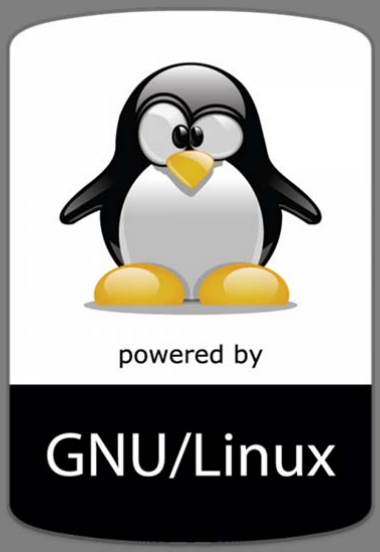
2015

اشتراک سیستم فایل با NFS

(ویرایش سوم)

نویسنده: حسام الدین توحید



skywan1@chmail.ir

بسم الله الرحمن الرحیم

مقدمه مولف :

آنچه پیش رو دارید ویرایش **سوم** مقاله اشتراک سیستم فایل با NFS است که 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 هایLPIc2 و RHCE و استفاده شده ولازم می دانم از **مهندس مهدوی فر** به خاطر راهنمایی های مفیدشان و **مرکز آموزشهای پیشرفته دانشگاه شریف (لایتک)** تشکر کافی را داشته باشم. این مطالب با نگاهی کاربردی و بدون پرداختن به بحث های تئوریک و بر اساس توزیع CentOS سری 6 گردآوری و عرضه شده ،که امیدوارم مطالب ارائه شده بتواند باعث ارتقاء دانش فنی کاربران لینوکس و متخصصین IT شود. زکات علم نشر آن است.

موفق باشید

حسام الدین توحید

بهمن 1395

skywan13@chmail.ir

فهرست

- مقدمه ای بر NFS 4
- مزایای استفاده از NFS 5
- ویژگی های NFS.v4 5
- RPC های مورد استفاده در NFS 6
- نصب و راه اندازی سرویس NFS 7
- Export دایرکتوری در NFS 9
- مثالهایی از Export دایرکتوری در NFS 11
- استفاده از دایرکتوری Export شده در کلاینت 13
- اختصاص پورت های ثابت به NFS 13
- مبحث Autofs 15
- بررسی دستورات مربوط به NFS 16

skywan13@chmail.ir

مقدمه ای بر NFS

NFS مخفف Network File System است و 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 وسیله‌ی 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 به بعد استفاده می شود. ورژن 4 این نرم افزار از سری 6 لینوکس های RedHat Base به آنها اضافه شد. با کمک NFS ، ما می‌توانیم اشتراک فایل بین سیستم های Unix Base را راه اندازی کنیم. تا پیش از NFSv4 تمامی نسخه های NFS بصورت Stateless بودند، به این معنی که در پروتکل هایStateless ، هر درخواست، هیچ اطلاعاتی از درخواست های پیشین ندارد و مستقل از درخواست های قبلی می باشد. انواع متفاوتی از سیستم فایل ها قابل دسترسی هستند که این کار را با استفاده از یک سری API انجام می‌دهند، به این api ها **Virtual File System** گفته می‌شود.

هدف اصلی از **Virtual File System** ، اجازه دادن به برنامه های کاربردی کلاینت برای دسترسی به انواع سیستم فایل ها در یک شکل واحد بدون آگاهی از این تفاوت ها میباشد، 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. با VFS دسترسی به سیستم فایل های مختلف بر روی ماشینها امکان‌پذیر 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. ساختار VFS سیستم فایل سرور را در ماشین کلاینت قرار می‌دهد.

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 هرکدام از این سیستم ها می توانند نقش سرویس دهنده NFS را داشته باشند و دایرکتوری ها و فایل های خود را به اشتراک بگذارند و همزمان به عنوان سرویس گیرنده به فعالیت بپردازد.

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 می‌کنند پس شاید بنابر سیاست کاری 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 هردو سمت سرویس دهنده و سرویس گیرنده است که برای کارکرد صحیح سرویس NFS بسیار لازم و ضروری می باشد.

