Scalable File Service Turbo

API Reference

Issue 01

Date 2025-07-22

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

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

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

Contents

1 Before You Start

Overview

Welcome to *Scalable File Service Turbo API Reference*. Scalable File Service Turbo (SFS Turbo) is a network-attached storage (NAS) service that provides scalable, high-performance file storage. With SFS Turbo, 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 Turbo resources, such as creating, querying, deleting, and updating a file system. For details about all supported operations, see **API Overview**.

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

API Calling

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

Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions.

Table 1-1 SFS Turbo endpoint information

Region Name	Region ID	Endpoint	Protocol
EU-Dublin	eu-west-101	sfs-turbo.eu-west-101.myhuaweicloud.eu	HTTPS

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 detailed constraints, see the constraints described in specific APIs.

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. Compute, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build 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 add resources to 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

These APIs allow you to use all SFS Turbo functions.

Lifecycle Management APIs

Table 2-1 Lifecycle management APIs

API	Description
Creating a File System	This API is used to create a file system.
Querying Details About a File System	This API is used to query details about an SFS Turbo file system.
Deleting a File System	This API is used to delete a file system.
Listing File Systems	This API is used to list file systems.
Expanding the Capacity of a File System	This API is used to expand the capacity of a file system.

Connection Management APIs

Table 2-2 Connection management APIs

API	Description
Changing the Security Group Associated with a File System	This API is used to change the security group associated with an SFS Turbo file system.

Tag Management APIs

Table 2-3 Tag management APIs

API	Description
Adding a Tag to a File System	This API is used to add a tag to a specified file system.
Querying Tags of a File System	This API is used to query all tags of a specified file system.
Deleting a Tag from a File System	This API is used to delete a tag from a specified file system.
Batch Adding Tags to a File System	This API is used to batch add tags to a specified file system.
Querying Tags of All File Systems of a Tenant	This API is used to query the tags of all file systems of a tenant.
Querying File Systems by Tag	This API is used to query file systems by tag.

Name Management APIs

Table 2-4 Name management APIs

API	Description
Changing the Name of a File System	This API is used to change the name of a file system.

Directory Management APIs

Table 2-5 Directory management APIs

API	Description
Creating Quota Limits for a Directory	This API is used to configure quota limits for a directory.
Updating Quota Limits of a Directory	This API is used to update the quota limits of a directory.
Querying Quota Limits of a Directory	This API is used to query the quota limits of a directory.
Removing Quota Limits from a Directory	This API is used to remove the quota limits from a directory.

API	Description
Creating a Directory	This API is used to create a directory.
Checking Whether a Directory Exists	This API is used to check whether a directory exists.
Deleting a Directory from a File System	This API is used to delete a directory from a file system.
Querying the Resource Usage of a Directory	This API is used to query the resource usage of a file system directory (including usages of subdirectories).

Permissions Management APIs

Table 2-6 Permissions management APIs

API	Description
Joining an AD Domain	This API is used to add a file system to an AD domain.
Querying the AD Domain Configuration	This API is used to query the AD domain configuration.
Modifying the AD Domain Configuration	This API is used to modify the AD domain configuration.
Leaving an AD Domain	This API is used to remove a file system from an AD domain.

Task Management APIs

Table 2-7 Task management APIs

API	Description
Querying Details About a Task	This API is used to query the details about a task.

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 "AK/SK Authentication" 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. If the request body contains full-width characters, these characters must be coded in UTF-8.

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.

SFS Turbo 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 store the token in ciphertext in the configuration file or environment variable 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 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

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:

- IAM API for obtaining a token
- API for creating an SFS Turbo file system. For details, see "Creating a File System" in the *Scalable File Service Turbo API Reference*.

Procedure

Step 1 Obtain the token by following instructions in **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 the SFS Turbo file system details based on the returned file system ID. For details, see "Querying Details About a Single File System" in the *Scalable File Service Turbo API Reference*.

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 **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 API

5.1 Lifecycle Management

5.1.1 Creating a File System

Function

This API is used to create a file system.

URI

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

Table 5-1 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Request Parameters

Table 5-2 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-3 Request body parameters

Parameter	Mandatory	Type	Description
share	Yes	**Share** object	The parameters for creating a file system.
bss_param	No	**BssInfo** object	This parameter is mandatory when the yearly/monthly billing mode is used.

Table 5-4 Share

Parameter	Mandatory	Type	Description
availability_zone	Yes	String	The code of the AZ where the file system resides.
description	No	String	The file system description, which can contain 0 to 255 characters. This parameter is not supported currently.
enterprise_project_id	No	String	The ID of the enterprise project that the file system will be added to.
metadata	No	**Metadata** object	The file system metadata. The value consists of one or multiple key-value pairs organized as a directory of strings.
name	Yes	String	The file system name. The name can contain 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.
security_group_id	Yes	String	The security group ID of a tenant in a region.

Parameter	Mandatory	Type	Description
share_proto	Yes	String	• Network File System (NFS) is a distributed file system protocol that allows different computers and OSs to share data over a network. You are advised to use NFS file systems with Linux servers. • Common Internet File System (CIFS) is a protocol used for network file sharing. CIFS is a dialect of the SMB protocol, meaning that it is a specific implementation or variation of the SMB protocol. CIFS is also a public or open version of the SMB protocol. It allows applications to access files on computers over the Internet and request for file services. Using CIFS, network files can be shared between Windows hosts. CIFS file systems cannot be mounted to Linux servers. You are advised to use CIFS file systems with Windows servers.

Parameter	Mandatory	Type	Description
share_type	Yes	String	The file system type. Valid values are **STANDARD** and **PERFORMANCE**. This field is not returned when the file system is being created. • For a previous-generation SFS Turbo file system, specify **STANDARD** for a Standard or Standard-Enhanced file system, and **PERFORMANCE** for a Performance or Performance-Enhanced file system. • For a 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, or 20 MB/s/TiB file system, this field is not verified. Specify either **STANDARD** or **PERFORMANCE**.

Parameter	Mandatory	Type	Description
size	Yes	Integer	• For a previous-generation SFS Turbo file system, the capacity ranges from **500** to **32768**, in GiB. • For a previous-generation SFS Turbo file system with **expand_type="bandwidth "** configured under **metadata**, the capacity ranges from **10240** to **327680**, in GiB. • For a 20 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="20M"** configured under **metadata**, the capacity ranges from **3686* to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system. • For a 40 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="40M"** configured under **metadata**, the capacity ranges from **1228* to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system. • For a 125 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="125M"**

Parameter	Mandatory	Type	Description
			configured under **metadata**, the capacity ranges from **1228* to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system. • For a 250 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="250M"** configured under **metadata**, the capacity ranges from **1228* to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system. • For a 500 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="500M"** configured under **metadata**, the capacity ranges from **1228** to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system.

Parameter	Mandatory	Type	Description
			• For a 1,000 MB/s/TiB file system with **expand_type="hpc"** and **hpc_bw="1000M"** configured under **metadata**, the capacity ranges from **1228** to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, specify 3686 GiB for a 3.6 TiB file system, 4915 GiB for a 4.8 TiB file system, and 8601 GiB for an 8.4 TiB file system.
subnet_id	Yes	String	The subnet ID of a tenant in a VPC.
vpc_id	Yes	String	The VPC ID of a tenant in a region.
backup_id	No	String	The backup ID. This parameter is mandatory if you create a file system from a backup.
tags	No	Array of **ResourceTag** objects	The tag list.

Table 5-5 Metadata

Parameter	Mandatory	Type	Description
crypt_key_id	No	String	The ID of a KMS professional key, which is used to create an encrypted file system.
dedicated_flavor	No	String	The VM flavor, which is used to create a dedicated file system.
dedicated_storage_id	No	String	The ID of the dedicated distributed storage, which is used to create a dedicated file system.

