Scalable File Service

API Reference

Issue 01

Date 2022-09-30

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.

Contents

1 Before You Start

1.1 Overview

Welcome to *Scalable File Service API Reference*. Scalable File Service (SFS) is a network attached storage (NAS) service that provides scalable, high-performance file storage. With SFS, you can enjoy shared file access spanning multiple Elastic Cloud Servers (ECSs), Bare Metal Servers (BMSs), and containers created on Cloud Container Engine (CCE).

This document describes how to use application programming interfaces (APIs) to perform operations on SFS resources, such as creating, querying, deleting, and updating a file system. For details about all supported operations, see **2 API Overview**.

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

1.2 API Calling

SFS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS requests. For details about API calling, see **3 Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoint of SFS, see "Regions and Endpoints."

1.4 Constraints

- The numbers of file systems that you can create and their capacities are determined by your quotas. To view or increase the quotas, see **Quotas**.
- For more constraints, see API description.

1.5 Concepts

- Account

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

- User

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

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

- Region

  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***.

2 API Overview

SFS Turbo APIs

By using the SFS Turbo APIs, you can create, delete, query SFS Turbo file systems as well as expanding the capacity of an SFS Turbo file system.

Table 2-1 API overview

Type	Subtype	Description
SFS Turbo API	Lifecycle management	Include creating file systems, deleting file systems, querying file system lists, and querying file system details.
SFS Turbo API	Storage capacity management	Expand the capacity of a specified file system.
SFS Turbo API	Connection management	Change the security group bound to an SFS Turbo file system.

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 **8.3 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 **3.2 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 as specified in the **Content-Type** header field. The request body transfers content except the request header.

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

In the case of the API used to **obtain a user token**, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, ******** (login password), and *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": "********",
                    "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.

A cloud service can be deployed as either a project-level service or global service.

- For a project-level service, you need to obtain a project-level token. When you call the API, set **auth.scope** in the request body to **project**.
- For a global service, you need to obtain a global token. When you call the API, set **auth.scope** in the request body to **domain**.

IMS 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",
                "password": "********",
                "domain": {
                    "name": "domainname"
                }
            }
        }
    },
    "scope": {
        "project": {
            "name": "xxxxxxxx"
        }
    }
  }
}
```

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 **8.1 Status Codes**.

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

Response Header

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

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

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

(Optional) Response Body

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

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

```
{
    "token": {
```



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

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

```
{
    "error_msg": "The format of message is error",
    "error_code": "AS.0001"
}
```

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

4 Getting Started (SFS Turbo)

This section describes how to use APIs by calling an API to create an SFS Turbo file system.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use one token for authentication, you can cache it to avoid frequently calling.

Involved APIs

If you use a token for authentication, you must obtain the token and add **X-Auth-Token** to the request header of the API when making a call. The following APIs are involved in the request for creating an SFS Turbo file system:

- API for obtaining tokens from IAM
- API for creating an SFS Turbo file system. For details, see Creating a File System.

Procedure

Step 1 Obtain the token by following instructions in **3.2 Authentication**.

Step 2 Add **X-Auth-Token** to the request header.

Step 3 Specify the following parameters in the request body:

```
{
  "share": {
    "name": "sfs-turbo-test",
    "share_proto": "NFS",
    "share_type": "STANDARD",
    "size": 100,
    "availability_zone": "az1",
    "vpc_id": "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9",
    "subnet_id": "b8884abe-f47b-4917-9f6c-f64825c365db",
    "security_group_id": "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4"
  }
}
```

Step 4 Send the request **POST https://*Endpoint of SFS Turbo*/v1/{project_id}/sfs-turbo/shares**.

Step 5 After the request is successfully responded, the ID and name of the SFS Turbo file system are returned.

If the request fails, an error code and error information are returned. For details about the error codes, see the abnormal return values of the corresponding API.

Query SFS Turbo file system details based on the returned file system ID. For details, see Querying Details About a Single File System.

If the returned status of the file system is **200**, the SFS Turbo file system is successfully created. For details about the return values of request exceptions, see the abnormal return values of the corresponding API. For other statuses, see **7.1 SFS Turbo File System Statuses**.

You can query and delete an SFS Turbo file system based on the file system ID.

----End

Configuration Example

If the token has been obtained, you can run the following **curl** command to create an SFS Turbo file system:

```
curl -k -i -X POST -H "X-Auth-Token: token_value" -H "Content-Type: application/json" -d '{"share": {"name": "sfs-turbo-test", "share_proto": "NFS", "share_type": "STANDARD", "size": 100, "availability_zone": "az1", "vpc_id": "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9", "subnet_id": "b8884abe-f47b-4917-9f6c-f64825c365db", "security_group_id": "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4"}}' "https://127.0.0.1:8979/v1/xxxbxbex5cfx41f0a08ay915fd79240d/sfs-turbo/shares"
```

5 SFS Turbo APIs

5.1 Lifecycle Management

5.1.1 Deleting a File System

Function

This API is used to delete a file system.

URI

DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}

Table 5-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

None

Example Requests

None

Example Responses

None

Status Codes

Status Code	Description
202	File system deletion request delivered.

Error Codes

See **Error Codes**.

5.1.2 Querying Details About a File System

Function

This API is used to query details about an SFS Turbo file system.

URI

GET /v1/{project_id}/sfs-turbo/shares/{share_id}

Table 5-3 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Parameter	Mandatory	Type	Description
share_id	Yes	String	File system ID

Request Parameters

Table 5-4 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

Status code: 200

Table 5-5 Response body parameters

Parameter	Type	Description
action_progress	ActionProgress object	Task progress of the SFS Turbo file system. For example, the file system creation progress.
version	String	Version of an SFS Turbo file system
avail_capacity	String	Available capacity of the SFS Turbo file system, in GB
availability_zone	String	Code of the AZ where the SFS Turbo file system resides
az_name	String	Name of the AZ where the SFS Turbo file system resides
created_at	String	Time when the file system was created. UTC time, for example: 2018-11-19T04:02:03
crypt_key_id	String	ID of the encryption key specified by the user. This parameter is not returned for non-encrypted file systems.
expand_type	String	File system extension type. If the requested file system is an enhanced file system, **bandwidth** will be returned. Or, this parameter is not returned.
export_location	String	Location where the SFS Turbo file system is mounted
id	String	ID of the SFS Turbo file system

Parameter	Type	Description
name	String	Name of the SFS Turbo file system specified during creation
pay_model	String	Billing mode of the SFS Turbo file system. Value **0** indicates pay-per-use, and **1** indicates yearly/monthly subscription. Enumeration values: • **0** • **1**
region	String	Region where the SFS Turbo file system resides
security_group_id	String	ID of the security group specified by the user
share_proto	String	Protocol used by the SFS Turbo file system. The valid value is **NFS**.
share_type	String	Storage class of the SFS Turbo file system. The valid values are **STANDARD** and **PERFORMANCE**.
size	String	Total capacity of the SFS Turbo file system, in GB
status	String	Status of the SFS Turbo file system. The value can be as follows: **100** (creating), **200** (available), **400** (deleted), and **303** (creation failed).
sub_status	String	Sub-status of the SFS Turbo file system. The value can be as follows: **121** (expanding), **221** (expansion succeeded), and **321** (expansion failed).
subnet_id	String	ID of the subnet specified by the user
vpc_id	String	ID of the VPC specified by the user

Table 5-6 ActionProgress

Parameter	Type	Description
CREATING	String	File system creation progress

Example Requests

None

Example Responses

Status code: 200

Response body for querying a file system


