Scalable File Service Turbo

User Guide

Issue	01
Date	2025-01-09

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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 Permissions Management

1.1 Creating an IAM 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.

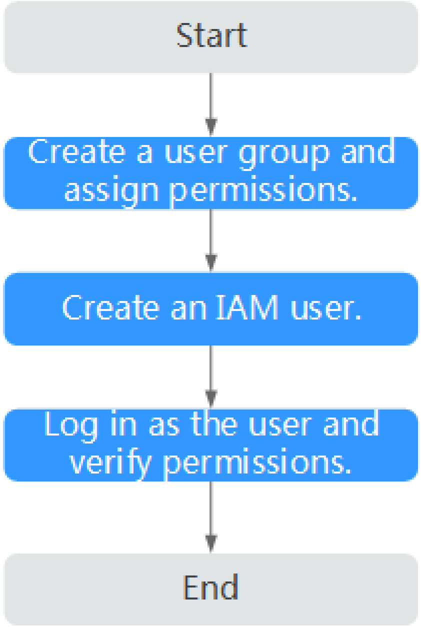
This section describes the procedure for granting user permissions. **Figure 1-1** shows the process flow.

Prerequisites

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

Process Flow

Figure 1-1 Process of granting SFS Turbo permissions


Start
Create a user group and assign permissions.
Create an IAM user.
Log in as the user and verify permissions.
End

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.

2 File System Management

2.1 Creating an SFS Turbo 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

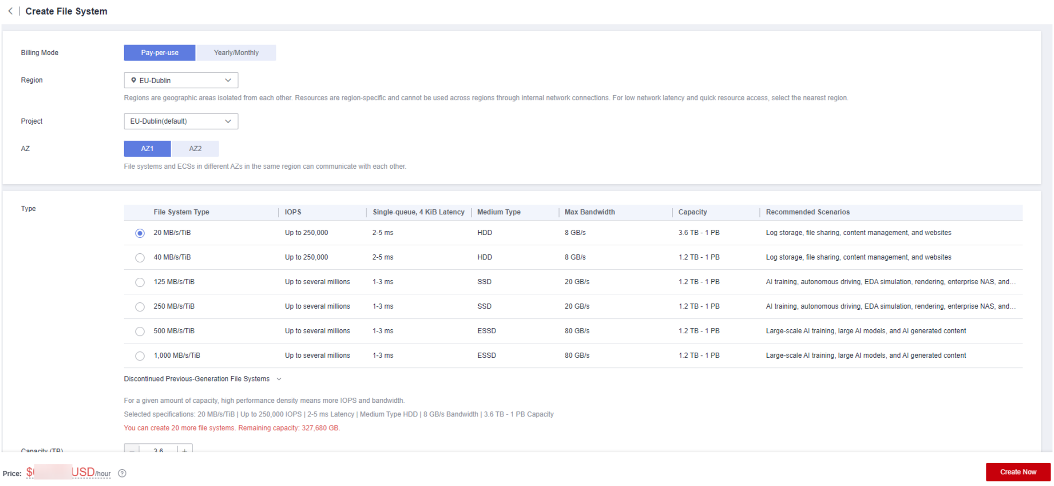
Step 1 Log in to the console using a cloud account.

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 2-1**, as shown in **Figure 2-1**.

Figure 2-1 Creating an SFS Turbo file system



Create File System

Billing Mode: Pay-per-use | Yearly/Monthly

Region: EU-Dublin

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

Project: EU-Dublin(default)

AZ: AZ1 | AZ2

File systems and ECSs in different AZs in the same region can communicate with each other.

Type

	File System Type	IOPS	Single-queue, 4 KiB Latency	Medium Type	Max Bandwidth	Capacity	Recommended Scenarios
●	20 MB/s/TiB	Up to 250,000	2-5 ms	HDD	8 GB/s	3.6 TB - 1 PB	Log storage, file sharing, content management, and websites
○	40 MB/s/TiB	Up to 250,000	2-5 ms	HDD	8 GB/s	1.2 TB - 1 PB	Log storage, file sharing, content management, and websites
○	125 MB/s/TiB	Up to several millions	1-3 ms	SSD	20 GB/s	1.2 TB - 1 PB	AI training, autonomous driving, EDA simulation, rendering, enterprise NAS, and...
○	250 MB/s/TiB	Up to several millions	1-3 ms	SSD	20 GB/s	1.2 TB - 1 PB	AI training, autonomous driving, EDA simulation, rendering, enterprise NAS, and...
○	500 MB/s/TiB	Up to several millions	1-3 ms	ESSD	80 GB/s	1.2 TB - 1 PB	Large-scale AI training, large AI models, and AI generated content
○	1,000 MB/s/TiB	Up to several millions	1-3 ms	ESSD	80 GB/s	1.2 TB - 1 PB	Large-scale AI training, large AI models, and AI generated content

Discontinued Previous-Generation File Systems

For a given amount of capacity, high performance density means more IOPS and bandwidth.

Selected specifications: 20 MB/s/TiB | Up to 250,000 IOPS | 2-5 ms Latency | Medium Type HDD | 8 GB/s Bandwidth | 3.6 TB - 1 PB Capacity

You can create 20 more file systems. Remaining capacity: 327,680 GB.

Capacity (TB): 3.6

Price: $[redacted]USD/hour

Table 2-1 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 of 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: ● 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.	● 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	Mandatory Select a VPC and a subnet. ● 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. **NOTE** ● 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. For details about VPC peering connections, see section "VPC Peering Connection" in *Virtual Private Cloud User Guide*.	-
Enterprise Project	This function is provided for enterprise users. When creating a file system, you can add the file system to an existing enterprise project. An enterprise project makes it easy to manage projects and groups of cloud resources and users. Use the default enterprise project or create one. Select an enterprise project from the drop-down list.	You can select only created enterprise projects. To create an enterprise project, click **Enterprise** in the upper right corner of the console page.

Parameter	Description	Remarks
Encryption	Optional 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: - **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 *Data Encryption Workshop User Guide*. - **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.	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**.

Parameter	Description	Remarks
Security Group	Mandatory 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. When creating an SFS Turbo file system, you can select only one security group. You are advised to use an independent security group for an SFS Turbo file system to isolate it from service nodes. 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 *Virtual Private Cloud User Guide*. 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.	-
Name	Mandatory User-defined file system name.	The name must start with a letter and can contain only letters, digits, underscores (_), and hyphens (-). It must contain at least four characters but no more than 64 characters.

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.

⚠ CAUTION

You can configure alarm policies for the file system inode usage and capacity usage by referring to **Creating an Alarm Rule**. Later, when such an alarm is reported, you can delete data to release space or expand the file system capacity in a timely manner. Otherwise, data can no longer be written to the file system when the capacity is used up, and services will be affected.

----End

2.2 Mounting an SFS Turbo File System

2.2.1 Mounting an SFS Turbo File System to Linux ECSs

After creating a file system, you need to mount it to cloud servers so that they can share the file system. This section describes how to mount a file system to ECSs.

NOTE

- The operations of mounting a file system to BMSs are the same as those of ECSs. To mount a file system to CCE, see **Storage**. Then complete the deployment on the CCE console.
- The mount operation may vary depending on the server OS. Perform the mount operation based on your server OS.

Notes and Constraints

NOTE

The constraints only apply to local paths (mount points) and do not affect other files or directories.

- SFS Turbo file systems cannot be mounted to Windows ECSs.
- The **atime**, **ctime**, and **mtime** attributes of a local path (root directory of the mount point) are the current time of the file system server. Each time a root directory attribute is queried, the current time of the server is returned.
- 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.

Prerequisites

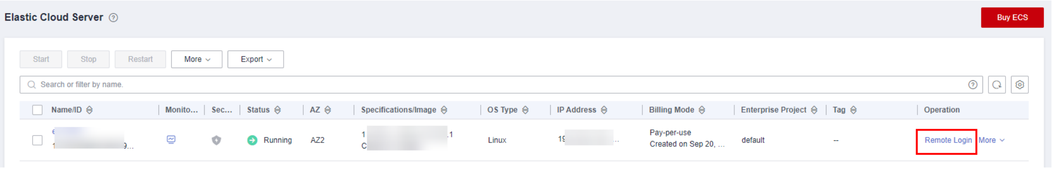
- You have checked the type of the OS on each ECS. Different OSs require different commands to install the NFS client.
- You have created an SFS Turbo file system and obtained its shared path from the file system list.
- At least an ECS is available.

Mounting an SFS Turbo File System

Step 1 Log in to the ECS console.

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

Step 2 Log in to the ECS as user **root**. You can log in using the console or a remote access tool (such as PuTTY).



