Step by Step Gluster Setup

Table of Contents

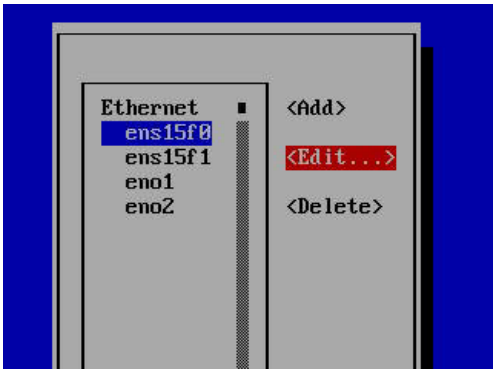
NOTE: All ***bold italicized*** words are commands to be entered in the command line.

Preflight

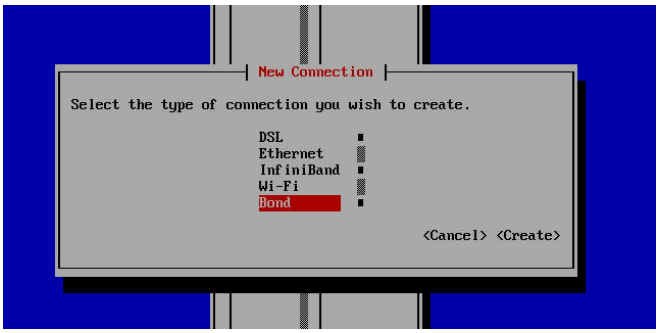
Configuring your Network (Do on all Nodes)

Make use of the Network Management tool

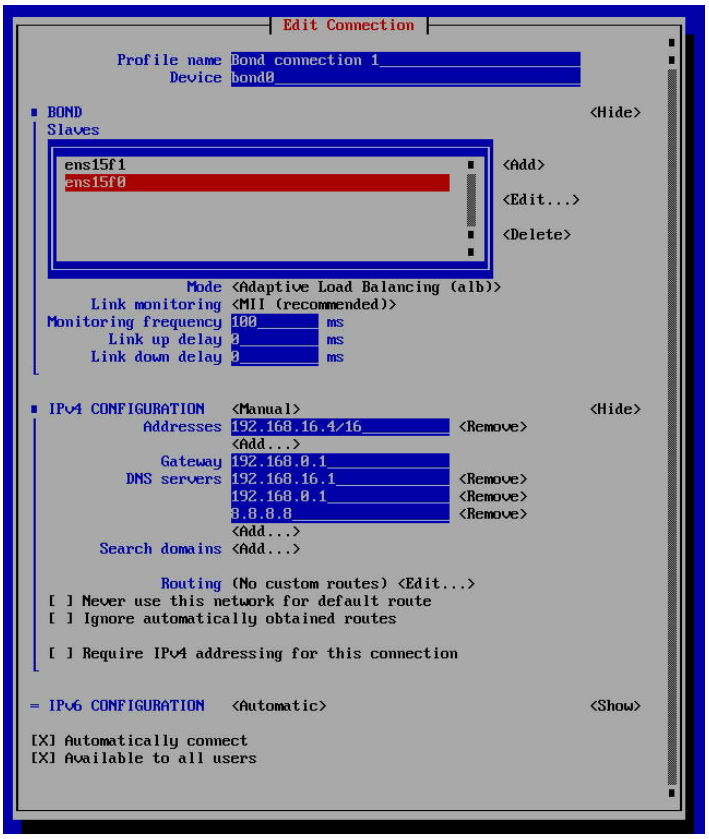
- ***nmtui***
- Edit a connection
- Eno1 and eno2 are onboard ports, other two are 10 GB NIC. Delete all interfaces to start fresh