Parameter	Mandatory	Type	Description
expand_type	No	String	The extension type. This parameter is not returned when the file system is being created. • This parameter is mandatory when you are creating an SFS Turbo 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, 20 MB/s/TiB, or Enhanced file system. • Specify **bandwidth** when you are creating a Standard-Enhanced or Performance-Enhanced file system. • Specify **hpc** when you are creating a 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, or 20 MB/s/TiB file system.
hpc_bw	No	String	The file system bandwidth. This parameter is mandatory when you are creating an SFS Turbo 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, or 20 MB/s/TiB file system. Specify **20M** for a 20 MB/s/TiB file system, **40M** for a 40 MB/s/TiB file system, **125M** for a 125 MB/s/TiB file system, and **250M** for a 250 MB/s/TiB file system. Specify **500M** for a 500 MB/s/TiB file system and **1000M** for a 1,000 MB/s/TiB file system.

Table 5-6 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).
value	Yes	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Table 5-7 BssInfo

Parameter	Mandatory	Type	Description
is_auto_renew	No	Long	Whether to enable auto-renewal. **0** disables auto-renewal. **1** enables auto-renewal. Enumeration values: • **0** • **1**
period_num	Yes	Long	The yearly/monthly subscription terms.

Parameter	Mandatory	Type	Description
period_type	Yes	Long	The yearly/monthly subscription type. The value can be **2** (monthly subscription) or **3** (yearly subscription). Enumeration values: • **2** • **3**
is_auto_pay	No	Long	Whether to automatically pay for the order after it is created. The value can be **0** (does not pay automatically) or **1** (pays automatically). Enumeration values: • **0** • **1**

Response Parameters

Status code: 202

Table 5-8 Response body parameters

Parameter	Type	Description
id	String	The ID of the created SFS Turbo file system.
name	String	The name of the created SFS Turbo file system.
status	String	The status of the created SFS Turbo file system.

Example Requests

- Previous-generation SFS Turbo file system type:

```
POST HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares

{
  "share" : {
    "name" : "sfs-turbo-test",
    "availability_zone" : "example",
    "security_group_id" : "8c4ebbd0-6edf-4aae-8353-xxx",
    "share_proto" : "NFS",
    "share_type" : "STANDARD",
    "size" : 500,
    "subnet_id" : "b8884abe-f47b-4917-9f6c-xxx",
```



```
    "vpc_id" : "d651ea2b-2b20-4c6d-8bbf-xxx"
  }
}
```

- Previous-generation SFS Turbo file system type in a dedicated scenario:

```
POST HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares

{
  "share" : {
    "name" : "sfs-turbo-dedicated-test",
    "availability_zone" : "example",
    "security_group_id" : "8c4ebbd0-6edf-4aae-8353-xxx",
    "share_proto" : "NFS",
    "share_type" : "STANDARD",
    "size" : 500,
    "subnet_id" : "b8884abe-f47b-4917-9f6c-xxx",
    "vpc_id" : "d651ea2b-2b20-4c6d-8bbf-xxx",
    "metadata" : {
      "dedicated_flavor" : "c6.xlarge.2",
      "dedicated_storage_id" : "198f0704-xxx-4d85-xxx-c25caa4d3264"
    }
  }
}
```

- 125 MB/s/TiB:

```
POST HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares

{
  "share" : {
    "name" : "sfs-turbo-test",
    "availability_zone" : "example",
    "security_group_id" : "8c4ebbd0-6edf-4aae-8353-xxx",
    "share_proto" : "NFS",
    "share_type" : "STANDARD",
    "size" : 3686,
    "subnet_id" : "b8884abe-f47b-4917-9f6c-xxx",
    "vpc_id" : "d651ea2b-2b20-4c6d-8bbf-xxx",
    "metadata" : {
      "expand_type" : "hpc",
      "hpc_bw" : "125M"
    }
  }
}
```

This example creates an SFS Turbo Standard file system in the AZ whose AZ code is **example**, with the file system name set to **sfs-turbo-test**, protocol type to NFS, capacity to 500 GB. The security group ID is **8c4ebbd0-6edf-4aae-8353-xxx**, the subnet ID is **b8884abe-f47b-4917-9f6c-xxx**, and the VPC ID is **d651ea2b-2b20-4c6d-8bbf-xxx**.

This example creates an SFS Turbo Standard file system in the AZ whose AZ code is **example**, with the file system name set to **sfs-turbo-dedicated-test**, protocol type to NFS, capacity to 500 GB. The dedicated storage pool ID is **198f0704-xxx-4d85-xxx-c25caa4d3264**, the dedicated ECS flavor is **c6.xlarge.2**, the security group ID is **8c4ebbd0-6edf-4aae-8353-xxx**, the subnet ID is **b8884abe-f47b-4917-9f6c-xxx**, and the VPC ID is **d651ea2b-2b20-4c6d-8bbf-xxx**.

This example creates an SFS Turbo 125 MB/s/TiB file system in the AZ whose AZ code is **example**, with the file system name set to **sfs-turbo-test**, protocol type to NFS, capacity to 3686 GB. The security group ID is **8c4ebbd0-6edf-4aae-8353-xxx**, the subnet ID is **b8884abe-f47b-4917-9f6c-xxx**, and the VPC ID is **d651ea2b-2b20-4c6d-8bbf-xxx**.

This example creates a 125 MB/s/TiB SFS Turbo file system in the AZ whose AZ code is **example**, with the file system name set to **sfs-turbo-test**, protocol type to

NFS, capacity to **3686GB**. The security group ID is **8c4ebbd0-6edf-4aae-8353-xxx**, the subnet ID is **b8884abe-f47b-4917-9f6c-xxx**, and the VPC ID is **d651ea2b-2b20-4c6d-8bbf-xxx**.

Example Responses

Status code: 202

Successful file system creation

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

Status Codes

Status Code	Description
202	Successful file system creation

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 5-9 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-10 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-11 Response body parameters

Parameter	Type	Description
action_progress	ActionProgress object	The file system creation progress. This field is only returned when the file system is being created.
version	String	The file system version.
avail_capacity	String	The available capacity of the file system, in GB.
availability_zone	String	The code of the AZ where the file system resides.
az_name	String	The name of the AZ where the file system resides.
created_at	String	The time when the file system was created. UTC time, for example: 2018-11-19T04:02:03.
crypt_key_id	String	The ID of the encryption key. This parameter is not returned for non-encrypted file systems.
expand_type	String	For an Enhanced file system, **bandwidth** is returned. For a 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, or 20 MB/s/TiB file system, **hpc** is returned. For other types of file systems, this field is not returned.
export_location	String	The file system location, for example, **192.168.0.90:/**. This field is not returned when the file system is being created.

Parameter	Type	Description
id	String	The file system ID.
name	String	The file system name specified during creation.
pay_model	String	The file system billing mode. Value **0** indicates pay-per-use. This field is not returned when the file system is being created. Enumeration values: • **0** • **1**
region	String	The region where the file system resides.
security_group_id	String	The ID of the specified security group.
share_proto	String	The protocol used by the SFS Turbo file system. The value can be **NFS** or **CIFS**.
share_type	String	The file system performance type. Valid values are **STANDARD** and **PERFORMANCE**.
size	String	The file system total capacity, in GB.
status	String	The file system status. The value can be as follows: **100** (creating), **200** (available), **303** (creation failed), and **800** (frozen)

Parameter	Type	Description
sub_status	String	The substatus of the SFS Turbo file system. The value can be as follows: This field is not returned if no modification is made to the file system. **121** (expanding capacity), **132** (changing security group), **137** (adding authorized VPC), **138** (removing authorized VPC), *150** (adding storage backend), **151** (removing storage backend) **221** (expansion succeeded), **232** (security group changed), **237** (authorized VPC added), **238** (authorized VPC removed), **250** (storage backend added), *251** (storage backend removed) **321** (expansion failed), **332** (changing security group failed), **337** (adding authorized VPC failed), **338** (removing authorized VPC failed), **350** (adding storage backend failed), **351** (removing storage backend failed)
subnet_id	String	The ID of the specified subnet.
vpc_id	String	The ID of the specified VPC.
enterprise_project_id	String	The ID of the enterprise project that file system is added to.
tags	Array of **ResourceTag** objects	The tag list.
optional_endpoint	String	The alternative IP addresses that can be used for mount. This field is not returned for previous-generation file systems.
hpc_bw	String	The file system bandwidth. • "20M": 20 MB/s/TiB • "40M": 40 MB/s/TiB • "125M": 125 MB/s/TiB • "250M": 250 MB/s/TiB • "500M": 500 MB/s/TiB • "1000M": 1,000 MB/s/TiB

Parameter	Type	Description
instanceId	String	The node ID of the file system type, which has no meaning. This is a reserved field.
instanceType	String	The node type of the file system type, which has no meaning. This is a reserved field.
statusDetail	String	The file system request ID, which has no meaning. This is a reserved field.
features	ShareInfoFeatures object	The file system feature settings.