Name/ID	Monito...	Sec...	Status	AZ	Specifications/Image	OS Type	IP Address	Billing Mode	Enterprise Project	Tag	Operation
[redacted]			Running	AZ2	[redacted]	Linux	[redacted]	Pay-per-use Created on Sep 20, ...	default	--	Remote Login More

NOTE

If you log in as a non-root user, mount the file system by referring to see **How Do I Mount a File System to a Linux as a Non-root User?**

Step 3 Install the NFS client.

1. 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, execute the **installation command**.
   - 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
```

2. 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 Mount the NFS file system.

- To mount the file system root directory, run the following commands:

```
mkdir <Local path>
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp <Shared path> <Local path>
```

- To mount a file system subdirectory, run the following commands:

```
mkdir <Local path>/<Subdirectory>
mount -t nfs -o vers=3,timeo=600,noresvport,nolock,tcp <File system domain name or IP address>:/
<Subdirectory> <Local path>
```

NOTE

- You can obtain the domain name or IP address from the shared path in the file system list.
- If any other resources, such as a disk, have been mounted on the desired 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.)
- A file system can only be mounted to the ECSs that are in the same VPC as the file system.

Table 2-2 describes the mount command variables.

Table 2-2 Mount command variables

Parameter	Description
<Local path>	A local directory on the ECS used to mount the file system, for example, **/local_path**.
<Shared path>	For an SFS Turbo Standard-Enhanced, Standard, Performance-Enhanced, or Performance file system, the format is *File system IP address***:/**, 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 *File System domain name***:/**, for example, **xxx.sfsturbo.internal:/**.
vers	File system version. Only NFSv3 is supported currently, so the value is fixed to **3**.

Parameter	Description
timeo	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.
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 client. It is invalid for applications on any other clients. **nolock** is recommended. If this parameter is not specified, **lock** is used. Then, other clients cannot write data to the file system.

When mounting the file system, you can add performance optimization options described in **Table 2-3**. Use commas (,) to separate them. A command example is provided as follows:

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

Table 2-3 Performance optimization options

Option	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**.

Option	Description
wsize	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. You are advised to set this parameter to the maximum value **1048576**.
soft/hard	**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. The default value is **hard**.
retrans	Number of retransmission times before the client returns an error. The recommended value is **1**.
ro/rw	● **ro** indicates that the file system is mounted as read-only. ● **rw** indicates that the file system is mounted as read/write. The default value is **rw**. If neither **ro** nor **rw** is specified, the file system will be mounted as read/write.
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.
sync/async	**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. **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.
noatime	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.
nodiratime	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.

NOTE

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

Step 5 View the mounted file system.

```
mount -l
```

If the command output contains the following information, the file system has been mounted. You can access the file system to read or write data from the ECS.

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

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.

Step 6 (Optional) Configure the **fstab** file to configure file system auto mount upon system startup.

After a client ECS is restarted, it loses the file system mount information. You can configure auto mount in the **fstab** file to ensure that the ECS automatically mounts the file system when it restarts.

1. Open the **/etc/fstab** file.

```
vi /etc/fstab
```

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 on the file system details page. Each record in the **/etc/fstab** file represents a mount. Each record has six fields, as described in **Table 2-4**.

Table 2-4 Mount fields

Field	Description
<Shared path>	The address of the file system to be mounted. Set it to the shared path in the **mount** command in **Step 4**.
</local_path>	A directory created on the ECS for mounting the file system. Set it to the local path in the **mount** command in **Step 4**.
nfs	The file system or partition mount type. Set it to **nfs**.

Field	Description
vers=3,timeo=600,noresvport,nolock,tcp	Mount options. Use commas (,) to separate multiple options. ● **vers**: The file system version. Value **3** indicates NFSv3. ● **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 client. It is invalid for applications on any other clients. **nolock** is recommended. If this parameter is not specified, **lock** is used. Then, other clients cannot write data to the file system. ● **tcp**: The TCP transmission protocol.
0	Choose whether to use **dump** to back up the file system. ● **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 priority.
0	Choose whether to use fsck to check file systems when the ECS is starting and specify the sequence for checking file systems. ● **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 one with a larger integer.

NOTICE

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

2. Press **Esc**, enter **:wq**, and press **Enter** to save and exit.
3. (Optional) View the updated content of the **/etc/fstab** file.