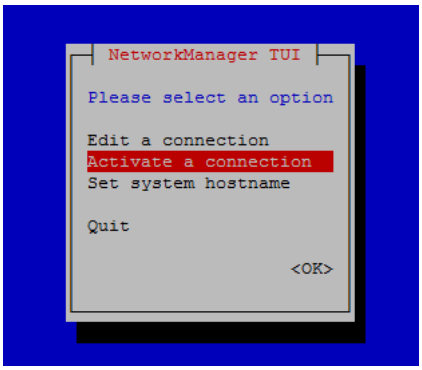
- Add → Bond



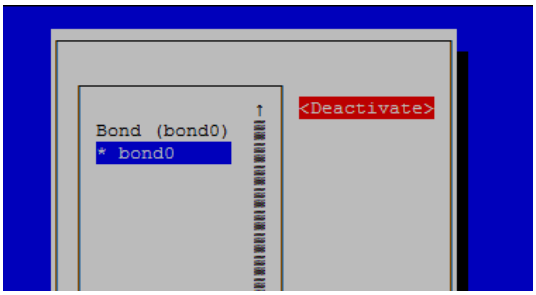
- Profile name & Device = bond0
  Add Bond Slaves, your two 10GB NIC names (ens15f0 ens15f1) if applicable
  Mode = Adaptive Load Balancing (alb)
  IPv4 Config = Automatic if using DHCP, IPv4 Config= Manual if you want Static
  See example below.



- Scroll down to Back, and then go to “Activate a connection”



- With “bond0” highlighted, go over to <Deactivate> and hit “Enter”. You will then see <Activate> and then hit “Enter” again.



- Then go down to Back, and then click OK to return to the command line.

- ***ip addr show*** → bond0 will show the IP address you can ping from the other servers.

```
[root@gluster1 ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
    link/ether 0c:c4:7a:6b:ea:64 brd ff:ff:ff:ff:ff:ff
3: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN qlen 1000
    link/ether 0c:c4:7a:6b:ea:65 brd ff:ff:ff:ff:ff:ff
4: ens15f0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP qlen 1000
    link/ether a0:36:9f:a2:8d:28 brd ff:ff:ff:ff:ff:ff
5: ens15f1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP qlen 1000
    link/ether a0:36:9f:a2:8d:2a brd ff:ff:ff:ff:ff:ff
7: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP qlen 1000
    link/ether a0:36:9f:a2:8d:2a brd ff:ff:ff:ff:ff:ff
    inet 192.168.16.4/16 brd 192.168.255.255 scope global bond0
       valid_lft forever preferred_lft forever
    inet6 fe80::b1e4:644d:5ee3:b79/64 scope link
       valid_lft forever preferred_lft forever
```

- ***ping 192.168.16.4***

```
[root@gluster1 ~]# ping 192.168.16.5
PING 192.168.16.5 (192.168.16.5) 56(84) bytes of data.
64 bytes from 192.168.16.5: icmp_seq=1 ttl=64 time=0.132 ms
64 bytes from 192.168.16.5: icmp_seq=2 ttl=64 time=0.165 ms
64 bytes from 192.168.16.5: icmp_seq=3 ttl=64 time=0.121 ms
64 bytes from 192.168.16.5: icmp_seq=4 ttl=64 time=0.123 ms
64 bytes from 192.168.16.5: icmp_seq=5 ttl=64 time=0.123 ms
^C
--- 192.168.16.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.121/0.132/0.165/0.022 ms
```

Install Required Packages (On all nodes)

- ***cd /root***
- ***ls***

```
[root@gluster1 ~]# cd /root
[root@gluster1 ~]# ls
anaconda-ks.cfg  gtools-2.1-1.noarch.rpm  preconfig
[root@gluster1 ~]# _
```

  if preconfig isn't there then ***wget images.45drives.com/gtools/preconfig***

- ***./preconfig –af***
- You'll need to reboot the system, log back in as root, and then ***./preconfig –af*** to finish the install.

