



NFS 网关

用户使用指南

天翼云科技有限公司

目录

1	简介	3
1.1	术语和缩略词.....	3
1.2	产品架构.....	3
2	前期准备工作	4
2.1	NFS 网关云主机创建	4
2.2	ZOS 对象存储创建	5
2.3	客户端准备工作.....	5
3	NFS 网关管理.....	6
3.1	NFS 共享创建与删除	6
3.2	NFS 共享查询与修改	8
3.3	NFS 共享暂停与恢复	10
4	NFS 网关使用.....	11
4.1	云主机作为客户端挂载.....	11
4.2	本地主机作为客户端挂载.....	12
5	附录	12
5.1	创建共享时 config.py 参数格式限制	12

1 简介

NFS 网关是以天翼云 ZOS 作为后端存储,为客户提供标准的文件存储服务。

1.1 术语和缩略词

NFS 网关	-	NFS 网关提供基本的协议转换和简单的连接性,用户本地通过标准存储协议(NFS)访问网关,实现本地至天翼云 ZOS 数据同步。
对象存储	ZOS	天翼云对象存储(简称 ZOS)是天翼云为客户提供的一种海量、弹性、高可靠、高性价比的存储产品,是专门针对云计算、大数据和非结构化数据的海量存储形态,通过 S3 协议和标准的服务接口,提供非结构化数据(图片、音视频、文本等格式文件)的无限存储服务。
Network File System	NFS	NFS(Network File System)即网络文件系统,是 FreeBSD 支持的文件系统中的一种,它允许网络中的计算机之间通过 TCP/IP 网络共享资源。

1.2 产品架构

NFS 网关部署在天翼云上,用户本地通过 NFS 协议将数据上传至 NFS 网关, NFS 网关将数据缓存,并定期同步至 ZOS 对象存储,用户可选择通过本地主机或天翼云云主机作为 NFS 客户端访问 NFS 网关。NFS 网关架构如下:

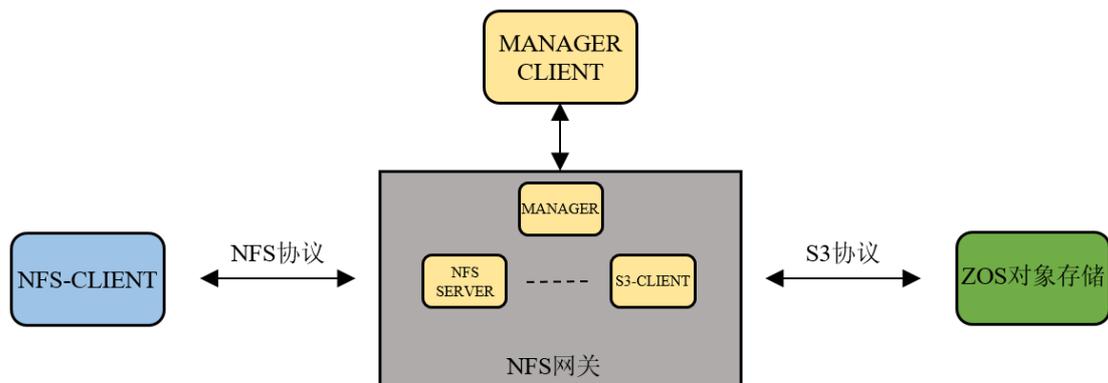


图 1-1 NFS 网关架构

2 前期准备工作

目前使用 NFS 网关需要首先创建部署网关的云主机，并申请天翼云对象存储 ZOS 作为后端存储。

2.1 NFS 网关云主机创建

(1) NFS 网关主要是通过 NFS 网关私有镜像（部署好 NFS 网关的镜像文件）来申请云主机进行部署，在官网控制台将私有镜像进行导入，并通过该私有镜像创建云主机，具体的私有镜像导入方式见《导入私有镜像用户操作指南》。

进入创建私有镜像方式：官网控制中心->弹性云主机->镜像服务->私有镜像->创建私有镜像。图 2-1 为创建私有镜像界面。



图 2-1 私有镜像创建界面

(2) 私有镜像创建完成，使用该私有镜像申请云主机，创建部署 NFS 网关的云主机。推荐云主机配置为：vCPU-8 核、内存-16G。并根据要在网关上申请的共享个数，创建对应个数的缓存盘。缓存盘推荐规格为超高 IO。缓存盘大小可根据业务场景中文件规模和数据量估算。申请云主机方法：官网控制台弹性云主机->创建云主机。图 2-2 为创建云主机时选择镜像位置与申请缓存盘规格位置。选择上一步创建的私有镜像来创建云主机，创建时记录创建云主机时选择的网卡设置，如图 2-3 所示，若有本地主机部署客户端的需求，可申请弹性公网 ip。