```
{
  "id" : "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
  "name" : "sfs-turbo-8468",
  "status" : "200",
  "version" : "1.0.0",
  "region" : "north-1",
  "created_at" : "2018-11-19T04:02:03",
  "export_location" : "192.168.0.90:/",
  "action_progress" : { },
  "share_type" : "STANDARD",
  "sub_status" : "330",
  "availability_zone" : "az1.dc1",
  "az_name" : "az1",
  "vpc_id" : "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
  "subnet_id" : "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
  "security_group_id" : "50586458-aec9-442c-bb13-e08ddc6f1b7a",
  "size" : "500.00",
  "pay_model" : "0",
  "avail_capacity" : "500.00",
  "share_proto" : "NFS"
}
```


Status Codes

Status Code	Description
200	Response body for querying a file system

Error Codes

See **Error Codes**.

5.1.3 Creating a File System

Function

This API is used to create a file system.

URI

POST /v1/{project_id}/sfs-turbo/shares

Table 5-7 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 5-8 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Table 5-9 Request body parameters

Parameter	Mandatory	Type	Description
share	Yes	**Share** object	Request body for creating a file system

Table 5-10 Share

Parameter	Mandatory	Type	Description
availability_zone	Yes	String	Code of the AZ where the file system resides
description	No	String	Description of the file system, which can contain 0 to 255 characters. This parameter is not supported by the current version. Minimum: **0** Maximum: **255**
enterprise_project_id	No	String	ID of the enterprise project to which the file system will be added
metadata	No	**Metadata** object	Metadata of the file system. The value consists of key and value pairs as a directory of strings.

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the SFS Turbo file system. The name contains 4 to 64 characters and must start with a letter. It can contain letters (case insensitive), digits, hyphens (-), and underscores (_), and cannot contain other special characters. Minimum: **4** Maximum: **64**
security_group_id	Yes	String	Security group ID of a tenant in a region
share_proto	Yes	String	File sharing protocol. The valid value is **NFS**. Network File System (NFS) is a distributed file system protocol that allows different computers and operating systems to share data over a network.
share_type	Yes	String	File system type. The valid values are **STANDARD** and **PERFORMANCE**.
size	Yes	Integer	File system capacity. For a standard file system, the value ranges from **500** to **32768**, in GB. For an enhanced file system, with the **expand_type** field set under **metadata**, the value ranges from **10240** to **327680**, in GB. Minimum: **500** Maximum: **327680**
subnet_id	Yes	String	Subnet ID of a tenant in a VPC
vpc_id	Yes	String	VPC ID of a tenant in a region
backup_id	No	String	Backup ID. This parameter is mandatory if you create a file system from a backup.

Table 5-11 Metadata

Parameter	Mandatory	Type	Description
crypt_key_id	No	String	ID of a KMS professional key. This parameter is used if you want to create an encrypted file system.
dedicated_flavor	No	String	VM flavor used for creating a dedicated file system
dedicated_storage_id	No	String	ID of the dedicated distributed storage used when creating a dedicated file system
expand_type	No	String	Extension type. The valid value is **bandwidth**, indicating an enhanced file system.

Response Parameters

Status code: 202

Table 5-12 Response body parameters

Parameter	Type	Description
id	String	ID of the created SFS Turbo file system
name	String	Name of the created SFS Turbo file system
status	String	Status of the SFS Turbo file system

Example Requests

```
{
  "share" : {
    "availability_zone" : "az1",
    "metadata" : {
      "crypt_key_id" : "015bf4b8-73cc-4235-8595-46931de7dfd0",
      "dedicated_flavor" : "c3.large.4",
      "dedicated_storage_id" : "198f0704-63fc-4d85-b15d-c25caa4d3264"
    },
    "name" : "sfs-turbo-test",
    "security_group_id" : "8c4ebbd0-6edf-4aae-8353-81ce6d06e1f4",
    "share_proto" : "NFS",
    "share_type" : "STANDARD",
    "size" : 500,
    "subnet_id" : "b8884abe-f47b-4917-9f6c-f64825c365db",
    "vpc_id" : "d651ea2b-2b20-4c6d-8bbf-2adcec18dac9"
  }
}
```

Example Responses

Status code: 202

Response body for creating a file system

