



BeeGFS®

BeeGFS

Typical Administration Tasks

BeeGFS.io



2022

Agenda



- 🐝 Installation, Upgrade, Versioning
- 🐝 Administration Tools
- 🐝 Advanced Configurations and Features
 - 🐝 Multiple BeeGFS instances
 - 🐝 Storage pools
 - 🐝 Data Striping
- 🐝 Monitoring
- 🐝 Quota Management

Installation



🐝 Detailed installation guide

https://doc.beegfs.io/latest/quick_start_guide/quick_start_guide.html

🐝 Package repositories

ubuntu



```
# wget -O /etc/yum.repos.d/beegfs-rhel8.repo
```

```
http://www.beegfs.com/release/beegfs\_7.4.2/dists/beegfs-rhel9.repo
```

Or

```
# curl https://www.beegfs.io/release/beegfs_7.4.2/dists/beegfs-rhel9.repo  
--output /etc/yum.repos.d/beegfs-rhel7.repo
```

Installation



🐝 Prepare Firewall TCP and UDP Ports

```
server01:~ # firewall-cmd --zone=public --permanent --add-port=8008/tcp --add-port=8008/udp
```

```
server01:~ # firewall-cmd --zone=public --permanent --add-port=8005/tcp --add-port=8005/udp
```

```
server01:~ # firewall-cmd --zone=public --permanent --add-port=8003/tcp --add-port=8003/udp
```

```
server01:~ # firewall-cmd --zone=public --permanent --add-port=8004/tcp --add-port=8004/udp
```

```
server01:~ # firewall-cmd --zone=public --permanent --add-port=8006/tcp --add-port=8006/udp
```

```
server01:~ # firewall-cmd --reload
```

🐝 Disable or set to permissive selinux

```
server01:~ # setenforce permissive
```



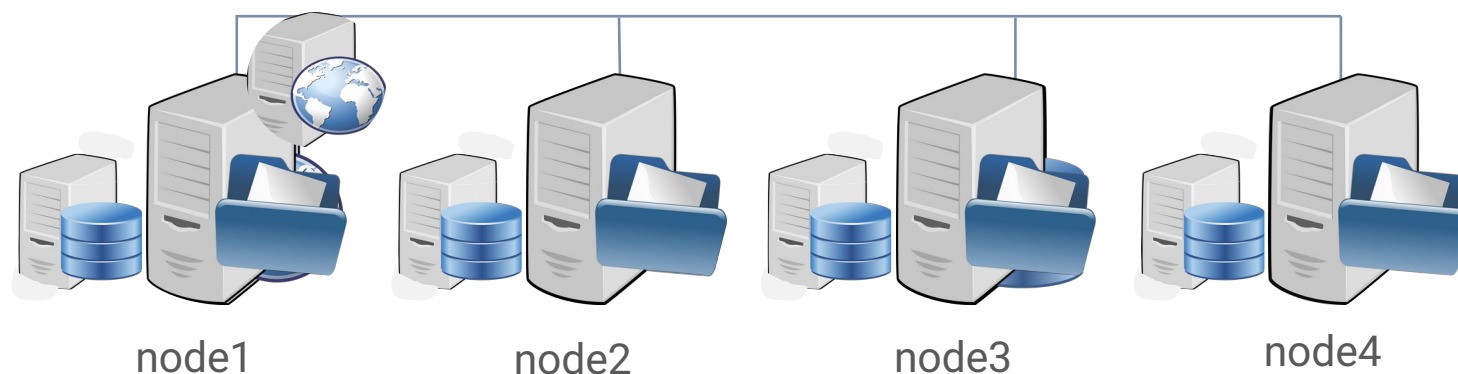
server01

DIY: Preparation for BeeGFS Installation



🐝 Prepare test environment

- 🐝 Format disk `/dev/nvme1n1`
- 🐝 Partition disk `/dev/nvme2n1` with two partitions of 10GB each
 - 🐝 Use `fdisk` or another tool.
- 🐝 Format the two partitions
- 🐝 Create mount points and mount the disks
 - 🐝 `/dev/nvme1n1` is for Management and Metadata
 - 🐝 `/dev/nvme2n1p*` is for Storage
- 🐝 Do this on all 4 nodes of the cluster



Installation BeeGFS Management Server



🐝 Management Service

```
server01:~ # yum install beegfs-mgmt
```

```
server01:~ # beegfs-setup-mgmt -p /BeeGFS/management
```

🐝 Create authentication file in /etc/beegfs and edit beegfs-mgmt.conf

```
Example: connAuthFile = /etc/beegfs/AuthFile
```

```
server01:~ # systemctl start beegfs-mgmt
```



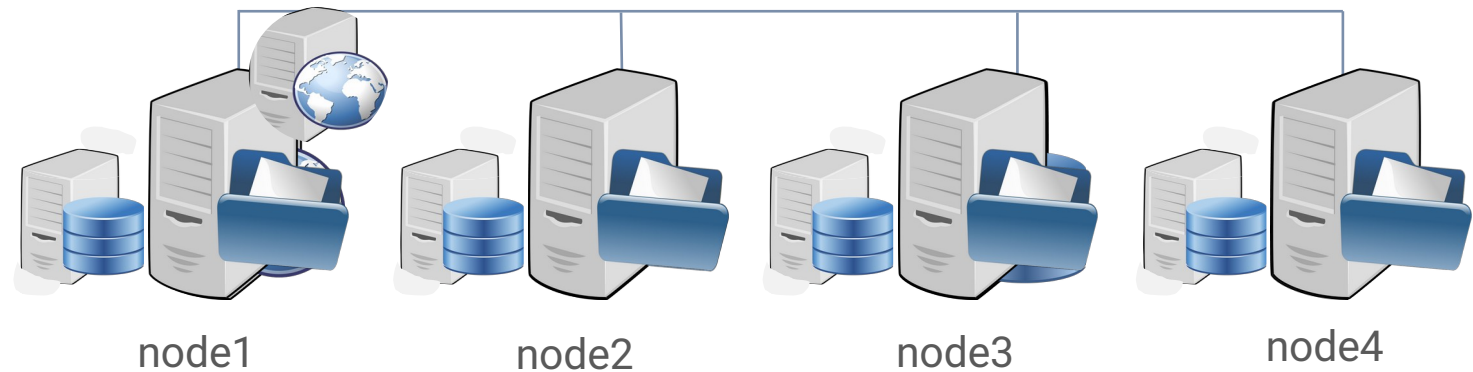
server01

DIY Install and configure BeeGFS Management



🐝 Install and configure Beegfs Management Server

🐝 Do this on the first node of the cluster



Installation Metadata

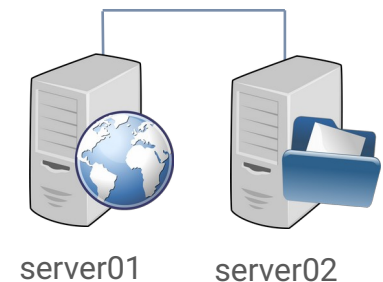


Metadata Service

```
server01:~ # yum install beegfs-meta
```

```
server01:~ # beegfs-setup-meta -p /BeeGFS/metadata -s 1 -m server01
```

```
server01:~ # systemctl start beegfs-meta
```

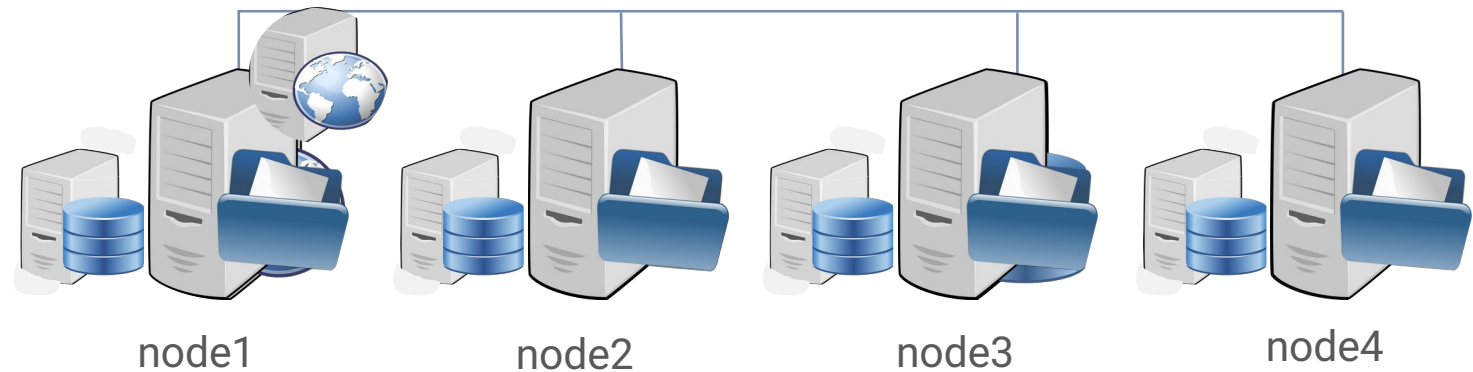


DIY: Metadata



🐝 Install and configure Beegfs Metadata Server

🐝 Do this on all 4 nodes of the cluster



Installation BeeGFS Storage Server



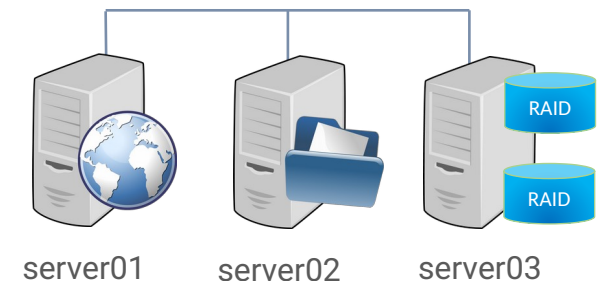
Storage Service

```
server01:~ # yum install beegfs-storage
```

```
server01:~ # beegfs-setup-storage -p /storage/stor1 -s 1 -i 101 -m server01
```

```
server01:~ # beegfs-setup-storage -p /storage/stor2 -s 1 -i 102 -m server01
```

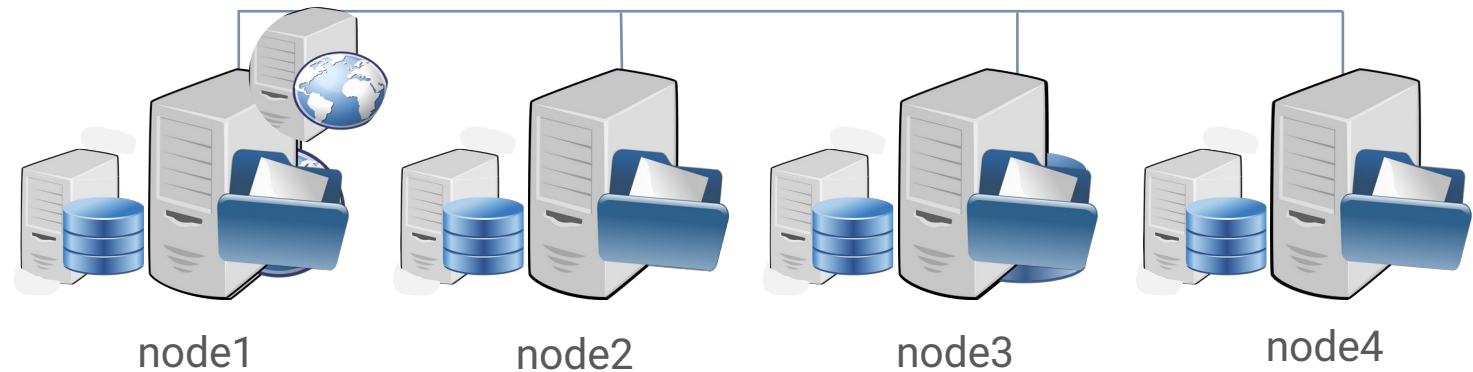
```
server01:~ # systemctl start beegfs-storage
```



DIY: BeeGFS Storage Server



- 🐝 Install and configure Beegfs Storage Server
 - 🐝 Do this on two (2) nodes of the cluster
 - 🐝 Each Storage Server has two targets of 10GB size



Installation Client and Helperd



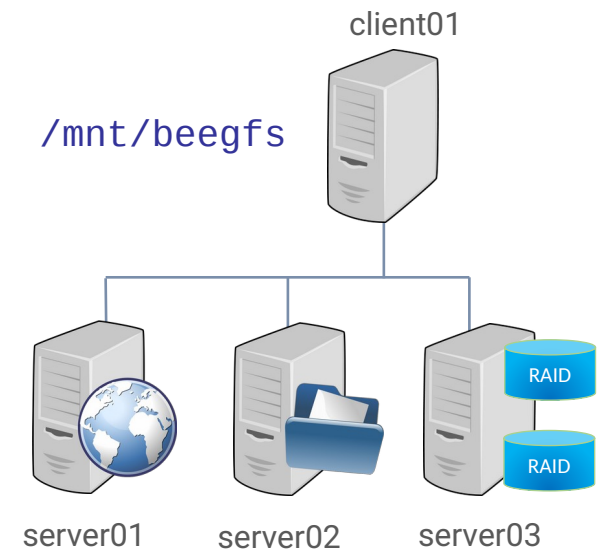
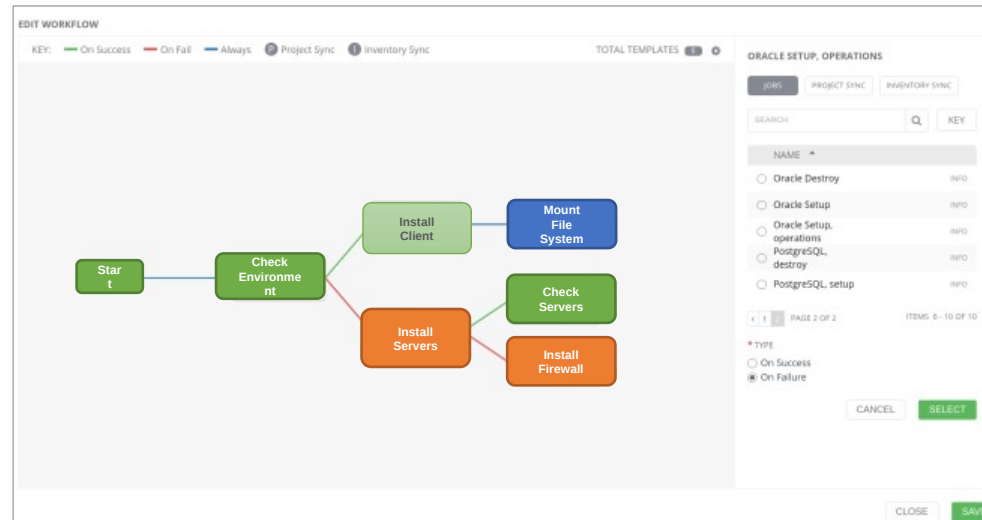
Client Service

```
client01:~ # yum install beegfs-client beegfs-helperd
```

```
client01:~ # beegfs-setup-client -m server01
```

```
client01:~ # systemctl start beegfs-helperd
```

```
client01:~ # systemctl start beegfs-client
```



Client: Important configuration files

🐝 > beegfs-client.conf

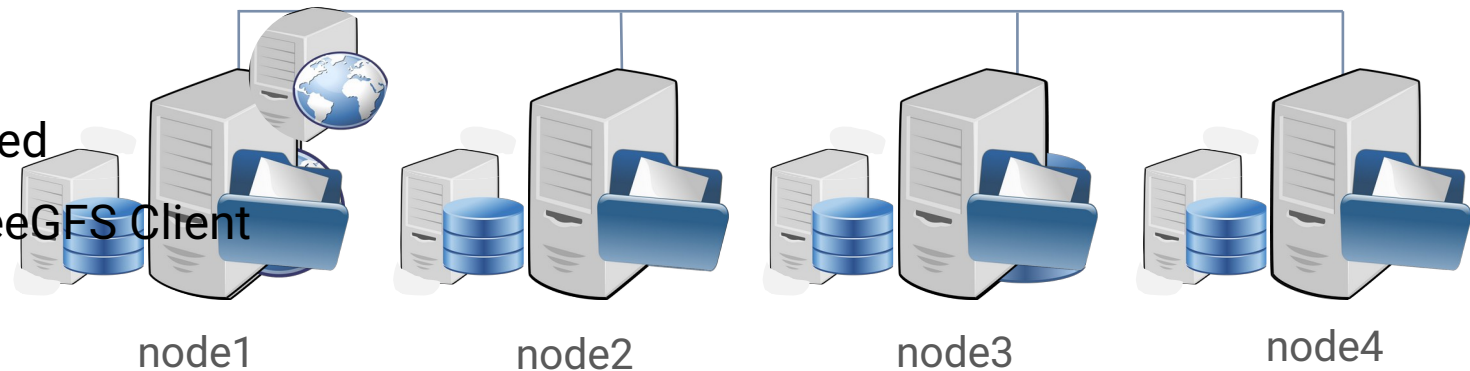
- 🐝 Defines connection to the BeeGFS Management Server
- 🐝 Authentication file