Table 5-12 ActionProgress

Parameter	Type	Description
CREATING	String	The file system creation progress.

Table 5-13 ResourceTag

Parameter	Type	Description
key	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).

Parameter	Type	Description
value	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Table 5-14 ShareInfoFeatures

Parameter	Type	Description
backup	ShareInfoFeature object	Whether the file system supports backup.

Table 5-15 ShareInfoFeature

Parameter	Type	Description
is_support	Boolean	Whether the file system supports this feature.
message	String	The message.
msg_code	String	The message code.

Example Requests

Querying the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
GET HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/77ba6f4b-6365-4895-8dda-bc7142af4dde
```

Example Responses

Status code: 200

Query response body

- Listing file systems in the project whose ID is **e1e45b08f3ea4480ab4655ef9c7160ba**

```
{
  "id" : "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
  "name" : "sfs-turbo-8468",
```



```
  "status" : "200",
  "version" : "1.0.0",
  "region" : "example",
  "availability_zone" : "example",
  "az_name" : "example",
  "created_at" : "2018-11-19T04:02:03",
  "export_location" : "192.168.xx.xx:/",
  "action_progress" : { },
  "share_type" : "STANDARD",
  "sub_status" : "221",
  "vpc_id" : "b24e39e1-bc0c-475b-ae0c-aef9cf240af3",
  "subnet_id" : "86fc01ea-8ec8-409d-ba7a-e0ea16d4fd97",
  "security_group_id" : "50586458-aec9-442c-bb13-e08ddc6f1b7a",
  "size" : "600.00",
  "avail_capacity" : "600.00",
  "pay_model" : "0",
  "share_proto" : "NFS"
}
```


Status Codes

Status Code	Description
200	Query response body

Error Codes

See **Error Codes**.

5.1.3 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 5-16 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-17 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 202

File system deletion request delivered

None

Example Requests

Deleting the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
DELETE HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/77ba6f4b-6365-4895-8dda-bc7142af4dde
```

Example Responses

None

Status Codes

Status Code	Description
202	File system deletion request delivered

Error Codes

See **Error Codes**.

5.1.4 Listing File Systems

Function

This API is used to list file systems.

URI

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

Table 5-18 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Table 5-19 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	The maximum number of returned file systems. If not specified, **1000** is used by default.
offset	No	Integer	The offset of the returned file systems.

Request Parameters

Table 5-20 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-21 Response body parameters

Parameter	Type	Description
shares	Array of **ShareInfo** objects	The list of SFS Turbo file systems.
count	Integer	The number of SFS Turbo file systems.

Table 5-22 ShareInfo

Parameter	Type	Description
action_progress	**ActionProgress** object	The file system creation progress. This field is only returned when the file system is being created.
version	String	The file system version.
avail_capacity	String	The available capacity of the file system, in GB.
availability_zone	String	The code of the AZ where the file system resides.
az_name	String	The name of the AZ where the file system resides.
created_at	String	The time when the file system was created. UTC time, for example: 2018-11-19T04:02:03.
crypt_key_id	String	The ID of the encryption key. This parameter is not returned for non-encrypted file systems.
expand_type	String	For an Enhanced file system, **bandwidth** is returned. For a 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB, 125 MB/s/TiB, 40 MB/s/TiB, or 20 MB/s/TiB file system, **hpc** is returned. For other types of file systems, this field is not returned.
export_location	String	The file system location, for example, **192.168.0.90:/**. This field is not returned when the file system is being created.
id	String	The file system ID.
name	String	The file system name specified during creation.
pay_model	String	The file system billing mode. Value **0** indicates pay-per-use. This field is not returned when the file system is being created. Enumeration values: • **0** • **1**
region	String	The region where the file system resides.

Parameter	Type	Description
security_group_id	String	The ID of the specified security group.
share_proto	String	The protocol used by the SFS Turbo file system. The value can be **NFS** or **CIFS**.
share_type	String	The file system performance type. Valid values are **STANDARD** and **PERFORMANCE**.
size	String	The file system total capacity, in GB.
status	String	The file system status. The value can be as follows: **100** (creating), **200** (available), **303** (creation failed), and **800** (frozen)
sub_status	String	The substatus of the SFS Turbo file system. The value can be as follows: This field is not returned if no modification is made to the file system. **121** (expanding capacity), **132** (changing security group), **137** (adding authorized VPC), **138** (removing authorized VPC), *150** (adding storage backend), **151** (removing storage backend) **221** (expansion succeeded), **232** (security group changed), **237** (authorized VPC added), **238** (authorized VPC removed), **250** (storage backend added), *251** (storage backend removed) **321** (expansion failed), **332** (changing security group failed), **337** (adding authorized VPC failed), **338** (removing authorized VPC failed), **350** (adding storage backend failed), **351** (removing storage backend failed)
subnet_id	String	The ID of the specified subnet.
vpc_id	String	The ID of the specified VPC.
enterprise_project_id	String	The ID of the enterprise project that file system is added to.
tags	Array of **ResourceTag** objects	The tag list.

Parameter	Type	Description
optional_endpoint	String	The alternative IP addresses that can be used for mount. This field is not returned for previous-generation file systems.
hpc_bw	String	The file system bandwidth. • "20M": 20 MB/s/TiB • "40M": 40 MB/s/TiB • "125M": 125 MB/s/TiB • "250M": 250 MB/s/TiB • "500M": 500 MB/s/TiB • "1000M": 1,000 MB/s/TiB
instanceId	String	The node ID of the file system type, which has no meaning. This is a reserved field.
instanceType	String	The node type of the file system type, which has no meaning. This is a reserved field.
statusDetail	String	The file system request ID, which has no meaning. This is a reserved field.
features	**ShareInfoFeatures** object	The file system feature settings.

Table 5-23 ActionProgress

Parameter	Type	Description
CREATING	String	The file system creation progress.

Table 5-24 ResourceTag

Parameter	Type	Description
key	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).
value	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Table 5-25 ShareInfoFeatures

Parameter	Type	Description
backup	**ShareInfoFeature** object	Whether the file system supports backup.

Table 5-26 ShareInfoFeature

Parameter	Type	Description
is_support	Boolean	Whether the file system supports this feature.
message	String	The message.
msg_code	String	The message code.

Example Requests

Listing file systems in the project whose ID is **e1e45b08f3ea4480ab4655ef9c7160ba**

```
GET HTTPS://{endpoint}/v1/e1e45b08f3ea4480ab4655ef9c7160ba/sfs-turbo/shares/detail
```

Example Responses

Status code: 200

Success response body

- Listing file systems in the project whose ID is **e1e45b08f3ea4480ab4655ef9c7160ba**

```
{
  "shares" : [ {
    "id" : "8fba8253-c914-439d-ae8b-d5c89d0bf5e8",
    "name" : "sfs-turbo-8468",
    "status" : "100",
    "version" : "1.0.0",
    "region" : "example",
    "created_at" : "2018-11-19T04:02:03",
    "action_progress" : {
      "CREATING" : "22%"
    },
    "share_type" : "STANDARD",
    "sub_status" : "",
    "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"
  } ]
}
```

- Listing file systems in the project whose ID is **5694xxx8f3e0000xxxx655exxxxx60b7**

```
{
  "shares" : [ {
    "id" : "65f2d30b-7b4e-4786-9608-4324faef6646",
    "name" : "sfs-turbo-df12",
    "status" : "200",
    "version" : "1.0.0",
    "region" : "example",
    "created_at" : "2018-11-15T02:32:10",
    "export_location" : "192.168.xx.xx:/",
    "optional_endpoint" : "192.168.xx.xx 192.168.xx.xx",
    "share_type" : "HPC_PERFORMANCE_250M",
    "expand_type" : "hpc",
    "sub_status" : "",
    "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" : "3686.00",
    "pay_model" : "0",
    "avail_capacity" : "3686.00",
    "share_proto" : "NFS"
  } ]
}
```


Status Codes

Status Code	Description
200	Success response body

Error Codes

See **Error Codes**.