```
cat /etc/fstab
```

NOTE

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

FAQs

SFS Turbo Mount

2.3 Viewing an SFS Turbo 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 2-5** describes the file system parameters.

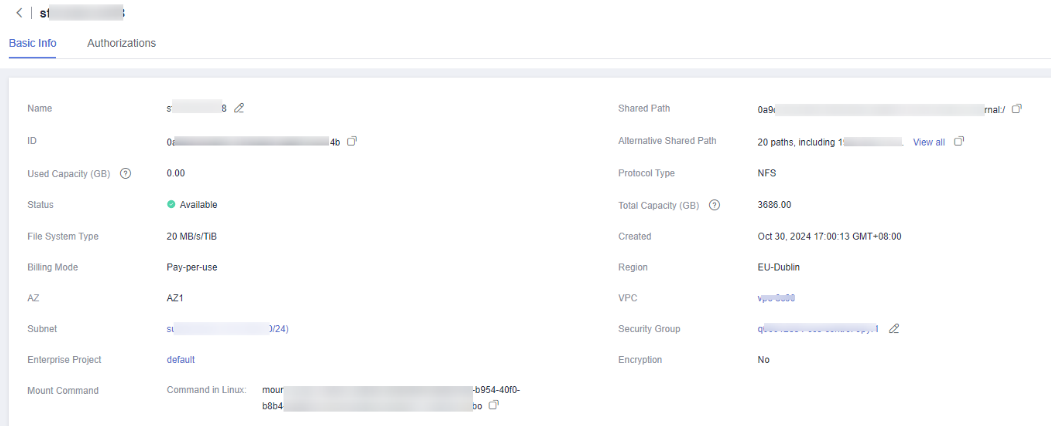
Table 2-5 File system parameters

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

Parameter	Description
Encryption	Encryption status of a file system. The value can be **Yes** or **No**.
Enterprise Project	Enterprise project to which a file system belongs
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 2-2 Details of an SFS Turbo file system



< | sf[redacted]

Basic Info Authorizations

Name	s[redacted]8	Shared Path	0a9[redacted]rnal:/
ID	0a[redacted]4b	Alternative Shared Path	20 paths, including 1[redacted]. View all
Used Capacity (GB)	0.00	Protocol Type	NFS
Status	Available	Total Capacity (GB)	3686.00
File System Type	20 MB/s/TiB	Created	Oct 30, 2024 17:00:13 GMT+08:00
Billing Mode	Pay-per-use	Region	EU-Dublin
AZ	AZ1	VPC	[redacted]
Subnet	su[redacted]0/24)	Security Group	q[redacted]
Enterprise Project	default	Encryption	No
Mount Command	Command in Linux: mour[redacted]-b954-40f0- b8b4[redacted]bo		

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

2.4 Expanding Capacity of an SFS Turbo File System

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 2-6 Capacity expansion parameters

Parameter	Description
Current Capacity	Current storage capacity of the file system
New Capacity	New storage capacity of the file system Constraints: ● Standard–Enhanced (discontinued), Standard, Performance–Enhanced (discontinued), and Performance: 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. ● 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: The expansion increment is 1.2 TB, and the new capacity must be an integer multiple of 1.2 TB and no more than 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.

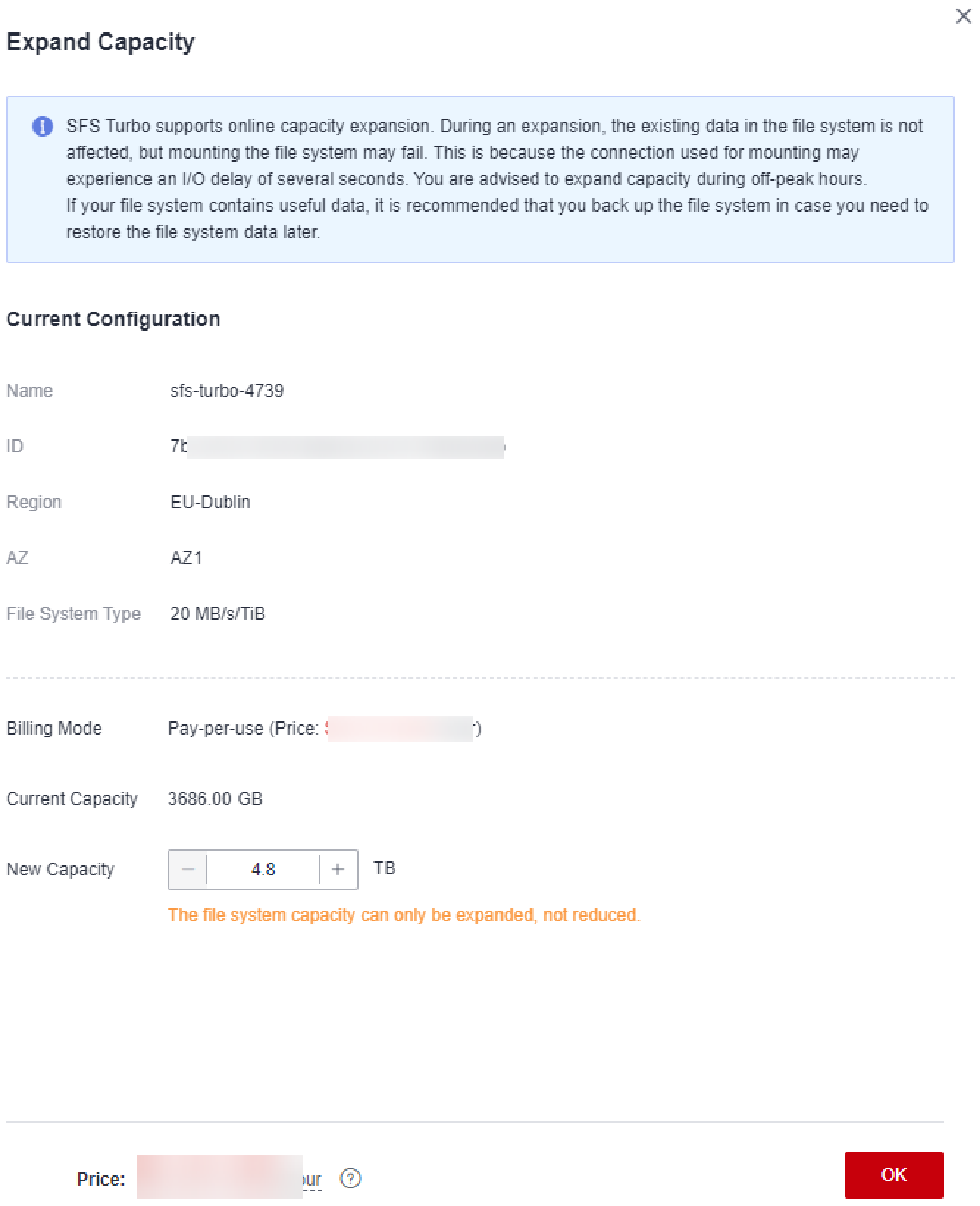
----End

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 2-3 Expanding the capacity of a pay-per-use SFS Turbo file system



Expand Capacity

SFS Turbo supports online capacity expansion. During an expansion, the existing data in the file system is not affected, but mounting the file system may fail. This is because the connection used for mounting may experience an I/O delay of several seconds. You are advised to expand capacity during off-peak hours.
If your file system contains useful data, it is recommended that you back up the file system in case you need to restore the file system data later.

Current Configuration

Name	sfs-turbo-4739
ID	7b[redacted]
Region	EU-Dublin
AZ	AZ1
File System Type	20 MB/s/TiB
Billing Mode	Pay-per-use (Price: [redacted])
Current Capacity	3686.00 GB
New Capacity	4.8 TB

The file system capacity can only be expanded, not reduced.

Price: [redacted]ur

Step 3 Enter a new capacity based on service requirements. For detailed parameter descriptions, see **Table 2-6**.

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

----End

2.5 Unmounting an SFS Turbo 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 **Compute** > **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

2.6 Deleting an SFS Turbo 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 an SFS Turbo File System**.

Procedure

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

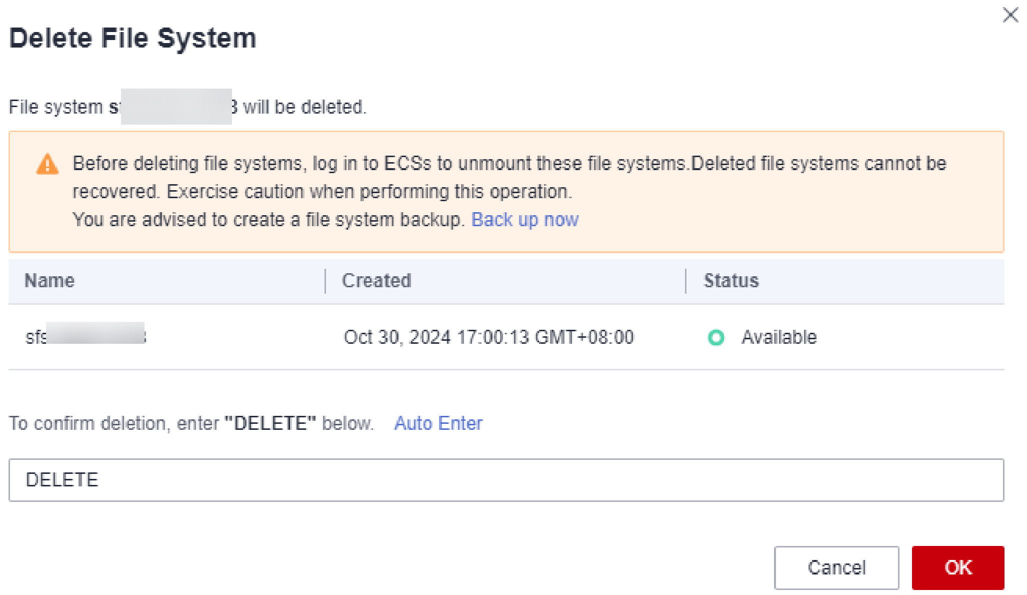
Step 2 For a pay-per-use SFS Turbo file system, a dialog box will be displayed. Confirm the information, enter **DELETE** in the text box, and click **OK**.

For a yearly/monthly SFS Turbo file system, complete the unsubscription as prompted after you click **Unsubscribe**.

NOTE

- Only **Available** and **Unavailable** file systems can be deleted or unsubscribed from.
- You can unsubscribe from yearly/monthly SFS Turbo file systems even when they are in a grace period or retention period.

Figure 2-4 Deleting a file system



Delete File System

File system s[redacted]3 will be deleted.

Before deleting file systems, log in to ECSs to unmount these file systems.Deleted file systems cannot be recovered. Exercise caution when performing this operation.
You are advised to create a file system backup. Back up now

Name	Created	Status
sfs[redacted]	Oct 30, 2024 17:00:13 GMT+08:00	Available

To confirm deletion, enter **"DELETE"** below. Auto Enter

DELETE

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

----End

3 Data Migration

3.1 Data Migration Overview

By default, an SFS Turbo file system can only be accessed by ECSs or CCE containers 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 using Direct Connect, VPN, or VPC peering connections.

- To access from on premises or another cloud, use Direct Connect or VPN.
- To access from a different VPC using the same account in the same region, use VPC peering.
- To access using a different account in the same region, use VPC peering.
- To access from a different region, use Cloud Connect.

This section mainly introduces how to migrate data from the local NAS storage to SFS Turbo through direct file system mount to an ECS that can access the Internet. There are two scenarios:

- Mount and access the SFS Turbo file system.

  Migrating Local Data to SFS Turbo Using Direct Connect
- If the connection cannot be established through file system mounting, migrate data through the ECS over the Internet.

  Migrating Local Data to SFS Turbo over the Internet

3.2 Migrating Local Data to SFS Turbo 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 you can only migrate the changed data.

Prerequisites

- You have purchased 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 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*.

3.3 Migrating Local Data to SFS Turbo over the Internet

Context

You can migrate data from a local NAS to SFS Turbo over 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 cloud server (ECS) is used to access SFS Turbo. Inbound and outbound traffic is allowed on port 22 of the 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 you can only migrate the changed data.

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 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 Install rclone on **client 1**.