تمامی نسخه های NFS از پروتکل TCP استفاده می کنند. NFSv2 امروزه کمتر استفاده می شود اماNFSv3 و NFSv4 بسیار کاربرد دارند. NFSv4 بواسطه ثابت بودن پورت و تنظیمات فایروال می تواند برروی اینترنت هم کار کرده و همچنین از ویژگی ACL نیز پشتیبانی کند. NFSv2 و NFSv3 از پروتکلUDP استفاده می‌کنند. مشکل اصلی ورژن های 3و2و1 رندوم بودن پورت های ارتباطی آنها می باشد چون نمی‌توانیم پورت های مربوطه را بر روی فایروال باز کنیم، ولی 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 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 شده‌اند را نداشت. این اطلاعات و تاریخچه‌ها برای قفل کردن فایل ها لازم می‌باشد. یک راه حل برای این مشکل بکار بردن ابزار های مجزا از NFS مانند statd و lockd برای قفل کردن فایل هاست که در NFSv2 و NFSv3 استفاده می‌شوند. اما NFSv4 ویژگی statfull که نشان دهنده مبتنی بودن بر حالت و نگه داشتن تاریخچه ای از فعالیت هاست را دارا می‌باشد.

NFSv4 مبتنی بر state است و هم برروی کلاینت و هم برروی سرویس دهنده اطلاعاتی را نگه می دارد مانند اینکه کدام فایل‌ها باز هستند و یا کدام‌ها قفل شده‌اند. این اطلاعات در صورتی که سرویس دهنده دچار مشکل شوند برای رفع اشکال سرویس دهنده میان آنها رد و بدل خواهد شد. در ادامه به مزایا استفاده از این سرویس و ویژگی‌های ورژن 4 آن می‌پردازیم.

مزایای استفاده از NFS

1. NFS اجازه دسترسی محلی به فایل‌های از راه دور را می‌دهد.
2. NFS از معماری سرویس گیرنده / سرویس دهنده استاندارد برای به اشتراک گذاری فایل بین همه ماشین‌های مبتنی بر unix استفاده می‌کند.
3. با NFS ، هیچ نیازی نیست که روی هر دو ماشین، سیستم‌عامل 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. نیاز به هیچ refresh دستی برای دسترسی به فایل‌های جدید نیست.
7. نسخه جدیدتر NFS از ACL پشتیبانی می‌کند.
8. می‌توان با Firewallها و Kerberos آن را امن‌تر کرد.

ویژگی های NFS.v4

1. سازگاری با فایروال ها و ابزار هایNAT
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 Uniqad
7. کارایی بالا حتی بر روی شبکه‌هایی با پهنای باند پایین
8. برمبنای TCP کار کرده و پورت ثابت آن 2049 می‌باشد و البته نیاز به نرم افزارهای کمکی مثل Portmapper را هم ندارد.
9. کلاینت های ویندوزی می‌توانند از share های NFS استفاده کنند.
10. ویرایش چهار NFS 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 نمی‌کرد. نرم افزارهای دیگری مثل Portmapper پیشنهاد دهنده پورت به کلاینت بودند. Portmapper چک می‌کند NFS با چه پورتی 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 سرویس‌ها را برای اجرای NFS استفاده می‌کنند. تمامی نسخه‌های NFS متکی بر RPC یا Remote Procedure Call میان کلاینت‌ها و سرورها هستند. سرویس‌های RPC در لینوکس تحت سرویس Portmap کار می‌کنند. در زیر فهرستی از سرویس‌ها که بصورت ترکیبی با هم در حال ارتباط و کار کردن برای اجرای NFS هستند آمده است:

- **nfs** : اصلی ترین سرویس که دیگر سرویس‌ها با فعال شدن آن نیز فعال می‌شوند.
- **nfslockd** : در سمت کلاینت اجازه می‌دهد تا فایل ها را بروی سرور lock کند.
- **portmap** : همانطور که گفته شد سرویس‌های RPC تحت لینوکس توسط portmap کنترل می‌شوند و مسئولیت تنظیم کردن اتصال ها برای سرویس‌های RPC درخواست شده را به عهده دارد. این نرم‌افزار هم در سرور و هم در کلاینت باید وجود داشته باشد.