5.1.5 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 5-27 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-28 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-29 Request body parameters

Parameter	Mandatory	Type	Description
extend	Yes	**Extend** object	The **extend** object.

Table 5-30 Extend

Parameter	Mandatory	Type	Description
new_size	Yes	Integer	The new capacity of the file system, in GiB. For a previous-generation Standard or Performance file system, the capacity ranges from **500** to **32768** (in GiB), and the minimum expansion increment is 100 GiB. For a previous-generation Standard-Enhanced or Performance-Enhanced file system, the capacity ranges from **10240** to **327680** (in GiB), and the minimum expansion increment is 100 GiB. For a 20 MB/s/TiB file system, the capacity ranges from **3686** to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, use 4915 GiB for a 4.8 TiB file system and 8601 GiB for an 8.4 TiB file system. The minimum expansion increment is 1.2 TiB. For a 40 MB/s/TiB file system, the capacity ranges from **1228** to **1048576** (in GiB) and must be a multiple of 1.2 TiB. The desired capacity must be converted to GiB and rounded down to the nearest integer. For example, use 4915 GiB for a 4.8 TiB file system and 8601 GiB for an 8.4 TiB file system. The minimum expansion increment is 1.2 TiB. The capacity range and minimum expansion increment of 1,000 MB/s/TiB, 500 MB/s/TiB, 250 MB/s/TiB and 125 MB/s/TiB file systems are the same as those of 40 MB/s/TiB file systems.

Parameter	Mandatory	Type	Description
new_bandwidth	No	Long	The new bandwidth, in GB. Only HPC Cache file systems support bandwidth change. The supported bandwidth values are **2**, **4**, **8**, **16**, **24**, **32**, and **48**.
bss_param	No	**BssInfoExtend** object	The billing parameter of expanding a yearly/monthly file system.

Table 5-31 BssInfoExtend

Parameter	Mandatory	Type	Description
is_auto_pay	No	Long	Whether to enable automatic payment. Enumeration values: • **0** • **1**

Response Parameters

Status code: 202

Table 5-32 Response body parameters

Parameter	Type	Description
id	String	The SFS Turbo file system ID.
name	String	The SFS Turbo file system name.

Example Requests

Expanding the capacity of a file system to 1,000 GB

```
{
  "extend" : {
    "new_size" : 1000
  }
}
```

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**.

5.2 Connection Management

5.2.1 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. Changing the security group is an asynchronous task. You can call the API for querying details of a file system and view the value of **sub_status** returned to check whether the security group change is successful. If value **232** is returned, the security group has been changed.

URI

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

Table 5-33 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-34 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-35 Request body parameters

Parameter	Mandatory	Type	Description
change_security_group	Yes	**ChangeSecurityGroup** object	The **change_security_group** object.

Table 5-36 ChangeSecurityGroup

Parameter	Mandatory	Type	Description
security_group_id	Yes	String	The ID of the new security group.

Response Parameters

Status code: 202

Table 5-37 Response body parameters

Parameter	Type	Description
id	String	The SFS Turbo file system ID.

Example Requests

Changing the security group of a file system (new security group ID **26f6b565-240e-43c3-8867-03f0bd975433**)

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

Example Responses

Status code: 202

The SFS Turbo file system ID.

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

Status Codes

Status Code	Description
202	The SFS Turbo file system ID.

Error Codes

See **Error Codes**.

5.3 Tag Management

5.3.1 Adding a Tag to a File System

Function

This API is used to add a tag to a file system.

A maximum of 20 tags can be added to a 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 you want to add, the tag will be updated.

URI

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

Table 5-38 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-39 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-40 Request body parameters

Parameter	Mandatory	Type	Description
tag	Yes	**ResourceTag** object	The **resource_tag** field description.

Table 5-41 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).
value	Yes	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Response Parameters

Status code: 204

Tag adding request delivered

None

Example Requests

Creating a file system tag, with tag value set to **key1** and tag key **value1**

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

Example Responses

None

Status Codes

Status Code	Description
204	Tag adding request delivered

Error Codes

See **Error Codes**.

5.3.2 Querying All Tags of a File System

Function

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

URI

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

Table 5-42 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-43 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-44 Response body parameters

Parameter	Type	Description
tags	Array of **ResourceTag** objects	The tag list.

Table 5-45 ResourceTag

Parameter	Type	Description
key	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).

Parameter	Type	Description
value	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Example Requests

Querying tags of the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
GET HTTPS://{endpoint}/v1/v1/{project_id}/sfs-turbo/77ba6f4b-6365-4895-8dda-bc7142af4dde/tags
```

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**.

5.3.3 Deleting a Tag from a File System

Function

This API is used to delete a tag from a file system. If the specified key is not found, error 404 will be returned.

URI

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

Table 5-46 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.
key	Yes	String	The tag key, which can contain a maximum of 128 characters. It cannot be left empty 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 5-47 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 204

Tag deleted

None

Example Requests

Deleting a tag (tag key is **test**) from the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
DELETE HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/77ba6f4b-6365-4895-8dda-bc7142af4dde/tags/test
```

Example Responses

None

Status Codes

Status Code	Description
204	Tag deleted

Error Codes

See **Error Codes**.

5.3.4 Batch Adding Tags to a File System

Function

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

A maximum of 20 tags can be added to a 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 you want to add, the tag will be updated.

URI

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

Table 5-48 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Parameter	Mandatory	Type	Description
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-49 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-50 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	The operation identifier. The value is **create**. Use **create** if you want to batch add tags to a file system. Enumeration values: • **create**
tags	Yes	Array of **ResourceTag** objects	The list of tags.

Table 5-51 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).

Parameter	Mandatory	Type	Description
value	Yes	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Response Parameters

Status code: 204

Batch adding tags succeeded

None

Example Requests

Batch adding tags to a file system, with tag key of the first tag set to **key1**, tag value of the first tag **value1**, tag key of the second tag **key2**, and tag value of the second tag **value1**

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

Example Responses

None

Status Codes

Status Code	Description
204	Batch adding tags succeeded

Error Codes

See **Error Codes**.

5.3.5 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 5-52 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Table 5-53 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Integer	The maximum number of tags that can be returned.
offset	No	Integer	The tag query offset.

Request Parameters

Table 5-54 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-55 Response body parameters

Parameter	Type	Description
tags	Array of **Tag** objects	The tag list.

Table 5-56 Tag

Parameter	Type	Description
key	String	The tag key. A key can contain a maximum of 128 characters and cannot be left blank.
values	Array of strings	The list the tag values. Each value can contain a maximum of 255 characters. An empty list for **values** indicates any value. The values are in the OR relationship.

Example Requests

Querying tags of all file systems in the project whose ID is **e1e45b08f3ea4480ab4655ef9c7160ba**

```
GET HTTPS://{endpoint}/v1/e1e45b08f3ea4480ab4655ef9c7160ba/sfs-turbo/tags
```

Example Responses

Status code: 200

Query response body

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

Status Codes

Status Code	Description
200	Query response body

Error Codes

See **Error Codes**.

5.3.6 Querying File Systems by Tag

Function

This API is used to query file systems by tag.