```
{
  "id" : "708c017c-54b5-429a-a098-7692e23fa518",
  "name" : "sfs-turbo-test",
  "status" : "100"
}
```

Status Codes

Status Code	Description
202	Response body for creating a file system

Error Codes

See **Error Codes**.

5.1.4 Obtaining the File System List

Function

This API is used to obtain the file system list.

URI

GET /v1/{project_id}/sfs-turbo/shares/detail

Table 5-13 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Table 5-14 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	Number of the file systems returned. The maximum value is **200**. Minimum: **0** Maximum: **200**

Parameter	Mandatory	Type	Description
offset	No	Integer	Offset where the file system listing starts Minimum: **0** Maximum: **200**

Request Parameters

Table 5-15 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

Status code: 200

Table 5-16 Response body parameters

Parameter	Type	Description
shares	Array of **Shares** objects	List of SFS Turbo file systems
count	Integer	Number of SFS Turbo file systems

Table 5-17 Shares

Parameter	Type	Description
action_progress	**ActionProgress** object	Task progress of the SFS Turbo file system. For example, the file system creation progress.
version	String	Version of an SFS Turbo file system
avail_capacity	String	Available capacity of the SFS Turbo file system, in GB
availability_zone	String	Code of the AZ where the SFS Turbo file system resides
az_name	String	Name of the AZ where the SFS Turbo file system resides

Parameter	Type	Description
created_at	String	Time when the file system was created. UTC time, for example: 2018-11-19T04:02:03
crypt_key_id	String	ID of the encryption key specified by the user. This parameter is not returned for non-encrypted file systems.
expand_type	String	File system extension type. If the requested file system is an enhanced file system, **bandwidth** will be returned. Or, this parameter is not returned.
export_location	String	Location where the SFS Turbo file system is mounted
id	String	ID of the SFS Turbo file system
name	String	Name of the SFS Turbo file system specified during creation
pay_model	String	Billing mode of the SFS Turbo file system. Value **0** indicates pay-per-use, and **1** indicates yearly/monthly subscription. Enumeration values: • **0** • **1**
region	String	Region where the SFS Turbo file system resides
security_group_id	String	ID of the security group specified by the user
share_proto	String	Protocol used by the SFS Turbo file system. The valid value is **NFS**.
share_type	String	Storage class of the SFS Turbo file system. The valid values are **STANDARD** and **PERFORMANCE**.
size	String	Total capacity of the SFS Turbo file system, in GB
status	String	Status of the SFS Turbo file system. The value can be as follows: **100** (creating), **200** (available), **400** (deleted), and **303** (creation failed).
sub_status	String	Sub-status of the SFS Turbo file system. The value can be as follows: **121** (expanding), **221** (expansion succeeded), and **321** (expansion failed).
subnet_id	String	ID of the subnet specified by the user

Parameter	Type	Description
vpc_id	String	ID of the VPC specified by the user

Table 5-18 ActionProgress

Parameter	Type	Description
CREATING	String	File system creation progress

Example Requests

None

Example Responses

Status code: 200

Response body for querying the file system list


```
{
  "shares" : [ {
    "id" : "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
    "name" : "sfs-turbo-8468",
    "status" : "200",
    "version" : "1.0.0",
    "region" : "north-1",
    "created_at" : "2018-11-19T04:02:03",
    "export_location" : "192.168.0.90:/",
    "action_progress" : {
      "CREATING" : "22%"
    },
    "share_type" : "STANDARD",
    "sub_status" : "230",
    "availability_zone" : "az1.dc1",
    "az_name" : "az1",
    "vpc_id" : "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
    "subnet_id" : "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
    "security_group_id" : "50586458-aec9-442c-bb13-e08ddc6f1b7a",
    "size" : "500.00",
    "pay_model" : "0",
    "avail_capacity" : "500.00",
    "share_proto" : "NFS"
  }, {
    "id" : "65f2d30b-7b4e-4786-9608-4324faef6646",
    "name" : "sfs-turbo-df12",
    "status" : "200",
    "version" : "1.0.0",
    "region" : "north-1",
    "created_at" : "2018-11-15T02:32:10",
    "export_location" : "192.168.0.197:/",
    "action_progress" : {
      "CREATING" : "22%"
    },
    "share_type" : "STANDARD",
    "sub_status" : "230",
    "availability_zone" : "az1.dc1",
    "az_name" : "az1",
    "vpc_id" : "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
    "subnet_id" : "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
```



```
    "security_group_id" : "50586458-aec9-442c-bb13-e08ddc6f1b7a",
    "size" : "500.00",
    "pay_model" : "0",
    "avail_capacity" : "500.00",
    "share_proto" : "NFS"
  } ]
}
```

Status Codes

Status Code	Description
200	Response body for querying the file system list

Error Codes

See **Error Codes**.

5.2 Storage Capacity Management

5.2.1 Expanding the Capacity of a File System

Function

This API is used to expand the capacity of a file system.

URI

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action

Table 5-19 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Table 5-21 Request body parameters

Parameter	Mandatory	Type	Description
extend	Yes	**Extend** object	Object of **extend**

Table 5-22 Extend

Parameter	Mandatory	Type	Description
new_size	Yes	Integer	New capacity of the file system, in GB The expansion increment is greater than or equal to 100 GB. For a standard file system, the value ranges from **500** to **32768**, in GB. For an enhanced file system, the value ranges from **10240** to **327680**, in GB. Minimum: **500** Maximum: **327680**

Response Parameters

Status code: 202

Table 5-23 Response body parameters

Parameter	Type	Description
id	String	ID of the SFS Turbo file system
name	String	Name of the SFS Turbo file system

Example Requests

