## 什么是NFS服务，如何安装和配置？

## 1、NFS 介绍

NFS 是 Network FileSystem 的缩写，顾名思义就是网络文件存储系统，它最早是由 Sun 公司发展出来的，也是 FreeBSD 支持的文件系统中的一个，它允许网络中的计算机之间通过 TCP/IP 网络共享资源。通过 NFS，我们本地 NFS 的客户端应用可以透明地读写位于服务端 NFS 服务器上的文件，就像访问本地文件一样方便。简单的理解，NFS 就是可以透过网络，让不同的主机、不同的操作系统可以共享存储的服务。

NFS 在文件传送或信息传送过程中依赖于 RPC（Remote Procedure Call） 协议，即远程过程调用， NFS 的各项功能都必须要向 RPC 来注册，如此一来 RPC 才能了解 NFS 这个服务的各项功能 Port、PID、NFS 在服务器所监听的 IP 等，而客户端才能够透过 RPC 的询问找到正确对应的端口，所以，NFS 必须要有 RPC 存在时才能成功的提供服务，简单的理解二者关系：NFS是 一个文件存储系统，而 RPC 是负责信息的传输。

## 2、环境、软件准备

本次演示环境，我是在虚拟机上安装 Linux 系统来执行操作，以下是安装的软件及版本：

* **Oracle VirtualBox**: 5.1.20 r114628 (Qt5.6.2)
* **System**: CentOS Linux release 7.3.1611 (Core)
* **rpcbind**: 0.2.0-38.el7.x86\_64
* **nfs-utils**: 1.3.0-0.54.el7.x86\_64

## 3、NFS 服务安装

通过上边简要的介绍，我们知道 NFS 服务需要依赖 RPC 服务，所以这里 NFS 服务端需要安装 rpcbind 和 nfs-utils，客户端只需要安装 nfs-utils 即可，由于我选用的为 CentOS 系统，所以可以使用 yum 快速的安装。

首先，确认下服务端系统是否已安装 NFS。

$ rpm -qa nfs-utils rpcbindnfs-utils-1.3.0-0.54.el7.x86\_64rpcbind-0.2.0-38.el7.x86\_64

注意：这里我已经安装完毕，若为空，则说明未安装。

然后，安装 NFS 服务

# 服务端$ yum install -y nfs-utils rpcbind# 客户端$ yum install -y nfs-utils

另：Ubuntu 16.04 安装命令为：

# 服务端apt install nfs-kernel-server# 客户端apt install nfs-common

## 4、NFS 配置及使用

我们在服务端创建一个共享目录 /data/share ，作为客户端挂载的远端入口，然后设置权限。

$ mkdir -p /data/share$ chmod 666 /data/share

然后，修改 NFS 配置文件 /etc/exports

$ vim /etc/exports/data/share 10.222.77.0/24(rw,sync,insecure,no\_subtree\_check,no\_root\_squash)

说明一下，这里配置后边有很多参数，每个参数有不同的含义，具体可以参考下边。此处，我配置了将 /data/share 文件目录设置为允许 IP 为该 10.222.77.0/24 区间的客户端挂载，当然，如果客户端 IP 不在该区间也想要挂载的话，可以设置 IP 区间更大或者设置为 \* 即允许所有客户端挂载，例如：/home \*(ro,sync,insecure,no\_root\_squash) 设置 /home 目录允许所有客户端只读挂载。

| **参数** | **说明** |
| --- | --- |
| ro | 只读访问 |
| rw | 读写访问 |
| sync | 所有数据在请求时写入共享 |
| async | nfs 在写入数据前可以响应请求 |
| secure | nfs 通过 1024 以下的安全 TCP/IP 端口发送 |
| insecure | nfs 通过 1024 以上的端口发送 |
| wdelay | 如果多个用户要写入 nfs 目录，则归组写入（默认） |
| no\_wdelay | 如果多个用户要写入 nfs 目录，则立即写入，当使用 async 时，无需此设置 |
| hide | 在 nfs 共享目录中不共享其子目录 |
| no\_hide | 共享 nfs 目录的子目录 |
| subtree\_check | 如果共享 /usr/bin 之类的子目录时，强制 nfs 检查父目录的权限（默认） |
| no\_subtree\_check | 不检查父目录权限 |
| all\_squash | 共享文件的 UID 和 GID 映射匿名用户 anonymous，适合公用目录 |
| no\_all\_squash | 保留共享文件的 UID 和 GID（默认） |
| root\_squash | root 用户的所有请求映射成如 anonymous 用户一样的权限（默认） |
| no\_root\_squash | root 用户具有根目录的完全管理访问权限 |
| anonuid=xxx | 指定 nfs 服务器 /etc/passwd 文件中匿名用户的 UID |
| anongid=xxx | 指定 nfs 服务器 /etc/passwd 文件中匿名用户的 GID |

