BMS), use the native OpenStack APIs of ECS. For details, see **Elastic Cloud Server API Reference**.
- To support function extension, Nova APIs can be distinguished by version. There are two types of versions:
  - Major version: independent URL
  - Microversion: Used by the HTTP request header X-OpenStack-Nova-API-Version. Since microversion 2.27, the new microversion header OpenStack-API-Version has been supported.

5.2 API Version Query

5.2.1 Querying API Versions (Native OpenStack API)

Function

This interface is used to query all available Nova versions.

URI

GET /

Request Parameters

None

Example Request

Querying all available API versions

```
GET https://{ECS Endpoint}/
```

Response Parameters

Parameter	Type	Description
versions	Array of objects	Specifies the list of all API versions.
id	String	Specifies the version ID, for example, v1.
links	Array of objects	Specifies the API URL.
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.
version	String	If the APIs of this version support minor versions, set this parameter to the maximum minor version supported. If not, leave this parameter blank.
status	String	Specifies the version status. Possible values are as follows: • **CURRENT**: indicates a primary version. • **SUPPORTED**: indicates an old version that is still supported. • **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If the APIs of this version support minor versions, set this parameter to the supported minimum minor version. If not, leave this parameter blank.

Example response

```
{
   "versions": [
```



```
    {
        "links": [
            {
                "rel": "self",
                "href": "https://192.168.82.231:443/v2/"
            }
        ],
        "id": "v2.0",
        "updated": "2018-09-21T12:33:21Z",
        "status": "SUPPORTED"
    }
  ]
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.2.2 Querying an API Version (Native OpenStack API)

Function

This API is used to query a specified API version.

URI

GET /{api_version}

Table 5-1 lists the parameters.

Table 5-1 Parameter description

Parameter	Mandatory	Description
api_version	Yes	Specifies the API version, for example v2.

Request Parameters

None

Example Request

Querying the API v2

```
GET https://{ECS Endpoint}/v2
```

Response Parameters

Parameter	Type	Description
version	Object	Specifies a specified version.
id	String	Specifies the version ID, for example, v1.
links	Array of objects	Specifies the API URL.
href	String	Specifies the reference address of the current API version.
rel	String	Specifies the relationship between the current API version and the referenced address.
version	String	If the APIs of this version support minor versions, set this parameter to the maximum minor version supported. If not, leave this parameter blank.
status	String	Specifies the version status. Possible values are as follows: • **CURRENT**: indicates a primary version. • **SUPPORTED**: indicates an old version that is still supported. • **DEPRECATED**: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of v1 is 2014-06-28T12:20:21Z.
min_version	String	If the APIs of this version support minor versions, set this parameter to the supported minimum minor version. If not, leave this parameter blank.

Example response

```
{
    "version": {
        "min_version": "",
        "media-types": [
            {
                "type": "application/vnd.openstack.compute+json;version=2",
                "base": "application/json"
            }
```



```
        ],
        "links": [
            {
                "rel": "self",
                "href": "https://ecs.service.domain.com:443/v2/"
            },
            {
                "rel": "describedby",
                "href": "http://docs.openstack.org/",
                "type": "text/html"
            }
        ],
        "id": "v2.0",
        "updated": "1999-02-20T11:33:21Z",
        "version": "",
        "status": "SUPPORTED"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.3 BMS Lifecycle Management

5.3.1 Creating a BMS (Native OpenStack API)

Function

This interface is used to create a BMS.

Constraints

- This interface cannot be used to create BMSs in batches.
- When you create a BMS using an image that supports Cloud-Init or Cloudbase-Init, only parameter **key_name** can be configured. (Parameter **adminPass** is invalid.) The password of a Linux BMS can be injected only using parameter **user_data**. The password of a Windows BMS can be injected only using metadata **admin_pass**.
- When you create a BMS using an image that does not support Cloud-Init or Cloudbase-Init, both parameters **adminPass** and **key_name** are invalid. You need to use the password or certificate of the image to log in to the BMS.

- File injection is not supported.
- BMS creation from a system volume is not supported.
- Parameter **port** in the three network parameters (**port**, **uuid**, and **fixed_ip**) has the highest priority. If parameter **fixed_ip** is set, you must specify the UUID.
- After a BMS is created, it is recommended that you attach the **__type_baremetal** tag to the BMS. This tag specifies that the created server is a BMS. Otherwise, the BMS may not be displayed in the BMS list on the management console.
- A BMS can have a maximum of two VPCs, in which case the first VPC will be used by the primary NIC.

URI

POST /v2.1/{project_id}/servers

Table 5-2 lists the parameters.

Table 5-2 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID.

Request Parameters

Parameter	Mandatory	Type	Description
server	Yes	Object	Specifies the BMS information, see Table 5-3.

Table 5-3 server field data structure description

Parameter	Mandatory	Type	Description
imageRef	Yes	String	Specifies the ID of the image used by the BMS or the image resource uniform resource locator (URL). • Example image ID: 3b8d6fef-af77-42ab-b8b7-5a7f0f0af8f2 • Example image URL: http://glance.openstack.example.com/images/3b8d6fef-af77-42ab-b8b7-5a7f0f0af8f2 **NOTE** • BMSs using certain flavors do not support all public images provided by the cloud service platform. To obtain the images supported by a BMS flavor, log in to the management console, view the images displayed on the **Create ECS** page, and obtain the image IDs on the **Image Management Service** page. • If the creation fails, modify the parameter settings.
flavorRef	Yes	String	Specifies the ID or URL of the flavor used by the BMS.
name	Yes	String	Specifies the BMS name. It contains a maximum of 255 characters and cannot be left blank.
metadata	No	Object	Specifies the BMS metadata. The maximum size for both the metadata **key** and **value** is 255 characters. For details, see **Table 5-4**.

Parameter	Mandatory	Type	Description
user_data	No	String	Specifies the user data to be injected during the BMS creation. Text, text files, and .gzip files can be injected. The content to be injected cannot be greater than 32 KB in size. The content to be injected must be encoded with base64.

Parameter	Mandatory	Type	Description
adminPass	No	String	Specifies the initial login password of the BMS administrator account. This parameter is invalid for a Linux BMS. The administrator account of a Windows BMS is **Administrator**. Password complexity requirements: • The password contains 8 to 26 characters. • The password must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (!@$%^-_=+[{}]:,./?). • The password cannot contain the username or the username in reverse. • The Windows BMS password cannot contain the username, the username in reverse order, or more than two consecutive characters in the username. Note: If this parameter is not specified, a random password will be generated. Special characters: !@$%^-_=+[{}]:,./?
security_groups	No	Array of objects	Specifies the security group of a BMS. The default value is **default**. This parameter is valid when you specify parameter **network**. You are not allowed to specify multiple security groups. For details, see **Table 5-5**.

Parameter	Mandatory	Type	Description
networks	Yes	Array of objects	Specifies the BMS NICs. For details, see **Table 5-6**. You can specify a maximum of four networks for a BMS, including two VXLAN networks and two GENEVE networks. The first network in the parameter must be a VXLAN network. The network is used as by the primary NIC of the BMS. If multiple groups of network parameters are specified, ensure that the parameters of each group belong to the same VPC.
key_name	No	String	Specifies the name of a key pair. This is an extended attribute.
availability_zone	Yes	String	Specifies information about the AZ to which the BMS belongs. You are not allowed to specify host information.

Table 5-4 metadata field data structure description

Parameter	Mandatory	Type	Description
User-defined field key and value pair	No	String	Specifies the key and value pair of the metadata. Each key or value contains a maximum of 255 characters.

Table 5-5 security_groups field data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the name of the security group to which the BMS belongs.

Table 5-6 networks field data structure description

Parameter	Mandatory	Type	Description
port	No	String	Specifies the UUID of the network port.
uuid	No	String	Specifies the network UUID.
fixed_ip	No	String	Specifies the fixed IP address.

Example Request

Creating a BMS (ID: 1a6635d8-afea-4f2b-abb6-27a202bad319; flavor: physical.o2.medium; name: bms_name01; AZ: az-dc-1; network UUID: 8470310b-bfa2-4edf-8f64-d15196b2b2c9)

```
POST https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers
{
    "server": {
        "imageRef": "1a6635d8-afea-4f2b-abb6-27a202bad319",
        "flavorRef": "physical.o2.medium",
        "name": "bms_name01",
        "availability_zone": "az-dc-1",
        "networks": [
            {
                "uuid": "8470310b-bfa2-4edf-8f64-d15196b2b2c9"
            }
        ]
    }
}
```

Response Parameters

Parameter	Type	Description
server	Object	Specifies the BMS information. For details, see **Table 5-7**.

Table 5-7 server field data structure description

Parameter	Type	Description
security_groups	Array of objects	Specifies information about the BMS security group. For details, see **Table 5-8**.

Parameter	Type	Description
OS-DCF:diskConfig	String	Specifies the disk configuration. The value can be: • **MANUAL**: The API uses the partitioning scheme in the image and the file system to create a BMS. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The API uses a single partition with the same size as the disk of the target flavor to create a BMS. The API automatically adjusts the file system to adapt to the entire partition.
id	String	Specifies the BMS ID.
links	Array of objects	Specifies the shortcut links of the BMS. For details, see **Table 5-9**.
adminPass	String	Specifies the initial login password of the BMS administrator account.

Table 5-8 security_groups field data structure description

Parameter	Type	Description
name	String	Specifies the name of the security group to which the BMS belongs.

Table 5-9 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Example Response


```
{
    "server": {
        "security_groups": [
            {
                "name": "default"
            }
        ],
        "OS-DCF:diskConfig": "MANUAL",
        "links": [
            {
                "rel": "self",
                "href": "https://openstack.example.com/v2/c685484a8cc2416b97260938705deb65/servers/
9ab74d89-61e7-4259-8546-465fdebe4944"
            },
            {
                "rel": "bookmark",
                "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/servers/
9ab74d89-61e7-4259-8546-465fdebe4944"
            }
        ],
        "id": "9ab74d89-61e7-4259-8546-465fdebe4944",
        "adminPass": "RjdD3h8U2DBe"
    }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.3.2 Deleting a BMS (Native OpenStack API)

Function

This interface is used to delete a BMS.

URI

DELETE /v2.1/{project_id}/servers/{server_id}

Table 5-10 lists the parameters.

Table 5-10 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Deleting a BMS (ID: 9ab74d89-61e7-4259-8546-465fdebe4944)

```
DELETE https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/
9ab74d89-61e7-4259-8546-465fdebe4944
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.3.3 Querying Details About a BMS (Native OpenStack API)

Function

This API is used to query details about a BMS based on the BMS ID.

URI

GET /v2.1/{project_id}/servers/{server_id}

Table 5-11 lists the parameters.

Table 5-11 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying details about a BMS (ID: 9ab74d89-61e7-4259-8546-465fdebe4944)

```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/
9ab74d89-61e7-4259-8546-465fdebe4944
```

Response Parameters

Parameter	Type	Description
server	Object	Specifies BMS information. For details, see **Table 5-12**.

Table 5-12 server field data structure description

Parameter	Type	Description
name	String	Specifies the BMS name.
id	String	Specifies the unique ID of the BMS.

Parameter	Type	Description
status	String	Specifies the BMS status. Value range: • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
created	String	Specifies the time when the BMS was created. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.
updated	String	Specifies the time when the BMS was last updated. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T04:30:52Z.
flavor	Object	Specifies the BMS flavor. For details, see **Table 5-13**.
image	Object	Specifies the BMS image. For details, see **Table 5-14**.
tenant_id	String	Specifies the ID of the tenant owning the BMS. The value is in UUID format. This parameter specifies the same meaning as **project_id**.
key_name	String	Specifies the SSH key name.
user_id	String	Specifies the ID of the user to which the BMS belongs.
metadata	Map<String,String>	Specifies the BMS metadata. It is a user-defined metadata key-value pair. Each key or value contains a maximum of 255 characters.
hostId	String	Specifies the host ID of the BMS.

Parameter	Type	Description
addresses	Map<String,Array of **address** objects>	Specifies BMS network addresses. Specifies the VPC used by the BMS. • **key**: indicates the ID of the VPC used by the BMS. • **value** indicates the VPC details.
security_groups	Array of objects	Specifies the security groups to which the BMS belongs. For details, see **Table 5-17**.
links	Array of objects	Specifies shortcut links of the BMS. For details, see **Table 5-15**.
OS-DCF:diskConfig	String	Specifies the disk configuration method. This is an extended attribute. The value can be: • **MANUAL**: The API uses the partitioning scheme in the image and the file system to create a BMS. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The API uses a single partition with the same size as the disk of the target flavor to create a BMS. The API automatically adjusts the file system to adapt to the entire partition.
OS-EXT-AZ:availability_zone	String	Specifies the AZ where the BMS is located.
OS-EXT-SRV-ATTR:host	String	Specifies the hostname of the BMS. This is an extended attribute.
OS-EXT-SRV-ATTR:hypervisor_hostname	String	Specifies the hostname of the hypervisor. This is an extended attribute provided by the Nova virt driver.
OS-EXT-SRV-ATTR:instance_name	String	Specifies the BMS alias. This is an extended attribute.

Parameter	Type	Description
OS-EXT-STS:power_state	Integer	Specifies the BMS power status. This is an extended attribute. Options: 0, 1, 2, 3, and 4 • **0**: pending • **1**: running • **2**: paused • **3**: shutdown • **4**: crashed
OS-EXT-STS:task_state	String	Specifies the BMS task status. This is an extended attribute. Value range: • **rebooting**: The BMS is being restarted. • **reboot_started**: The BMS is normally restarted. • **reboot_started_hard**: The BMS is forcibly restarted. • **powering-off**: The BMS is being powered off. • **powering-on**: The BMS is being powered on. • **rebuilding**: The BMS is being rebuilt. • **scheduling**: The BMS is being scheduled. • **deleting**: The BMS is being deleted.
OS-EXT-STS:vm_state	String	Specifies the BMS status. This is an extended attribute. Value range: • **active**: The BMS is running. • **stopped**: The BMS is stopped.
OS-SRV-USG:launched_at	String	Specifies the time when the BMS was started. This is an extended attribute. The timestamp format is ISO 8601, for example, **2019-05-22T03:23:59.000000**.
OS-SRV-USG:terminated_at	String	Specifies the time when the BMS was deleted. This is an extended attribute. The timestamp format is ISO 8601, for example, **2019-05-22T04:23:59.000000**.

Parameter	Type	Description
os-extended-volumes:volumes_attached	Array of objects	Specifies the EVS disks attached to the BMS. For details, see **Table 5-18**.
accessIPv4	String	This is a reserved attribute.
accessIPv6	String	This is a reserved attribute.
fault	Object	Specifies the fault cause. If the BMS is faulty, this field is returned. For details, see **Table 5-19**.
config_drive	String	This is a reserved attribute.
progress	Integer	This is a reserved attribute.
description	String	Provides supplementary information about the pool. This parameter is added in micro version 2.19.
host_status	String	Specifies the status of the host that accommodates the BMS. • **UP**: The nova-compute is normal. • **UNKNOWN**: The nova-compute status is unknown. • **DOWN**: the nova-compute status is abnormal. • **MAINTENANCE**: The nova-compute is in the maintenance state. • Empty string: The BMS does not have host information. Added in micro version 2.16.
OS-EXT-SRV-ATTR:hostname	String	Specifies the hostname of the BMS. Added in micro version 2.3.
OS-EXT-SRV-ATTR:reservation_id	String	Specifies the reserved BMS IDs in a batch BMS creation situation. Added in micro version 2.3.
OS-EXT-SRV-ATTR:launch_index	Integer	Specifies the BMS startup sequence in a batch BMS creation situation. Added in micro version 2.3.
OS-EXT-SRV-ATTR:kernel_id	String	Specifies the UUID of the kernel image when the AMI image is used. In other scenarios, leave this parameter blank. Added in micro version 2.3.

Parameter	Type	Description
OS-EXT-SRV-ATTR:ramdisk_id	String	Specifies the UUID of the Ramdisk image when the AMI image is used. In other scenarios, leave this parameter blank. Added in micro version 2.3.
OS-EXT-SRV-ATTR:root_device_name	String	Specifies the device name of the BMS system disk, for example, **/dev/sda**. Added in micro version 2.3.
OS-EXT-SRV-ATTR:user_data	String	Specifies the **user_data** specified during BMS creation. The value is encoded using Base64 or an empty string.
locked	Boolean	Specifies whether the BMS is locked. • **true**: The BMS is locked. • **false**: The BMS is not locked. Added in micro version 2.9.
tags	Array of strings	Specifies tags of the BMS. This parameter is added in microversion 2.26. If the microversion is not used for query, the response does not contain the **tags** field. The value of this field meets the following requirements: • The key and value of a tag are connected using an equal sign (=), for example, **key=value**. • If the value is empty, only the key is returned.

Table 5-13 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the BMS type ID.
links	Array of objects	Specifies shortcut links of the BMS type. For details, see **Table 5-15**.

Table 5-14 image field data structure description

Parameter	Type	Description
id	String	Specifies the image ID of the BMS.
links	Array of objects	Specifies shortcut links of the BMS image. For details, see **Table 5-15**.