```
{
  "extend" : {
    "new_size" : 500
  }
}
```

Example Responses

Status code: 202

Response body for expanding the capacity of a file system

```
{
  "id" : "67d4bd5e-7b2f-4c24-9a0b-c0038940c6f8",
```



```
  "name" : "sfs-turbo-test"
}
```

Status Codes

Status Code	Description
202	Response body for expanding the capacity of a file system

Error Codes

See **Error Codes**.

5.3 Connection Management

5.3.1 Changing the Security Group Associated with a File System

Function

This API is used to change the security group associated with an SFS Turbo file system. Security group change is an asynchronous task. You can check whether the security group is changed based on the value of **sub_status** returned after calling the API to query details of a file system. If value **232** is returned, the security group has been changed.

URI

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action

Table 5-24 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-25 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	MIME type

Table 5-26 Request body parameters

Parameter	Mandatory	Type	Description
change_security_group	Yes	ChangeSecurityGroup object	Object of **change_security_group**

Table 5-27 ChangeSecurityGroup

Parameter	Mandatory	Type	Description
security_group_id	Yes	String	ID of the security group to be changed

Response Parameters

Status code: 202

Table 5-28 Response body parameters

Parameter	Type	Description
id	String	ID of the SFS Turbo file system

Example Requests

```
{
  "change_security_group" : {
    "security_group_id" : "26f6b565-240e-43c3-8867-03f0bd975433"
  }
}
```

Example Responses

Status code: 202

ID of the SFS Turbo file system

```
{
  "id" : "67d4bd5e-7b2f-4c24-9a0b-c0038940c6f8"
}
```


Status Codes

Status Code	Description
202	ID of the SFS Turbo file system

Error Codes

See **Error Codes**.

5.4 Tag Management

5.4.1 Adding a Tag for a File System

Function

This API is used to add a tag to a specified file system. A maximum of 10 tags can be added to one file system. Tag keys added to the same file system must be unique. This API is idempotent. If the file system already has the key to be added, the tag will be updated.

URI

POST /v1/{project_id}/sfs-turbo/{share_id}/tags

Table 5-29 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-30 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Table 5-31 Request body parameters

Parameter	Mandatory	Type	Description
tag	Yes	**ResourceTag** object	Description of the **resource_tag** field

Table 5-32 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	Tag key. It can contain a maximum of 36 characters. It cannot be left blank and cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **1** Maximum: **36**
value	Yes	String	Tag value. It can contain a maximum of 43 characters and can be left blank. It cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **43**

Response Parameters

None

Example Requests

```
{
  "tag" : {
    "key" : "key1",
    "value" : "value1"
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
204	Tag adding request delivered.

Error Codes

See **Error Codes**.

5.4.2 Deleting a Tag of a File System

Function

This API is used to delete a tag of a specified file system. If the key to be deleted does not exist, error 404 will be returned.

URI

DELETE /v1/{project_id}/sfs-turbo/{share_id}/tags/{key}

Table 5-33 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Parameter	Mandatory	Type	Description
key	Yes	String	Tag key, which can contain a maximum of 36 characters. It cannot be left blank and cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). When this API is called to delete a tag, if the tag key contains special characters that cannot be directly resolved by the URL, the key needs to be escaped.

Request Parameters

Table 5-34 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

None

Example Requests

None

Example Responses

None

Status Codes

Status Code	Description
204	File system tag deleted.

Error Codes

See **Error Codes**.

5.4.3 Querying Tags of a File System

Function

This API is used to query all tags of a specified file system.