🐝 > beegfs-mounts.conf

- 🐝 Defines the mountpoint of the BeeGFS Client
- 🐝 Default: /mnt/beegfs

🐝 > beegfs-client-autobuild.conf

- 🐝 Must be edited if Mellanox OFED is used
- 🐝 Must be edited before you start the BeeGFS Client

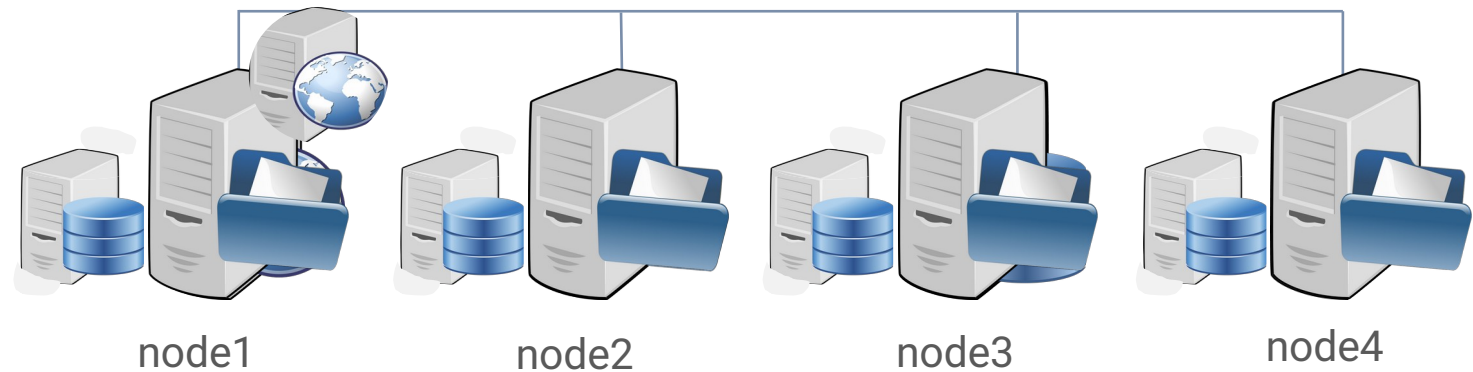


DIY: Client



🐝 Install and configure Beegfs Helperd and BeeGFS Client

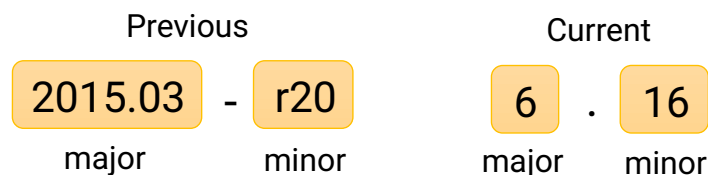
🐝 Do this on all 4 nodes of the cluster



Updating, Upgrading and Versioning



Version scheme



```
# yum update beegfs-mgmt
```

```
# systemctl restart beegfs-mgmt
```

Minor release update

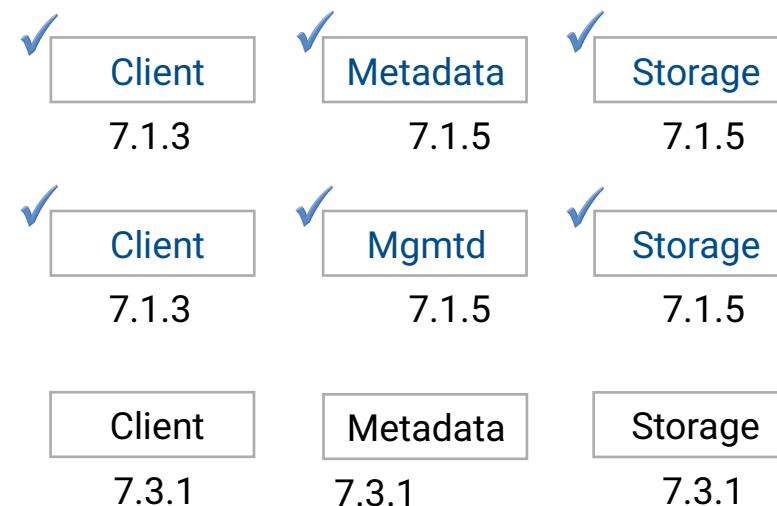
- Downtime unnecessary
- Service restart required
- Minor releases can be mixed

Major release upgrade

- Downtime needed
- Major releases cannot be mixed

Major upgrade are:

6.x	➡	7.1.x
7.1.x	➡	7.2.x
7.2.x	➡	7.3.x



Command-line Administration



🐝 Configuration files

- 🐝 Reasonable default configuration values
- 🐝 Located at /etc/beegfs/*.conf
- 🐝 All options documented

```
Server01:~ # vi /etc/beegfs/beegfs-mgmt.conf
...
storeMgmtDirectory      = /data/beegfs/mgmt
storeAllowFirstRunInit  = false
sysAllowNewServers      = true
sysAllowNewTargets      = true
...
logLevel                = 2
logNoDate               = false
logNumLines             = 50000
logNumRotatedFiles      = 5
logStdFile              = /var/log/beegfs-mgmt.log
...
# [storeMgmtDirectory]
# The absolute path and name of a directory where the file
# system can store its management data.
# Default: <none>
```

Command-line Administration



🐝 Log files

- 🐝 Located at /var/log/beegfs-*.log
- 🐝 Human-readable messages

```
client01:~ # less /var/log/beegfs-client.log
```

```
...
Apr18 10:04:34 *mount(18876) [DatagramListener (init sock)] >> Listening for UDP datagrams: Port 18004
Apr18 10:04:34 *mount(18876) [App_logInfos] >> BeeGFS Client Version: 6.9
Apr18 10:04:34 *mount(18876) [App_logInfos] >> ClientID: 49BC-58F5C892-seislab-master3
Apr18 10:04:34 *mount(18876) [App_logInfos] >> Usable NICs: ib0(RDMA) ib0(TCP) eth0(TCP)
...
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Init] >> Management node found.
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [NodeConn...] >> Connected: beegfs-mgmt@192.168.74.1:8008 (TCP)
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Sync] >> Nodes added: 4 (Type: beegfs-meta)
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Sync] >> Nodes added: 10 (Type: beegfs-storage)
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Init] >> Node registration...
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Registration] >> Node registration successful.
Apr18 10:04:34 *beegfs_XNodeSyn(18878) [Init] >> Init complete.
```

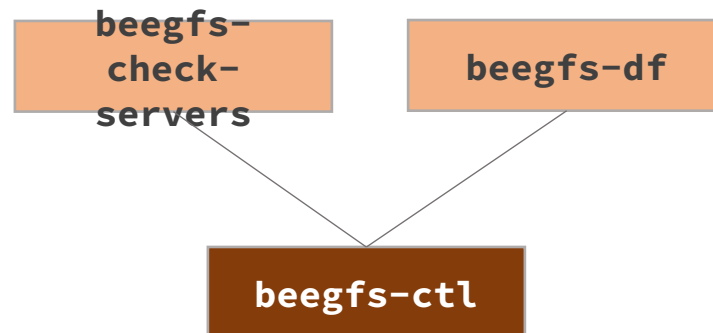
Command-line Administration



🐝 Command beegfs-ctl

- 🐝 General command-line administration tool
- 🐝 Part of the beegfs-utils package
- 🐝 Usually installed on a client node
- 🐝 Some helpers just call beegfs-ctl with special arguments for convenience

```
# yum install beegfs-utils
```



Command beegfs-ctl



```
client01:~ # beegfs-ctl --help
BeeGFS Command-Line Control Tool (http://www.beegfs.com)
```

GENERAL USAGE:

```
$ beegfs-ctl --<modename> --help
$ beegfs-ctl --<modename> [mode_arguments] [client_arguments]
```

Modes:

--listnodes	=> List registered clients and servers.
--listtargets	=> List metadata and storage targets.
--removenode (*)	=> Remove (unregister) a node.
--removetarget (*)	=> Remove (unregister) a storage target.
--getentryinfo	=> Show file system entry details.
--find	=> Find files located on certain servers.
--migrate	=> Migrate files to other storage servers.
--serverstats	=> Show server IO statistics.
--clientstats	=> Show client IO statistics.
--userstats	=> Show user IO statistics.
--storagebench (*)	=> Run a storage targets benchmark.
--getquota	=> Show quota information for users or groups.
--setquota (*)	=> Sets the quota limits for users or groups.
--listmirrorgroups	=> List mirror buddy groups.
--addmirrorgroup (*)	=> Add a mirror buddy group.

*) Marked modes require root privileges.

Command beegfs-ctl Contextual Help



```
client01:~ # beegfs-ctl --listnodes --help
```

GENERAL USAGE:

```
$ beegfs-ctl --<modename> [mode_arguments] [client_arguments]
```

CLIENT ARGUMENTS:

Optional:

```
--mount=<path>
```

Use config settings from this BeeGFS mountpoint

```
--logEnabled
```

Enable detailed logging.

...

MODE ARGUMENTS:

Mandatory:

```
--nodetype=<nodetype> The node type (management, metadata, storage, client).
```

Optional:

```
--details
```

Print additional node details, such as network ports and interface order.

```
--nicdetails
```

Print additional network interconnect details, such as IP address of each node interface.

...

USAGE:

This mode queries the management daemon for information about registered clients and servers.

Example: List registered storage nodes with their network interface order

```
$ beegfs-ctl --listnodes --nodetype=storage --details
```

Listing Registered Nodes



```
client01:~ # beegfs-ctl --listnodes --nodetype=storage
```

```
server03 [ID: 3]
```

```
server04 [ID: 4]
```

```
client01:~ # beegfs-ctl --listnodes --nodetype=storage --nicdetails
```

```
server03 [ID: 3]
```

```
Ports: UDP: 8003; TCP: 8003
```

```
Interfaces:
```

```
+ ib0[ip addr: 10.12.20.3; type: RDMA]
```

```
+ ib0[ip addr: 10.12.20.3; type: TCP]
```

```
+ eth0[ip addr: 10.10.20.3; type: TCP]
```

```
server04 [ID: 4]
```

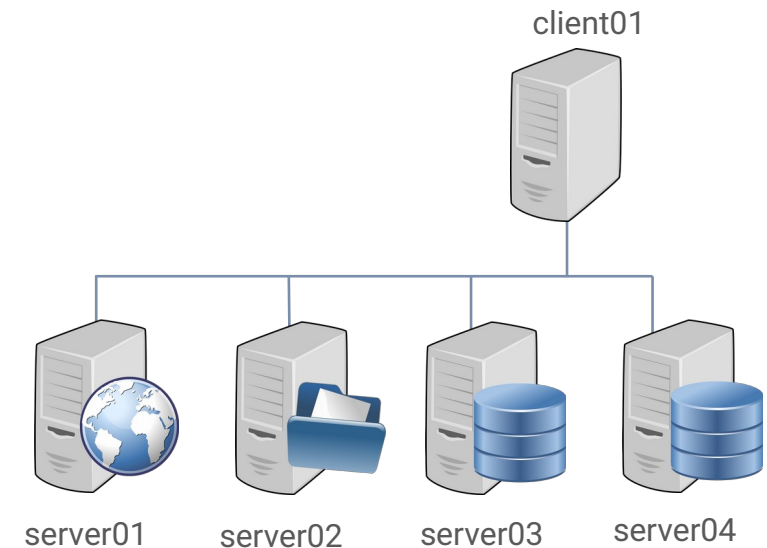
```
Ports: UDP: 8003; TCP: 8003
```

```
Interfaces:
```

```
+ ib0[ip addr: 10.12.20.4; type: RDMA]
```

```
+ ib0[ip addr: 10.12.20.4; type: TCP]
```

```
+ eth1[ip addr: 10.10.20.4; type: TCP]
```



Checking Servers



```
client01:~ # beegfs-check-servers
```

Management

```
=====
```

```
server01 [ID: 1]: reachable at 10.12.20.3:8008 (protocol: TCP)
```

Metadata

```
=====
```

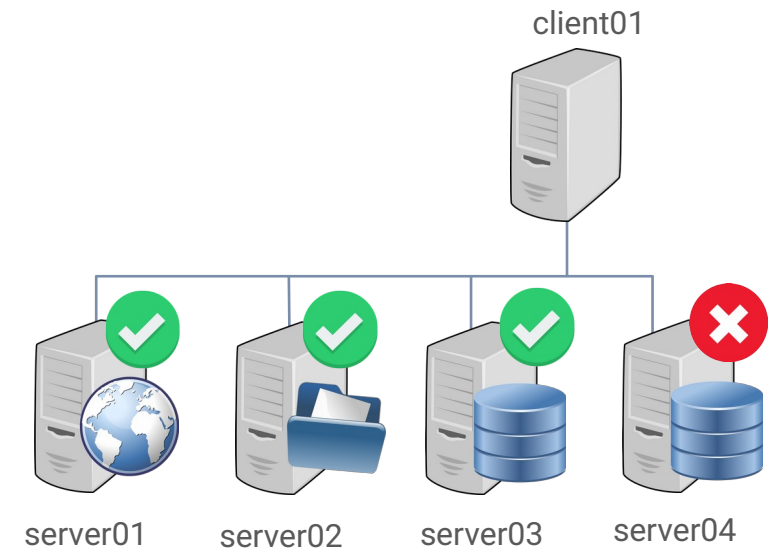
```
server02 [ID: 2]: reachable at 10.12.20.4:8005 (protocol: RDMA)
```

Storage

```
=====
```

```
server03 [ID: 3]: reachable at 10.12.20.3:8003 (protocol: RDMA)
```

```
server04 [ID: 4]: UNREACHABLE
```



Checking Servers



```
client01:~ # beegfs-net
```

TCP is normal for management service

```
mgmt_nodes
```

```
=====
```

```
server01 [ID: 1]
```

```
Connections: TCP: 1 (10.12.20.1:8008);
```

```
storage_nodes
```

```
=====
```

```
server03 [ID: 3]
```

```
Connections: RDMA: 3 (10.12.20.3:8003);
```

```
server04 [ID: 4]
```

```
Connections: RDMA: 3 (10.12.20.4:8003);
```

```
meta_nodes
```

```
=====
```

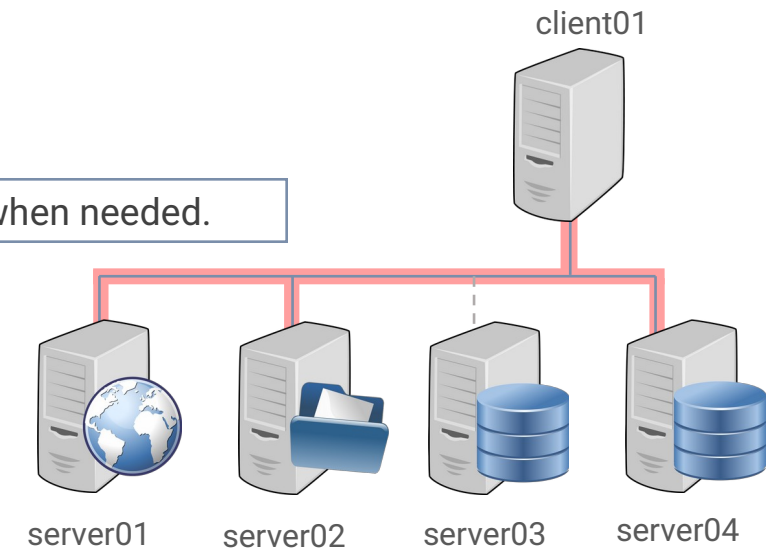
```
server02 [ID: 2]
```

```
Connections: RDMA: 1 (10.12.20.2:8005);
```

```
server03 [ID: 3]
```

```
Connections: <none>
```

Connections established only when needed.