URI

POST /v1/{project_id}/sfs-turbo/resource_instances/action

Table 5-57 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Request Parameters

Table 5-58 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-59 Request body parameters

Parameter	Mandatory	Type	Description
action	Yes	String	The operation type of listing file systems by tag. The value can be **filter** or **count**. Enumeration values: • **filter** • **count**
limit	No	String	The maximum number of file systems returned.
offset	No	String	The offset of the returned file systems.

Parameter	Mandatory	Type	Description
without_any_tag	No	Boolean	The resources to be queried contain no tags. If this parameter is set to **true**, all resources without specified tags are queried. In this case, the **tags** field is ignored. If this parameter is set to **false** or not specified, it does not take effect, meaning that all resources are returned or resources are filtered by **tags** or **matches**.
tags	No	Array of **Tag** objects	The resources to be queried contain tags specified in this field. Each resource can contain a maximum of 20 keys. Each tag key can have a maximum of 20 tag values. The tag value corresponding to each tag key can be an empty array but the structure cannot be missing. Each tag key must be unique, and tag values of the same tag must be unique. The response returns resources containing all tags in this list. Keys in this list are in the AND relationship and values in each key-value structure are in the OR relationship. If no tag filtering criteria is specified, all data is returned.
matches	No	Array of **ResourceTag** objects	The search criteria. **key** is the field to match. Only **resource_name** is supported. **value** is the matched value. If the value ends with *, prefix search will be performed. For example, if the value is sfsturbo*, all resources whose names start with **sfsturbo** will be returned.

Table 5-60 Tag

Parameter	Mandatory	Type	Description
key	Yes	String	The tag key. A key can contain a maximum of 128 characters and cannot be left blank.
values	Yes	Array of strings	The list the tag values. Each value can contain a maximum of 255 characters. An empty list for **values** indicates any value. The values are in the OR relationship.

Table 5-61 ResourceTag

Parameter	Mandatory	Type	Description
key	Yes	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).
value	Yes	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Response Parameters

Status code: 200

Table 5-62 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-63 Response body parameters

Parameter	Type	Description
resources	Array of **ListSharesByTagResource** objects	The list of file systems queried by tag.
total_count	Integer	The total number of file systems returned.

Table 5-64 ListSharesByTagResource

Parameter	Type	Description
resource_id	String	The resource ID.
resource_name	String	The resource name.
resource_detail	String	The resource details.
tags	Array of **ResourceTag** objects	The resource tag list.

Table 5-65 ResourceTag

Parameter	Type	Description
key	String	The tag key. It can contain a maximum of 128 characters. It cannot be left empty 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 (_).
value	String	The tag value. Each tag value can contain a maximum of 255 characters and can be an empty string. 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 (_).

Status code: 400

Table 5-66 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-67 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-68 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-69 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

- Filtering file systems by file system tag

```
{
  "action" : "filter",
  "limit" : 10,
  "offset" : 10,
  "tags" : [ {
    "key" : "key1",
    "values" : [ "value1", "value2" ]
  }, {
    "key" : "key2",
    "values" : [ "value1", "value2" ]
  } ]
}
```

- Filtering file systems by file system name

```
{
  "action" : "filter",
  "matches" : [ {
    "key" : "resource_name",
    "value" : "sfsturbo*"
  } ]
}
```

Example Responses

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "parameter error"
}
```

Status code: 500

The error response.

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "internal server error"
}
```


Status Codes

Status Code	Description
200	Success
400	Error response
500	The error response.

Error Codes

See **Error Codes**.

5.4 Name Management

5.4.1 Changing the Name of a File System

Function

This API is used to change the name of a file system.

URI

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

Table 5-70 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-71 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-72 Request body parameters

Parameter	Mandatory	Type	Description
change_name	Yes	ShareName object	The file system to be modified.

Table 5-73 ShareName

Parameter	Mandatory	Type	Description
name	Yes	String	The name of the file system to be modified.

Response Parameters

Status code: 204

Successful request

None

Example Requests

Changing the name of an SFS Turbo file system to **sfs-turbo-test1**

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

Example Responses

None

Status Codes

Status Code	Description
204	Successful request
400	Incorrect parameter
409	File system name already exists
500	Internal error

Error Codes

See **Error Codes**.

5.5 File System Management

5.5.1 Obtaining IP Addresses of the Clients Who Have Mounted the File System

Function

This API is used to obtain the IP addresses of the clients who have mounted the file system.

URI

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

Table 5-74 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-75 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-76 Request body parameters

Parameter	Mandatory	Type	Description
get_client_ips	Yes	**ClientIpInfo** object	Obtains the IP addresses of the clients who have mounted the file system.

Table 5-77 ClientIpInfo

Parameter	Mandatory	Type	Description
ips	No	String	The IP addresses of the clients who have mounted the file system.

Response Parameters

Status code: 200

Table 5-78 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-79 Response body parameters

Parameter	Type	Description
id	String	The file system ID.
ips	Array of strings	The IP addresses of the clients who have mounted the file system.

Status code: 400

Table 5-80 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-81 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-82 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-83 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

None

Example Responses

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "parameter error"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Success
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6 Permissions Management

5.6.1 Joining an AD Domain

Function

This API is used to add a file system to an AD domain. Active Directory Domain Services (AD DS) is a directory service that provides unified identity and permissions management through domain networks. By adding an SFS Turbo file system to an AD domain, you can easily authenticate and manage user identities and control file permissions in that domain.

Constraints

This API is only supported for SMB file systems.

URI

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

Table 5-84 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-85 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-86 Request body parameters

Parameter	Mandatory	Type	Description
service_account	Yes	String	The service account, which is specified when the domain server is created. **administrator** is used normally.
password	Yes	String	The password of the service account.
domain_name	Yes	String	The domain name of the domain controller. It is specified when the domain server is created.
system_name	Yes	String	The name of the file system in the AD domain.
overwrite_same_account	No	Boolean	If the option is enabled and the domain controller already has the file system name you specified, the information you specified will overwrite the existing information in the domain controller.
dns_server	Yes	Array of strings	The IP address of the DNS server. It is used to resolve the AD domain name.
organization_unit	No	String	A group of domain objects, such as users, computers, and printers. If you add the file system to an organizational unit (OU), it will become a member of that OU. If this parameter is left blank, the file system will be added to the **computers** OU.
vpc_id	No	String	The VPC ID.

Response Parameters

Status code: 202

Table 5-87 Response body parameters

Parameter	Type	Description
job_id	String	The ID of the asynchronous task for joining the AD domain.

Status code: 400

Table 5-88 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-89 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-90 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-91 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

Joining an AD domain

```
POST HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

{
  "service_account" : "administrator",
```



```
  "password" : "pwdxxxxx",
  "domain_name" : "SFSTURBO.COM",
  "system_name" : "sfs",
  "dns_server" : "100.xxx:xxx"
}
```

Example Responses

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "parameter domain_name is required and cannot be empty"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
202	Join request delivered
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.2 Querying the AD Domain Configuration

Function

This API is used to query the AD domain configuration. AD DS is a directory service that provides unified identity and permissions management through domain networks. By adding an SFS Turbo file system to an AD domain, you can easily authenticate and manage user identities and control file permissions in that domain.

Constraints

This API is only supported for SMB file systems.

URI

GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

Table 5-92 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-93 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-94 Response body parameters

Parameter	Type	Description
domain_name	String	The domain name of the domain controller. It is specified when the domain server is created.
system_name	String	The name of the file system in the AD domain.
dns_server	Array of strings	The IP address of the DNS server. It is used to resolve the AD domain name.
organization_unit	String	A group of domain objects, such as users, computers, and printers. If you add the file system to an OU, it will become a member of that OU. If this parameter is left blank, the file system will be added to the **computers** OU.
vpc_id	String	The VPC ID.

Parameter	Type	Description
status	String	The current status of the AD domain. Enumeration values: • **JOINING** • **AVAILABLE** • **EXITING** • **FAILED**

Status code: 500

Table 5-95 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-96 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

None

Example Responses

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Successful query
500	Error response

Error Codes

See **Error Codes**.

5.6.3 Modifying the AD Domain Configuration

Function

This API is used to modify the AD domain configuration. AD DS is a directory service that provides unified identity and permissions management through domain networks. By adding an SFS Turbo file system to an AD domain, you can easily authenticate and manage user identities and control file permissions in that domain.

Constraints

This API is only supported for SMB file systems.

URI

PUT /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

Table 5-97 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-98 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-99 Request body parameters

Parameter	Mandatory	Type	Description
service_account	Yes	String	The service account, which is specified when the domain server is created. **administrator** is used normally.
password	Yes	String	The password of the service account.
domain_name	Yes	String	The domain name of the domain controller. It is specified when the domain server is created.
system_name	Yes	String	The name of the file system in the AD domain.
overwrite_same_account	No	Boolean	If the option is enabled and the domain controller already has the file system name you specified, the information you specified will overwrite the existing information in the domain controller.
dns_server	Yes	Array of strings	The IP address of the DNS server. It is used to resolve the AD domain name.
organization_unit	No	String	A group of domain objects, such as users, computers, and printers. If you add the file system to an OU, it will become a member of that OU. If this parameter is left blank, the file system will be added to the **computers** OU.
vpc_id	No	String	The VPC ID.

Response Parameters

Status code: 202

Table 5-100 Response body parameters

Parameter	Type	Description
job_id	String	The ID of the asynchronous task for modifying the AD domain.

Status code: 400

Table 5-101 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-102 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-103 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-104 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

```
PUT HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

{
  "service_account" : "administrator",
  "password" : "pwdxxxxx",
  "domain_name" : "SFSTURBO.COM",
  "system_name" : "sfs",
```



```
  "dns_server" : "100.xxx:xxx"
}
```

Example Responses

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "Incorrect account or password."
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
202	Modification request delivered
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.4 Leaving the AD Domain

Function

This API is used to remove a file system from an AD domain. AD DS is a directory service that provides unified identity and permissions management through domain networks. By adding an SFS Turbo file system to an AD domain, you can easily authenticate and manage user identities and control file permissions in that domain.

Constraints

This API is only supported for SMB file systems.

URI

DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain

Table 5-105 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-106 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-107 Request body parameters

Parameter	Mandatory	Type	Description
service_account	Yes	String	The service account, which is specified when the domain server is created. **administrator** is used normally.
password	Yes	String	The password of the service account.

Response Parameters

Status code: 202

Table 5-108 Response body parameters

Parameter	Type	Description
job_id	String	The ID of the asynchronous task for leaving the AD domain.

Status code: 400

Table 5-109 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-110 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-111 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-112 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

None

Example Responses

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "Invalid rule id"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
```



```
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
202	AD domain left successfully
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.5 Creating a Permission Rule

Function

This API is used to create a permission rule.

Constraints

A maximum of 64 permission rules can be added for a file system.

This API is only supported for NFS file systems.

URI

POST /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules

Table 5-113 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Request Parameters

Table 5-114 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.

Parameter	Mandatory	Type	Description
Content-Type	Yes	String	The MIME type.

Table 5-115 Request body parameters

Parameter	Mandatory	Type	Description
rules	Yes	Array of **OnePermRuleRequestInfo** objects	The permission rule details. You can add a maximum of five rules at a time.

Table 5-116 OnePermRuleRequestInfo

Parameter	Mandatory	Type	Description
ip_cidr	Yes	String	The IP address or IP address range of the object to be authorized. Once configured, this parameter cannot be modified.
rw_type	Yes	String	The read/write permission of the object to be authorized. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission

Parameter	Mandatory	Type	Description
user_type	Yes	String	The file system access permission granted to the user of the object to be authorized. The value can be: • **no_root_squash**: allows any user including root on the client to access the file system as who they are, instead of mapping them to another user. • **root_squash**: allows root on the client to access the file system as **nfsnobody**. Client access using a non-root user will be retained as who they are, instead of being mapped to another user. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**.

Response Parameters

Status code: 200

Table 5-117 Response body parameters

Parameter	Type	Description
rules	Array of **OnePermRuleResponseInfo** objects	The permission rule information.

Table 5-118 OnePermRuleResponseInfo

Parameter	Type	Description
id	String	The permission rule ID.
ip_cidr	String	The IP address or IP address range of the authorized object.

Parameter	Type	Description
rw_type	String	The read/write permission of the authorized object. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission
user_type	String	The file system access permission granted to the user of the authorized object. The value can be: • **no_root_squash**: allows root on the client to access the file system as **root**. • **root_squash**: allows root on the client to access the file system as **nfsnobody**. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**. This is the default value.

Status code: 400

Table 5-119 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-120 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

- Request example for creating permission rules

```
{
  "rules" : [ {
```



```
    "ip_cidr" : "192.168.xx.xx/16",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  }, {
    "ip_cidr" : "192.32.xx.xx/16",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  } ]
}
```

Example Responses

Status code: 200

Successful creation

- Response example for creating permission rules

```
{
  "rules" : [ {
    "id" : "1131ed520xxxxxxebedb6e57xxxxxxxx",
    "ip_cidr" : "192.32.0.0/16",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  }, {
    "id" : "1131ed520xxxxxxebedb6e57xxxxxxxx",
    "ip_cidr" : "192.32.0.1",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  } ]
}
```

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "Rules not allowed empty"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Successful creation
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.6 Querying Permission Rules of a File System

Function

This API is used to query permission rules of a file system.

Constraints

A maximum of 64 permission rules can be added for a file system.

This API is only supported for NFS file systems.

URI

GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules

Table 5-121 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.

Table 5-122 Query Parameters

Parameter	Mandatory	Type	Description
limit	No	Long	The maximum number of permission rules that can be returned.
offset	No	Long	The offset of the returned permission rules.

Request Parameters

Table 5-123 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-124 Response body parameters

Parameter	Type	Description
rules	Array of **OnePermRuleResponseInfo** objects	The permission information.

Table 5-125 OnePermRuleResponseInfo

Parameter	Type	Description
id	String	The permission rule ID.
ip_cidr	String	The IP address or IP address range of the authorized object.
rw_type	String	The read/write permission of the authorized object. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission
user_type	String	The file system access permission granted to the user of the authorized object. The value can be: • **no_root_squash**: allows root on the client to access the file system as **root**. • **root_squash**: allows root on the client to access the file system as **nfsnobody**. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**. This is the default value.

Status code: 500

Table 5-126 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

Querying the permission rules of the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
GET HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/77ba6f4b-6365-4895-8dda-bc7142af4dde/fs/perm-rules
```

Example Responses

Status code: 200

Successful query

- Response example of querying the permission rules of a file system

```
{
  "rules" : [ {
    "id" : "1131ed520xxxxxxebedb6e57xxxxxxxx",
    "ip_cidr" : "192.168.xx.xx/16",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  }, {
    "id" : "1231ed520xxxxxxebedb6e57xxxxxxxx",
    "ip_cidr" : "192.32.xx.xx/16",
    "rw_type" : "rw",
    "user_type" : "no_root_squash"
  } ]
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Successful query
500	Error response

Error Codes

See **Error Codes**.

5.6.7 Querying a Permission Rule of a File System

Function

This API is used to query a permission rule of a file system.

Constraints

This API is only supported for NFS file systems.

URI

GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}

Table 5-127 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.
rule_id	Yes	String	The permission rule ID.

Request Parameters

Table 5-128 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-129 Response body parameters

Parameter	Type	Description
id	String	The permission rule ID.
ip_cidr	String	The IP address or IP address range of the authorized object.

Parameter	Type	Description
rw_type	String	The read/write permission of the authorized object. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission
user_type	String	The file system access permission granted to the user of the authorized object. The value can be: • **no_root_squash**: allows root on the client to access the file system as **root**. • **root_squash**: allows root on the client to access the file system as **nfsnobody**. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**. This is the default value.

Status code: 400

Table 5-130 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-131 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

Querying details about the permission rule whose ID is **11abef677ac40f46644d1d5cfc2424a4** for the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
GET HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/77ba6f4b-6365-4895-8dda-bc7142af4dde/fs/perm-rules/11abef677ac40f46644d1d5cfc2424a4
```

Example Responses

Status code: 200

Successful query

- Response example of querying a specific permission rule of a file system

```
{
  "id" : "1131ed520xxxxxxebedb6e57xxxxxxxx",
  "ip_cidr" : "192.168.xx.xx/16",
  "rw_type" : "rw",
  "user_type" : "no_root_squash"
}
```

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "Invalid rule id"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Successful query
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.8 Modifying a Permission Rule

Function

This API is used to modify a permission rule.

Constraints

This API is only supported for NFS file systems.

URI

PUT /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}

Table 5-132 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.
rule_id	Yes	String	The permission rule ID.

Request Parameters

Table 5-133 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Table 5-134 Request body parameters

Parameter	Mandatory	Type	Description
rw_type	No	String	The read/write permission of the object to be authorized. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission

Parameter	Mandatory	Type	Description
user_type	No	String	The file system access permission granted to the user of the object to be authorized. The value can be: • **no_root_squash** (default value): allows any user including root on the client to access the file system as who they are, instead of mapping them to another user. • **root_squash**: allows root on the client to access the file system as **nfsnobody** and allows a non-root user on the client to access as who they are, instead of being mapped to another user. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**.

Response Parameters

Status code: 200

Table 5-135 Response body parameters

Parameter	Type	Description
id	String	The permission rule ID.
ip_cidr	String	The IP address or IP address range of the authorized object.
rw_type	String	The read/write permission of the authorized object. • **rw**: read and write permission, which is the default option • **ro**: read-only permission • **none**: no permission

Parameter	Type	Description
user_type	String	The file system access permission granted to the user of the authorized object. The value can be: • **no_root_squash**: allows root on the client to access the file system as **root**. • **root_squash**: allows root on the client to access the file system as **nfsnobody**. • **all_squash**: allows any user on the client to access the file system as **nfsnobody**. This is the default value.

Status code: 400

Table 5-136 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-137 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

```
{
  "rw_type" : "rw",
  "user_type" : "no_root_squash"
}
```

Example Responses

Status code: 200

Success


```
{
  "id" : "1131ed520xxxxxxebedb6e57xxxxxxxx",
  "ip_cidr" : "192.32.0.0/16",
  "rw_type" : "rw",
  "user_type" : "no_root_squash"
}
```

Status code: 400

Error response

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "Invalid rule id"
}
```

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Success
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.6.9 Deleting a Permission Rule

Function

This API is used to delete a permission rule.

Constraints

This API is only supported for NFS file systems.

URI

DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}

Table 5-138 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.
share_id	Yes	String	The file system ID.
rule_id	Yes	String	The permission rule ID.

Request Parameters

Table 5-139 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 204

Successful deletion

Status code: 400

Table 5-140 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-141 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

Deleting the permission rule whose ID is **11abef677ac40f46644d1d5cfc2424a4** for the file system whose ID is **77ba6f4b-6365-4895-8dda-bc7142af4dde**

```
DELETE HTTPS://{endpoint}/v1/{project_id}/sfs-turbo/shares/77ba6f4b-6365-4895-8dda-bc7142af4dde/fs/perm-rules/11abef677ac40f46644d1d5cfc2424a4
```

Example Responses

Status code: 500

Error response

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
204	Successful deletion
400	Error response
500	Error response

Error Codes

See **Error Codes**.

5.7 Task Management

5.7.1 Querying the Status of a Task

Function

This API is used to query the execution status of the SFS Turbo asynchronous API.

URI

GET /v1/{project_id}/sfs-turbo/jobs/{job_id}

Table 5-142 Path Parameters

Parameter	Mandatory	Type	Description
project_id	Yes	String	The project ID.

Parameter	Mandatory	Type	Description
job_id	Yes	String	job ID

Request Parameters

Table 5-143 Request header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	The account token.
Content-Type	Yes	String	The MIME type.

Response Parameters

Status code: 200

Table 5-144 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-145 Response body parameters

Parameter	Type	Description
status	String	The task status, which can be **success**, **running**, **failed**, or **waiting**. Enumeration values: • **success** • **failed** • **waiting** • **running**
job_id	String	The task ID.
job_type	String	The task type.
begin_time	String	The task start time in UTC format, for example, **2016-01-02 15:04:05**.
end_time	String	The task end time in UTC format, for example, **2016-01-02 15:04:05**.

Parameter	Type	Description
error_code	String	The error code returned if the task execution fails.
fail_reason	String	The cause of the task execution failure.
sub_jobs	Array of **GetSubJobDetail** objects	The subtask list.

Table 5-146 GetSubJobDetail

Parameter	Type	Description
status	String	The subtask status. The value can be **success**, **running**, **failed**, or **waiting**.
job_id	String	The subtask ID.
job_type	String	The subtask type.
begin_time	String	The subtask start time in UTC format, for example, **2016-01-02 15:04:05**.
end_time	String	The subtask end time in UTC format, for example, **2016-01-02 15:04:05**.
error_code	String	The error code returned if the subtask execution fails.
fail_reason	String	The cause of the subtask execution failure.

Status code: 400

Table 5-147 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-148 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 404

Table 5-149 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-150 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Status code: 500

Table 5-151 Response header parameters

Parameter	Type	Description
X-request-id	String	The request ID.

Table 5-152 Response body parameters

Parameter	Type	Description
errCode	String	The error code.
errMsg	String	The error message.

Example Requests

None

Example Responses

Status code: 200

Response body parameter

```
{
  "job_id" : "26f6b565-xxxx-XXXX-xxxx-03f0bd975433",
  "status" : "success",
  "job_type" : "bind_ldap",
  "begin_time" : "2023-07-26 09:33:58",
  "end_time" : "2023-07-26 09:33:58"
}
```


Status code: 400

Client error

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "parameter error"
}
```

Status code: 404

Resource not found

```
{
  "errCode" : "SFS.TURBO.0001",
  "errMsg" : "parameter error"
}
```

Status code: 500

Internal error

```
{
  "errCode" : "SFS.TURBO.0005",
  "errMsg" : "Internal server error"
}
```

Status Codes

Status Code	Description
200	Response body parameter
400	Client error
404	Resource not found
500	Internal error

Error Codes

See **Error Codes**.

6 Permissions Policies and Supported Actions

6.1 Introduction

This section describes fine-grained permissions management for your SFS Turbo resources. If your HUAWEI ID 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.

Each 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 projects/Enterprise projects: Authorization scope of custom policies, which can be IAM projects, enterprise projects, or both. Custom policies that contain actions for both IAM projects and enterprise projects can be assigned to user groups and be applied in both IAM and Enterprise Management. Policies that contain actions only for IAM projects can be assigned to user groups and be applied only in IAM. They cannot be applied in Enterprise Management.
- 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 Turbo supports the following actions that can be defined in custom policies:

Lifecycle Management: include actions supported by all SFS Turbo file system APIs, such as the APIs for creating file systems, querying file system details, deleting file systems, listing file systems, and expanding capacities of file systems.

6.2 SFS Turbo Actions

Lifecycle Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Creating a File System	POST /v1/{project_id}/sfs-turbo/shares	sfsturbo:shares:createShare	• VPC actions, including verifying VPCs, subnets, and security groups, creating virtual IP addresses and ports, and creating security group rules – "vpc:*:*" • KMS Administrator permissions • DSS actions (required only in dedicated scenarios) – "dss:*:get", – "dss:*:list", – "dss:*:count"	√	√
Querying Details About a File System	GET /v1/{project_id}/sfs-turbo/shares/{share_id}	sfsturbo:shares:getShare	-	√	√

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Deleting a File System	DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}	sfsturbo:shares:deleteShare	• VPC actions, including deleting virtual IP addresses and ports and deleting security group rules – "vpc:*:*" • DSS actions (required only in dedicated scenarios) – "dss:*:get", – "dss:*:list", – "dss:*:count"	√	√
Listing File Systems	GET /v1/{project_id}/sfs-turbo/shares/detail	sfsturbo:shares:getAllShares	-	√	√
Expanding the Capacity of a File System	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action	sfsturbo:shares:shareAction	-	√	√

Connection Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Changing the Security Group Associated with a File System	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action	sfsturbo:shares:shareAction	Security group actions • vpc:securityGroups:* • vpc:securityGroupRules:*	√	√

Tag Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Adding a Tag to a File System	POST /v1/{project_id}/sfs-turbo/{share_id}/tags	sfsturbo:shares:addTag	-	√	√
Querying Tags of a File System	GET /v1/{project_id}/sfs-turbo/{share_id}/tags	sfsturbo:shares:getTag	-	√	√
Deleting a Tag from a File System	DELETE /v1/{project_id}/sfs-turbo/{share_id}/tags/{key}	sfsturbo:shares:deleteTag	-	√	√
Batch Adding Tags to a File System	POST /v1/{project_id}/sfs-turbo/{share_id}/tags/action	sfsturbo:shares:batchResTag	-	√	√

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying Tags of All File Systems of a Tenant	GET /v1/{project_id}/sfs-turbo/tags	sfsturbo:shares:getAllTag	-	√	√
Querying File Systems by Tag	POST /v1/{project_id}/sfs-turbo/resource_instances/action	sfsturbo:shares:getAllTag	-	√	√

Name Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Changing the Name of a File System	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action	sfsturbo:shares:shareAction	-	√	√

File System Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Obtaining IP Addresses of the Clients Who Have Mounted the File System	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/action	sfsturbo:shares:shareAction	-	√	√

Permissions Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Joining an AD Domain	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain	sfsturbo:shares:addActiveDirectoryDomain	-	√	√
Querying the AD Domain Configuration	GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain	sfsturbo:shares:showActiveDirectoryDomain	-	√	√
Modifying the AD Domain Configuration	PUT /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain	sfsturbo:shares:updateActiveDirectoryDomain	-	√	√
Leaving an AD Domain	DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/active-directory-domain	sfsturbo:shares:deleteActiveDirectoryDomain	-	√	√
Creating a Permission Rule	POST /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules	sfsturbo:shares:createPermRule	-	√	√

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying Permission Rules of a File System	GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules	sfsturbo:shares:listPermRules	-	√	√
Querying a Permission Rule of a File System	GET /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}	sfsturbo:shares:showPermRule	-	√	√
Modifying a Permission Rule	PUT /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}	sfsturbo:shares:updatePermRule	-	√	√
Deleting a Permission Rule	DELETE /v1/{project_id}/sfs-turbo/shares/{share_id}/fs/perm-rules/{rule_id}	sfsturbo:shares:deletePermRule	-	√	√

Task Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying Details About a Task	GET /v1/{project_id}/sfs-turbo/jobs/{job_id}	sfsturbo:shares:getJob	-	√	√

Operations Management

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Changing the Billing Mode of a File System from Pay-per-Use to Yearly/Monthly	POST /v2/{project_id}/sfs-turbo/shares/{share_id}/change-charge-mode	sfsturbo:shares:changeChargeMode	-	√	√

File System Type and Quota Query

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying File System Types and Quotas	GET /v1/{project_id}/sfs-turbo/share-types	sfsturbo:shares:getFlavors	-	√	√

Console Reference

Permission	API	Action	Dependencies	IAM Project (Project)	Enterprise Project (Enterprise Project)
Querying the SFS Turbo Quota	Console reference	sfsturbo:shares:getQuota	-	√	√
Obtaining the AZ Information	Console reference	sfsturbo:shares:getAZInfo	-	√	√
Obtaining SFS Turbo Specifications	Console reference	sfsturbo:shares:getFlavors	-	√	√
Checking the Name of a File System	Console reference	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.
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.

Returned Value	Description
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 Error Codes

Status Code	Error Code	Error Message	Description	Solution
400/404	SFS.TURBO.0001	Parameter error	Invalid parameters.	Use valid parameters and try again.
400/404	SFS.TURBO.0002	Cluster not found	The requested object is not found 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.
400	SFS.TURBO.0004	Invalid vpc	Invalid VPC.	Use valid parameters and try again.

Status Code	Error Code	Error Message	Description	Solution
400/500	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.
409	SFS.TURBO.0009	Name has existed	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.
400/403	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.0015	do not have the operation permission	Insufficient permissions.	Apply for the required permissions.
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.
404	SFS.TURBO.0019	Invalid Job Id	Invalid job ID.	Use a valid job ID.
400	SFS.TURBO.0020	Invalid flavor	Invalid flavor.	Use a valid flavor.
400	SFS.TURBO.0021	file system not match	Unmatched type. The background disk type is not supported by this file system type.	Ensure that the background disk type is supported by the file system type.
400	SFS.TURBO.0022	backup name already exists	The backup name already exists.	Change the backup name.
400	SFS.TURBO.0023	Invalid flavor ref	Invalid specification code.	Use a valid specification code.
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.

Status Code	Error Code	Error Message	Description	Solution
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.
400	SFS.TURBO.0031	cache type not exist	The cache type is not found.	Use a valid cache type.
400	SFS.TURBO.0032	EVS Resource Not Enough	Insufficient EVS resources.	Enlarge EVS resources.
500	SFS.TURBO.0033	Get Client Ips Error	Failed to obtain client IP addresses.	Try again. If the fault persists, contact technical support.
400	SFS.TURBO.0034	dedicated storage resource not enough	Insufficient resources in the dedicated storage pool.	Expand the storage pool.
400	SFS.TURBO.0035	The current type does not support backup.	Unsupported type for backup.	Unsupported type for backup.
500	SFS.TURBO.0036	Failed to obtain the used capacity of the directory	Failed to obtain the used capacity of the directory.	Try again. If the fault persists, contact technical support.
400	SFS.TURBO.0037	Operation conflict, client retry	Operation conflict.	Try again with valid operations.
400	SFS.TURBO.0038	unknown error	Unknown error.	Contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	SFS.TURBO.0039	The VIP quota is insufficient	Insufficient virtual IP address quota.	Apply for a higher quota.
400	SFS.TURBO.0040	Insufficient Security Group Quota	Insufficient security group quota.	Apply for a higher quota.
400	SFS.TURBO.0041	Operation is not allowed	File system version too early.	Contact technical support.
404	SFS.TURBO.0042	Invalid NIC ID	The specified NIC ID is not found or is empty.	Use valid parameters and try again.
400	SFS.TURBO.0100	Invalid file system path	Invalid file system path.	Use a valid file system path.
404	SFS.TURBO.0101	The file system path does not exist	The file system path is not found.	Select a valid file system path.
400	SFS.TURBO.0102	The file system path is not a directory	The file system path is not a directory.	Select a valid file system path.
400	SFS.TURBO.0103	The file system is being processed	The file system is being processed.	Wait until the processing is complete.
500	SFS.TURBO.0104	Failed to import or export OBS data	Failed to import or export OBS data.	Try again. If the fault persists, contact technical support.
500	SFS.TURBO.0105	Failed to obtain OBS import and export task data	Failed to obtain the OBS import and export data.	Try again. If the fault persists, contact technical support.

Status Code	Error Code	Error Message	Description	Solution
400	SFS.TURBO.0106	The OBS task does not exist	The OBS task is not found.	Select an existing OBS task or create an OBS task.
400	SFS.TURBO.0107	OBS protocol error	Backend parameter **type** is incorrectly configured.	Contact technical support.
400	SFS.TURBO.0108	The OBS endpoint name is incorrect	Incorrect OBS domain name.	Contact technical support.
400	SFS.TURBO.0109	The OBS bucket name is incorrect	Incorrect OBS bucket name.	Use the correct OBS bucket name.
400	SFS.TURBO.0110	OBS agent error	The import or export task failed.	Contact technical support.
400	SFS.TURBO.0111	The OBS configuration list is empty	The OBS configuration list is empty.	Use valid OBS configuration information.

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 **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 the 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.

