

Scalable File Service Turbo

User Guide

Issue 01
Date 2024-11-06



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI 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 File System Management	1
1.1 Creating a File System	1
1.2 Viewing a File System	8
1.3 Deleting a File System	9
2 Permissions Management	11
2.1 Creating a User and Granting SFS Turbo Permissions	11
3 Mount and Access	13
3.1 Mounting a File System	13
3.1.1 Mounting an NFS File System to ECSs (Linux)	13
3.1.2 Mounting a File System Automatically	18
3.2 Unmounting a File System	20
4 Data Security	21
4.1 Encryption	21
5 Backup and DR	22
5.1 Backup	22
6 Data Management	24
6.1 Capacity Expansion	24
6.2 SFS Turbo Quotas	27
7 Monitoring and Auditing	28
7.1 Monitoring SFS Turbo File Systems Using Cloud Eye	28
7.1.1 SFS Turbo Metrics	28
7.2 Auditing SFS Turbo File Systems Using CTS	30
7.2.1 Supported SFS Turbo Operations	30
8 Typical Applications	33
8.1 Enterprise Website/App Background	33
8.2 Log Printing	34
9 Other Operations	36
9.1 Testing SFS Turbo Performance	36
9.2 Mounting a File System to a Linux as a Non-root User	42
9.3 Mounting a Subdirectory of an NFS File System to ECSs (Linux)	45

9.4 Data Migration.....	46
9.4.1 Migration Description.....	46
9.4.2 Migrating Data Using Direct Connect.....	47
9.4.3 Migrating Data Using the Internet.....	48

1 File System Management

1.1 Creating a File System

You can create an SFS Turbo file system and mount it to multiple servers. Then the servers can share this file system.

Prerequisites

1. A VPC is available.
If no VPC is available, create one by referring to section "Creating a VPC" in the *Virtual Private Cloud User Guide*.
2. ECSs are available and they belong to the created VPC.
If no ECSs are available, create ECSs by referring to section "Creating an ECS" in the *Elastic Cloud Server User Guide*.
3. Creating SFS Turbo file systems depends on the following services: VPC, Billing Center, DSS, and ECS. Ensure that required roles or policies have been configured.
 - The permissions of the **SFS Turbo FullAccess** policy already include the permissions of **VPC FullAccess**, which are required for creating file systems. An IAM user assigned the **SFS Turbo Full Access** policy does not need to have the **VPC FullAccess** policy assigned explicitly.
 - To create yearly/monthly file systems, the **BSS Administrator** policy is required.
 - To create file systems in dedicated projects, the **DSS FullAccess** and **ECS FullAccess** policies are required.

Signing In to the Console

Step 1 Visit the [Huawei Cloud website](#).

Step 2 Sign up for an account.

Before using SFS Turbo, you need to sign up for a HUAWEI ID and enable Huawei Cloud services. You can use this account to access all Huawei Cloud services, including SFS Turbo. If you already have an account, start from [Step 3](#).

1. In the upper right corner of the page, click **Sign Up**.
2. Complete the registration as instructed.
After the registration is complete, you will be redirected to your personal information page.

Step 3 Sign in to the console.

1. In the upper right corner of the displayed page, click **Console**.
2. Enter the username and password as prompted, and click **Sign In**.

Step 4 In the upper left corner of the page, select the region where the service is located from the drop-down list.

Step 5 Choose **Storage > Scalable File Service** to go to the SFS Turbo console.

Step 6 (Recommended) Top up your account and then buy and use SFS Turbo file systems.

----End

Creating an SFS Turbo File System

Step 1 Log in to the console using a cloud account.

1. Log in to the console and select a region and a project.
2. Choose **Storage > Scalable File Service**.

Step 2 In the upper right corner of the page, click **Create File System**.

Step 3 Configure the parameters based on [Table 1-1](#), as shown in [Figure 1-1](#).

Figure 1-1 Creating an SFS Turbo file system

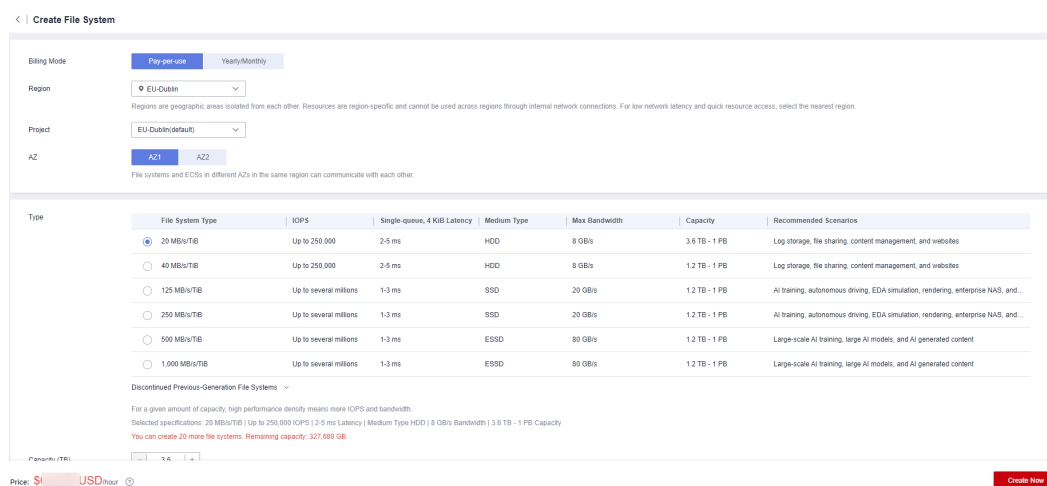


Table 1-1 File system parameters

Parameter	Description	Remarks
Billing Mode	Mandatory	-

Parameter	Description	Remarks
Region	Mandatory Region of the tenant. Select a region from the drop-down list in the upper left corner of the page.	You are advised to select the region where the servers reside.
AZ	Mandatory A geographical area with an independent network and an independent power supply.	There is certain performance loss when a file system is accessed from a different AZ. You are advised to select the AZ where your servers reside.
Type	Mandatory The following types are supported: Standard, Standard-Enhanced (Discontinued), Performance, Performance-Enhanced (Discontinued), 20 MB/s/TiB, 40 MB/s/TiB, 125 MB/s/TiB, 250 MB/s/TiB, 500 MB/s/TiB, and 1,000 MB/s/TiB.	Select Standard . NOTE After a file system is created, its type cannot be changed. If you want to change the type, you need to create another file system. So, plan the file system type in advance.
Capacity	Maximum capacity allowed for a single file system. When the used capacity of a file system reaches this value, no more data can be written to the file system. You need to expand the file system capacity. The capacity of an SFS Turbo file system cannot be reduced. Set an appropriate file system capacity based on your service needs.	Supported ranges: <ul style="list-style-type: none"> • Standard: 500 GB to 32 TB • Performance: 500 GB to 32 TB • Standard-Enhanced (Discontinued): 10 TB to 320 TB • Performance-Enhanced (Discontinued): 10 TB to 320 TB • 20 MB/s/TiB: 3.6 TB to 1 PB • 40 MB/s/TiB: 1.2 TB to 1 PB • 125 MB/s/TiB: 1.2 TB to 1 PB • 250 MB/s/TiB: 1.2 TB to 1 PB • 500 MB/s/TiB: 1.2 TB to 1 PB • 1,000 MB/s/TiB: 1.2 TB to 1 PB

Parameter	Description	Remarks
Bandwidth (GB/s)	Defines the cache bandwidth, which is recommended for workloads with frequent reads but infrequent writes. The higher the bandwidth, the larger the capacity required.	<ul style="list-style-type: none"> • If you select the 20 MB/s/TiB, 40 MB/s/TiB, 125 MB/s/TiB, 250 MB/s/TiB, 500 MB/s/TiB, or 1,000 MB/s/TiB file system type, this parameter and its value will show up. Bandwidth size = Capacity x Bandwidth density (type value). The minimum bandwidth is 150 MB/s. If the calculated bandwidth is less than 150, 150 MB/s will be used.. • If you select the Standard-Enhanced (Discontinued), Standard, Performance-Enhanced (Discontinued), or Performance type, this parameter will not show up.
Protocol Type	Mandatory SFS Turbo file systems support file access from clients using NFS.	The default value is NFS .

Parameter	Description	Remarks
VPC	<p>Mandatory</p> <p>Select a VPC and a subnet.</p> <ul style="list-style-type: none"> • VPC: A server cannot access file systems in a different VPC. Select the VPC to which the servers reside. • Subnet: A subnet is a unique IP address range in a VPC. A subnet provides dedicated network resources that are logically isolated from other networks to improve network security. <p>NOTE</p> <ul style="list-style-type: none"> • To achieve the optimal network performance, select the VPC where your servers reside. You can also use VPC peering connections to connect two or more VPCs to share files between VPCs. When a file system is accessed across VPCs, the latency, bandwidth, and IOPS loss may be high. Therefore, intra-VPC access is recommended. <p>For details about VPC peering connections, see section "VPC Peering Connection" in <i>Virtual Private Cloud User Guide</i>.</p>	-
Enterprise Project	<p>This function is provided for enterprise users. When creating a file system, you can add the file system to an existing enterprise project.</p> <p>An enterprise project makes it easy to manage projects and groups of cloud resources and users. Use the default enterprise project or create one.</p> <p>Select an enterprise project from the drop-down list.</p>	<p>You can select only created enterprise projects. To create an enterprise project, click Enterprise in the upper right corner of the console page.</p>

Parameter	Description	Remarks
Encryption	<p>Optional</p> <p>Specifies whether a file system is encrypted. You can create a file system that is encrypted or not, but you cannot change the encryption attribute of an existing file system. If you enable encryption, the following parameters will be displayed:</p> <ul style="list-style-type: none"> ● KMS key name A key name is the identifier of the key, and you can use KMS key name to specify a KMS key and use it for encryption. Select an existing key from the drop-down list, or click View KMS List to create a new key. For details, see Creating a CMK in the <i>Data Encryption Workshop User Guide</i>. ● KMS Key ID After you select a key name, the system automatically shows the key ID. ● Key Encryption Algorithm After you select a key name, the system automatically shows the encryption algorithm of the key. 	<p>You are advised to enable encryption to ensure core data security. If you use KMS encryption, any usage beyond the free quota given by KMS will be billed. For details, see DEW Pricing Details.</p>

Parameter	Description	Remarks
Security Group	<p>Mandatory</p> <p>A security group is a virtual firewall that provides network access control policies for file systems. You can define access rules for a security group. Then these rules will apply to all file systems added to this security group.</p> <p>When creating an SFS Turbo file system, you can select only one security group.</p> <p>You are advised to use an independent security group for an SFS Turbo file system to isolate it from service nodes.</p> <p>The security group rules affect the normal access and use of an SFS Turbo file system. For details about how to configure a security group rule, see section "Adding a Security Group Rule" in the <i>Virtual Private Cloud User Guide</i>. After an SFS Turbo file system is created, the system automatically enables the security group ports required by NFS. This ensures that the SFS Turbo file system can be successfully mounted to your servers. The inbound ports required by NFS are ports 111, 2049, 2051, 2052, and 20048. If you need to change the enabled ports, go to the VPC console, choose Access Control > Security Groups, locate the target security group, and change the ports.</p>	-
Name	<p>Mandatory</p> <p>User-defined file system name.</p>	<p>The name must start with a letter and can contain only letters, digits, underscores (_), and hyphens (-). It must contain more than four characters but no more than 64 characters.</p>

Step 4 Click **Create Now**.