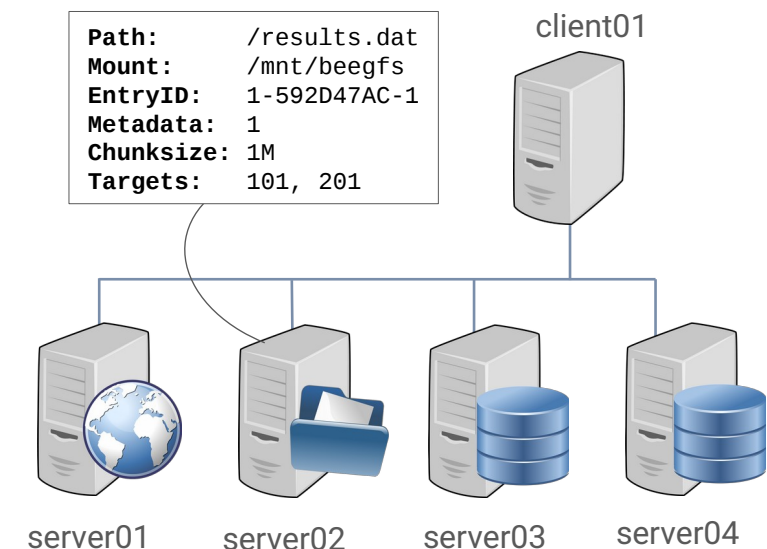


Getting Information About Directory Entry



```
client01:~ # beegfs-ctl --getentryinfo --verbose /mnt/beegfs/results.dat
```

```
Path: /results.dat
Mount: /mnt/beegfs
EntryID: 1-592D47AC-1
Metadata node: server02 [ID: 1]
Stripe pattern details:
+ Chunksize: 1M
+ Number of storage targets: desired: 4; actual: 2
+ Storage targets:
  + 101 @ server03 [ID: 1]
  + 201 @ server04 [ID: 2]
```



Checking Free Space



```
client01:~ # beegfs-df
```

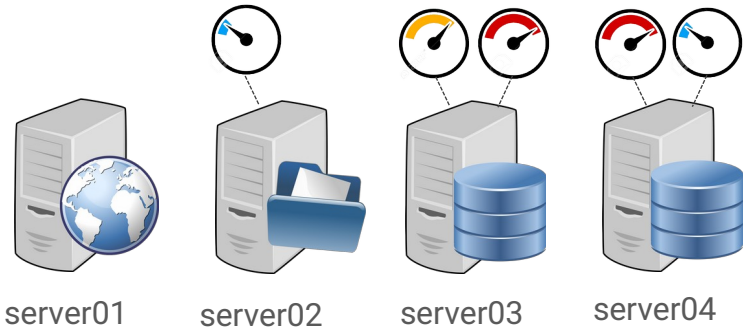
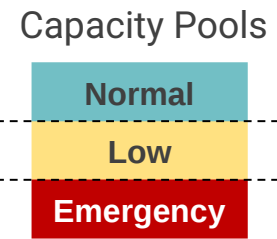
METADATA SERVERS:

TargetID	Capacity Pool	Total	Free	InodesTotal	InodesFree
=====	=====	=====	=====	=====	=====
2	[normal]	240.0GB	230.0GB	158.8M	148.7M

STORAGE TARGETS:

TargetID	Capacity Pool	Total	Free	InodesTotal	InodesFree
=====	=====	=====	=====	=====	=====
301	[emergency]	9168.7GB	4.5GB	582.2M	87.2M
302	[low]	9168.7GB	102.5GB	582.2M	42.0M
401	[emergency]	9168.7GB	2.5GB	582.2M	75.2M
402	[normal]	9168.7GB	2112.5GB	582.2M	92.2M

tuneStorageSpaceLowLimit = 200G
tuneStorageSpaceEmergencyLimit = 5G



DIY: beegfs-utils



- ✚ Populate the system
 - ✚ Create directories, subdirectories
 - ✚ Create files of different sizes
 - ✚ Create symbolic links
- ✚ Test the basic commands
 - ✚ Check if all servers are reachable
 - ✚ Check if all clients are reachable
 - ✚ List all metadata nodes and discover which version of BeeGFS each one is using
 - ✚ List all storage targets
 - ✚ Check which storage target has more data at the moment
 - ✚ Etc

DIY: beegfs-utils



- 🐝 Check the BeeGFS System
 - 🐝 List registered Nodes
 - 🐝 Check Servers
 - 🐝 Check Network
- 🐝 Check the BeeGFS File System
 - 🐝 Get directory information
 - 🐝 Check free space

File System Consistency Check and Repair



```
client01:~ # beegfs-fsck --checkfs --readonly
```

```
-----  
Started BeeGFS fsck in forward check mode [Mon Sep 29 21:47:57 2015]  
Log will be written to /var/log/beegfs-fsck.log  
Database will be saved as /var/lib/beegfs/beegfs-fsck.db  
-----
```

Step 1: Check reachability of nodes: Finished

Step 2: Gather data from nodes:

Fetch data > Directory entries: 504598 | Inodes: 504598 | Chunks: 904831

Step 3: Check for errors...

- * Target is used, but does not exist... Finished
- * File has a missing target in stripe pattern... Finished
- * Dentry-by-ID file is present, but no corresponding dentry... Finished
- * Dentry-by-ID file is broken or missing... Finished
- * Chunk is saved in wrong path... Finished
- * Wrong owner node saved in inode... Finished
- * Dentry points to inode on wrong node... Finished
- ...

Data Striping

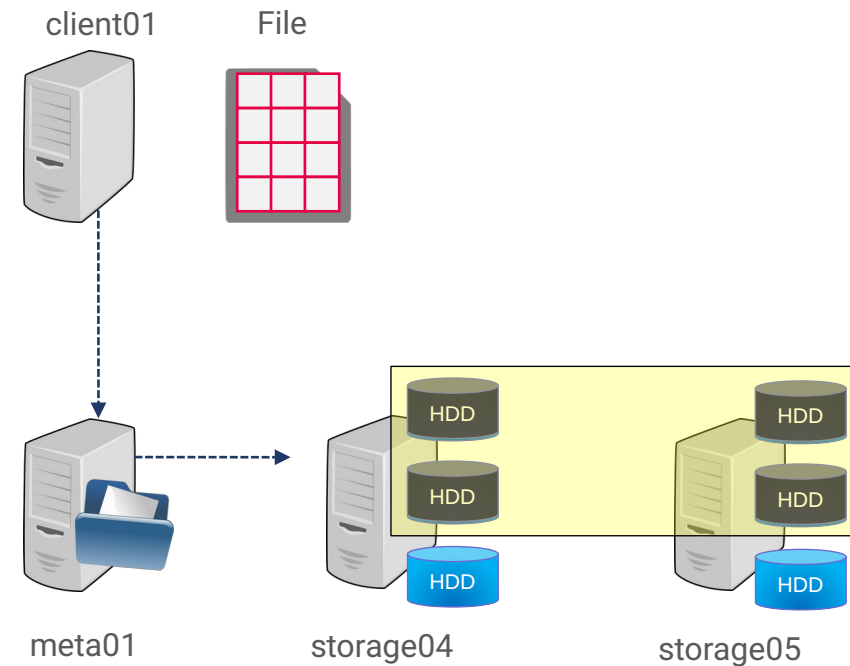


When new file is created

- Metadata service assigns up to 4 storage targets to file
- By default, data chunks have the size of 512 KB

Default settings

- Work well for most systems
- Number of targets: 4
- Chunk size: 512 KB



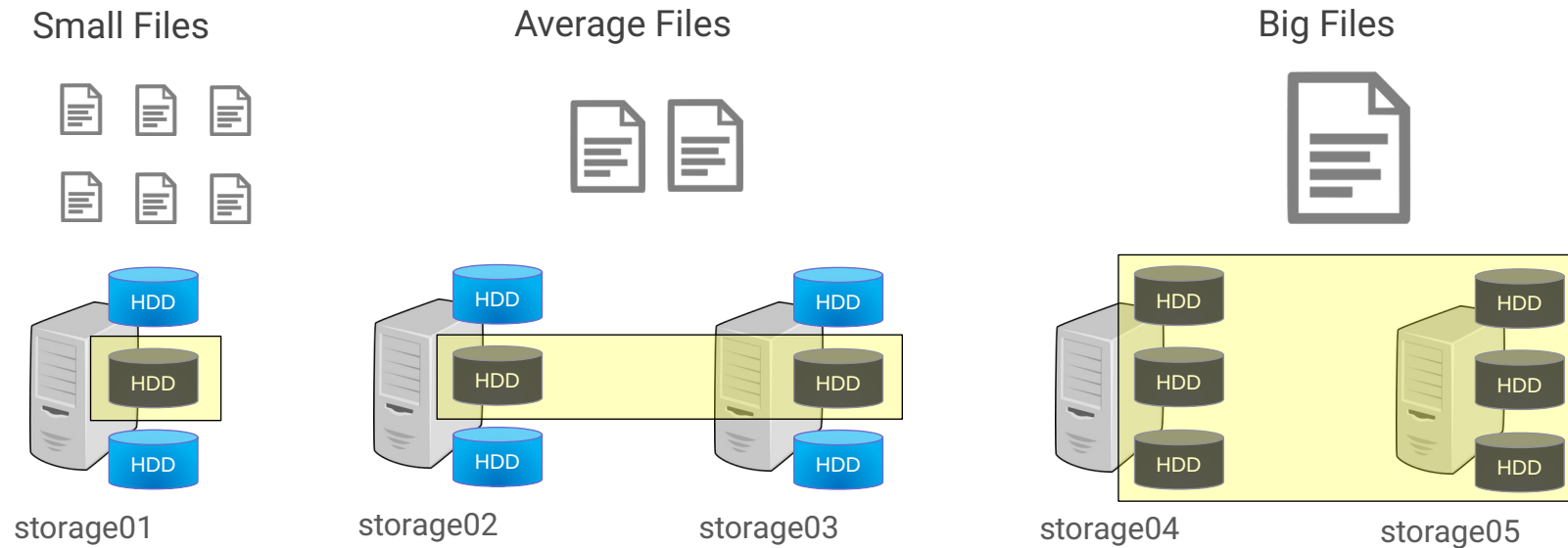
Data Striping



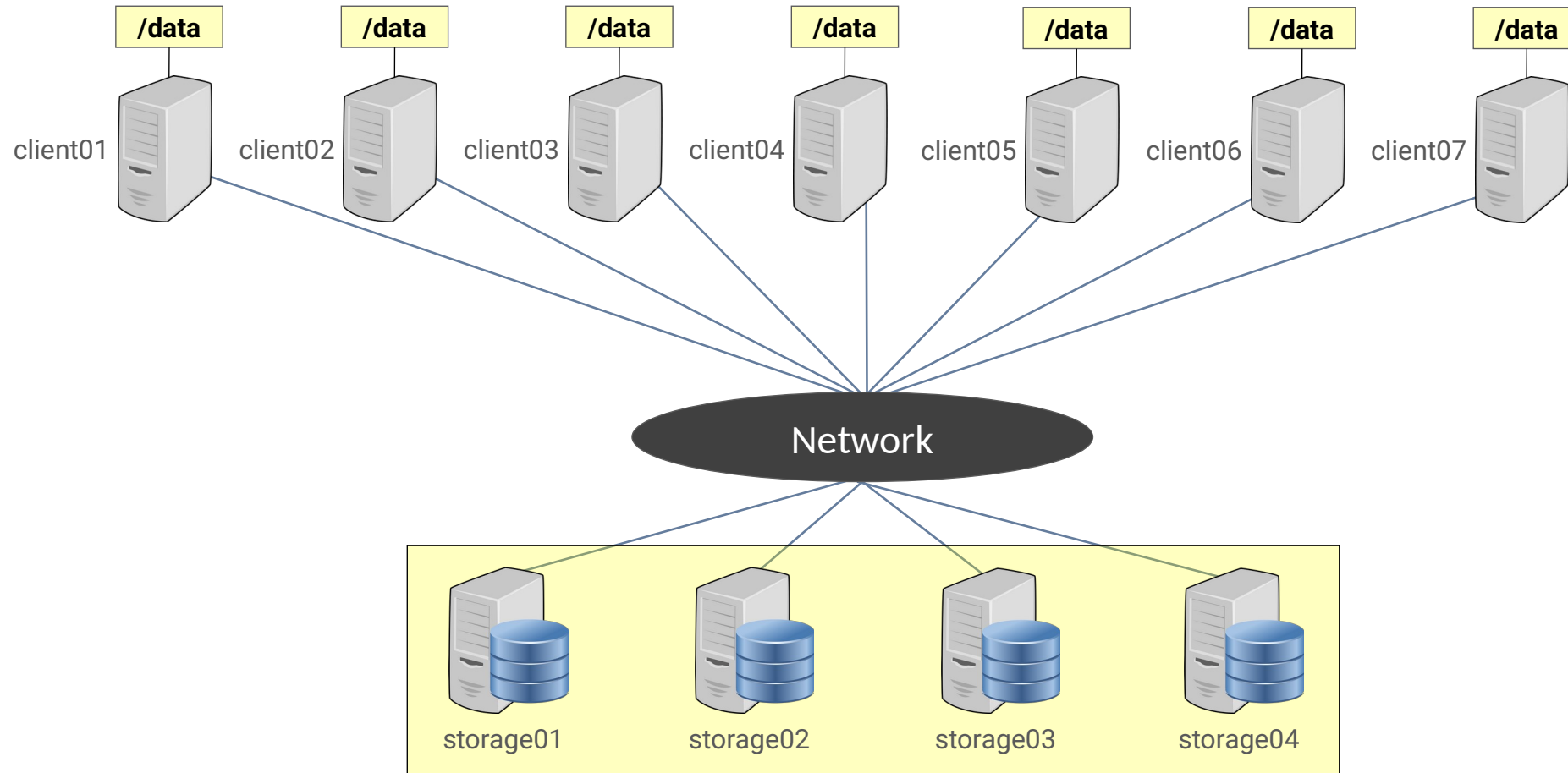
- Setting data striping pattern per path

```
client01:~ # beegfs-ctl --setpattern --chunksize=1m --numtargets=6 /data/simulations
```