RPC های مورد استفاده در NFS

وقتی دو کلاینت لینوکسی قصد دارند به منابع اشتراکی هم متصل شوند از پروتکل NFS استفاده می‌کنند و خود NFS در پس زمینه از RPCها برای اتصال استفاده می‌کند. NFS برای ارتباط نیاز به شش نوع RPC دارد. تمام فایل سیستم‌هایی که تحت شبکه کار می‌کنند برای عملکرد صحیح خود از سرویسی به نام RPC استفاده می‌کنند. در واقع RPC پروسه‌ای است که Computing دستورات بین مبدا و مقصد را انجام می‌دهد تا کلاینت درگیر پروسه ارتباط نشود.

در ادامه این rpc ها به همراه توضیحات آمده است:

- **rpc.mountd :** این سرویس در خواست‌های mount را از سمت کلاینت دریافت کرده و بررسی می‌کند که دایرکتوری درخواست شده در حال حاظر به اشتراک گذاشته شده یا خیر. علاوه بر mount ، عمل umount شدن منبع اشتراکی را هم انجام می‌دهد. فهرست دایرکتوری‌ها صادر شده در فایل /etc/exports قرار می‌گیرند.

Exports شدن به این معنی است که یک دایرکتوری را در شبکه به اشتراک بگذاریم. این سرویس در هنگام فعال کردن NFS بصورت خودکار فعال می‌شود.

- **rpc.nfsd :** این سرویس با هسته لینوکس برای مواجه شدن با در خواست‌های پویا از سمت کلاینت کار میکند.
- **rpc.nfslock :** به کلاینت اجازه می‌دهد فایل‌های 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 نمی‌شود.
- **rpc.statd :** کلاینت ها را در صورتی که سرور restart شود باخبر می‌کند. کار این RPC مانیتور کردن مبداء و مقصد می‌باشد یعنی اگر هر کدام از طرفین ریبوت شوند به طرف مقابل خبر داده می‌شود تا فایل ها lock شده را از حالت قفل خارج کنند. این سرویس توسط nfslock فعال می‌شود و در NFSv4 استفاده نمی‌شود.
- **rpc.rquotad :** این سرویس اطلاعات سهمیه‌بندی را برای کاربران راه دور (remote users) فراهم می‌کند. سهمیه‌بندی یعنی اینکه هر کاربر راه دور چه مقدار فضا می‌تواند برای ایجاد فایل و ... داشته باشد. این سرویس به همراه فعال کردن سرویس NFS خودکار فعال می‌شود.
- **rpc.idmapd :** بعنوان نگاشت کننده نام ها و ID ها در NFS به کار می‌رود. این سرویس UIDها و GID ها را به نام ترجمه می‌کند. فایل مرتبط با آن **/etc/idmapd.conf** می‌باشد.
- **rpc.gssd :** یک پروتکل امنیتی است که در ایجاد ارتباط، Security Conetxt های لازم برای کرنل را load می‌کند. نکته ضروری دیگر اینکه، برای استفاده از NFS برروی سیستم مطمئن شوید که بسته های portmap, nfs-utils و nfs-utils-lib نصب شده باشند. بسته سرور NFS شامل سه سرویس گنجانده شده، در بسته‌های portmap و nfs-utils می‌باشد.