```
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 Configure the environment on **client1**.

```
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 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

4 Data Encryption

4.1 Creating and Unmounting an Encrypted File System

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 an SFS Turbo 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 (30 seconds by default). Exercise caution in this case.

For details about how to unmount an encrypted file system, see **Unmounting an SFS Turbo File System**.

5 Backing Up an SFS Turbo File System

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 (such as an accidental deletion, a hacker attack, or virus infection), you can use data backups to quickly restore data.

Creating a Backup on the CBR Console

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 on the CBR Console

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 2-1**.

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 Managing Service 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 SFS Turbo File System to Linux ECSs**.

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 SFS Turbo File System to Linux ECSs**.

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 Performance Tuning

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

File System Performance Data

Table 9-1 lists the performance metrics of SFS Turbo file systems including IOPS and throughput.

Table 9-1 File system performance data

Parameter	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
Max. capacity	1 PB	1 PB	1 PB	1 PB	1 PB	1 PB
Max. IOPS	250,000	250,000	1 million	1 million	4 million You can **submit a service ticket** to get up to 10 million of IOPS.	4 million You can **submit a service ticket** to get up to 10 million of IOPS.
Max. throughput	8 GB/s You can **submit a service ticket** to get up to 20 Gbit/s of throughput.	8 GB/s You can **submit a service ticket** to get up to 20 Gbit/s of throughput.	20 GB/s You can **submit a service ticket** to get up to 100 Gbit/s of throughput.	20 GB/s You can **submit a service ticket** to get up to 100 Gbit/s of throughput.	80 GB/s You can **submit a service ticket** to get up to 200 Gbit/s of throughput.	80 GB/s You can **submit a service ticket** to get up to 200 Gbit/s of throughput.
IOPS calculation formula	IOPS = Min. (250,000, 600 x Capacity) Capacity unit: TB	IOPS = Min. (250,000, 1,200 x Capacity) Capacity unit: TB	IOPS = Min. (1,000,000, 6,000 x Capacity) Capacity unit: TB	IOPS = Min. (1,000,000, 12,500 x Capacity) Capacity unit: TB	IOPS = Min. (4,000,000, 25,000 x Capacity) Capacity unit: TB	IOPS = Min. (4,000,000, 50,000 x Capacity) Capacity unit: TB

Common Test Configuration Examples

NOTE

Test results provided in the following examples are obtained using a single ECS. You are advised to use multiple ECSs to test the **SFS Turbo** performance.

In the following examples, the Performance type 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

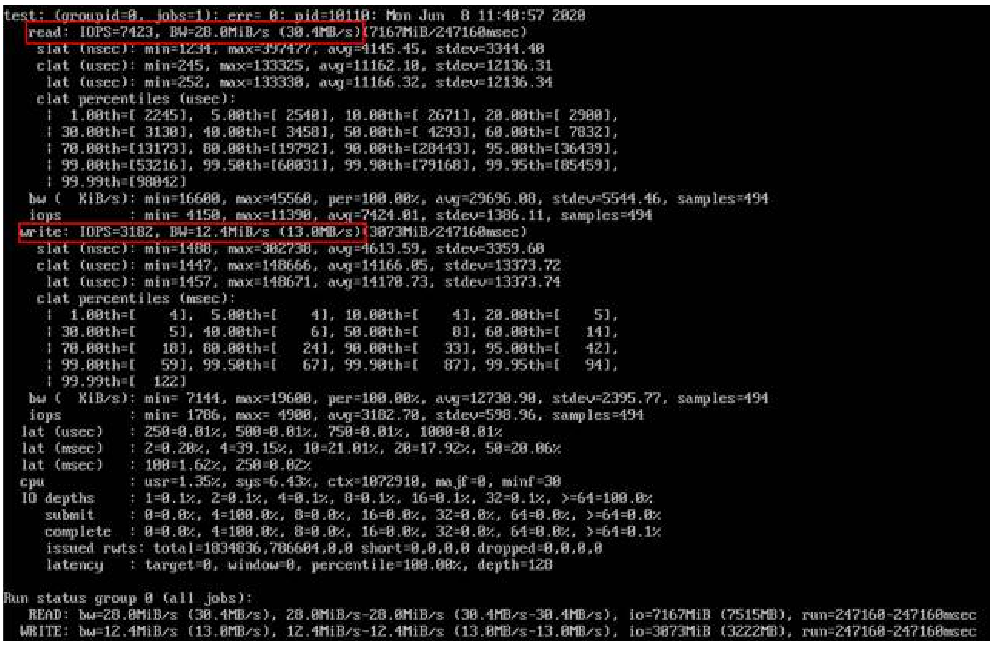
- fio command:

  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

  > Variable */mnt/nfs/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/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 11:40:57 2020
  read: IOPS=7423, BW=28.0MiB/s (30.4MB/s)(7167MiB/247160msec)
    slat (nsec): min=1234, max=397477, avg=4145.45, stdev=3344.40
    clat (usec): min=245, max=133325, avg=11162.10, stdev=12136.31
     lat (usec): min=252, max=133330, avg=11166.32, stdev=12136.34
    clat percentiles (usec):
     |  1.00th=[ 2245],  5.00th=[ 2540], 10.00th=[ 2671], 20.00th=[ 2900],
     | 30.00th=[ 3130], 40.00th=[ 3458], 50.00th=[ 4293], 60.00th=[ 7832],
     | 70.00th=[13173], 80.00th=[19792], 90.00th=[28443], 95.00th=[36439],
     | 99.00th=[53216], 99.50th=[60031], 99.90th=[79168], 99.95th=[85459],
     | 99.99th=[98042]
   bw (  KiB/s): min=16600, max=45560, per=100.00%, avg=29696.08, stdev=5544.46, samples=494
   iops        : min= 4150, max=11390, avg=7424.01, stdev=1386.11, samples=494
  write: IOPS=3182, BW=12.4MiB/s (13.0MB/s)(3073MiB/247160msec)
    slat (nsec): min=1400, max=302730, avg=4613.59, stdev=3359.60
    clat (usec): min=1447, max=148666, avg=14166.05, stdev=13373.72
     lat (usec): min=1457, max=148671, avg=14170.73, stdev=13373.74
    clat percentiles (msec):
     |  1.00th=[    4],  5.00th=[    4], 10.00th=[    4], 20.00th=[    5],
     | 30.00th=[    5], 40.00th=[    6], 50.00th=[    8], 60.00th=[   14],
     | 70.00th=[   18], 80.00th=[   24], 90.00th=[   33], 95.00th=[   42],
     | 99.00th=[   59], 99.50th=[   67], 99.90th=[   87], 99.95th=[   94],
     | 99.99th=[  122]
   bw (  KiB/s): min= 7144, max=19600, per=100.00%, avg=12730.90, stdev=2395.77, samples=494
   iops        : min= 1786, max= 4900, avg=3182.70, stdev=598.96, samples=494
  lat (usec)   : 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=1834836,786604,0,0 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=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
```

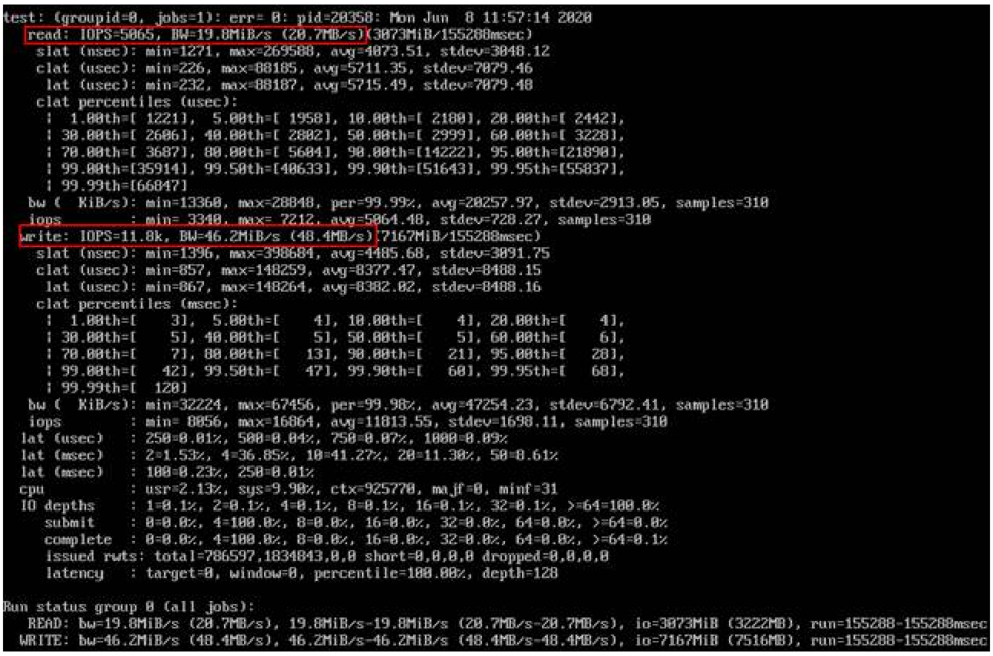
- fio command:

  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

  > Variable */mnt/nfs/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/nfs** directory in this example. Set it based on the site requirements.

