

准备

hostname	ip	
test-n1	10.8.30.152	nfs server
test-n2	10.8.30.150	nfs client
...		

搭建 NFS 服务器

1. 安装 NFS 服务

```
$ sudo apt update  
  
$ sudo apt install nfs-kernel-server
```

一旦安装完成，NFS 服务将会自动启动。默认情况下，在 Ubuntu 18.04 及以上版本中 NFS 版本 2 被禁用。NFS 3 和 NFS 4 都可以用。你可以运行下面的命令去验证：

```
$ sudo cat /proc/fs/nfsd/versions  
-2 +3 +4 +4.1 +4.2
```

NFSv2 非常古老，没有理由去启用它。NFS 服务器配置选项在 `/etc/default/nfs-kernel-server` 和 `/etc/default/nfs-common` 文件。默认的设置对于我们的使用场景已经足够了。

2. 创建文件系统

当配置 NFSv4 服务器的时候，最好的实践就是使用一个全局的 NFS 根目录，并且在这里挂载实际的目录。在这个例子中，我们将会使用 `/data` 作为 NFS root 目录。

我们将会分享两个目录 (`/var/www` 和 `/opt/backups`)，使用不同的配置，来更好的解释如何配置 NFS 挂载。

`/var/www/` 归属于用户和用户组 `www-data`，并且 `/opt/backups` 归属于 `root`。

使用 `mkdir` 命令创建导出文件系统：

```
$ sudo mkdir -p /data/{www,backups}
```

挂载实际的目录:

```
$ sudo mount --bind /opt/backups /data/backups
```

```
$ sudo mount --bind /var/www /data/www
```

想要这个挂载持久化, 开机自动挂载, 修改 `/etc/fstab` 文件

```
$ sudo vi /etc/fstab
```

```
/opt/backups /data/backups none bind 0 0  
/var/www /data/www none bind 0 0
```

3. 导出文件系统

下一步就是定位将要被 NFS 服务器导出的文件系统, 共享选项和被允许访问文件系统的客户端。想要这么做, 打开 `/etc/exports`

```
$ sudo nano /etc/exports
```

`/etc/exports` 文件包含了注释, 解释如何导出一个目录。

```
/data/ 10.8.30.0/24(rw,sync,no_subtree_check,crossmnt,fsid=0)  
/data/backups 10.8.30.0/24(ro,sync,no_subtree_check)  
10.8.30.150(rw,sync,no_subtree_check)  
/data/www 10.8.30.150(rw,sync,no_subtree_check)
```

第一行包含 `fsid=0` 定义了 NFS 根目录 `/data/` 来自 `10.8.30.0/24` 网络的所有客户端被允许访问 NFS 卷。 `crossmnt` 选项是必要的, 用来分享被导出目录的子目录。

第二行显示了如何针对一个文件系统指定多个导出规则。它导出了 `/data/backups` 目录, 并且允许来自 `10.8.30.0/24` 的客户端只读访问, 而来自 `10.8.30.150` 的客户端同时读写可访问。这个 `sync` 选项告诉了 NFS 在回复之前将修改写入磁盘。

最后一行应该是自解释的。想要了解更多可用选项，在终端输入 `man exports`

保存文件并且导出分享：

```
$ sudo exportfs -ra
```

每次你修改 `/etc/exports` 文件你都需要运行一次上面的命令。如果有任何的错误或者警告，它们会被显示在终端上。

想要查看当前活跃的导出和它们的状态，使用：

```
$ sudo exportfs -v
```

输出将会包含所有分享以及它们的选项。就像你能看到的，还有我们没有在 `/etc/exports` 文件定义的选项。那些是默认选项，如果你想修改他们，你需要显式的设置那些选项。

```
/data/jenkins/jenkins_home
10.8.30.0/24(rw,wdelay,root_squash,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
/data/jenkins/versions
10.8.30.0/24(rw,wdelay,root_squash,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
/data
10.8.30.0/24(rw,wdelay,no_root_squash,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)
```

在 Ubuntu 系统上，`root_squash` 默认被启用。它是一个最重要的选项，关系到 NFS 安全性。它阻止来自客户端 root 用户拥有被挂载分享目录的 root 权限。它将会将 root `UID` 和 `GID` 映射到 `nobody/nogroup` 的 `UID` 和 `GID`。