نصب و راه اندازی سرویس NFS

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

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

در صورت نصب نبودن ، در سیستم های ردهت جهت نصب nfs و portmap از yum استفاده کرده :

```
#yum  -y  install  nfs-utils.*
#yum  -y  install  portmap
```

و 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 دوباره query می‌گیریم :

#rpm -qa | grep nfs

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

#rpm -ql nfs

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

#rpm -qi nfs

سپس با دستور chkconfig مشخص می‌کنیم این سرویس در چه runlevel هایی فعال باشد :

#chkconfig --level 35 nfs on
#chkconfig --level 35 portmap on

و در انتها سرویس را reset می‌کنیم :

#service nfs restart
#service portmap restart

با این دستور می توانیم لیست export های سیستم را مشاهده کنیم. همانطور که گفته شد export اشاره به منابع به اشتراک کذاشته شده دارد.

#exportfs -rva

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

v : جزئیات را کاملا نشان میدهد.

a : export های جدید را نشان می دهد.

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

#rpcinfo -p

برای بررسی وضعیت سرور از دستور زیر استفاده کنید و به جای server-name می‌توانید نشانی ip و یا نام سرور NFS را وارد کنید.

#rpcinfo -p server-name

خروجی این دستور منابع اشتراکی شبکه را نشان می دهد

#showmount -e 192.168.10.1

نکته مهم: سرویسی که باعث می‌شود در هر بار ریست سرویس پورت جدیدی به آن تعلق بگیرد portmapper است که در ورژن 4 هیچ کاربردی ندارد.

skywan13@chmail.ir

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 رویه تغییر کرده است.

روند کار برای پیاده‌سازی سرویس NFS بدین گونه است که ابتدا باید سرویس‌ها و برنامه‌های لازم را هم برروی سرویس دهنده (NFS Server) و هم برروی سرویس‌گیرنده (NFS Client) نصب کنیم. سپس تعیین می‌کنیم چه دایرکتوری‌هایی باید اشتراکی شوند و در نهایت دایرکتوری‌های اشتراکی شده را در سیستم‌های کلاینت mount می‌کنیم. مطلب دیگر اینکه لازم است Firewall نیز بصورت مناسب برای اجازه تردد 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:

#vi /etc/exports

shared_directory IP or machin_name(OPTIONS)

در ادمه قسمت‌های این خط توضیح داده می‌شود:

- **shared_directory** نام دایرکتوریی از NFS Server است که برای دیگر ماشین‌های شبکه Share شده است .
- **IP or machin_name** آدرس یا نام ماشین (های) کلاینتی است که مجاز به mount کردن دایرکتوری در سیستم محلیشان می‌باشند.
- **OPTIONS**گزینه‌هایی هستند که بر شیوه استفاده کلاینت از دایرکتوری اشتراکی تاثیر می‌گذارند. در زیر تعدادی از مهمترین این Option ها توضیح داده شده است :
  - ***ro*** : مخفف Read Only است که کلاینت‌های تنظیم شده با این گزینه، تنها دسترسی فقط خواندنی روی دایرکتوری Mount شده دارند.
  - ***rw :*** مخفف Read&Write که کلاینت‌های تنظیم شده با این گزینه، دسترسی خواندن و نوشتن بر روی دایرکتوری اشتراکی دارند.
  - ***sync :*** این گزینه باعث می‌شود که سرور تنها پس از اینکه نوشتن داده‌ها (اعمال تغییرات) به طور کامل انجام شد، به کلاینت پاسخ دهد. این گزینه، بعنوان پیشفرض فعال است و اگر هم آن را ننویسید، باز هم همین گزینه در نظر گرفته می‌شود. با sync، آنچه که در حافظه است به اجبار بر روی دیسک سخت نوشته می‌شود.

skywan13@chmail.ir

- ***async :*** نقطه مقابل sync است و یعنی اینکه سرور قبل از تکمیل تغییرات بر روی Storage ، به در خواست های دیگر نیز پاسخ خواهد داد و یوزرهای مقابل می‌تواند از هر مقدار فایل که کپی شده به صورت Real Time استفاده کنند. async اجازه ذخیره فعالیت کاربر در حافظه را می‌دهد.