图 2-2 私有镜像与缓存盘选择



图 2-3 网卡网段信息

(3) 填写云主机的用户名和密码，并完成创建。

(4) 想要使用 NFS 网关，需要用户申请普通云主机作为客户端来挂载访问 NFS 共享，若创建云主机时申请了弹性公网 IP 则也可通过本地主机进行挂载访问。

2.2 ZOS 对象存储创建

NFS 网关使用 ZOS 对象存储作为后端存储，申请云主机后，需要开通 ZOS 服务，并在相应资源池申请对象存储 bucket。

(1) 进入官网控制中心->对象存储，选择并创建 Bucket。

(2) 进入接入管理(VPC)，添加创建云主机时选择的网卡，来实现内网访问。

(3) 在 Access Key 管理中查看对象存储的 AK、SK。

(4) 进入创建完成的 Bucket 中查看域名，如图 2-4 所示，由于配置了 VPC，可以使用 VPC 内网访问域名进行访问。

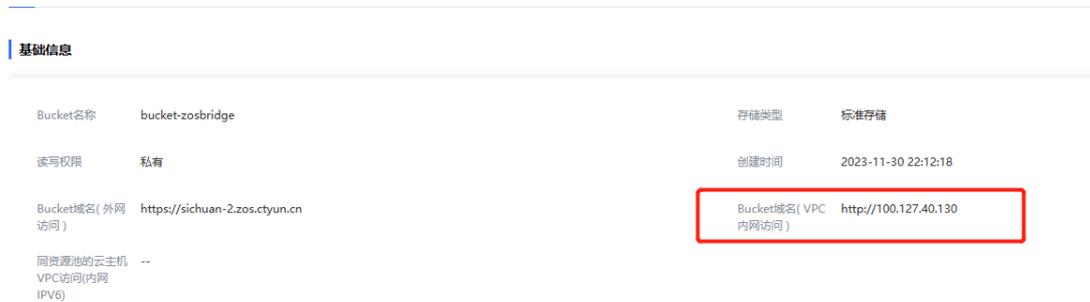


图 2-4 内网 Bucket 域名

2.3 客户端准备工作

客户端主机在使用 NFS 网关完成挂载需要进行的前置准备：

- 通过指令进行安装 NFS：`yum install -y rpcbind nfs-utils`,
- 执行 `systemctl start rpcbind` 与 `systemctl start nfs` 开启 NFS 服务。

3 NFS 网关管理

当前需要使用脚本进行 NFS 网关的管理, 下载管控脚本至 NFS 网关云主机。脚本目录结构如下图 2-5 所示, 其中 config.py 为配置文件, 用于填写网关创建共享时所需参数。其余脚本用于完成 NFS 网关共享的创建、删除、查询、修改、暂停和恢复六个功能。

```
[root@hhcc-test wy-script]# ll
total 36
-rw-r--r-- 1 root root 728 Dec 12 11:28 config.py
-rw-r--r-- 1 root root 1288 Dec 12 11:28 create.py
-rw-r--r-- 1 root root 338 Dec 12 13:59 delete.py
-rw-r--r-- 1 root root 285 Dec 12 10:11 get_all.py
-rw-r--r-- 1 root root 321 Dec 12 10:11 get_one.py
-rw-r--r-- 1 root root 583 Dec 12 14:37 patch.py
drwxr-xr-x 2 root root 35 Dec 12 13:59 __pycache__
-rw-r--r-- 1 root root 110 Dec 12 10:11 README.md
-rw-r--r-- 1 root root 498 Dec 12 10:11 resume.py
-rw-r--r-- 1 root root 499 Dec 12 10:11 suspend.py
```