Step 5 Confirm the file system information and click **Submit**.

Step 6 When the creation is complete, go back to the file system list.

If the status of the created file system is **Available**, the file system is created successfully. If the status is **Creation failed**, contact the administrator.

----End

1.2 Viewing a File System

You can search for file systems by file system name, status or other properties, and view their basic information.

NOTE

Viewing details of SFS Turbo file systems depends on the VPC service. Ensure that the required role or policy has been configured.

The permissions of the **SFS Turbo ReadOnlyAccess** policy already include the permissions of **VPC ReadOnlyAccess**, which are required for querying file system details. An IAM user assigned the **SFS Turbo ReadOnlyAccess** policy does not need to have the **VPC ReadOnlyAccess** policy assigned explicitly.

Procedure

Step 1 Log in to the SFS Turbo console.

Step 2 In the file system list, view the file systems you have created. [Table 1-2](#) describes the file system parameters.

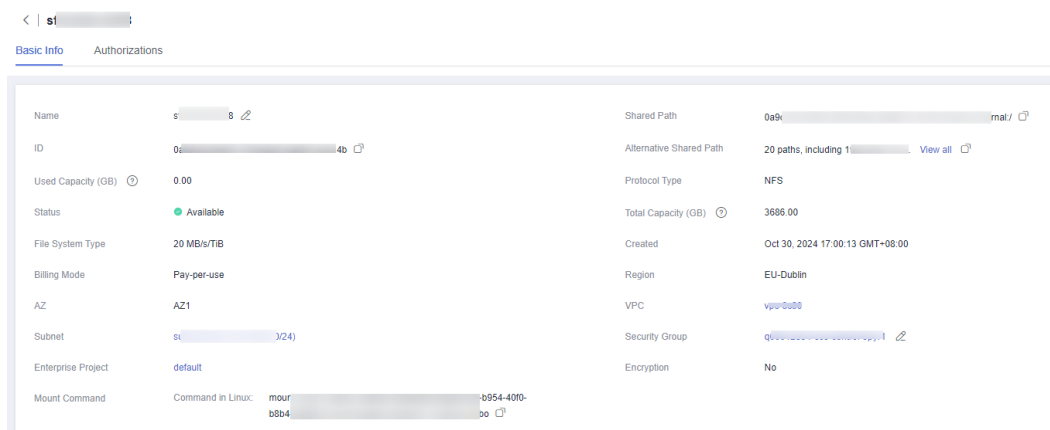
Table 1-2 File system parameters

Parameter	Description
Name	Name of a file system, for example, sfs-turbo-name001
Status	Possible values are Available , Unavailable , Frozen , Creating , Deleting .
Type	File system type
Protocol Type	File system protocol, which is NFS
Used Capacity (GB)	File system space already used for data storage NOTE This information is refreshed every 15 minutes.
Maximum Capacity (GB)	Maximum capacity of the file system
Encryption	Encryption status of a file system. The value can be Yes or No .
Enterprise Project	Enterprise project to which a file system belongs

Parameter	Description
Billing Mode	Billing mode of a file system. The value can be Pay-per-use or Yearly/Monthly . In addition, the creation time is displayed for a pay-per-use file system, and the expiration time is displayed for a yearly/monthly file system.
Operation	Provides the Expand Capacity , Delete , View Metric , Create Backup , Renew , and Unsubscribe buttons. NOTE You can renew or unsubscribe from a yearly/monthly SFS Turbo file system about 1 or 2 minutes after it has been created.

Step 3 Click the name of a file system to view its basic information.

Figure 1-2 Details of an SFS Turbo file system



Step 4 (Optional) Search for file systems by file system name, ID, AZ, type, protocol type, used capacity, or status, and view their basic information.

----End

1.3 Deleting a File System

After you delete a file system, data in it cannot be restored. To prevent data loss, ensure that files in a file system have been properly stored or backed up before you delete a file system.

Prerequisites

You are advised to unmount the file system before deleting it. For details, see [Unmounting a File System](#).

Procedure

Step 1 In the file system list, locate the file system you want to delete and choose **More > Delete** or **Unsubscribe** in the **Operation** column.

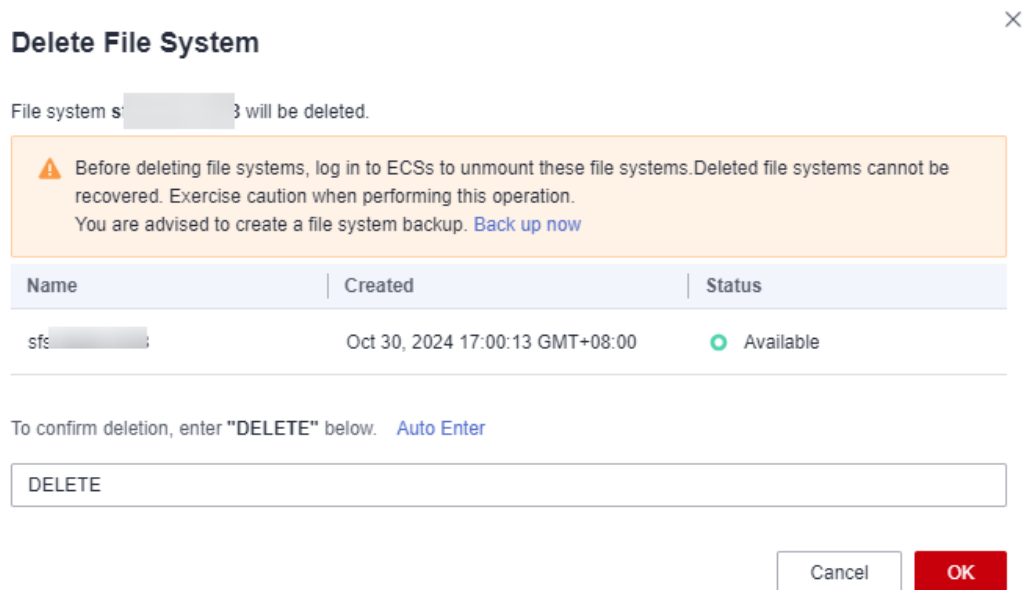
Step 2 In the dialog box, confirm the information, enter **DELETE** in the text box, and click **OK**.

After clicking **Unsubscribe** for a yearly/monthly SFS Turbo file system, complete the unsubscription as prompted.

 **NOTE**

Only **Available** and **Unavailable** file systems can be deleted or unsubscribed from.

Figure 1-3 Deleting a file system



Cancel

OK

Step 3 Check that the file system disappears from the file system list.

----End

2 Permissions Management

2.1 Creating a User and Granting SFS Turbo Permissions

This section describes how to use IAM to implement fine-grained permissions control for your SFS Turbo resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing SFS Turbo resources.
- Grant users only the permissions required to perform a given task based on their job responsibilities.

If your Huawei Cloud account does not require individual IAM users, skip this section.

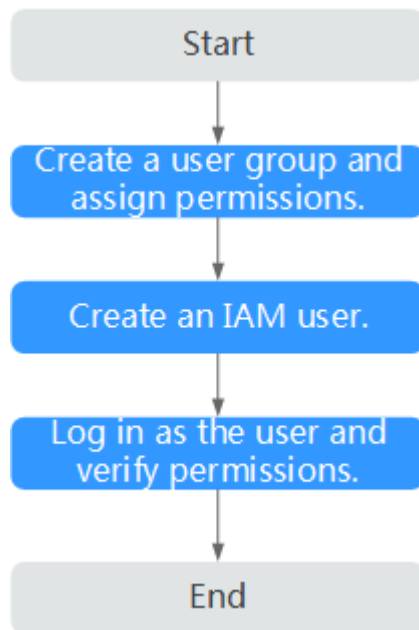
This section describes the procedure for granting user permissions. [Figure 2-1](#) shows the process flow.

Prerequisites

Before granting permissions to user groups, learn about SFS Turbo system-defined permissions.

Process Flow

Figure 2-1 Process for granting SFS Turbo permissions



1. Create a user group and assign permissions to it.
Create a user group on the IAM console and assign the **SFS Turbo ReadOnlyAccess** permissions to the group.
2. Create a user and add it to a user group.
Create a user on the IAM console and add the user to the group created in **1**.
3. Log in and verify permissions.
In the authorized region, perform the following operations:
 - Choose **Service List > Scalable File Service Turbo**. On the SFS Turbo console, click **Create File System** in the upper right corner. If a message appears indicating that you have insufficient permissions to perform the operation, the **SFS Turbo ReadOnlyAccess** permissions are in effect.
 - Choose another service from **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **SFS Turbo ReadOnlyAccess** permissions are in effect.

3 Mount and Access

3.1 Mounting a File System

3.1.1 Mounting an NFS File System to ECSs (Linux)

After creating a file system, you need to mount it to ECSs so that they can share the file system.

In this section, ECSs are used as example servers. Operations on BMSs are the same as those on ECSs.

To use SFS Turbo file systems as storage backends for CCE, see [Storage](#). Then complete the deployment on the CCE console.

SFS Turbo file systems cannot be mounted to Windows ECSs.

Prerequisites

- You have checked the type of the OS on each ECS. Different OSs use different commands to install the NFS client.
- You have created an SFS Turbo file system and obtained its shared path.
- At least an ECS is available.

Notes and Constraints

NOTE

This constraint only applies to local paths (mount points) and does not affect other files or directories.

Metadata of the local paths (mount points) cannot be modified. Specifically, the following operations cannot be performed on the local paths' metadata:

- **touch**: Update file access time and modification time.
- **rm**: Delete files or directories.
- **cp**: Replicate files or directories.

- **mv**: Move files or directories.
- **rename**: Rename files or directories.
- **chmod**: Modify permissions on files or directories.
- **chown**: Change the owners of files or directories.
- **chgrp**: Change the group of a file or directory.
- **ln**: Create hard links.
- **link**: Create hard links.
- **unlink**: Delete hard links.

The **atime**, **ctime**, and **mtime** attributes of a local path (root directory of the mount point) are the current time. So each time the root directory attribute is queried, the current time of the server is returned.

Procedure

Step 1 Log in to the console using a cloud account.

1. Log in to the console and select a region and a project.
2. Choose **Computing** > **Elastic Cloud Server** to go to the ECS console.

Step 2 Log in to the ECS as user **root**.

NOTE

If you log in to the ECS as a non-root user, see [Mounting a File System to a Linux as a Non-root User](#).

Step 3 Install the NFS client.

1. **Install the NFS client.**

a. Check whether the NFS software package is installed.

