

GlusterFS NFS Alpha Release Notes

March 19th , 2010

Table of Contents

Introduction.....	2
Disclaimer.....	2
Comparison of knfsd, unfsd and GlusterFS NFS.....	2
Pre-Installation Information.....	7
Product License.....	7
System Requirements.....	7
Installation.....	7
Installing from RPMs.....	7
Building from Source Tarball.....	7
Building from Source Checkout.....	7
Co-locating NFS Alpha with Existing GlusterFS Version.....	8
Compatibility.....	8
Upgrade.....	8
User Guide.....	8
Quick Start.....	8
NFS Translator Options.....	14
Filing Defects.....	15
Feedback.....	16
Product Documentation.....	16
Copyright/Trademarks.....	16

Introduction

Development of native NFS support in GlusterFS was motivated by the functional and performance limitations imposed by the currently available NFS servers, namely, the kernel NFS server and unfsd. Both servers impose significant performance overhead because of the need to use FUSE between the NFS server and GlusterFS process.

Native NFS for GlusterFS is being implemented as a translator thus completely avoiding the overhead of FUSE and of switching between GlusterFS and the NFS servers. Like all other translators, NFS can be added to an existing volume file in order to export any of the volumes in the volfile to NFS clients. Volumes can be exported through NFS by addition of only 4 to 5 lines in the volfile.

This ALPHA version of NFS translator supports NFSv3 and MOUNTv3. Support for NLMv4, the NFSv3 locking protocol, will be introduced in a future release.

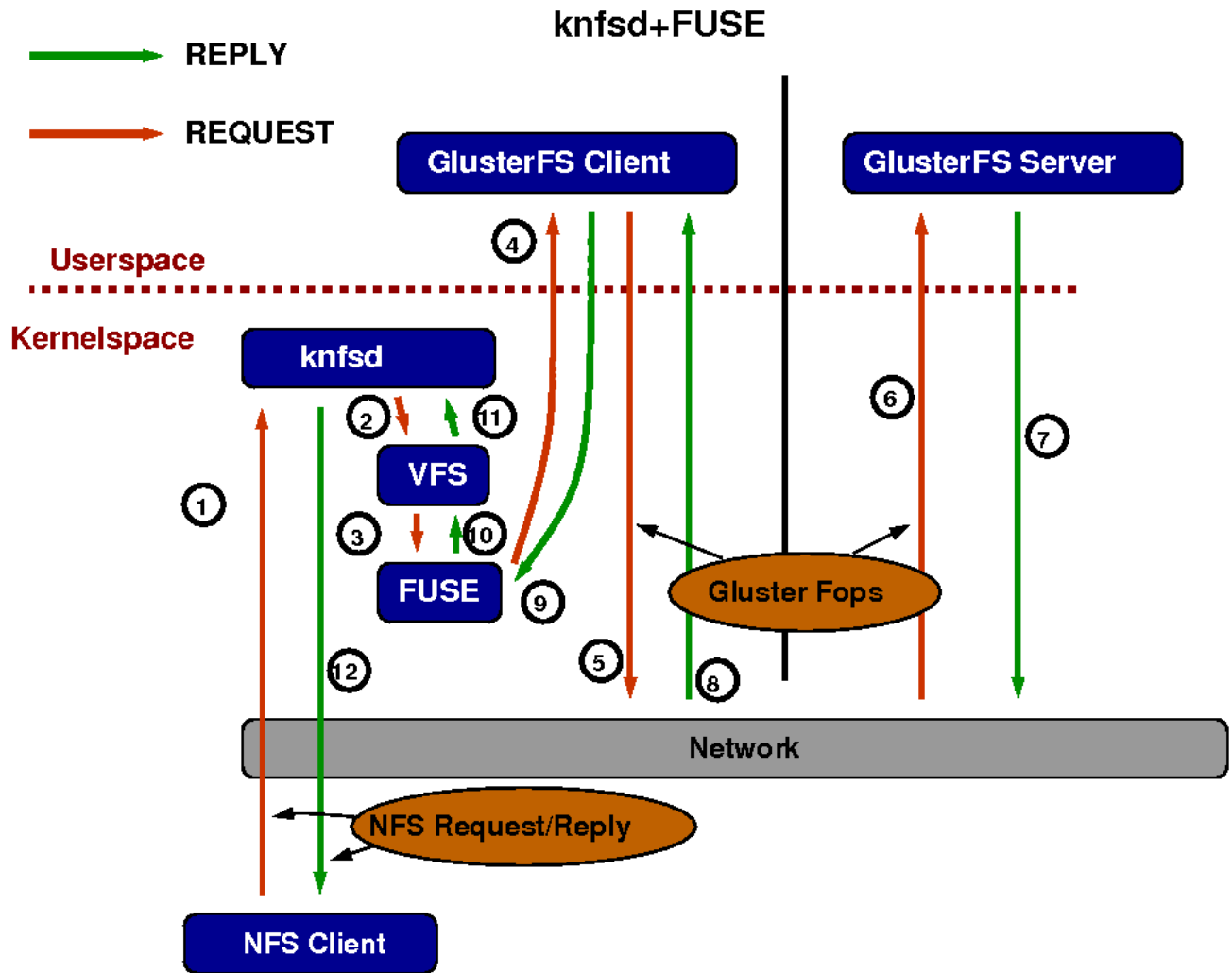
This ALPHA release has been alpha tested to be interoperable with NFS clients of Linux, Solaris and Vmware.