Table 5-15 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Table 5-16 address field data structure description

Parameter	Type	Description
addr	String	Specifies the IP address.
version	Integer	Specifies the type of the IP address. The value 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 type. This is an extended attribute. • **fixed**: indicates the private IP address. • **floating**: indicates the EIP.

Table 5-17 security_groups field data structure description

Parameter	Type	Description
name	String	• If no security group is specified during BMS creation, the **default** value is used. • If a security group is specified when you create the BMS, the value of this parameter is the security group name.

Table 5-18 os-extended-volumes:volumes_attached field data structure description

Parameter	Type	Description
id	String	Specifies the EVS disk ID.
delete_on_termination	Boolean	Specifies whether to delete the disk when deleting the BMS. • **true**: Yes • **false**: No Added in micro version 2.3.

Table 5-19 fault field data structure description

Parameter	Type	Description
message	String	Specifies the fault information.
code	Integer	Specifies the fault code.
details	String	Specifies the fault details.
created	String	Specifies the time when the fault occurred. The time is in ISO 8601 format.

Example Response

```
{
   "server": {
      "tenant_id": "c685484a8cc2416b97260938705deb65",
      "addresses": {
         "08a7715f-7de6-4ff9-a343-95ba4209f24a": [
{
                  "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:0e:c3:77",
                  "OS-EXT-IPS:type": "fixed",
                  "addr": "192.168.0.107",
                  "version": 4
               }
```



```
            ]
        },
        "metadata": {
            "op_svc_userid": "1311c433dd9b408886f57d695c229cbe"
        },
        "OS-EXT-STS:task_state": null,
        "OS-DCF:diskConfig": "MANUAL",
        "OS-EXT-AZ:availability_zone": "az-dc-1",
        "links": [
{
                "rel": "self",
                "href": "https://openstack.example.com/v2.1/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
            },
{
                "rel": "bookmark",
                "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
                }
        ],
        "OS-EXT-STS:power_state": 1,
        "id": "95bf2490-5428-432c-ad9b-5e3406f869dd",
        "os-extended-volumes:volumes_attached": [
{
                "id": "dfa375b5-9856-44ad-a937-a4802b6434c3"
            },
{
                "id": "bb9f1b27-843b-4561-b62e-ca18eeaec417"
            },
{
                "id": "86e801c3-acc6-465d-890c-d43ba493f553"
            },
{
                "id": "0994d3ac-3c6a-495c-a439-c597a4f08fa6"
                }
        ],
        "OS-EXT-SRV-ATTR:host": "bms.az1",
        "image": {
            "links": [
{
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/images/
1a6635d8-afea-4f2b-abb6-27a202bad319"
                }
            ],
            "id": "1a6635d8-afea-4f2b-abb6-27a202bad319"
        },
        "OS-SRV-USG:terminated_at": null,
        "accessIPv4": "",
        "accessIPv6": "",
        "created": "2017-05-24T06:14:05Z",
        "hostId": "e9c3ee0fcc58ab6085cf30df70b5544eab958858fb50d925f023e53e",
        "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova004@2",
        "key_name": "$key_name",
        "flavor": {
            "links": [
{
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/flavors/
physical.83.medium"
                }
            ],
            "id": "physical.83.medium"
        },
        "security_groups": [
{
                "name": "0011b620-4982-42e4-ad12-47c95ca495c4"
                }
        ],
```



```
    "config_drive": "",
    "OS-EXT-STS:vm_state": "active",
    "OS-EXT-SRV-ATTR:instance_name": "instance-0000ebd3",
    "user_id": "1311c433dd9b408886f57d695c229cbe",
    "name": "bms-83",
    "progress": 0,
    "OS-SRV-USG:launched_at": "2017-05-25T03:40:25.066078",
    "updated": "2017-05-25T03:40:25Z",
    "status": "ACTIVE"
        }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.3.4 Querying BMSs (Native OpenStack API)

Function

This API is used to query BMSs.

Constraints

- The query result returned by this interface includes both ECSs and BMSs. You need to filter out the BMSs using the flavor used to create the BMSs or the tags added to the BMSs during BMS creation.
- If the image is used as the search criteria, other search criteria and pagination criteria are not supported. If both the image and other search criteria are used, the BMSs are filtered out by image. If the image is not used as the search criteria, this interface has no restrictions.

URI

GET /v2.1/{project_id}/servers{?changes-since={changes-since}&image={image}&flavor={flavor}&name={name}&status={status}&limit={limit}&marker={marker}&tags={tags}&not-tags={not-tags}&reservation_id={reservation_id}&sort_key={sort_key}&sort_dir={sort_dir}}

Table 5-20 lists the parameters.

Table 5-20 Parameter description

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

Request Parameters

Parameter	Mandatory	Type	Description
changes-since	No	String	Specifies the timestamp of the last BMS status update. The parameter is in ISO 8601 time format, for example, **2013-06-09T06:42:18Z**.
image	No	String	Specifies the image ID. You can obtain the image ID from the IMS console or by calling the **Querying Images** API. **NOTE** If the image is used as the search criteria, other search criteria and pagination criteria are not supported. If both the image and other search criteria are used, the BMS details are filtered out by image. If the image is not used as the search criteria, this interface has no restrictions.
flavor	No	String	Specifies the flavor ID. You can obtain the flavor ID from the BMS console or using the **Querying BMS Flavors (Native OpenStack API)** API.

Parameter	Mandatory	Type	Description
name	No	String	Specifies the BMS name. This parameter supports fuzzy matching. For example, the regular expression **?name=bob** will return both **bob** and **bobb**. To obtain only **bob**, you can use a regular expression matching the basic database syntax, such as MySQL or PostgreSQL (official website: **https://www.postgresql.org/docs/9.2/static/functions-matching.html**).
status	No	String	Specifies the BMS status. Value range: • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
limit	No	Integer	Specifies the number of BMSs displayed on each page.
marker	No	String	Specifies the BMS ID to which the marker corresponds. The query will start from the next ID.
tags	No	String	Queries the BMSs with specified tags. Added in micro version 2.26.

Parameter	Mandatory	Type	Description
not-tags	No	String	Queries the BMSs with tags not containing the specified value. The value is a list of tag keys. **NOTE** If the tags added before the function upgrade are in the format of "Key.Value", query tags using "Key". For example, an existing tag is **a.b**. After the tag function upgrade, query the tag using "not-tags=a". Added in micro version 2.26.
reservation_id	No	String	Specifies the reserved ID, which can be used to query BMSs created in a batch. Added in micro version 2.26.
sort_key	No	String	Specifies the BMS sorting attribute, which can be the BMS UUID (**uuid**), BMS status (**vm_state**), BMS name (**display_name**), BMS task status (**task_state**), power status (**power_state**), creation time (**created_at**), last time when the BMS is updated (**updated_at**), and AZ (**availability_zone**). You can specify multiple **sort_key** and **sort_dir** pairs. The default sorting is the reverse order by **created_at**.
sort_dir	No	String	Specifies the sorting direction. • **asc**: The query results are displayed in ascending order. • **desc** (default value): The query results are displayed in descending order.

Example Request

- Querying BMSs
  GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers
- Querying BMSs with the **__type_baremetal** tag
  GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers?tags=__type_baremetal
- Querying BMSs with the **__type_baremetal** tag and with **bms-test01** contained in the name


```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers?
tags=__type_baremetal&name=bms-test01
```

Response Parameters

Parameter	Type	Description
servers	Array of objects	Specifies the BMS list. For details, see **Table 5-21**.

Table 5-21 servers field data structure description

Parameter	Type	Description
name	String	Specifies the BMS name.
id	String	Specifies the unique ID of the BMS.
links	Array of objects	Specifies shortcut links of the BMS. For details, see **Table 5-22**.

Table 5-22 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Example Response

```
{
   "servers": [
      {
         "name": "bms",
         "links": [
            {
               "rel": "self",
               "href": "https://openstack.example.com/v2.1/c685484a8cc2416b97260938705deb65/servers/
820abbd0-2d8b-4bc5-ae46-69cacfd4fbaa"
            },
            {
               "rel": "bookmark",
               "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/servers/
820abbd0-2d8e-4bc5-ae46-69cacfd4fbaa"
            }
```

```
        ],
        "id": "820abbd0-2d8e-4bc5-ae46-69cacfd4fbaa"
      }
   ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.3.5 Querying Details About BMSs (Native OpenStack API)

Function

This API is used to query details about BMSs.

Constraints

- The query result returned by this interface includes both ECS and BMS details. You need to filter out the BMS details using the flavor used to create the BMSs or the tags added to the BMSs during BMS creation.
- If the image is used as the search criteria, other search criteria and pagination criteria are not supported. If both the image and other search criteria are used, the BMS details are filtered out by image. If the image is not used as the search criteria, this interface has no restrictions.

URI

GET /v2.1/{project_id}/servers/detail{?changes-since={changes-since}&image={image}&flavor={flavor}&name={name}&status={status}&limit={limit}&marker={marker}&tags={tags}&not-tags={not-tags}&reservation_id={reservation_id}&sort_key={sort_key}&sort_dir={sort_dir}}

Table 5-23 lists the parameters.

Table 5-23 Parameter description

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

Request Parameters

Parameter	Mandatory	Type	Description
changes-since	No	String	Specifies the timestamp of the last BMS status update. The parameter is in ISO 8601 time format, for example, **2013-06-09T06:42:18Z**.
image	No	String	Specifies the image ID. You can obtain the image ID from the IMS console or by calling the **Querying Images** API. **NOTE** If the image is used as the search criteria, other search criteria and pagination criteria are not supported. If both the image and other search criteria are used, the BMS details are filtered out by image. If the image is not used as the search criteria, this interface has no restrictions.
flavor	No	String	Specifies the flavor ID. You can obtain the flavor ID from the BMS console or using the **Querying BMS Flavors (Native OpenStack API)** API.

Parameter	Mandatory	Type	Description
name	No	String	Specifies the BMS name. This parameter supports fuzzy matching. For example, the regular expression **?name=bob** will return both **bob** and **bobb**. To obtain only **bob**, you can use a regular expression matching the basic database syntax, such as MySQL or PostgreSQL (official website: **https://www.postgresql.org/docs/9.2/static/functions-matching.html**).
status	No	String	Specifies the BMS status. Value range: • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
limit	No	Integer	Specifies the number of BMSs displayed on each page.
marker	No	String	Specifies the BMS ID to which the marker corresponds. The query will start from the next ID.
tags	No	String	Queries the BMSs with specified tags. Added in micro version 2.26.

Parameter	Mandatory	Type	Description
not-tags	No	String	Queries the BMSs with tags not containing the specified value. The value is a list of tag keys. **NOTE** If the tags added before the function upgrade are in the format of "Key.Value", query tags using "Key". For example, an existing tag is **a.b**. After the tag function upgrade, query the tag using "not-tags=a". Added in micro version 2.26.
reservation_id	No	String	Specifies the reserved ID, which can be used to query BMSs created in a batch. Added in micro version 2.26.
sort_key	No	String	Specifies the BMS sorting attribute, which can be the BMS UUID (**uuid**), BMS status (**vm_state**), BMS name (**display_name**), BMS task status (**task_state**), power status (**power_state**), creation time (**created_at**), last time when the BMS is updated (**updated_at**), and AZ (**availability_zone**). You can specify multiple **sort_key** and **sort_dir** pairs. The default sorting is the reverse order by **created_at**.
sort_dir	No	String	Specifies the sorting direction. • **asc**: The query results are displayed in ascending order. • **desc** (default value): The query results are displayed in descending order.

Example Request

- Querying details about BMSs
  https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/detail
- Querying details about BMSs with the **__type_baremetal** tag
  https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/detail?tags=__type_baremetal
- Querying details about BMSs with the **__type_baremetal** tag and with **bms-test01** contained in the name


```
https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/detail?
tags=__type_baremetal&name=bms-test01
```

Response Parameters

Parameter	Type	Description
servers	Array of objects	Specifies details about the BMS. For details, see **Table 5-24**.

Table 5-24 server field data structure description

Parameter	Type	Description
name	String	Specifies the BMS name.
id	String	Specifies the unique ID of the BMS.
status	String	Specifies the BMS status. Value range: • **ACTIVE**: The BMS is running, stopped, or being deleted. • **BUILD**: The BMS is being created. • **ERROR**: The BMS is faulty. • **HARD_REBOOT**: The BMS is being forcibly restarted. • **REBOOT**: The BMS is being restarted. • **SHUTOFF**: The BMS is stopped, being started, being deleted, being rebuilt, being reinstalled, failed to be reinstalled, or frozen.
created	String	Specifies when the BMS was created. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.
updated	String	Specifies the last time when the BMS was updated. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T04:30:52Z.
flavor	Object	Specifies the BMS flavor. For details, see **Table 5-27**.
image	Object	Specifies the BMS image. For details, see **Table 5-28**.

Parameter	Type	Description
tenant_id	String	Specifies the ID of the tenant owning the BMS. The value is in UUID format. This parameter equals to **project_id**.
key_name	String	Specifies the SSH key name.
user_id	String	Specifies the user ID of the BMS.
metadata	Map<String,String>	Specifies the BMS metadata. It is a user-defined metadata key-value pair. Each key or value contains a maximum of 255 characters.
hostId	String	Specifies the host ID of the BMS.
addresses	Map<String,Array of **address** objects>	Specifies the network address of the BMS. Specifies the VPC used by the BMS. • **key**: indicates the ID of the VPC used by the BMS. • **value** indicates the VPC details.
security_groups	Array of objects	Specifies the security groups of the BMS. For details, see **Table 5-31**.
links	Array of objects	Specifies shortcut links of the BMS. For details, see **Table 5-29**.
OS-DCF:diskConfig	String	Specifies the disk configuration method. This is an extended attribute. The value can be: • **MANUAL**: The BMS is created using the partitioning scheme and file system in the image. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The BMS is created using a single partition with the same size as the disk of the target flavor. The API automatically adjusts the file system to adapt to the entire partition.
OS-EXT-AZ:availability_zone	String	Specifies the name of the AZ where the BMS is deployed. This is an extended attribute.
OS-EXT-SRV-ATTR:host	String	Specifies the hostname of the BMS. This is an extended attribute.

Parameter	Type	Description
OS-EXT-SRV-ATTR:hypervisor_hostname	String	Specifies the hostname of the hypervisor. This is an extended attribute provided by the Nova virt driver.
OS-EXT-SRV-ATTR:instance_name	String	Specifies the BMS alias. This is an extended attribute.
OS-EXT-STS:power_state	Integer	Specifies the BMS power status. This is an extended attribute. Value range: • **0**: pending • **1**: running • **2**: paused • **3**: shutdown • **4**: crashed
OS-EXT-STS:task_state	String	Specifies the BMS task status. This is an extended attribute. Value range: • **rebooting**: The BMS is being restarted. • **reboot_started**: The BMS is normally restarted. • **reboot_started_hard**: The BMS is forcibly restarted. • **powering-off**: The BMS is being powered off. • **powering-on**: The BMS is being powered on. • **rebuilding**: The BMS is being rebuilt. • **scheduling**: The BMS is being scheduled. • **deleting**: The BMS is being deleted.
OS-EXT-STS:vm_state	String	Specifies the BMS status. This is an extended attribute. Value range: • **active**: The BMS is running. • **shutoff**: The BMS is stopped. • **reboot**: The BMS is restarted.

Parameter	Type	Description
OS-SRV-USG:launched_at	String	Specifies the time when the BMS was started. This is an extended attribute. The timestamp format complies with ISO 8601, for example, **2019-05-22T03:23:59.000000**.
OS-SRV-USG:terminated_at	String	Specifies the time when the BMS was deleted. This is an extended attribute. The timestamp format complies with ISO 8601, for example, **2019-05-22T04:23:59.000000**.
os-extended-volumes:volumes_attached	Array of objects	Specifies the attached EVS disks of the BMS. For details, see **Table 5-32**.
accessIPv4	String	This is a reserved attribute.
accessIPv6	String	This is a reserved attribute.
fault	Object	Specifies the fault cause. If the BMS is faulty, this field will be returned. For details, see **Table 5-33**.
config_drive	String	This is a reserved attribute.
progress	Integer	This is a reserved attribute.
description	String	Specifies description about the BMS. This is a new parameter in micro version 2.19.
host_status	String	Specifies the status of theBMS host. • **UP**: The host is normal. • **UNKNOWN**: The host status is unknown. • **DOWN**: the host is abnormal. • **MAINTENANCE**: The host is being maintained. • Empty string: There is no BMS host information. This is a new parameter in micro version 2.16.
OS-EXT-SRV-ATTR:hostname	String	Specifies the hostname of the BMS. This is a new parameter in micro version 2.3.