- On CentOS, Red Hat, Oracle Enterprise Linux, SUSE, EulerOS, Fedora, or OpenSUSE, run the following command:

```
rpm -qa|grep nfs
```

- On Debian or Ubuntu, run the following command:

```
dpkg -l nfs-common
```

If no such command output is displayed, go to [Step 3.1.b](#).

- On CentOS, Red Hat, EulerOS, Fedora, or Oracle Enterprise Linux, the command output is as follows:

```
libnfsidmap  
nfs-utils
```

- On SUSE or OpenSUSE, the command output is as follows:

```
nfsidmap  
nfs-client
```

- On Debian or Ubuntu, the command output is as follows:

```
nfs-common
```

b. Install the NFS software package.

 NOTE

The following commands require that the ECSs be connected to the Internet. Or, the installation will fail.

- On CentOS, Red Hat, EulerOS, Fedora, or Oracle Enterprise Linux, run the following command:
sudo yum -y install nfs-utils
- On Debian or Ubuntu, run the following command:
sudo apt-get install nfs-common
- On SUSE or OpenSUSE, run the following command:
zypper install nfs-client

Step 4 Create a local path for mounting the file system.

mkdir *Local path*

 NOTE

If any other resources, such as a disk, have been mounted on the local path, create a new path. (NFS clients do not refuse repeated mounts. If there are repeated mounts, information of the last successful mount is displayed.)

Step 5 Mount the file system to the ECSs in the same VPC as the file system. You can mount the file system to Linux ECSs using NFSv3 only.

Table 3-1 describes the mount parameters.

To mount an SFS Turbo file system, run the following command: **mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp** *Shared path Local path*

NOTICE

After a mounted ECS is restarted, it loses the file system mount information. You can configure auto mount in the **fstab** file to ensure that an ECS automatically mounts the file system when it restarts. For details, see [Mounting a File System Automatically](#).

Table 3-1 Parameters required for mounting file systems

Parameter	Description
vers	File system version. Only NFSv3 is supported currently, so the value is fixed to 3.
timeo	Waiting time before the NFS client retransmits a request. The unit is 0.1 second. The recommended value is 600 .

Parameter	Description
noresvport	Whether the NFS client uses a new TCP port when it re-establishes a network connection to the NFS server. It is strongly recommended that you specify noresvport , which ensures that your file system remains uninterrupted after a network reconnection or recovery.
lock/nolock	Whether to use the NLM protocol to lock files on the server. If nolock is specified, the lock is valid only for applications on the same host. It is invalid for applications on any other hosts. The recommended value is nolock . If this parameter is not specified, lock is used by default. Then, other servers cannot write data to the file system.
<i>Shared path</i>	For an SFS Turbo Standard-Enhanced, Standard, Performance-Enhanced, or Performance file system, the format is <i>File system IP address:/</i> , for example, 192.168.0.0:/ . For an SFS Turbo 20 MB/s/TiB, 40 MB/s/TiB, 125 MB/s/TiB, 250 MB/s/TiB, 500 MB/s/TiB, 1,000 MB/s/TiB, or HPC cache file system, the format is <i>File System domain name:/</i> , for example, xxx.sfsturbo.internal:/ .
<i>Local path</i>	Local path on the ECS used to mount the file system, for example, /local_path .

For more performance optimization mount options, see [Table 3-2](#). Use commas (,) to separate parameters. A command example is provided as follows:

```
mount -t nfs -o vers=3,timeo=600,nolock,rsize=1048576,wsizer=1048576,hard,retrans=3,noresvport,ro,async,noatime,nodiratime Shared path Local path
```

Table 3-2 Mount options for performance optimization

Parameter	Description
rsize	Maximum number of bytes in each read request that the client can receive when reading data from a file on the server. The actual data size is less than or equal to this parameter setting. The value of rsize must be a positive integral multiple of 1024 . Specified values less than 1024 are automatically replaced with 4096 , and values greater than 1048576 are automatically replaced with 1048576 . By default, this parameter is set through a negotiation between the server and the client. You are advised to set this parameter to the maximum value 1048576 .

Parameter	Description
wsize	<p>Maximum number of bytes in each write request that the client can send when writing data to a file on the server. The actual data size is less than or equal to this parameter setting. The value of wsize must be a positive integral multiple of 1024. Specified values less than 1024 are automatically replaced with 4096, and values greater than 1048576 are automatically replaced with 1048576. By default, this parameter is set through a negotiation between the server and the client.</p> <p>You are advised to set this parameter to the maximum value 1048576.</p>
soft/hard	<p>soft indicates soft mounts. With soft specified, if an NFS request times out, the client returns an error to the calling program. hard indicates hard mounts. With hard specified, if an NFS request times out, the client continues to request until the request is successful.</p> <p>The default value is hard.</p>
retrans	<p>Number of retransmission times before the client returns an error. The recommended value is 1.</p>
ro/rw	<ul style="list-style-type: none">• ro: indicates that the file system is mounted as read-only.• rw: indicates that the file system is mounted as read/write. <p>The default value is rw. If this parameter is not specified, the file system will be mounted as read/write.</p>
noresvport	<p>Whether the NFS client uses a new TCP port when it re-establishes a network connection to the NFS server.</p> <p>It is strongly recommended that you specify noresvport, which ensures that your file system remains uninterrupted after a network reconnection or recovery.</p>
sync/async	<p>sync indicates that data is written to the server immediately. async indicates that data is first written to the cache and then to the server.</p> <p>async is recommended. Synchronous writes require that an NFS server returns a success message after all data is written to the server, which brings long latency.</p>
noatime	<p>If you do not need to record the file access time, set this parameter. This prevents overheads caused by frequent access to modify the time.</p>
nodiratime	<p>If you do not need to record the directory access time, set this parameter. This prevents overheads caused by frequent access to modify the time.</p>

 NOTE

You are advised to use the default values for the parameters with no usage recommendations provided.

Step 6 View the mounted file system.

mount -l

If the command output contains the following information, the file system has been mounted:

```
Shared path on /local_path type nfs (rw,vers=3,timeo=600,nolock,addr=)
```

Step 7 Check that you can access the file system on the ECSs to read or write data.

 NOTE

The maximum size of a file that can be written to an SFS Turbo file system is 32 TB, and that for an SFS Turbo Enhanced file system is 320 TB.

----End

3.1.2 Mounting a File System Automatically

File system mount information may be lost after a server is restarted. You can configure auto mount on the server to avoid losing the mount information.

Restrictions

Because service startup sequences in different OSs vary, some servers running CentOS may not support the following auto mount plans. In this case, manually mount the file system.

Procedure (Linux)

Step 1 Log in to the console using a cloud account.

1. Log in to the console and select a region and a project.
2. Choose **Computing** > **Elastic Cloud Server** to go to the ECS console.

Step 2 Log in to the server as user **root**.

Step 3 Run the **vi /etc/fstab** command to edit the **/etc/fstab** file.

At the end of the file, add the file system information, for example:

```
Shared path /local_path nfs vers=3,timeo=600,noresvport,nolock,tcp 0 0
```

Replace *Shared path* and */local_path* with actual values. You can obtain the shared path from the **Shared Path** column of the file system. Each record in the **/etc/fstab** file corresponds to a mount. Each record has six fields, as described in [Mount Fields](#).

NOTICE

For optimal system performance, configure file system information based on the previous example. If needed, you can customize certain mount options. However, the customization may affect system performance.

Step 4 Press **Esc**, enter **:wq**, and press **Enter** to save and exit.

After the preceding configurations are complete, the system reads the mount information from the **/etc/fstab** file to automatically mount the file system when the server restarts.

Step 5 (Optional) View the updated content of the **/etc/fstab** file.

```
cat /etc/fstab
```

Step 6 If auto mount fails due to a network issue, add the **sleep** option and a time in front of the mount command in the **rc.local** file, and mount the file system after the NFS service is started.

```
sleep 10s && sudo mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared_path/local_path
```

```
----End
```

Mount Fields

Table 1 describes the mount fields.

Table 3-3 Mount fields

Field	Description
<i>Shared path</i>	The address of the file system to be mounted. Set it to the shared path in the mount command in Mounting an NFS File System to ECSs (Linux) .
/local_path	The directory created on the server for mounting the file system. Set it to the local path in the mount command in Mounting an NFS File System to ECSs (Linux) .
nfs	The file system or partition mount type. Set it to nfs .
vers=3,timeo=600,noresvport,nolock,tcp	Mount options. Use commas (,) to separate multiple options. <ul style="list-style-type: none"> • vers: The file system version. Value 3 indicates the NFSv3 protocol. • timeo: The waiting time before the NFS client retransmits a request. The unit is 0.1 second. The recommended value is 600. • noresvport: Whether the NFS client uses a new TCP port when it re-establishes a network connection to the NFS server. It is strongly recommended that you specify noresvport, which ensures that your file system remains uninterrupted after a network reconnection or recovery. • nolock: Whether to use the NLM protocol to lock files on the server. If nolock is specified, the lock is valid only for applications on the same host. It is invalid for applications on any other hosts. nolock is recommended. If this parameter is not specified, lock is used by default. Then, other servers cannot write data to the file system. • tcp: The TCP transmission protocol.

Field	Description
0	Choose whether to use dump to back up the file system. <ul style="list-style-type: none">• 0: dump backup is not used.• An integer greater than zero means that the file system is backed up. A smaller value has a higher check priority.
0	Choose whether to use fsck to check the file system when the server starts and specify the check sequence. <ul style="list-style-type: none">• 0: File systems are not checked.• By default, this field is set to 1 for the root directory. The values for other directories start from 2, and one with a smaller integer is checked earlier than that with a larger integer.

3.2 Unmounting a File System

If an SFS Turbo file system is no longer required, you can unmount it and then delete it.

Prerequisites

Stop the process and read/write operations before you unmount a file system.

Linux OS

Step 1 Log in to the console using a cloud account.

1. Log in to the console and select a region and a project.
2. Choose **Computing** > **Elastic Cloud Server** to go to the ECS console.

Step 2 Log in to the ECS.

Step 3 Run the following command:

```
umount Local path
```

Variable *Local path* is an ECS local directory where the file system is mounted, for example, **/local_path**.

NOTE

Before running the **umount** command, stop all read and write operations related to the SFS Turbo file system and exit from the local path. Or, the unmounting will fail.

----End

4 Data Security

4.1 Encryption

Creating an Encrypted File System

You can directly use the encryption function when creating SFS Turbo file systems. No authorization is required. For details, see [File System Encryption](#).

You can create a file system that is encrypted or not, but you cannot change the encryption attribute of an existing file system.

For details about how to create an encrypted file system, see [Creating a File System](#).

Unmounting an Encrypted File System

If the custom key used by an encrypted file system is disabled or scheduled for deletion, the file system can only be used within a certain period of time (30s by default). Exercise caution in this case.

For details about how to unmount an encrypted file system, see [Unmounting a File System](#).

5 Backup and DR

5.1 Backup

You to back up SFS Turbo file systems using CBR.

Scenarios

A backup is a complete copy of an SFS Turbo file system at a specific time. It records all configuration data and service data at that time.

If a file system is faulty or encounters a logical error (for example, accidental deletion, hacker attacks, and virus infection), you can use data backups to restore data quickly.