URI

GET /v1/{project_id}/sfs-turbo/{share_id}/tags

Table 5-35 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-36 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

Status code: 200

Table 5-37 Response body parameters

Parameter	Type	Description
tags	Array of **ResourceTag** objects	Tag list
sys_tags	Array of **ResourceTag** objects	Only users with the op_service permission can obtain this field. 1. This field currently contains only one resource_tag structure key, **_sys_enterprise_project_id**. 2. The key contains only value **0** currently, which indicates the default enterprise project. This field is not returned for users without the op_service permission.

Table 5-38 ResourceTag

Parameter	Type	Description
key	String	Tag key. It can contain a maximum of 36 characters. It cannot be left blank and cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **1** Maximum: **36**
value	String	Tag value. It can contain a maximum of 43 characters and can be left blank. It cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **43**

Example Requests

None

Example Responses

Status code: 200

Response body for query all tags of a specified file system

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

Status Codes

Status Code	Description
200	Response body for query all tags of a specified file system

Error Codes

See **Error Codes**.

5.4.4 Batch Adding Tags to a File System

Function

This API is used to batch add tags for a specified file system.

A maximum of 10 tags can be added to one file system. Tag keys added to the same file system must be unique. This API is idempotent. If the file system already has the key to be added, the tag will be updated.

URI

POST /v1/{project_id}/sfs-turbo/{share_id}/tags/action

Table 5-39 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-40 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Table 5-41 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	Operation identifier. The value is **create**. Use **create** if you want to batch add tags to a file system. Enumeration values: • **create**
tags	No	Array of **ResourceTag** objects	Tag list. This field is mandatory for users. For users with the op_service permission, choose either this field or **sys_tags**.
sys_tags	No	Array of **ResourceTag** objects	System tag list. This field is available only to users with the op_service permission. Choose either this field or **tags**. Only one resource_tag structure key, **_sys_enterprise_project_id**, is used in TMS calls.

Table 5-42 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	Tag key. It can contain a maximum of 36 characters. It cannot be left blank and cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **1** Maximum: **36**
value	Yes	String	Tag value. It can contain a maximum of 43 characters and can be left blank. It cannot contain the following characters: ASCII (0-31), equal signs (=), asterisks (*), left angle brackets (<), right angle brackets (>), backslashes (\), commas (,), vertical bars (|), and slashes (/). It can contain only letters, digits, hyphens (-), and underscores (_). Minimum: **0** Maximum: **43**

Response Parameters

None

Example Requests

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


Example Responses

None

Status Codes

Status Code	Description
204	File system tags added.

Error Codes

See **Error Codes**.

5.4.5 Querying Tags of All File Systems of a Tenant

Function

This API is used to query the tags of all file systems of a tenant.

URI

GET /v1/{project_id}/sfs-turbo/tags

Table 5-43 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID

Request Parameters

Table 5-44 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Response Parameters

Status code: 200

Table 5-45 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	Tag list

Table 5-46 Tag

Parameter	Type	Description
key	String	Tag key. It can contain a maximum of 127 characters. It cannot be left blank. Minimum: **1** Maximum: **127**
values	Array of strings	Tag values. Each value can contain a maximum of 255 characters. An empty list of values can match with any value. All values of a tag key are in the OR relationship. Minimum: **0** Maximum: **255**

Example Requests

None

Example Responses

Status code: 200

Response body for querying a file system

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

Status Codes

Status Code	Description
200	Response body for querying a file system

Error Codes

See **Error Codes**.

5.5 Name Management

5.5.1 Changing the Name of a File System

Function

This API is used to change the name of an SFS Turbo file system.

URI

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action

Table 5-47 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID
share_id	Yes	String	File system ID

Request Parameters

Table 5-48 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Account token
Content-Type	Yes	String	MIME type

Table 5-49 Request body parameters

Parameter	Mandatory	Type	Description
change_name	Yes	**ChangeShareName** object	Target SFS Turbo file system

Table 5-50 ChangeShareName

Parameter	Mandatory	Type	Description
name	Yes	String	Name of the target SFS Turbo file system

Response Parameters

None

Example Requests

```
{
  "change_name" : {
    "name" : "sfs-turbo-test1"
  }
}
```

Example Responses

None

Status Codes

Status Code	Description
204	Request successful
400	Invalid parameter
500	Internal error
409	The file system name already exists.

Error Codes

See **Error Codes**.