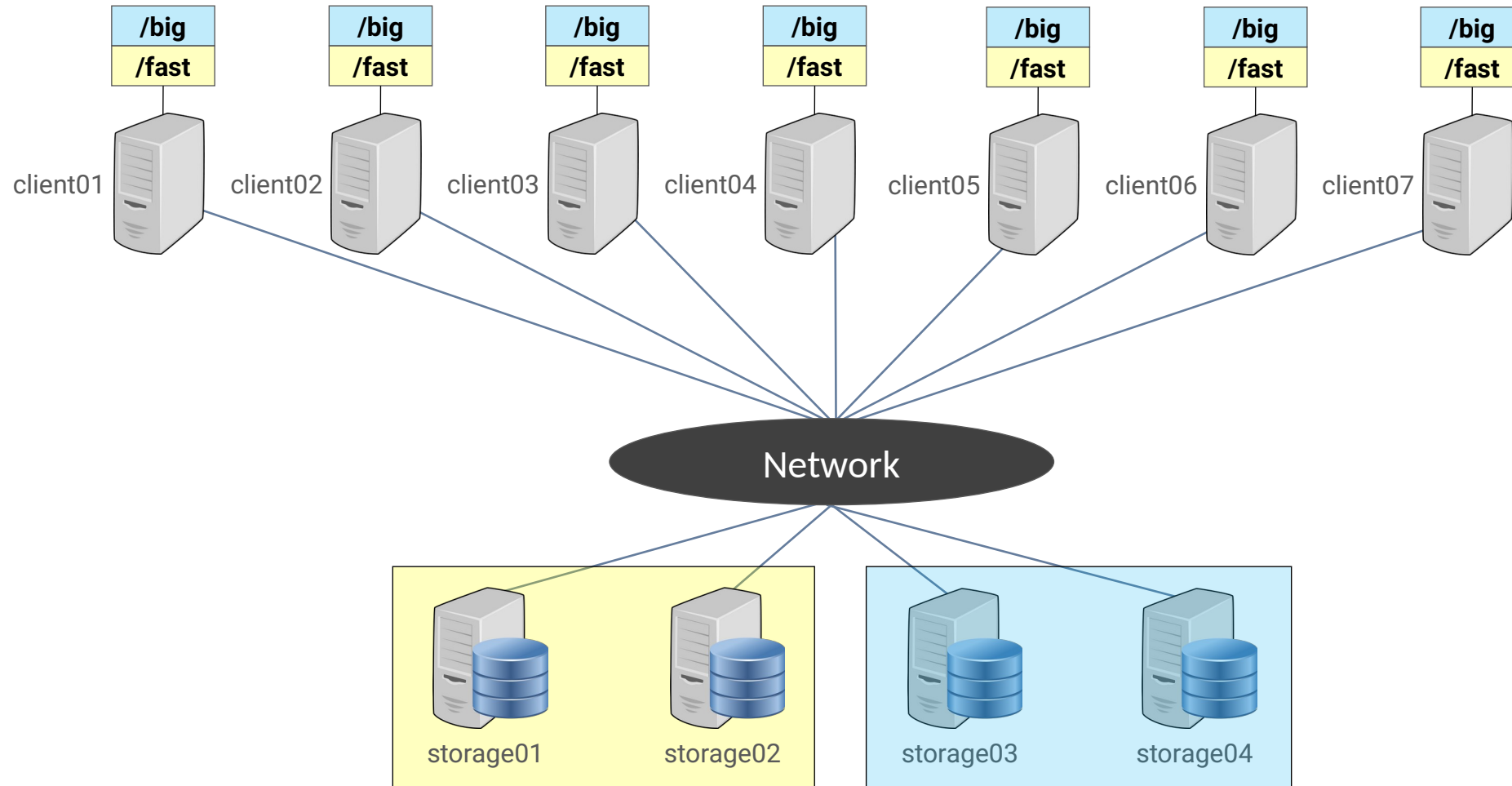
- Application can set pattern via API



Multiple BeeGFS Instances



Multiple BeeGFS Instances

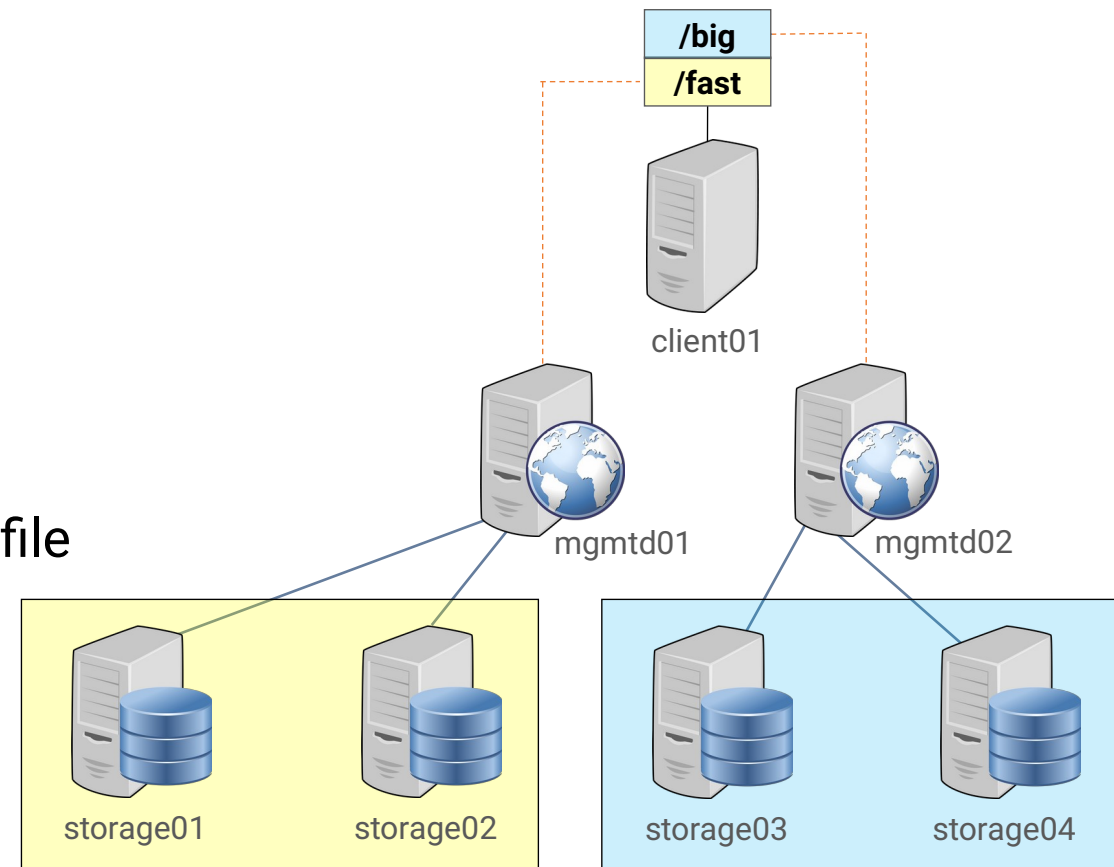


Multiple BeeGFS Instances

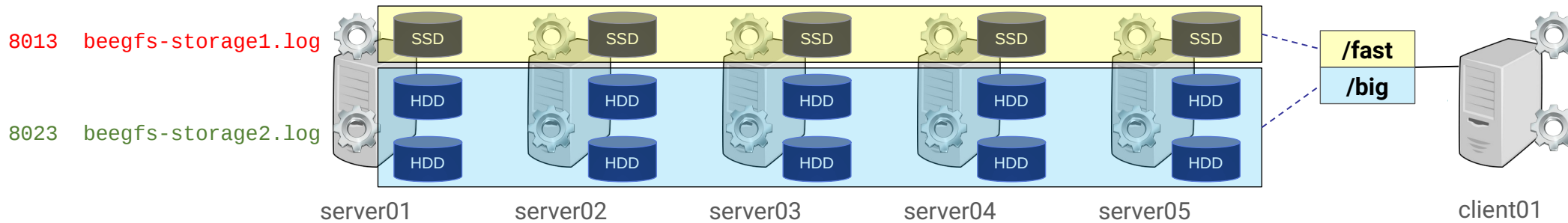


- Setup a new management service
 - Each BeeGFS instance must have its own management service
- Create copies of `/etc/beegfs/beegfs-client.conf`
 - ... with new names
 - ... pointing to a second management server
- Edit `/etc/beegfs/beegfs-mounts.conf`
 - Map each mountpoint to a different client config file

```
/fast /etc/beegfs/beegfs-client-fast.conf beegfs rw  
/big  /etc/beegfs/beegfs-client-big.conf  beegfs ro
```



Multiple BeeGFS Instances - Same Servers



- 🐝 Enable “multi-mode” in /etc/default/beegfs-...

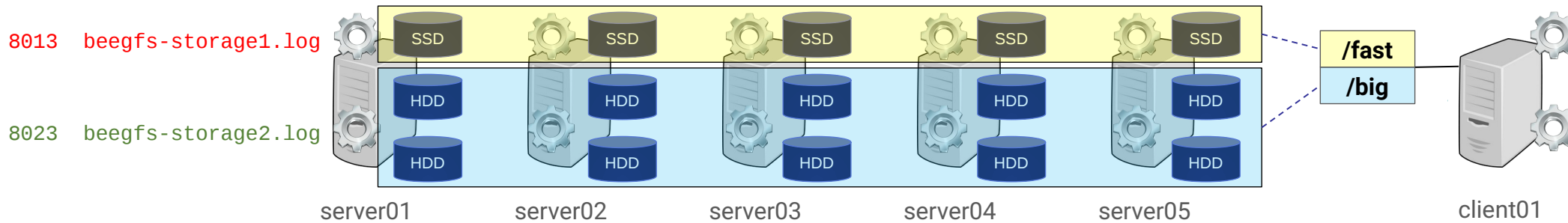
```
server01:~ # cat /etc/default/beegfs-storage  
MULTI_MODE="YES"
```

This is no longer necessary in BeeGFS > 7.1.x

- 🐝 Create a separate directory in /etc/beegfs for each instance with config files

```
/etc/beegfs/fast.d  
/etc/beegfs/big.d
```

Multiple BeeGFS Instances - Same Servers



🐝 Copy beegfs-storage.conf into directories

```
server01:~ # cp /etc/beegfs/beegfs-storage.conf /etc/beegfs/fast.d/  
server01:~ # cp /etc/beegfs/beegfs-storage.conf /etc/beegfs/big.d/
```

🐝 Edit each *.conf file

```
/etc/beegfs/fast.d/beegfs-storage.conf  
/etc/beegfs/big.d/beegfs-storage.conf
```

connStoragePortTCP and connStoragePortUDP must have different port numbers

BeeGFS-storage.conf



Example of beegfs-storage.conf

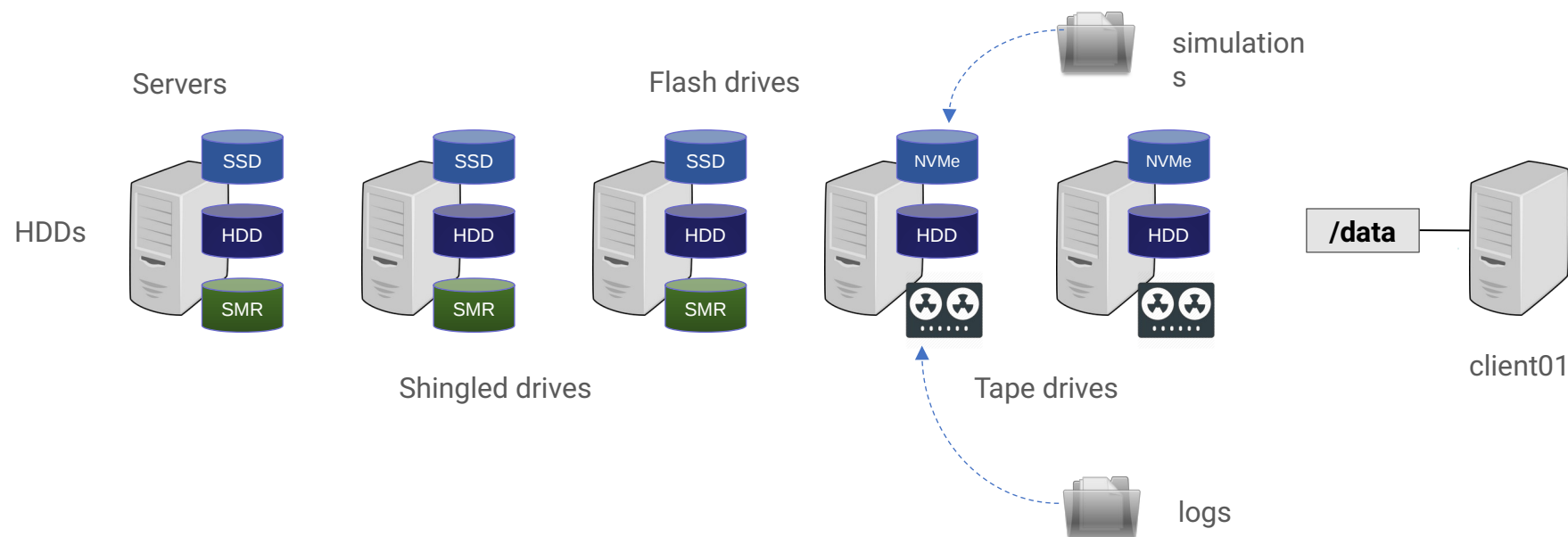
- Create two storage partitions
- Create directories stor1.d and stor2.d in /etc/beegfs
- Copy beegfs-storage.conf to /etc/beegfs/stor1.d/ and /etc/beegfs/stor2.d/
- Edit beegfs-storage.conf in both directories
 - connStoragePortTCP = 8004
 - connStoragePortUDP = 8004
 - logStdFile = /var/log/beegfs-storage-1.log
- Register partitions at Management Server
 - /opt/beegfs/sbin/beegfs-setup-storage -p <dir1> -s 31 -i 3101 -m <mgmtd> -S <nodename> -c /etc/beegfs/stor1.d/beegfs-storage.conf
 - /opt/beegfs/sbin/beegfs-setup-storage -p /<dir2> -s 32 -i 3201 -m <mgmtd> -S <diff-nodename> -c /etc/beegfs/stor2.d/beegfs-storage.conf
- Start with systemctl start beegfs-storage@stor1

DIY: Multimode

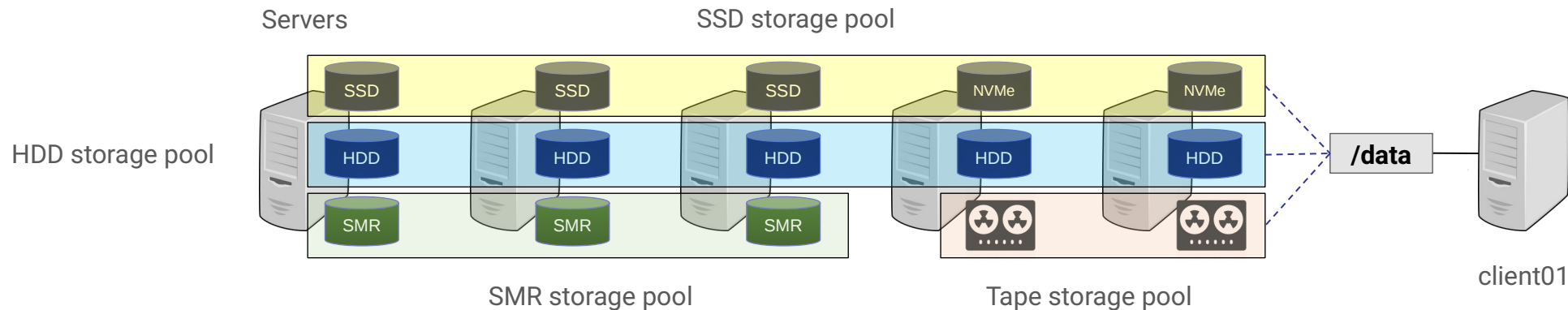


- 🐝 Create a multimode storage System
 - 🐝 Use the two Storage Servers, that aren't used yet.
 - 🐝 Create the mount points for the partitions (if not done yet)
 - 🐝 And mount the partitions
 - 🐝 Register BeeGFS Storage partitions at Management Server
 - 🐝 Start storage partitions

Storage Pools



Storage Pools



🐝 Add storage pools

```
client01:~ # beegfs-ctl --addstoragepool --desc=SSDs --targets=101,201,301,401,501 --id=2
```

```
client01:~ # beegfs-ctl --liststoragepools
```

🐝 Set data striping pattern based on pools

```
client01:~ # beegfs-ctl --setpattern --storagepoolid=2 --chunksize=1m --numtargets=4 /data/simulations
```

DIY: Storagepools



🐝 Create Storagepools on your system

Bind mounts

🐝 Enable bind mounts

- 🐝 Edit /etc/default/beegfs-client

```
client01:~ # vi /etc/default/beegfs-client
```

🐝 Copy /etc/beegfs/beegfs-client-mount-hook.example to a new name

- 🐝 Edit the new file and add the directory you want use
- 🐝 Remove the „exit 1“ line at the beginning

```
client01:~ # vi /etc/beegfs/beegfs-client-mount-hook
```

