Linux Networking

MANAGEMENT

Table of Contents

General Networking

2. Interface Configuration

2.1. To GUI or not to GUI

Recent Linux distributions often include a graphical application to configure the network. Some people complain that these applications mess networking configurations up when used simultaneously with command line configurations. Notably Network Manager (often replaced by wicd) and yast are known to not care about configuration changes via the command line. Since the goal of this course is server administration, we will assume our Linux servers are always administered through the command line. This chapter only focuses on using the command line for network interface configuration!
Unfortunately, there is no single combination of Linux commands and /etc files that works on all Linux distributions.

2.1.1. Configuring a Network Connection with Yast

There are many supported networking types on Linux. Most of them use different device names and the configuration files are spread over several locations in the file system.

On SUSE Linux Enterprise Desktop, where NetworkManager is active by default, all network cards are configured. If NetworkManager is not active, only the first interface with link up (with a network cable connected) is automatically configured. Additional hardware can be configured any time on the installed system. The following sections describe the network configuration for all types of network connections supported by SUSE Linux Enterprise Server.

To configure your wired or wireless network card in YaST, select Network Devices > Network Settings. After starting the module, YaST displays the Network Settings dialog with four tabs: Global Options, Overview, Hostname/DNS and Routing.

The Global Options tab allows you to set general networking options such as the use of NetworkManager, IPv6 and general DHCP options. For more information, see Configuring Global Networking Options.

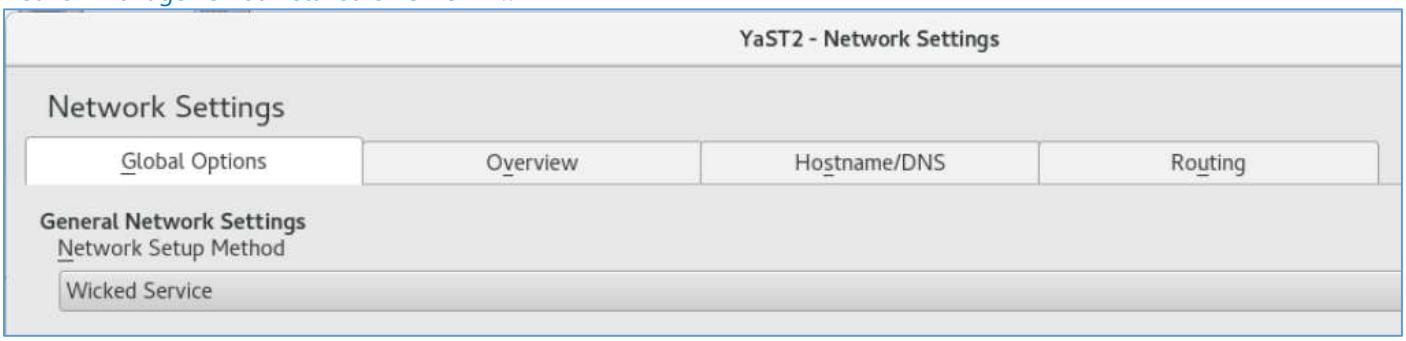
The Overview tab contains information about installed network interfaces and configurations. Any properly detected network card is listed with its name. You can manually configure new cards, remove or change their configuration in this dialog. If you want to manually configure a card that was not automatically detected, see Configuring an Undetected Network Card. If you want to change the configuration of an already configured card, see Changing the Configuration of a Network Card.

The Hostname/DNS tab allows to set the hostname of the machine and name the servers to be used. For more information, see Configuring Hostname and DNS.
The Routing tab is used for the configuration of routing. See Configuring Routing for more information.

2.1.2. NetworkManager

However, NetworkManager is not a suitable solution for all cases, so you can still choose between the traditional method for managing network connections (ifup) and NetworkManager. If you want to manage your network connection with NetworkManager, enable NetworkManager in the YaST Network Settings module and configure your network connections with NetworkManager.

NetworkManager is not installed on SLES 12 !!



2.2. Configuring a Network Connection Manually

Manual configuration of the network software should be the last alternative. Using YaST is recommended. However, this background information about the network configuration can also assist your work with YaST.

2.2.1. The wicked Network Configuration

The tool and library called wicked provides a new framework for network configuration.
One of the challenges with traditional network interface management is that different layers of network management get jumbled together into one single script, or at most two different scripts, that interact with each other in a not-really-well-defined way, with side effects that are difficult to be aware of, obscure constraints and conventions, etc. Several layers of special hacks for a variety of different scenarios increase the maintenance burden. Address configuration protocols are being used that are implemented via daemons like dhcpcd, which interact rather poorly with the rest of the infrastructure. Funky interface naming schemes that require heavy udev support are introduced to achieve persistent identification of interfaces.