Parameter	Type	Description
OS-EXT-SRV-ATTR:reservation_id	String	Specifies the reserved BMS IDs when BMSs are created in a batch. This is a new parameter in micro version 2.3.
OS-EXT-SRV-ATTR:launch_index	Integer	Specifies the startup sequence of the BMSs created in a batch. This is a new parameter in micro version 2.3.
OS-EXT-SRV-ATTR:kernel_id	String	Specifies the UUID of the kernel image when an AMI image is used. In other cases, leave this parameter blank. This is a new parameter in micro version 2.3.
OS-EXT-SRV-ATTR:ramdisk_id	String	Specifies the UUID of a RAM disk image when the AMI image is used. In other cases, leave this parameter blank. This is a new parameter in micro version 2.3.
OS-EXT-SRV-ATTR:root_device_name	String	Specifies the device name of the BMS system disk, for example, **/dev/sda**. This is a new parameter in micro version 2.3.
OS-EXT-SRV-ATTR:user_data	String	Specifies the **user_data** specified during BMS creation. The value is encoded using Base64 or an empty string.
locked	Boolean	Specifies whether the BMS is locked. • **true**: The BMS is locked. • **false**: The BMS is not locked. This is a new parameter in micro version 2.9.
tags	Array of strings	Specifies tags of the BMS. This is a new parameter in micro version 2.26. If you are not using this micro version, the response will not contain the **tags** parameter. A tag must meet the following requirements: • The key and value of a tag are connected using an equal sign (=), for example, **key=value**. • If the value is empty, only the key is returned.

Parameter	Type	Description
sys_tags	Array of objects	Specifies system tags of the BMS. For details, see **Table 5-25**.
enterprise_project_id	String	Specifies the enterprise project ID.
os:scheduler_hints	Object	Specifies scheduling information of the BMS. For details, see **Table 5-26**.

Table 5-25 sys_tags field data structure description

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

Table 5-26 os:scheduler_hints field data structure description

Parameter	Type	Description
group	Array of strings	Specifies the BMS group ID in UUID format.

Table 5-27 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the BMS type ID.
links	Array of objects	Specifies shortcut links of the BMS type. For details, see **Table 5-29**.

Table 5-28 image field data structure description

Parameter	Type	Description
id	String	Specifies the image ID of the BMS.
links	Array of objects	Specifies shortcut links of the BMS image. For details, see **Table 5-29**.

Table 5-29 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Table 5-30 address field data structure description

Parameter	Type	Description
addr	String	Specifies the IP address.
version	Integer	Specifies the type of the IP address. The value 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 type. This is an extended attribute. • **fixed**: indicates the private IP address. • **floating**: indicates the EIP.

Table 5-31 security_groups field data structure description

Parameter	Type	Description
name	String	• If no security group is specified during BMS creation, the **default** value is used. • If a security group is specified when you create the BMS, the value of this parameter is the security group name.

Table 5-32 os-extended-volumes:volumes_attached field data structure description

Parameter	Type	Description
id	String	Specifies the EVS disk ID.
delete_on_termination	Boolean	Specifies whether to delete the disk when deleting the BMS. • **true**: Yes • **false**: No Added in micro version 2.3.

Table 5-33 fault field data structure description

Parameter	Type	Description
message	String	Specifies the fault information.
code	Integer	Specifies the fault code.
details	String	Specifies the fault details.
created	String	Specifies the time when the fault occurred. The time is in ISO 8601 format.

Example Response


```
{
   "servers": [
{
        "tenant_id": "c685484a8cc2416b97260938705deb64",
        "addresses": {
            "08a7715f-7de6-4ff9-a343-95ba4209f24a": [
{
                    "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:0e:c4:77",
                    "OS-EXT-IPS:type": "fixed",
                    "addr": "192.168.0.107",
                    "version": 4
                }
            ]
        },
        "metadata": {
            "op_svc_userid": "1311c433dd9b408886f57d695c229cbe"
        },
        "OS-EXT-STS:task_state": null,
        "OS-DCF:diskConfig": "MANUAL",
        "OS-EXT-AZ:availability_zone": "az-dc-1",
        "links": [
{
                "rel": "self",
                "href": "https://openstack.example.com/v2.1/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
            },
{
                "rel": "bookmark",
                "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/servers/
```



```
95bf2490-5428-432c-ad9b-5e3406f869dd"
                }
            ],
            "OS-EXT-STS:power_state": 1,
            "id": "95bf2490-5428-432c-ad9b-5e3406f869dd",
            "os-extended-volumes:volumes_attached": [
{
                "id": "dfa375b5-9856-44ad-a937-a4802b6434c3"
            },
{
                "id": "bb9f1b27-843b-4561-b62e-ca18eeaec417"
            },
{
                "id": "86e801c3-acc6-465d-890c-d43ba493f553"
            },
{
                "id": "0994d3ac-3c6a-495c-a439-c597a4f08fa6"
                }
            ],
            "OS-EXT-SRV-ATTR:host": "bms.az1",
            "image": {
                "links": [
{
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/images/
1a6635d8-afea-4f2b-abb6-27a202bad319"
                }
                ],
                "id": "1a6635d8-afea-4f2b-abb6-27a202bad319"
            },
            "OS-SRV-USG:terminated_at": null,
            "accessIPv4": "",
            "accessIPv6": "",
            "created": "2017-05-24T06:14:05Z",
            "hostId": "e9c3ee0fcc58ab6085cf30df70b5544eab958858fb50d925f023e53e",
            "OS-EXT-SRV-ATTR:hypervisor_hostname": "nova004@2",
            "key_name": "$key_name",
            "flavor": {
                "links": [
{
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/flavors/
physical.83.medium"
                }
                ],
                "id": "physical.83.medium"
            },
            "security_groups": [
{
                "name": "0011b620-4982-42e4-ad12-47c95ca495c4"
                }
            ],
            "config_drive": "",
            "OS-EXT-STS:vm_state": "active",
            "OS-EXT-SRV-ATTR:instance_name": "instance-0000ebd3",
            "user_id": "1311c433dd9b408886f57d695c229cbe",
            "name": "bms",
            "progress": 0,
            "OS-SRV-USG:launched_at": "2017-05-25T03:40:25.066078",
            "updated": "2017-05-25T03:40:25Z",
            "status": "ACTIVE"
                }
    ]
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.4 BMS Status Management

5.4.1 Changing the Name of a BMS (Native OpenStack API)

Function

This interface is used to modify BMS information. Currently, only the name of the BMS can be changed.

URI

PUT /v2.1/{project_id}/servers/{server_id}

Table 5-34 lists the parameters.

Table 5-34 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
server	Yes	Object	Specifies the BMS data structure. For details, see **Table 5-35**.

Table 5-35 server field data structure description

Parameter	Mandatory	Type	Description
name	Yes	String	Specifies the new BMS name.

Example Request

Changing the name of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) to **new-server-test**

```
PUT https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd
{
    "server": {
        "name": "new-server-test"
    }
}
```

Response Parameters

Parameter	Type	Description
name	String	Specifies the BMS name.
id	String	Specifies the unique ID of the BMS.
status	String	Specifies the current status of the BMS. • **ACTIVE**: Running, Stopping, Deleting • **BUILD**: Creating • **ERROR**: Faulty • **HARD_REBOOT**: Forcibly Restarting • **REBOOT**: Restarting • **SHUTOFF**: Stopped, Starting, Deleting, Rebuilding, Reinstalling OS, OS Reinstallation Failed, Frozen
created	String	Specifies the time when the BMS was created. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T03:30:52Z.
updated	String	Specifies the time when the BMS was last updated. The timestamp format is YYYY-MM-DDTHH:MM:SSZ (ISO 8601), for example, 2019-05-22T04:30:52Z.

Parameter	Type	Description
flavor	Object	Specifies the BMS flavor information. For details, see **Table 5-36**.
image	Object	Specifies the BMS image. For details, see **Table 5-37**.
tenant_id	String	Specifies the ID of the tenant owning the BMS. The ID is in UUID format. This parameter specifies the same meaning as **project_id**.
user_id	String	Specifies the ID of the user to which the BMS belongs.
metadata	Object	Specifies the BMS metadata. For details, see **Table 5-38**.
hostId	String	Specifies the host ID of the BMS.
addresses	Object	Specifies the BMS network address. For details, see **Table 5-40**.
links	Array of objects	Specifies the shortcut links of the BMS. For details, see **Table 5-39**.
accessIPv4	String	This is a reserved attribute.
accessIPv6	String	This is a reserved attribute.
OS-DCF:diskConfig	String	Specifies the disk configuration method. This is an extended attribute. The value can be: • **MANUAL**: The API uses the partitioning scheme in the image and the file system to create a BMS. If the target flavor has a large disk, the API does not partition the remaining disk space. • **AUTO**: The API uses a single partition with the same size as the disk of the target flavor to create a BMS. The API automatically adjusts the file system to adapt to the entire partition.
progress	Integer	This is a reserved attribute.

Table 5-36 flavor field data structure description

Parameter	Type	Description
id	String	Specifies the flavor ID.
links	Array of objects	Specifies the shortcut link of the BMS flavor. For details, see **Table 5-39**.

Table 5-37 image field data structure description

Parameter	Type	Description
id	String	Specifies the ID of the BMS image.
links	Array of objects	Specifies the shortcut links of the BMS image. For details, see **Table 5-39**.

Table 5-38 metadata field data structure description

Parameter	Type	Description
User-defined field key and value pair	String	Specifies the key and value pair of the metadata. Each key or value contains a maximum of 255 characters.

Table 5-39 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. The value can be: • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Table 5-40 addresses field data structure description

Parameter	Type	Description
vpc_id	Array of objects	Specifies the VPC used by the BMS. • **key**: indicates the ID of the VPC used by the BMS. • **value**: indicates the VPC details. For details, see **Table 5-41**.

Table 5-41 address field data structure description

Parameter	Type	Description
addr	String	Specifies the IP address.
version	String	Specifies the version of the IP address.

Example Response


```
{
    "server": {
        "tenant_id": "c685484a8cc2416b97260938705deb65",
        "image": {
            "links": [
                {
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/images/
1a6635d8-afea-4f2b-abb6-27a202bad319"
                }
            ],
            "id": "1a6635d8-afea-4f2b-abb6-27a202bad319"
        },
        "accessIPv4": "",
        "addresses": {
            "08a7715f-7de6-4ff9-a343-95ba4209f24a": [
                {
                    "addr": "192.168.0.107",
                    "version": 4
                }
            ]
        },
        "metadata": {
            "op_svc_userid": "1311c433dd9b408886f57d695c229cbe"
        },
        "accessIPv6": "",
        "created": "2017-05-24T06:14:05Z",
        "hostId": "e9c3ee0fcc58ab6085cf30df70b5544eab958858fb50d925f023e53e",
        "flavor": {
            "links": [
                {
                    "rel": "bookmark",
                    "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/flavors/
physical.83.medium"
                }
            ],
            "id": "physical.83.medium"
        },
        "OS-DCF:diskConfig": "MANUAL",
```



```
      "user_id": "1311c433dd9b408886f57d695c229cbe",
      "name": "new-server-test",
      "progress": 0,
      "links": [
         {
            "rel": "self",
            "href": "https://openstack.example.com/v2/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
         },
         {
            "rel": "bookmark",
            "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/servers/
95bf2490-5428-432c-ad9b-5e3406f869dd"
         }
      ],
      "id": "95bf2490-5428-432c-ad9b-5e3406f869dd",
      "updated": "2017-05-25T03:40:25Z",
      "status": "ACTIVE"
   }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.4.2 Starting a BMS (Native OpenStack API)

Function

This API is used to start a single BMS.

URI

POST /v2.1/{project_id}/servers/{server_id}/action

Table 5-42 lists the parameters.

Table 5-42 Parameter description

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

Parameter	Mandatory	Description
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
os-start	Yes	null	Specifies the operation of starting the BMS. The data structure is empty.

Example Request

Starting a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
POST https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/action
{
    "os-start": {}
                }
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.4.3 Restarting a BMS (Native OpenStack API)

Function

This API is used to restart a single BMS.

Constraints

Currently, only forcible restart is supported.

URI

POST /v2.1/{project_id}/servers/{server_id}/action

Table 5-43 lists the parameters.

Table 5-43 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
reboot	Yes	Object	Specifies the operation of restarting the BMS. For details, see **Table 5-44**.

Table 5-44 reboot field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the type of the restart operation. • **SOFT**: soft restart • **HARD**: forcible restart **NOTE** Currently, value **SOFT** is invalid. All BMS restart operations are forcible restart.

Example Request

Restarting a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
POST https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/action
{
    "reboot": {
        "type": "HARD"
    }
}
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.4.4 Stopping a BMS (Native OpenStack API)

Function

This API is used to stop a single BMS. To stop multiple BMSs, you can use the **Stopping BMSs** API.

Constraints

- The **OS-EXT-STS:vm_state** attribute (BMS status) must be **active** or **error**.
- Currently, only forcible stopping is supported.

URI

POST /v2.1/{project_id}/servers/{server_id}/action

Table 5-45 lists the parameters.

Table 5-45 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
os-stop	Yes	Object	Specifies the operation of stopping the BMS. For details, see **Table 5-46**.

Table 5-46 os-stop field data structure description

Parameter	Mandatory	Type	Description
type	No	String	Specifies the type of the BMS stopping operation. • **SOFT**: normal BMS stopping • **HARD**: Forcible BMS stopping **NOTE** Currently, this parameter is invalid. All BMS stopping operations are forcible stopping.

Example Request

Stopping a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
POST https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/action
{
    "os-stop": {}
}
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.4.5 Obtaining the Remote Login Address (Native OpenStack API)

Function

This interface is used to obtain the address for remotely logging in to a BMS.

URL

POST /v2.1/{project_id}/servers/{server_id}/action

Table 5-47 lists the parameters.

Table 5-47 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Message

- Request parameters

Parameter	Mandatory	Type	Description
os-getSerialConsole	Yes	Object	Specifies the action to obtain the BMS remote login address. For details, see **Table 5-48**.

Table 5-48 os-getSerialConsole field data structure description

Parameter	Mandatory	Type	Description
type	Yes	String	Specifies the object. Set the value to **serial**.

- Example request

  Obtaining the remote login address of a BMS (ID: 47e9be4e-a7b9-471f-92d9-ffc83814e07a)

```
POST https://{ECS Endpoint}/v2/9c53a566cb3443ab910cf0daebca90c4/servers/47e9be4e-a7b9-471f-92d9-ffc83814e07a/action
{
    "os-getSerialConsole": {
        "type": "serial"
    }
}
```

Response Message

- Response parameters

  None

- Example response

```
{
    "console": {
        "url": "https://baremetal-consoleproxy.az1.dc1.domainname.com:8003/?
token=040134bb-9195-4029-9a62-550bce390258",
        type": "serial"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.5 BMS Metadata Management

5.5.1 Querying BMS Metadata (Native OpenStack API)

Function

The BMS metadata includes BMS basic information on the cloud platform, such as the BMS ID, hostname, and network information. This API is used to query the BMS metadata.

Constraints

Pagination query is not supported.

URI

GET /v2.1/{project_id}/servers/{server_id}/metadata

Table 5-49 lists the parameters.

Table 5-49 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying metadata of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/metadata
```

Response Parameters

Parameter	Type	Description
metadata	Map<String,String>	Specifies a user-defined metadata key-value pair. Each key or value contains a maximum of 255 characters.

Example Response

```
{
    "metadata": {
        "key": "value"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.5.2 Updating BMS Metadata (Native OpenStack API)

Function

This API is used to update BMS metadata.

- If the metadata does not contain the target field, the field is automatically added to the field.
- If the metadata contains the target field, the field value is automatically updated.

Constraints

The BMS **OS-EXT-STS:vm_state** attribute (BMS status) must be **active**, **stopped**, or **paused**.

URI

POST /v2.1/{project_id}/servers/{server_id}/metadata

Table 5-50 lists the parameters.

Table 5-50 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
metadata	Yes	Object	Specifies the user-defined metadata key and value pair. For details, see **Table 5-51**.

Table 5-51 metadata field data structure description

Parameter	Mandatory	Type	Description
User-defined field key and value pair	Yes	String	Specifies the user-defined metadata key and value pair. • Each key or value contains a maximum of 255 characters. • The key does not support the following special characters: :`~!@#$%^&*()=+<,>?/'";{[]}|\ • The value does not support the following special characters: \"

Example Request

Updating the metadata of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) to customize a key-value pair

```
POST https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/metadata
{
    "metadata": {
        "key": "value"
    }
}
```

Response Parameters

Parameter	Type	Description
metadata	Object	Specifies the user-defined metadata key and value pair. For details, see **Table 5-52**.

Table 5-52 metadata field data structure description

Parameter	Type	Description
User-defined field key and value pair	String	Specifies the key and value pair of the metadata. • Each key or value contains a maximum of 255 characters. • The value does not support the following special characters: \"

Example Response

```
{
   "metadata": {
      "key": "value"
   }
   }
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.5.3 Modifying Specified BMS Metadata (Native OpenStack API)

Function

This API is used to modify specified BMS metadata.

Constraints

The BMS **OS-EXT-STS:vm_state** attribute (BMS status) must be **active**, **stopped**, or **paused**.

URI

PUT /v2.1/{project_id}/servers/{server_id}/metadata/{key}

Table 5-53 lists the parameters.

Table 5-53 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
key	Yes	Specifies the BMS metadata key value to be modified.

Request Parameters

Parameter	Mandatory	Type	Description
meta	Yes	Object	Specifies the user-defined metadata key and value pair. For details, see **Table 5-54**.