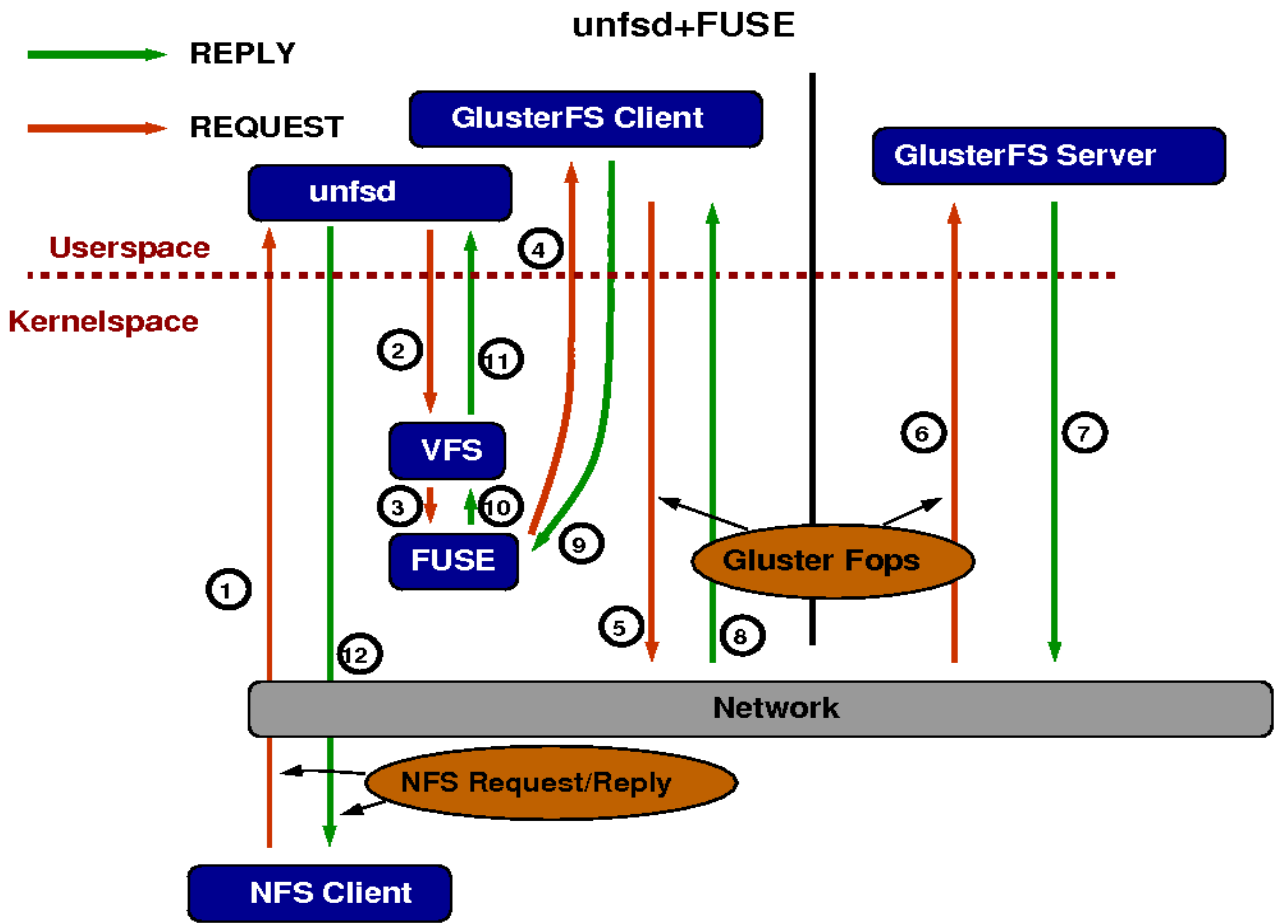
We look forward to the community for trying out other implementations of NFS clients and other virtualization platforms that utilize NFS and providing feedback to us at: nfs-alpha@gluster.com.

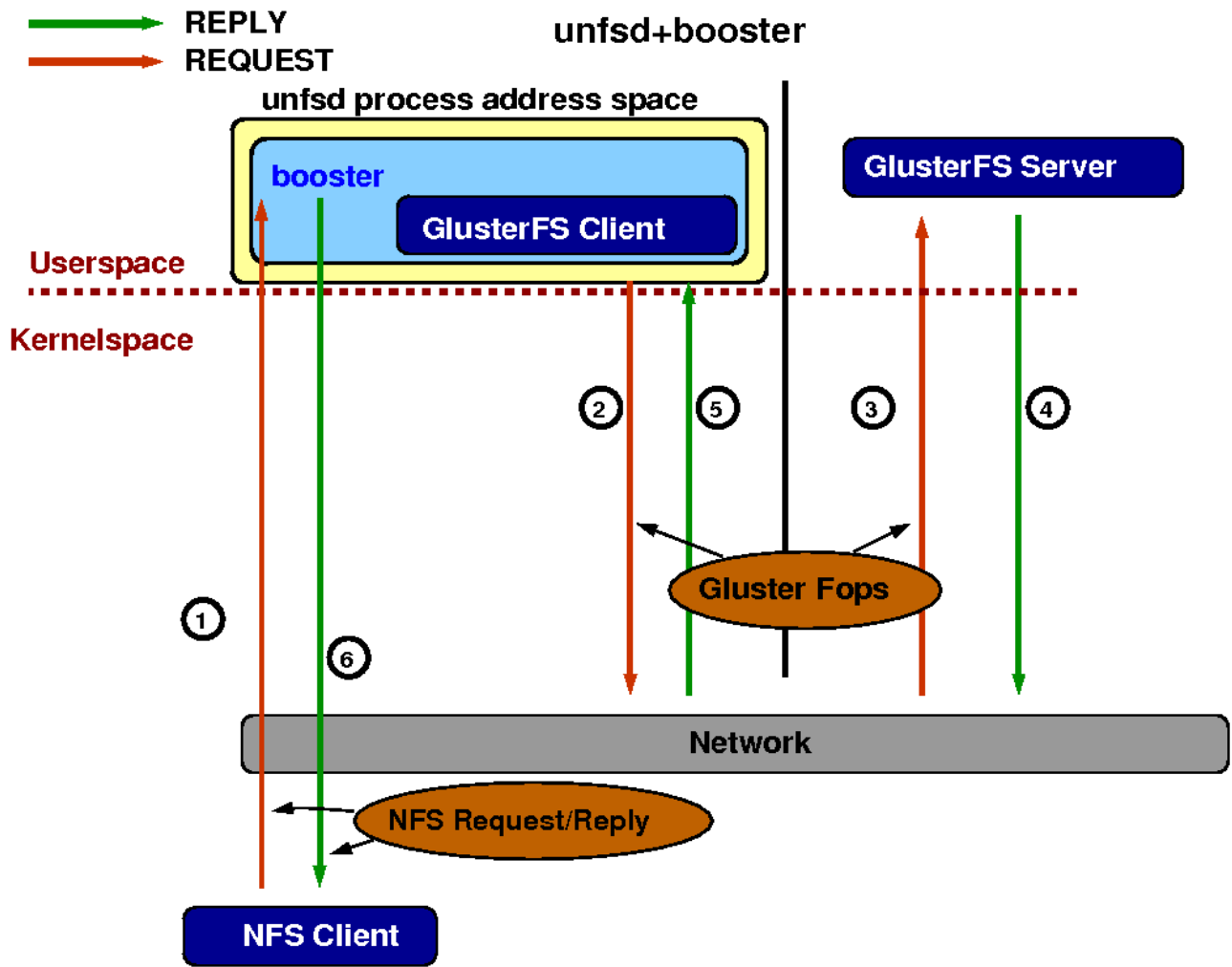
Disclaimer

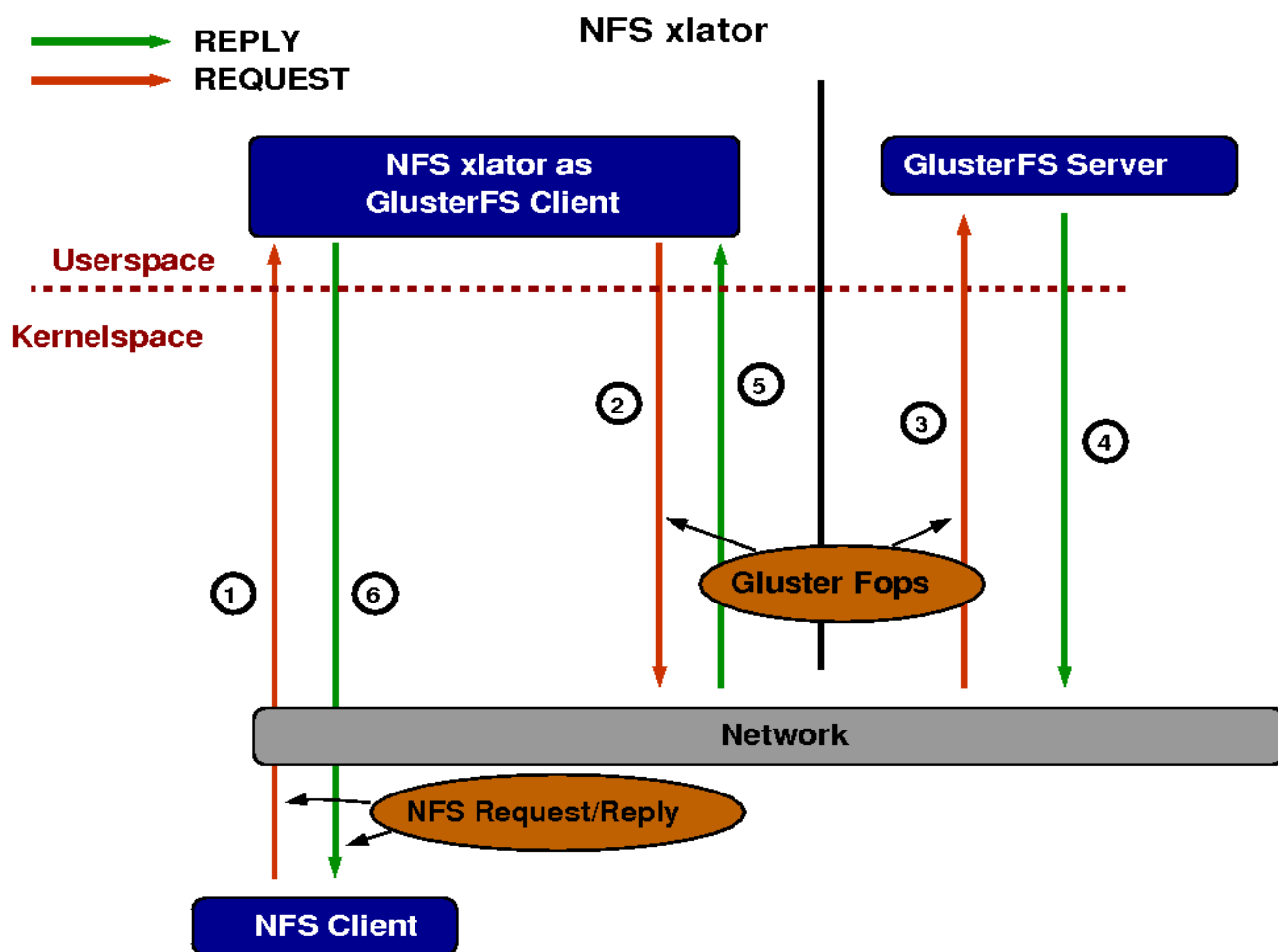
This GlusterFS release that includes NFS translator in ALPHA state is provided AS IS. We recommend that you do not use this on production setups and that Gluster, Inc, is in no way responsible for any loss of data or services.

Comparison of knfsd, unfsd and GlusterFS NFS









With GlusterFS setups, Knfsd and unfsd export GlusterFS mount points, or sections thereof. NFS translator, in contrast, does not need a mount point. Instead, it directly exports a volume using a modified GlusterFS client volfile. In effect, the mount path argument for the mount command at the NFS client is either the name of the volume or the more familiar - a “/” prepended to the volume name. See later sections for examples.

The implication is that, unlike with knfsd and unfsd where subsets of an existing GlusterFS mount-point could be exported separately as different NFS exports, with NFS translator each GlusterFS volume is exactly one NFS export by itself.

Pre-Installation Information

Product License

GlusterFS is released under GNU General Public License v3 or later. Documentation is released under GNU Free Documentation License 1.2 or later. The license for GlusterFS NFS Alpha can be viewed at <http://www.gluster.com/company/legal.php>

System Requirements

System requirements are same as those documented in the Release Notes for GlusterFS 3.0.3 release available at:

http://ftp.gluster.com/pub/gluster/glusterfs/3.0/3.0.3/GlusterFS_3.0.3_Release_Notes.pdf

Installation

Installing from RPMs

RPMs for the NFS Alpha release are available for Fedora 11 and Centos 5 at:

<http://ftp.gluster.com/pub/gluster/glusterfs/qa-releases/nfs-alpha>

Instructions to setup GlusterFS from RPMs are available at:

http://www.gluster.com/community/documentation/index.php/Storage_Server_Installation_and_Configuration

Building from Source Tarball

Although the following source tarball can be used for a fresh build and install, a later section provides instructions for setting up NFS Alpha release on machines which have an existing GlusterFS installation. GlusterFS NFS Alpha source tarball is available at:

<http://ftp.gluster.com/pub/gluster/glusterfs/qa-releases/nfs-alpha/glusterfs-nfsalpha1.tar.gz>

Building from Source Checkout

Latest ALPHA code can be obtained from the source code repository. Although the following source checkout can be used for a fresh build and install, the next section provides instructions for setting up NFS Alpha release on a machines which have an existing GlusterFS installation.

```
$ git clone git://git.gluster.com/users/shehjart/gluster-nfs-alpha.git gluster-nfs-alpha
```

```
$ cd gluster-nfs-alpha
$ ./autogen.sh
$ ./configure
$ make install;
```

Co-locating NFS Alpha with Existing GlusterFS Version

To co-locate the NFS alpha release with an existing version to facilitate testing, the Gluster NFS alpha source needs to be downloaded using either the tarball approach or directly from the repository using git. Next, in the source directory:

```
$ ./autogen.sh
$ ./configure --prefix=<NEW-GLUSTERFS-DIRECTORY>
$ make install;
```

Now, the GlusterFS NFS Alpha release binary resides in <NEW-GLUSTERFS-DIRECTORY>.

As described in detail in a later section, existing volumes files will need modifications to enable volumes defined in them to be exported via NFS.

Compatibility

NFS Alpha release requires that both the GlusterFS NFS server as well as the GlusterFS bricks run this same Alpha version. No other GlusterFS versions are supported. NFS clients must support NFS version 3.

Upgrade

If users of GlusterFS 2.0.x and 3.0.x want to use GlusterFS NFS Alpha, they need to install all GlusterFS servers and clients with GlusterFS NFS Alpha software.

Gluster recommends users backup their prior configuration and volume files prior to installing GlusterFS NFS Alpha.

User Guide

GlusterFS NFS server translator is a new translator that provides NFS version 3 file serving functionality for GlusterFS volumes.

Quick Start

Exporting GlusterFS volumes through NFS requires simply adding the NFS translator into the volume files generated using *volgen*. Since NFS is a translator, adding the following volume section to the GlusterFS client volfile allows the volumes in the file to be exported via NFS.

Please note that *glusterfs-volgen* tool does not yet support addition of NFS translator to GlusterFS volumes.

```
volume nfs-server
```

```

type nfs/server
subvolumes <vol1> <vol2> <vol3>...<volN>
option rpc-auth.addr.allow *
end-volume

```

Where <vol1>, <vol2>...<volN> are typical GlusterFS volumes which are traditionally mounted on the GlusterFS clients.

By default, all NFS clients are forbidden from mounting any of the exports in the volume file. The simplest way to reverse this restriction is to allow all NFS clients using the “rpc-auth.addr.allow” option.

To export GlusterFS volume through the NFS translator the command is similar to starting up a GlusterFS server, for eg:

```
glusterfsd -f <volfile> -L <debug-level> -l <log-file>
```

Example Posix backend for the following examples

```

volume posix
    type storage/posix
    option directory /export/homes
end-volume
volume posix-locked
    type features/locks
    subvolumes posix
end-volume
volume posix-server
    type protocol/server
    subvolumes posix-locked
    option transport-type tcp
    option transport.socket.listen-port 6102
    option auth.addr.posix-locked.allow *
end-volume

```

Exporting a single GlusterFS volume

Following is a volume file importing posix backends from two GlusterFS servers and combining them in a replicate configuration. The replicated volume is then exported using the NFS translator.

```

volume client0
    type protocol/client

```

```

        option transport-type tcp
        option remote-port 6102
        option remote-host 192.168.1.101
        option remote-subvolume posix-locked
end-volume
volume client1
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.102
    option remote-subvolume posix-locked
end-volume
volume repl-0-1
    type cluster/replicate
    subvolumes client0 client1
end-volume
#Export above replicate through NFS
volume nfs-server
    type nfs/server
    subvolumes repl-0-1
    option rpc-auth.addr.allow *
end-volume

```

On a Linux NFS client, the replicate export will be mounted by the command:

```
$ mount nfsserver:/repl-0-1 /mnt
```

Exporting multiple GlusterFS volumes

Next, suppose the NFS server needs to export two GlusterFS volumes as described by the volume file below:

```

volume client0
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.101
    option remote-subvolume posix-locked

```

```

end-volume
volume client1
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.102
    option remote-subvolume posix-locked
end-volume
volume repl-0-1
    type cluster/replicate
    subvolumes client0 client1
end-volume
volume client2
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.103
    option remote-subvolume posix-locked
end-volume
volume client3
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.104
    option remote-subvolume posix-locked
end-volume
volume dist-2-3
    type cluster/distribute
    subvolumes client2 client 3
end-volume
#Export above replicate and distribute as two separate
volumes through NFS
volume nfs-server
    type nfs/server
    subvolumes repl-0-1 dist-2-3

```

```
        option rpc-auth.addr.allow *
    end-volume
```

On a Linux NFS client, the replicate export can be mounted by the command:

```
$ mount nfsserver:/repl-0-1 /mnt
$ mount nfsserver:/dist-2-3 /mnt2
```

Exporting a single GlusterFS volume to a single NFS client

The NFS translator allows restricting the exports to certain client machines.

```
    volume client0
        type protocol/client
        option transport-type tcp
        option remote-port 6102
        option remote-host 192.168.1.101
        option remote-subvolume posix-locked
    end-volume
    volume client1
        type protocol/client
        option transport-type tcp
        option remote-port 6102
        option remote-host 192.168.1.102
        option remote-subvolume posix-locked
    end-volume
    volume repl-0-1
        type cluster/replicate
        subvolumes client0 client1
    end-volume
    #Export above replicate through NFS
    volume nfs-server
        type nfs/server
        subvolumes repl-0-1
        option rpc-auth.addr.allow 192.168.1.107
    end-volume
```

On a Linux NFS client machine 192.168.1.107, the replicate export will be mounted by the

command:

```
$ mount nfsserver:/repl-0-1 /mnt
```

For all other machines, the mount request will be denied.

Exporting a single GlusterFS volume to a subnet of clients

The NFS translator allows restricting the exports to certain client machines.

```
volume client0
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.101
    option remote-subvolume posix-locked
end-volume
volume client1
    type protocol/client
    option transport-type tcp
    option remote-port 6102
    option remote-host 192.168.1.102
    option remote-subvolume posix-locked
end-volume
volume repl-0-1
    type cluster/replicate
    subvolumes client0 client1
end-volume
#Export above replicate through NFS
volume nfs-server
    type nfs/server
    subvolumes repl-0-1
    option rpc-auth.addr.allow 192.168.1.*
end-volume
```

On a Linux NFS client machine on the given subnet, the replicate export will be mounted by the command:

```
$ mount nfsserver:/repl-0-1 /mnt
```

For machines with IP addresses other than 192.168.1.*, the mount request will be denied.

NFS Translator Options

Option	Type	Description
rpc-auth.auth-unix	Boolean eg. <onloff>	Use this option to enable or disable the AUTH_UNIX authentication scheme for all exported volumes. By default, this scheme is enabled for all exports and in fact, some NFS clients will not allow mounting an export without AUTH_UNIX enabled. Enabled by default.
rpc-auth.auth-null	Boolean eg. <onloff>	Use this option to enable or disable the AUTH_NULL authentication scheme for all exported volumes. By default, this scheme is enabled for all exports and turn it off if you require that all NFS clients be forced to use a better authentication scheme since AUTH_NULL is basically no authentication. Enabled by default.
rpc-auth.auth-null.*	Boolean eg. <onloff>	Use this option to enable or disable the AUTH_NULL authentication scheme for a specific export. By default, this scheme is enabled for all exports. This per-volume option can be used to over-ride the global value set using the previous two options.
rpc-auth.auth-unix.*	Boolean eg. <onloff>	Use this option to enable or disable the AUTH_UNIX authentication scheme for a specific export. By default, this scheme is enabled for all exports. This per-volume option can be used to over-ride the global value set using the previous two options.

rpc-auth.addr.reject	* or comma-separated list of Ipv4 addresses or hostnames	Reject only the given addresses/hostnames
rpc-auth.addr.allow	* or comma-separated list of Ipv4 addresses or hostnames	Allow only the given addresses/hostnames
rpc-auth.addr.<export>.allow	* or comma-separated list of Ipv4 addresses or hostnames	Allow access to <export> only from the given addresses/hostnames.
rpc-auth.addr.<export>.reject	* or comma-separated list of Ipv4 addresses or hostnames	Reject access to <export> only from the given addresses/hostnames.
rpc-auth.ports.insecure	Boolean, eg, <on/off>	Allow clients to mount exports from unprivileged ports.
rpc-auth.ports.<export>.insecure	Boolean, eg, <on/off>	Allow clients to only mount <export> from unprivileged ports.
nfs3.<export>.volume-access	String: “read-only” or “read-write”	Read-write access to <export> can be restricted using this option.
rpc-auth.addr.namelookup	Boolean, eg, <on/off>	Users have the option of turning off name lookup for incoming client connections using this option. In some setups, the name server can take too long to reply to DNS queries resulting in timeouts of mount requests. Use this option to turn off name lookups during address authentication. Note, turning this off will prevent you from using hostnames in rpc-auth.addr.* filters. By default name lookup is on.

Filing Defects

Defects can be filed at the Gluster Bug tracker at:

<http://bugs.gluster.com>

To help us identify NFS Alpha related bugs, please use the following values when filing the bugs:

Product: GlusterFS

Component: nfs

Release: nfs-alpha

Feedback

Apart from the Gluster-users community list, feedback for this release can be emailed to:

nfs-alpha@gluster.com

Product Documentation

Complete GlusterFS product documentation is available at:

http://www.gluster.com/community/documentation/index.php/GlusterFS_User_Guide

Copyright/Trademarks

Gluster, the Gluster logo and GlusterFS are all trademarks of Gluster, Inc. All other trademarks, registered trademarks, and product names may be trademarks of their respective owners. Additional legal information is at: <http://www.gluster.com/company/legal.php>