On SUSE Linux Enterprise, wicked is running by default. In case you want to check what is currently enabled and whether it is running, call :

2.2.1.1. Using wicked

On SUSE Linux Enterprise, `wicked` is running by default. In case you want to check what is currently enabled and whether it is running, call:

```
linsrv1:~ # systemctl status network
● wicked.service - wicked managed network interfaces
   Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled; vendor prese
t: disabled)
   Active: active (exited) since Thu 2016-12-15 15:06:27 CET; 7h ago
 Main PID: 1329 (code=exited, status=0/SUCCESS)
    Tasks: 0 (limit: 512)
   CGroup: /system.slice/wicked.service

Dec 15 15:06:23 linsrv1 systemd[1]: Starting wicked managed network interfa.....
Dec 15 15:06:27 linsrv1 wicked[1329]: lo              up
Dec 15 15:06:27 linsrv1 wicked[1329]: eth1            up
Dec 15 15:06:27 linsrv1 systemd[1]: Started wicked managed network interfaces.
Hint: Some lines were ellipsized, use -l to show in full.
```

```
linsrv1:/etc/sysconfig/network # wicked show all
lo              up
      link:     #1, state up
      type:     loopback
      config:   compat:suse:/etc/sysconfig/network/ifcfg-lo
      leases:   ipv4 static granted
      addr:     ipv4 127.0.0.1/8 [static]

eth0            device-unconfigured
      link:     #2, state down, mtu 1500
      type:     ethernet, hwaddr 00:50:04:89:68:08

eth1            up
      link:     #3, state up, mtu 1500
      type:     ethernet, hwaddr 00:19:99:cd:51:f7
      config:   compat:suse:/etc/sysconfig/network/ifcfg-eth1
      leases:   ipv4 static granted
      addr:     ipv4 192.168.0.40/24 [static]
      route:    ipv4 default via 192.168.0.1 proto boot
```

2.2.1.2. Bringing Up Multiple Interfaces

For bonds and bridges, it may make sense to define the entire device topology in one file (ifcfg-bondX), and bring it up in one go. wicked then can bring up the whole configuration if you specify the top-level interface names (of the bridge or bond):

```
linsrv1:~ # wicked ifup all
lo              up
eth1            up
```

2.2.2. Configuration Files

This section provides an overview of the network configuration files and explains their purpose and the format used.

/etc/wicked/common.xml

The /etc/wicked/common.xml file contains common definitions that should be used by all applications. It is sourced/included by the other configuration files in this directory. Even though you can use this file to, for example, enable debugging across all wicked components, we recommend to use the file /etc/wicked/local.xml for this purpose. After applying maintenance updates you might lose your changes as the /etc/wicked/common.xml might be overwritten.
The /etc/wicked/common.xml file includes the /etc/wicked/local.xml in the default installation, thus you typically do not need to modify the /etc/wicked/common.xml.

In case you want to disable nanny by setting the <use-nanny> to false, restart the wickedd.service and then run the following command to apply all configurations and policies:

```
wicked ifup all
```

/etc/sysconfig/network/ifcfg-*

These files contain the traditional configurations for network interfaces. In SUSE Linux Enterprise 11, this was the only supported format besides iBFT firmware.

NOTE: wicked and the ifcfg-* Files

wicked reads these files if you specify the compat: prefix. According to the SUSE Linux Enterprise Server 12 default configuration in /etc/wicked/client.xml, wicked tries these files before the XML configuration files in /etc/wicked/ifconfig. The --ifconfig switch is provided mostly for testing only. If specified, default configuration sources defined in /etc/wicked/ifconfig are not applied.

The ifcfg-* files include information such as the start mode and the IP address. Possible parameters are described in the manual page of ifup. Additionally, most variables from the dhcp and wireless files can be used in the ifcfg-* files if a general setting should be used for only one interface. However, most of the/etc/sysconfig/network/config variables are global and cannot be overridden in ifcfg-files. For example, NETCONFIG_* variables are global.

For ifcfg.template, see /etc/sysconfig/network/config, /etc/sysconfig/network/dhcp, and /etc/sysconfig/network/wireless.

```
linsrv1:/etc/sysconfig/network # cat ifcfg-eth1
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='192.168.0.40/24'
MTU=''
NAME=''
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='auto'
DHCLIENT_SET_DEFAULT_ROUTE='yes'
```

```
linsrv1:/etc/sysconfig/network # cat ifcfg-lo
# Loopback (lo) configuration
IPADDR=127.0.0.1/8
NETMASK=255.0.0.0
NETWORK=127.0.0.0
STARTMODE=nfsroot
BOOTPROTO=static
USERCONTROL=no
FIREWALL=no
```

/etc/nsswitch.conf

The introduction of the GNU C Library 2.0 was accompanied by the introduction of the Name Service Switch (NSS). Refer to the nsswitch.conf(5) man page and The GNU C Library Reference Manual for details.