Table 5-54 meta field data structure description

Parameter	Mandatory	Type	Description
User-defined field key and value pair	Yes	String	Specifies the user-defined metadata key and value pair. • Each key or value contains a maximum of 255 characters. • The key does not support the following special characters: :`~!@#$%^&*()=+<,>?/'"; {[]}|\ • The value does not support the following special characters: \"

Example Request

Modifying the metadata of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) to customize a key-value pair

```
PUT https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/metadata/{key}
{
    "meta": {
        "key": "value"
    }
}
```

Response Parameters

Parameter	Type	Description
meta	Object	Specifies the user-defined metadata key and value pair. For details, see **Table 5-55**.

Table 5-55 meta field data structure description

Parameter	Type	Description
User-defined field key and value pair	String	Specifies the user-defined metadata key and value pair. Each key or value contains a maximum of 255 characters.

Example Response

```
{
    "meta": {
        "key": "value"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.5.4 Deleting Specified BMS Metadata (Native OpenStack API)

Function

This API is used to delete specified BMS metadata.

Constraints

The BMS **OS-EXT-STS:vm_state** attribute (BMS status) must be **active**, **stopped**, or **paused**.

URI

DELETE /v2.1/{project_id}/servers/{server_id}/metadata/{key}

Table 5-56 lists the parameters.

Table 5-56 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
key	Yes	Specifies the BMS metadata key value to be deleted.

Request Parameters

None

Example Request

Deleting metadata of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
DELETE https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/metadata/{key}
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.6 BMS IP Address Query

5.6.1 Querying IP Addresses of a BMS (Native OpenStack API)

Function

This API is used to query private IP addresses of a BMS.

Constraints

Pagination query is not supported.

URI

GET /v2.1/{project_id}/servers/{server_id}/ips

Table 5-57 lists the parameters.

Table 5-57 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying the private IP addresses of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/ips
```

Response Parameters

Parameter	Type	Description
addresses	Map<String,Array of **address** objects>	Specifies the VPC used by the BMS. • **key**: indicates the ID of the VPC used by the BMS. • **value** indicates the VPC details.

Table 5-58 address parameter structure description

Parameter	Type	Description
version	Integer	Specifies the IP address version. The value can be: • **4**: IPv4 address • **6**: IPv6 address
addr	String	Specifies the IP address.

Example Response

```
{
    "addresses": {
        "08a7715f-7de6-4ff9-a343-95ba4209f24a": [
            {
                "version": 4,
                "addr": "192.168.2.90"
            }
        ]
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.6.2 Querying the Specified IP Address of a BMS (Native OpenStack API)

Function

This API is used to query the specified IP address of a BMS based on the network name.

URI

GET /v2.1/{project_id}/servers/{server_id}/ips/{vpc_id}

Table 5-59 lists the parameters.

Table 5-59 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
vpc_id	Yes	Specifies the ID of the VPC where the BMS is located.

Request Parameters

None

Example Request

Querying the IP address of a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) in a specified VPC

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/ips/{vpc_id}
```

Response Parameters

Parameter	Type	Description
Vpc_id	Array of **address** objects	Specifies the VPC used by the BMS. **Vpc_id** indicates the ID of the VPC used by the BMS.

Table 5-60 Network parameter structure description

Parameter	Type	Description
version	Integer	Specifies the IP address version. The value can be: • **4**: IPv4 address • **6**: IPv6 address
addr	String	Specifies the IP address.

Example Response

```
{
    "Vpc_id": [
        {
            "version": 4,
            "addr": "192.168.1.159"
        }
    ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.7 BMS Flavor Query

5.7.1 Querying BMS Flavors (Native OpenStack API)

Function

This API is used to query BMS flavors.

Constraints

The flavors you obtained using this API are all the flavors in the system. The flavors whose names starting with **physical** are BMS flavors and can be used to create BMSs.

URI

GET /v2.1/{project_id}/flavors/detail{?minDisk={minDisk}&minRam={minRam}&sort_key={sort_key}&sort_dir={sort_dir}}

Table 5-61 lists the parameters.

Table 5-61 Parameter description

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

Request Parameters

Table 5-62 lists the optional parameters that can be used to query BMS flavors.

Table 5-62 Optional parameters

Parameter	Mandatory	Type	Description
minDisk	No	String	Specifies the minimum disk size in GB. Only the BMSs with a disk size greater than or equal to the minimum size can be queried.
minRam	No	String	Specifies the minimum memory size in MB. Only the BMSs with the memory size greater than or equal to the minimum size can be queried.

Parameter	Mandatory	Type	Description
sort_key	No	String	Specifies the sorting field. The default value is **flavorid**. The value of this parameter can also be **name**, **memory_mb**, **vcpus**, **root_gb**, or **flavorid**.
sort_dir	No	String	Specifies the sorting of BMS flavors. The value can be **asc** or **desc**, and is **asc** by default.

Example Request

- Querying BMS flavors
  GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/flavors/detail
- Querying BMS flavors with the minimum disk of 3,725 GB
  GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/flavors/detail?minDisk=3725

Response Parameters

Parameter	Type	Description
flavors	Array of objects	Specifies BMS flavors. For details, see **Table 5-63**.

Table 5-63 flavors field data structure description

Parameter	Type	Description
id	String	Specifies the BMS flavor ID.
name	String	Specifies the BMS flavor name.
vcpus	Integer	Specifies the number of CPU cores in the BMS flavor.
ram	Integer	Specifies the memory size (MB) in the BMS flavor.
disk	Integer	Specifies the disk size (GB) in the BMS flavor.
swap	String	This is a reserved attribute.
OS-FLV-EXT-DATA:ephemeral	Integer	This is a reserved attribute.

Parameter	Type	Description
OS-FLV-DISABLED:disabled	Boolean	This is a reserved attribute.
rxtx_factor	Float	This is a reserved attribute.
os-flavor-access:is_public	Boolean	This is a reserved attribute.
links	Array of objects	Specifies shortcut links of the BMS flavor. For details, see **Table 5-64**.

Table 5-64 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Example Response

```
{
   "flavors": [
      {
         "name": "physical.o2.medium",
         "links": [
            {
               "href": "https://openstack.example.com/v2/c685484a8cc2416b97260938705deb65/flavors/
physical.o2.medium",
               "rel": "self"
            },
            {
               "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/flavors/
physical.o2.medium",
               "rel": "bookmark"
             }
         ],
         "ram": 321725,
         "OS-FLV-DISABLED:disabled": false,
         "vcpus": 56,
         "swap": "",
         "os-flavor-access:is_public": true,
         "rxtx_factor": 1,
         "OS-FLV-EXT-DATA:ephemeral": 0,
         "disk": 3725,
         "id": "physical.o2.medium"
      }
```



```
    ]
        }
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.7.2 Querying Details About a BMS Flavor (Native OpenStack API)

Function

This API is used to query details about a BMS flavor, such as the flavor name, CPU cores, and memory.

URI

GET /v2.1/{project_id}/flavors/{flavor_id}

Table 5-65 lists the parameters.

Table 5-65 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
flavor_id	Yes	Specifies the flavor ID. You can obtain the flavor ID from the BMS console or using the **Querying BMS Flavors (Native OpenStack API)** API.

Request Parameters

None

Example Request

Query details about a flavor (ID: physical.o2.medium)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/flavors/physical.o2.medium
```

Response Parameters

Parameter	Type	Description
flavor	Object	Specifies a BMS flavor. For details, see **Table 5-66**.

Table 5-66 flavor field description

Parameter	Type	Description
id	String	Specifies the BMS flavor ID.
name	String	Specifies the BMS flavor name.
vcpus	Integer	Specifies the number of CPU cores in the BMS flavor.
ram	Integer	Specifies the memory size (MB) in the BMS flavor.
disk	Integer	Specifies the disk size (GB) in the BMS flavor.
swap	String	This is a reserved attribute.
OS-FLV-EXT-DATA:ephemeral	Integer	This is a reserved attribute.
OS-FLV-DISABLED:disabled	Boolean	This is a reserved attribute.
rxtx_factor	Float	This is a reserved attribute.
os-flavor-access:is_public	Boolean	This is a reserved attribute.
links	Array of objects	Specifies shortcut links of the BMS flavor. For details, see **Table 5-67**.

Table 5-67 links field data structure description

Parameter	Type	Description
rel	String	Specifies the shortcut link marker name. • **self**: resource link that contains the version number. It is used when immediate tracing is required. • **bookmark**: resource link that can be stored for a long time.
href	String	Specifies the corresponding shortcut link.

Example Response

```
{
   "flavor": {
      "name": "physical.o2.medium",
      "links": [
         {
            "href": "https://openstack.example.com/v2/c685484a8cc2416b97260938705deb65/flavors/
physical.o2.medium",
            "rel": "self"
         },
         {
            "href": "https://openstack.example.com/c685484a8cc2416b97260938705deb65/flavors/
physical.o2.medium",
            "rel": "bookmark"
         }
      ],
      "ram": 192705,
      "OS-FLV-DISABLED:disabled": false,
      "vcpus": 24,
      "swap": "",
      "os-flavor-access:is_public": true,
      "rxtx_factor": 1,
      "OS-FLV-EXT-DATA:ephemeral": 0,
      "disk": 1862,
      "id": "physical.o2.medium"
   }
            }
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.7.3 Querying Details About extra_specs Parameters of a BMS Flavor (Native OpenStack API)

Function

extra_specs parameters specify the key-value pair of a BMS flavor. For example, **baremetal:extBootType** specifies the boot device of the BMS. Its value can be **LocalDisk** (local disk) or **Volume** (EVS disk). If you want to check whether a flavor supports quick provisioning, you can call this API.

URI

GET /v2.1/{project_id}/flavors/{flavor_id}/os-extra_specs

Table 5-68 lists the parameters.

Table 5-68 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
flavor_id	Yes	Specifies the flavor ID. You can obtain the flavor ID from the BMS console or using the **Querying BMS Flavors (Native OpenStack API)** API.

Request Parameters

None

Example Request

Querying details about extra_specs parameters of a flavor (ID: physical.s2.medium)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/flavors/physical.s2.medium/os-extra_specs
```

Response Parameters

Parameter	Type	Description
extra_specs	Object	Specifies the key-value pair of a BMS flavor. • **capabilities:cpu_arch**: specifies the CPU architecture of the BMS. The value can be **x86_64** (for x86 servers) or **aarch64** (for ARM servers). • **baremetal:disk_detail**: specifies the disk description. • **capabilities:hypervisor_type**: specifies the hypervisor type. The value is fixed at **ironic**. • **baremetal:__support_evs**: specifies whether to support EVS disks. The value can be **true** or **false**. If the flavor does not contain this parameter, EVS disks are not supported either. • **baremetal:extBootType**: specifies the boot device of the BMS. The value can be **LocalDisk** (local disk) or **Volume** (EVS disk). • **baremetal:net_num**: specifies the number of NICs that can be attached to a BMS. • **baremetal:netcard_detail**: specifies description of the NIC. • **baremetal:cpu_detail**: specifies description of the CPU. • **resource_type**: specifies the resource type. The value is fixed at **ironic**. • **baremetal:memory_detail**: specifies description of the memory.

Example Response

```
{
    "extra_specs": {
        "capabilities:cpu_arch": "x86_64",
        "baremetal:disk_detail": "SAS 8T",
        "capabilities:hypervisor_type": "ironic",
        "baremetal:__support_evs": "true",
        "baremetal:extBootType": "LocalDisk",
        "capabilities:board_type": "s2m",
        "baremetal:net_num": "2",
        "baremetal:netcard_detail": "2*10GE",
        "baremetal:cpu_detail": "2*8coreIntel Xeon E5-2667 V43.2GHz",
        "resource_type": "ironic",
        "baremetal:memory_detail": "256GB DDR4 RAM(GB)"
    }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.8 BMS NIC Management

5.8.1 Querying Information About BMS NICs (Native OpenStack API)

Function

This API is used to query information about BMS NICs, such as the MAC addresses and private IP addresses.

URI

GET /v2.1/{project_id}/servers/{server_id}/os-interface

Table 5-69 lists the parameters.

Table 5-69 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying information about NICs bound to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-interface
```

Response Parameters

Parameter	Type	Description
interfaceAttachments	Array of objects	Specifies information about NICs of the BMS. For details, see **Table 5-70**.

Table 5-70 interfaceAttachments field data structure description

Parameter	Type	Description
port_state	String	Specifies the status of the NIC port. The value can be **ACTIVE**, **BUILD**, or **DOWN**.
fixed_ips	Array of objects	Specifies the NIC private IP address. For details, see **Table 5-71**.
net_id	String	Specifies the ID of the subnet (**network_id**) to which the NIC ports belong.
port_id	String	Specifies the ID of the NIC port.
mac_addr	String	Specifies the MAC address of the NIC.

Table 5-71 fixed_ips field data structure description

Parameter	Type	Description
subnet_id	String	Specifies the ID of the subnet (**subnet_id**) corresponding to the private IP address of the NIC.
ip_address	String	Specifies the NIC private IP address.

Example Response

```
{
   "interfaceAttachments": [
      {
         "port_state": "ACTIVE",
         "fixed_ips": [
            {
               "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef",
               "ip_address": "192.168.1.3"
            }
         ],
         "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
         "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
         "mac_addr": "fa:16:3e:4c:2c:30"
      }
   ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.8.2 Querying Information About a Specified BMS NIC (Native OpenStack API)

Function

This API is used to query information about a specified BMS NIC based on the NIC ID.

URI

GET /v2.1/{project_id}/servers/{server_id}/os-interface/{id}

Table 5-72 lists the parameters.

Table 5-72 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
id	Yes	Specifies the ID of the NIC. You can obtain the NIC ID from the **NICs** tab page on the BMS details page or by calling the **Querying Information About BMS NICs (Native OpenStack API)** API. (The NIC ID is the value of **port_id**).

Request Parameters

None

Example Request

Querying information about a NIC (ID: ce531f90-199f-48c0-816c-13e38010b442) bound to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-interface/ce531f90-199f-48c0-816c-13e38010b442
```

Response Parameters

Parameter	Type	Description
interfaceAttachment	Object	Specifies information about the specified BMS NIC. For details, see **Table 5-73**.

Table 5-73 interfaceAttachment field data structure description

Parameter	Type	Description
port_state	String	Specifies the status of the NIC port. The value can be **ACTIVE**, **BUILD**, or **DOWN**.
fixed_ips	Array of objects	Specifies the IP addresses of NICs. For details, see **Table 5-74**.
net_id	String	Specifies the ID of the subnet (**network_id**) to which the NIC ports belong.
port_id	String	Specifies the ID of the NIC port.
mac_addr	String	Specifies the MAC address of the NIC.

Table 5-74 fixed_ips field data structure description

Parameter	Type	Description
subnet_id	String	Specifies the ID of the subnet (**subnet_id**) corresponding to the private IP address of the NIC.
ip_address	String	Specifies the NIC IP address.

Example Response

```
{
   "interfaceAttachment": {
      "port_state": "ACTIVE",
      "fixed_ips": [
{
            "subnet_id": "f8a6e8f8-c2ec-497c-9f23-da9616de54ef",
            "ip_address": "192.168.1.3"
               }
         ],
      "net_id": "3cb9bc59-5699-4588-a4b1-b87f96708bc6",
      "port_id": "ce531f90-199f-48c0-816c-13e38010b442",
      "mac_addr": "fa:16:3e:4c:2c:30"
               }
               }
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.9 BMS Disk Management

5.9.1 Querying Information About the Disks Attached to a BMS (Native OpenStack API)

Function

This API is used to query information about the EVS disks attached to a BMS.

URI

GET /v2.1/{project_id}/servers/{server_id}/os-volume_attachments

Table 5-75 lists the parameters.

Table 5-75 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying EVS disks attached to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-volume_attachments
```

Response Parameters

Parameter	Type	Description
volumeAttachments	Object	Specifies information about the disks attached to the BMS. For details, see Table 5-76.

Table 5-76 volumeAttachments field data structure description

Parameter	Type	Description
device	String	Specifies the mount directory, for example, **/dev/vdb**.
id	String	Specifies the ID of the attached resource.
serverId	String	Specifies the ID of the BMS to which the disks are attached.
volumeId	String	Specifies the IDs of the EVS disks attached to the BMS.

Example Response

```
{
    "volumeAttachments": {
        "device": "/dev/vdb",
        "serverId": "820abbd0-2d8e-4bc5-ae46-69cacfd4fbaa",
        "id": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
        "volumeId": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.9.2 Querying Information About a Disk Attached to a BMS (Native OpenStack API)

Function

This API is used to query information about a single disk attached to a BMS based on the disk ID.

URI

GET /v2.1/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id}

Table 5-77 lists the parameters.

Table 5-77 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
volume_id	Yes	Specifies the EVS disk ID. You can query attached EVS disks attached to a BMS using the **Querying Information About the Disks Attached to a BMS (Native OpenStack API)** API.

