

弹性文件服务

# 最佳实践

文档版本 01  
发布日期 2024-11-29



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

---

# 目录

---

<b>1 弹性文件服务数据迁移</b> .....	<b>1</b>
1.1 方案概述.....	1
1.2 通过云专线迁移（rclone 工具）.....	1
1.3 跨服务器迁移（rclone 工具）.....	3
1.4 通过云专线迁移（rsync 工具）.....	7
1.5 文件系统之间迁移数据.....	9
1.6 SFS 容量型文件系统迁移至其他文件系统.....	11
<b>2 SFS Turbo 性能测试</b> .....	<b>13</b>
<b>3 SFS Turbo 时延测试</b> .....	<b>21</b>
<b>4 为文件系统创建普通用户可读写的子目录</b> .....	<b>24</b>
4.1 方案概述.....	24
4.2 资源和成本规划.....	25
4.3 实施步骤.....	26
4.3.1 创建文件系统本地目录.....	26
4.3.2 为每个用户创建有读写权限的子目录.....	28

# 1 弹性文件服务数据迁移

## 1.1 方案概述

SFS Turbo默认只能被云上同一个VPC内的ECS/CCE访问，用户可通过云专线/VPN/对等连接等方式打通网络，实现多种访问方式。

- 云下或其他云访问：云专线/VPN
- 云上同区域同一账号不同VPC：VPC对等连接
- 云上同区域跨账号访问：VPC对等连接
- 不同区域访问：云连接

迁移数据分为两种网络条件，通过可访问公网的ECS直接挂载迁移。

- 通过mount方式挂载访问SFS Turbo，将本地NAS存储中的数据迁移至SFS Turbo。
  - [通过云专线迁移（rclone工具）](#)
  - [通过云专线迁移（rsync工具）](#)
- 无法打通网络直接挂载，可以通过公网访问华为云ECS绑定的EIP，也可进行传输迁移。
  - [跨服务器迁移（rclone工具）](#)

## 1.2 通过云专线迁移（rclone 工具）

### 方案概述

用户可以将本地NAS存储中的数据，通过云专线和rclone工具迁移至云上SFS Turbo文件系统中，进行云上业务拓展。

此方案通过在云上创建一台Linux操作系统的云服务器，来连接本地NAS存储以及云上SFS Turbo文件系统的通信，并通过这台服务器将本地NAS存储中的数据迁移至云上。

将云上NAS存储数据迁移至SFS Turbo文件系统中也可以参考本方案，具体说明请参考[云上NAS数据迁移至弹性文件服务](#)。

## 约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- 文件UID和GID在同步操作后将不再保持一致。
- 文件访问模式同步操作后将不再保持一致。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 通过 rclone 工具进行同步时，在执行命令之前写入的数据，可以完全同步；在执行命令之后写入的数据，不能保证完全同步。

## 前提条件

- 已经完成云专线的购买和配置，详情请参考《云专线服务用户指南》。
- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。

## 资源规划

本章节介绍案例中使用的资源信息。具体如表1-1所示。

表 1-1 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs   16GB   c7.2xlarge.2 操作系统：Linux 区域：中国-香港 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

## 操作步骤

**步骤1** 登录弹性云服务器管理控制台。

**步骤2** 登录已创建好的Linux系统云服务器，用于同时访问本地NAS存储和云上SFS Turbo文件系统。

**步骤3** 输入以下挂载命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

**步骤4** 输入以下挂载命令，用于访问云上文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统挂载地址 /mnt/dst
```

**步骤5** 在Linux云服务器中执行以下命令安装rclone工具。

```
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-*
```

**步骤6** 执行以下命令，进行数据同步。

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

### 📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。  
--copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

## 验证

**步骤1** 登录已创建好的Linux系统云服务器。

**步骤2** 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst  
ls | wc -l
```

**步骤3** 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

## 云上 NAS 数据迁移至弹性文件服务

云上NAS和SFS Turbo文件系统只需配置在同一个VPC下，或使用云连接（CC）配置网络后，再参考以上操作步骤，即可进行云上NAS数据迁移至弹性文件服务中。

云连接配置操作请参考[云连接用户指南](#)。

## 1.3 跨服务器迁移（rclone 工具）

### 方案概述

用户可以将本地NAS存储中的数据，使用rclone工具通过公网或内网迁移至云上SFS Turbo中，进行云上业务拓展。

此方案通过在云上和本地分别创建一台Linux操作系统的服务器，将本地NAS存储中的数据迁移至云上。两台服务器端口号22的出入方向需允许访问。本地服务器用于访问本地NAS存储，云上服务器用于访问云上SFS Turbo。

通过公网或内网将云上NAS存储数据迁移至SFS Turbo中也可以参考本方案完成。

### 约束与限制

- 暂不支持使用公网迁移本地NAS存储至SFS容量型文件系统。
- 仅支持使用Linux系统的云服务器进行数据迁移。
- 文件UID和GID在同步操作后将不再保持一致。

- 文件访问模式同步操作后不再保持一致。
- 端口号22的出入方向需允许访问。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 通过 rclone 工具进行同步时，在执行命令之前写入的数据，可以完全同步；在执行命令之后写入的数据，不能保证完全同步。

## 前提条件

- 已在云上和本地分别创建一台Linux操作系统的服务器。
- 服务器已完成弹性IP的配置，保证两台服务器间可以相互访问。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。

## 资源规划

本章节介绍案例中使用的资源信息。具体如表1-2所示。

表 1-2 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs   16GB   c7.2xlarge.2 操作系统：Linux 区域：中国-香港 VPC名称：VPC1 端口开放：22 弹性IP：xxx.xxx.xxx.xxx	已创建/mnt/dst的目录

## 操作步骤

**步骤1** 登录弹性云服务器管理控制台。

**步骤2** 登录已创建的本地服务器client1，执行以下命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

**步骤3** 登录已创建的Linux云服务器client2，执行以下命令，用于访问云上SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock SFS Turbo挂载地址 /mnt/dst
```

**步骤4** 在本地服务器client1中执行以下命令安装rclone工具。

```
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-*
```

**步骤5** 在本地服务器client1中执行以下命令，进行环境配置。

```
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 (新建名字)
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 (选择ssh/sftp号码)
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 (client2机器ip地址)
SSH username, leave blank for current username, root
Enter a string value. Press Enter for the default ("").
user> user name(client2机器的用户名)
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:(client2机器的登录密码)
Confirm the password:
password:(确认client2机器的登录密码)
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> (默认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> (默认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> (默认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
```

#### 说明

**client2机器ip地址**为公网地址。

**步骤6** 执行以下命令查看rclone.conf文件，路径为/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 = ***
```

**步骤7** 在本地服务器client1执行以下命令，同步数据。

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

#### 说明

- 命令中的`remote_name`需替换为实际的远端名称。
- 参数说明如下，`transfers`和`checkers`数目可以根据系统规格自行配置：
  - `transfers`：传输文件的并发数目。
  - `checkers`：扫描本地文件的并发数目。
  - `P`：数据拷贝进度。

等待数据完成同步后，可前往目标SFS Turbo文件系统查看是否已成功迁移。

----结束

## 验证

**步骤1** 登录已创建好的Linux系统云服务器。

**步骤2** 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

步骤3 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

## 1.4 通过云专线迁移（rsync 工具）

### 方案概述

用户可以将本地NAS存储中的数据，通过云专线和rsync工具迁移至云上SFS Turbo文件系统中，进行云上业务拓展。

此方案通过在云上创建一台Linux操作系统的云服务器，来连接本地NAS存储以及云上SFS Turbo文件系统的通信，并通过这台服务器将本地NAS存储中的数据迁移至云上。

将云上NAS存储数据迁移至SFS Turbo文件系统中也可以参考本方案，云上NAS和SFS Turbo文件系统需要配置在同一个VPC下。

### 约束与限制

- 支持拷贝特殊文件，如连接文件、设备等。
- 支持单文件断点续传。
- 支持保持原文件或目录的权限、时间、软硬链接、属主、组等所有属性。
- 支持使用rcp、rsh、ssh等方式来配合传输文件。
- 支持实现增量迁移，即只迁移发生变化的数据。
- 如果存在迁移多层级目录且小文件数量较多的情况时，rsync迁移效率将会非常低，建议使用多进程脚本解决或rclone工具迁移。

### 前提条件

- 已经完成云专线的购买和配置，详情请参考《云专线服务用户指南》。
- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS Turbo文件系统，并获取到文件系统的挂载地址。
- 已经获取到本地NAS存储的挂载地址。
- 本地NAS和云服务器已完成互信关系建立。

### 资源规划

本章节介绍案例中使用的资源信息。具体如表1-3所示。

表 1-3 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs   16GB   c7.2xlarge.2 操作系统：Linux 区域：中国-香港 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

## 操作步骤

**步骤1** 登录已创建好的Linux系统云服务器，用于同时访问本地NAS存储和SFS Turbo文件系统。

**步骤2** 输入以下挂载命令，用于访问本地NAS存储。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 本地NAS挂载地址 /mnt/src
```

**步骤3** 输入以下挂载命令，用于访问SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统挂载地址 /mnt/dst
```

**步骤4** 在Linux云服务器中执行以下命令安装rsync工具。

```
yum install rsync
```

图 1-1 安装 rsync 工具

```
[root@ ~]# yum install rsync
Loaded plugins: fastestmirror
Determining fastest mirrors
epel/x86_64/metalink
 * base: mirrors.huaweicloud.com
 * epel: mirrors.bfsu.edu.cn
 * extras: mirrors.ustc.edu.cn
 * updates: mirrors.huaweicloud.com
```

### 说明

源服务器和目的端服务器均需要安装rsync工具，否则将会报错。

**步骤5** 安装完成后，可以输入以下命令查询rsync工具的安装和版本情况。

```
rsync -version
```

图 1-2 查看工具安装情况

```
[root@ ~]# rsync -version
rsync version 3.1.2 protocol version 31
Copyright (C) 1996-2015 by Andrew Tridgell, Wayne Davison, and others.
Web site: http://rsync.samba.org/
Capabilities:
 64-bit files, 64-bit inums, 64-bit timestamps, 64-bit long ints,
 socketpairs, hardlinks, symlinks, IPv6, batchfiles, inplace,
 append, ACLs, xattrs, iconv, symtimes, prealloc

rsync comes with ABSOLUTELY NO WARRANTY. This is free software, and you
are welcome to redistribute it under certain conditions. See the GNU
General Public Licence for details.

rsync is a file transfer program capable of efficient remote update
via a fast differencing algorithm.

Usage: rsync [OPTION]... SRC [SRC]... DEST
       or rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
       or rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
       or rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
       or rsync [OPTION]... [USER@]HOST:SRC [DEST]
       or rsync [OPTION]... [USER@]HOST::SRC [DEST]
       or rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]
The ':' usages connect via remote shell, while '::' & 'rsync://' usages connect
to an rsync daemon, and require SRC or DEST to start with a module name.
```

**步骤6** 执行以下命令，将源端服务器/mnt/src目录中的数据全量同步迁移至目的端服务器/mnt/dst目录（文件系统）中。

```
rsync -avP /mnt/src /mnt/dst
```

图 1-3 全量同步数据

```
103669.dat
      1,024 100% 38.46kB/s  0:00:00 (xfr#4080, to-chk=295920/300001)
10367.dat
      1,024 100% 37.04kB/s  0:00:00 (xfr#4081, to-chk=295919/300001)
103670.dat
      1,024 100% 37.04kB/s  0:00:00 (xfr#4082, to-chk=295918/300001)
103671.dat
      1,024 100% 35.71kB/s  0:00:00 (xfr#4083, to-chk=295917/300001)
103672.dat
      1,024 100% 35.71kB/s  0:00:00 (xfr#4084, to-chk=295916/300001)
103673.dat
      1,024 100% 34.48kB/s  0:00:00 (xfr#4085, to-chk=295915/300001)
103674.dat
      1,024 100% 34.48kB/s  0:00:00 (xfr#4086, to-chk=295914/300001)
103675.dat
      1,024 100% 33.33kB/s  0:00:00 (xfr#4087, to-chk=295913/300001)
```

----结束

## 验证结果

**步骤1** 登录已创建好的Linux系统云服务器。

**步骤2** 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

**步骤3** 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

## 1.5 文件系统之间迁移数据

### 方案概述

用户可以将SFS容量型文件系统中的数据迁移至SFS Turbo文件系统中，也可以将SFS Turbo文件系统中的数据迁移至SFS容量型文件系统中，进行云上业务拓展。

此方案通过创建一台Linux操作系统的云服务器，来连接SFS容量型文件系统和SFS Turbo文件系统的通信。

### 约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- Linux系统云服务器、SFS容量型文件系统和SFS Turbo文件系统需在同一VPC下。
- 支持实现增量迁移，即只迁移发生变化的数据。

### 前提条件

- 已创建一台操作系统为Linux的云服务器。

- 已创建SFS容量型或SFS Turbo文件系统，并获取到文件系统的挂载地址。

## 资源规划

本章节介绍案例中使用的资源信息。具体如表1-4所示。

表 1-4 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs   16GB   c7.2xlarge.2 操作系统：Linux 区域：中国-香港 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

## 操作步骤

**步骤1** 登录弹性云服务器管理控制台。

**步骤2** 登录已创建好的Linux系统云服务器，用于同时访问SFS容量型文件和SFS Turbo文件系统。

**步骤3** 输入以下挂载命令，用于访问文件系统1。文件系统1可以是SFS容量型文件系统或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统1挂载地址 /mnt/src
```

**步骤4** 输入以下挂载命令，用于访问文件系统2。文件系统2可以是SFS容量型文件或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统2挂载地址 /mnt/dst
```

**步骤5** 下载并安装rclone工具。下载地址请参见<https://rclone.org/downloads/>。

**步骤6** 执行以下命令，进行数据同步。

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

### 📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- /mnt/src：源路径
- /mnt/dst：目标路径
- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。
- --copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

## 验证

**步骤1** 登录已创建好的Linux系统云服务器。

**步骤2** 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst  
ls | wc -l
```

**步骤3** 数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

## 1.6 SFS 容量型文件系统迁移至其他文件系统

### 方案概述

用户可以将SFS容量型文件系统中的数据迁移至通用文件系统或SFS Turbo文件系统中。

此方案通过创建一台Linux操作系统的云服务器，来连接SFS容量型文件系统和其他文件系统的通信。

### 约束与限制

- 仅支持使用Linux系统的云服务器进行数据迁移。
- Linux系统云服务器、SFS容量型文件系统和其他文件系统需在同一VPC下。如果目标端是通用文件系统，需配置VPC终端节点，具体操作，请参见[配置VPC终端节点](#)。
- 支持实现增量迁移，即只迁移发生变化的数据。

### 前提条件

- 已创建一台操作系统为Linux的云服务器。
- 已创建SFS容量型和其他文件系统，并获取到文件系统的挂载地址。

### 资源规划

本章节介绍案例中使用的资源信息。具体如[表1-5](#)所示。

表 1-5 资源规划

产品	配置示例	说明
弹性云服务器 ECS	规格：8vCPUs   16GB   c7.2xlarge.2 操作系统：Linux 区域：中国-香港 VPC名称：VPC1	已创建/mnt/src和/mnt/dst的目录

## 操作步骤

**步骤1** 登录弹性云服务器管理控制台。

**步骤2** 登录已创建好的Linux系统云服务器，用于同时访问SFS容量型文件系统和其他文件系统。

**步骤3** 输入以下挂载命令，用于访问文件系统1。文件系统1是SFS容量型文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统1挂载地址 /mnt/src
```

**步骤4** 输入以下挂载命令，用于访问文件系统2。文件系统2可以是通用文件系统或SFS Turbo文件系统。

```
mount -t nfs -o vers=3,timeo=600,noresvport,nolock 文件系统2挂载地址 /mnt/dst
```

**步骤5** 在Linux云服务器中执行以下命令安装rclone工具。

```
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-*
```

### 📖 说明

rclone工具不保留源端的文件权限或属组信息，有相关需求请使用rsync工具。

**步骤6** 执行以下命令，进行数据同步。

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

### 📖 说明

参数说明如下，transfers和checkers数目可以根据系统规格自行配置：

- /mnt/src：源路径
- /mnt/dst：目标路径
- --transfers：传输文件的并发数目。
- --checkers：扫描本地文件的并发数目。
- -P：数据拷贝进度。
- --links：复制源端的软链接，目的端保持为软链接的形式。
- --copy-links：复制源端软链接指向的文件内容，目的端变成文件的形式，不再是软链接。
- --create-empty-src-dirs：复制源端的空目录到目的端。

等待数据完成同步后，可前往目标文件系统查看是否已成功迁移。

----结束

## 验证

**步骤1** 登录已创建好的Linux系统云服务器。

**步骤2** 在目的端服务器执行以下命令，验证文件同步情况。

```
cd /mnt/dst
ls | wc -l
```

数据量与源端服务器中的一致时，表示数据迁移成功。

----结束

# 2 SFS Turbo 性能测试

fio是一个开源的I/O压力测试工具，可以使用fio工具对SFS进行吞吐量和IOPS的性能测试。

## 前提条件

已在云服务器上安装fio工具。fio可从[官网](#)或[GitHub](#)下载。

## 注意和说明

测试性能依赖client和server之间的网络带宽及文件系统的容量大小。

## 安装 fio

以Linux CentOS系统为例说明：

1. 在官网下载fio。  
**yum install fio**
2. 安装libaio引擎。  
**yum install libaio-devel**
3. 查看fio版本。  
**fio --version**

## 文件系统性能数据

SFS Turbo文件系统的性能主要有IOPS和吞吐量等指标，具体各指标数据参见[表2-1](#)。

表 2-1 性能数据表

参数	普通型		HPC型	
	SFS Turbo标准型	SFS Turbo性能型	125MB/s/TiB	250MB/s/TiB
最大容量	32TB	32TB	1PB	1PB

最大 IOPS	5K	20K	1000K	1000K
最大吞吐量	150 MB/s	350 MB/s	20 GB/s	20 GB/s
IOPS性能计算公式	IOPS = min (5000, 1200 + 6 × 容量) 其中, 容量单 位为GB	IOPS = min (20000, 1500 + 20 × 容量) 其中, 容量单 位为GB	IOPS = min (1000000, 6000 × 容量) 其中, 容量单位 为TB	IOPS = min (1000000, 12500 × 容量) 其中, 容量单位 为TB

## 通用测试配置样例

### 说明

以下提供的预估值为单台弹性云服务器ECS测试的结果。建议使用多台ECS测试，以达到弹性文件服务的性能指标。

本文以SFS Turbo性能型，云服务器规格如下为例说明。

规格：通用计算增强型 | c3.xlarge.4 | 4vCPUs | 16GB

镜像：CentOS 7.564bit

- fio命令：

```
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
```

### 说明

其中，“/mnt/nfs/test\_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/nfs”目录下的“test\_fio”文件，请根据实际情况填写。

- fio结果：

```
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=3344.48
clat (nsec): min=245, max=133325, avg=11162.10, stdev=12136.31
lat (nsec): min=252, max=133330, avg=11166.32, stdev=12136.34
clat percentiles (nsec):
| 1.00th=[ 2245],  5.00th=[ 2540], 10.00th=[ 2671], 20.00th=[ 2900],
| 30.00th=[ 3130], 40.00th=[ 3450], 50.00th=[ 4293], 60.00th=[ 7832],
| 70.00th=[13173], 80.00th=[19792], 90.00th=[20443], 95.00th=[36439],
| 99.00th=[53216], 99.50th=[60031], 99.90th=[79160], 99.95th=[85459],
| 99.99th=[90042]
bw ( KIB/s): min=16600, max=45560, per=100.00%, avg=29696.00, stdev=5544.46, samples=494
iops      : min= 4150, max=11390, avg=7424.01, stdev=1306.11, samples=494
write: IOPS=3182, BW=12.4MiB/s (13.0MB/s)(3073MiB/247160msec)
slat (nsec): min=1480, max=302730, avg=4613.59, stdev=3359.60
clat (nsec): min=1447, max=140666, avg=14166.05, stdev=13373.72
lat (nsec): min=1457, max=140671, avg=14170.73, stdev=13373.74
clat percentiles (nsec):
| 1.00th=[  41],  5.00th=[  41], 10.00th=[  41], 20.00th=[  51],
| 30.00th=[  51], 40.00th=[  61], 50.00th=[  81], 60.00th=[ 141],
| 70.00th=[ 101], 80.00th=[ 241], 90.00th=[ 331], 95.00th=[ 421],
| 99.00th=[ 591], 99.50th=[ 671], 99.90th=[ 871], 99.95th=[ 941],
| 99.99th=[ 122]
bw ( KIB/s): min= 7144, max=19600, per=100.00%, avg=12730.90, stdev=2395.77, samples=494
iops      : min= 1706, max= 4900, avg=3182.70, stdev=590.96, samples=494
lat (nsec): 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%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete : 0=0.0%, 4=100.0%, 0=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=1034036,706604,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=120

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

- fio命令:

```
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
```

#### 📖 说明

其中，“/mnt/nfs/test\_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/nfs”目录下的“test\_fio”文件，请根据实际填写。

- fio结果:

```
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/155280msec)
slat (msec): min=1271, max=269588, avg=4873.51, stdev=3848.12
clat (usec): min=226, max=88185, avg=5711.35, stdev=7879.46
lat (usec): min=232, max=88187, avg=5715.49, stdev=7879.48
clat percentiles (msec):
| 1.00th=[ 1221], 5.00th=[ 1958], 10.00th=[ 2188], 20.00th=[ 2442],
| 30.00th=[ 2686], 40.00th=[ 2882], 50.00th=[ 2999], 60.00th=[ 3228],
| 70.00th=[ 3687], 80.00th=[ 5684], 90.00th=[14222], 95.00th=[21898],
| 99.00th=[35914], 99.50th=[48633], 99.90th=[51643], 99.95th=[55837],
| 99.99th=[66847]
bw ( KIB/s): min=13368, max=28848, per=99.99%, avg=28257.97, stdev=2913.85, samples=318
iops      : min= 3348, max= 7212, avg=5864.48, stdev=728.27, samples=318
write: IOPS=11.8k, BW=46.2MiB/s (48.4MB/s)(7167MiB/155280msec)
slat (msec): min=1396, max=398684, avg=4485.68, stdev=3891.75
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=[  31], 5.00th=[  41], 10.00th=[  41], 20.00th=[  41],
| 30.00th=[  51], 40.00th=[  51], 50.00th=[  51], 60.00th=[  61],
| 70.00th=[  71], 80.00th=[ 131], 90.00th=[ 211], 95.00th=[ 281],
| 99.00th=[ 421], 99.50th=[ 471], 99.90th=[ 681], 99.95th=[ 681],
| 99.99th=[ 1281]
bw ( KIB/s): min=32224, max=67456, per=99.96%, avg=47254.23, stdev=6792.41, samples=318
iops      : min= 8856, max=16864, avg=11813.55, stdev=1698.11, samples=318
lat (usec) : 250=0.81%, 500=0.84%, 750=0.87%, 1000=0.89%
lat (msec) : 2=1.53%, 4=36.85%, 10=41.27%, 20=11.38%, 50=0.61%
lat (msec) : 100=0.23%, 250=0.81%
cpu       : usr=2.13%, sys=9.98%, ctx=925778, 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=100.8%
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
```

#### 顺序读IOPS

- fio命令:

```
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
```

#### 📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际填写。

- fio结果:

```
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)(18.0GiB/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=913130, majf=0, minf=159
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=2621440,0,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=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=18.0GiB (18.7GB), run=2
```

## 随机读IOPS

- fio命令:

```
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
```

### 📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果:

```
test: (g=0): rw=randread, bs=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
```

## 顺序写IOPS

- fio命令:

```
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
```

## 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
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=368726, avg=4388.87, stdev=3688.87
clat (usec): min=1953, max=186548, avg=11588.61, stdev=5876.84
lat (usec): min=1959, max=186552, avg=11593.86, stdev=5876.86
clat percentiles (usec):
| 1.00th=[ 4015], 5.00th=[ 5932], 10.00th=[ 6652], 20.00th=[ 7439],
| 30.00th=[ 8029], 40.00th=[ 8848], 50.00th=[ 9634], 60.00th=[10814],
| 70.00th=[12518], 80.00th=[15533], 90.00th=[19268], 95.00th=[22676],
| 99.00th=[32637], 99.50th=[37487], 99.90th=[49821], 99.95th=[53748],
| 99.99th=[69731]
bw ( KIB/s): min=31712, max=52431, per=99.99%, avg=44158.84, stdev=3987.31, samples=474
iops      : min= 7928, max=13187, avg=11839.58, stdev=996.83, samples=474
lat (msec): 2=0.01%, 4=1.00%, 10=51.94%, 20=38.58%, 50=8.39%
lat (msec): 100=0.00%, 250=0.01%
cpu       : usr=1.33%, sys=5.47%, ctx=392117, majf=0, minf=27
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=8,2621448,0,0 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=
```

## 随机写IOPS

- fio命令：

```
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
```

## 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果：

```
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, aggrbw=18462KB/s, minbw=18462KB/s, maxbw=18462KB/s, mint=567947msec, maxt=567947msec
```

## 顺序读带宽

- fio命令:

```
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
```

#### 📖 说明

其中，“/mnt/sfs-turbo/test\_fio”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfs-turbo”目录下的“test\_fio”文件，请根据实际情况填写。

- fio结果:

```
test: (groupid=0, jobs=1): err= 0: pid=28962: Mon Jun 8 14:37:48 2020
read: IOPS=390, BW=391MiB/s (409MB/s)(10.06iB/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=327.40, stdev=99.63
clat percentiles (msec):
| 1.00th=[ 155], 5.00th=[ 161], 10.00th=[ 167], 20.00th=[ 180],
| 30.00th=[ 360], 40.00th=[ 372], 50.00th=[ 380], 60.00th=[ 381],
| 70.00th=[ 388], 80.00th=[ 393], 90.00th=[ 401], 95.00th=[ 411],
| 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=0.20%, 250=27.61%, 500=71.37%, 750=0.58%
cpu       : usr=0.00%, sys=4.21%, ctx=18395, majf=0, minf=97
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued   : 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=10.06iB (10.76B), run=26221-26221msec
```

## 随机读带宽

- fio命令:

```
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
```

#### 📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果:

```
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, aggrbw=154129KB/s, minbw=154129KB/s, maxbw=154129KB/s, mint=68032msec, maxt=68032msec
```

## 顺序写带宽

- fio命令:

```
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
```

### 📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果:

```
test: (groupid=0, jobs=1): err=0: pid=21889: Mon Jun 8 14:53:44 2020
write: IOPS=243, BW=244MiB/s (255MB/s)(10.0GiB/42048msec)
slat (usec): min=183, 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=[ 100], 10.00th=[ 167], 20.00th=[ 292],
| 30.00th=[ 422], 40.00th=[ 468], 50.00th=[ 506], 60.00th=[ 550],
| 70.00th=[ 625], 80.00th=[ 760], 90.00th=[ 902], 95.00th=[ 970],
| 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=249500.99, stdev=147656.62, samples=83
iops      : min=   4, max=  458, avg=243.63, stdev=144.22, samples=83
lat (msec): 20=0.03%, 50=0.96%, 100=3.36%, 250=12.55%, 500=31.63%
lat (msec): 750=30.07%, 1000=18.96%
cpu       : usr=2.28%, sys=2.50%, ctx=3972, majf=0, minf=27
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued rwts: total=0,10240,0,0 short=0,0,0,0 dropped=0,0,0,0
latency   : target=0, window=0, percentile=100.00%, depth=128

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

## 随机写带宽

- fio命令:

```
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
```

### 📖 说明

其中，“/mnt/sfs-turbo/”为待测试的目标文件的挂载路径，需具体到文件名，请根据实际情况填写。

- fio结果:

```
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=[ 97],
| 30.00th=[ 97], 40.00th=[ 98], 50.00th=[ 98], 60.00th=[ 99],
| 70.00th=[ 99], 80.00th=[ 100], 90.00th=[ 100], 95.00th=[ 101],
| 99.00th=[ 102], 99.50th=[ 102], 99.90th=[ 103], 99.95th=[ 104],
| 99.99th=[ 195]
bw (KB /s): min=150104, max=180654, per=98.76%, avg=154058.04, stdev=3404.48
lat (msec): 20=0.04%, 50=13.44%, 100=1.04%, 250=0.73%, 500=1.05%
lat (msec): 750=0.04%, 1000=60.69%, 2000=22.97%
cpu      : usr=0.91%, sys=1.52%, ctx=2011, majf=0, minf=28
IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.2%, 32=0.3%, >=64=99.4%
submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%
issued    : total=r=0/w=10240/d=0, short=r=0/w=0/d=0
latency   : target=0, window=0, percentile=100.00%, depth=128

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

# 3 SFS Turbo 时延测试

本文以SFS Turbo性能增强型，云服务器规格如下为例说明。

规格：通用计算增强型 | c6.4xlarge.4 | 16vCPUs | 64GB

镜像：EulerOS 2.5

## 单队列随机读

- fio命令：

```
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
```

### 📖 说明

其中，“/mnt/sfsturbo/fio\_test\_01 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio\_test\_01 -name”文件，请根据实际情况填写。

- fio结果：

```
[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
reads: IOPS=1401, 0m=5600KiB/s (5740KB/s) (328MiB/60001msec)
slat (nsec): min=2379, max=51192, avg=4228.52, stdev=1319.26
clat (usec): min=562, max=8407, 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=[ 1350], 99.50th=[ 2040], 99.90th=[ 3523], 99.95th=[ 3982],
  | 99.99th=[ 5755]
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=80.70%, 1000=9.21%
lat (msec) : 2=1.39%, 4=0.40%, 10=0.05%
cpu       : usr=0.86%, sys=1.77%, ctx=84892, 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=5600KiB/s (5740KB/s), 5600KiB/s-5600KiB/s (5740KB/s-5740KB/s), io=328MiB (344MB), run=60001.60001msec
[root@100 ~]#
```

## 单队列随机写

- fio命令：

```
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
```

### 说明

其中，“/mnt/sfsturbo/fio\_test\_02 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio\_test\_02 -name”文件，请根据实际填写。

- fio结果:

```
[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_test
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=25783: Tue Nov 8 21:37:13 2022
write: IOPS=1661, BW=6645KIB/s (6884kB/s) (389MB/6000msec)
slat (msec): min=2566, max=34191, avg=4528.29, stdev=1661.31
clat (usec): min=473, max=8238, avg=595.67, stdev=169.48
lat (usec): min=477, max=8234, avg=600.38, stdev=169.55
clat percentiles (usec):
| 1.00th=[ 515], 5.00th=[ 529], 10.00th=[ 537], 20.00th=[ 553],
| 30.00th=[ 562], 40.00th=[ 578], 50.00th=[ 578], 60.00th=[ 586],
| 70.00th=[ 594], 80.00th=[ 611], 90.00th=[ 635], 95.00th=[ 668],
| 99.00th=[ 938], 99.50th=[ 1383], 99.90th=[ 3130], 99.95th=[ 3949],
| 99.99th=[ 5669]
bw ( KIB/s): min= 6288, max= 6968, per=100.00%, avg=6044.03, stdev=152.58, samples=119
iops    : min=1572, max=1742, avg=1668.99, stdev=30.16, samples=119
lat (msec)  : 500=0.12%, 750=97.53%, 1000=1.51%
lat (msec)  : 2=0.59%, 4=0.20%, 10=0.05%
cpu        : usr=1.12%, sys=2.03%, ctx=98979, 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 rts: total=0, short=0, 0 short=0, 0 dropped=0, 0 0
latency    : target=0, window=0, percentile=100.00%, depth=1
Run status group 0 (all jobs):
WRITE: bw=6645KIB/s (6884kB/s), 6645KIB/s-6645KIB/s (6884kB/s-6884kB/s), io=389MB (408MB), run=60001.60001msec
[root@100 ~]#
```

### 单队列顺序读

- fio命令:

```
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
```

### 说明

其中，“/mnt/sfsturbo/fio\_test\_03 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio\_test\_03 -name”文件，请根据实际填写。

- fio结果:

```
[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
Starting 1 process
Jobs: 1 (f=1): [R(1)][100.0%][r=6412KIB/s,w=0KIB/s][r=1602,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) (370MB/6000msec)
slat (msec): min=2349, max=30953, avg=4152.29, stdev=1412.71
clat (usec): min=472, max=8642, avg=629.77, 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= 5888, 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 (msec)  : 500=0.01%, 750=96.40%, 1000=2.54%
lat (msec)  : 2=0.01%, 4=0.20%, 10=0.04%
cpu        : usr=0.69%, 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 rts: total=94593, 0 0 short=0, 0 0 dropped=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=370MB (387MB), run=60001.60001msec
[root@100 ~]#
```

### 单队列顺序写

- fio命令:

```
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
```

### 说明

其中，“/mnt/sfsturbo/fio\_test\_04 -name”为待测试的目标文件的挂载路径，需具体到文件名，即这里要测试的是“/mnt/sfsturbo”目录下的“fio\_test\_04 -name”文件，请根据实际填写。

- fio结果:

```
[root@100 ~]# fio --direct=1 --ioengine=libaio --bs=4k --size=100 --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): [(w1)][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=389: Tue Nov 9 21:40:23 2022
write: IOPS=1799, BW=7194KIB/s (7367kB/s)(422MiB/6000Imsec)
slat (msec): 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
lat 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=1799.35, stdev=53.20, 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 dropped=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 (442MiB), run=60001-60001msec
[root@100 ~]#
```

# 4 为文件系统创建普通用户可读写的子目录

---

## 4.1 方案概述

### 应用场景

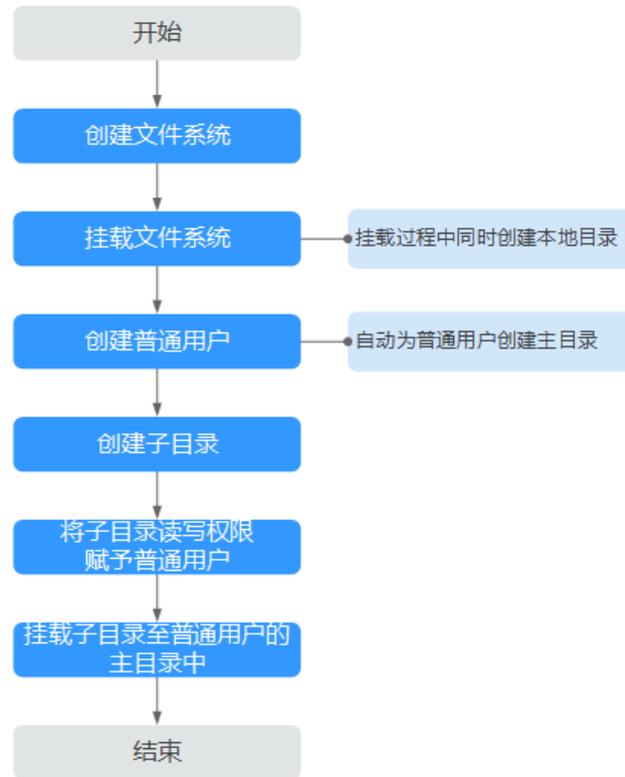
文件系统创建成功后，文件系统默认情况下只有root用户具有读写权限。

root用户可以根据需要，可以授权多个普通用户访问权限，并在文件主目录下为每个普通用户创建多个子目录并挂载至主目录上，同时使普通用户拥有读写权限。

### 操作流程

为文件系统创建普通用户可读写的子目录的流程如[图4-1](#)所示。

图 4-1 操作流程



## 4.2 资源和成本规划

本节介绍最佳实践中资源规划情况，包含以下内容：

表 4-1 资源和成本规划

资源	资源说明
弹性云服务器ECS	云服务器和文件系统归属于同一项目下，用于挂载共享路径实现数据共享。
虚拟私有云VPC	VPC为弹性云服务器构建隔离的、用户自主配置和管理的虚拟网络环境，提升用户云中资源的安全性，简化用户的网络部署。 云服务器无法访问不在同一VPC下的文件系统，使用弹性文件服务时需将文件系统和云服务器归属于同一VPC下。
文件系统	文件系统通过标准的NFS协议和CIFS协议为客户提供文件存储服务，用于网络文件远程访问，用户通过管理控制台创建挂载地址后，即可在多个云服务器上进行挂载，并通过标准的POSIX接口对文件系统进行访问。

## 4.3 实施步骤

### 4.3.1 创建文件系统本地目录

当创建文件系统后，您需要将文件系统挂载至云服务器上，在为root用户创建一个本地目录。

如已成功挂载文件系统，可跳过此章节。请记录[步骤4](#)的本地目录后，按照[为每个用户创建有读写权限的子目录](#)执行操作。

#### 前提条件

- 确定云服务器操作系统类型，不同操作系统安装NFS客户端的命令不同。
- 已完成创建文件系统，并获取到文件系统的挂载地址。
- 选择与文件系统所属VPC相同的云服务器进行挂载。
- 云服务器上已配置了用于内网解析文件系统域名的DNS服务器的IP地址。

#### 操作步骤

**步骤1** 已经成功在北京一可用区2创建一台CentOS系统的云服务器如ecs-whm，如[图4-2](#)所示。

图 4-2 成功创建云服务器



**步骤2** 以root用户登录弹性云服务器。安装NFS客户端。

1. 查看系统是否安装NFS软件包。

- CentOS、Red Hat、Oracle Enterprise Linux、SUSE、Euler OS、Fedora或OpenSUSE系统下，执行如下命令：

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

- Debian或Ubuntu系统下，执行如下命令：

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

不同操作系统回显会有所不同，如果回显如下类似信息，说明已经成功安装NFS软件包，执行[步骤3](#)。如未显示，执行[步骤2.2](#)。

- CentOS、Red Hat、Euler OS、Fedora或Oracle Enterprise Linux系统下，回显如下类似信息：

```
libnfsidmap  
nfs-utils
```

- SUSE或OpenSUSE系统下，回显如下类似信息：

```
nfsidmap  
nfs-client
```

- Debian或Ubuntu系统下，回显如下类似信息：

```
nfs-common
```

2. 如果查看到未安装，根据不同的操作系统，执行不同命令。

### 📖 说明

执行以下命令前要求云服务器已连接到互联网，否则安装NFS客户端失败。

- CentOS、Red Hat、Euler OS、Fedora或Oracle Enterprise Linux系统下，执行如下命令：

```
sudo yum -y install nfs-utils
```

- Debian或Ubuntu系统下，执行如下命令：

```
sudo apt-get install nfs-common
```

- SUSE或OpenSUSE系统下，执行如下命令：

```
zypper install nfs-client
```

**步骤3** 执行如下命令，查看是否能解析文件系统挂载地址中的域名。如图4-3所示。

**nslookup 文件系统域名**

```
nslookup sfs-nas1.xx-xxxx-xx.xxxxxxxxxxxx.com
```

### 📖 说明

- 文件系统域名仅为域名，如：sfs-nas1.xxx.com。文件系统域名请从文件系统的挂载地址中获取，不需要输入整个挂载地址。
- 无法使用nslookup命令时，需要先安装bind-utils软件包。（可通过执行yum install bind-utils命令安装）
- 解析成功，执行**步骤4**。
- 解析失败，请先完成DNS服务器IP地址的配置再执行挂载文件系统的操作，具体配置操作请参见**配置DNS**。

图 4-3 解析域名

```
[root@ecs-whm ~]# rpm -qa|grep nfs
libnfsidmap-0.25-19.el7.x86_64
nfs-utils-1.3.0-0.61.el7.x86_64
[root@ecs-whm ~]# nslookup sfs-nas1.
-bash: nslookup: command not found
[root@ecs-whm ~]# yum install bind-utils
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
```

**步骤4** 执行如下命令，创建用于挂载文件系统的本地目录。此时记录本地目录名字，此处以root001为例。

**mkdir 本地目录**

```
mkdir root001
```

**步骤5** 执行如下命令，将文件系统挂载到云服务器上。文件系统目前仅支持NFSv3协议挂载到Linux云服务器，其中变量说明见**表4-2**。

```
mount -t nfs -o vers=3,timeo=600,nolock 挂载地址 本地目录
```

### 须知

已挂载文件系统的云服务器重启后，该云服务器上的挂载信息将会丢失，您可以通过在fstab文件中配置自动挂载来保证云服务器重启时自动挂载文件系统，具体操作请参见**自动挂载文件系统**。

表 4-2 参数说明

参数	说明
vers	文件系统版本，目前只支持NFSv3。取值：3。
timeo	NFS客户端重传请求前的等待时间(单位为0.1秒)。建议值：600。
lock/nolock	选择是否使用NLM协议在服务器上锁文件。当选择nolock选项时，锁对于同一主机的应用有效，对不同主机不受锁的影响。建议值：nolock。
挂载地址	SFS文件系统的格式为：文件系统域名:/路径，例如：example.com:/share-xxx。SFS Turbo文件系统的格式为：文件系统IP:/，例如192.168.0.0:/。 <b>说明</b> x是数字或字母。 由于挂载地址名称较长，需要拉宽该栏以便完整显示。
本地目录	云服务上用于挂载文件系统的本地目录，例如“/local_path”。

图 4-4 挂载地址



**步骤6** 挂载完成后，执行如下命令，查看已挂载的文件系统。

```
mount -l
```

如果回显包含如下类似信息，说明挂载成功。

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

**步骤7** 挂载成功后，参考下一个章节创建普通用户和子目录。

如果挂载失败或超时，请参考[故障排除](#)处理。

#### 📖 说明

支持写入的单个文件最大容量为240TB。

----结束

## 4.3.2 为每个用户创建有读写权限的子目录

### 前提条件

- 已创建文件系统，并能通过root账号成功挂载到云服务器上。操作步骤请参考[创建文件系统本地目录](#)。
- 已获取到文件系统的挂载地址。

### 操作步骤

**步骤1** 以root账号登录弹性云服务器。

**步骤2** 使用以下命令在root账号下使用以下命令添加普通用户账号，下面以添加普通用户Tom为例。

```
adduser Tom  
passwd Tom
```

根据回显提示修改普通用户Tom的密码，创建成功后会自动创建用户Tom的主目录/home/Tom。

**步骤3** 添加普通用户Tom成功后，再在root的本地目录下使用以下命令为普通用户Tom创建子目录。

由[创建文件系统本地目录](#)章节的**步骤4**可知root的本地目录为root001，使用以下命令为普通用户Tom创建子目录Tom。其中root001需替换为实际的本地目录。

```
mkdir /root/root001/Tom
```

**步骤4** 使用以下命令将子目录Tom的读写权限赋予普通用户Tom。其中root001需替换为实际的本地目录。

```
chown Tom:Tom /root/root001/Tom
```

创建完后可以通过以下命令验证普通用户Tom是否已有子目录Tom的读写权限。如[图4-5](#)所示。

```
cd /home  
cd /root/root001  
ll
```

图 4-5 查询权限

```
[root@ecs-whm home]# cd /root/root001  
[root@ecs-whm root001]# ll  
total 4  
drwxr-xr-x 2 Tom Tom 4096 May 13 14:31 Tom
```

可以看到用户Tom已经获取了子目录Tom的读写权限。

**步骤5** 将root001/Tom子目录使用挂载命令mount挂载到Tom的主目录/home/Tom中。其中xx-xxxx-xx需要替换为文件系统所在区域，share-xxxx需要修改为实际的文件系统，第一个Tom需要修改为实际的子目录名称。如[图4-6](#)所示。

```
mount -t nfs sfs-nas1.xx-xxxx-xx.xxxxxxxxxx.com:/share-xxxx/Tom /home/Tom
```

图 4-6 挂载目录

```
[root@ecs-whm root001]# mount -t nfs sfs-nas1.xxxxxxxxxx.com:/share-xxxx/Tom /home/Tom  
[1282638.340788] Key type dns_resolver registered  
[1282638.381991] NFS: Registering the id_resolver key type  
[1282638.382851] Key type id_resolver registered  
[1282638.383689] Key type id_legacy registered
```

执行以下命令可以查看到该子目录已经成功挂载至Tom的主目录中。如[图4-7](#)所示。

```
df -h
```

图 4-7 查询挂载结果

```
[root@ecs-whm root001]# df -h
Filesystem                Size      Used Avail Use% Mounted on
/dev/vda1                  40G       1.9G   36G   5% /
devtmpfs                   486M         0  486M   0% /dev
tmpfs                       496M         0  496M   0% /dev/shm
tmpfs                       496M       26M  471M   6% /run
tmpfs                       496M         0  496M   0% /sys/fs/cgroup
tmpfs                       100M         0   100M   0% /run/user/0
sfs-nas1.                  100G         0   100G   0% /root/root001
sfs-nas1.                  100G         0   100G   0% /home/Tom
```

----结束