

Configuring: How do I run NFSv4 with Bright?

How can I use NFS version 4 with Bright?

By configuring it as shown in the example later on. But first, some background.

Why have NFSv3 as the default anyway?

NFS version 3 is provided by the parent distributions by default. Bright is therefore configured to work with NFS version 3 by default.

By default, Bright uses NFS for the shared filesystems, `/cm/shared` and `/home`. The Bright cluster manager relies on UIDs and usernames being the same across all the filesystems being managed.

Why use NFSv4? Any problems if we use it?

NFSv4 has some features that NFSv3 does not have. Indeed, sometimes, the cluster administrator really needs features that are only provided by NFSv4. This can lead to a compatibility problem, because NFSv4 has a feature where it can map users to the appropriate IDs across systems. NFSv3 does not support this mapping. In addition, the underlying RPC authentication that NFSv4 uses to open files is still not able to support this mapping. Relying on this feature can therefore lead to confusion and applications that do not work as expected if applications rely on the NFSv3 kind of behavior.

So, for example, in NFSv4, while a filename belonging to a common username can be seen, the contents of the file cannot be opened unless an ID matches as well. This behavior often results in broken applications in NFSv4. In practice, the files that cannot be mapped to the user via RPC are simply treated as files owned by user `nobody`, which will indeed often lead to broken applications.

So, how do we make NFSv4 work across the filesystems? For what applications?

The way to avoid the broken behavior is to make sure that system and LDAP users have their UIDs/GIDs completely synchronized across the filesystems that are used, just like in NFSv3. The only way to ensure this is to have the administrator synchronize the user and his UIDs/GIDs

Configuring: How do I run NFSv4 with Bright?

across the filesystems. So, typically the administrator must have control over the shared storage NFS server, and be able to change the UID/GIDs, as well as have control over the UID/GIDs on the rest of the systems used by the applications that use NFS.

Applications that need NFSv4 features:

- Some workload managers (OGS, UGE) need NFS version 4 when an HA setup is used:
- NFS over RDMA requires NFSv4.

In some cases NFSv4 is needed and the user--->UID/GID mapping issue is not relevant. In this case the NFS version can be set to version 4.

Example: Implementing NFSv4 for a category:

For the 'default' category, the NFS-shared filesystems '/cm/shared' and '/home' can be set to use NFSv4 as follows:

```
[root@mycluster ~]# cmsh
[mycluster]% category use default
[mycluster->category[default]]% fsmounts
[mycluster->category[default]->fsmounts]% use /cm/shared
[mycluster->category[default]->fsmounts[/cm/shared]]% append
mountoptions ",vers=4"
[mycluster->category*[default*]->fsmounts*/cm/shared*]]% commit
[mycluster->category[default]->fsmounts[/cm/shared]]% use /home
[mycluster->category[default]->fsmounts[/home]]% append mountoptions
",vers=4"
[mycluster->category*[default*]->fsmounts*/home*]]% commit
[mycluster->category[default]->fsmounts[/home]]%
```

Appending the 'vers=4' mount option means that NFS version 4 is used instead of the default version 3.

After the changes are carried out, the nodes in the category should be rebooted, so that the new settings are used.

Unique solution ID: #1262

Author: Frank Furter

Last update: 2015-02-10 21:33