Request Parameters

None

Example Request

Querying information about an EVS disk (ID: b53f23bd-ee8f-49ec-9420-d1acfeaf91d6) attached to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
GET https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-volume_attachments/b53f23bd-ee8f-49ec-9420-d1acfeaf91d6
```

Response Parameters

Parameter	Type	Description
volumeAttachment	Object	Specifies information about the disk attached to the BMS. For details, see **Table 5-78**.

Table 5-78 volumeAttachment field data structure description

Parameter	Type	Description
device	String	Specifies the mount directory, for example, **/dev/vdb**.
id	String	Specifies the ID of the attached resource.
serverId	String	Specifies the ID of the BMS to which the disks are attached.
volumeId	String	Specifies the ID of the disk attached to the BMS.

Example Response

```
{
   "volumeAttachment": {
      "device": "/dev/vdb",
      "serverId": "820abbd0-2d8e-4bc5-ae46-69cacfd4fbaa",
      "id": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
      "volumeId": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6"
   }
            }
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.9.3 Attaching an EVS Disk to a BMS (Native OpenStack API)

Function

This API is used to attach an EVS disk to a BMS.

Constraints

- A bootable disk cannot be attached to a BMS.
- A disk cannot be attached to a BMS when the BMS is in the **PAUSED** state, which is specified using the **OS-EXT-STS:vm_state** parameter.
- Only a shared disk or a disk in the **available** state can be attached to a BMS.
- Only EVS disks whose device type is **SCSI** can be attached to a BMS.

URI

POST /v2.1/{project_id}/servers/{server_id}/os-volume_attachments

Table 5-79 lists the parameters.

Table 5-79 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
volumeAttachment	Yes	Object	Specifies the disks to be attached. For details, see **Table 5-80**.

Table 5-80 volumeAttachment field data structure description

Parameter	Mandatory	Type	Description
volumeId	Yes	String	Specifies the ID of the disk to be attached to a BMS.
device	No	String	Specifies the mount point, such as **/dev/sda** and **/dev/sdb**. The new disk mount point cannot be the same as an existing one. The mount point must be specified based on the sequence of existing device names. Otherwise, the system automatically generates a mount point.

Example Request

Attaching an EVS disk (ID: b53f23bd-ee8f-49ec-9420-d1acfeaf91d6) to a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd) with the mount point of **/dev/sdb**

```
POST https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-volume_attachments
{
    "volumeAttachment": {
        "volumeId": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
        "device": "/dev/sdb"
    }
}
```

Response Parameters

Parameter	Type	Description
volumeAttachment	Object	Specifies the disks attached to a BMS. For details, see **Table 5-81**.

Table 5-81 volumeAttachment field data structure description

Parameter	Type	Description
device	String	Specifies the device name, for example, **/dev/vdb**.
serverId	String	Specifies the ID of the BMS to which the disk is to be attached. The ID is in UUID format.

Parameter	Type	Description
id	String	Specifies the disk UUID.
volumeId	String	Specifies the attaching ID, which is the same as the UUID.

Example Response

```
{
    "volumeAttachment": {
        "id": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
        "volumeId": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
        "serverId": "820abbd0-2d8e-4bc5-ae46-69cacfd4fbaa",
        "device": "/dev/vdb"
    }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.9.4 Detaching an EVS Disk from a BMS (Native OpenStack API)

Function

This API is used to detach an EVS disk from a BMS.

Constraints

If a BMS is stopped, disks can be detached from it without any limitation on the OS. If a BMS is in running state, the constraints are as follows:

- Before detaching an EVS disk from a Linux BMS, log in to the BMS, run the **unmount** command to disassociate the disk to be detached from the file system, and ensure that no program is reading data from or writing data to the disk. Otherwise, the disk will fail to be detached.
- Before detaching an EVS disk from a running Windows BMS, ensure that no program is reading data from or writing data to the disk. Otherwise, data will be lost.

URI

DELETE /v2.1/{project_id}/servers/{server_id}/os-volume_attachments/{volume_id}

Table 5-82 lists the parameters.

Table 5-82 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
volume_id	Yes	Specifies the EVS disk ID. You can query attached EVS disks attached to a BMS using the **Querying Information About the Disks Attached to a BMS (Native OpenStack API)** API.

Request Parameters

None

Example Request

Detaching an EVS disk (ID: b53f23bd-ee8f-49ec-9420-d1acfeaf91d6) attached from a BMS (ID: 95bf2490-5428-432c-ad9b-5e3406f869dd)

```
DELETE https://{ECS Endpoint}/v2.1/c685484a8cc2416b97260938705deb65/servers/95bf2490-5428-432c-ad9b-5e3406f869dd/os-volume_attachments/b53f23bd-ee8f-49ec-9420-d1acfeaf91d6
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.10 BMS SSH Key Pair Management

5.10.1 Querying SSH Key Pairs (Native OpenStack API)

Function

This interface is used to query SSH key pairs and to display the query results in a list.

Constraints

Pagination query is not supported.

URI

GET /v2.1/{project_id}/os-keypairs

Table 5-83 lists the parameters.

Table 5-83 Parameter description

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

Request Parameters

None

Example Request

Querying SSH key pairs in a project (ID: bbf1946d374b44a0a2a95533562ba954)

```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/os-keypairs
```


Response Parameters

Parameter	Type	Description
keypairs	Array of objects	Specifies key pairs. For details, see **Table 5-84**.

Table 5-84 keypairs field data structure description

Parameter	Type	Description
keypair	Object	Specifies details about a key pair. For details, see **Table 5-85**.

Table 5-85 keypair field data structure description

Parameter	Type	Description
fingerprint	String	Specifies fingerprint information about the key pair.
name	String	Specifies the key pair name.
type	String	Specifies the key type, which is **ssh** by default. This field is supported in microversions later than 2.2.
public_key	String	Specifies information about the public key in the key pair.

Example Response

```
{
   "keypairs": [
      {
         "keypair": {
            "fingerprint": "15:b0:f8:b3:f9:48:63:71:cf:7b:5b:38:6d:44:2d:4a",
            "name": "keypair-test",
            "type": "ssh",
            "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQC+Eo/
RZRngaGTkFs7I62ZjsIlO79KklKbMXi8F+KITD4bVQHHn+kV
+4gRgkgCRbdoDqoGfpaDFs877DYX9n4z6FrAIZ4PES8TNKhatifpn9NdQYWA+IkU8CuvlEKGuFpKRi/k7JLos/
gHi2hy7QUwgtRvcefvD/vgQZOVw/mGR9Q== Generated-by-Nova"
         }
      }
   ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.10.2 Querying an SSH Key Pair (Native OpenStack API)

Function

This interface is used to query a specified SSH key pair based on the key pair name.

URI

GET /v2.1/{project_id}/os-keypairs/{keypair_name}

Table 5-86 lists the parameters.

Table 5-86 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
keypair_name	Yes	Specifies the key pair name. You can obtain the key pair name by calling the **Querying SSH Key Pairs (Native OpenStack API)** API.

Request Parameters

None

Example Request

Querying details about a key pair (name: keypair-test)

```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/os-keypairs/keypair-test
```

Response Parameters

Parameter	Type	Description
keypair	Object	Specifies the SSH key pair. For details, see **Table 5-87**.

Table 5-87 keypair field data structure description

Parameter	Type	Description
public_key	String	Specifies information about the public key in the key pair.
name	String	Specifies the key pair name.
fingerprint	String	Specifies fingerprint information about the key pair.
created_at	String	Specifies the time when the key pair was created. The timestamp format is ISO 8601, for example, **2019-05-07T12:06:13.681238**.
deleted	Boolean	Specifies the deleted key pair. • **true**: indicates that the key has been deleted. • **false**: indicates that the key is not deleted.
deleted_at	String	Specifies the time when the key pair was deleted. The timestamp format is ISO 8601, for example, **2019-05-07T12:06:13.681238**.
id	String	Specifies the key pair ID.
updated_at	String	Specifies the time when the key pair was updated. The timestamp format is ISO 8601, for example, **2019-05-07T12:06:13.681238**.
user_id	String	Specifies information about the user to which the key pair belongs.

Example Response

```
{
    "keypair": {
```

```
        "created_at": "2019-05-07T12:06:13.681238",
        "deleted": false,
        "deleted_at": null,
        "fingerprint": "9d:00:f4:d7:26:6e:52:06:4c:c1:d3:1d:fd:06:66:01",
        "id": 1,
        "name": "keypair-3582d8b7-e588-4aad-b7f7-f4e76f0e4314",
        "public_key": "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDYJrTVpcMwFqQy/
oMvtUSRofZdSRHEwrsX8AYkRvn2ZnCXM+b6+GZ2NQuuWj+ocznlnwiGFQDsL/yeE+/
kurqcPJFKKp60mToXIMyzioFxW88fJtwEWawHKAclbHWpR1t4fQ4DS+/sIbX/Yd9btlVQ2tpQjodGDbM9Tr9/+/
3i6rcR+EoLqmbgCgAiGiVV6VbM2Zx79yUwd+GnQejHX8BlYZoOjCnt3NREsITcmWE9FVFy6TnLmahs3FkEO/
QGgWGkaohAJlsgaVvSWGgDn2AujKYwyDokK3dXyeX3m2Vmc3ejiqPa/C4nRrCOlko5nSgV/
9IXRx1ERImsqZnE9usB Generated-by-Nova",
        "updated_at": null,
        "user_id": "fake"
    }
}
```


Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.10.3 Creating or Importing an SSH Key Pair (Native OpenStack API)

Function

This API is used to create an SSH key pair or import a public key to generate a key pair.

After an SSH key pair is generated, download the private key to a local directory. Then, you can use this private key to log in to the BMS. For BMS security purposes, a private key can be downloaded only once. Keep it secure.

URI

POST /v2.1/{project_id}/os-keypairs

Table 5-88 lists the parameters.

Table 5-88 Parameter description

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

Request Parameters

NOTE

When creating an SSH key pair, you only need to configure **name**. When importing a public SSH key, you must also configure **public_key**.

Parameter	Mandatory	Type	Description
keypair	Yes	Object	Specifies the created or imported SSH key pair. For details, see **Table 5-89**.

Table 5-89 keypair field data structure description

Parameter	Mandatory	Type	Description
public_key	No	String	Specifies the imported public key. The maximum size of the imported public key is 1024 bytes. Note: If the length of the public key to be imported exceeds 1024 bytes, the public key import to the BMS will fail.
name	Yes	String	Specifies the key pair name. The new key pair name cannot be the same as an existing one.

Example Request

Creating or importing an SSH key pair (name: keypair-7d7c3650-dabe-4eb0-b904-5c464453c043) with the public key **ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQC9mC3WZN9UGLxgPBpP7H5jZMc6pKwOoSgre8yun6REFktn/Kz7DUt9jaR1UJyRzHxITfCfAIgSxPdGqB/oF1suMyWgu5i0625vavLB5z5kC8Hq3qZJ9zJO1poE1kyD**

+htiTtPWJ88e12xuH2XB/CZN9OpEiF98hAagiOE0EnOS5Q== Generated by Nova\n

```
POST https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/os-keypairs
{
    "keypair": {
        "name": "keypair-7d7c3650-dabe-4eb0-b904-5c464453c043",
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQC9mC3WZN9UGLxgPBpP7H5jZMc6pKwOoSgre8yun6REFktn/
Kz7DUt9jaR1UJyRzHxlTfCfAIgSxPdGqB/oF1suMyWgu5i0625vavLB5z5kC8Hq3qZJ9zJO1poE1kyD
+htiTtPWJ88e12xuH2XB/CZN9OpEiF98hAagiOE0EnOS5Q== Generated by Nova\n"
    }
}
```

Response Parameters

Parameter	Type	Description
keypair	Object	Specifies the SSH key pair. For details, see **Table 5-90**.

Table 5-90 keypair field data structure description

Parameter	Type	Description
fingerprint	String	Specifies fingerprint information about the key pair.
name	String	Specifies the key pair name.
public_key	String	Specifies the public key.
private_key	String	Specifies the private key. • The information about the private key is contained in the response for creating an SSH key. • The information about the private key is not contained in the response for importing an SSH key.
user_id	String	Specifies the ID of the user to which the key pair belongs.

Example Response

```
{
    "keypair": {
        "public_key": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgQC9mC3WZN9UGLxgPBpP7H5jZMc6pKwOoSgre8yun6REFktn/
Kz7DUt9jaR1UJyRzHxlTfCfAIgSxPdGqB/oF1suMyWgu5i0625vavLB5z5kC8Hq3qZJ9zJO1poE1kyD
+htiTtPWJ88e12xuH2XB/CZN9OpEiF98hAagiOE0EnOS5Q== Generated by Nova\n",
        "user_id": "f882feb345064e7d9392440a0f397c25",
        "name": "keypair-7d7c3650-dabe-4eb0-b904-5c464453c043",
        "fingerprint": "35:9d:d0:c3:4a:80:d3:d8:86:f1:ca:f7:df:c4:f9:d8"
```

```
  }
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.10.4 Deleting an SSH Key Pair (Native OpenStack API)

Function

This interface is used to delete a specified SSH key pair based on the key pair name.

URI

DELETE /v2.1/{project_id}/os-keypairs/{keypair_name}

Table 5-91 lists the parameters.

Table 5-91 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
keypair_name	Yes	Specifies the key pair name. You can obtain the key pair name by calling the **Querying SSH Key Pairs (Native OpenStack API)** API.

Request Parameters

None

Example Request

Deleting an SSH key pair (name: KeyPair-123)

```
DELETE https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/os-keypairs/keypair-test
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11 BMS 1D Tag Management

5.11.1 Querying BMS Tags (Native OpenStack API)

Function

This API is used to query all tags of a BMS.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

URI

GET /v2.1/{project_id}/servers/{server_id}/tags

Table 5-92 lists the parameters.

Table 5-92 Parameter description

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

Parameter	Mandatory	Description
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Querying tags of a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/53206ed0-56de-4d6b-b7ee-ffc62ca26f43/tags
```

Response Parameters

Parameter	Type	Description
tags	Array of strings	Specifies user-defined tags of a BMS.

Example Response

```
{
   "tags": [
      "baz",
      "foo",
      "qux"
   ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11.2 Adding Tags for a BMS (Native OpenStack API)

Function

This API is used to add tags for a BMS.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

Constraints

A BMS can have a maximum of 50 tags.

NOTE

- It is recommended that you add the **__type_baremetal** tag to BMSs to distinguish BMSs from ECSs. Otherwise, BMSs will be available only on the ECS console.
- A new tag will overwrite the existing one. If you want to retain the original tag, add it to the list of new tags. You are advised to add **__type_baremetal** to the added tags list each time you add a tag.

URI

PUT /v2.1/{project_id}/servers/{server_id}/tags

Table 5-93 lists the parameters.

Table 5-93 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

Parameter	Mandatory	Type	Description
tags	Yes	Array of strings	• Specifies the tags to be added. Each tag can contain a maximum of 80 characters. • The tag cannot start with a period (.). • A BMS can have a maximum of 50 tags. • An empty tag cannot be created.

Example Request

Adding tags **baz**, **foo**, and **qux** for a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
PUT https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/53206ed0-56de-4d6b-b7ee-ffc62ca26f43/tags
{
    "tags": [
        "baz",
        "foo",
        "qux"
    ]
}
```

Response Parameters

Parameter	Type	Description
tags	Array of strings	Specifies user-defined tags of a BMS.

Example Response

```
{
    "tags": [
        "baz",
        "foo",
        "qux"
    ]
}
```

Returned Values

Normal values

Returned Values	Description
200	The request has been successfully processed.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11.3 Deleting Tags of a BMS (Native OpenStack API)

Function

This API is used to delete all tags of a BMS.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

Constraints

NOTE

- Tag **__type_baremetal** is used to identify a BMS. You are not advised to delete this tag. Otherwise, the BMS will be displayed only on the ECS console.
- After deleting the **__type_baremetal** tag, you can add it again by following the instructions in **Adding a Tag for a BMS (Native OpenStack API)**. After the tag is added, the BMS will be displayed on the BMS console.

URI

DELETE /v2.1/{project_id}/servers/{server_id}/tags

Table 5-94 lists the parameters.

Table 5-94 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.

Request Parameters

None

Example Request

Deleting tags of a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
DELETE https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/53206ed0-56de-4d6b-b7ee-ffc62ca26f43/tags
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11.4 Adding a Tag for a BMS (Native OpenStack API)

Function

This API is used to add a tag for a BMS.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

Constraints

- A BMS can have a maximum of 50 tags.
- The tag contains a maximum of 80 characters.
- The tag cannot start with a period (.).
- An empty tag cannot be created.

NOTE

It is recommended that you add the **__type_baremetal** tag to BMSs to distinguish BMSs from ECSs.

URI

PUT /v2.1/{project_id}/servers/{server_id}/tags/{tag}

Table 5-95 lists the parameters.

Table 5-95 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
tag	Yes	Specifies the tag information. Constraints: • The tag contains a maximum of 80 characters. • The tag cannot start with a period (.). • An empty tag cannot be created. • URL encoding is required for special characters.

Request Parameters

None

Example Request

Adding a tag for a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
PUT https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/53206ed0-56de-4d6b-b7ee-ffc62ca26f43/tags/{tag}
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11.5 Checking for Tags of a BMS (Native OpenStack API)

Function

This API is used to check whether a BMS has a specified tag.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

URI

GET /v2.1/{project_id}/servers/{server_id}/tags/{tag}

Table 5-96 lists the parameters.

Table 5-96 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
tag	Yes	Specifies the key of the tag to be queried. Constraints: • URL encoding is required for special characters. • If no tag key is specified, all tags of the BMS are displayed.

