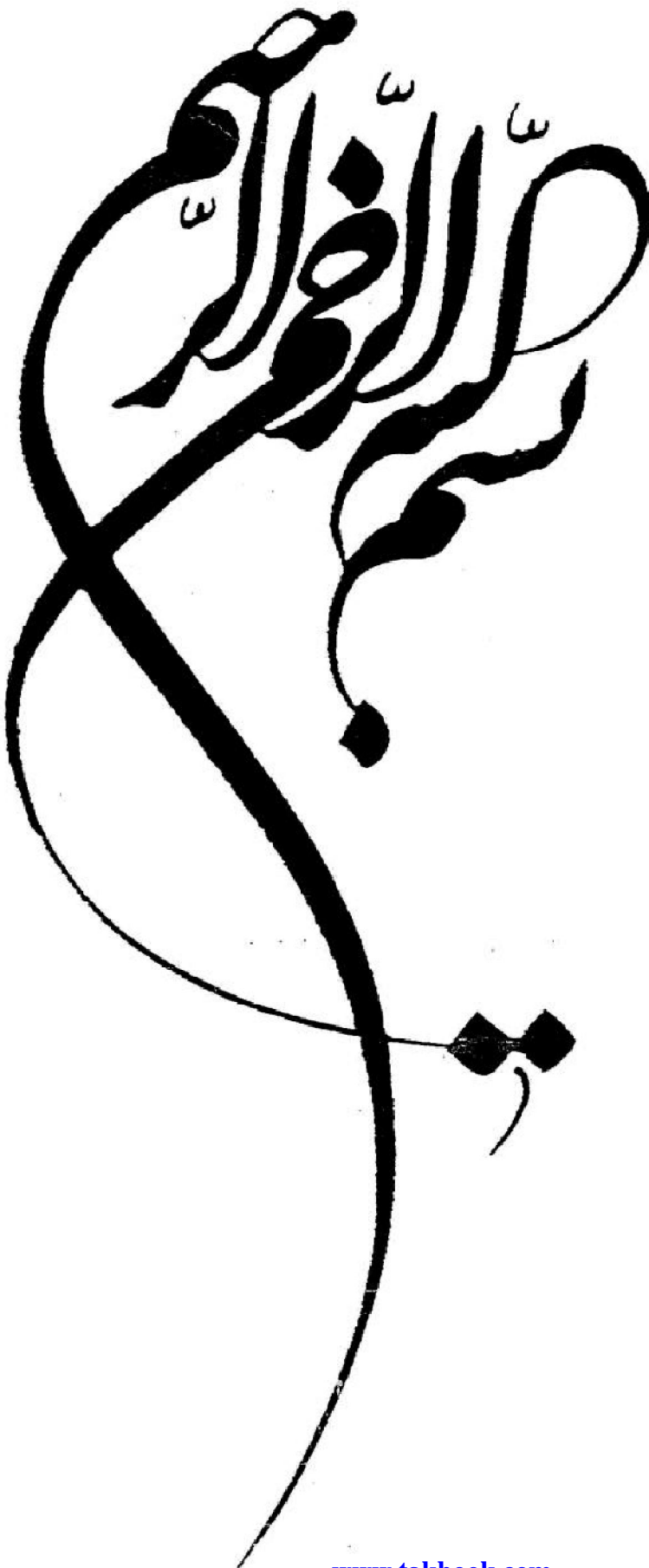


NFS در لینوکس

نویسنده: حسام الدین توحید

SKYWAN13@YAHOO.COM



مقدمه مولف :

آنچه پیش رو دارید به صورت رایگان و تحت لیسانس GNU GPLv3 به علاقه مندان لینوکس هدیه می گردد . در تهیه این مقاله از سر فصل های درسی گفته شده در دوره های LPic2 و RHCE استفاده شده و لازم می دانم از مهندس مهدوی فر به خاطر راهنمایی های مفیدشان و مرکز آموزشهای پیشرفته دانشگاه شریف (لایتك) تشکر کافی را داشته باشم. این مطالب با نگاهی کاربردی و بدون پرداختن به بحث های تئوریک گردآوری و عرضه شده است. امیدوارم مطالب ارائه شده بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. این آموزش بر اساس توزیع CentOS می باشد، لذا خواهشمندم هرگونه نقص در محتوا را به ایمیل نگارنده ارسال فرمائید. انتشار این فایل با ذکر منبع بلامانع است و هرگونه استفاده نامناسب از محتوای ارائه شده بر عهده کاربر بوده و تمام حقوق معنوی این اثر به نویسنده آن تعلق دارد. زکات علم نشر آن است.

موفق باشید

حسام الدین توحید

خرداد 1393

فهرست :

اشتراک سیستم فایل در لینوکس

4	مقدمه ای بر NFS
6	مزایای استفاده از NFS
6	ویژگی های NFS.v4
7	RPC های مورد استفاده در NFS
9	نصب و راه اندازی NFS
10	Export دایرکتوری در NFS
13	Export دایرکتوری به سبک NFSv4
14	Export دایرکتوری Home
14	اختصاص پورت های ثابت به NFS
16	دستورات مهم برای NFS
17	استفاده از دایرکتوری Export شده در کلاینتها
18	مبحث Autofs

اشتراک سیستم فایل در لینوکس

مقدمه ای بر NFS :

NFS که مخفف Network File System است بطور خلاصه امکانیست که اجازه می دهد تا یک سیستم فایل محلی قابل mount شدن توسط دیگر سیستم ها در شبکه مورد استفاده قرار بگیرد. NFS یک روش سنتی برای share کردن دایرکتوری بین سیستم های Unix Base می باشد که به وسیله ی Sun Microsystems در سال ۱۹۸۰ توسعه و به طور پیش فرض بر روی سیستم های Redhat Base نصب است و در توزیع CentOS از ورژن 3 آن استفاده می شود. ورژن 4 از سری 6 لینوکس های Redhat Base به آنها اضافه شد. با کمک NFS ، ما می توانیم اشتراک فایل بین سیستم یونیکس به لینوکس و لینوکس به یونیکس را راه اندازی کنیم. تا پیش از NFSv4 تمامی نسخه های NFS بصورت Stateless بودند، به این معنی که در پروتکل های Stateless ، هر درخواست هیچ اطلاعاتی از درخواست های پیشین ندارد و هر درخواست مستقل از درخواست های قبلی می باشد. انواع متفاوتی از سیستم فایل ها قابل دسترسی هستند که این کار با استفاده از یک سری API انجام می گیرد ، به این api ها Virtual File System گفته می شود.

هدف اصلی از VFS ، اجازه دادن به برنامه های کاربردی کلاینت برای دسترسی به انواع سیستم فایل ها در یک شکل واحد بدون آگاهی از این تفاوت هاست ، که در تمامی سیستم های عامل مانند ویندوز، لینوکس، مکینتاش و تمامی شبه یونیکس ها استفاده می شود. یک سیستم فایل NFS که بر روی ماشینی Mount شده بسیار شبیه به سیستم فایل محلی همان ماشین است .

هدف از VFS دسترسی به سیستم فایل های مختلف بر روی ماشین است. تمام اعمال روی سیستم فایل ماشین محلی که یک سیستم فایل راه دور بر روی آن Mount شده است، از طریق VFS انجام می گیرد.

در ارائه سرویس NFS هیچ محدودیتی در نوع کاربرد سیستم نیست یعنی در یک شرکت با 100 سیستم نباید حتما یکی از آنها بعنوان سروردهنده NFS باشد و مابقی از آن سرویس بگیرند بلکه هر کدام از این سیستم ها می توانند نقش سرویس دهنده NFS را داشته باشند و دایرکتوری ها و فایل های خود را به اشتراک بگذارند.

اما می توان از دیدگاهی دیگر راه اندازی آنرا به دو بخش تنظیم سرویس دهنده و تنظیم سرویس گیرنده تقسیم کرد چون بالاخره یکی دایرکتوری را به اشتراک می گذارد و بقیه استفاده می کنند پس شاید بنا به سیاست یک سیستم هم دایرکتوری را برای دیگران به اشتراک بگذارد و هم از اشتراکات دیگران استفاده کند. تنها مطلبی که

باید دقت شود اطمینان از فعال بودن سرویس های لازم در هر دو سمت سروس دهنده و سرویس گیرنده بود که برای کارکرد صحیح سرویس NFS بسیار لازم و ضروری هستند.

تمامی نسخه های NFS از پروتکل TCP استفاده می کنند. NFSv2 امروزه کمتر استفاده می شود اما NFSv3 و NFSv4 بسیار کاربرد دارند. NFSv4 بواسطه ثابت بودن پورت و تنظیمات فایروال می تواند بر روی اینترنت کار کند و همچنین از ویژگی ACL که نیز پشتیبانی می کند. NFSv2 و NFSv3 از پروتکل UDP نیز می توانند استفاده کنند. مشکل اصلی ورژن های 2 و 3 و 1 رندوم بودن پورت های ارتباطی آنها می باشد چون نمی توانیم پورت های مربوطه را بر روی فایروال باز کنیم، که این مشکل در ورژن 4 برطرف گردیده است. فایل `/etc/exports` بخش اصلی تنظیم NFS است که فهرستی از دایرکتوری های اشتراکی را درون خود نگه می دارد. در این فایل تعیین می شود کدام دایرکتوری ها با چه مجوز هایی باید برای چه کسانی و با چه مجوز هایی به اشتراک گذاشته شوند. به صورت پیش فرض این فایل بدون محتوا بوده و باید توسط کاربر مقدار دهی شود. در اوایل سرویس های NFS بصورت Stateless بودن یعنی توانایی ثبت و نگهداری تاریخچه ای از فعالیت ها و اینکه کدام فایل توسط کدام ماشین ها و کاربران راه دور استفاده شده اند را نداشتند. این اطلاعات و تاریخچه ها برای قفل کردن فایل ها لازم می باشند. یک راه حل برای این مشکل بکار بردن ابزار های مجزا از NFS مانند `lockd` و `statd` برای قفل کردن فایل هاست که در NFSv2 و NFSv3 استفاده می شوند. اما NFSv4 ویژگی `statfull` را که نشان دهنده مبتنی بودن بر حالت و نگه داشتن تاریخچه ای از فعالیت ها را دارا می باشد. NFSv4 مبتنی بر state است و هم بر روی کلاینت و هم بر روی سرویس دهنده اطلاعاتی را نگه می دارد مانند اینکه کدام فایل ها باز هستند و یا کدام ها قفل شده اند. این اطلاعات در صورتی که سرویس دهنده دچار مشکل شوند برای رفع اشکال سرویس دهنده میان آنها رد و بدل خواهد شد.

مزایای استفاده از NFS

1. NFS اجازه‌ی دسترسی محلی به فایل‌های از راه دور را می‌دهد.
2. NFS از معماری سرویس گیرنده / سرویس دهنده استاندارد برای به اشتراک گذاری فایل بین همه‌ی ماشین‌های مبتنی بر nix*، استفاده می‌کند.
3. با NFS، هیچ نیازی نیست که روی هر دو ماشین، سیستم عامل مشابه اجرا شده باشد.
4. با کمک NFS ما می‌توانیم راه حل‌های ذخیره‌سازی مرکزی را پیکربندی کنیم.
5. کاربران می‌توانند اطلاعات خود را بدون توجه به موقعیت فیزیکی، دریافت کنند.
6. هیچ refresh دستی برای فایل‌های جدید نیاز نمی‌شود.
7. نسخه جدیدتر NFS همچنین ACL و مانت ریشه‌ی کاذب را پشتیبانی می‌کند.
8. می‌توان با Firewall ها و Kerberos آن را امن تر کرد.

ویژگی های NFS.v4

1. سازگاری با فایروال ها و ابزار های NAT
2. امنیت بالا
3. بر طرف کردن مشکلات Authentication
4. پشتیبانی از کلاینت های یونیکسی (Linux,BSD,Mac OSX) و ویندوزی
5. پشتیبانی از نوشتن ACL
6. پشتیبانی از نام فایل های یونیکد
7. کارایی بالا حتی بر روی شبکه ها با پهنای باند پایین
8. بر مبنای tcp کار کرده و پورت ثابت آن 2049 می باشد و البته نیاز به نرم افزارهای کمکی مثل Portmapper را ندارد.
9. کلاینت های ویندوزی می توانند از share های nfs استفاده کنند.
10. NFSv4 بر روی شبکه های wan هم کار میکند.
11. مبحث RootDirectory به آن اضافه شده است.

تا ورژن 3 ، سرویس NFS از پورت استاتیک استفاده نمی کرد. نرم افزارهای دیگری مثل Portmapper پیشنهاد دهنده پورت به کلاینت بودند. Portmapper چک می کند NFS با چه پورته کار میکند ، به محض اینکه درخواستی از یک کلاینت برسد می آید پورت مربوطه را به کلاینتها پیشنهاد می دهد.

RHEL و دیگر توزیع های لینوکسی ترکیبی از سرویس ها را برای انجام NFS استفاده می کنند. تمامی نسخه های NFS متکی بر RPC یا Remote Procedure Call میان کلاینت ها و سرور ها هستند. سرویس های RPC در لینوکس تحت سرویس portmap کار می کنند. در زیر فهرستی از سرویس ها که بصورت ترکیبی با هم در حال ارتباط و کار کردن برای اجرای NFS هستند آمده است:

nfs : اصلی ترین سرویس که دیگر سرویس ها با فعال شدن آن نیز فعال می شوند.

nfslockd : در سمت کلاینت اجازه می دهد تا فایل ها را بروی سرور قفل یا lock کند.

portmap : همانطور که گفته شد سرویس های RPC تحت لینوکس توسط این سرویس کنترل می شوند و مسئول تنظیم کردن اتصال ها برای سرویس های RPC درخواست شده است.

RPC های مورد استفاده در NFS

وقتی دو کلاینت لینوکسی قصد دارند به منابع اشتراکی هم متصل شوند از پروتکل NFS استفاده می کنند و خود NFS در پس زمینه از RPC ها برای اتصال استفاده میکند. NFS برای ارتباط نیاز به شش نوع RPC دارد. تمام فایل سیستم هایی که تحت شبکه کار می کنند برای عملکرد صحیح خود از سرویس به نام RPC استفاده می کنند. در واقع RPC پروسه ای است که Computing دستورات بین مبدا و مقصد را انجام می دهد تا کلاینت درگیر پروسه ارتباط نشود. فهرست این rpc ها به همراه توضیح در زیر آمده است:

rpc.mountd : این سرویس در خواست های mount را از سمت کلاینت دریافت کرده و بررسی می کند که دایرکتوری درخواست شده در حال حاضر به اشتراک گذاشته شده یا خیر. این سرویس عمل mount شدن منبع اشتراکی را انجام می دهد. فهرست دایرکتوری ها صادر شده در فایل `etc/exports/` قرار می گیرند . exports شدن به این معنی است که یک دایرکتوری را در شبکه به اشتراک بگذاریم. این سرویس در هنگام فعال کردن سرویس nfs بصورت خودکار فعال می شود.

rpc.nfsd : این سرویس با هسته لینوکس برای مواجه شدن با در خواست های پویا از سمت کلاینت کار می کند.

rpc.nfslock : به کلاینت اجازه می دهد فایل های خود را بر روی سرور قفل کند اگر فایل مورد استفاده lock نشود ممکن است توسط شخص دیگری مورد تغییر قرار بگیرد. این سرویس در NFSv4 استفاده نمی شود.

rpc.statd : کلاینت ها را در صورتی که سرور restart شود باخبر می کند. کار این RPC مانیتور کردن مبداء و مقصد می باشد یعنی اگر هر کدام از طرفین ریوت شوند به طرف مقابل خبر داده می شود تا lock فایل ها برداشته شئد. این سرویس توسط nfslock فعال می شود و در NFSv4 استفاده نمی شود.

rpc.rquotad : این سرویس اطلاعات سهمیه بندی را برای کاربران راه دور (remote users) فراهم می کند. سهمیه بندی یعنی اینکه هر کاربر راه دور چه مقدار فضا می تواند برای ایجاد فایل و ... داشته باشد. این سرویس به همراه فعال کردن سرویس nfs خودکار فعال می شود.

rpc.idmapd : بعنوان نگاشت کننده نام ها و ID ها در NFS است. این سرویس UID ها و GID ها را به نام ها ترجمه می کند. فایل مرتبط با آن /etc/idmapd.conf می باشد.

rpc.gssd : یک پروتکل امنیتی است که در ایجاد ارتباط Security Conetxt های لازم برای کرنل را load می کند . نکته ضروری دیگر اینکه برای استفاده از NFS بر روی سیستم تان مطمئن شوید که بسته های portmap, nfs-utils و nfs-utils-lib نصب شده باشد که در بخش نصب سرویس انها را نصب خواهیم کرد.

بسته سرور NFS شامل سه سرویس گنجانده شده در بسته های portmap و nfs-utils می باشد.

نصب و راه اندازی سرویس NFS

ابتدا باید از نصب بودن پکیج NFS اطمینان حاصل کنیم لذا با دستور زیر از سیستم query می گیریم :

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

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب nfs از yum استفاده می کنیم :

```
#yum -y install nfs-utils
```

بعد از نصب ، باید اطمینان حاصل کنیم که آیا پکیج NFS بر روی سیستم نصب شده است یا خیر لذا با دستور زیر از سیستم query می گیریم :

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

سپس با دستور زیر شاخه ها و مسیرهایی که فایل های این سرویس در آن ایجاد شده است را چک می کنیم :

```
#rpm -ql nfs
```

و با این دستور هم اطلاعات لازم را در مورد پکیج این سرویس به دست می آوریم :

```
#rpm -qi nfs
```

سپس با دستور chkconfig مشخص می کنیم در چه runlevel هایی فعال باشد :

```
#chkconfig --level 35 nfs on
```

و در انتها سرویس را reset می کنیم :

```
#service nfs restart
```

```
#exportfs -rva
```

برای اینکه بفهمیم چه پورتهایی به این سرویس اختصاص داده شده از این دستور بهر می بریم -p

```
#showmount -e
```

```
192.168.10.1
```

نکته : سرویسی که باعث می شود در هر بار ریست سرویس پورت جدیدی به آژن تعلق بگیرد portmapper می باشد که در ورژن 4 هیچ کاربردی ندارد.

توضیح دستور exportfs :

با این دستور می توانیم لیست export های سیستم را مشاهده کنیم.

r : این سوئیچ export های حذف شده را پاک و یک لیست به روز را نشان میدهد.

v : جزئیات را کاملاً نشان میدهد.

a : export های جدید را نشان می دهد.

export دایرکتوری در NFS

توضیحات سمت سرور :

Export کردن به این معنی است که تعیین کنیم کدام یک از دایرکتوری ها بر روی ماشین محلی برای کدام یک از ماشین ها راه دور و با چه مجوز هایی قابل mount شدن باشد. در NFSv2 و NFSv3 هر دایرکتوری صادر شده بعنوان یک ورودی مستقل بود اما در NFSv4 این طور نمی باشد.

NFS سرویسی است که امکان اشتراک گذاری سیستم فایل مابین سیستم عامل های یونیکسی مانند لینوکس ها، بی اس دی ها و مکینتاش را فراهم می کند، حتی امکان استفاده از این سرویس بین سیستم عامل های یونیکسی و ویندوز سرور 2008R2 نیز فراهم شده است.

روند کار برای پیاده سازی سرویس NFS بدین گونه است که ابتدا باید سرویس ها و برنامه های لازم را هم بر روی سرویس دهنده (NFS Server) و هم بر روی (NFS Client) نصب کنیم. سپس تعیین کنیم چه دایرکتوری هایی که باید اشتراکی شوند و در نهایت دایرکتوری های اشتراکی شده در سیستم های کلاینت mount می کنیم مطلب دیگر اینکه لازم است Firewall نیز بصورت مناسب برای اجازه به ترافیک NFS تنظیم شود.

همانطور که گفته شد NFS دارای فایلی به نام exports در زیر دایرکتوری /etc است که باید آنرا ویرایش کنیم.

شکل کلی هر خط این فایل بصورت زیر است:

shared_directory IP or machin_name(OPTIONS)

که در آن shared_directory نام دایرکتوری از NFS Server است که برای دیگر ماشین های شبکه Share شده است و IP or machin_name آدرس یا نام ماشین(های) کلاینتی است که مجاز به mount کردن دایرکتوری در سیستم محلیشان می باشند و OPTIONS هم گزینه هایی هستند که بر شیوه استفاده کلاینت از دایرکتوری اشتراکی تاثیر می گذارند. در زیر تعدادی از مهمترین این Option توضیح داده شده است :

ro : مخفف Read Only است که کلاینت های تنظیم شده با این گزینه، تنها دسترسی فقط خواندنی روی دایرکتوری Mount شده دارند.

rw : مخفف Read Write است که کلاینت های تنظیم شده با این گزینه، دسترسی خواندن و نوشتن بر روی دایرکتوری اشتراکی را دارند.

sync : این گزینه باعث می شود که سرور تنها پس از اینکه نوشتن داده ها (اعمال تغییرات) به طور کامل انجام شد، به کلاینت پاسخ دهد. این گزینه، بعنوان پیشفرض است و اگر ننویسید، همین گزینه در نظر گرفته می شود.

async : نقطه مقابل sync است و یعنی اینکه سرور قبل از تکمیل تغییرات بر روی Storage، به درخواست های دیگر نیز پاسخ خواهد داد. و یوزرهای مقابل می تواند از هر مقدار فایل که کپی شده به صورت Real Time استفاده کند.

نکته : sync قابلیت اطمینان و امنیت بالاتری دارد اما async دارای کارایی و سرعت بالاتری می باشد.

no_root_squash : گزینه ای بسیار مهم در تنظیم یک دایرکتوری برای اشتراک گذاشتن است، چون که باعث کاهش امنیتی در سرور NFS خواهد. اگر از این گزینه استفاده می کنید باید بدانید که کاربر root روی ماشین کلاینت، روی دایرکتوری اشتراک شده، دسترسی root مطابق با root ماشین سرور را خواهد داشت و توصیه می شود که از این گزینه استفاده نکنید.