Configure Services

NTP

- Unless you have your own NTP server, or Active Directory, you can use the CentOS defaults.
- To edit, ***vim /etc/ntp.conf*** → press ***i*** to enter text, and the ESC key when done, followed by ***:wq***

- ***systemctl enable ntpd***
- ***systemctl start ntpd***
- Test that all is working with ***ntpq –p*** → output should be the same format as below

```
[root@gluster1 ~]# ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*muug.ca         200.98.196.212   2 u   40   64    1   49.203    2.976   0.186
 159.203.31.244  24.141.214.195   2 u   40   64    1   30.573   26.837   0.338
 penguin.hopcoun 142.66.101.13    2 u   40   64    1   32.273    3.306   0.231
 sanction.trebor 192.5.41.209     2 u   40   64    1   30.547  103.504   0.471
```

Password less SSH

- ***vim /etc/hosts*** → and enter the IP and host name for all nodes being setup

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.16.4    gluster1
192.168.16.5    gluster2
```

- ***ssh-keygen –t rsa*** → (leave input blank just hit enter three times for simplicity)
- ***ssh-copy-id root@hostname*** → (Do for all hosts in /etc/hosts including itself)

Creating Storage

ZFS Storage Poll Setup (Do on every node)

Configure Drive Mapping

- ***dmap*** → options are as follows:

  Controller:
  - R750, r750, r (HighPoint R750)
  - LSI, lsi, l (LSI 9201 -24i)
  - Adaptec, adaptec, a (Adaptec HBA-1000i, ASR-81605Z)
  - Rr3740, rr (HighPoint RR3740)

  Chassis
  - 30, 45, or 60

- ***lsdev*** → (Grey = empty slot, Orange = clean drive, Green = Drive in a storage volume)

```
[root@gluster1 ~]# lsdev

| Disk Controller: HighPointR750 |
| DriverVersion:   |

| 1-15 | 1-16 | 2-21 |
| 1-14 | 1-17 | 2-20 |
| 1-13 | 1-18 | 2-19 |
| 1-12 | 1-19 | 2-18 |
| 1-11 | 1-20 | 2-17 |
| 1-10 | 1-21 | 2-16 |
| 1-9  | 1-22 | 2-15 |
| 1-8  | 1-23 | 2-14 |
| 1-7  | 1-24 | 2-13 |
| 1-6  | 2-1  | 2-12 |
| 1-5  | 2-2  | 2-11 |
| 1-4  | 2-3  | 2-10 |
| 1-3  | 2-4  | 2-9  |
| 1-2  | 2-5  | 2-8  |
| 1-1  | 2-6  | 2-7  |

| ROW1 | ROW2 | ROW3 |
```

Build ZFS Storage Pool

- ***zcreate -n (insert pool name) –l (insert RAID level (raidz2 suggested)) –v (# of VDEVs) –b (build flag)***

Below is a table of our suggested VDEV configurations:

Chassis Size	Maximum Storage Efficiency	Maximum IO per Second
Q30	3VDEVs of 10 Drives	5VDEVs of 6 Drives
S45	3VDEVs of 15 Drives	5VDEVs of 9 Drives
XL60	4VDEVs of 15 Drives	6VDEVs of 10 Drives

- now ***lsdev*** will show the slots are green
- ***systemctl enable zfs.target; systemctl start zfs.target***
- ***vim /usr/lib/systemd/system/zfs-import-cache.service***
- change line "ExecStart=" to be ***"ExecStart=/usr/local/libexec/zfs/startzfscache.sh"***
- ***mkdir /usr/local/libexec/zfs***
- ***vim /usr/local/libexec/zfs/startzfscache.sh*** and add the following in the file:
  #!/bin/sh
  sleep 10
  /sbin/zpool import –c /etc/zfs/zpool.cache –aN
  zfs mount –a
- ***chmod +x /usr/local/libexec/zfs/startzfscache.sh***

Gluster Volume Setup

Create Bricks (Do on all nodes)

To set up a cluster with GlusterFS, you must break up your big ZFS storage pool into several bricks to allow for the replication and/or distribution of data.
-A is for an Arbiter brick. An Arbiter brick is a brick that will store filenames and metadata, but no physical data. It is helpful in avoiding a split-brain, by knowing which file belongs to which brick etc.
-C is for a CTDB brick. A CTDB brick controls the sharing of the clustered volume. If one server goes down, the volume can still be access through the other servers etc.
There are a few things to consider when deciding how many bricks you want to create:

1. We recommend that a single brick shouldn't be more than 100TB in size.
2. Your brick needs to be larger in size than any single file that you plan to store on it.
3. More bricks mean more processes, so it can handle more clients better.

- ***mkbrick –n (ZFS pool name) -C -A -b (# of bricks wanted)***
- ***df –H*** → this will show you all that is mounted an you should see your *ZpoolName/volX*