Request Parameters

None

Example Request

Checking whether a BMS (ID: 2d85af7c-cbfe-40c5-a378-4d03b42fb0e2) has a specified tag

```
GET https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/2d85af7c-cbfe-40c5-a378-4d03b42fb0e2/tags/{tag}
```

Example Response

If the specified tag exists, no response is returned.

If the specified tag does not exist, the response is as follows:

```
{
   "itemNotFound": {
      "message": "Server 2d85af7c-cbfe-40c5-a378-4d03b42fb0e2 has no tag 'abc'",
      "code": 404
   }
}
```

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

5.11.6 Deleting a Tag of a BMS (Native OpenStack API)

Function

This API is used to delete a tag of a BMS.

You are required to use the HTTP header **X-OpenStack-Nova-API-Version: 2.26** to specify the micro version on the client.

Constraints

- The tag contains a maximum of 80 characters.
- If a tag contains non-URL-safe characters, perform URL encoding.

NOTE

- Tag **__type_baremetal** is used to identify a BMS. You are not advised to delete this tag. Otherwise, the BMS will be displayed only on the ECS console.
- After deleting the **__type_baremetal** tag, you can add it again by following the instructions in **Adding a Tag for a BMS (Native OpenStack API)**. After the tag is added, the BMS will be displayed on the BMS console.

URI

DELETE /v2.1/{project_id}/servers/{server_id}/tags/{tag}

Table 5-97 lists the parameters.

Table 5-97 Parameter description

Parameter	Mandatory	Description
project_id	Yes	Specifies the project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
server_id	Yes	Specifies the BMS ID. You can obtain the BMS ID from the BMS console or by calling the **Querying BMSs (Native OpenStack API)**.
tag	Yes	Specifies the tag information. Constraints: • A tag can contain a maximum of 80 characters. If a tag contains non-URL-safe characters, perform URL encoding. • If no key is specified, all tags of the BMS are deleted.

Request Parameters

None

Example Request

Deleting a tag from a BMS (ID: 53206ed0-56de-4d6b-b7ee-ffc62ca26f43)

```
DELETE https://{ECS Endpoint}/v2.1/bbf1946d374b44a0a2a95533562ba954/servers/53206ed0-56de-4d6b-b7ee-ffc62ca26f43/tags/{tag}
```

Response

N/A

Returned Values

Normal values

Returned Values	Description
204	The server has processed the request but did not return any content.

For details about other returned values, see **Status Codes**.

Error Codes

See **Error Codes**.

6 Examples

6.1 Example 1: Creating a BMS

Scenarios

BMSs are recommended for key applications and services that require high performance (such as Big Data clusters and enterprise middleware systems) and a secure and reliable running environment.

This section describes how to create a BMS using the **Creating BMSs** API. For details about how to call the API, see **Calling APIs**.

NOTE

Currently, only yearly/monthly BMSs can be created.

Prerequisites

- The BMS boot device can be a local disk or EVS disk. Quickly provisioned BMSs boot from an EVS disk. You can use either of the following methods to check the boot device:
  - After selecting a flavor on the **BMS console**, if **System Disk** is displayed in the **Disk** option, the flavor boots from an EVS disk. Otherwise, the flavor boots from a local disk.
  - Call the **Querying Parameter extra_specs of a BMS Flavor** API and check the value of **baremetal:extBootType** in the response. If the value is **LocalDisk**, the boot device is a local disk. If the value is **Volume**, the boot device is an EVS disk.

Create a BMS with Local Disks

The following example uses the most basic configuration required for creating a BMS with local disks.

```
{
    "server": {
        "availability_zone": "cn-north-4a",
        "name": "bms-test01",
```



```
    "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
    "flavorRef": "physical.s3.large",
    "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
    "nics": [
      {
        "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616"
      }
    ],
    "extendparam": {
      "chargingMode": "prePaid",
      "periodType": "month",
      "periodNum": 1,
      "isAutoRenew": "true",
      "isAutoPay": "true",
      "regionID": "cn-north-4"
    },
    "metadata": {
      "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81"
    },
    "key_name": "$key_name"
  }
}
```

- **availability_zone**: AZ where you want to create the BMS.
- **name**: name of the BMS, for example, **bms-test01**
- **imageRef**: ID of the image (OS) used to create the BMS You can query the image ID on the IMS console.
- **flavorRef**: BMS flavor ID in physical.*x.x* format You can obtain the flavor ID from the BMS console and determine the boot device of the flavor based on **Prerequisites**. In this example, a flavor that boots from a local disk is used, for example, **physical.s3.large**.
- **vpcid**: ID of the VPC where the BMS resides. You can query the ID on the VPC console or using the **Querying VPCs** API.
- **nics.subnet_id**: ID of the VPC subnet. You can query the ID on the VPC console or using the **Querying Subnets** API.
- **extendparam**: **chargingMode** specifies the billing mode, and **prePaid** indicates prepayment, that is, the yearly/monthly mode. **periodType** specifies the period type, and **month** indicates the monthly mode. **periodNum** specifies the number of periods, and **1** in this example indicates one month. **isAutoRenew** specifies whether to enable auto-renewal. **isAutoPay** specifies whether to automatically deduct the charges from your account balance without requiring manual payment. **regionID** specifies the ID of the region where the BMS resides.
- **metadata.op_svc_userid**: user ID, which you can obtain from **My Credentials** on the management console.
- **key_name**: key pair name. To log in to the BMS using an SSH key pair, set the value to the name of an existing key pair. You can obtain the key pair name from the **Key Pair** page on the Cloud Server Console.

Create a BMS Supporting Quick Provisioning

The following example uses the most basic configuration required for creating a BMS supporting quick provisioning.

```
{
  "server": {
    "availability_zone": "cn-north-4a",
    "name": "bms-test02",
```



```
    "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
    "flavorRef": "physical.h2.large",
    "root_volume": {
          "volumetype": "SAS",
          "size": 150
    },
    "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
    "nics": [
      {
        "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616"
      }
    ],
    "extendparam": {
      "chargingMode": "prePaid",
      "periodType": "month",
      "periodNum": 1,
      "isAutoRenew": "true",
      "isAutoPay": "true",
      "regionID": "cn-north-4"
    },
    "metadata": {
      "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81"
    },
    "key_name": "$key_name"
  }
}
```

Compared with creating a BMS with local disks, you need to select a flavor whose boot device is EVS disk and set system disk parameters.

- **volumetype**: type of the system disk. **SAS** indicates high I/O. You can specify other types. For details, see the parameter descriptions in **Creating BMSs**.
- **size**: system disk size (GB). The value ranges from **40** to **1024**. The system disk size cannot be less than the minimum size of the system disk (**min_disk**) in the image. You can obtain the value of **min_disk** using the **Querying Images** API.

Create a BMS with Data Disks

You can configure data disks for BMSs by adding the **data_volumes** field to the request body. An example is as follows:

```
{
  "server": {
    "availability_zone": "cn-north-4a",
    "name": "bms-test03",
    "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
    "flavorRef": "physical.s3.large",
    "data_volumes": [
      {
        "volumetype": "SAS",
        "size": 10,
        "shareable": false
      }
    ],
    "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
    "nics": [
      {
        "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616"
      }
    ],
    "extendparam": {
      "chargingMode": "prePaid",
      "periodType": "month",
      "periodNum": 1,
      "isAutoRenew": "true",
```



```
            "isAutoPay": "true",
            "regionID": "cn-north-4"
        },
        "metadata": {
            "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81"
        },
        "key_name": "$key_name"
    }
}
```

- **volumetype**: data disk type. **SAS** indicates high I/O.
- **size**: data disk size (GB). The value ranges from **10** to **32768**.
- **shareable**: whether the disk is shareable. **true** indicates a shared disk and **false** indicates a common disk.

Create a BMS with an EIP

You can bind an EIP to the BMS by adding the **publicip** field to the request body. During the BMS creation, an EIP is assigned and bound to the BMS. An example is as follows:

```
{
    "server": {
        "availability_zone": "cn-north-4a",
        "name": "bms-test04",
        "imageRef": "766a545a-02e1-433e-b1d1-733b5dc95e94",
        "flavorRef": "physical.s3.large",
        "data_volumes": [
            {
                "volumetype": "SAS",
                "size": 10,
                "shareable": false
            }
        ],
        "vpcid": "8df83bf2-fd2e-4ee0-9692-c0b7736513fb",
        "nics": [
            {
                "subnet_id": "c6bb8788-4fd1-4a8f-adab-7eba3bed8616"
            }
        ],
        "extendparam": {
            "chargingMode": "prePaid",
            "periodType": "month",
            "periodNum": 1,
            "isAutoRenew": "true",
            "isAutoPay": "true",
            "regionID": "cn-north-4"
        },
        "metadata": {
            "op_svc_userid": "59781460e9e54886a7d03df7d3f3fc81"
        },
        "publicip": {
            "eip": {
                "iptype": "5_bgp",
                "bandwidth": {
                    "chargemode": "bandwidth",
                    "name": "bms-test04-bandwidth",
                    "size": 1,
                    "sharetype": "PER"
                },
                "extendparam": [
                    "chargingMode": "prePaid"
                ]
            }
        },
        "key_name": "$key_name"
```



```
  }
}
```

- **iptype**: EIP type. **5_bgp** indicates dynamic BGP.
- **chargemode**: bandwidth billing type. The value can be **traffic** or **bandwidth**.
- **name**: bandwidth name
- **size**: bandwidth (Mbit/s). The value ranges from 1 to 2000.
- **sharetype**: bandwidth sharing type. **PER** indicates dedicated bandwidth.
- **extendparam.chargingMode**: billing mode of the EIP. **prePaid** indicates prepayment, that is, the yearly/monthly mode.

6.2 Example 2: Querying Details About BMSs

Scenarios

You can specify criteria to query details about desired BMSs.

Involved APIs

The following API is involved:

Querying Details About BMSs

Query Details About BMSs By Specifying a Flavor

- API

  URI format: GET /v1/{project_id}/baremetalservers/detail

  For details, see **Querying Details About BMSs**.

- Example request

  GET https://{BMS Endpoint}/v1/5233f5df59f9491ba7c14d8b5d535f6b/baremetalservers/detail?flavor=physical.comtest04.large

- Example response

```
{
   "count": 1,
   "servers": [
      {
         "fault": null,
         "id": "cb52b43d-8ba7-4b4d-84c4-7ef332715073",
         "name": "kd1ne4xl-c00420527-nodel-nonstop",
         "addresses": {
            "fdbf76aa-697a-4f49-8835-36c0e6903e28": [
               {
                  "version": "4",
                  "addr": "172.16.0.178",
                  "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:8a:3a:50",
                  "OS-EXT-IPS:type": "fixed",
                  "OS-EXT-IPS:port_id": "0c111191-9785-49e6-8b42-66ef2802bd8f"
               },
               {
                  "version": "4",
                  "addr": "124.71.38.23",
                  "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:8a:3a:50",
                  "OS-EXT-IPS:type": "floating",
                  "OS-EXT-IPS:port_id": "0c111191-9785-49e6-8b42-66ef2802bd8f"
               }
            ]
```

```
        },
        "flavor": {
            "disk": "192010",
            "vcpus": "128",
            "ram": "524288",
            "id": "physical.comtest04.large",
            "name": "physical.comtest04.large"
        },
        "accessIPv4": "",
        "accessIPv6": "",
        "status": "SHUTOFF",
        "progress": null,
        "hostId": "f57b3421b5cdc214fabf127984ef779b7433acc1d6379214ed834219",
        "updated": "2020-11-07T09:28:40Z",
        "created": "2020-09-21T09:07:28Z",
        "metadata": {
            "metering.order_id": "CS20110717284WW42",
            "metering.image_id": "c13c91f7-038a-443e-ad76-608e255b4d85",
            "metering.imagetype": "shared",
            "metering.resourcespeccode": "physical.comtest04.large.linux",
            "image_name": "euler2u8-cbg-sdi3-20200330v3",
            "op_svc_userid": "48b5cec3f26e4bedab54cb996b124079",
            "__bms_support_evs": "false",
            "EcmResStatus": "",
            "metering.product_id": "00301-103881-0--0",
            "metering.resourcetype": "__type_baremetal",
            "vpc_id": "fdbf76aa-697a-4f49-8835-36c0e6903e28",
            "os_type": "Linux",
            "os_bit": "64",
            "chargingMode": "1"
        },
        "tags": [
            "__type_baremetal"
        ],
        "description": "kd1ne4xl-c00420527-nodel-nonstop",
        "locked": false,
        "config_drive": "",
        "tenant_id": "5233f5df59f9491ba7c14d8b5d535f6b",
        "user_id": "fc3a5ac3fe244ca6bb079c8069f9143f",
        "key_name": null,
        "os-extended-volumes:volumes_attached": [
            {
                "device": "/dev/sda",
                "bootIndex": "0",
                "id": "405ae92b-2d59-4d61-8eb0-79946e391c33",
                "delete_on_termination": "false"
            }
        ],
        "OS-EXT-STS:task_state": null,
        "OS-EXT-STS:power_state": 4,
        "OS-EXT-STS:vm_state": "stopped",
        "OS-EXT-SRV-ATTR:host": "f57b3421b5cdc214fabf127984ef779b7433acc1d6379214ed834219",
        "OS-EXT-SRV-ATTR:instance_name": "instance-00567827",
        "OS-EXT-SRV-ATTR:hypervisor_hostname":
"bea2f2c61b2def48c3ad001446ffada47bbd2b1cbf75d4950c9609a0",
        "OS-DCF:diskConfig": "MANUAL",
        "OS-EXT-AZ:availability_zone": "cn-south-1e",
        "os:scheduler_hints": {
            "dec_baremetal": [
                "share"
            ]
        },
        "OS-EXT-SRV-ATTR:root_device_name": "/dev/vda",
        "OS-EXT-SRV-ATTR:ramdisk_id": "",
        "enterprise_project_id": "0",
        "OS-EXT-SRV-ATTR:user_data": "$USER_DATA",
        "OS-SRV-USG:launched_at": "2020-09-21T09:14:01.000000",
        "OS-EXT-SRV-ATTR:kernel_id": "",
        "OS-EXT-SRV-ATTR:launch_index": 0,
```



```
            "host_status": "UP",
            "OS-EXT-SRV-ATTR:reservation_id": "r-18mfg3us",
            "OS-EXT-SRV-ATTR:hostname": "kd1ne4xl-c00420527-nodel-nonstop",
            "OS-SRV-USG:terminated_at": null,
            "sys_tags": [
                {
                    "key": "_sys_enterprise_project_id",
                    "value": "0"
                }
            ],
            "security_groups": [
                {
                    "id": "2beb2b46-2a55-4b83-b362-20698eeda10b",
                    "name": "d310-l00415626-cce-control-8xsdf"
                }
            ],
            "image": {
                "id": "c13c91f7-038a-443e-ad76-608e255b4d85"
            },
            "hypervisor": null
        }
    ]
}
```


Query Details About BMSs By Specifying an Enterprise Project ID

- API

  URI format: GET /v1/{project_id}/baremetalservers/detail

  For details, see **Querying Details About BMSs**.