Creating a Backup

Ensure that the status of the file system you want to back up is **Available**. Or, the backup task cannot start. This procedure describes how to manually create a file system backup

NOTE

When an SFS Turbo Standard, Standard-Enhanced (Discontinued), Performance, or Performance-Enhanced (Discontinued) file system is being backed up, mounting the file system may fail. This is because the connection used for mounting may experience an I/O delay about 30 seconds. You are advised to perform backup during off-peak hours.

Step 1 In the navigation pane on the left, choose **SFS Turbo Backups**.

Step 2 Wait for CBR to automatically create a file system backup.

You can view the backup creation status on the **Backups** tab. When the **Status** of the backup changes to **Available**, the backup has been created.


Step 3 Create a new file system from the backup if the file system becomes faulty or encounters an error occurred. For details, see [Using a Backup to Create a File System](#).

----End

Creating a File System from a Backup

In case of a virus attack, accidental deletion, or software or hardware fault, you can use an SFS Turbo backup to create a new SFS Turbo file system. Data on the new file system is the same as that in the backup.

Step 1 Log in to the CBR console.

1. Log in to the console.
2. Click  in the upper left corner and select a region.
3. Choose **Storage > Cloud Backup and Recovery > SFS Turbo Backups**.

Step 2 Click the **Backups** tab and locate the desired backup.

Step 3 Click **Create File System** in the **Operation** column of the backup. The button is available only when the backup status is **Available**.

Step 4 Configure the file system parameters.

 **NOTE**

- To learn more about these parameters, see [Table 1-1](#).

Step 5 Click **Next**.

Step 6 Go back to the file system list and check whether the file system is successfully created.

You will see the file system status change as follows: **Creating, Available, Restoring, Available**. After the file system status has changed from **Creating** to **Available**, the file system is successfully created. After the status has changed from **Restoring** to **Available**, backup data has been successfully restored to the created file system.

----End

6 Data Management

6.1 Capacity Expansion

Scenarios

You can expand the capacity of a file system if it is insufficient.

Notes and Constraints

SFS Turbo file systems can only have their capacities expanded, not reduced. And only **In-use** file systems can be expanded.

Expanding the Capacity of a Yearly/Monthly SFS Turbo File System

- Step 1** Log in to the SFS Turbo console.
- Step 2** In the file system list, locate the SFS Turbo file system you want to expand capacity and click **Expand Capacity** in the **Operation** column to open the **Expand Capacity** page.

Table 6-1 Capacity expansion parameters

Parameter	Description
Current Capacity	Current storage capacity of the file system

Parameter	Description
New Capacity	<p>New storage capacity of the file system</p> <p>Constraints:</p> <ul style="list-style-type: none"> • For a Standard-Enhanced (Discontinued), Standard, Performance-Enhanced (Discontinued), or Performance file system, the expansion increment is 100 GB. A Standard or Performance file system can be expanded to up to 32 TB, and a Standard-Enhanced or Performance-Enhanced file system can be expanded to up to 320 TB. • For a 20 MB/s/TiB, 40 MB/s/TiB, 125 MB/s/TiB, 250 MB/s/TiB, 500 MB/s/TiB, or 1,000 MB/s/TiB file system, the expansion increment is 1.2 TB, and a file system can be expanded to up to 1 PB.

Step 3 Enter a new capacity based on service requirements and then click **Next**.

Step 4 Confirm the resource information and click **Submit**.

Step 5 Complete the payment as instructed and return to the file system list. Click the name of the expanded file system and check that the capacity has been expanded.

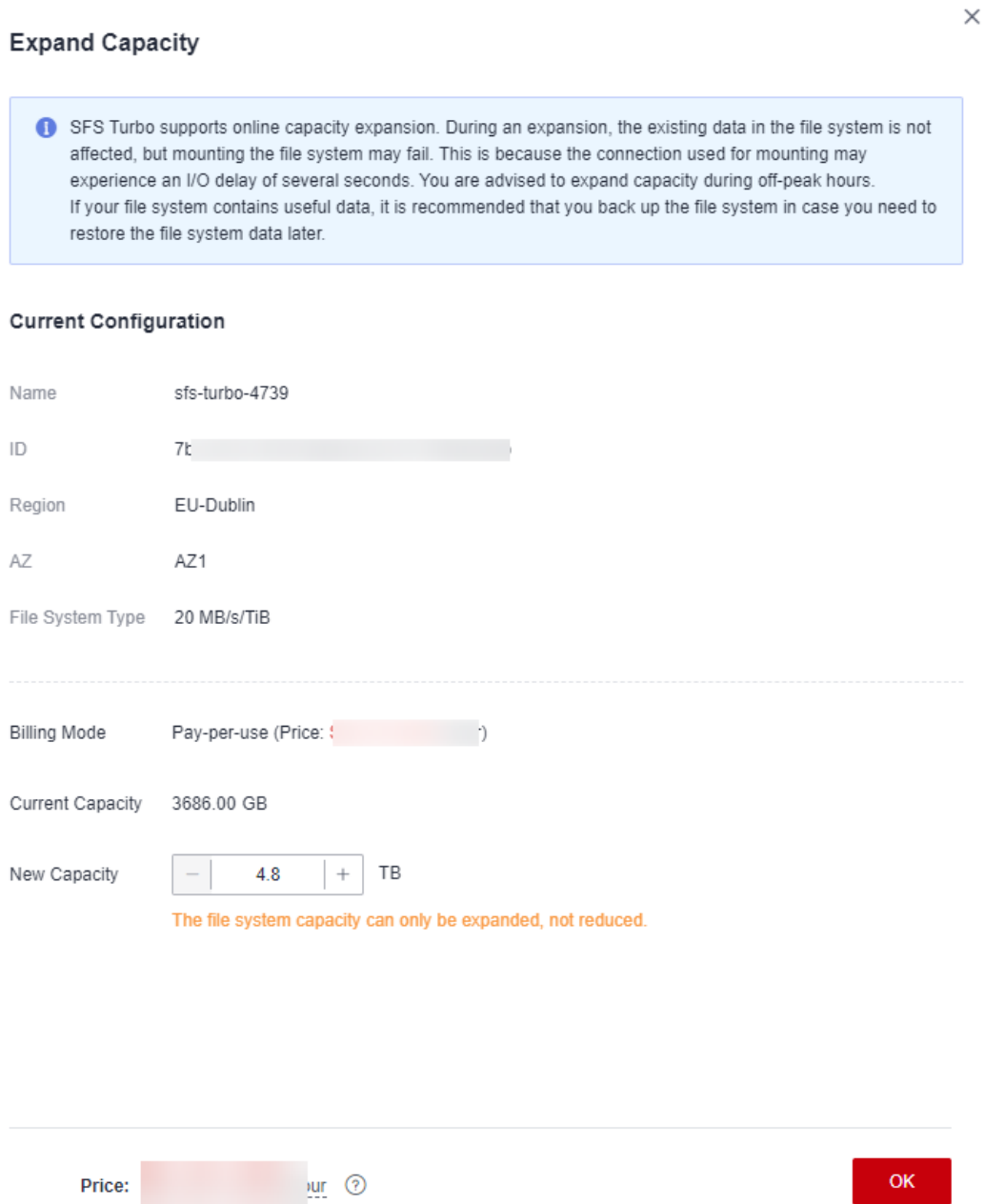
----End

Expanding the Capacity of a Pay-per-Use SFS Turbo File System

Step 1 Log in to the SFS Turbo console.

Step 2 In the file system list, locate the SFS Turbo file system you want to expand capacity and click **Expand Capacity** in the **Operation** column to open the **Expand Capacity** page.

Figure 6-1 Expanding the capacity of a pay-per-use SFS Turbo file system



Step 3 Enter a new capacity based on service requirements. For detailed parameter descriptions, see [Table 6-1](#).

Step 4 Click **OK**. In the file system list, check that the file system capacity has been expanded.

----End


6.2 SFS Turbo Quotas

What Is Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources that you can use, for example, the maximum number of file systems that you can create.

If a quota cannot meet your needs, apply for a higher quota.

How Do I View My Quotas?

1. Log in to the console.
2. Click  in the upper left corner and select a region.
3. In the upper right corner of the page, choose **Resources > My Quotas**.
The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.

If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

1. Log in to the console.
2. In the upper right corner of the page, choose **Resources > My Quotas**.
The **Service Quota** page is displayed.
3. Click **Increase Quota** in the upper right corner of the page.
4. On the **Create Service Ticket** page, configure parameters as required.
In the **Problem Description** area, fill in the content and reason for adjustment.
5. After all necessary parameters are configured, select **I have read and agree to the Ticket Service Protocol and Privacy Statement** and click **Submit**.

7 Monitoring and Auditing

7.1 Monitoring SFS Turbo File Systems Using Cloud Eye

7.1.1 SFS Turbo Metrics

Function

This section describes metrics reported by SFS Turbo to Cloud Eye as well as their namespaces and dimensions. You can use the console or APIs provided by Cloud Eye to query the metrics generated for SFS Turbo.

Namespace

SYS.EFS

Metrics

Table 7-1 SFS Turbo metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
client_connections	Client Connections	Number of client connections NOTE Only active client connections are counted. A network connection is automatically disconnected when the client has no I/Os for a long time and is automatically re-established when there are I/Os.	≥ 0	SFS Turbo file system	1 minute
data_read_io_bytes	Read Bandwidth	Data read I/O load Unit: byte/s	≥ 0 bytes/s	SFS Turbo file system	1 minute
data_write_io_bytes	Write Bandwidth	Data write I/O load Unit: byte/s	≥ 0 bytes/s	SFS Turbo file system	1 minute
metadata_io_bytes	Metadata Read and Write Bandwidth	Metadata read and write I/O load Unit: byte/s	≥ 0 bytes/s	SFS Turbo file system	1 minute
total_io_bytes	Total Bandwidth	Total I/O load Unit: byte/s	≥ 0 bytes/s	SFS Turbo file system	1 minute
iops	IOPS	I/O operations per unit time	≥ 0	SFS Turbo file system	1 minute
used_capacity	Used Capacity	Used capacity of a file system Unit: byte	≥ 0 bytes	SFS Turbo file system	1 minute
used_capacity_percent	Capacity Usage	Percentage of used capacity in the total capacity Unit: percent	0% to 100%	SFS Turbo file system	1 minute

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
used_inode	Used inodes	Number of inodes used in a file system	≥ 1	SFS Turbo file system	1 minute
used_inode_percent	Inode Usage	Percentage of used inodes to total inodes in a file system Unit: percent	0% to 100%	SFS Turbo file system	1 minute

Dimension

Key	Value
efs_instance_id	Instance

Viewing Monitoring Statistics

Step 1 Log in to the console.

Step 2 View the monitoring graphs using either of the following methods.

- Method 1: Choose **Service List > Storage > Scalable File Service**. In the file system list, click **View Metric** in the **Operation** column of the desired file system.
- Method 2: Choose **Management & Deployment > Cloud Eye > Cloud Service Monitoring > SFS Turbo**. In the file system list, click **View Metric** in the **Operation** column of the desired file system.