```
[root@gluster1 ~]# df -H
Filesystem      Size  Used Avail Use% Mounted on
/dev/md125      108G  1.6G  106G   2% /
devtmpfs         17G     0   17G   0% /dev
tmpfs            17G     0   17G   0% /dev/shm
tmpfs            17G  9.7M   17G   1% /run
tmpfs            17G     0   17G   0% /sys/fs/cgroup
/dev/md126      1.1G  140M  925M  14% /boot
tmpfs           3.4G     0  3.4G   0% /run/user/0
zpool           200T  132k  200T   1% /zpool
zpool/ctdb      2.2G  263k  2.2G   1% /zpool/ctdb
zpool/vol1       45T  263k   45T   1% /zpool/vol1
zpool/vol2       45T  263k   45T   1% /zpool/vol2
zpool/vol3       45T  263k   45T   1% /zpool/vol3
zpool/vol4       45T  263k   45T   1% /zpool/vol4
```

Firewall Ports

- ***firewall-cmd --permanent --add-port=24007-24008/tcp***
- ***firewall-cmd --permanent --add-port=4379/tcp***
- ***firewall-cmd –reload***
- ***systemctl enable glusterd; systemctl start glusterd***
- ***gluster peer probe HostName*** → do this from one node, and probe all other nodes.

Creating your Gluster Volume (Only do on ONE node)

- ***vim /root/vol.conf***

  Linked list (4 nodes, 4 bricks)

```
gluster volume create tank replica 2 \
HOST1:/zpool/vol1/brick HOST2:/zpool/vol2/brick \
HOST2:/zpool/vol1/brick HOST3:/zpool/vol2/brick \
HOST3:/zpool/vol1/brick HOST4:/zpool/vol2/brick \
HOST4:/zpool/vol1/brick HOST1:/zpool/vol2/brick \
HOST1:/zpool/vol3/brick HOST2:/zpool/vol4/brick \
HOST2:/zpool/vol3/brick HOST3:/zpool/vol4/brick \
HOST3:/zpool/vol3/brick HOST4:/zpool/vol4/brick \
HOST4:/zpool/vol3/brick HOST1:/zpool/vol4/brick \
force
```

  Distributed Replica (4 nodes, 4 bricks)

```
gluster vol create tank replica 2 \
HOST1:/zpool/vol1/brick HOST2:/zpool/vol1/brick \
HOST1:/zpool/vol2/brick HOST2:/zpool/vol2/brick \
HOST1:/zpool/vol3/brick HOST2:/zpool/vol3/brick \
HOST1:/zpool/vol4/brick HOST2:/zpool/vol4/brick \
HOST3:/zpool/vol1/brick HOST4:/zpool/vol1/brick \
HOST3:/zpool/vol2/brick HOST4:/zpool/vol2/brick \
HOST3:/zpool/vol3/brick HOST4:/zpool/vol3/brick \
HOST3:/zpool/vol4/brick HOST4:/zpool/vol4/brick \
force
```

  Distributed (4 nodes, 4 bricks)

```
gluster volume create tank \
HOST1:/zpool/vol1/brick HOST1:/zpool/vol2/brick \
HOST1:/zpool/vol3/brick HOST1:/zpool/vol4/brick \
HOST2:/zpool/vol1/brick HOST2:/zpool/vol2/brick \
HOST2:/zpool/vol3/brick HOST2:/zpool/vol4/brick \
HOST3:/zpool/vol1/brick HOST3:/zpool/vol2/brick \
HOST3:/zpool/vol3/brick HOST3:/zpool/vol4/brick \
HOST4:/zpool/vol1/brick HOST4:/zpool/vol2/brick \
HOST4:/zpool/vol3/brick HOST4:/zpool/vol4/brick \
force
```

- ***gcreate -c /root/vol.conf -b X -n Y -n Z …***

  X = # of bricks per node. Y,Z,…= hostname of all other nodes.

Creating your CTDB Volume (Only do on ONE node)

- ***vim /root/ctdb.conf***

```
gluster volume create ctdb replica 2 \
gluster1:/zpool/ctdb/brick gluster2:/zpool/ctdb/brick \
force
```

NOTE: if using 3 servers or more, make it a replica 3.

- ***gcreate -c /root/ctdb.conf -b 1 -n Y -n Z …***
  -b 1 (only one CTDB brick per node), Y,Z,…= hostname of all other nodes.
- ***mkdir /mnt/ctdb*** → /mnt/ctdb is just our example.
- ***echo localhost:/ctdb /mnt/ctdb glusterfs defaults,_netdev 0 0 >> /etc/fstab***
- ***mount /mnt/ctdb***

Firewall ports

- ***gluster volume status*** → this will output a table similar to the below

```
[root@gluster2 /]# gluster volume status
Status of volume: ctdb
Gluster process                                 TCP Port  RDMA Port  Online  Pid
------------------------------------------------------------------------------
Brick gluster1:/zpool/ctdb/brick                49152     0          Y       6621
Brick gluster2:/zpool/ctdb/brick                49152     0          Y       4300
Self-heal Daemon on localhost                   N/A       N/A        Y       5061
Self-heal Daemon on gluster1                    N/A       N/A        Y       6610

Task Status of Volume ctdb
------------------------------------------------------------------------------
There are no active volume tasks

Status of volume: tank
Gluster process                                 TCP Port  RDMA Port  Online  Pid
------------------------------------------------------------------------------
Brick gluster1:/zpool/vol1/brick                49153     0          Y       6629
Brick gluster1:/zpool/vol2/brick                49154     0          Y       6637
Brick gluster1:/zpool/vol3/brick                49155     0          Y       6645
Brick gluster1:/zpool/vol4/brick                49156     0          Y       6651
Brick gluster2:/zpool/vol1/brick                49153     0          Y       4308
Brick gluster2:/zpool/vol2/brick                49154     0          Y       4316
Brick gluster2:/zpool/vol3/brick                49155     0          Y       4323
Brick gluster2:/zpool/vol4/brick                49156     0          Y       4330
NFS Server on localhost                         2049      0          Y       1808
NFS Server on gluster1                          2049      0          Y       3772

Task Status of Volume tank
------------------------------------------------------------------------------
There are no active volume tasks
```

- ***firewall-cmd --permanent --add-ports=49152-49156/tcp***
- ***firewall-cmd --permanent --add-ports=2049/tcp***
- ***firewall-cmd --reload***

Sharing

Check to see if your CTDB volume is mounted with the ***df*** command.
Should say “localhost:ctdb” at the bottom of the output.

SMB

- ***mkdir /mnt/ctdb/files***
- ***vim /mnt/ctdb/files/ctdb*** → enter the following information

```
CTDB_RECOVERY_LOCK=/mnt/ctdb/.CTDB-lockfile
CTDB_NODES=/etc/ctdb/nodes
CTDB_PUBLIC_ADDRESSES=/etc/ctdb/public_addresses
CTDB_MANAGES_SAMBA=yes
```

- ***vim /mnt/ctdb/files/nodes*** → enter the IP addresses of all nodes being set up like below

```
192.168.16.4
192.168.16.5
```

- ***vim /mnt/ctdb/files/public_addresses*** → enter an IP which will be used to access the share
  Ex: 192.168.16.160/16 bond0 (/16 is the Subnet Mask & bond0 is the interface)
- ***vim /mnt/ctdb/files/smb.conf*** → below is the basic config, you’ll need to adjust permissions
  [gluster-tank] is the share name.