🐝 Test the bind mount

🐝 Restart the client and check that the bind mount is done automatically



Limit Traffic to Specific Network Interfaces



Limit incoming connections

- In the daemon configuration file at `/etc/beegfs/beegfs-*.conf`
 - `connInterfacesFile`: Registered interfaces for incoming connections

```
client01:~ # beegfs-ctl --listnodes --nodetype=storage --nicdetails
```

```
storage01 [ID: 1]
```

```
Ports: UDP: 8003; TCP: 8003
```

```
Interfaces:
```

```
+ ib1[ip addr: 10.12.20.3; type: RDMA]
```

```
+ ib1[ip addr: 10.12.20.3; type: TCP]
```

```
+ ib0[ip addr: 10.12.21.5; type: RDMA]
```

```
+ ib0[ip addr: 10.12.21.5; type: TCP]
```

```
storage01 [ID: 2]
```

```
Ports: UDP: 8003; TCP: 8003
```

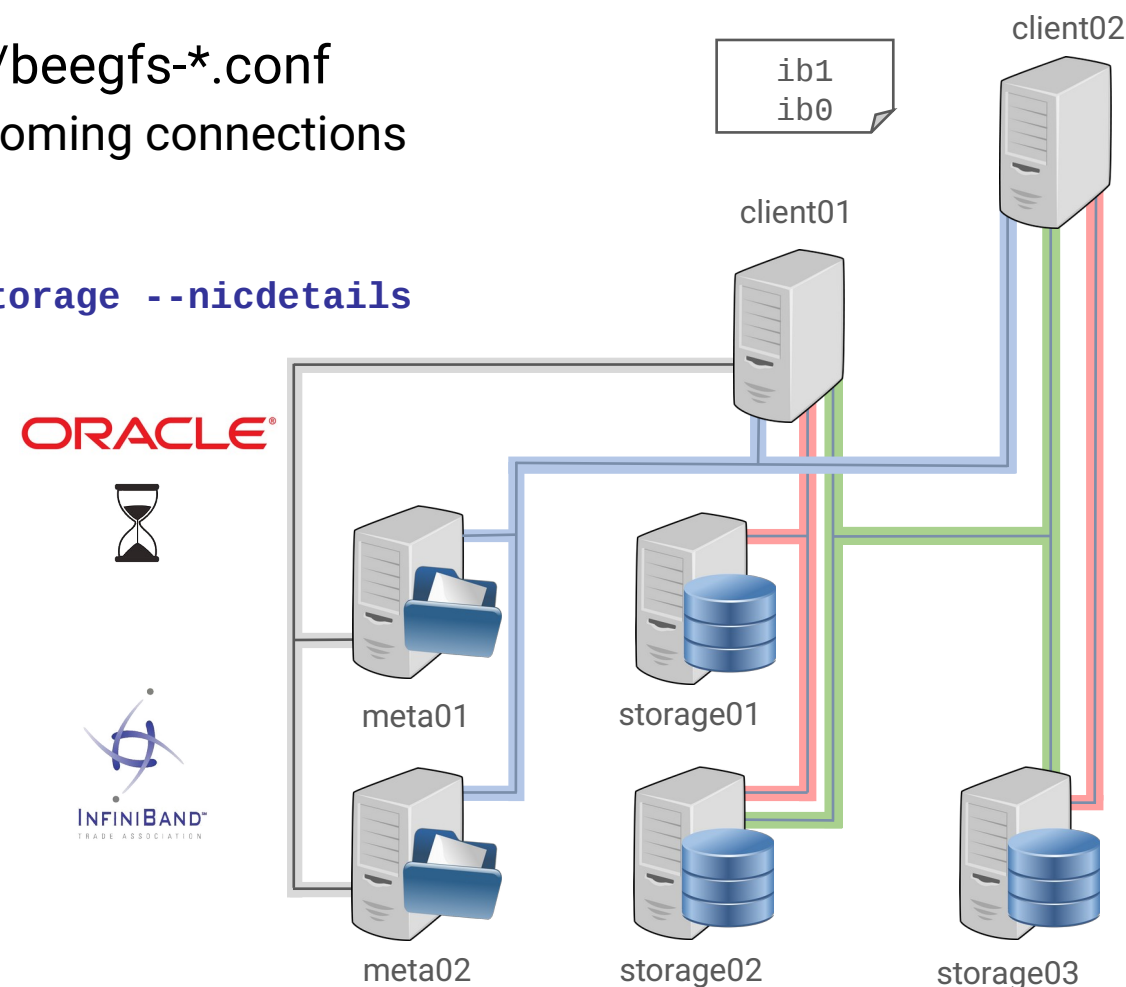
```
Interfaces:
```

```
+ ib1[ip addr: 10.12.20.4; type: RDMA]
```

```
+ ib1[ip addr: 10.12.20.4; type: TCP]
```

```
+ ib0[ip addr: 10.12.21.6; type: RDMA]
```

```
+ ib0[ip addr: 10.12.21.6; type: TCP]
```



Limit Traffic to Specific Networks



- Limit outgoing connections to certain IP address ranges
 - In the daemon configuration file at `/etc/beegfs/beegfs-*.conf`
 - `connNetFilterFile`: Limit outgoing connections to certain IP address ranges

```
client01:~ # beegfs-ctl --listnodes --nodetype=client --nicdetails
```

```
14E8-53E4DE10-client01
```

```
Ports: UDP: 18004; TCP: 0
```

```
Interfaces:
```

```
+ ib1[ip addr: 10.12.20.201; type: RDMA]
```

```
+ ib2[ip addr: 10.12.29.200; type: RDMA]
```

```
...
```

```
15E7-33D4DE20-client02
```

```
Ports: UDP: 18004; TCP: 0
```

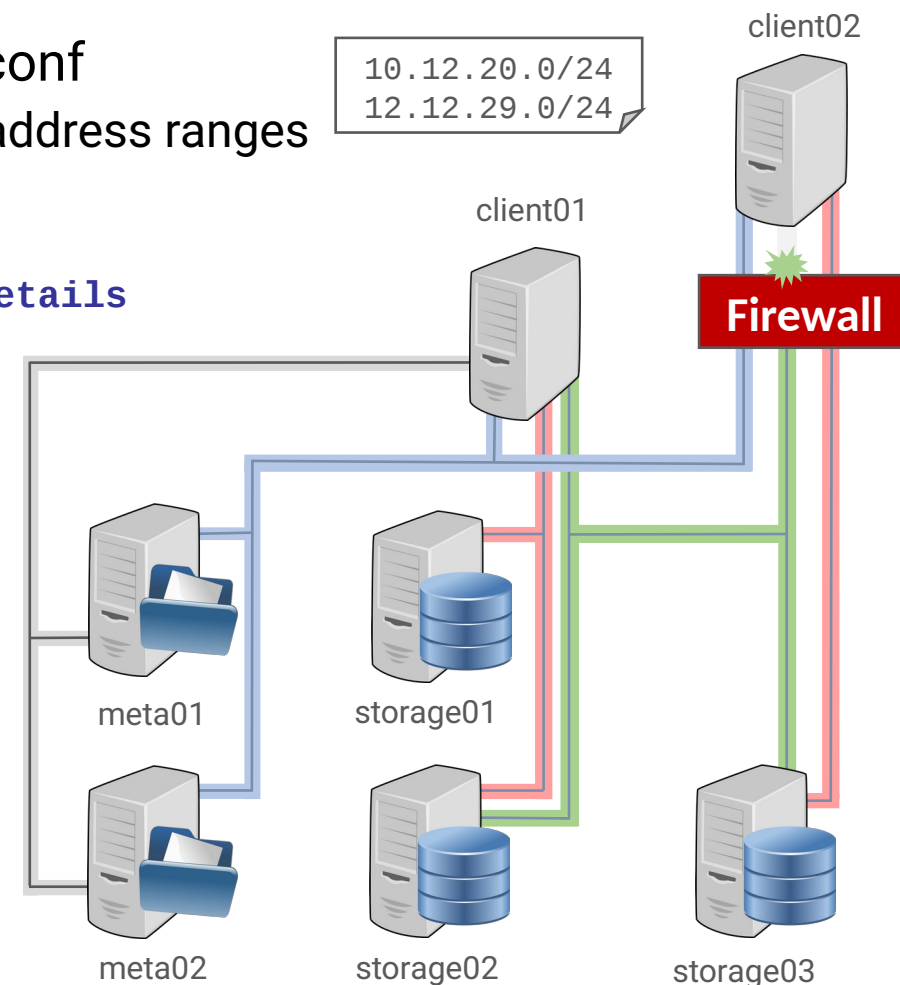
```
Interfaces:
```

```
+ ib0[ip addr: 10.12.21.100; type: RDMA]
```

```
+ ib1[ip addr: 10.12.20.202; type: RDMA]
```

```
+ ib2[ip addr: 10.12.29.202; type: RDMA]
```

```
...
```



Enhance security with connAuthFile

- 🐝 Mandatory since BeeGFS 7.2.7 (CentOS 7) and 7.3.1
 - 🐝 In the daemon configuration file at `/etc/beegfs/beegfs-*.conf`
 - 🐝 `connAuthFile`: Limit connections to those services who have a `connAuthFile`
 - 🐝 BeeGFS Services will not start without a authentication file

```
client01:~ # less /etc/beegfs/beegfs-client.conf
```

```
#  
# --- Section 1.2: [Advanced Settings] ---  
#
```

```
connAuthFile          =  
connDisableAuthentication = false
```

```
client01:~ # vi /etc/beegfs/AuthFile
```

```
#  
# Authentication file for BeeGFS  
#
```

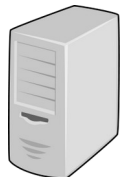
```
abc123
```



metadata



storage



client

Getting Server Statistics



```
client01:~ # beegfs-ctl --serverstats --nodetype=storage --interval=1 --perserver
```

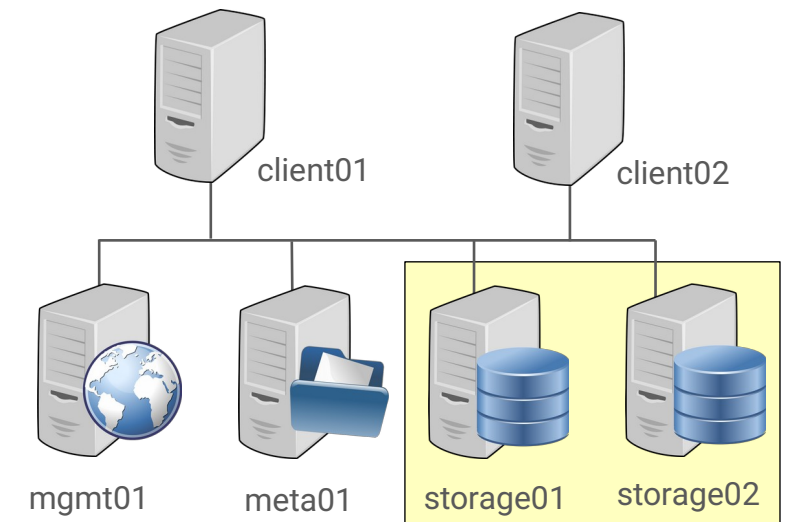
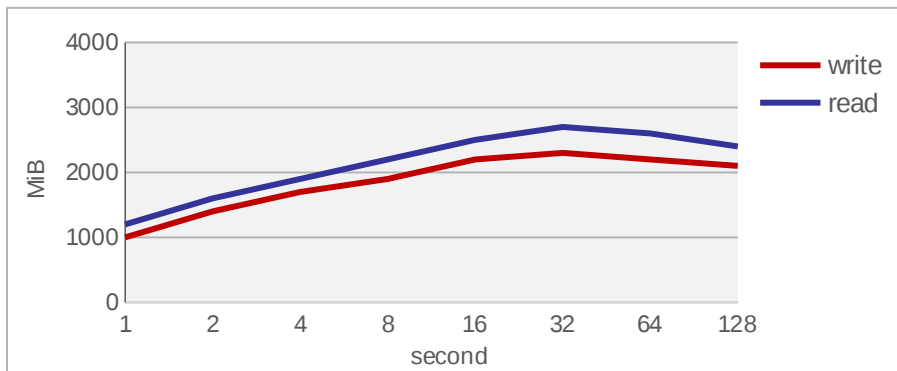
```
=== time index: 1412011239 (values show last second only)
```

nodeID	write_KiB	read_KiB	reqs	qlen	bsy
1	1436430	5633	18098	0	0
2	1536430	14430	18386	0	0

```
=== time index: 1412011240 (values show last second only)
```

nodeID	write_KiB	read_KiB	reqs	qlen	bsy
1	1404430	9633	15613	0	0
2	1336330	12403	17274	0	0

...



Getting Client Statistics



```
client01:~ # beegfs-ctl --clientstats --nodetype=storage --interval=5 --names
```

```
===== 5 s =====
```

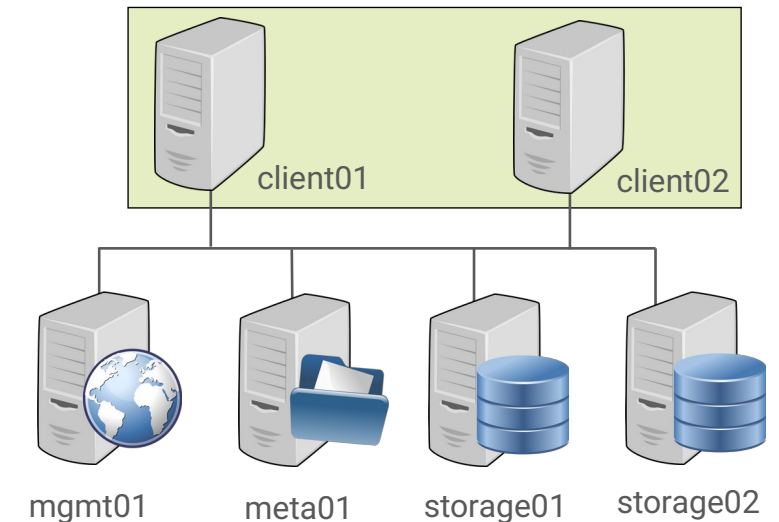
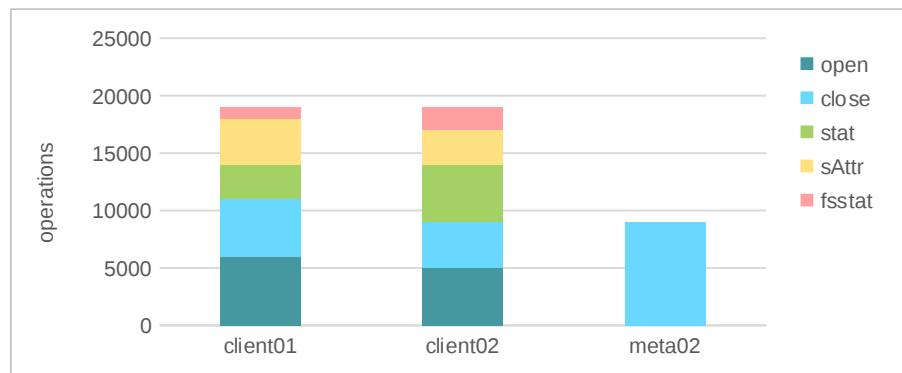
```
Sum          22967 [sum] 4720 [close] 65 [sAttr] 5706 [open] 2456 [stat] ...
client01      2847 [sum] 700 [close] 700 [open] 1447 [stat] ...
client02      2847 [sum] 700 [close] 700 [open] 1447 [stat] ...
meta01        1060 [sum] 1060 [close] ...
```

```
...
```

```
===== 10 s =====
```

```
Sum:          28138 [sum] 6511 [close] 1800 [open] 1804 [stat] 4 [sAttr] ...
client01      5803 [sum] 1403 [close] 1400 [open] ...
client02      5803 [sum] 1403 [close] 1400 [open] ...
meta01        2454 [sum] 2454 [close] ...
```

```
...
```