root_squash : نقطه مقابل no_root_squash است. با استفاده از این گزینه، درخواست های آمده از uid=0 و gid=0 به ک کاربر anonymous که به nobody user یا nfsnobody user شناخته می شود، نگاشت خواهد شد. یعنی دسترسی کاربر root روی ماشین کلاینت بر روی دایرکتوری Share شده، معادل با دسترسی کاربر root روی ماشین سرور **نخواهد بود** که این کار باعث افزایش امنیت خواهد شد. از این به بعد سطح دسترسی فایل مربوطه nfsnobody می باشد. به طور پیش فرض این آپشن فعال است.

no_subtree_check : این گزینه باعث عدم پیمایش در دایرکتوری بالا دستی دایرکتوری اشتراکی می شود. یعنی وقتی یک دایرکتوری اشتراکی می شود، با تنظیم این گزینه نمی گذاریم که کلاینت ها به دایرکتوری های بالایی دایرکتوری اشتراکی شده دسترسی پیدا کنند و تنها به زیر دایرکتوری های، دایرکتوری اشتراکی دسترسی دارند.

نکته : هر یوزری که با nfs به منبعی وصل می شود، یوزر پسورد پرسیده نمی شود به جای آن وقتی وارد یک سیستم می شود با یوزر nfsnobody به آن وصل می شود به این تکنیک squash گفته می شود.

در زیر چندین مثال از فرمت های گوناگون exports شدن یک منبع آورده شده است:

در مثال زیر دایرکتوری /nfs/nfs-share/ برای یک کلاینت با آدرس 192.168.10.1 به اشتراک گذاشته شده و در خط دوم برای دو ماشین به آدرس های 192.168.10.2 و 192.168.10.1 به اشتراک گذاشته شده است.

/nfs-share/ 192.168.10.1(rw, sync)

```
/nfs-share/ 192.168.10.1(rw,sync) 192.168.10.2 (rw,sync)
```

در مثال زیر یک محدوده IP ها (IP Range) در نظر گرفته می شود.

```
/nfs-share/ 192.168.10.0/24(rw,sync)
```

و در این مثال یک ماشین عضو دامنه skywan13.local را شامل می شود.

```
/nfs-share/ pc2.skywan13.local(rw,sync)
```

در مثال زیر تمامی ماشین های عضو دامنه skywan13.local را شامل میشود.

```
/nfs-share/ *.skywan13.local(rw,sync)
```

برای دیدن مثال های بیشتر به exports 5 man رجوع کنید.

Example :

فرض کنید می خواهیم دایرکتوری /nfs-share/ را export کنیم تا کاربران شبکه 192.168.10.0/24 با گزینه های ro و root_squash از آن استفاده کنند.

```
#mkdir /nfs-share
#groupadd nfs-users
#chgrp nfs-users /nfs-share
#chmod g+s /nfs-share
#chmod -R 777 /nfs-share
#vi /etc/exports
    /mnt/nfs-share 192.168.10.10(rw,sync)
#service nfs restart
```

Export دایرکتوری به سبک NFSv4

در ورژن های قبل از NFSv4 باید مسیر کامل دایرکتوری را به در فایل eports و هم در خط mount وارد می کردیم . یعنی حتما باید از ریشه مسیره می شود. به کار rootDirectory گفته می شود. این یک نقص امنیتی است که در ورژن 4 اصلاح گردیده است. جهت بررسی از ورژن 3 و 4 مثالهای آورده شده است .

NFS.v3

```
#mkdir -p /mnt/nfs-tes
#vi /etc/exports
    /mnt/nfs-test *(ro)
#service nfs restart
#exportfs -rva
#mount -t nfs -o vers=4 192.168.10.1:/mnt/nfs-test
```

NFS.v4

```
#mkdir -p /mnt/nfs-test
#vi /etc/exports
    /mnt *(ro,fsid=0)
    /mnt/nfs-test *(rw,nohide)
#service nfs restart

#mount -t nfs -o vers=4 192.168.10.1:/nfs-test
```

یا اینکه به این صورت می نویسیم :

```
#mount -t nfs4 192.168.10.1:/nfs-test
```

توضیح :

Fsid=0 : این گزینه نشان می دهد /mnt با NFS ورژن 4 به اشتراک گذاشته شده است.

Nohide : یعنی محتوا را نشان دهد .

Export دایرکتوری Home

هیچ یوزری دایرکتوری /home واقعی خودش را نباید در اختیار یوزر دیگری قرار بدهد. این کار بهترین روش امنیتی برای share کردن بین دو لینوکس می باشد ، مزیت این کار جلوگیری از rootdirectory است .

```
#mkdir /mnt/home
#mount --bind /home /mnt/home
#vi /etc/exports
    /mnt *(ro,fsid=0)
    /mnt/home *(rw,nohide)
#service nfs restart
```

--bind : این آپشن باعث می شود محتوای home اصلی با /mnt/home یکسان شود .

اختصاص پورت های ثابت به NFS

اگر از NFS زیر ورژن 4 استفاده می کنید می توانید پورتهای آن را ثابت کنید تا دیگر درگیر مشکلات فایروال نشوید. برای این کار به مسیر زیر رفته و فایل nfs را باز میکنیم. فایل nfs اصلی پیکربندی سرویس NFS میباشد. هر زمان که NFS اجرا می شود محتوای این فایل را چک می کند تا اگر تغییری دید آنها را اعمال کند.

```
#vi /etc/sysconfig/nfs
```

پس از اجرای دستور بالا باید خطوط زیر را Uncomment کنید یعنی علامت # ابتدای خطوط زیر را بردارید

```
LOCKD_TCPPORT=32803
LOCKD_UDPSPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

بعد از ذخیره تغییرات ، سرویس را ریست کرده و دوباره از پورت های NFS یک لیست می گیریم :

```
#service nfs restart
#rpcinfo -p
```


این کار برای کسانی مناسب است که می خواهند فایروال سیستم روشن باشد و این پورتهای را درون آن باز کنند و در انتها به منظور پیکربندی iptables خطوط زیر را به فایل /etc/sysconfig/iptables اضافه می کنیم .

#/etc/sysconfig/iptables]

Firewall configuration written by system-config-firewall #

.Manual customization of this file is not recommended #

filter*

```
A INPUT -m state --state NEW -m udp -p udp --dport 2049 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 2049 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 111 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 111 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 32769 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 32803 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 892 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 892 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 875 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 875 -j ACCEPT-
A INPUT -m state --state NEW -m udp -p udp --dport 662 -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 662 -j ACCEPT-
[INPUT ACCEPT [0:0:
[FORWARD ACCEPT [0:0:
[OUTPUT ACCEPT [0:0:
A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT-
A INPUT -p icmp -j ACCEPT-
A INPUT -i lo -j ACCEPT-
A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT-
A INPUT -j REJECT --reject-with icmp-host-prohibited-
A FORWARD -j REJECT --reject-with icmp-host-prohibited-
COMMIT
```

در آخر هم سرویس iptables را ریست می کنیم .

#service iptables restart]

iptables: Flushing firewall rules:

iptables: Setting chains to policy ACCEPT: filter

iptables: Unloading modules:

iptables: Applying firewall rules:

دستورات مهم برای NFS

برخی از دستورات مهم برای NFS

showmount -e :

نمایش share های در دسترس روی ماشین محلی

showmount -e server-ip or hostname :

لیست share های در دسترس در سرور از راه دور (remote)

showmount -d :

لیست تمام زیر دایرکتوری ها (sub directorie)

exportfs -v :

نمایش یک لیست از فایل های share شده و آپشن های روی یک سرور

exportfs -a :

Export همه ی share های لیست شده در /etc/exports، با توجه به نام

exportfs -u :

Unexport همه ی share های لیست شده در /etc/exports، با توجه به نام

exportfs -r :

تازه کردن (Refresh) لیست سرور پس از تغییر /etc/exports

استفاده از دایرکتوری export شده در کلاینت ها

سمت کلاینت :

پس از تمامی کارهای بالا، نوبت به پیکربندی کلاینت ها می رسد. پیش از هر کاری باید در هر کلاینت در مسیر مناسب یک دایرکتوری بسازیم تا دایرکتوری اشتراکی را به آن متصل کنیم. بهترین جا برای اتصال سیستم فایل های اشتراکی دایرکتوری /mnt است.

```
#cd /mnt
#mkdir nfs-share
```

سپس به ازای هر خط فایل exports باید یک دستور mount بصورت زیر اجرا کنیم.

```
#mount -t nfs -o vers=4 192.16.10.1:/mnt/nfs-share
#mount -t nfs4 192.168.10.1:/mnt/nfs-share
```

یا به این صورت:

مشکل دستور mount موقتی بودن آن است یعنی پس از خاموش شدن سیستم، نقطه اتصال دایرکتوری اشتراکی mount شده از بین می رود اگر می خواهید که دایرکتوری مورد نظر بصورت دائمی باشد با یک خط به فایل etc/fstab/ اضافه کنید .

```
#vi /etc/fstab
192.168.10.1:/mnt/nfs-share /mnt nfs defaults 0 0
```

با اضافه کردن این خط به فایل fstab نقطه اتصال ما به دایرکتوری اشتراکی مورد نظر دائمی می باشد. در صورتی که با خطای زیر در هنگام mount کردن یک دایرکتوری در کلاینت مواجه شدید دلیل آن بسته بودن پورت های 111 و 2049 است. بطور کلی برای تست می توانید در سرور و کلاینت فایروال را خاموش یا غیر فعال کنید.

```
mount.nfs: mount to NFS server '192.168.10.1' failed: System Error: No route to host
```

سرویس nfs بر روی پورت 2049 و portmap بر روی پورت 111 کار می کنند.

مبحث Autofs

وقتی یک کلاینت دایرکتوری share شده توسط سرور را mount می کند یعنی یک session و کانال ارتباطی شکل گرفته است ، حال اگر ارتباط سرویس دهنده قطع شود کلاینت متوجه این موضوع نشده و سیستم درگیر چک کردن مداوم آن می شود . این حالت در کار سیستم کلاینت اختلال ایجاد کرده و کلاً آن را از کار می اندازد و به بیج وجه اجازه umount شدن را نمی دهد مگر اینکه سیستم را ریست کنیم .

این که ما همیشه یک دایرکتوری را به صورت ثابت mount شده داشته باشیم یک ضعف به حساب می آید چون اگر سرور از کار بیافتد کلاینتها در کارکرد با مشکل مواجه خواهند شد. برای رفع این ضعف سرویس autofs معرفی شد. لذا هر وقت به دایرکتوری خاصی احتیاج داشتیم autofs آن را برایمان mount کرده و بعد از طی شدن زمان خاصی آن دایرکتوری را umount می کند.

لازم است این سرویس بر روی سیستم کلاینت نصب شود چون کلاینت استفاده کننده از دایرکتوری اشتراکی میباشد.

با نصب این پکیج در زیر دایرکتوری /etc تعدادی فایل ایجاد می شود که با auto شروع می شوند :

auto.ftp
auto.master
auto.misc

auto.master کانفیگ اصلی و گلوبال سرویس auto.master در این فایل قرار دارد. در این فایل آدرس دایرکتوری سیستم local را که قرار است چیزی در آن mount شود را مشخص می کنیم.

#vi /etc/auto.master

/media /etc/auto.media ==timeout=20

توضیح خطوط تعریف شده :

/media : آدرس دایرکتوری از سیستم local که قرار است چیزی درون آن mount شود.

/etc/auto.media : فایل کانفیگ دایرکتوری که قرار است چیزی درون آن mount شود.

==timeout=20 : این خط مشخص میکند بعد چه مدتی دایرکتوری mount شده umount گردد.(بر

حسب ثانیه)

در مرحله دوم باید فایل `auto.media` را ایجاد کنیم. برای این کار از فایل `auto.misc` یک کپی با نام `auto.media` می‌سازیم.

```
#cp /etc/auto.misc /etc/auto.media
```

سپس فایل `auto.media` را باز کرده تغییرات لازم را اعمال می‌کنیم.

```
#vi /etc/auto.media
```

```
nfs-share                -rw,sync                192.168.10.1:/mnt/nfs-share
```

توضیح خطوط تعریف شده :

`nfs-share` : نام فولدیری که قرار است در سیستم کلاینت `mount` شود. این نام به دلخواه انتخاب می‌گردد.

`-rw,sync` : دایرکتوری مورد نظر به چه صورتی `mount` شود.

192.168.10.1:/mnt/nfs-share : در اینجا آدرس سروری که قرار است از دایرکتوری `export` شده

آن استفاده کنیم را وارد می‌کنیم.

```
#service autofs restart
```

بعد از انجام تغییرات، `autofs` را ریستارت می‌کنیم.

ما مشخص کردیم دایرکتوری `/mnt/nfs-share` از سرور به دایرکتوری `/media` کلاینت و با نام `nfs-share` مونت شود. وقتی در کلاینت وارد `/media` می‌شویم و `ls` می‌گیریم محتویات پیش فرض در آن قرار دارد. ولی به محض اینکه دستور `cd nfs-share` را تایپ کنیم دایرکتوری `/mnt/nfs-share` برای ما `mount` می‌شود. دقت کنید حتما باید در مسیر مشخص شده باشیم و اگر طبق زمان تعریف شده استفاده ای از دایرکتوری `mount` شده نداشته باشیم توسط این سرویس `umount` می‌گردد.

منابع :

مطالب متفرقه منتشر شده در اینترنت

سر فصل دوره های RHCE و LPic2

راهنمای NFS

سایت centos.org