- Example request

  GET https://{BMS Endpoint}/v1/5233f5df59f9491ba7c14d8b5d535f6b/baremetalservers/detail?enterprise_project_id=98135c62-f0c1-4003-9c9f-7f5523d7f3a2

- Example response


```
{
    "count": 1,
    "servers": [
        {
            "fault": null,
            "id": "cb52b43d-8ba7-4b4d-84c4-7ef332715073",
            "name": "kd1ne4xl-c00420527-nodel-nonstop",
            "addresses": {
                "fdbf76aa-697a-4f49-8835-36c0e6903e28": [
                    {
                        "version": "4",
                        "addr": "172.16.0.178",
                        "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:8a:3a:50",
                        "OS-EXT-IPS:type": "fixed",
                        "OS-EXT-IPS:port_id": "0c111191-9785-49e6-8b42-66ef2802bd8f"
                    },
                    {
                        "version": "4",
                        "addr": "124.71.38.23",
                        "OS-EXT-IPS-MAC:mac_addr": "fa:16:3e:8a:3a:50",
                        "OS-EXT-IPS:type": "floating",
                        "OS-EXT-IPS:port_id": "0c111191-9785-49e6-8b42-66ef2802bd8f"
                    }
                ]
            },
            "flavor": {
                "disk": "192010",
                "vcpus": "128",
                "ram": "524288",
                "id": "physical.comtest04.large",
```



```
            "name": "physical.comtest04.large"
        },
        "accessIPv4": "",
        "accessIPv6": "",
        "status": "SHUTOFF",
        "progress": null,
        "hostId": "f57b3421b5cdc214fabf127984ef779b7433acc1d6379214ed834219",
        "updated": "2020-11-07T09:28:40Z",
        "created": "2020-09-21T09:07:28Z",
        "metadata": {
            "metering.order_id": "CS20110717284WW42",
            "metering.image_id": "c13c91f7-038a-443e-ad76-608e255b4d85",
            "metering.imagetype": "shared",
            "metering.resourcespeccode": "physical.comtest04.large.linux",
            "image_name": "euler2u8-cbg-sdi3-20200330v3",
            "op_svc_userid": "48b5cec3f26e4bedab54cb996b124079",
            "__bms_support_evs": "false",
            "EcmResStatus": "",
            "metering.product_id": "00301-103881-0--0",
            "metering.resourcetype": "__type_baremetal",
            "vpc_id": "fdbf76aa-697a-4f49-8835-36c0e6903e28",
            "os_type": "Linux",
            "os_bit": "64",
            "chargingMode": "1"
        },
        "tags": [
            "__type_baremetal"
        ],
        "description": "kd1ne4xl-c00420527-nodel-nonstop",
        "locked": false,
        "config_drive": "",
        "tenant_id": "5233f5df59f9491ba7c14d8b5d535f6b",
        "user_id": "fc3a5ac3fe244ca6bb079c8069f9143f",
        "key_name": null,
        "os-extended-volumes:volumes_attached": [
            {
                "device": "/dev/sda",
                "bootIndex": "0",
                "id": "405ae92b-2d59-4d61-8eb0-79946e391c33",
                "delete_on_termination": "false"
            }
        ],
        "OS-EXT-STS:task_state": null,
        "OS-EXT-STS:power_state": 4,
        "OS-EXT-STS:vm_state": "stopped",
        "OS-EXT-SRV-ATTR:host": "f57b3421b5cdc214fabf127984ef779b7433acc1d6379214ed834219",
        "OS-EXT-SRV-ATTR:instance_name": "instance-00567827",
        "OS-EXT-SRV-ATTR:hypervisor_hostname":
"bea2f2c61b2def48c3ad001446ffada47bbd2b1cbf75d4950c9609a0",
        "OS-DCF:diskConfig": "MANUAL",
        "OS-EXT-AZ:availability_zone": "cn-south-1e",
        "os:scheduler_hints": {
            "dec_baremetal": [
                "share"
            ]
        },
        "OS-EXT-SRV-ATTR:root_device_name": "/dev/vda",
        "OS-EXT-SRV-ATTR:ramdisk_id": "",
        "enterprise_project_id": "98135c62-f0c1-4003-9c9f-7f5523d7f3a2",
        "OS-EXT-SRV-ATTR:user_data": "$USER_DATA",
        "OS-SRV-USG:launched_at": "2020-09-21T09:14:01.000000",
        "OS-EXT-SRV-ATTR:kernel_id": "",
        "OS-EXT-SRV-ATTR:launch_index": 0,
        "host_status": "UP",
        "OS-EXT-SRV-ATTR:reservation_id": "r-18mfg3us",
        "OS-EXT-SRV-ATTR:hostname": "kd1ne4xl-c00420527-nodel-nonstop",
        "OS-SRV-USG:terminated_at": null,
        "sys_tags": [
            {
```



```
                "key": "_sys_enterprise_project_id",
                "value": "0"
            }
        ],
        "security_groups": [
            {
                "id": "2beb2b46-2a55-4b83-b362-20698eeda10b",
                "name": "d310-l00415626-cce-control-8xsdf"
            }
        ],
        "image": {
            "id": "c13c91f7-038a-443e-ad76-608e255b4d85"
        },
        "hypervisor": null
    }
  ]
}
```


6.3 Example 3: Attaching Disks to a BMS

Scenarios

If the existing disks of a BMS fail to meet service requirements, for example, due to insufficient disk space or poor disk performance, you can attach more available disks to the BMS, or call the EVS disk creation API to create disks and attach them to the BMS.

Prerequisites

BMSs using certain flavors cannot have EVS disks attached because the servers do not have SDI iNICs or for other reasons. Therefore, perform the following operations to check whether EVS disks can be attached to a BMS:

Call the **Querying Flavor Details and Extended Flavor Information** API and check the response. If the value of **baremetal:__support_evs** for the BMS flavor is **true**, EVS disks can be attached to the BMS. If the value is **false** or the BMS flavor does not contain this parameter, EVS disks cannot be attached to the BMS.

Involved APIs

The following APIs are involved:

- **Creating EVS Disks**
- **Attaching an EVS Disk to a BMS**
- **Querying EVS disks attached to a BMS**

Procedure

1. Create an EVS disk.
   - API

     URI format: POST /v2/{project_id}/volumes

     For details, see **Creating EVS Disks**.
   - Example request

     POST https://{EVS Endpoint}/v2/05041fffa40025702f6dc009cc6f8f33/volumes

Body:

```
{
    "volume": {
        "name": "openapi_vol01",
        "availability_zone": "cn-north-4a",
        "description": "create for api test",
        "volume_type": "SSD",
        "metadata": {
            "hw:passthrough": "true"
        },
        "size": 40
    }
}
```

NOTE

BMSs support only SCSI EVS disks. Therefore, the value of **hw:passthrough** must be **true**.

- Example response

```
{
    "volume": {
        "attachments": [],
        "links": [
            {
                "href": "https://evs.cn-
north-4.myhuaweicloud.com/v2/05041fffa40025702f6dc009cc6f8f33/volumes/
8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111",
                "rel": "self"
            },
            {
                "href": "https://evs.cn-north-4.myhuaweicloud.com/
05041fffa40025702f6dc009cc6f8f33/volumes/8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "cn-north-4a",
        "encrypted": false,
        "updated_at": "2020-11-09T08:19:22.077967",
        "replication_status": "disabled",
        "storage_cluster_id": null,
        "snapshot_id": null,
        "id": "8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111",
        "size": 40,
        "user_id": "0504186e6a8010e01f3ec009a7279baa",
        "metadata": {
            "hw:passthrough": "true"
        },
        "status": "creating",
        "description": "create for api test",
        "multiattach": false,
        "source_volid": null,
        "consistencygroup_id": null,
        "name": "openapi_vol01",
        "bootable": "false",
        "created_at": "2020-11-09T08:19:22.061273",
        "volume_type": "SSD",
        "shareable": false
    }
}
```

Record the value of **id** in **volume**.

2. Attach the EVS disk to the BMS.
   - API

     URI format: POST /v1/{project_id}/baremetalservers/{server_id}/attachvolume

     For details, see **Attaching an EVS Disk to a BMS**.

- Example request

  POST https://{BMS Endpoint}/v1/05041fffa40025702f6dc009cc6f8f33/baremetalservers/9ab74d89-61e7-4259-8546-465fdebe4944/attachvolume

  Body:

```
{
   "volumeAttachment": {
      "volumeId": "8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111",
      "device": "/dev/sdb"
   }
}
```

- Example response

```
{
   "volumeAttachment": {
      "id": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
      "volumeId": "8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111",
      "serverId": "9ab74d89-61e7-4259-8546-465fdebe4944",
      "device": "/dev/sdb"
   }
}
```

3. Check whether the EVS disk is successfully attached to the BMS.

   - API

     URI format: GET /v1/{project_id}/baremetalservers/{server_id}/os-volume_attachments

     For details, see **Querying EVS Disks Attached to a BMS**.

   - Example request

     GET https://{BMS Endpoint}/v1/05041fffa40025702f6dc009cc6f8f33/baremetalservers/9ab74d89-61e7-4259-8546-465fdebe4944/os-volume_attachments

   - Example response

```
{
   "volumeAttachments": [
      {
         "device": "/dev/sdd",
         "id": "a26887c6-c47b-4654-abb5-dfadf7d3f803",
         "serverId": "9ab74d89-61e7-4259-8546-465fdebe4944",
         "volumeId": "a26887c6-c47b-4654-abb5-dfadf7d3f803"
      },
      {
         "device": "/dev/sdb",
         "id": "b53f23bd-ee8f-49ec-9420-d1acfeaf91d6",
         "serverId": "9ab74d89-61e7-4259-8546-465fdebe4944",
         "volumeId": "8ae4ff5b-7e6e-492f-b83a-8a51fdfa3111"
      }
   ]
}
```


7 Public Parameters

7.1 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.

Returned Value	Description
408 Request Timeout	The request timed out.
409 Conflict	Failed to complete the request due to a conflict.
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.

7.2 Error Codes

Context

- An error code returned by an API does not correspond to one error message. The following table lists only common error messages.
- Most BMS APIs are asynchronous. Some error codes are displayed in the returned messages for task viewing requests. HTTP status codes may not be accurate.
- The BMS service is strongly dependent on other services, such as network and storage. When error messages are provided for the BMS-depended services, contact technical support for troubleshooting.

Error Code Description

If an error code starting with **APIGW** is returned after you call an API, rectify the fault by referring to the instructions provided in **API Gateway Error Codes**.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0001	Request error.	Fail to parse request, reason: %s.	Check the request body according to the returned error message.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0002	BMS has not been launched.	The BMS service is unavailable.	Select a region where the BMS service is available.
400	BMS.0003	Request error.	Create BareMetal Server error, request is null.	Check the request body according to the returned error message.
400	BMS.0004	Insufficient permission.	Role check fail, reason: %s.	See the returned error message or contact technical support.
400	BMS.0005	Insufficient permission.	Role check fail, reason: You do not have permission or your balance is insufficient.	See the returned error message or contact technical support.
400	BMS.0008	Failed to query a flavor.	Fail to query flavor [%s], reason: %s.	See the returned error message or contact technical support.
400	BMS.0006	Failed to create the task.	Fail to operate baremetal server.	See the returned error message or contact technical support.
400	BMS.0009	Failed to query flavor attributes.	Fail to query flavor extra specs[%s], reason: %s.	See the returned error message or contact technical support.
400	BMS.0010	Failed to query an image.	Fail to query image [%s], reason: %s.	See the returned error message or contact technical support.
400	BMS.0011	The images do not support BMS.	The selected images cannot be used to apply for BMSs, %s.	See the returned error message or contact technical support.
400	BMS.0012	Insufficient IP addresses in the selected subnet.	The number of IP addresses in the selected subnet[%s] is insufficient.	Check whether the IP addresses of the subnet are used up.
400	BMS.0013	Failed to query the port.	Fail to query ports by subnet [%s], reason: %s.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0014	IP address conflict.	The specified IP address conflicts with an existing IP address in subnet[%s].	Modify the NIC IP address.
400	BMS.0015	Failed to query the NIC.	Fail to query subnet, reason: %s.	See the returned error message or contact technical support.
400	BMS.0017	The EIP quota is insufficient.	The number[%d] of EIPs has reached the maximum[%d] allowed. Apply for a higher quota and try again.	Apply for a higher EIP quota.
400	BMS.0218	Failed to create the order.	Fail to create order, reason: %s.	See the returned error message or contact technical support.
400	BMS.0219	Failed to change the BMS name.	OpenStack nova change ServerName fail.	Contact technical support.
400	BMS.0018	Invalid request parameters.	Request parameter is invalid.	Modify the request parameters based on the returned error message.
400	BMS.0019	Invalid NIC parameters.	publicIp parameter is illegal, reason: %s.	See the returned error message or contact technical support.
400	BMS.0020	Failed to check the billing mode.	Fail to check chargingMode, reason: %s.	See the returned error message or contact technical support.
400	BMS.0021	The flavor is invalid for creating the BMS. Select another flavor.	Flavor parameter is illegal, reason: %s.	Select a valid flavor.
400	BMS.0022	Invalid request parameters.	Request parameter is invalid.	Modify the request parameters based on the returned error message.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0023	Failed to query the quota.	Fail to query limits, reason: %s	See the returned error message or contact technical support.
400	BMS.0025	The number of BMSs exceeds the quota.	The number of cloud servers has reached the maximum allowed.	Apply for a higher quota.
400	BMS.0026	The number of BMS CPUs exceeds the quota.	The number of CPU cores used by all cloud servers has reached the maximum allowed.	Apply for a higher quota.
400	BMS.0027	The BMS memory exceeds the quota.	The memory space used by all cloud servers has reached the maximum allowed.	Apply for a higher quota.
400	BMS.0028	The number of BMSs and that of CPUs exceed the quotas.	The number of CPU cores used by all cloud servers and that of cloud servers have reached the maximum allowed.	Apply for a higher quota.
400	BMS.0029	The number of BMSs and the memory exceed the quotas.	The memory space used by all cloud servers and the number of cloud servers have reached the maximum allowed.	Apply for a higher quota.
400	BMS.0030	The BMS memory and the number of CPUs exceed the quotas.	The memory space and number of CPU cores used by all cloud servers have reached the maximum allowed.	Apply for a higher quota.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0031	The number of BMSs, the number of CPUs, and the memory exceed the quotas.	The number of cloud servers, the memory space used by all cloud servers, and the number of CPU cores used by all cloud servers have reached the maximum allowed.	Apply for a higher quota.
400	BMS.0032	Contact technical support to apply for a BMS.	Token check fail.	See the returned error message or contact technical support.
403	BMS.0033	You do not have operation rights. Contact technical support.	Fail to check roles, reason: %s.	See the returned error message or contact technical support.
400	BMS.0034	Currently, BMSs cannot be automatically provisioned.	Not support create Bare Metal Server.	This operation is not supported.
400	BMS.0047	Invalid system disk.	Root volume is illegal, reason: %s.	See the returned error message or contact technical support.
400	BMS.0049	Failed to query **key_name**.	Query keypair fail, reason is: %s.	See the returned error message or contact technical support.
400	BMS.0089	Failed to pass the verification of tag policy compliance.	The tag policy does not allow the specified value for the following tag key: %s.	Change the tag values according to the policy or contact technical support.
400	BMS.0102	This image does not support volume attaching.	The image does not support attach volume.	Contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0103	The disk does not exist.	Attach volume %s fail, volume info is null.	Check whether the disk information is correct.
400	BMS.0104	The data volume cannot be attached to the mount point of the system volume.	Attach volume %s fail, data volume can not attach in root volume device.	Select another mount point.
400	BMS.0105	The mount point is invalid.	Attach volume %s fail, device is illegal.	Check whether the mount point is valid.
400	BMS.0106	The AZ of data disks is different from that of the BMS.	Attach volume %s fail, volume's az is not equal with server's az.	Ensure that the AZ of data disks is the same as that of the BMS.
400	BMS.0108	The system disk does not match the BMS.	Attach volume %s fail, root volume does not match the VM, do not change image.	Attach a matching system disk to the BMS.
400	BMS.1001	This operation can be performed only when the BMS is in running or stopped.	Volume can only be attached when server %s stopped or active.	Try again later or contact technical support.
400	BMS.1002	The number of data disks that can be attached to the BMS exceeds the quota.	Attach volume fail, server %s attachment num over limit.	Contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.1003	The disk status is invalid.	Attach shareable volume %s fail, volume status is %s, not available or inuse.	Check whether the disk is in attached or other unavailable status.
400	BMS.1004	The number of shared data disks that can be attached to the BMS exceeds the quota.	Attach shareable volume %s fail, volume status is %s, not available or inuse.	Contact technical support.
400	BMS.1006	The running status of the BMS cannot be obtained.	Server %s info is null or its status or its metadata is null.	Try again later or contact technical support.
400	BMS.1007	Failed to call the API.	Calling interface failed	See the returned error message or contact technical support.
400	BMS.1008	The disk type does not match the BMS.	The server[%s] is not HANA server, volume[%s] does not match the server.	Check whether the disk type matches the BMS or contact the technical support.
400	BMS.1009	The data disk has been attached to the BMS.	Attach shareable volume %s fail, volume status is %s, the volume has been attached to the server.	Select another data disk.
400	BMS.1011	The data disk is being attached.	Attach shareable volume %s fail, volume status is %s, not available or inuse.	Do not repeatedly attach the data disk.
400	BMS.3001	Failed to query the security group or subnet.	Query security group failed: %s. or Query vpcId for subnet failed: %s.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.3011	System error.	Decoded token is null.	See the returned error message or contact technical support.
400	BMS.3025	The EVS disk type is incorrect.	Not support create shareable data volumes.	See the returned error message or contact technical support.
400	BMS.3035	No matching disk is found for the BMS.	Attach volume fail: the bmsid [%s] in volume metadata is not the same with bmsid [%s] in url.	Ensure that the disk matches the BMS or contact technical support.
400	BMS.3039	Failed to verify password complexity.	The password is illegal, reason: %s.	See the password rules.
400	BMS.0246	Failed to query the VPC.	The vpcId[%s] is invalid or not-existing.	See the returned error message or contact technical support.
400	BMS.0201	The parameters for creating the BMS are incorrectly configured.	Fail to check the baremetal server params, reason: %s.	See the returned error message or contact technical support.
400	BMS.0202	The data disk type is invalid.	All volumes must be in same type.	See the returned error message or contact technical support.
400	BMS.0203	Parameter **Volume** is invalid.	Volume is illegal, %s.	See the returned error message or contact technical support.
400	BMS.0204	The number of data disks exceeds the upper limit.	The number of data volumes is illegal. The number is %d, but maximum number allow is %d.	See the returned error message or contact technical support.
400	BMS.0205	The BMS quantity is invalid.	The number of baremetal servers is out of range for one quest.	Change or delete the BMS quantity.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0206	The name contains invalid characters.	The baremetal server name [%s] is include invalid char.	Change the name as prompted.
400	BMS.0207	No IP address can be specified when BMSs are created in a batch.	VPC is illegal, reason: %s.	See the returned error message or contact technical support.
400	BMS.0208	Failed to query the AZ.	Fail to get RegionInfo by tenant [%s], reason: %s.	See the returned error message or contact technical support.
400	BMS.0210	The length of the injected data exceeds the upper limit.	The size of userdata [%d] is over quota limits [%d].	Modify the injected data.
400	BMS.0211	Invalid key.	The image platform is [%s], support publicKey, reason: the publicKey is illegal, null is not allowed.	Select a valid key.
400	BMS.0114	The disk to be detached is not in the disk list of the BMS.	Volume %s is not in server %s attach volume list.	Check whether the disk exists.
400	BMS.0212	The system is overloaded.	System is overloaded, please try again later.	Try again later.
400	BMS.0213	Insufficient permission.	OBTAZ role verify fail: not allowed role.	See the returned error message or contact technical support.
400	BMS.0214	Failed to query the NIC.	Query subnet[%s] failed: response is null.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0215	Failed to create the order.	The response of inquiry order info is null or invalid.	See the returned error message or contact technical support.
400	BMS.0216	Failed to submit the order.	Submit order[%s] failed: %s.	See the returned error message or contact technical support.
400	BMS.0217	Failed to verify metadata.	Create server fail, reason: metaData is illegal.	See the returned error message or contact technical support.
400	BMS.0222	The primary NIC cannot be deleted.	primary port can not be deleted.	See the returned error message or contact technical support.
400	BMS.0223	Currently, only SCSI disks are supported.	Only SCSI disks are supported.	Select the SCSI disk type.
400	BMS.0039	The BMS has been shut down.	Server is stopped, not allow to stop.	Do not stop the BMS again.
400	BMS.0040	The BMS does not exist.	The server does not exist.	See the returned error message or contact technical support.
400	BMS.0035	The BMS is being powered on and no other operations are allowed.	Server is powering on, not allow to %s.	See the returned error message or contact technical support.
400	BMS.0036	The BMS is being powered off and no other operations are allowed.	Server is powering off, not allow to %s.	See the returned error message or contact technical support.
400	BMS.0038	The BMS is running and cannot be started.	Server is running, not allow to start.	Do not start the BMS again.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0037	The BMS is being restarted and no other operations are allowed.	Server is rebooting, not allow to %s.	See the returned error message or contact technical support.
403	BMS.0288	You do not have permissions to perform this operation.	Policy doesn't allow %s to be performed.	See the returned error message or contact technical support.
400	BMS.0290	The disk is not a shared disk and cannot be attached to multiple BMSs.	Batch attach volume type must be sharable.	Select a shared disk.
400	BMS.0291	The enterprise project ID cannot be empty.	enterprise_project_id can not be empty.	See the returned error message or contact technical support.
400	BMS.0294	Failed to invoke EPS to query the project.	Query enterprise_project_id[%s] failed %.	See the returned error message or contact technical support.
400	BMS.0293	Failed to query the project information.	Query enterprise_project_id[%s] failed: response is null.	See the returned error message or contact technical support.
400	BMS.0292	The enterprise project has been suspended.	enterprise_project_id[%s] is deactive.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0071	If an EIP has been specified for creating the BMS, no other EIP can be created for the BMS.	Using an existing EIP and creating a new EIP can't be used at the same time.	See the returned error message or contact technical support.
400	BMS.0072	An EIP cannot be used to create multiple BMSs.	An existing EIP cannot be assigned to the ECSs created in batches.	See the returned error message or contact technical support.
400	BMS.0073	The bandwidth parameter is empty.	Bandwidth info in eip is null.	See the returned error message or contact technical support.
400	BMS.0074	The EIP extension parameter is unavailable.	Parameter exetendparam or chargingMode is null.	See the returned error message or contact technical support.
400	BMS.0075	The bandwidth ID is empty.	Bandwidth info in eip is null.	See the returned error message or contact technical support.
400	BMS.0076	The EIP billing mode is unavailable.	Parameter chargingMode only supports postPaid when shared type is whole.	See the returned error message or contact technical support.
400	BMS.0077	Invalid bandwidth size.	PublicIp parameter is illegal, reason: %s.	See the returned error message or contact technical support.
400	BMS.0078	EIP quota is insufficient.	Shared bandwidth has been bound to %d EIPs, quota is %d.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0079	The EIP bandwidth type is unavailable.	Specifies the bandwidth of the shared type and the id cannot be empty.	See the returned error message or contact technical support.
400	BMS.0080	The subnet status is unavailable.	Check subnet status failed.	Check whether the subnet exists or whether it is in **ACTIVE** state.
400	BMS.0297	The submitted EVS disk order has not been paid. The EVS disk cannot be attached to or detached from the BMS.	Fail to attach volume[%s]: volume is locked.	See the returned error message or contact technical support.
400	BMS.0054	Failed to verify the capacity.	Check capacity fail, the number of capacity is: [%d], and req num is: [%d].	The capacity is insufficient. Contact technical support.
400	BMS.0055	Failed to query the quota.	Query capacity fail, Flavor id is [%s], reason: [%s].	See the returned error message or contact technical support.
400	BMS.2005	Failed to query the network.	Network could not be found.	Check the network information or contact technical support.
400	BMS.3037	Insufficient resources or failed to start the BMS.	Insufficient resources or failed to start the BMS.	Contact technical support.
400	BMS.3004	Failed to create the BMS due to an internal system error.	Failed to create the BMS due to an internal system error.	Contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.3005	Failed to create the port.	Failed to create the port.	Contact technical support.
400	BMS.3033	Failed to create the system disk.	Failed to create the system disk.	Contact technical support.
400	BMS.3029	Failed to create the system disk. The disk status is abnormal.	Failed to create the system disk. The disk status is abnormal.	Contact technical support.
400	BMS.3006	Failed to assign the EIP.	Failed to assign the floating IP address.	Contact technical support.
400	BMS.3021	Failed to create the data disk.	Failed to create the data disk.	Contact technical support.
400	BMS.3019	Failed to attach the data disk.	Failed to attach the data disk.	Contact technical support.
400	BMS.3038	Failed to assign an EIP.	Failed to assign the EIP.	Contact technical support.
400	BMS.0315	Disks cannot be attached to a BMS using this flavor.	Disks cannot be attached to a BMS using this flavor[%s].	See the returned error message or contact technical support.
400	BMS.0316	The RAID level does not match the RAID type.	RAID level and RAID type do not match.	See the returned error message or contact technical support.
400	BMS.0317	Disk information is empty.	Disk information is empty.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0320	This flavor does not support RAID formed by disks of this type.	This flavor does not support RAID formed by disks of this type.	See the returned error message or contact technical support.
400	BMS.0321	This flavor does not support RAID formed by disks of this size.	This flavor does not support RAID formed by disks of this size.	See the returned error message or contact technical support.
400	BMS.0322	Disk quantity and RAID level do not match.	Disk quantity and RAID level do not match.	See the returned error message or contact technical support.
400	BMS.0323	Disk quantity does not match that in the flavor.	Disk quantity does not match that in the flavor.	See the returned error message or contact technical support.
400	BMS.0325	Disk quantity exceeds that supported by the flavor.	Disk quantity exceeds that supported by the flavor.	See the returned error message or contact technical support.
400	BMS.0327	Only JBOD is supported.	Only JBOD is supported.	See the returned error message or contact technical support.
400	BMS.0328	Only single-disk RAID 0 is supported.	Only single-disk RAID 0 is supported.	See the returned error message or contact technical support.
400	BMS.0329	Disk type is empty.	Disk type is empty.	See the returned error message or contact technical support.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0330	RAID is not supported.	RAID is not supported.	See the returned error message or contact technical support.
400	BMS.0331	No IPv6 Open Beta Test permissions.	No IPv6 Open Beta Test permissions.	See the returned error message or contact technical support.
400	BMS.0333	IPv6 is not supported.	IPv6 is not supported.	See the returned error message or contact technical support.
400	BMS.0400	The image does not support IPv6.	the image[%s] is not support IPv6.	See the returned error message or contact technical support.
400	BMS.0360	The disk has been frozen and cannot be operated for the BMS.	This operation cannot be performed because EVS Disk %s is frozen.	See the returned error message or contact technical support.
400	BMS.3026	The password does not meet requirements.	Password does not meet the requirements of the rule.	See the returned error message or contact technical support.
400	BMS.3027	Invalid VPC parameters.	VPC parameter is illegal.	See the returned error message or contact technical support.
400	BMS.3028	Disk quota is abnormal.	cinder quota check fail.	See the returned error message or contact technical support.
400	Common.0018	tenant_id in the token is different from that in the URL.	tenantId in token is not the same with in URL.	Check whether the tenant token is correct.