对于那些通过客户端机器访问的用户，NFS 预期会把客户端的用户和用户组 ID 匹配服务器上的用户和用户组。另外一个选项，就是使用 NFSv4 idmapping 特性，它能将用户和用户组 ID 转换成名字或者其他的方式。

就这些。此时，你已经在你的 Ubuntu 服务器上建立了一个 NFS 服务器。你可以看下一步，并且配置客户端，以便连接 NFS 服务器。

4. 防火墙配置

如果你在网络上运行了防火墙，你将需要添加一个规则，允许流量通过 NFS 端口。

假设你使用 `UFW` 管理你的防火墙，你需要运行下面的命令，允许来自 `10.8.30.0/24` 的访问

```
$ sudo ufw allow from 10.8.30.0/24 to any port nfs
```

想要验证修改，运行：

```
$ sudo ufw status
```

输出显示，流量允许从 2049 通过：

To	Action	From
--	-----	----
2049	ALLOW	10.8.30.0/24
22/tcp	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)

NFS 客户端

以上完成了 NFS 服务端配置，下一步就是配置客户端。

1. 安装 NFS 客户端

在客户端机器上需要安装，远程挂载 NFS 文件系统的工具软件

```
# 在 Ubuntu/Debian 上安装 `nfs-common`  
$ sudo apt update  
$ sudo apt install nfs-common  
  
# 在 centos 上安装 `nfs-utils`  
$ sudo yum install nfs-utils
```

2. 挂载文件系统

我们将在 IP 为 10.8.30.150 的客户端机器上操作。这台机器拥有对 /data/www 的读写操作权限，和对 /data/backups 文件的只读访问权限。

创建两个新目录作为挂载点。你可以在任何位置创建这些目录：

```
$ sudo mkdir -p /{backups,www}
```

使用 mount 命令挂载文件系统

```
$ sudo mount -t nfs -o vers=4 10.8.30.152:/data/backups /backups
```

```
$ sudo mount -t nfs -o vers=4 10.8.30.152:/data/www /www
```

`10.8.30.152` 是 NFS 服务器的 IP, 你也可使用主机名

查看 文件系统是否成功挂载, 使用 `df` 命令:

```
$ df -h
```

```
Filesystem                Size      Used Avail Use% Mounted on
udev                      16G         0    16G   0% /dev
tmpfs                     3.2G   4.9M    3.2G   1% /run
/dev/sdd2                 915G    60G   809G   7% /
tmpfs                     16G         0    16G   0% /dev/shm
tmpfs                     5.0M         0   5.0M   0% /run/lock
tmpfs                     16G         0    16G   0% /sys/fs/cgroup
/dev/mapper/lvm_data-lvmda 1.8T    77M   1.7T   1% /data
/dev/sdd1                  511M    6.7M   505M   2% /boot/efi
/dev/sda                   916G    6.8G   863G   1% /home/fs-test
tmpfs                     3.2G         0   3.2G   0% /run/user/1000
/dev/loop1                 99M    99M     0 100% /snap/core/11081
/dev/loop0                 100M   100M     0 100% /snap/core/11167
10.8.30.152:/data/jenkins/jenkins_home 1.8T   198G   1.6T  12%
/home/fastest/jenkins_home
```

想要持久化这些挂载, 打开 `/etc/fstab` 文件:

添加

```
10.8.30.152:/backups /backups nfs defaults,timeo=900,retrans=5,_netdev 0
0
10.8.30.152:/www /www nfs defaults,timeo=900,retrans=5,_netdev 0
0
```

另外一个挂载远程文件系统的选项是: `autofs`.

3. 测试 nfs 访问

让我们通过在共享目录中创建新文件来测试对共享文件夹的访问。

首先, 通过使用 `touch` 命令在 `/data/backups` 目录下创建一个测试文件:

```
sudo touch /backups/test.txt
```

`backup` 文件系统被导出为只读，并且你应该会看到一个类似 `Permission denied` 的错误信息：

```
touch: cannot touch '/backups/test': Permission denied
```

下一步，通过 `sudo` 命令以 `root` 用户身份在 `/data/www` 目录下创建一个测试文件：

再一次，你将会看到 `Permission denied` 信息。

```
touch: cannot touch '/srv/www': Permission denied
```

4. 取消挂载

```
sudo umount /backups
```