6 Permissions Policies and Supported Actions

6.1 Introduction

This section describes fine-grained permissions management for your SFS. If your Huawei Cloud account does not need individual IAM users, then you may skip over this section.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant users permissions by using roles and policies. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user 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 the permissions required to call all APIs, but IAM users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query ECSs using an API, the user must have been granted permissions that allow the **ecs:servers:list** action.

Supported Actions

SFS 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. Operations supported by policies are specific to APIs. The following are common concepts related to policies:

- Permissions: Statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.
- Actions: Specific operations that are allowed or denied.
- Related actions: Actions on which a specific action depends to take effect. When assigning permissions for the action to a user, you also need to assign permissions for the related actions.
- IAM or enterprise projects: Type of projects for which an action will take effect. Policies that contain actions for both IAM and enterprise projects can be used and take effect for both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM.

  NOTE

  The check mark (√) and cross symbol (x) indicate that an action takes effect or does not take effect for the corresponding type of projects.

SFS supports the following actions that can be defined in custom policies:

- **File Systems**, including actions supported by all SFS Turbo file system APIs, such as the APIs for creating file systems, querying file system lists, querying details about a single file system, and deleting file systems.

6.2 Supported Actions

6.2.1 SFS Turbo Actions

File Systems

Permission	API	Action	Dependent Permission	IAM Project	Enterprise Project
Creating a File System	POST /v1/{project_id}/sfs-turbo/shares	sfsturbo:shares:createShare	• You must have VPC-related permissions when creating an SFS Turbo instance, including the permissions for verifying VPCs, subnets, and security groups, creating virtual IP addresses and ports, and creating security group rules. You must add the following action: – "vpc:*:*" • The KMS Administrator permission needs to be configured for the encrypted instance in the project. • If you have deployed and subscribed to Dedicated Distributed Storage Service (DSS) and want to	√	√

Permission	API	Action	Dependent Permission	IAM Project	Enterprise Project
			create SFS Turbo file systems, add the following actions: – "dss:*:get", – "dss:*:list", – "dss:*:count"		
Querying Details About All File Systems	GET /v1/{project_id}/sfs-turbo/shares/detail	sfsturbo:shares:getAllShares	-	√	√
Querying Details About a File System	GET /v1/{project_id}/sfs-turbo/shares/{share_id}	sfsturbo:shares:getShare	-	√	√

Permission	API	Action	Dependent Permission	IAM Project	Enterprise Project
Deleting a File System	DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}	sfsturbo:shares:deleteShare	• To delete an SFS Turbo instance, you must have VPC-related permissions, including the permissions for deleting virtual IP addresses, ports, and security group rules. You must add the following action: – "vpc:*:*" • If you have deployed and subscribed to DSS and want to create SFS Turbo file systems, add the following actions: – "dss:*:get", – "dss:*:list", – "dss:*:count"	√	√

File System Expansion

Permission	API	Action	IAM Project	Enterprise Project
Expanding the Capacity of a File System	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action	sfsturbo:shares:shareAction	√	√

APIs for Console Only

Permission	API	Action	Dependent Permission	IAM Project	Enterprise Project
Changing a Security Group	For Console only	sfsturbo:shares:shareAction	To modify a security group, you must have the following permissions: • vpc:securityGroups:* • vpc:securityGroupRules:*	√	√
Querying the SFS Turbo Quota	For Console only	sfsturbo:shares:getQuota	-	√	√
Obtaining the AZ Information	For Console only	sfsturbo:shares:getAZInfo	-	√	√
Obtaining SFS Turbo Specifications	For Console only	sfsturbo:shares:getFlavors	-	√	√
Checking the Name of a File System	For Console only	sfsturbo:shares:checkShareName	-	√	√

7 Common Parameters

7.1 SFS Turbo File System Statuses

- SFS Turbo file system status elements

Returned Value	Description
100	CREATING: The file system is being created.
200	ACTIVE: The file system is active. An SFS Turbo file system can be mounted in this status.
300	FAILED: The job failed.
303	CREATE_FAILED: The cluster failed to be created.
400	DELETED: The cluster has been deleted.
800	FROZEN: The cluster has been frozen.

7.2 SFS Turbo File System Substatuses

- SFS Turbo file system substatus elements

Returned Value	Description
121	Expanding the capacity online.
221	Online capacity expansion succeeded.
321	Failed to perform online capacity expansion.

8 Appendix

8.1 Status Codes

- Normal

Returned Value	Description
200 OK	Specifies the normal response for the GET and PUT operations.
201 Created	Specifies the normal response for the POST operation.
202 Accepted	The request has been accepted for processing.
204 No Content	Specifies the normal response for the DELETE operation.

- Abnormal

Returned Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter a username and the password to access the requested page.
403 Forbidden	Access to the requested page is forbidden.
404 Not Found	The requested page was not found.

Returned Value	Description
405 Method Not Allowed	You are not allowed to use the method specified in the request.
406 Not Acceptable	The response generated by the server could not be accepted by the client.
407 Proxy Authentication Required	You must use the proxy server for authentication. Then the request can be processed.
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	The request is not completed because of a service error.
501 Not Implemented	The request is not completed because the server does not support the requested function.
502 Bad Gateway	The request is not completed because the server receives an invalid response from an upstream server.
503 Service Unavailable	The request is not completed because the service is unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

8.2 SFS Turbo Error Codes

Status Code	Error Code	Error Message	Description	Solution
400	SFS.TURBO.0001	Parameter error	Invalid parameters.	Use valid parameters and try again.
404	SFS.TURBO.0002	Cluster not found	The requested object does not exist or you do not have permissions to access it.	Use valid parameters and try again.
400	SFS.TURBO.0003	Invalid name	Invalid name.	Use valid parameters and try again.

Status Code	Error Code	Error Message	Description	Solution
400	SFS.TURBO.0004	Invalid vpc	Invalid VPC.	Use valid parameters and try again.
500	SFS.TURBO.0005	Internal server error	Internal error.	Contact technical support.
400	SFS.TURBO.0006	Invalid subnet	Invalid subnet.	Use valid parameters and try again.
400	SFS.TURBO.0007	Invalid share type	Invalid file system type.	Use valid parameters and try again.
400	SFS.TURBO.0008	Invalid size	Unsupported file system size.	Use valid parameters and try again.
400	SFS.TURBO.0009	Name has existed	The file system name already exists.	Use valid parameters and try again.
400	SFS.TURBO.0010	Quota exceeds	Insufficient quota.	Submit a service order to increase quota.
400	SFS.TURBO.0011	Cluster is doing something	Another operation is being performed on the file system.	Wait until that operation is complete and try again.
400	SFS.TURBO.0012	Operation is not allowed	A yearly/monthly file system cannot be resized or deleted via API.	Manage yearly/monthly file systems on the console.
400	SFS.TURBO.0016	Res tag count already reach max value	The maximum number of tags has been reached for the resource.	Delete unnecessary tags.

Status Code	Error Code	Error Message	Description	Solution
400	SFS.TURBO.0017	Invalid tag key param	The length of the resource tag key is invalid.	Use valid parameters and try again.
400	SFS.TURBO.0018	Invalid tag value param	The length of the resource tag value is invalid.	Use valid parameters and try again.
400	SFS.TURBO.0024	Operation is not allowed	Unsupported operation.	Contact technical support.
400	SFS.TURBO.0025	Invalid tag key param	The resource tag key contains invalid characters.	Use valid parameters and try again.
400	SFS.TURBO.0026	Invalid tag value param	The resource tag value contains invalid characters.	Use valid parameters and try again.
400	SFS.TURBO.0027	Invalid security group	Invalid security group.	Use valid parameters and try again.
400	SFS.TURBO.0028	Invalid crypt key	Invalid KMS key.	Use valid parameters and try again.
400	SFS.TURBO.0029	Subnet has not enough ips	Insufficient IP addresses in the subnet.	Use valid parameters and try again.
400	SFS.TURBO.0030	Ecs resource not enough	The ECS specification is sold out in the selected AZ.	Change the AZ and try again.

8.3 Obtaining a Project ID

Scenarios

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

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

Obtain the Project ID by Calling an API

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

The API used to obtain a project ID is GET https://{Endpoint}/v3/projects. {Endpoint} is the IAM endpoint and can be obtained from Regions and Endpoints. For details about API authentication, see **3.2 Authentication**.

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

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

Obtain a Project ID from the Console

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

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

A Change History

Released On	Description
2022-09-30	This issue is the first official release.