HTTP Status Code	Returned Value	Description	Error Message	Handling Measure
400	BMS.0111	Password or key pair is not specified, or both specified.	none or multiple passwords specified, please specify one.	The password and key pair cannot be specified at the same time. Select one of them based on the login mode.
400	BMS.0395	Windows BMSs do not support remote login.	The server[%s] is windows system and can not support remote login.	See the returned error message or contact technical support.
400	BMS.0605	The BMS is locked.	the BMS is locked.	See the returned error message or contact technical support.

7.3 Task Responses

7.3.1 Task ID Response

Normal Response

Table 7-1 Normal response

Parameter	Type	Description
job_id	String	Specifies the task ID after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

Abnormal Response

Table 7-2 Abnormal response

Parameter	Type	Description
error	Dictionary data structure	Specifies the error returned when a task submission encounters an exception. For details, see **Table 7-3**.

Table 7-3 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response
```
{
    "job_id": "70a599e0-31e7-49b7-b260-868f441e862b"
}
```
- Abnormal response
```
{
    "error": {"message": "", "code": XXX}
}
```

7.3.2 Order ID Response

Normal Response

Table 7-4 Normal response

Parameter	Type	Description
order_id	String	Specifies the order ID returned after an order is submitted. You can query the order processing progress based on the ID. To query the order processing progress, log in to the management console and choose **Billing** > **My Orders**.
job_id	String	Specifies the task ID returned after a task command is issued. The task ID can be used to query the execution status of the task. For details about how to query the task execution status based on **job_id**, see **Querying Task Statuses**.

Abnormal Response

Table 7-5 Abnormal response

Parameter	Type	Description
error	Dictionary data structure [1]	Specifies the error returned when a task submission encounters an exception. For details, see **Table 7-6**.

Table 7-6 error data structure

Parameter	Type	Description
message	String	Specifies the error message.
code	String	Specifies the error code.

Example Response

- Normal response

```
{
    "order_id": "CS2009141523OQSEQ",
    "job_id": "ff808081748b760c01748b7f80370003"
}
```

- Abnormal response

```
{
    "error": {"message": "", "code": XXX}
}
```


8 Permissions and Supported Actions

8.1 Introduction

This section describes fine-grained permissions management for your BMS. If your Huawei Cloud account does not need individual IAM users, you may skip over this section.

By default, new IAM users do not have any permissions granted. You need to add a user to one or more groups, and assign policies or roles to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on BMS based on the permissions.

Roles are provided by IAM to define service-based permissions depending on users' job responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The permissions required for calling an API are determined by the actions supported by the API. Only users who have the permissions allowing the actions can call the API. For example, if an IAM user queries BMSs using an API, the user must have been granted permissions that allow the **bms:servers:list** action.

Supported Actions

BMS provides system-defined policies that can be directly used in IAM. You can also create custom policies and use them to supplement system-defined policies, implementing more refined access control. Actions supported by policies are specific to APIs. The following are common concepts related to policies:

- Permission: a statement in a policy that allows or denies certain operations.
- APIs: REST APIs that can be called in a custom policy.

- Actions: Added to a custom policy to control permissions for specific operations.
- Dependent actions: When assigning an action to users, you also need to assign dependent permissions for that action to take effect.
- IAM project/Enterprise project: A custom policy can be applied to IAM projects or enterprise projects or both. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management.

NOTE

√: supported; x: not supported

BMS supports the following actions that can be defined in custom policies:

- **Lifecycle management actions**, including actions supported by BMS's lifecycle management APIs, such as the APIs for creating BMSs and querying BMS details.
- **Status management actions**, including actions supported by BMS's status management APIs, such as the APIs for stopping, restarting, and starting BMSs in batches.
- **Password management actions**, including actions supported by BMS's password-related APIs, such as the APIs for querying whether BMS passwords can be reset.
- **Flavor query actions**, including actions supported by BMS's flavor-related APIs, such as the APIs for querying details about flavors and extended flavor information.
- **NIC management actions**, including actions supported by BMS's NIC-related APIs, such as the APIs for querying BMS NICs.
- **Disk management actions**, including actions supported by BMS's disk-related APIs, such as the APIs for attaching disks to a BMS and detaching disks from a BMS.
- **Metadata management actions**, including actions supported by BMS's metadata-related APIs, such as the API for updating the BMS metadata.
- **Tenant quota management actions**, including actions supported by BMS's quota-related APIs, such as the API for querying tenant quotas.

Action Descriptions

- The actions and scopes of native OpenStack APIs of BMS are the same as those of ECS. For details, see in *Elastic Cloud Server API Reference*.
- When customizing a BMS user policy in IAM, add the ecs:*:get and ecs:*:list permissions. Otherwise, functions on some pages cannot work properly.
- High-speed networks and user-defined networks do not support enterprise projects. To use the two functions, use the primary account.
- The vpc:ports:get permission must be assigned to the sub-account of an enterprise project. Otherwise, the EIP and security group cannot be properly displayed on the BMS details page.

8.2 Lifecycle Management

Permissions	API	Action	IAM Project	Enterprise Project
Creating a BMS	POST /v1/project_id}/baremetalservers	bms:servers:create	√	√
Querying Details About BMSs	GET /v1/{project_id}/baremetalservers/detail	bms:servers:list	√	√
Querying Details About a BMS	GET /v1/{project_id}/baremetalservers/{server_id}	bms:servers:get	√	√

8.3 Status Management

Permissions	API	Action	IAM Project	Enterprise Project
Change the Name of a BMS	PUT /v1/{project_id}/baremetalservers/{server_id}	bms:servers:put	√	√
Reinstalling the BMS OS	POST /v1/{project_id}/baremetalservers/{server_id}/reinstallos	bms:servers:reInstallOS	√	√
Stopping BMSs in a Batch	POST /v1/{project_id}/baremetalservers/action	bms:servers:stop	√	√
Restarting BMSs in a Batch	POST /v1/{project_id}/baremetalservers/action	bms:servers:reboot	√	√
Starting BMSs in a Batch	POST /v1/{project_id}/baremetalservers/action	bms:servers:start	√	√

8.4 Flavor Query

Permissions	API	Action	IAM Project	Enterprise Project
Querying Details About Flavors and Extended Flavor Information	GET /v1/{project_id}/baremetalservers/flavors	bms:serverFlavors:get	√	√

8.5 NIC Management

Permissions	API	Action	IAM Project	Enterprise Project
Querying Information About BMS NICs	GET /v1/{project_id}/baremetalservers/{server_id}/os-interface	bms:servers:get	√	√

8.6 Disk Management

Permissions	API	Action	Dependent Actions	IAM Project	Enterprise Project
Detaching a Disk from a BMS	DELETE /v1/{project_id}/baremetalservers/{server_id}/detachvolume/{attachment_id}	bms:servers:detachVolume	-	√	√
Attaching a Disk to a BMS	POST /v1/{project_id}/baremetalservers/{server_id}/attachvolume	bms:servers:attachVolume	evs:volumes:use	√	√

Permissions	API	Action	Dependent Actions	IAM Project	Enterprise Project
Querying EVS disks attached to a BMS	GET /v1/{project_id}/baremetalservers/{server_id}/os-volume_attachments	bms:servers:get	-	√	√

8.7 Metadata Management

Permissions	APIs	Actions	IAM Project	Enterprise Project
Updating the Metadata of a BMS	POST /v1/{project_id}/baremetalservers/{server_id}/metadata	bms:servers:updateMetadata	√	√

8.8 Tenant Quota Management

Permissions	APIs	Actions	IAM Project	Enterprise Project
Querying the Tenant Quota	GET /v1/{project_id}/baremetalservers/limits	bms:serverQuotas:get	√	x

8.9 Password Management

Permissions	API	Action	IAM Project	Enterprise Project
Querying Whether a BMS Supports Password Reset	GET /v1/{project_id}/baremetalservers/{server_id}/os-resetpwd-flag	bms:servers:get	√	√

Permissions	API	Action	IAM Project	Enterprise Project
Obtaining a password from a Windows BMS	GET /v1/{project_id}/baremetalservers/{server_id}/os-server-password	bms:servers:get	√	√
Deleting the password of a Windows BMS	DELETE /v1/{project_id}/baremetalservers/{server_id}/os-server-password	bms:servers:deletePassword	√	√

A Appendix

A.1 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": "65ewtrgaggshhk1223245sghjlse684b",
            "is_domain": false,
            "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
            "name": "project_name",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4adasfjljaaaakla12334jklga9sasfg"
            },
            "id": "a4adasfjljaaaakla12334jklga9sasfg",
            "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.

A.2 Obtaining an Account ID

An account ID is required for some URLs when an API is called. To obtain an account ID, 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 **My Credentials** page, view the **Account ID**.

B Change History

Released On	Description
2023-05-23	This issue is the third official release. Added error code BMS.0089. For details, see **Error Codes**.
2023-02-28	This issue is the second official release. Added new error codes and updated existing error codes in **Error Codes**. Added constraints in **Creating BMSs**.
2022-09-30	This issue is the first official release.