```
[global]
        workgroup = SAMBA
        security = user
        passdb backend = tdbsam
        printing = cups
        printcap name = cups
        load printers = yes
        cups options = raw

[gluster-tank]
        comment = For samba share of volume tank
        vfs objects = glusterfs
        glusterfs:volume = tank
        glusterfs:logfile = /var/log/samba/gluster-tank.log
        glusterfs:loglevel = 7
        path = /
        read only = no
        guest ok = yes
        kernel share modes = No
```

- These files need to be on <u>every node</u> at the following locations:
  - ctdb = /etc/sysconfig/ctdb
  - nodes = /etc/ctdb/nodes
  - public_addresses = /etc/ctdb/public_addresses
  -smb.conf = /etc/samba/smb.conf
  -This can all be done from one node using passwordless SSH:
  Ex: ***ssh root@gluster2 “cp /mnt/ctdb/files/nodes /etc/ctdb nodes”***
- ***touch /mnt/ctdb/files/.CTDB-lockfile***
- ***firewall-cmd --permanent --add-service=samba; firewall-cmd --reload***
- ***systemctl enable ctdb; systemctl start ctdb***
- ***systemctl disable smb; systemctl disable nfs***
- ***testparm*** → This will check the smb.conf file for any issues.

Creating Groups/Users to Access your SMB Share

- Create a group which will be given access to the share → ***groupadd groupName***
- Create a user within that group → ***useradd username -G groupName***
- Add user to Samba database → ***smbpasswd -a username***
- Edit the smb.conf, in the share section to add → ***valid users = @groupName***
- If you only want one user to access the volume, do not include the -G option when creating the user, and make ***valid users = username***

NFS

- ***mkdir /mnt/ctdb/files***
- ***vim /mnt/ctdb/files/ctdb*** → enter the following information

```
CTDB_RECOVERY_LOCK=/mnt/ctdb/.CTDB-lockfile
CTDB_NODES=/etc/ctdb/nodes
CTDB_PUBLIC_ADDRESSES=/etc/ctdb/public_addresses
CTDB_MANAGES_NFS=yes
```

- ***vim /mnt/ctdb/files/nodes*** → enter the IP addresses of all nodes being set up like below

```
192.168.16.4
192.168.16.5
```

- ***vim /mnt/ctdb/files/public_addresses*** → enter an IP which will be used to access the share
  Ex: 192.168.16.160/16 bond0 (/16 is the Subnet Mask & bond0 is the interface)
- These files need to be on every node at the following locations:
  - ctdb = /etc/sysconfig/ctdb
  - nodes = /etc/ctdb/nodes
  - public_addresses = /etc/ctdb/public_addresses
  -This can all be done from one node using passwordless SSH:
  Ex: ***ssh root@gluster2 "cp /mnt/ctdb/files/nodes /etc/ctdb nodes"***
- ***touch /mnt/ctdb/files/.CTDB-lockfile***
- ***firewall-cmd --permanent --add-service=nfs***
- ***firewall-cmd --permanent --add-port=111/tcp***
- ***firewall-cmd --permanent --add-port=38465-38467/tcp***
- ***firewall-cmd --reload***
- ***gluster volume set (volume name) nfs.disable off***
- ***gluster volume set (volume name) nfs.rpc-auth-allow <ip range>***
  Ex: on a 255.255.0.0 Subnet, we put 192.168.*.* so anyone on network can access.
- ***gluster volume set (volume name) nfs.export-volumes on***
- ***systemctl enable ctdb; systemctl start ctdb***
- ***systemctl disable smb; systemctl disable nfs***

Creating Groups/Users to access your NFS share

- Create a group which will be given access to the share → ***groupadd groupName***
- Create a user within that group → ***useradd username -G groupName***
- Set a user and group to be the owner of the share → chown username:groupName /mnt/tank
- Mount on client using credentials:
  mount -t nfs <externalIP>:VolumeName -o username=X,password=Y /directoryOfChoice

Firewall Cheat Sheet

Application	Add-port
NFS (RPC Bind)	111/tcp
Communication for Gluster nodes	24007-24008/tcp
GlusterFS NFS Service	38465-38467/tcp & 2049/tcp
Communication for CTDB	4379/tcp
Gluster Bricks	49152-4915X/tcp

Application	Add-service
Samba	samba
NFS	nfs