نکته مهم : sync قابلیت اطمینان و امنیت بالاتری دارد چون در آن اطلاعات کاربر باید ابتدا بر روی دیسک به طور کامل نوشته شود، اما **async** دارای کارائی و سرعت بالاتری است چون از ترافیک شبکه استفاده بهتری می‌کند.

- ***no_root_squash :*** گزینه‌ای بسیار مهم در تنظیم یک دایرکتوری برای به اشتراک گذاشتن است، چون که استفاده از آن باعث کاهش امنیت در سرور NFS خواهد شد. اگر از این گزینه استفاده می‌کنید باید بدانید که کاربر root روی ماشین کلاینت، روی دایرکتوری اشتراک شده، دسترسی root مطابق با root ماشین سرور را خواهد داشت و توصیه می‌شود که از این گزینه اصلا استفاده نکنید.

- ***root_squash :*** نقطه مقابل no_root_squash است. با استفاده از این گزینه، درخواست‌های آمده از uid=0 و gid=0 از کاربر anonymous که به nobody user یا nfsnobody user شناخته می‌شود، 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** و 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 می‌شود، با تنظیم این گزینه نمی گذاریم که کلاینت ها به دایرکتوری‌های بالایی دایرکتوری اشتراکی شده دسترسی پیدا کنند و تنها به زیر دایرکتوری‌های، دایرکتوری اشتراکی دسترسی دارند.

نکته مهم : از 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.

مثالهایی از Export دایرکتوری در NFS

در این قسمت ابتدا مثال‌های ساده‌ای با فرمت های گوناگون از export شدن یک منبع بیان می‌شود و سپس به اشتراکی کردن یک دایرکتوری به سبک NFS v4 می‌پردازیم.

Examples:

❖ **مثال 1:** در مثال زیر دایرکتوری /nfs/nfs-share/ برای یک کلاینت با آدرس 192.168.10.1 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)
```

❖ **مثال 2:** در مثال زیر یک محدوده IP ها (IP Range)در نظر گرفته می‌شود.

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

❖ **مثال 4:** و در این مثال یک ماشین عضو دامنه skywan13.local

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

❖ **مثال 5:** و در اینجا تمامی ماشین‌های عضو دامنه skywan13.local را شامل می‌شود.

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

❖ **مثال 6:** فرض کنید می‌خواهیم 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 لذا به شیوه زیر عمل می‌کنیم:

```
#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
```

❖ **مثال 7:** در ورژن های قبل از NFSv4 باید مسیر کامل دایرکتوری را هم در فایل exports و هم در خط mount وارد می‌کردیم یعنی حتما باید از ریشه مسیردهی می‌شد. به اینکار RootDirectory گفته می‌شود. این یک نقص امنیتی بود که در ورژن 4 اصلاح شد. جهت بررسی از ورژن 3 به 4 مثالهای زیر را دنبال کنید:

skywan13@chmail.ir

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
```

نکته:

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

nohide: یعنی محتوا را نشان دهد .

❖ **مثال 8:** 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 .

skywan13@chmail.ir

استفاده از دایرکتوری Export شده در کلاینت ها

توضیحات سمت کلاینت :