Getting User Statistics



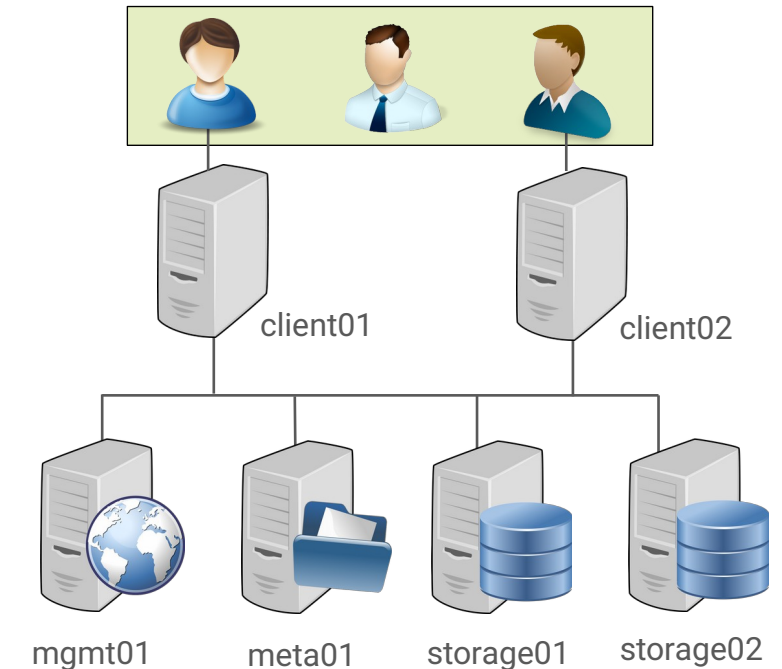
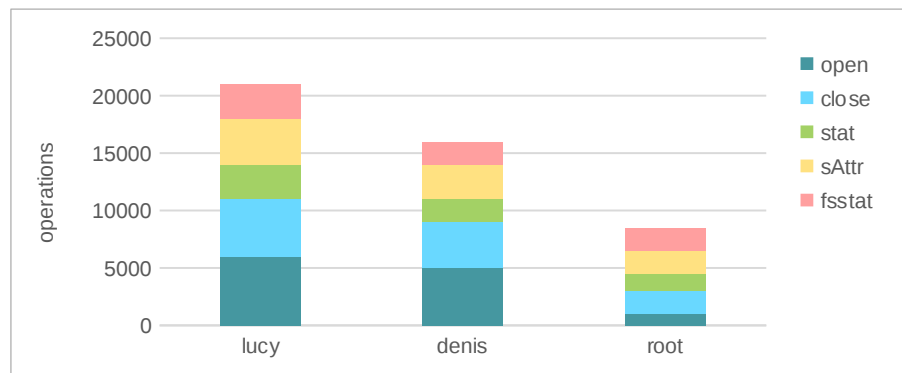
```
client01:~ # beegfs-ctl --userstats --nodetype=meta --interval=5 --names
```

```
===== 5 s =====
```

Sum:	60	[sum]	10	[close]	10	[open]	1	[sAttr]	6	[unlnk]	...
lucy	36	[sum]	3	[close]	3	[open]	4	[stat]	1	[unlnk]	...
root	12	[sum]	6	[stat]	4	[unlnk]	...				
denis	8	[sum]	1	[sAttr]	3	[stat]	...				

```
===== 10 s =====
```

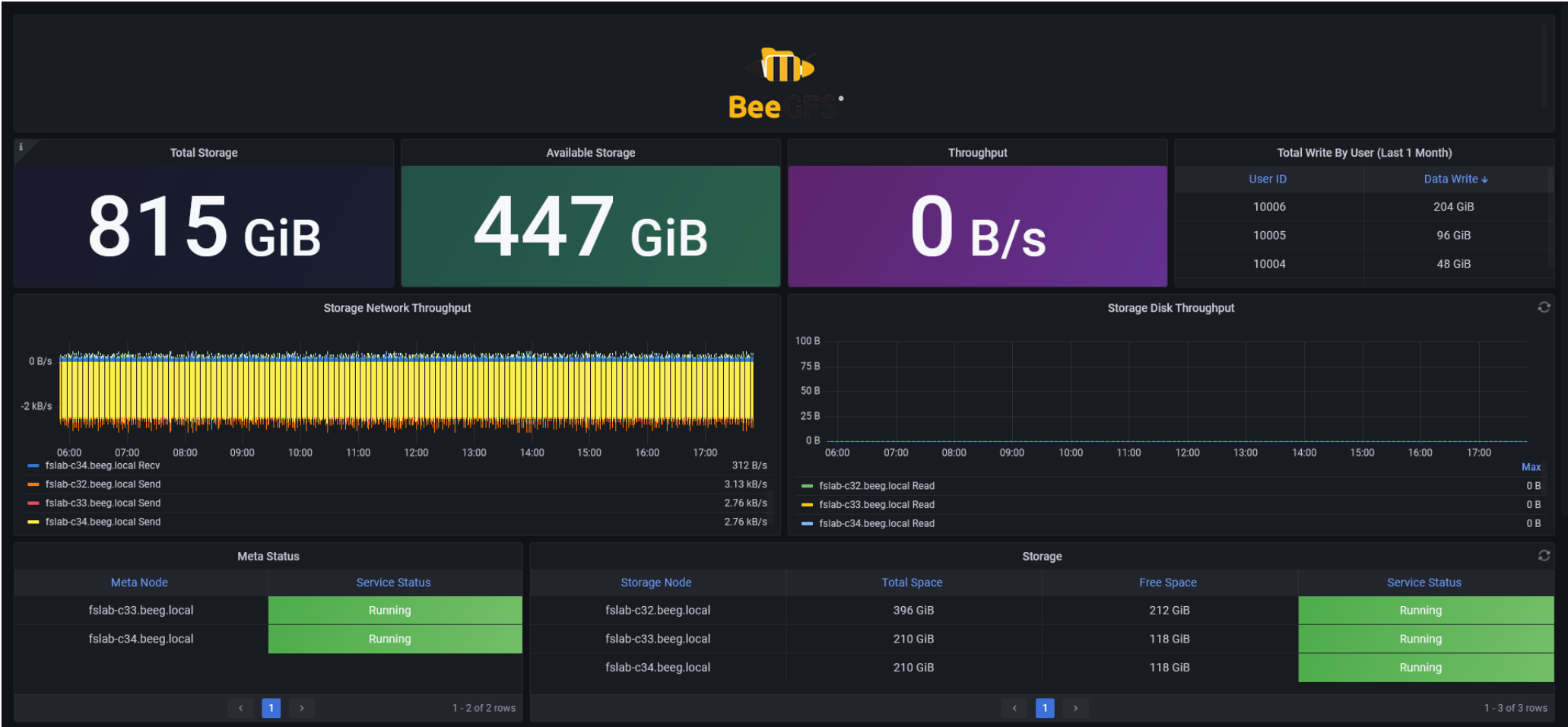
Sum:	2314	[sum]	24	[close]	1	[create]	...
lucy	2029	[sum]	23	[close]	23	[open]	...
denis	16	[sum]	1	[close]	1	[create]	...
root	8	[sum]	4	[stat]	...		



Monitoring GUI (old version)



Monitoring GUI (new version, available with BeeGFS 7.3.4)



Installation Monitoring: 3rd party software



Grafana

Setup grafana repo

```
server01:~ # cat << EOF | sudo tee /etc/yum.repos.d/grafana.repo
[grafana]
baseurl = https://packages.grafana.com/oss/rpm
repo_gpgcheck = 1
enabled = 1
gpgcheck = 1
gpgkey = https://packages.grafana.com/gpg.key
EOF
```

```
server01:~ # yum install grafana
```

```
server01:~ # systemctl start grafana-server
```



server01

Installation Monitoring: 3rd party software



Influxdb

Setup influxdb repository

```
server01:~ # cat <<EOF | sudo tee /etc/yum.repos.d/influxdb.repo  
[influxdb]  
name = InfluxDB Repository - RHEL \"$releasever  
baseurl = https://repos.influxdata.com/stable/x86_64/main  
enabled = 1  
gpgcheck = 1  
gpgkey = https://repos.influxdata.com/influxdata-archive_compat.key  
EOF
```

```
server01:~ # yum install influxdb2 influxdb2-cli
```

```
server01:~ # systemctl start influxdb2
```



server01

Installation Monitoring: 3rd party software

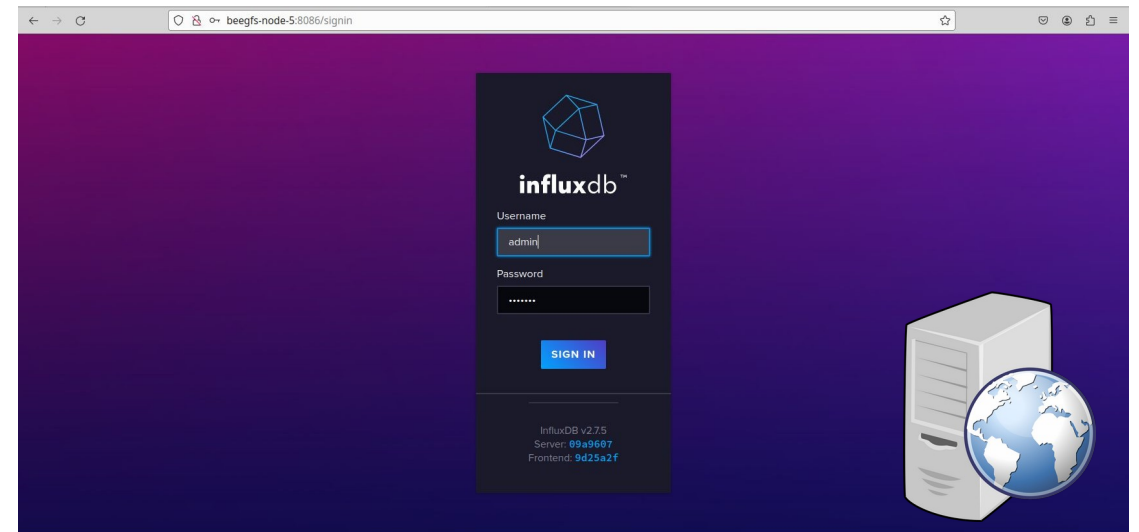
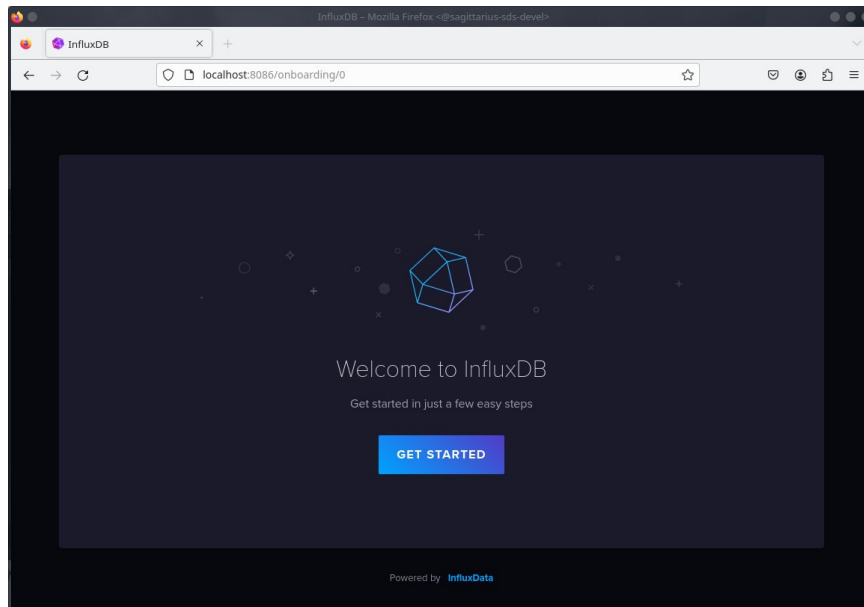


🐝 Configure Influxdb

🐝 This is for Influxdb2

Influxdb2 needs a token and a bucket

Both can be declared via command line or the web interface of influxdb



server01

Installation Monitoring: 3rd party software

🐝 **Telegraf is part of the influxdb repository**

```
server01:~ # yum install telegraf
```

```
server01:~ # vi /etc/telegraf/telegraf.d/beegfs_mon_telegraf.conf
```

```
[[outputs.influxdb_v2]]
urls = ["http://beegfs-node-5:8086"] # Replace with the actual InfluxDB URL
token = "" # Replace with your InfluxDB 2.x token
organization = "" # Replace with your InfluxDB 2.x organization
bucket = "" # Replace with your InfluxDB 2.x bucket

[[inputs.cpu]]
percpu = true
totalcpu = true
collect_cpu_time = false
report_active = false
core_tags = false

[[inputs.disk]]
ignore_fs = ["tmpfs", "devtmpfs", "devfs", "iso9660", "overlay", "aufs", "squashfs"]

[[inputs.diskio]]
[[inputs.mem]]
[[inputs.processes]]
[[inputs.system]]
[[inputs.procstat]]
systemd_unit = "beegfs-meta.service"
[[inputs.procstat]]
systemd_unit = "beegfs-storage.service"
```



server01

Installation Monitoring: BeeGFS Software



BeeGFS

```
server01:~ # yum install beegfs-mon
```

```
server01:~ # yum install beegfs-mon-grafana
```

```
server01:~ # configure /etc/beegfs/beegfs-mon.conf
```

```
Server01:~ # vi /etc/beegfs/beegfs-mon.conf
```

```
sysMgmtHost          = <hostname of BeeGFS Mgmt Server>
```



server01

Installation Monitoring: BeeGFS Software



BeeGFS mon.conf

```
Server01:~ # vi /etc/beegfs/beegfs-mon.conf
```

```
sysMgmtHost          = <hostname of BeeGFS Mgmt Server>
dbType                = influxdb2
dbHostName            = <influx database host>
dbHostPort            = 8086
dbAuthFile            = /etc/beegfs/beegfs-mon.auth

# used by influxdb V2 only

dbBucket              = beegfs
```

```
server01:~ # systemctl start beegfs-mon
```



server01

Installation Monitoring: BeeGFS Software



BeeGFS mon.auth

```
Server01:~ # vi /etc/beegfs/beegfs-mon.auth
```

```
# This file configures the credentials needed to connect to your monitoring database instance.
```

```
# This currently only works with InfluxDB.
```

```
username = admin
```

```
password = admin0123
```

```
# used by influxdb V2 only
```

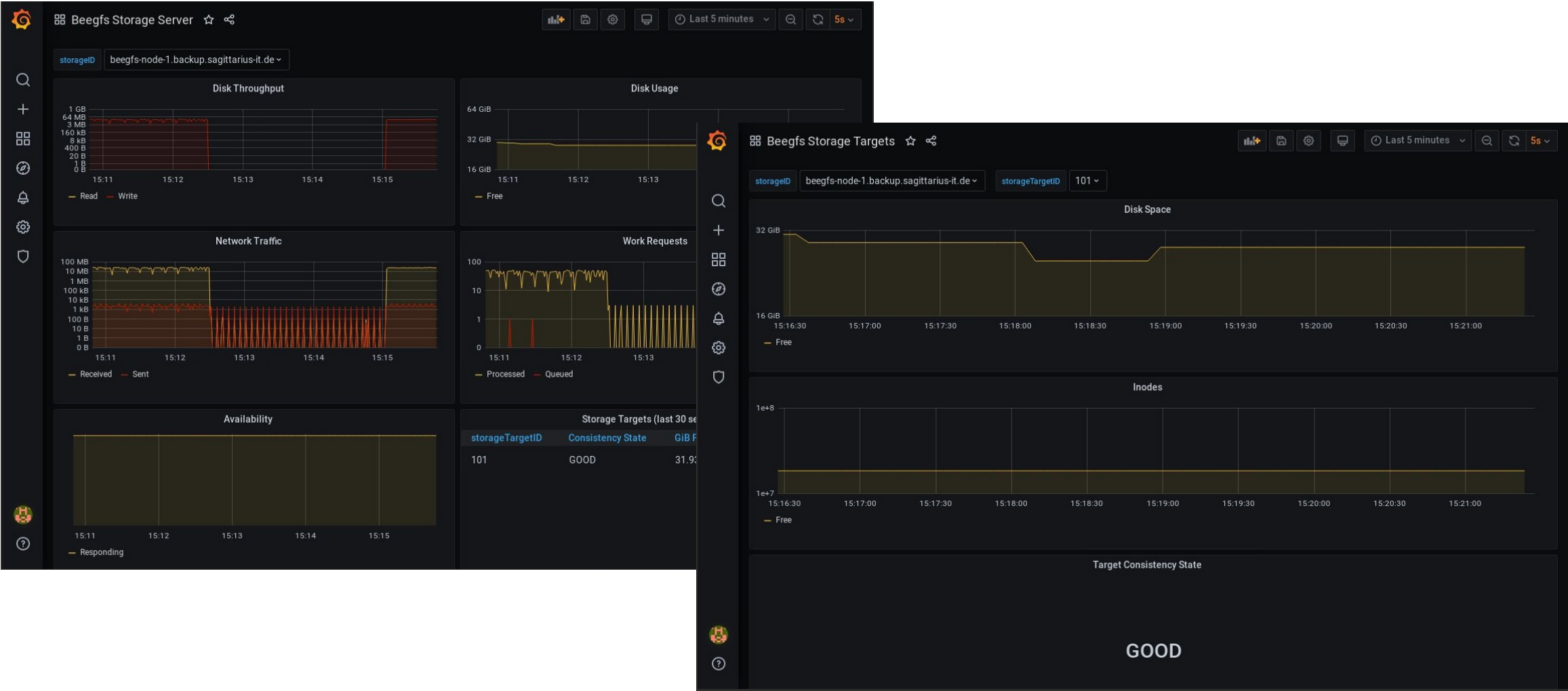
```
organization = <your organisation>
```

```
token = <your admin token>
```