- fio result:



```
test: (groupid=0, jobs=1): err= 0: pid=20358: Mon Jun  8 11:57:14 2020
  read: IOPS=5065, BW=19.8MiB/s (20.7MB/s)(3073MiB/155288msec)
   slat (nsec): min=1271, max=269588, avg=4073.51, stdev=3048.12
   clat (usec): min=226, max=88185, avg=5711.35, stdev=7079.46
    lat (usec): min=232, max=88187, avg=5715.49, stdev=7079.48
   clat percentiles (usec):
    |  1.00th=[ 1221],  5.00th=[ 1958], 10.00th=[ 2180], 20.00th=[ 2442],
    | 30.00th=[ 2606], 40.00th=[ 2802], 50.00th=[ 2999], 60.00th=[ 3228],
    | 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=[66847]
  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=728.27, samples=310
 write: IOPS=11.8k, BW=46.2MiB/s (48.4MB/s)(7167MiB/155288msec)
   slat (nsec): min=1396, max=398684, avg=4485.68, stdev=3091.75
   clat (usec): min=857, max=148259, avg=8377.47, stdev=8488.15
    lat (usec): min=867, max=148264, avg=8382.02, stdev=8488.16
   clat percentiles (msec):
    |  1.00th=[    3],  5.00th=[    4], 10.00th=[    4], 20.00th=[    4],
    | 30.00th=[    5], 40.00th=[    5], 50.00th=[    5], 60.00th=[    6],
    | 70.00th=[    7], 80.00th=[   13], 90.00th=[   21], 95.00th=[   28],
    | 99.00th=[   42], 99.50th=[   47], 99.90th=[   60], 99.95th=[   68],
    | 99.99th=[  120]
  bw (  KiB/s): min=32224, max=67456, per=99.98%, avg=47254.23, stdev=6792.41, samples=310
  iops        : min= 8056, max=16864, avg=11813.55, stdev=1698.11, 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=11.38%, 50=8.61%
 lat (msec)   : 100=0.23%, 250=0.01%
 cpu          : usr=2.13%, sys=9.98%, 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=786597,1834843,0,0 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=19.8MiB/s (20.7MB/s), 19.8MiB/s-19.8MiB/s (20.7MB/s-20.7MB/s), io=3073MiB (3222MB), run=155288-155288msec
  WRITE: bw=46.2MiB/s (48.4MB/s), 46.2MiB/s-46.2MiB/s (48.4MB/s-48.4MB/s), io=7167MiB (7516MB), run=155288-155288msec
```

Sequential read IOPS

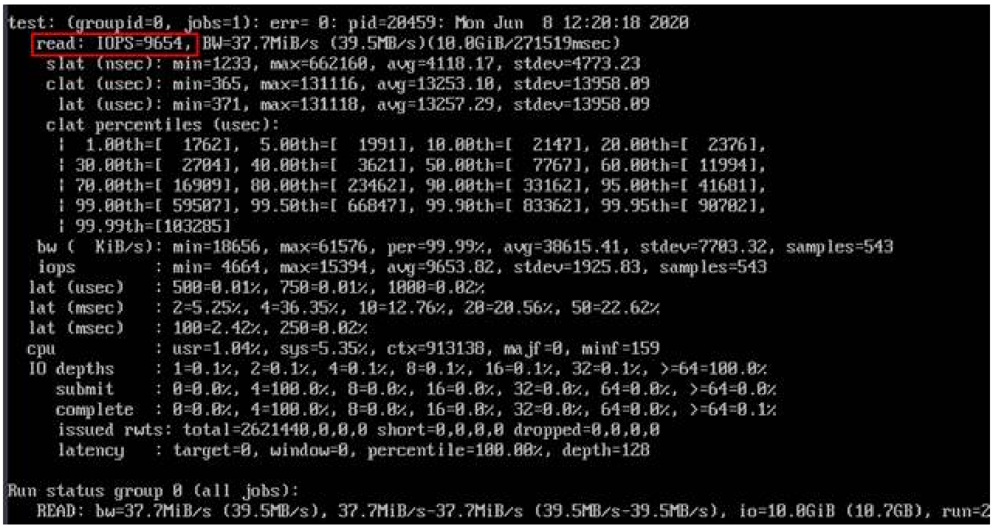
- fio command:

  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=13958.09
    lat (usec): min=371, max=131118, avg=13257.29, stdev=13958.09
   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=18656, max=61576, per=99.99%, avg=38615.41, stdev=7703.32, samples=543
  iops        : min= 4664, max=15394, avg=9653.82, stdev=1925.83, 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=913138, 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=128

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:

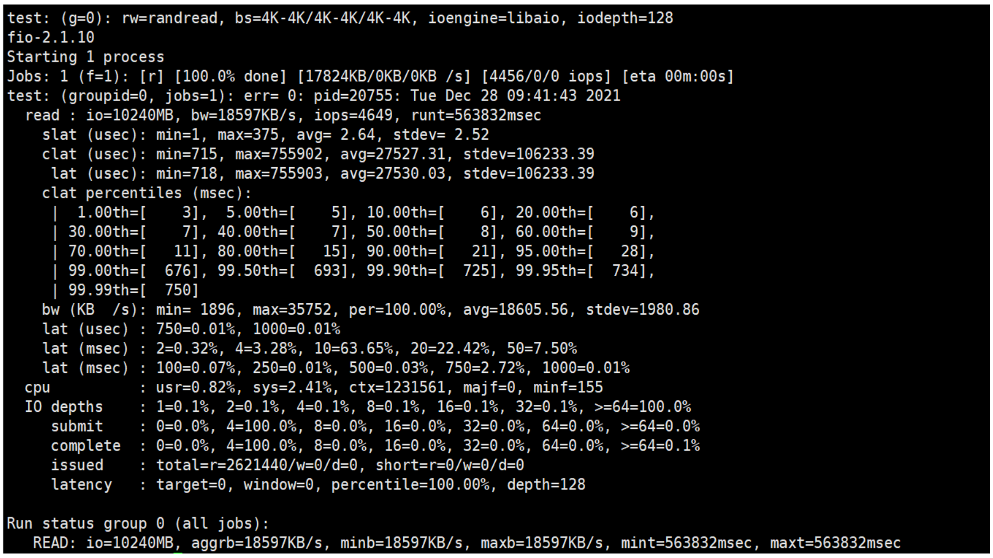
  fio --ioengine=libaio --direct=1 --fallocate=none --time_based=1 --group_reporting=1 --name=iops_fio --directory=*/mnt/sfs-turbo/* --

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, aggrb=18597KB/s, minb=18597KB/s, maxb=18597KB/s, mint=563832msec, maxt=563832msec
```

Sequential write IOPS

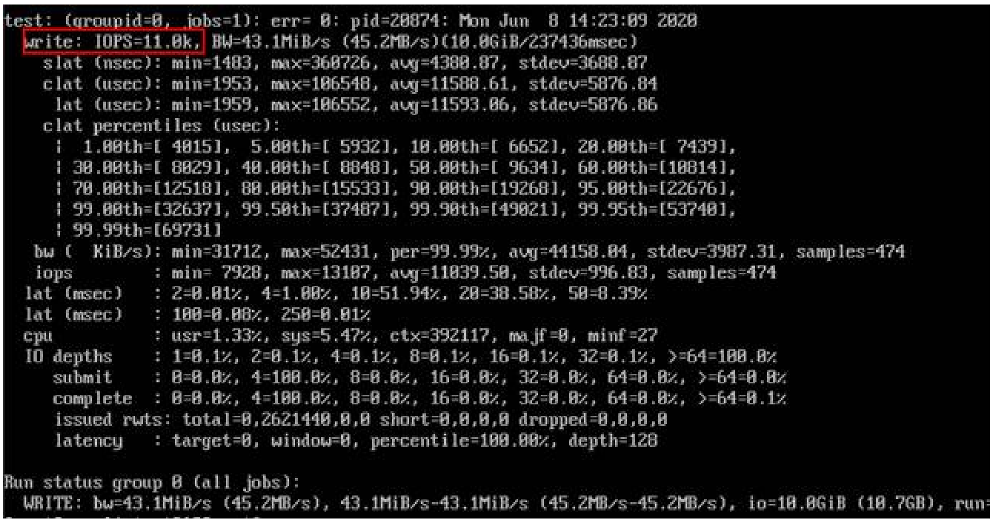
- fio command:

  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)(10.0GiB/237436msec)
    slat (nsec): min=1483, max=360726, avg=4380.87, stdev=3608.87
    clat (usec): min=1953, max=106548, avg=11588.61, stdev=5876.84
     lat (usec): min=1959, max=106552, avg=11593.06, 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=[49021], 99.95th=[53740],
     | 99.99th=[69731]
   bw (  KiB/s): min=31712, max=52431, per=99.99%, avg=44158.04, stdev=3987.31, samples=474
   iops        : min= 7928, max=13107, avg=11039.50, stdev=996.83, samples=474
  lat (msec)   : 2=0.01%, 4=1.00%, 10=51.94%, 20=38.58%, 50=8.39%
  lat (msec)   : 100=0.08%, 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=0,2621440,0,0 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=43.1MiB/s (45.2MB/s), 43.1MiB/s-43.1MiB/s (45.2MB/s-45.2MB/s), io=10.0GiB (10.7GB), run=
```