پس از تمامی کارهای انجام گرفته در سرور، نوبت به پیکربندی کلاینت ها می‌رسد. 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 دستورmount بصورت زیر اجرا شود:

```
#mount -t nfs -o vers=4 192.16.10.1:0/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 باید 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 می‌شود. در صورتی که با خطای زیر در هنگام mount کردن یک دایرکتوری در کلاینت موجه شدید دلیل آن بسته بودن پورت های 111 و 2049 است.

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

بطور کلی برای تست می‌توانید در سرور و کلاینت فایروال را خاموش یا غیر فعال کنید.

نکته مهم: سرویس NFS برروی پورت 2049 و Portmap برروی پورت 111 کار می کنند.

اختصاص پورت های ثابت به NFS

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

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

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

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=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 پورتها را درون آن باز کنند. در انتها به منظور پیکربندی iptables خطوط زیر را به فایل /etc/sysconfig/iptables اضافه می کنیم .

#vi /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 می‌کنیم .

#service iptables restart

مبحث Autofs

وقتی یک کلاینت دایرکتوری share شده توسط سرور را mount می‌کند یعنی یک session و کانال ارتباطی بین خود و سرور ایجاد کرده است، حال اگر ارتباط سرویس دهنده قطع شود کلاینت متوجه این موضوع نشده و سیستم درگیر چک کردن مداوم آن می‌شود. این حالت در کار سیستم کلاینت اخلال ایجاد کرده و کلا آن را از کار می‌اندازد و به هیچ وجه اجازه umount شدن دایرکتوری mount شده را نمی‌دهد مگر اینکه سیستم را ریست کنید.

این که ما همیشه یک دایرکتوری را به صورت ثابت mount شده داشته باشیم یک ضعف به حساب می‌آید چون اگر سرور از کار بیافتد کلاینتها در کارکرد با مشکل مواجه خواهند شد. برای رفع این ضعف سرویس autofs معرفی شد. NFS از ویژگی‌های autofs برای غلبه بر این نقص استفاده می‌کند. این کار بوسیله صرف نظر کردن از فایل fstab و استفاده از یک فایل مخصوص برای mount که می‌تواند به صورت توزیع شده بین کلاینتها وجود داشته باشد، صورت می‌گیرد. 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 می‌کند. گذشته از این می‌توان زمان تداوم NFS mount را بعد از هر بار mount به صورت خودکار مشخص کرد.

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

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**

Map file اصلی مربوط به auotofs فایل auto.master است که ساختار و قالب ساده ای دارد. در ستون اول آن نام Mount point و در ستون دوم، فایل کمکی که عملیات آن را به عهده دارد آمده است. در ستون سوم هم می‌توان کلیدها و گزینه‌های مربوط به دستور mount را مشخص کرد.

auto.master کانفیگ اصلی و گلوبال سرویس autofs 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 دایرکتوریی 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)

skywan13@chmail.ir

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

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

سپس فایل auto.media را باز کرده تغییرات لازم را اعمال می کنیم.

#vi /etc/auto.media

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

توضیح خطوط تعریف شده :

nfs-share: نام فولدری که قرار است در سیستم کلاینت mount شود. این نام به دلخواه انتخاب می گردد.

-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 وارد می‌شود.

بعد از انجام تغییرات ، autofs را ریستارت می کنیم. **#service autofs restart**

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 استفاده‌ای از دایرکتوری mount شده نداشته باشیم دایرکتوری توسط این سرویس umount می گردد. و در آخر مراحل زیر را برای راه اندازی دایمی autofs به کار ببرید.

#chkconfig autofs on

بررسی دستورات مربوط به NFS

در اینجا به بررسی برخی از دستورات مهم و پرکاربرد مرتبط با NFS پرداخته می‌شود. برای اطلاعات بیشتر در مورد آنها می‌توانید به اینترنت و یا صفحات راهنمای NFS مراجعه کنید.

showmount -e

نمایش share های در دسترس روی ماشین محلی را انجام می‌دهد.

showmount -e server-ip or hostname

لیست و اطلاعات دایرکتوری‌های mount شده یک سرور NFS را نشان می‌دهد.

showmount -d

لیست تمام زیر دایرکتوری‌ها(sub directorie)

skywan13@chmail.ir

exportfs -V

یک لیست از فایل های share شده وآپشن‌های روی یک سرور را تهیه و نمایش می‌دهد.

exportfs -a

Exportهمهshare های لیست شده در /etc/exports با توجه به نام

exportfs -u

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

exportfs -r

از این دستور برای تازه کردن (Refresh) لیست سرور پس از تغییر در /etc/exports استفاده می‌شود.

df –F nfs

فهرست سیستم فایل‌های mount شده به پایانه را نشان می‌دهد.

nfsstat –s

این دستور خطاهای سرور NFS را گزارش می‌کند.

nfsstat –c

این دستور گزارش خطاها را برای کلاینتها تولید می‌کند.

skywan13@chmail.ir