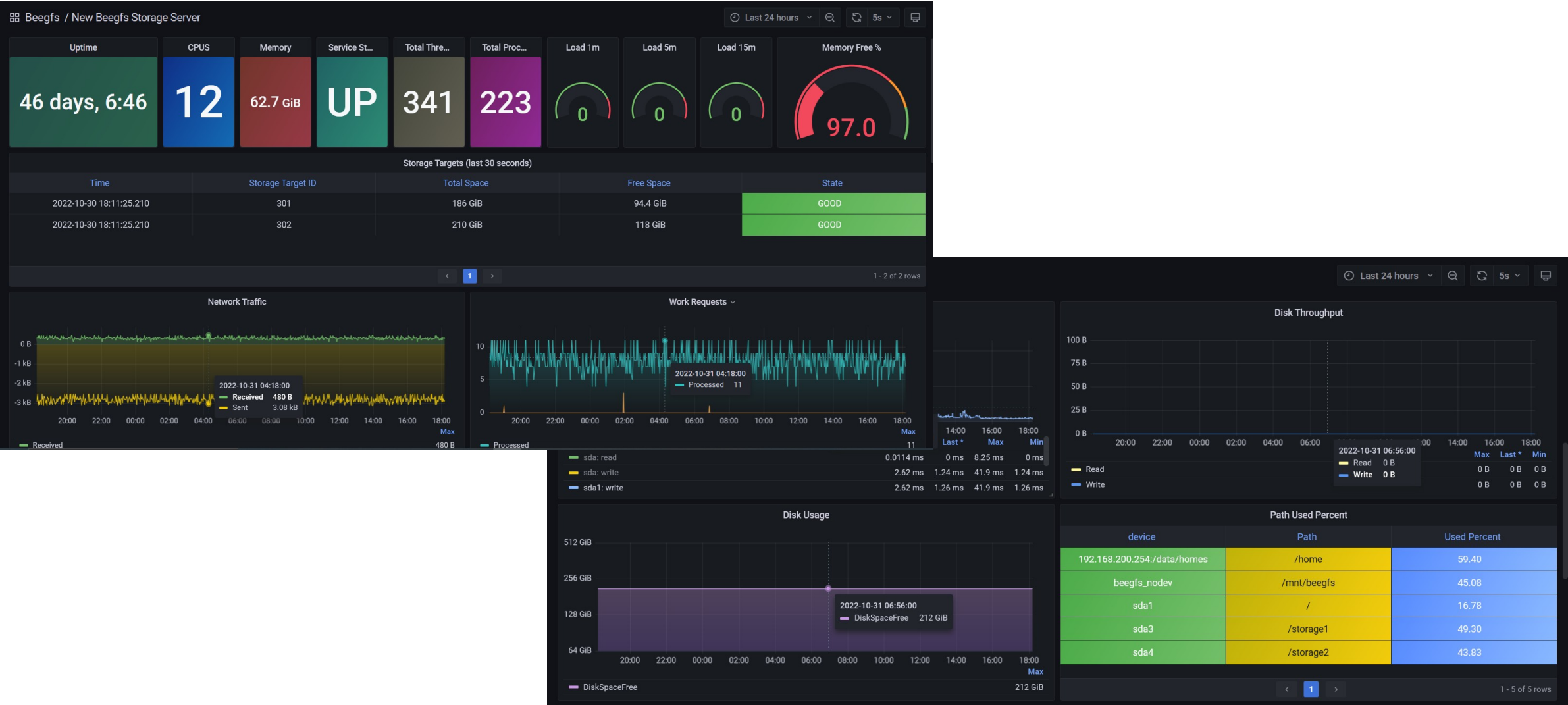


server01

grafana GUI - Live Storage Statistics



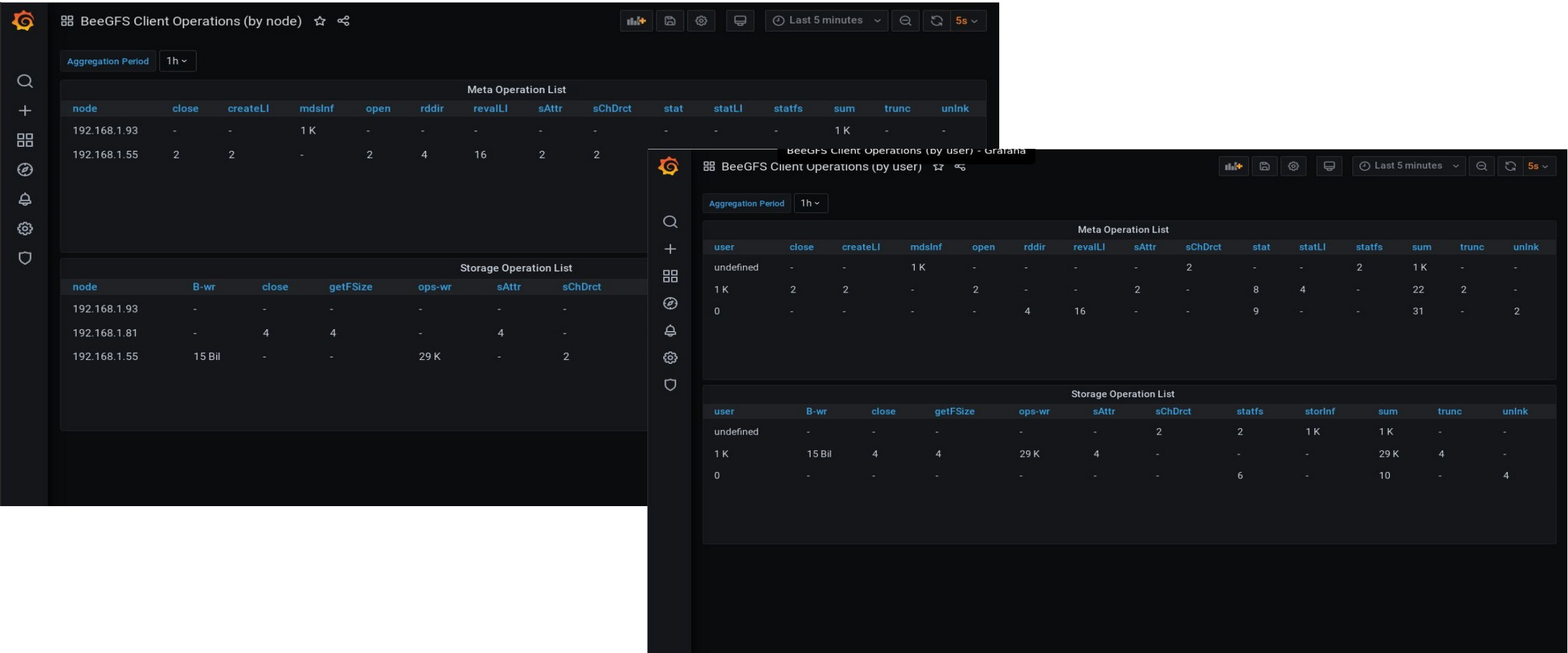
grafana GUI - Live Storage Statistics (new Version)



grafana GUI - Live Metadata Statistics (new Version)



Grafana GUI – Live per-client and per-user statistics



Grafana GUI – Event Notification

The screenshot shows the Grafana Alerting "Notification channels" configuration page. The left sidebar contains navigation icons for home, search, dashboard, alerting, settings, and help. The main content area is titled "Alerting" with the subtitle "Alert rules & notifications". Below this, there are two tabs: "Alert Rules" and "Notification channels", with the latter being active. The "New Notification Channel" form includes a "Name" text input, a "Type" dropdown menu set to "Email", and five toggle switches for "Default (send on all alerts)", "Include image", "Disable Resolve Message", and "Send reminders", all of which are currently turned off. Under the "Email settings" section, there is a "Single email" toggle (turned off) and an "Addresses" text area. A note below the text area states, "You can enter multiple email addresses using a ';' separator". At the bottom of the form are three buttons: "Save" (highlighted in blue), "Send Test", and "Back".

Quota Management

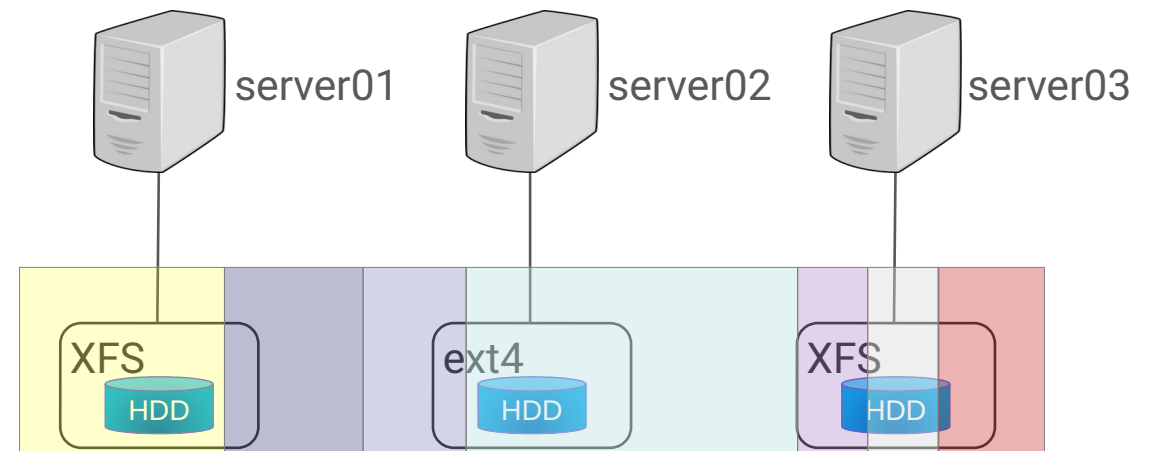
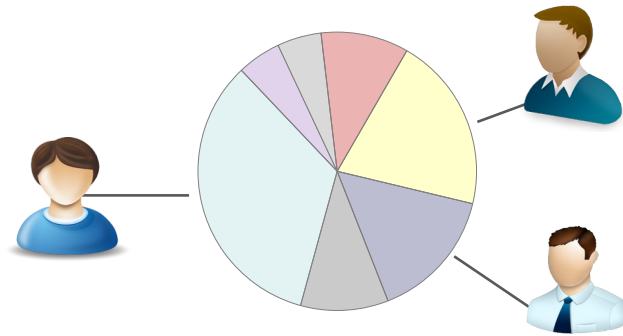


Definition of system-wide quotas

Types

- Storage space
- Inodes

For users or user groups



Quota Management

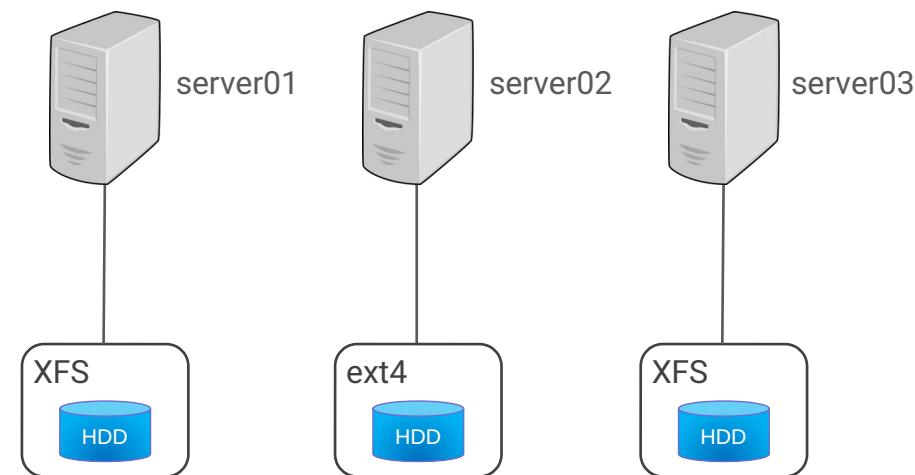


🐝 Requirements

- 🐝 Underlying FS of storage targets must support syscall quotactl()
 - 🐝 e.g. XFS and ext4
 - 🐝 ZFS only supports quotas for used space, not for inodes
 - 🐝 They must be mounted with quota enabled

```
server01:~ # $ mount /dev/sdb /mnt/raid1 -t xfs -orw,uqnoenforce,gqnoenforce,...
```

```
server01:~ # $ mkfs.ext4 -O quota /dev/sda  
Server01:~ # $ mount /dev/sda /mnt/raid1 -t ext4
```



Quota Management



Requirements

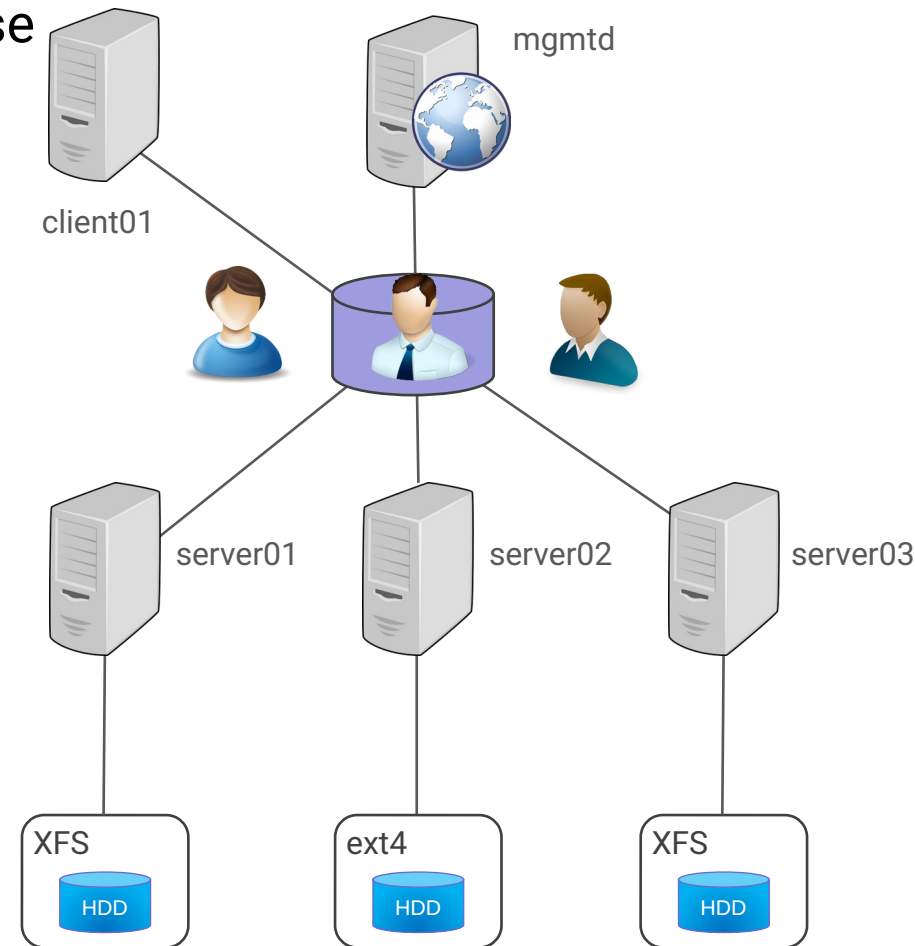
- All nodes must be able to query passwd and group database

```
server01:~ # $ getent passwd
```

```
root:x:0:0:root:/root:/bin/bash
lucy:x:1000:1000:Lucy Smith:/home/lucy:/bin/bash
denis:x:1001:1001:Denis Doe:/home/denis:/bin/bash
marco:x:1002:1002:Marco Moe:/home/marco:/bin/bash
...
```

```
server01:~ # $ getent group
```

```
root:x:0:
adm:x:4:lucy,denis
sudo:x:27:lucy
developers:x:29:denis,marco
...
```