Random write IOPS

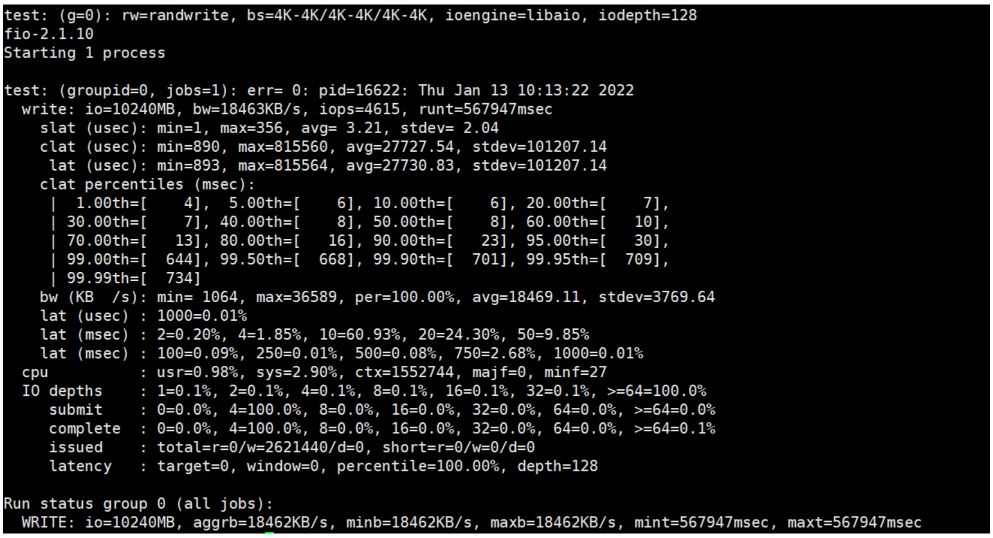
- fio command:

  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

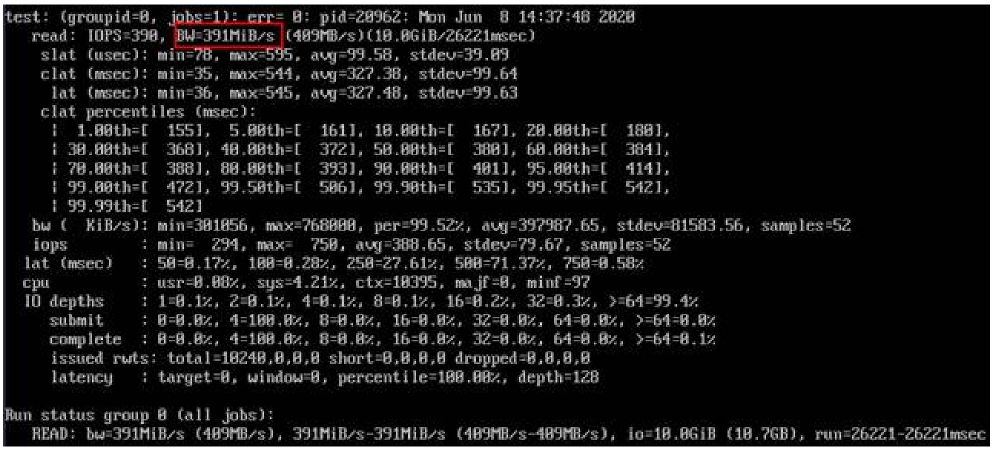
- fio command:

  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

  Variable */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=20962: Mon Jun  8 14:37:48 2020
   read: IOPS=390, BW=391MiB/s (409MB/s)(10.0GiB/26221msec)
    slat (usec): min=78, max=595, avg=99.58, stdev=39.09
    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=[  180],
     | 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=[  506], 99.90th=[  535], 99.95th=[  542],
     | 99.99th=[  542]
   bw (  KiB/s): min=301056, max=768000, per=99.52%, avg=397987.65, stdev=81583.56, samples=52
   iops        : min=  294, max=  750, 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=0.08%, sys=4.21%, ctx=10395, 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 (409MB/s), 391MiB/s-391MiB/s (409MB/s-409MB/s), io=10.0GiB (10.7GB), run=26221-26221msec
```

Random read bandwidth

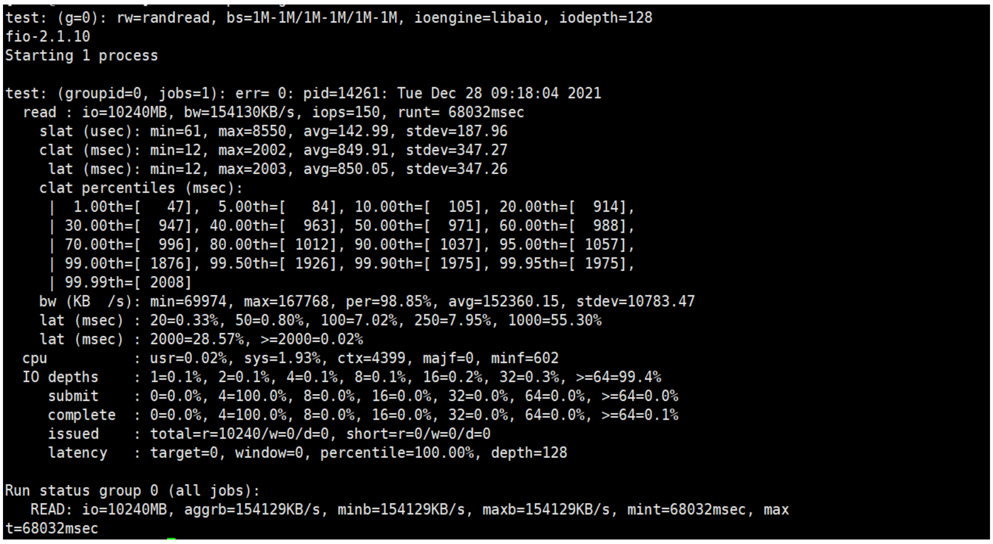
- fio command:

  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

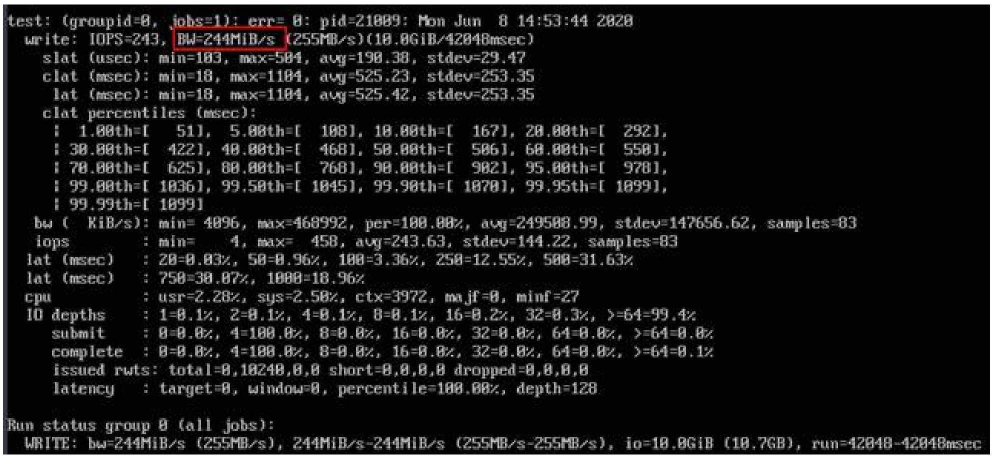
- fio command:

  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=21009: 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=[  550],
     | 70.00th=[  625], 80.00th=[  768], 90.00th=[  902], 95.00th=[  978],
     | 99.00th=[ 1036], 99.50th=[ 1045], 99.90th=[ 1070], 99.95th=[ 1099],
     | 99.99th=[ 1099]
   bw (  KiB/s): min= 4096, max=468992, per=100.00%, avg=249508.99, 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=2.58%, 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

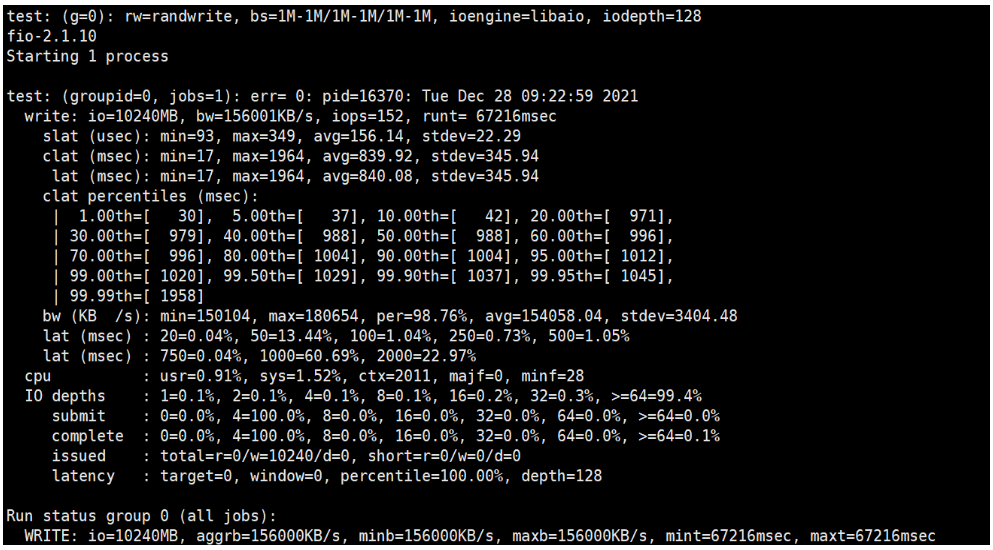
- fio command:

  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=156001KB/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=[  971],
     | 30.00th=[  979], 40.00th=[  988], 50.00th=[  988], 60.00th=[  996],
     | 70.00th=[  996], 80.00th=[ 1004], 90.00th=[ 1004], 95.00th=[ 1012],
     | 99.00th=[ 1020], 99.50th=[ 1029], 99.90th=[ 1037], 99.95th=[ 1045],
     | 99.99th=[ 1958]
    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 Testing SFS Turbo Latency

This section uses the Performance-Enhanced type to test the file system latency. The specifications of the ECSs to be used are as follows:

Specifications: General computing-plus | c6.4xlarge.4 | 16 vCPUs | 64 GB

Image: EulerOS 2.5

Single-queue random read

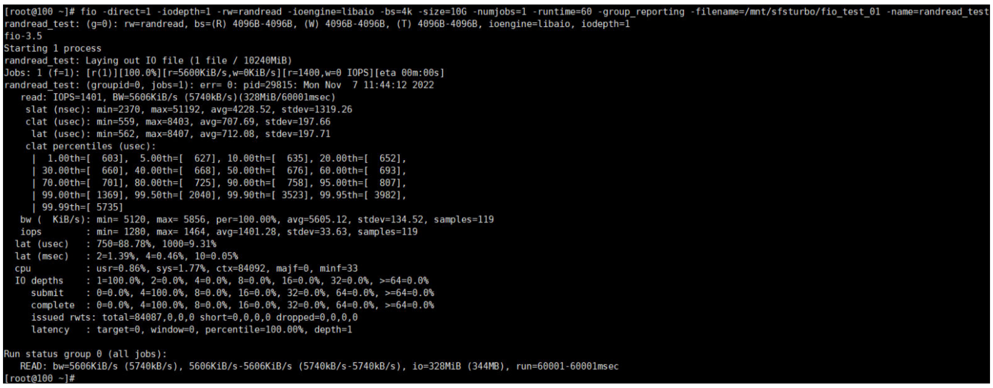
- fio command:

  fio -direct=1 -iodepth=1 -rw=randread -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=*/mnt/sfsturbo/fio_test_01 -name*=randread_test

  NOTE

  Variable */mnt/sfsturbo/fio_test_01 -name* is the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_01 -name** file in the **/mnt/sfsturbo** directory in this example. Set it based on the site requirements.

- fio result:



```
[root@100 ~]# fio -direct=1 -iodepth=1 -rw=randread -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_01 -name=randread_test
randread_test: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
randread_test: Laying out IO file (1 file / 10240MiB)
Jobs: 1 (f=1): [r(1)][100.0%][r=5600KiB/s,w=0KiB/s][r=1400,w=0 IOPS][eta 00m:00s]
randread_test: (groupid=0, jobs=1): err= 0: pid=29815: Mon Nov  7 11:44:12 2022
   read: IOPS=1401, BW=5606KiB/s (5740kB/s)(328MiB/60001msec)
    slat (nsec): min=2370, max=51192, avg=4228.52, stdev=1319.26
    clat (usec): min=559, max=8403, avg=707.69, stdev=197.66
     lat (usec): min=562, max=8407, avg=712.08, stdev=197.71
    clat percentiles (usec):
     |  1.00th=[  603],  5.00th=[  627], 10.00th=[  635], 20.00th=[  652],
     | 30.00th=[  660], 40.00th=[  668], 50.00th=[  676], 60.00th=[  693],
     | 70.00th=[  701], 80.00th=[  725], 90.00th=[  758], 95.00th=[  807],
     | 99.00th=[ 1369], 99.50th=[ 2040], 99.90th=[ 3523], 99.95th=[ 3982],
     | 99.99th=[ 5735]
   bw (  KiB/s): min= 5120, max= 5856, per=100.00%, avg=5605.12, stdev=134.52, samples=119
   iops        : min= 1280, max= 1464, avg=1401.28, stdev=33.63, samples=119
  lat (usec)   : 750=88.78%, 1000=9.31%
  lat (msec)   : 2=1.39%, 4=0.46%, 10=0.05%
  cpu          : usr=0.86%, sys=1.77%, ctx=84092, majf=0, minf=33
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
     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=0.0%
     issued rwts: total=84087,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=5606KiB/s (5740kB/s), 5606KiB/s-5606KiB/s (5740kB/s-5740kB/s), io=328MiB (344MB), run=60001-60001msec
[root@100 ~]#
```

Single-queue random write

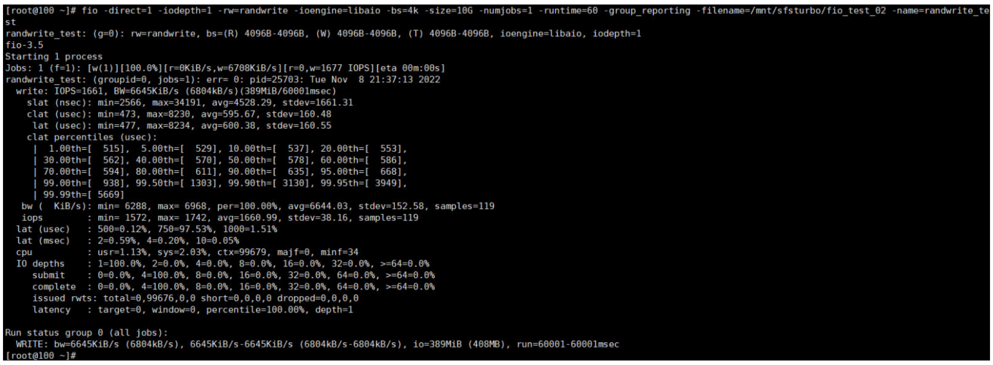
- fio command:

  fio -direct=1 -iodepth=1 -rw=randwrite -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=*/mnt/sfsturbo/fio_test_02 -name*=randwrite_test

  NOTE

  Variable */mnt/sfsturbo/fio_test_02 -name* is the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_02 -name** file in the **/mnt/sfsturbo** directory in this example. Set it based on the site requirements.

- fio result:



```
[root@100 ~]# fio -direct=1 -iodepth=1 -rw=randwrite -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_02 -name=randwrite_te
st
randwrite_test: (g=0): rw=randwrite, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
Jobs: 1 (f=1): [w(1)][100.0%][r=0KiB/s,w=6708KiB/s][r=0,w=1677 IOPS][eta 00m:00s]
randwrite_test: (groupid=0, jobs=1): err= 0: pid=25703: Tue Nov  8 21:37:13 2022
  write: IOPS=1661, BW=6645KiB/s (6804kB/s)(389MiB/60001msec)
    slat (nsec): min=2566, max=34191, avg=4528.29, stdev=1661.31
    clat (usec): min=473, max=8230, avg=595.67, stdev=160.48
     lat (usec): min=477, max=8234, avg=600.38, stdev=160.55
    clat percentiles (usec):
     |  1.00th=[  515],  5.00th=[  529], 10.00th=[  537], 20.00th=[  553],
     | 30.00th=[  562], 40.00th=[  570], 50.00th=[  578], 60.00th=[  586],
     | 70.00th=[  594], 80.00th=[  611], 90.00th=[  635], 95.00th=[  668],
     | 99.00th=[  938], 99.50th=[ 1303], 99.90th=[ 3130], 99.95th=[ 3949],
     | 99.99th=[ 5669]
   bw (  KiB/s): min= 6288, max= 6968, per=100.00%, avg=6644.03, stdev=152.58, samples=119
   iops        : min= 1572, max= 1742, avg=1660.99, stdev=38.16, samples=119
  lat (usec)   : 500=0.12%, 750=97.53%, 1000=1.51%
  lat (msec)   : 2=0.59%, 4=0.20%, 10=0.05%
  cpu          : usr=1.13%, sys=2.03%, ctx=99679, majf=0, minf=34
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
     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=0.0%
     issued rwts: total=0,99676,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  WRITE: bw=6645KiB/s (6804kB/s), 6645KiB/s-6645KiB/s (6804kB/s-6804kB/s), io=389MiB (408MB), run=60001-60001msec
[root@100 ~]#
```

Single-queue sequential read

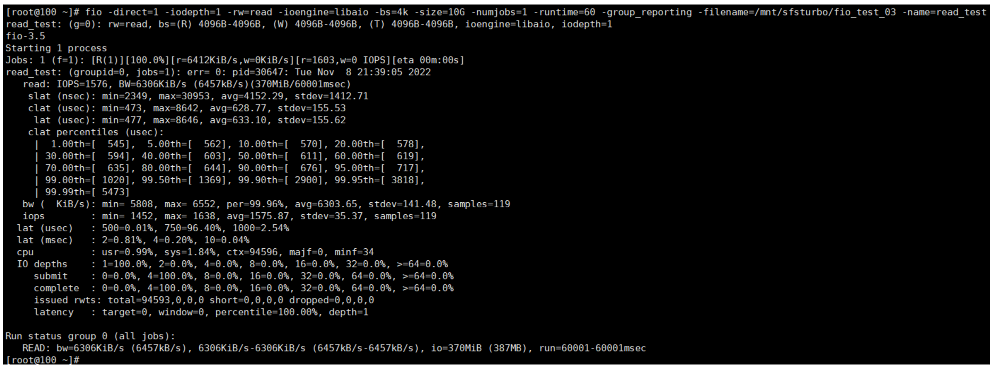
- fio command:

  fio -direct=1 -iodepth=1 -rw=read -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=*/mnt/sfsturbo/fio_test_03 -name*=read_test

NOTE

Variable */mnt/sfsturbo/fio_test_03 -name* is the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_03 -name** file in the **/mnt/sfsturbo** directory in this example. Set it based on the site requirements.

- fio result:



```
[root@100 ~]# fio -direct=1 -iodepth=1 -rw=read -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_03 -name=read_test
read_test: (g=0): rw=read, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
Jobs: 1 (f=1): [R(1)][100.0%][r=6412KiB/s,w=0KiB/s][r=1603,w=0 IOPS][eta 00m:00s]
read_test: (groupid=0, jobs=1): err= 0: pid=30647: Tue Nov  8 21:39:05 2022
   read: IOPS=1576, BW=6306KiB/s (6457kB/s)(370MiB/60001msec)
    slat (nsec): min=2349, max=30953, avg=4152.29, stdev=1412.71
    clat (usec): min=473, max=8642, avg=628.77, stdev=155.53
     lat (usec): min=477, max=8646, avg=633.10, stdev=155.62
    clat percentiles (usec):
     |  1.00th=[  545],  5.00th=[  562], 10.00th=[  570], 20.00th=[  578],
     | 30.00th=[  594], 40.00th=[  603], 50.00th=[  611], 60.00th=[  619],
     | 70.00th=[  635], 80.00th=[  644], 90.00th=[  676], 95.00th=[  717],
     | 99.00th=[ 1020], 99.50th=[ 1369], 99.90th=[ 2900], 99.95th=[ 3818],
     | 99.99th=[ 5473]
   bw (  KiB/s): min= 5808, max= 6552, per=99.96%, avg=6303.65, stdev=141.48, samples=119
   iops        : min= 1452, max= 1638, avg=1575.87, stdev=35.37, samples=119
  lat (usec)   : 500=0.01%, 750=96.40%, 1000=2.54%
  lat (msec)   : 2=0.81%, 4=0.20%, 10=0.04%
  cpu          : usr=0.99%, sys=1.84%, ctx=94596, majf=0, minf=34
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
     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=0.0%
     issued rwts: total=94593,0,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
   READ: bw=6306KiB/s (6457kB/s), 6306KiB/s-6306KiB/s (6457kB/s-6457kB/s), io=370MiB (387MB), run=60001-60001msec
[root@100 ~]#
```

Single-queue sequential write

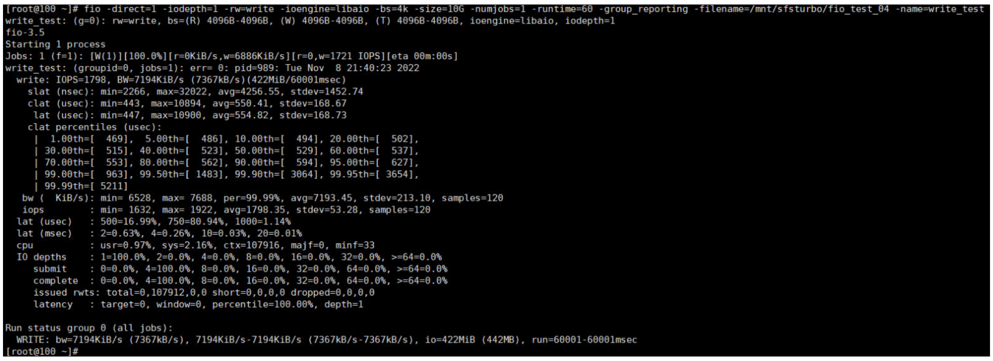
- fio command:

  fio -direct=1 -iodepth=1 -rw=write -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=*/mnt/sfsturbo/fio_test_04 -name*=write_test

NOTE

Variable */mnt/sfsturbo/fio_test_04 -name* is the location of the file to be tested. The location must be specific to the file name, which is the **fio_test_04 -name** file in the **/mnt/sfsturbo** directory in this example. Set it based on the site requirements.

- fio result:



```
[root@100 ~]# fio -direct=1 -iodepth=1 -rw=write -ioengine=libaio -bs=4k -size=10G -numjobs=1 -runtime=60 -group_reporting -filename=/mnt/sfsturbo/fio_test_04 -name=write_test
write_test: (g=0): rw=write, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=1
fio-3.5
Starting 1 process
Jobs: 1 (f=1): [W(1)][100.0%][r=0KiB/s,w=6886KiB/s][r=0,w=1721 IOPS][eta 00m:00s]
write_test: (groupid=0, jobs=1): err= 0: pid=989: Tue Nov  8 21:40:23 2022
  write: IOPS=1798, BW=7194KiB/s (7367kB/s)(422MiB/60001msec)
    slat (nsec): min=2266, max=32022, avg=4256.55, stdev=1452.74
    clat (usec): min=443, max=10894, avg=550.41, stdev=168.67
     lat (usec): min=447, max=10900, avg=554.82, stdev=168.73
    clat percentiles (usec):
     |  1.00th=[  469],  5.00th=[  486], 10.00th=[  494], 20.00th=[  502],
     | 30.00th=[  515], 40.00th=[  523], 50.00th=[  529], 60.00th=[  537],
     | 70.00th=[  553], 80.00th=[  562], 90.00th=[  594], 95.00th=[  627],
     | 99.00th=[  963], 99.50th=[ 1483], 99.90th=[ 3064], 99.95th=[ 3654],
     | 99.99th=[ 5211]
   bw (  KiB/s): min= 6528, max= 7688, per=99.99%, avg=7193.45, stdev=213.10, samples=120
   iops        : min= 1632, max= 1922, avg=1798.35, stdev=53.28, samples=120
  lat (usec)   : 500=16.99%, 750=80.94%, 1000=1.14%
  lat (msec)   : 2=0.63%, 4=0.26%, 10=0.03%, 20=0.01%
  cpu          : usr=0.97%, sys=2.16%, ctx=107916, majf=0, minf=33
  IO depths    : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
     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=0.0%
     issued rwts: total=0,107912,0,0 short=0,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  WRITE: bw=7194KiB/s (7367kB/s), 7194KiB/s-7194KiB/s (7367kB/s-7367kB/s), io=422MiB (442MB), run=60001-60001msec
[root@100 ~]#
```