接下来，我们先启动 RPC 服务。

$ service rpcbind start# 或者使用如下命令亦可$ /bin/systemctl start rpcbind.service# 查看 NFS 服务项 rpc 服务器注册的端口列表$ rpcinfo -p localhost program vers proto port service 100000 4 tcp 111 portmapper 100000 3 tcp 111 portmapper 100000 2 tcp 111 portmapper 100000 4 udp 111 portmapper 100000 3 udp 111 portmapper

注意：此时我们还没有启动 NFS 服务，只监听了 111 端口，接着我们来启动 NFS 服务，再来看下注册的端口列表。

# 启动 NFS 服务$ service nfs start# 或者使用如下命令亦可/bin/systemctl start nfs.service# 启动 NFS 服务后 rpc 服务已经启用了对 NFS 的端口映射列表# rpcinfo -p localhost program vers proto port service 100000 4 tcp 111 portmapper 100000 3 tcp 111 portmapper 100000 2 tcp 111 portmapper 100000 4 udp 111 portmapper 100000 3 udp 111 portmapper 100000 2 udp 111 portmapper 100024 1 udp 33745 status 100024 1 tcp 36980 status 100005 1 udp 20048 mountd 100005 1 tcp 20048 mountd 100005 2 udp 20048 mountd 100005 2 tcp 20048 mountd 100005 3 udp 20048 mountd 100005 3 tcp 20048 mountd 100003 3 tcp 2049 nfs 100003 4 tcp 2049 nfs 100227 3 tcp 2049 nfs\_acl 100003 3 udp 2049 nfs 100003 4 udp 2049 nfs 100227 3 udp 2049 nfs\_acl 100021 1 udp 38960 nlockmgr 100021 3 udp 38960 nlockmgr 100021 4 udp 38960 nlockmgr 100021 1 tcp 38362 nlockmgr 100021 3 tcp 38362 nlockmgr 100021 4 tcp 38362 nlockmgr

我们发现，启动了 NFS 服务后，rpc 注册的端口列表明显增多。OK 现在服务端都启动起来了，在服务端看下是否正确加载了设置的 /etc/exports 配置。

$ showmount -e localhostExport list for localhost:/data/share 10.222.77.0/24

## 5、NFS 测试

最后，在另一台 Linux 虚拟机上测试一下，是否能够正确挂载吧。首先，我们可以在客户端查看下 NFS 服务端 (上边服务端 IP 为：10.222.77.86) 设置可共享的目录信息。

# showmount -e 10.222.77.86Export list for 10.222.77.86:/data/share 10.222.77.0/24

然后，在客户端创建挂在目录 /share

$ mkdir -p /share

最后，挂载远端目录到本地 /share 目录。

$ mount 10.222.77.86:/data/share /share$ df -h | grep 10.222.77.86Filesystem Size Used Avail Use% Mounted on10.222.77.86:/data/share 27G 11G 17G 40% /share

可以看到，可以正确将远端 NFS 目录挂载到本地。注意：挂载点 /share 目录必须已经存在，而且目录中没有文件或子目录。

最后，我们在 NFS 服务端 /data/share 目录下创建一个文件，看下客户端是否能够正确读取并修改。

# 服务端写入$ echo "This is NFS server." > /data/share/nfs.txt# ll /data/share/total 4-rw-r--r-- 1 root root 20 Nov 5 16:49 nfs.txt# 客户端读取$ ll /share/total 4-rw-r--r-- 1 root root 20 Nov 5 16:49 nfs.txt$ cat /share/nfs.txtThis is NFS server.# 客户端写入$ echo "This is NFS client." >> /share/nfs.txt# 服务端读取$ cat /data/share/nfs.txtThis is NFS server.This is NFS client.

都是木有问题的，这是因为上边设置了 NFS 远端目录权限为 rw 拥有读写权限，如果设置为 ro，那么客户端只能读取，不能写入，根据实际应用场景合理配置，这里就不在演示了。这里提一下，NFS 默认使用用 UDP 协议来进行挂载，为了提高 NFS 的稳定性，可以使用 TCP 协议挂载，那么客户端挂载命令可使用如下命令：

$ mount 10.222.77.86:/data/share /share -o proto=tcp -o nolock

最后，如果客户端要卸载 NFS 挂载的话，使用如下命令即可。

$ umount /share