Quota Management



🐝 Enabling quota tracking

- 🐝 Client service: file /etc/beegfs/beegfs-client.conf setting

`quotaEnabled = true`

🐝 Enable quota enforcement

- 🐝 In management configuration file

`quotaEnableEnforcement = true`

`quotaUpdateIntervalMin = 10`

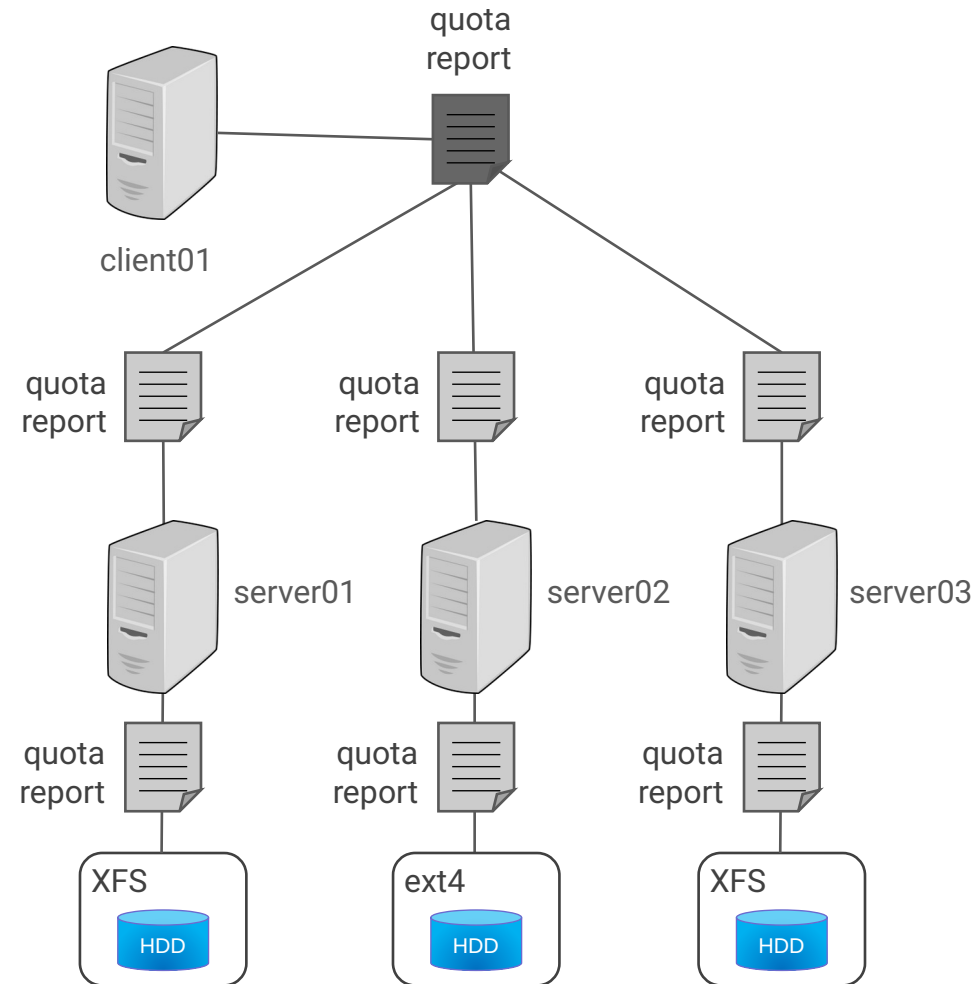
- 🐝 In Metadata and Storage configuration file

`quotaEnableEnforcement = true`

🐝 Restart all BeeGFS services

- 🐝 Enable quota tracking of all data chunk files

`client01:~ # beegfs-fsck --enablequota`



Quota Management



Querying quota information

```
client01:~ # beegfs-ctl --getquota --uid --all
```

user/group		size		chunk files	
name	id	used	hard	used	hard
lucy	1000	15 GiB	2 TiB	9020	3000000
dennis	1001	81 MiB	9 TiB	120	9000000
...					

```
client01:~ # beegfs-ctl --getquota --uid 1000
```

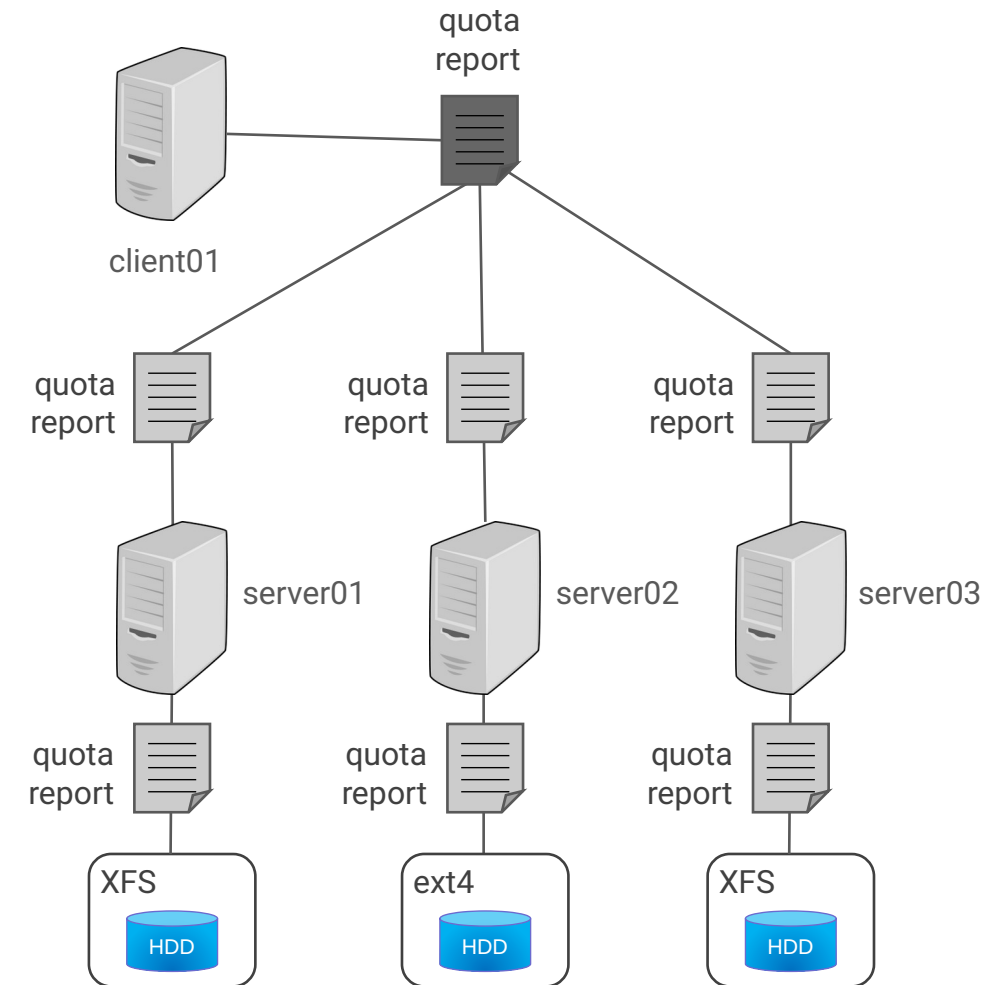
...

```
client01:~ # beegfs-ctl --getquota --uid --range 1000 1500
```

...

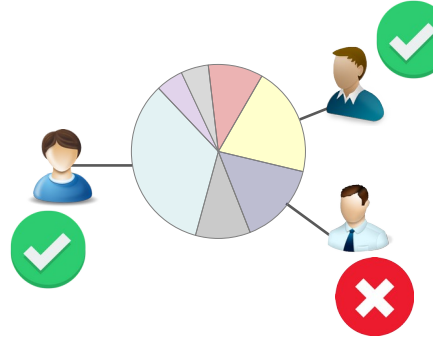
```
client01:~ # beegfs-ctl --getquota --gid 1000
```

...



Quota Management

Setting quota limits

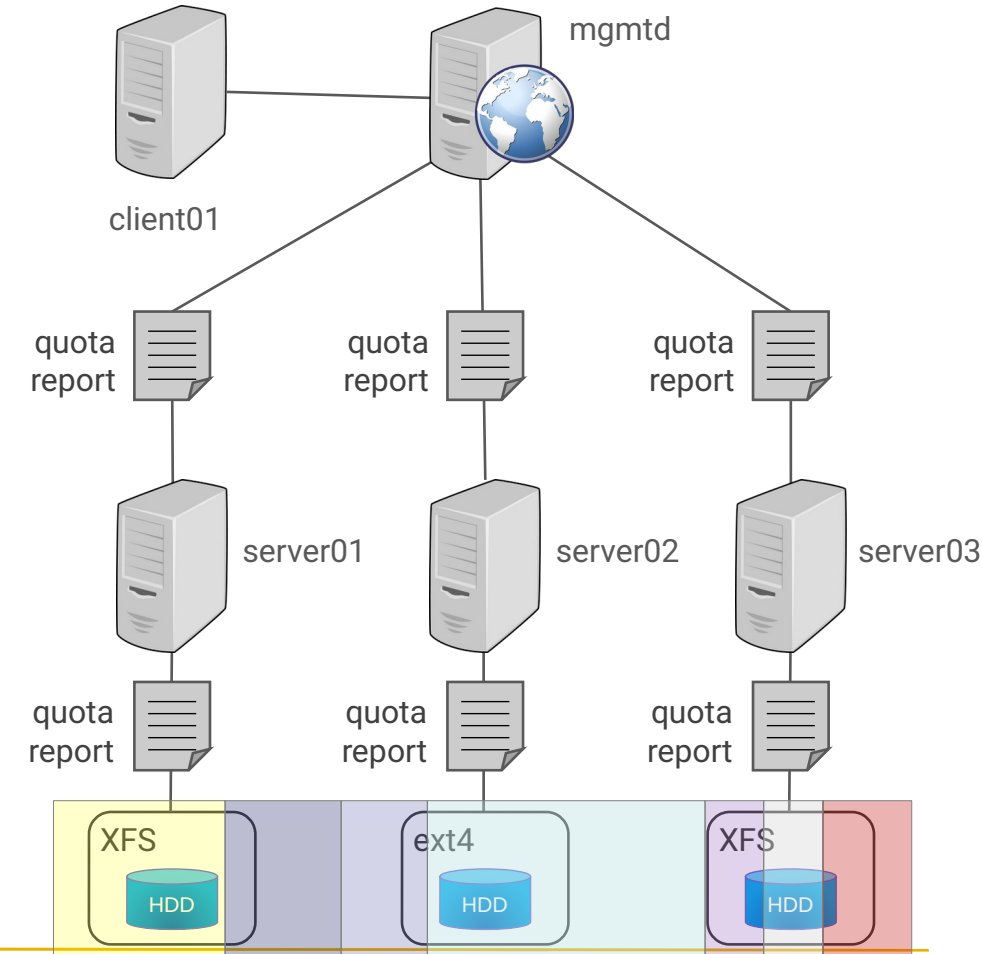


```
client01:~ # beegfs-ctl --setquota --uid 1000  
            --sizelimit=1G --inodelimit=5000
```

```
client01:~ # beegfs-ctl --setquota --gid 2000  
            --sizelimit=unlimited --inodelimit=5000
```

```
client01:~ # beegfs-ctl --setquota --uid --default  
            --sizelimit=10G --inodelimit=unlimited
```

```
client01:~ # beegfs-ctl --getquota -uid 1000
```



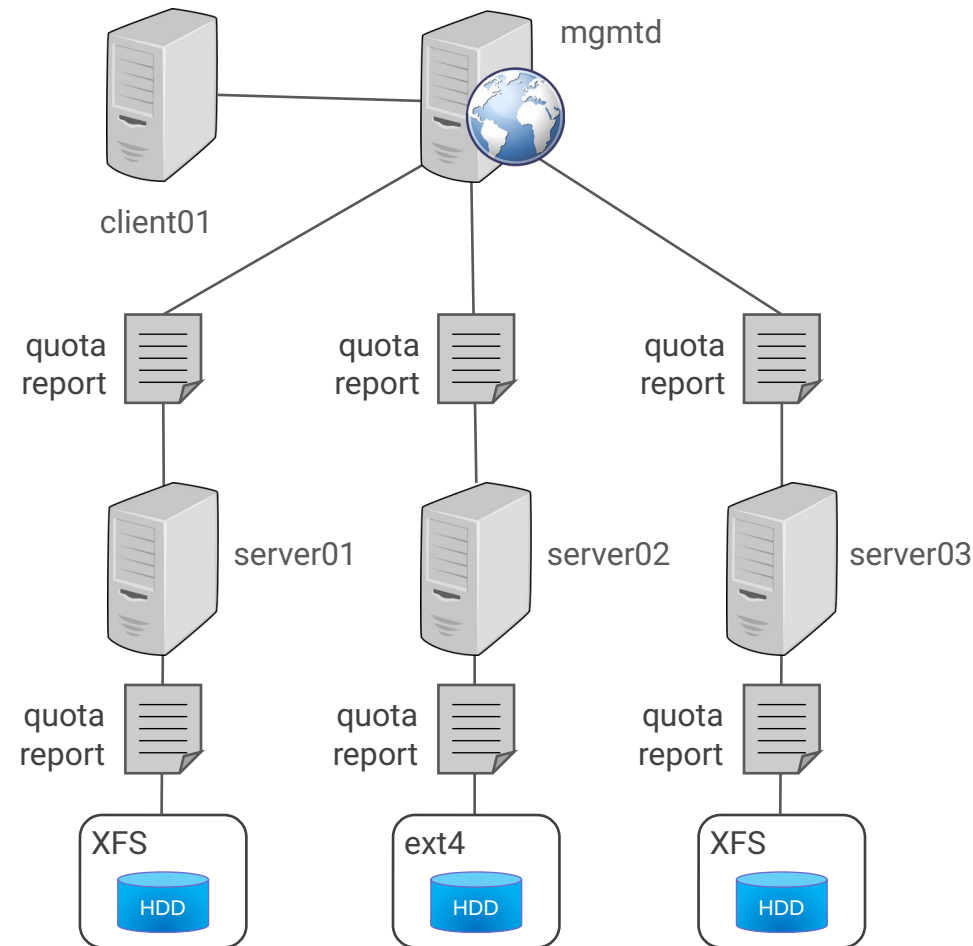
Quota Management



- Enable directory quotas (sort of)
 - BeeGFS is configured with quota support
 - Create a directory and set the „setgid“ flag

```
#> chmod g+s /mnt/beegfs/project01
```
 - Create a separate group for the project

```
#> chown root:project01 /mnt/beegfs/project01
```
 - Set the quota for the project01 group



DIY



Quota Management

-  Stop BeeGFS Cluster (Client, Storage, Metadata, Management)
-  Remount the disks with quota support enabled
-  Enable Quotas in BeeGFS
-  Set quotas for user training
-  Control the Quota settings

What's Next?

🐝 Typical Administration Task ✓

🐝 Sizing & Tuning

🐝 BeeGFS Resiliency Tools

🐝 BeeOND: BeeGFS on Demand

🐝 BeeGFS Hive Index

🐝 Conclusion



Thank You

Follow **BeeGFS**:

