



链滴









Linux 通过 NFS 实现文件共享

作者: [ymxfi](#)

原文链接: <https://ld246.com/article/1602396608594>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

在项目生产环境我们经常需要实现文件共享，传统的常见方案是通过 NFS，实现服务之间共享某一块磁盘，通过网络传输将分散的文件集中存储在一块指定的共享磁盘，实现基本的文件共享。实现这种方案，分服务端和客户端，将服务端的磁盘 mount 到客户端指定目录下，在客户端操作就像操作本地磁盘一样，nfs 基本的安装如下：

模拟环境：

服务端/客户端	服务器 IP
服务端	192.168.3.10
客户端	192.168.3.9



















环境要求：

- 需要安装 `nfs` 和 `rpcbind`

1. 服务端安装 NFS

1.1 检查 `nfs` 和 `rpcbind` 是否安装

```
$ rpm -qa |grep nfs  
$ rpm -qa |grep rpcbind
```

1.2 创建一个共享目录，并配置共享信息

```
# 创建共享目录，这里假设共享目录为/data01/nfs  
$ mkdir -p /data01/nfs  
# 配置共享信息  
#在/etc/exports添加以下内容  
$ echo '/data01/nfs 192.168.3.9(rw,no_root_squash,no_all_squash,sync)' >> /etc/exports
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">权限说明:
</span></span><span class="highlight-line"><span class="highlight-cl">    ro
    只读访问
</span></span><span class="highlight-line"><span class="highlight-cl">    rw
    读写访问
</span></span><span class="highlight-line"><span class="highlight-cl">    sync
    所有数据在请求时写入共享
</span></span><span class="highlight-line"><span class="highlight-cl">    async
    NFS在写入数据前可以相应请求
</span></span><span class="highlight-line"><span class="highlight-cl">    secure
    NFS通过1024以下的安全TCP/IP端口发送
</span></span><span class="highlight-line"><span class="highlight-cl">    insecure
    NFS通过1024以上的端口发送
</span></span><span class="highlight-line"><span class="highlight-cl">    wdelay
    如果多个用户要写入NFS目录, 则归组写入 (默认)
</span></span><span class="highlight-line"><span class="highlight-cl">    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。
</span></span><span class="highlight-line"><span class="highlight-cl">    hide
    在NFS共享目录中不共享其子目录
</span></span><span class="highlight-line"><span class="highlight-cl">    no_hide
    共享NFS目录的子目录
</span></span><span class="highlight-line"><span class="highlight-cl">    subtree_chec
    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)
</span></span><span class="highlight-line"><span class="highlight-cl">    no_subtree_c
    和上面相对, 不检查父目录权限
</span></span><span class="highlight-line"><span class="highlight-cl">    all_squash
    共享文件的UID和GID映射匿名用户anonymous, 适合公用目录。
</span></span><span class="highlight-line"><span class="highlight-cl">    no_all_squash
    保留共享文件的UID和GID (默认)
</span></span><span class="highlight-line"><span class="highlight-cl">    root_squash
    root用户的所有请求映射成如anonymous用户一样的权限 (默认)
</span></span><span class="highlight-line"><span class="highlight-cl">    no_root_squa
    root用户具有根目录的完全管理访问权限
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-nv">anonuid</span><span class="highlight-o">=</span><span>xxx
    指定NFS服
    器/etc/passwd文件中匿名用户的UID
</span></span><span class="highlight-line"><span class="highlight-cl">    <span class="highlight-nv">anongid</span><span class="highlight-o">=</span><span>xxx
    指定NFS服
    器/etc/passwd文件中匿名用户的GID
</span></span></code></pre>

```

1.3 启动 `nfs` 和 `rpcbind`

```

<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ systemctl <span class="highlight-nb">enable</span> rpcbind
</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl start r
cbind
</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl <span class="highlight-nb">enable</span> nfs
</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl start n
s
</span></span></code></pre>

```

1.4 查看是否共享成功

```

<pre><code class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ showmount -e localhost
</span></span></code></pre>

```

2. 客户端安装并挂载

2.1 检查 `nfs` 和 `rpcbind` 是否安装

```
$ rpm -qa |grep nfs
$ rpm -qa |grep rpcbind
```

2.2 启动 `nfs` 和 `rpcbind`

```
$ systemctl enable rpcbind
$ systemctl start rpcbind
$ systemctl enable nfs
$ systemctl start nfs
```

2.3 创建挂载目录

```
$ mkdir -p /data01/share
```

2.4 检测远程共享服务器是否共享成功

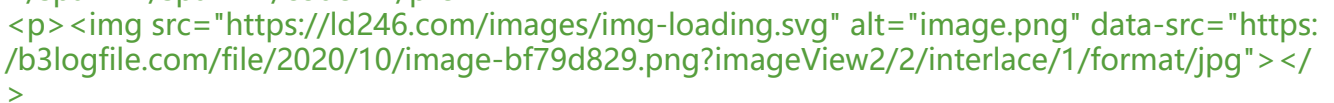
```
$ showmount -e 192.168.3.11
```

2.5 挂载

```
# mount -t nfs -o nolock 服务端IP:共享目录绝对路径 本地挂载目录
$ mount -t nfs -o nolock 192.168.3.11:/data01/nfs /data01/share
```

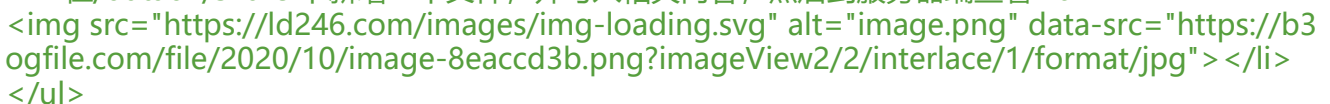
2.6 查看挂载结果

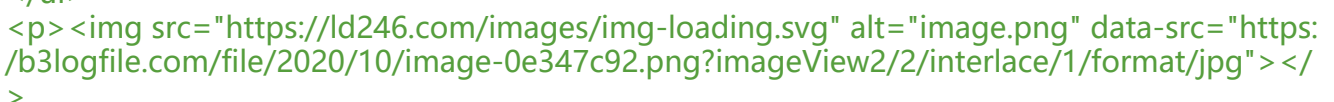
```
$ df -Th
```

如果显示了远程磁盘的挂载信息，则挂载成功

2.7 测试

在/data01/share 下新增一个文件，并写入相关内容，然后到服务器端查看





2.8 卸载

卸载直接在客户端服务器上执行如下操作即可：<code>umount 192.168.3.11:/data01/nfs</code>

2.9 防火墙规则设置

可以通过 `rpcinfo -p` 查看具体用到的端口。
 NFS 启动时会随机启动多个端口并向 RPC 注册，为了设置安全组以及 iptables 规则需要设置 NFS 固定端口。

NFS 服务需要开启 `mountd,nfs,nlockmgr,portmapper,rquotad` 这 5 个服务，其中 `nfs`、`ortmapper` 的端口是固定的，另外三个服务的端口是随机分配的，所以需要给 `mountd,nlockmgr,rqotad` 设置固定的端口。

-

其中，给 `mountd`、`rquotad` 设置端口的方式很简单，在 `/etc/sysconfig/nf` 中添加一下设置即可：

```
class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nv">RQUOTAD_PORT</span><span class="highlight-o">=</span><span class="highlight-m">30001</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nv">LOCKD_TCPPORT</span><span class="highlight-o">=</span><span class="highlight-m">30002</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nv">LOCKD_UDPPORT</span><span class="highlight-o">=</span><span class="highlight-m">30002</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nv">MOUNTD_PORT</span><span class="highlight-o">=</span><span class="highlight-m">30003</span></span></span><span class="highlight-line"><span class="highlight-cl"><span class="highlight-nv">STATD_PORT</span><span class="highlight-o">=</span><span class="highlight-m">30004</span></span></span></pre>
```

添加完后重启 `rpc`、`nfs` 的配置与服务

```
class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ systemctl restart rpcbind</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl restart nfs</span></span></pre>
```

-

还需在 `/etc/modprobe.d/lockd.conf` 中添加以下设置：

```
class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">options lockd <span class="highlight-nv">nlm_tcpport</span><span class="highlight-o">=</span><span class="highlight-m">30002</span></span></span><span class="highlight-line"><span class="highlight-cl">options lockd <span class="highlight-nv">nlm_udpport</span><span class="highlight-o">=</span><span class="highlight-m">30002</span></span></span></pre>
```

添加完后重新加载 NFS 配置和服务

```
class="language-bash highlight-chroma"><span class="highlight-line"><span class="highlight-cl">$ systemctl restart nfs-config</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl restart nfs-idmap</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl restart nfs-lock</span></span><span class="highlight-line"><span class="highlight-cl">$ systemctl restart nfs-server</span></span></pre>
```

```
</span> </span> </code> </pre>
```

```
</li>
```

```
<li>
```

```
<p> <strong>修改完后查看端口</strong> <br>
```

```
 </p>
```

```
</li>
```

```
</ul>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th>端口</th>
```

```
<th>proto</th>
```

```
<th>作用</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>111</td>
```

```
<td>tcp</td>
```

```
<td>portmapper 进程使用，负责分配 rpc server 的端口，并在 client 端请求时，负责响应目的 rpc server 端口返回给 client 端。 </td>
```

```
</tr>
```

```
<tr>
```

```
<td>111</td>
```

```
<td>udp</td>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<td>30002</td>
```

```
<td>udp</td>
```

```
<td>NFS lock manager 进程使用，它允许 NFS 客户端在服务器上对文件加锁。 </td>
```

```
</tr>
```

```
<tr>
```

```
<td>30002</td>
```

```
<td>tcp</td>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<td>2049</td>
```

```
<td>tcp</td>
```

```
<td>nfs 的守护进程，负责接收到用户的调用请求后与内核发出请求并得到调用结果响应给用户。 </td>
```

```
</tr>
```

```
<tr>
```

```
<td>2049</td>
```

```
<td>udp</td>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<td>30003</td>
```

```
<td>udp</td>
```

```
<td>是 nfs 服务的认证服务的守护进程，client 在收到返回的真正端口时，就会去连接 mountd，证取得令牌。 </td>
```

```
</tr>
<tr>
<td>30004</td>
<td>tcp</td>
<td></td>
</tr>
<tr>
<td>30004</td>
<td>udp</td>
<td>rpc.statd 进程，这个进程实现了网络状态监控(NSM)RPC 协议，通知 NFS 客户端什么时候一个
NFS 服务器非正常重启动。 </td>
</tr>
</tbody>
</table>
```