图 2-5 脚本目录格式

3.1 NFS 共享创建与删除

首先打开 config.py 配置文件。配置文件内容如图 2-6 所示。其中各个参数含义为:

```
SERVER_ADDR = '127.0.0.1:8080'
#Access Key
AK='ak'
#Secret Key
SK='sk'
#Bucket域名(VPC内网访问)
ZOS_ENDPOINT = 'endpoint'
#用户指定的挂载目录下, 允许该网段客户端挂载:可以为固定ip, 也可以为一个网段。
#例:192.168.0.76 或 192.168.0.0/24
IP = "ip"
#读写权限:rw\ro
ALLOW = "rw"
#对象存储桶名
BUCKET = "bucket"
#若需要挂载至对象存储桶的子目录则指定, 否则指定为空即可
SUBDIR = "/subdir"
#mount_point挂载点, 前缀固定为/exports/, 这里只需指定挂载点的名称即可
MP = "gntest1"
#用于缓存的缓存盘名称
CACHE = "/dev/vdb"
```

图 2-6 config.py 文件信息

- SERVER_ADDR: 服务器地址, 管控服务监听 8080 端口号, 在 NFS 网关服务端执行脚本。该处填写 127.0.0.1:8080 即可。
- AK 和 SK: 用户密钥, 此处填写在 2.2 中创建 ZOS 对象存储时保存的 AK、SK。

- ZOS_ENDPOINT: 对象存储域名, 此处填写 2.2 中保存的对象存储内网 VPC 域名。
- IP: 客户端 IP, 此处填写想要挂载至 NFS 网关共享的客户端的 IP 地址或网段, 当客户端为天翼云云主机时, 可通过内网访问, 此处填写客户端内网 IP, 若部署至本地主机, 此处应填写客户端公网 IP。若指定为"*"则表示允许所有 IP 的客户端访问。
- ALLOW: NFS 共享读写权限, 可选为"ro"只读或"rw"读写。
- BUCKET: 对象存储桶名, 为在 2.2 中创建 ZOS 指定的 bucket 名称, 需注意在不指定 subdir 时, 创建不同的网关共享需要指定不同的 bucket。
- SUBDIR: 对象存储 Bucket 子目录名, 用户可选择存储至 Bucket 中的某个 subdir 子目录中, 该参数可不指定。若指定该参数, 则不同共享可存储至相同 Bucket 中, 但不同 Subdir 不能存在包含关系。
- MP: 服务端挂载点, 该参数代表后续客户端挂载时需要指定的服务端挂载点, 由于挂载点前缀固定为"/exports/"因此进行隐藏, 此处只需填写挂载点名称, 可自行指定, 同时该处可以指定为多级目录, 但是不同挂载点之间不可存在包含关系。
- CACHE: 缓存盘, 在 2.1 中申请云主机时的数据盘名称, 可通过 `fdisk -l` 查看, 此处一个缓存盘唯一对应一个网关共享, 且该参数必须指定。
- USER_MAPPING: NFS 客户端用户与 NFS 服务器之间的映射关系, 可选参数为"no_root_squash"、"root_squash"、"all_squash"三种, 可不指定该字段, 若进行指定则默认选择为第一个。

参数填写完毕后, 保存退出。

(1) NFS 共享创建

- 执行脚本进行创建: `python3 create.py`
- 注意事项:
 - 需要保证 `create.py` 和 `config.py` 在同一目录下。
 - 保证所有参数和配置均按规定输入, 否则会创建失败。

```
挂载点 : NFS网关ip:/exports/test1
client mount usage: mount -t nfs -o sync NFS网关ip:/exports/test1 [client mount dir]
```

创建成功后会出现上图界面, 根据上图显示的挂载点和挂载命令提示, 在客户端进行 nfs 挂载。其中 `client mount dir` 为用户在客户端中想要指定的挂载点。若创建失败会返回错误消息, 提示出错原因。

```
{"msg": "cache disk not exists"}
```

(2) NFS 共享删除

- 执行脚本删除 config.py 中指定创建的共享: `python3 delete.py`
- 注意事项:
 - 保证 delete.py 与 config.py 在同一目录下
 - config.py 与创建时保持一致

删除成功后出现以下提示, 返回 204 说明删除成功, 失败会返回错误信息:

```
[root@hhcc-test wy-script]# python3 delete.py
204
```

3.2 NFS 共享查询与修改

NFS 网关共享创建完成后, 可选择查询已创建的共享信息或修改已存在的共享权限。

(1) NFS 共享查询

- 执行脚本用于查看 config.py 中指定创建的共享: `python3 get_one.py`
 - 注意事项:
 - 保证 get_one.py 与 config.py 在同一目录下
 - config.py 与创建时保持一致
 - 保证待查询的共享已创建。
 - 执行脚本查看当前 NFS 网关中已经创建的所有共享: `python3 get_all.py`。
- 查询成功后会返回指定共享或所有共享的信息, 失败会返回错误信息:

```
[root@hhcc-test wy-script]# python3 get_one.py
{
  "s3_path": {
    "url": "http://100.127.40.130",
    "bucket": "nfs-xntest4",
    "subdir": ""
  },
  "cache": "/dev/vde",
  "sync_status": "Synchronized",
  "active": true,
  "mount_point": "/exports/xntest4",
  "permission": [
    {
      "ip": "10.0.1.53",
      "allow": "rw"
    }
  ],
  "user_mapping": "no_root_squash",
  "client": []
}
```

- s3_path: 对象存储信息, 返回该共享使用的 ZOS 信息, 其中 url 对应对象存储域名, bucket 与 subdir 分别是对象存储桶名与桶的子目录 (若存在)。
- cache: 共享使用的缓存盘, NFS 网关中当前 NFS 共享使用的缓存盘名称。
- sync_status: 缓存盘同步状态, 表示当前 NFS 共享的缓存盘是否存在待同步

数据，若不存在未同步数据为”Synchronized”，若存在未同步数据则为”Synchronizing”。

- active: 当前 NFS 共享的活动状态，true 表示运行中，false 表示已暂停。
- mount_point: 当前 NFS 共享的服务端挂载点。
- permission: 当前 NFS 共享允许访问的客户端 IP 与读写权限。
- client: 当前 NFS 共享已经挂载的客户端 IP。

(2) NFS 共享修改

- 打开脚本 patch.py，修改 payload 中的 permission 字段中 ip 与 allow，该字段含义同 config.py 中的含义，其中 Type 字段指定为 Modify，表示进行 NFS 共享权限修改。

```
# 脚本用于修改已有的NFS导出的权限控制，修改payload中的权限配置后执行脚本
conn = http.client.HTTPConnection(config.SERVER_ADDR)
payload = json.dumps({
    "Type": "Modify",
    "permission": [
        {
            "ip": "10.0.2.61",
            "allow": "rw" #rw,ro
        }
    ]
})
```

- 执行脚本修改 config.py 中指定创建的共享: `python3 patch.py`
- 注意事项:
 - 保证 patch.py 与 config.py 在同一目录下
 - config.py 与创建时保持一致
 - 保证待修改的共享已创建。
 - payload 中 permission 填写格式正确。

修改成功后会返回修改后的 NFS 共享信息，失败会返回错误信息：

```
[root@hhcc-test wy-script]# python3 patch.py
{
  "s3_path": {
    "url": "http://100.127.40.130",
    "bucket": "nfs-xntest4",
    "subdir": ""
  },
  "cache": "/dev/vde",
  "sync_status": "Synchronized",
  "active": true,
  "mount_point": "/exports/xntest4",
  "permission": [
    {
      "ip": "10.0.2.61",
      "allow": "rw"
    }
  ],
  "user_mapping": "no_root_squash",
  "client": []
}
```

3.3 NFS 共享暂停与恢复

NFS 网关共享支持暂停与恢复功能，当某一共享暂时无需使用时，可以通过脚本来进行共享的暂停和恢复。

(1) NFS 共享暂停

- 执行脚本暂停 `config.py` 中指定创建的共享：`python3 suspend.py`
- 注意事项：
 - 保证 `suspend.py` 与 `config.py` 在同一目录下
 - `config.py` 与创建时保持一致
 - 保证待暂停的共享处于活动状态。

暂停成功后，通过查询脚本查看 NFS 共享信息，可以观察 `active` 字段为 `false` 表示暂停成功。

```
[root@hhcc-test wy-script]# python3 suspend.py
null
[root@hhcc-test wy-script]# python3 get_one.py
{
  "s3_path": {
    "url": "http://100.127.40.130",
    "bucket": "nfs-xntest4",
    "subdir": ""
  },
  "cache": "/dev/vde",
  "sync_status": "Synchronized",
  "active": false,
  "mount_point": "/exports/xntest4",
  "permission": [
    {
      "ip": "10.0.2.61",
      "allow": "rw"
    }
  ],
  "user_mapping": "no_root_squash",
  "client": []
}
```

(2) NFS 共享恢复

- 执行脚本恢复 `config.py` 中指定创建的共享：`python3 resume.py`
- 注意事项：
 - 保证 `resume.py` 与 `config.py` 在同一目录下
 - `config.py` 与创建时保持一致
 - 保证待恢复的共享处于暂停状态。

恢复成功后，通过查询脚本查看 NFS 共享信息，可以观察 `active` 字段为 `true` 表示恢复成功。

```
[root@hhcc-test wy-script]# python3 resume.py
null
[root@hhcc-test wy-script]# python3 get_one.py
{
  "s3_path": {
    "url": "http://100.127.40.130",
    "bucket": "nfs-xntest4",
    "subdir": ""
  },
  "cache": "/dev/vde",
  "sync_status": "Synchronized",
  "active": true,
  "mount_point": "/exports/xntest4",
  "permission": [
    {
      "ip": "10.0.2.61",
      "allow": "rw"
    }
  ],
  "user_mapping": "no_root_squash",
  "client": []
}
```

4 NFS 网关使用

使用前提:

- 已经执行第 2 章前期准备工作，申请部署了 NFS 网关的天翼云云主机、ZOS 对象存储(VPC 指定云主机子网)、客户端主机已经安装并开启 NFS 服务。
- 执行了 3.1 中 NFS 网关共享创建，并指定正确的客户端 IP 与读写权限。
- 若需要使用本地主机作为客户端进行挂载使用，确保部署了 NFS 网关的云主机已申请弹性公网 IP。

4.1 云主机作为客户端挂载

(1) 客户端挂载

客户端云主机与部署了 NFS 网关的云主机配置完成后可以开始挂载使用。

- 客户端执行 `mkdir -p /mnt/挂载目录` 创建需要挂载的目录
- 执行命令: `mount -t nfs -o sync [挂载点]/mnt/[挂载目录]`。挂载命令可以根据 3.1 中创建共享时脚本返回的提示进行挂载，其中挂载点为[NFS 网关 ip:NFS 网关共享目录]。
- 挂载成功后，可通过 `df -h` 与 `mount` 命令查看挂载是否成功，也可以在挂载目录写入文件并查看对象存储对应 bucket 是否上传成功。

```
10.0.3.107:/exports/xntest4 256T 0 256T 0% /mnt/xntest4
10.0.3.107:/exports/xntest4 on /mnt/xntest4 type nfs (rw,relatime,sync,vers=3,rsize=1048576,wsz=1048576,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,mountaddr=10.0.3.107,mountvers=3,mountport=30001,mountproto=udp,local_lock=none,addr=10.0.3.107)
```

(2) 客户端卸载

客户端卸载只需执行命令: `umount /mnt/[挂载目录]`。即可成功卸载。

4.2 本地主机作为客户端挂载

创建 NFS 网关共享时指定客户端公网 IP 为参数，创建共享。

(1) 客户端挂载

客户端挂载命令同 4.1。挂载点 ip 需要指定为部署 NFS 网关的服务端云主机的弹性公网 IP。

(2) 客户端卸载

同 4.1。

5 附录

5.1 创建共享时 config.py 参数格式限制

名称	必填	输入要求
共享个数		一个 NFS 网关允许创建 4 个共享。
AK	是	不为空、a-z、A-Z、0-9、-、_
SK	是	不为空、a-z、A-Z、0-9、-、_、*
BUCKET	是	不为空、a-z、A-Z、0-9、-、_
ZOS_ENDPOINT	是	IP、IP: 端口、URL(注: AK、SK、URL 和 BucketName 需要能找到桶)
SUBDIR	否	名称前需指定"/"，如"/subdir"，subdir 不能包含特殊字符，长度限制在 255 之内。同时多个共享间的 subdir 不能存在包含关系。
CACHE	是	需要指定存在的数据盘，且必须指定，每个共享对应唯一缓存盘
MP	是	用户指定的挂载点名称，不同共享之间不能存在包含关系，长度限制在 255 之内，且不能包含特殊字符
IP	是	指定可以与服务端进行通信的客户端 IP
ALLOW	是	指定允许客户端的访问读写权限，限制指定为"ro"、"rw"中的一个
USER_MAPPING	否	指定为 3.1 中规定的三种类型，可以不指定，但若指定错误则无法创建成功。