Step 3 View the SFS Turbo file system monitoring data by metric or monitored duration.

For more information about Cloud Eye, see the *User Guide*.

----End

7.2 Auditing SFS Turbo File Systems Using CTS

7.2.1 Supported SFS Turbo Operations

Scenarios

Cloud Trace Service (CTS) records operations performed on SFS Turbo file systems, facilitating query, audit, and backtracking.

Prerequisites

You have enabled CTS and the tracker is normal. For details about how to enable CTS, see [Enabling CTS](#) in the *Cloud Trace Service Getting Started*.


Operations

Table 7-2 SFS Turbo operations traced by CTS

Operation	Resource Type	Trace
Creating a file system	sfs_turbo	createShare
Deleting a file system	sfs_turbo	deleteShare

Querying Traces

Step 1 Log in to the console.

Step 2 Click  in the upper left corner and select a region.

Step 3 Choose **Management & Deployment > Cloud Trace Service**.

The **Cloud Trace Service** page is displayed.

Step 4 In the navigation pane on the left, choose **Trace List**.

Step 5 On the trace list page, set **Trace Source**, **Resource Type**, and **Search By**, and click **Query** to query the specified traces.

For details about other operations, see section "Querying Real-Time Traces" in the *Cloud Trace Service User Guide*.

----End

Disabling or Enabling a Tracker

The following procedure describes how to disable a tracker on the CTS console. After the tracker is disabled, CTS will stop recording operations, but you can still view existing operation records.

Step 1 Log in to the console.

Step 2 Choose **> Cloud Trace Service**.

The **Cloud Trace Service** page is displayed.

Step 3 Choose **Tracker List** in the left navigation pane.

Step 4 Find the tracker you want to disable, and click **Disable** in the **Operation** column.

Step 5 Click **OK**.

Step 6 After the tracker is disabled, the available operation changes from **Disable** to **Enable**. To re-enable the tracker, click **Enable** and then click **OK**. CTS will start recording operations again.

----End

8 Typical Applications

8.1 Enterprise Website/App Background

Context

For I/O-intensive website services, SFS Turbo can provide shared website source code directories and storage for multiple web servers, enabling low-latency and high-IOPS concurrent share access. Features of such services are as follows:

- Massive small files: Static website files need to be stored, including HTML files, JSON files, and static images.
- Intensive read I/Os: Heavy read of small files, less data writes
- Concurrent access: Multiple web servers access an SFS Turbo background for high availability of website services.

Configuration Process

1. Sort out the website files.
2. Log in to the SFS Turbo console and create an SFS Turbo file system to store the website files.
3. Log in to the cloud server that functions as the compute node and mount the file system.
4. On the head node, upload the files to the file system.
5. Start the web server.

Prerequisites

- A VPC has been created.
- that function as the head node and compute node have been created, and are in the created VPC.
- SFS Turbo has been enabled.

Example Configuration

- Step 1** Log in to the SFS Turbo console.
 - Step 2** In the upper right corner of the page, click **Create File System**.
 - Step 3** On the **Create File System** page, configure parameters as instructed.
 - Step 4** To mount a file system to Linux ECSs, see [Mounting an NFS File System to ECSs \(Linux\)](#).
 - Step 5** Log in to the head node and upload the files to the file system.
 - Step 6** Start the web server.
- End

8.2 Log Printing

Context

SFS Turbo can provide multiple service nodes for shared log output directories, facilitating log collection and management of distributed applications. Features of such services are as follows:

- **Sharing:** A file system is mounted to multiple service hosts and logs are printed concurrently.
- **Large file size and small I/Os:** The size of a single log file is large, but the I/O of each log write is small.
- **Intensive write I/Os:** Most service I/Os are write I/Os of small blocks.

Configuration Process

1. Log in to the SFS Turbo console and create an SFS Turbo file system to store the log files.
2. Log in to the cloud server that functions as the compute node and mount the file system.
3. Configure the file system path as the log directory. It is recommended that each host use different log files.
4. Start applications.

Prerequisites

- A VPC has been created.
- that function as the head node and compute node have been created, and are in the created VPC.
- SFS Turbo has been enabled.

Example Configuration

- Step 1** Log in to the SFS Turbo console.
- Step 2** In the upper right corner of the page, click **Create File System**.

- Step 3** On the **Create File System** page, configure parameters as instructed.
 - Step 4** To mount a file system to Linux ECSs, see [Mounting an NFS File System to ECSs \(Linux\)](#).
 - Step 5** Configure the file system path as the log directory. It is recommended that each host use different log files.
 - Step 6** Start applications.
- End

9 Other Operations

9.1 Testing SFS Turbo Performance

Fio is an open-source I/O tester. You can use fio to test the throughput and IOPS of SFS Turbo file systems.

Prerequisites

Fio has been installed on the cloud server. You can download fio from [the official website](#) or [GitHub](#).

Note and Description

The test performance depends on the network bandwidth between the client and server, as well as the capacity of the file system.

Installing fio

The following uses a Linux CentOS system as an example:

1. Download fio.
yum install fio
2. Install the libaio engine.
yum install libaio-devel
3. Check the fio version.
fio --version

Common Test Configuration Example

NOTE

The following estimated values are obtained from the test on a single ECS. You are advised to use multiple ECSs to test the [SFS Turbo](#) performance.

In the following examples, SFS Turbo Performance and cloud servers with the following specifications are used for illustration.

Specifications: General computing-plus | c3.xlarge.4 | 4 vCPUs | 16 GB

Image: CentOS 7.5 64-bit

- fio command:

```
fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log --direct=1 --filename=/mnt/nfs/test_fio --bs=1M --iodepth=128 --size=10240M --readwrite=rw --rwmixwrite=30 --fallocate=none
```

 NOTE

`/mnt/nfs/test_fio` indicates the location of the file to be tested. The location must be specific to the file name, which is the `test_fio` file in the `/mnt/nfs` directory in this example. Set it based on the site requirements.

- fio result:

```
test: (groupid=0, jobs=1): err=0: pid=10110: Mon Jun 8 11:48:57 2020
read: IOPS=7423, BW=28.0MiB/s (30.4MB/s)(7167MiB/247160msec)
slat (msec): min=1234, max=397477, avg=3145.45, stdev=3344.48
clat (msec): min=245, max=133325, avg=11162.10, stdev=12136.31
lat (msec): min=252, max=133330, avg=11166.32, stdev=12136.34
clat percentiles (msec):
| 1.00th=[ 2245],  5.00th=[ 2540], 10.00th=[ 2671], 20.00th=[ 2900],
| 30.00th=[ 3130], 40.00th=[ 3450], 50.00th=[ 4293], 60.00th=[ 7832],
| 70.00th=[13173], 80.00th=[19792], 90.00th=[20443], 95.00th=[36439],
| 99.00th=[53216], 99.50th=[60031], 99.90th=[79160], 99.95th=[85459],
| 99.99th=[90042]
bw ( KIB/s): min=16600, max=45560, per=100.00%, avg=29696.00, stdev=5544.46, samples=494
jobs   : min= 4150, max=11390, avg=7424.01, stdev=1306.11, samples=494
write: IOPS=3182, BW=12.4MiB/s (13.0MB/s)(3073MiB/247160msec)
slat (msec): min=1400, max=302730, avg=4613.59, stdev=3359.60
clat (msec): min=1447, max=140666, avg=14166.05, stdev=13373.72
lat (msec): min=1457, max=140671, avg=14170.73, stdev=13373.74
clat percentiles (msec):
| 1.00th=[  41],  5.00th=[  41], 10.00th=[  41], 20.00th=[  51],
| 30.00th=[  51], 40.00th=[  61], 50.00th=[  81], 60.00th=[ 141],
| 70.00th=[ 101], 80.00th=[ 241], 90.00th=[ 331], 95.00th=[ 421],
| 99.00th=[ 591], 99.50th=[ 671], 99.90th=[ 871], 99.95th=[ 941],
| 99.99th=[ 1221]
bw ( KIB/s): min= 7144, max=19600, per=100.00%, avg=12730.90, stdev=2395.77, samples=74
jobs   : min= 1706, max= 4900, avg=3182.70, stdev=590.96, samples=494
lat (msec) : 250=0.01%, 500=0.01%, 750=0.01%, 1000=0.01%
lat (msec) : 2=0.20%, 4=39.15%, 10=21.01%, 20=17.92%, 50=20.06%
lat (msec) : 100=1.62%, 250=0.02%
cpu       : usr=1.35%, sys=6.43%, ctx=1072910, majf=0, minf=30
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=1034036,706604,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=120

Run status group 0 (all jobs):
READ: bw=28.0MiB/s (30.4MB/s), 28.0MiB/s-28.0MiB/s (30.4MB/s-30.4MB/s), io=7167MiB (7515MB), run=247160-247160msec
WRITE: bw=12.4MiB/s (13.0MB/s), 12.4MiB/s-12.4MiB/s (13.0MB/s-13.0MB/s), io=3073MiB (3222MB), run=247160-247160msec
```

- fio command:

```
fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log --direct=1 --filename=/mnt/nfs/test_fio --bs=1M --iodepth=128 --size=10240M --readwrite=rw --rwmixwrite=70 --fallocate=none
```

 NOTE

`/mnt/nfs/test_fio` indicates the location of the file to be tested. The location must be specific to the file name, which is the `test_fio` file in the `/mnt/nfs` directory in this example. Set it based on the site requirements.

- fio result:

```

test: (groupid=0, jobs=1): err= 0: pid=20350: Mon Jun 8 11:57:14 2020
read: IOPS=5065, BW=19.8MiB/s (20.7MB/s)(3073MiB/155200msec)
slat (nsec): min=1271, max=269500, avg=4073.51, stdev=3040.12
clat (usec): min=226, max=80185, avg=5711.35, stdev=7079.46
lat (usec): min=232, max=80187, avg=5715.49, stdev=7079.48
clat percentiles (usec):
| 1.00th=[ 1221], 5.00th=[ 1950], 10.00th=[ 2100], 20.00th=[ 2442],
| 30.00th=[ 2606], 40.00th=[ 2802], 50.00th=[ 2999], 60.00th=[ 3220],
| 70.00th=[ 3687], 80.00th=[ 5604], 90.00th=[14222], 95.00th=[21890],
| 99.00th=[35914], 99.50th=[40633], 99.90th=[51643], 99.95th=[55837],
| 99.99th=[66047]
bw ( KIB/s): min=13360, max=28848, per=99.99%, avg=20257.97, stdev=2913.05, samples=310
iops      : min= 3340, max= 7212, avg=5064.48, stdev=720.27, samples=310
write: IOPS=11.8k, BW=46.2MiB/s (48.4MB/s)(7167MiB/155200msec)
slat (nsec): min=1396, max=390604, avg=4405.68, stdev=3091.75
clat (usec): min=857, max=140259, avg=8377.47, stdev=8400.15
lat (usec): min=867, max=140264, avg=8382.02, stdev=8400.16
clat percentiles (nsec):
| 1.00th=[  31], 5.00th=[  41], 10.00th=[  41], 20.00th=[  41],
| 30.00th=[  51], 40.00th=[  51], 50.00th=[  51], 60.00th=[  61],
| 70.00th=[  71], 80.00th=[ 131], 90.00th=[ 211], 95.00th=[ 201],
| 99.00th=[ 421], 99.50th=[ 471], 99.90th=[ 601], 99.95th=[ 601],
| 99.99th=[ 1201]
bw ( KIB/s): min=32224, max=67456, per=99.90%, avg=47254.23, stdev=6792.41, samples=310
iops      : min= 8056, max=16864, avg=11813.55, stdev=1690.11, samples=310
lat (usec) : 250=0.01%, 500=0.04%, 750=0.07%, 1000=0.09%
lat (msec) : 2=1.53%, 4=36.85%, 10=41.27%, 20=11.30%, 50=0.61%
lat (msec) : 100=0.23%, 250=0.01%
cpu       : usr=2.13%, sys=9.90%, ctx=925770, majf=0, minf=31
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=706597,1834043,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=120

Run status group 0 (all jobs):
READ: bw=19.8MiB/s (20.7MB/s), 19.8MiB/s-19.8MiB/s (20.7MB/s-20.7MB/s), io=3073MiB (3222MB), run=155200-155200msec
WRITE: bw=46.2MiB/s (48.4MB/s), 46.2MiB/s-46.2MiB/s (48.4MB/s-48.4MB/s), io=7167MiB (7516MB), run=155200-155200msec

```

Sequential read IOPS

- fio command:

```

fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --rw=read
--bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10

```

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```

test: (groupid=0, jobs=1): err= 0: pid=20459: Mon Jun 8 12:20:18 2020
read: IOPS=9654, BW=37.7MiB/s (39.5MB/s)(10.0GiB/271519msec)
slat (nsec): min=1233, max=662160, avg=4118.17, stdev=4773.23
clat (usec): min=365, max=131116, avg=13253.10, stdev=13950.09
lat (usec): min=371, max=131118, avg=13257.29, stdev=13950.09
clat percentiles (usec):
| 1.00th=[ 1762], 5.00th=[ 1991], 10.00th=[ 2147], 20.00th=[ 2376],
| 30.00th=[ 2704], 40.00th=[ 3621], 50.00th=[ 7767], 60.00th=[ 11994],
| 70.00th=[ 16909], 80.00th=[ 23462], 90.00th=[ 33162], 95.00th=[ 41681],
| 99.00th=[ 59507], 99.50th=[ 66847], 99.90th=[ 83362], 99.95th=[ 90702],
| 99.99th=[103285]
bw ( KIB/s): min=10656, max=61576, per=99.99%, avg=30615.41, stdev=7703.32, samples=543
iops      : min= 4664, max=15394, avg=9653.02, stdev=1925.03, samples=543
lat (usec) : 500=0.01%, 750=0.01%, 1000=0.02%
lat (msec) : 2=5.25%, 4=36.35%, 10=12.76%, 20=20.56%, 50=22.62%
lat (msec) : 100=2.42%, 250=0.02%
cpu       : usr=1.04%, sys=5.35%, ctx=913130, majf=0, minf=159
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=2621440,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=120

Run status group 0 (all jobs):
READ: bw=37.7MiB/s (39.5MB/s), 37.7MiB/s-37.7MiB/s (39.5MB/s-39.5MB/s), io=10.0GiB (10.7GB), run=2

```

Random read IOPS

- fio command:

```

fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --

```

rw=randread --bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (g=0): rw=randread, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
fio-2.1.10
Starting 1 process
Jobs: 1 (f=1): [r] [100.0% done] [17824KB/0KB/0KB /s] [4456/0/0 iops] [eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=20755: Tue Dec 28 09:41:43 2021
read : io=10240MB, bw=18597KB/s, iops=4649, runt=563832msec
slat (usec): min=1, max=375, avg= 2.64, stdev= 2.52
clat (usec): min=715, max=755902, avg=27527.31, stdev=106233.39
lat (usec): min=718, max=755903, avg=27530.03, stdev=106233.39
clat percentiles (msec):
| 1.00th=[ 3], 5.00th=[ 5], 10.00th=[ 6], 20.00th=[ 6],
| 30.00th=[ 7], 40.00th=[ 7], 50.00th=[ 8], 60.00th=[ 9],
| 70.00th=[ 11], 80.00th=[ 15], 90.00th=[ 21], 95.00th=[ 28],
| 99.00th=[ 676], 99.50th=[ 693], 99.90th=[ 725], 99.95th=[ 734],
| 99.99th=[ 750]
bw (KB /s): min= 1896, max=35752, per=100.00%, avg=18605.56, stdev=1980.86
lat (usec) : 750=0.01%, 1000=0.01%
lat (msec) : 2=0.32%, 4=3.28%, 10=63.65%, 20=22.42%, 50=7.50%
lat (msec) : 100=0.07%, 250=0.01%, 500=0.03%, 750=2.72%, 1000=0.01%
cpu        : usr=0.82%, sys=2.41%, ctx=1231561, majf=0, minf=155
IO depths  : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit     : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued    : total=r=2621440/w=0/d=0, short=r=0/w=0/d=0
latency    : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
READ: io=10240MB, agrb=18597KB/s, minb=18597KB/s, maxb=18597KB/s, mint=563832msec, maxt=563832msec
```

Sequential write IOPS

- fio command:

fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --rw=write --bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (groupid=0, jobs=1): err= 0: pid=20874: Mon Jun  8 14:23:09 2020
write: IOPS=11.0k, BW=43.1MiB/s (45.2MB/s)(10.06GiB/237436msec)
slat (msec): min=1483, max=368726, avg=4388.87, stdev=3688.87
clat (usec): min=1953, max=186548, avg=11588.61, stdev=5876.84
lat (usec): min=1959, max=186552, avg=11593.86, stdev=5876.86
clat percentiles (usec):
| 1.00th=[ 4015], 5.00th=[ 5932], 10.00th=[ 6652], 20.00th=[ 7439],
| 30.00th=[ 8029], 40.00th=[ 8848], 50.00th=[ 9634], 60.00th=[10814],
| 70.00th=[12518], 80.00th=[15533], 90.00th=[19268], 95.00th=[22676],
| 99.00th=[32637], 99.50th=[37487], 99.90th=[49821], 99.95th=[53748],
| 99.99th=[69731]
bw ( KiB/s): min=31712, max=52431, per=99.99%, avg=44158.84, stdev=3987.31, samples=474
iops       : min= 7928, max=13187, avg=11839.58, stdev=996.83, samples=474
lat (msec) : 2=0.01%, 4=1.00%, 10=51.94%, 20=38.58%, 50=0.39%
lat (msec) : 100=0.00%, 250=0.01%
cpu        : usr=1.33%, sys=5.47%, ctx=392117, majf=0, minf=27
IO depths  : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit     : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=r=8,2621448,0,0 short=r=0,0,0,0 dropped=0,0,0,0
latency    : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
WRITE: bw=43.1MiB/s (45.2MB/s), 43.1MiB/s-43.1MiB/s (45.2MB/s-45.2MB/s), io=10.06GiB (10.7GB), run=
```

Random write IOPS

- fio command:
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --rw=randwrite --bs=4k --size=1G --iodepth=128 --runtime=120 --numjobs=10

 NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (g=0): rw=randwrite, bs=4K-4K/4K-4K/4K-4K, ioengine=libaio, iodepth=128
fio-2.1.10
Starting 1 process

test: (groupid=0, jobs=1): err= 0: pid=16622: Thu Jan 13 10:13:22 2022
write: io=10240MB, bw=18463KB/s, iops=4615, runt=567947msec
slat (usec): min=1, max=356, avg= 3.21, stdev= 2.04
clat (usec): min=890, max=815560, avg=27727.54, stdev=101207.14
lat (usec): min=893, max=815564, avg=27730.83, stdev=101207.14
clat percentiles (msec):
| 1.00th=[ 4], 5.00th=[ 6], 10.00th=[ 6], 20.00th=[ 7],
| 30.00th=[ 7], 40.00th=[ 8], 50.00th=[ 8], 60.00th=[ 10],
| 70.00th=[ 13], 80.00th=[ 16], 90.00th=[ 23], 95.00th=[ 30],
| 99.00th=[ 644], 99.50th=[ 668], 99.90th=[ 701], 99.95th=[ 709],
| 99.99th=[ 734]
bw (KB /s): min=1064, max=36589, per=100.00%, avg=18469.11, stdev=3769.64
lat (usec): 1000=0.01%
lat (msec): 2=0.20%, 4=1.85%, 10=60.93%, 20=24.30%, 50=9.85%
lat (msec): 100=0.09%, 250=0.01%, 500=0.08%, 750=2.68%, 1000=0.01%
cpu : usr=0.98%, sys=2.90%, ctx=1552744, majf=0, minf=27
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued : total=r=0/w=2621440/d=0, short=r=0/w=0/d=0
latency : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
WRITE: io=10240MB, aggrb=18462KB/s, minb=18462KB/s, maxb=18462KB/s, mint=567947msec, maxt=567947msec
```

Sequential read bandwidth

- fio command:
fio --randrepeat=1 --ioengine=libaio --name=test -output=output.log --direct=1 --filename=/mnt/sfs-turbo/test_fio --bs=1M --iodepth=128 --size=10240M --readwrite=read --fallocate=none

 NOTE

`/mnt/sfs-turbo/test_fio` is the location of the file to be tested. The location must be specific to the file name, which is the `test_fio` file in the `/mnt/sfs-turbo` directory in this example. Set it based on the site requirements.

- fio result:

```
test: (groupid=0, jobs=1): err= 0: pid=28962: Mon Jun 8 14:37:48 2020
read: IOPS=398, BW=391MiB/s (489MB/s)(10.8GiB/26221msec)
slat (usec): min=78, max=595, avg=99.58, stdev=39.89
clat (msec): min=35, max=544, avg=327.38, stdev=99.64
lat (msec): min=36, max=545, avg=327.48, stdev=99.63
clat percentiles (msec):
| 1.00th=[ 155], 5.00th=[ 161], 10.00th=[ 167], 20.00th=[ 188],
| 30.00th=[ 368], 40.00th=[ 372], 50.00th=[ 380], 60.00th=[ 384],
| 70.00th=[ 388], 80.00th=[ 393], 90.00th=[ 401], 95.00th=[ 414],
| 99.00th=[ 472], 99.50th=[ 586], 99.90th=[ 535], 99.95th=[ 542],
| 99.99th=[ 542]
bw ( KiB/s): min=381856, max=768000, per=99.52%, avg=397987.65, stdev=81583.56, samples=52
iops : min= 294, max= 758, avg=388.65, stdev=79.67, samples=52
lat (msec): 50=0.17%, 100=0.28%, 250=27.61%, 500=71.37%, 750=0.58%
cpu : usr=0.80%, sys=4.21%, ctx=18395, majf=0, minf=97
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=10240,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
READ: bw=391MiB/s (489MB/s), 391MiB/s-391MiB/s (489MB/s-489MB/s), io=10.8GiB (10.7GB), run=26221-26221msec
```


Random read bandwidth

- fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --rw=randread --bs=1M --size=10G --iodepth=128 --runtime=120 --numjobs=1
```

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (g=0): rw=randread, bs=1M-1M/1M-1M/1M-1M, ioengine=libaio, iodepth=128
fio-2.1.10
Starting 1 process

test: (groupid=0, jobs=1): err= 0: pid=14261: Tue Dec 28 09:18:04 2021
read : io=10240MB, bw=154130KB/s, iops=150, runt= 68032msec
slat (usec): min=61, max=8550, avg=142.99, stdev=187.96
clat (msec): min=12, max=2002, avg=849.91, stdev=347.27
lat (msec): min=12, max=2003, avg=850.05, stdev=347.26
clat percentiles (msec):
| 1.00th=[ 47], 5.00th=[ 84], 10.00th=[ 105], 20.00th=[ 914],
| 30.00th=[ 947], 40.00th=[ 963], 50.00th=[ 971], 60.00th=[ 988],
| 70.00th=[ 996], 80.00th=[ 1012], 90.00th=[ 1037], 95.00th=[ 1057],
| 99.00th=[ 1876], 99.50th=[ 1926], 99.90th=[ 1975], 99.95th=[ 1975],
| 99.99th=[ 2008]
bw (KB /s): min=69974, max=167768, per=98.85%, avg=152360.15, stdev=10783.47
lat (msec) : 20=0.33%, 50=0.80%, 100=7.02%, 250=7.95%, 1000=55.30%
lat (msec) : 2000=28.57%, >=2000=0.02%
cpu       : usr=0.02%, sys=1.93%, ctx=4399, majf=0, minf=602
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued   : total=r=10240/w=0/d=0, short=r=0/w=0/d=0
latency   : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
  READ: io=10240MB, aggrb=154129KB/s, minb=154129KB/s, maxb=154129KB/s, mint=68032msec, max
t=68032msec
```

Sequential write bandwidth

- fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --rw=write --bs=1M --size=10G --iodepth=128 --runtime=120 --numjobs=1
```

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (groupid=0, jobs=1): err= 0: pid=21889: Mon Jun 8 14:53:44 2020
write: IOPS=243, BW=244MiB/s (255MB/s)(10.0GiB/42048msec)
slat (usec): min=103, max=504, avg=190.38, stdev=29.47
clat (msec): min=18, max=1104, avg=525.23, stdev=253.35
lat (msec): min=18, max=1104, avg=525.42, stdev=253.35
clat percentiles (msec):
| 1.00th=[ 51], 5.00th=[ 108], 10.00th=[ 167], 20.00th=[ 292],
| 30.00th=[ 422], 40.00th=[ 468], 50.00th=[ 506], 60.00th=[ 558],
| 70.00th=[ 625], 80.00th=[ 768], 90.00th=[ 902], 95.00th=[ 978],
| 99.00th=[ 1036], 99.50th=[ 1045], 99.90th=[ 1078], 99.95th=[ 1099],
| 99.99th=[ 1099]
bw ( KiB/s): min= 4896, max=468992, per=100.00%, avg=249588.99, stdev=147656.62, samples=83
iops      : min=    4, max=  458, avg=243.63, stdev=144.22, samples=83
lat (msec) : 20=0.03%, 50=0.96%, 100=3.36%, 250=12.55%, 500=31.63%
lat (msec) : 750=30.07%, 1000=18.96%
cpu       : usr=2.28%, sys=2.50%, ctx=3972, majf=0, minf=27
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=0,10240,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
WRITE: bw=244MiB/s (255MB/s), 244MiB/s-244MiB/s (255MB/s-255MB/s), io=10.0GiB (10.7GB), run=42048-42048msec
```

Random write bandwidth

- fio command:

```
fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --
group_reporting=1 --name=iops_fio --directory=/mnt/sfs-turbo/ --
rw=randwrite --bs=1M --size=10G --iodepth=128 --runtime=120 --
numjobs=1
```

NOTE

Variable `/mnt/sfs-turbo/` is the location of the file to be tested. The location must be specific to the file name. Set it to the actual file name.

- fio result:

```
test: (g=0): rw=randwrite, bs=1M-1M/1M-1M/1M-1M, ioengine=Libaio, iodepth=128
fio-2.1.10
Starting 1 process

test: (groupid=0, jobs=1): err= 0: pid=16370: Tue Dec 28 09:22:59 2021
write: io=10240MB, bw=156000KB/s, iops=152, runt= 67216msec
slat (usec): min=93, max=349, avg=156.14, stdev=22.29
clat (msec): min=17, max=1964, avg=839.92, stdev=345.94
lat (msec): min=17, max=1964, avg=840.08, stdev=345.94
clat percentiles (msec):
| 1.00th=[ 30], 5.00th=[ 37], 10.00th=[ 42], 20.00th=[ 97],
| 30.00th=[ 97], 40.00th=[ 98], 50.00th=[ 98], 60.00th=[ 99],
| 70.00th=[ 99], 80.00th=[ 100], 90.00th=[ 100], 95.00th=[ 101],
| 99.00th=[ 102], 99.50th=[ 102], 99.90th=[ 103], 99.95th=[ 104],
| 99.99th=[ 195]
bw (KB /s): min=150104, max=180654, per=98.76%, avg=154058.04, stdev=3404.48
lat (msec) : 20=0.04%, 50=13.44%, 100=1.04%, 250=0.73%, 500=1.05%
lat (msec) : 750=0.04%, 1000=60.69%, 2000=22.97%
cpu       : usr=0.91%, sys=1.52%, ctx=2011, majf=0, minf=28
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued   : total=r=0/w=10240/d=0, short=r=0/w=0/d=0
latency   : target=0, window=0, percentile=100.00%, depth=128

Run status group 0 (all jobs):
WRITE: io=10240MB, aggrb=156000KB/s, minb=156000KB/s, maxb=156000KB/s, mint=67216msec, maxt=67216msec
```

9.2 Mounting a File System to a Linux as a Non-root User

Scenarios

By default, a Linux ECS allows only the **root** user to use the **mount** command to mount file systems, but you can grant the permissions of user **root** to other users.

Such users can then use the **mount** command to mount file systems. The following describes how to mount a file system to a Linux ECS as a common user. EulerOS is used in this example.

Prerequisites

- A non-**root** user has been created on the ECS.
- A file system has been created and can be mounted to the ECS as **root**.
- The shared path of the file system has been obtained.

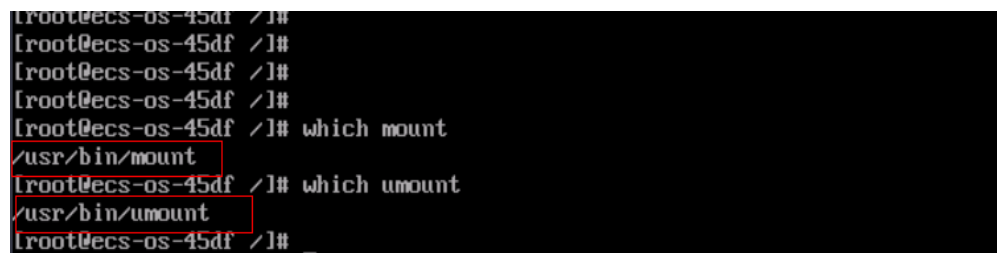
Procedure

Step 1 Log in to the ECS as user **root**.

Step 2 Assign the permissions of user **root** to a non-**root** user.

1. Run **chmod 777 /etc/sudoers** to make the **sudoers** file editable.
2. Use the **which** command to view the **mount** and **umount** command paths.

Figure 9-1 Viewing command paths



```
root@ecs-os-45df ~/#  
root@ecs-os-45df ~/#  
root@ecs-os-45df ~/#  
root@ecs-os-45df ~/#  
root@ecs-os-45df ~/# which mount  
/usr/bin/mount  
root@ecs-os-45df ~/# which umount  
/usr/bin/umount  
root@ecs-os-45df ~/#
```

3. Run **vi /etc/resolv.conf** to edit the **sudoers** file.
4. Add a common user under the **root** account. In this example, user **mike** is added.

Figure 9-2 Adding a user

```
# Defaults    env_keep += "HOME"

Defaults    secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
mike    ALL=(ALL)    NOPASSWD: /usr/bin/mount
mike    ALL=(ALL)    NOPASSWD: /usr/bin/umount

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)    ALL

## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
```

5. Press **Esc**, enter **:wq**, and press **Enter** to save and exit.
6. Run **chmod 440 /etc/sudoers** to make the **sudoers** file read-only.

Step 3 Log in to the ECS as user **mike**.

Step 4 Run the following command to mount the file system. For details about the mount parameters, see [Table 9-1](#).

sudo mount -t nfs -o vers=3,timeo=600,noresvport,nolock Shared path Local path

Table 9-1 Parameter description

Parameter	Description
<i>Shared path</i>	The format is <i>File system IP address:/</i> , for example, 192.168.0.0:/ . NOTE Variable <i>x</i> is a digit or letter. If the shared path is too long to display completely, you can adjust the column width.
<i>Local path</i>	Local path on the used to mount the file system, for example, /local_path .

Step 5 View the mounted file system.

mount -l

If the command output contains the following information, the file system has been mounted:


```
example.com:/share-xxx on /local_path type nfs (rw,vers=3,timeo=600,nolock,addr=)
```

----End

9.3 Mounting a Subdirectory of an NFS File System to ECSs (Linux)

This section describes how to mount a subdirectory of an NFS file system to Linux ECSs.

Prerequisites

You have mounted the file system to a Linux ECS by referring to [Mounting an NFS File System to ECSs \(Linux\)](#).

Procedure

Step 1 Create a subdirectory in the local path.

```
mkdir Local_path/Subdirectory
```

NOTE

Variable *Local_path* is a local directory on the ECS used to mount the file system, for example, **/local_path**. Specify the local path used to mount the root directory.

Step 2 Mount the subdirectory to the ECSs that are in the same VPC as the file system. You can mount the file system to Linux ECSs using NFSv3 only.

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Domain_name_or_IP  
address_of_the_file_system:/Subdirectory Local_path
```

NOTE

- *Domain name or IP address of the file system*: You can obtain it from the file system list or details page on the console.
 - SFS Turbo: *xx.xx.xx.xx;/subdirectory*
- *Subdirectory* is the subdirectory created in the previous step.
- *Local_path* is an ECS local directory where the file system is mounted, for example, **/local_path**. Specify the local path used to mount the root directory.

Step 3 View the mounted file system.

```
mount -l
```

If the command output contains the following information, the file system has been mounted:

```
Shared_path on /local_path type nfs (rw,vers=3,timeo=600,nolock,addr=)
```

Step 4 Check that you can access the subdirectory on the ECSs to read or write data.

----End

Troubleshooting

If a subdirectory is not created before mounting, the mount will fail.

Figure 9-3 Mounting without a subdirectory created

```
[root@ecs-eos-0891 workstation]# mount -t nfs -o nolock,vers=3 [redacted] -vvv
mount.nfs: timeout set for Sun Oct 24 20:44:13 2021
mount.nfs: trying text-based options 'nocheck,vers=3,addr=[redacted]'
mount.nfs: prog 100003, trying vers=3, prot=6
mount.nfs: trying [redacted] prog 100003 vers 3 prot TCP port 2049
mount.nfs: prog 100005, trying vers=3, prot=17
mount.nfs: trying [redacted] prog 100005 vers 3 prot UDP port 20048
mount.nfs: mount(2): Permission denied
mount.nfs: access denied by server while mounting [redacted] :/subdir
```

In the preceding figure, the root directory does not have the **subdir** subdirectory created so that the mount fails. In this case, error message "Permission denied" is reported.

To troubleshoot this issue, mount the root directory, create a subdirectory, and then mount the subdirectory.

Figure 9-4 Mounting a subdirectory

```
[root@ecs-eos-0891 workstation]# mount -t nfs -o nolock,vers=3 [redacted] .82:/mnt/sfsturbo -vvv
mount.nfs: timeout set for Sun Oct 24 20:47:26 2021
mount.nfs: trying text-based options 'nocheck,vers=3,addr=[redacted] .82' Mount the root directory.
mount.nfs: prog 100003, trying vers=3, prot=6
mount.nfs: trying [redacted] .82 prog 100003 vers 3 prot TCP port 2049
mount.nfs: prog 100005, trying vers=3, prot=17
mount.nfs: trying [redacted] .82 prog 100005 vers 3 prot UDP port 20048
[root@ecs-eos-0891 workstation]# mkdir /mnt/sfsturbo/subdir Create a subdirectory.
[root@ecs-eos-0891 workstation]# umount /mnt/sfsturbo
[root@ecs-eos-0891 workstation]# mount -t nfs -o nolock,vers=3 [redacted] .82:/subdir /mnt/sfsturbo -vvv
mount.nfs: timeout set for Sun Oct 24 20:47:50 2021
mount.nfs: trying text-based options 'nocheck,vers=3,addr=[redacted] .82' Mount the subdirectory.
mount.nfs: prog 100003, trying vers=3, prot=6
mount.nfs: trying [redacted] .82 prog 100003 vers 3 prot TCP port 2049
mount.nfs: prog 100005, trying vers=3, prot=17
mount.nfs: trying [redacted] .82 prog 100005 vers 3 prot UDP port 20048
[root@ecs-eos-0891 workstation]#
```

9.4 Data Migration

9.4.1 Migration Description

By default, an SFS Turbo file system can only be accessed by ECSs or CCE containers that reside in the same VPC as the file system. To access an SFS Turbo file system from an on-premises data center or a different VPC, you need to establish network connections by using Direct Connect, VPN, or VPC peering connections.

- Access from on premises or another cloud: Use Direct Connect or VPN.
- Access from a different VPC under the same account and in the same region: Use VPC peering.
- Access from a different account in the same region: Use VPC peering.
- Access from a different region: Use Cloud Connect.

Data can be migrated to SFS Turbo by using an ECS that can access the Internet.

- Mount the SFS Turbo file system to the ECS and migrate data from the local NAS storage to the SFS Turbo file system.

Migrating Data Using Direct Connect

- If communication cannot be enabled through file system mounting, migrate data using the Huawei Cloud ECS via the Internet.

[Migrating Data Using the Internet](#)

9.4.2 Migrating Data Using Direct Connect

Context

You can migrate data from a local NAS to SFS Turbo using Direct Connect.

In this solution, a Linux ECS is created to connect the local NAS and SFS Turbo, and data is migrated to the cloud using this ECS.

You can also refer to this solution to migrate data from an on-cloud NAS to SFS Turbo. For details, see [Migrating Data from On-Cloud NAS to SFS Turbo](#).

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The UID and GID of your file will no longer be consistent after data migration.
- The file access modes will no longer be consistent after data migration.
- Incremental migration is supported, so that only changed data is migrated.

Prerequisites

- You have enabled and configured Direct Connect. For details, see *Direct Connect User Guide*.
- You have created a Linux ECS.
- You have created an SFS Turbo file system and have obtained its shared path.
- You have obtained the shared path of the local NAS.

Procedure

Step 1 Log in to the ECS console.

Step 2 Log in to the Linux ECS.

Step 3 Mount the local NAS to the ECS.

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared path of the local NAS /mnt/src
```

Step 4 Mount the SFS Turbo file system to the ECS.

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared path of the file system /mnt/dst
```

Step 5 Install rclone on the ECS.

```
wget https://downloads.rclone.org/v1.53.4/rclone-v1.53.4-linux-amd64.zip --no-check-certificate
unzip rclone-v1.53.4-linux-amd64.zip
chmod 0755 ./rclone-*/rclone
cp ./rclone-*/rclone /usr/bin/
rm -rf ./rclone-*
```

Step 6 Synchronize data to the SFS Turbo file system.

```
rclone copy /mnt/src /mnt/dst -P --transfers 32 --checkers 64 --links --create-empty-src-dirs
```

 NOTE

The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.

- **--transfers**: number of files that can be transferred concurrently
- **--checkers**: number of local files that can be scanned concurrently
- **-P**: data copy progress
- **--links**: replicates the soft links from the source. They are saved as soft links in the destination.
--copy-links: replicates the content of files to which the soft links point. They are saved as files rather than soft links in the destination.
- **--create-empty-src-dirs**: replicates the empty directories from the source to the destination.

After data synchronization is complete, go to the SFS Turbo file system to check whether data is migrated.

----End

Migrating Data from On-Cloud NAS to SFS Turbo

To migrate data from an on-cloud NAS to your SFS Turbo file system, ensure that the NAS and SFS Turbo file system are in the same VPC, or you have established the network using Cloud Connect.

For details about how to configure Cloud Connect, see *Cloud Connect User Guide*.

9.4.3 Migrating Data Using the Internet

Context

You can migrate data from a local NAS to SFS Turbo using the Internet.

In this solution, to migrate data from the local NAS to the cloud, a Linux server is created both on the cloud and on-premises. The on-premises server is used to access the local NAS, and the ECS is used to access SFS Turbo. Inbound and outbound traffic is allowed on port 22 of these two servers.

You can also refer to this solution to migrate data from an on-cloud NAS to SFS Turbo.

Notes and Constraints

- Only Linux ECSs can be used to migrate data.
- The UID and GID of your file will no longer be consistent after data migration.
- The file access modes will no longer be consistent after data migration.
- Inbound and outbound traffic must be allowed on port 22.
- Incremental migration is supported, so that only changed data is migrated.

Prerequisites

- A Linux server has been created on the cloud and on-premises respectively.
- An EIP has been bound to the ECS to ensure that the two servers can communicate with each other.

- You have created an SFS Turbo file system and have obtained its shared path.
- You have obtained the shared path of the local NAS.

Procedure

Step 1 Log in to the ECS console.

Step 2 Log in to the on-premises server **client1** and run the following command to mount the local NAS:

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared path of the local NAS /mnt/src
```

Step 3 Log in to the Linux ECS **client2** and run the following command to mount the SFS Turbo file system:

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp Shared path of the SFS Turbo file system /mnt/dst
```

Step 4 Run the following commands on **client1** to install rclone:

```
wget https://downloads.rclone.org/v1.53.4/rclone-v1.53.4-linux-amd64.zip --no-check-certificate
unzip rclone-v1.53.4-linux-amd64.zip
chmod 0755 ./rclone-*/rclone
cp ./rclone-*/rclone /usr/bin/
rm -rf ./rclone-*
```

Step 5 Run the following commands on **client1** to configure the environment:

```
rclone config
No remotes found - make a new one
n) New remote
s) Set configuration password
q) Quit config
n/s/q> n
name> remote name (New name)
Type of storage to configure.
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
24 / SSH/SFTP Connection
  \ "sftp"
Storage> 24 (Select the SSH/SFTP number)
SSH host to connect to
Enter a string value. Press Enter for the default ("").
Choose a number from below, or type in your own value
 1 / Connect to example.com
  \ "example.com"
host> ip address (IP address of client2)
SSH username, leave blank for current username, root
Enter a string value. Press Enter for the default ("").
user> user name (Username of client2)
SSH port, leave blank to use default (22)
Enter a string value. Press Enter for the default ("").
port> 22
SSH password, leave blank to use ssh-agent.
y) Yes type in my own password
g) Generate random password
n) No leave this optional password blank
y/g/n> y
Enter the password:
password: (Password for logging in to client2)
Confirm the password:
password: (Confirm the password for logging in to client2)
Path to PEM-encoded private key file, leave blank or set key-use-agent to use ssh-agent.
Enter a string value. Press Enter for the default ("").
key_file> (Press Enter)
The passphrase to decrypt the PEM-encoded private key file.

Only PEM encrypted key files (old OpenSSH format) are supported. Encrypted keys
in the new OpenSSH format can't be used.
y) Yes type in my own password
```

```

g) Generate random password
n) No leave this optional password blank
y/g/n> n
When set forces the usage of the ssh-agent.
When key-file is also set, the ".pub" file of the specified key-file is read and only the associated key is
requested from the ssh-agent. This allows to avoid `Too many authentication failures for *username*` errors
when the ssh-agent contains many keys.
Enter a boolean value (true or false). Press Enter for the default ("false").
key_use_agent> (Press Enter)
Enable the use of the aes128-cbc cipher. This cipher is insecure and may allow plaintext data to be
recovered by an attacker.
Enter a boolean value (true or false). Press Enter for the default ("false").
Choose a number from below, or type in your own value
 1 / Use default Cipher list.
  \ "false"
 2 / Enables the use of the aes128-cbc cipher.
  \ "true"
use_insecure_cipher> (Press Enter)
Disable the execution of SSH commands to determine if remote file hashing is available.
Leave blank or set to false to enable hashing (recommended), set to true to disable hashing.
Enter a boolean value (true or false). Press Enter for the default ("false").
disable_hashcheck>
Edit advanced config? (y/n)
y) Yes
n) No
y/n> n
Remote config
-----
[remote_name]
type = sftp
host=(client2 ip)
user=(client2 user name)
port = 22
pass = *** ENCRYPTED ***
key_file_pass = *** ENCRYPTED ***
-----
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
Current remotes:

Name          Type
=====
remote_name   sftp

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q

```

Step 6 Run the following command to view the **rclone.conf** file in **/root/.config/rclone/rclone.conf**:

```

cat /root/.config/rclone/rclone.conf
[remote_name]
type = sftp
host=(client2 ip)
user=(client2 user name)
port = 22
pass = ***
key_file_pass = ***

```

Step 7 Run the following command on **client1** to synchronize data:

```

rclone copy /mnt/src remote_name:/mnt/dst -P --transfers 32 --checkers 64

```

 **NOTE**

- Replace *remote_name* in the command with the remote name in the environment.
- The parameters are described as follows. Set **transfers** and **checkers** based on the system specifications.
 - **transfers**: number of files that can be transferred concurrently
 - **checkers**: number of local files that can be scanned concurrently
 - **P**: data copy progress

After data synchronization is complete, go to the SFS Turbo file system to check whether data is migrated.

----End