The order for queries is defined in the file /etc/nsswitch.conf. A sample nsswitch.conf is shown in Example 16-10. Comments are preceded by # signs. In this example, the entry under the hosts database means that a request is sent to /etc/hosts (files) via DNS.

```
linsrv1:~ # cat /etc/nsswitch.conf | grep hosts
hosts:  files dns wins
```

/etc/hosts

In this file, shown in Example 16-7, IP addresses are assigned to host names. If no name server is implemented, all hosts to which an IP connection will be set up must be listed here. For each host, enter a line consisting of the IP address, the fully qualified host name, and the host name into the file. The IP address must be at the beginning of the line and the entries separated by blanks and tabs. Comments are always preceded by the # sign.

```
linsrv1:~ # cat /etc/hosts
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.
# Syntax:
#
# IP-Address  Full-Qualified-Hostname  Short-Hostname
#

127.0.0.1       localhost

# special IPv6 addresses
::1             localhost ipv6-localhost ipv6-loopback

fe00::0         ipv6-localnet

ff00::0         ipv6-mcastprefix
ff02::1         ipv6-allnodes
ff02::2         ipv6-allrouters
ff02::3         ipv6-allhosts
192.168.0.40    linsrv1.terwal.local linsrv1
```

/etc/HOSTNAME

/etc/HOSTNAME contains the fully qualified host name (FQHN). The fully qualified host name is the host name with the domain name attached. This file must contain only one line (in which the host name is set). It is read while the machine is booting.

```
linsrv1:~ # cat /etc/HOSTNAME
linsrv1.terwal.local
```

2.2.3. Testing the Configuration

Before you write your configuration to the configuration files, you can test it. To set up a test configuration, use the ip command. To test the connection, use the ping command.

The command ip changes the network configuration directly without saving it in the configuration file. Unless you enter your configuration in the correct configuration files, the changed network configuration is lost on reboot.

NOTE : ifconfig and route Are Obsolete

The ifconfig and route tools are obsolete. Use ip instead. ifconfig, for example, limits interface names to 9 characters.

2.2.3.1. Configuring a Network Interface with ip

ip is a tool to show and configure network devices, routing, policy routing, and tunnels.

```
linsrv1:~ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t 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
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen
 1000
    link/ether 00:50:04:89:68:08 brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 00:19:99:cd:51:f7 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.40/24 brd 192.168.0.255 scope global eth1
       valid_lft forever preferred_lft forever
```

2.2.3.2. Testing a Connection with ping

```
linsrv1:~ # ping -c 3 linsrv1.terwal.local
PING linsrv1.terwal.local (192.168.0.40) 56(84) bytes of data.
64 bytes from linsrv1.terwal.local (192.168.0.40): icmp_seq=1 ttl=64 time=0.017
ms
64 bytes from linsrv1.terwal.local (192.168.0.40): icmp_seq=2 ttl=64 time=0.011
ms
64 bytes from linsrv1.terwal.local (192.168.0.40): icmp_seq=3 ttl=64 time=0.009
ms

--- linsrv1.terwal.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.009/0.012/0.017/0.004 ms
```

2.2.4. Unit Files and Start-Up Scripts

Apart from the configuration files described above, there are also systemd unit files and various scripts that load the network services while the machine is booting. These are started when the system is switched to the multi-user.target target. Some of these unit files and scripts are described in Some Unit Files and Start-Up Scripts for Network Programs. For more information about systemd, see Section 14.0, The systemd Daemon and for more information about the systemd targets, see the man page ofsystemd.special (man systemd.special).

Some Unit Files and Start-Up Scripts for Network Programs

- network.target

network.target is the systemd target for networking, but its mean depends on the settings provided by the system administrator.
For more information, see http://www.freedesktop.org/wiki/Software/systemd/NetworkTarget/.

- multi-user.target

multi-user.target is the systemd target for a multiuser system with all required network services.

- xinetd

Starts xinetd. xinetd can be used to make server services available on the system. For example, it can start vsftpd whenever an FTP connection is initiated.

- rpcbind

Starts the rpcbind utility that converts RPC program numbers to universal addresses. It is needed for RPC services, such as an NFS server.

- ypserv

Starts the NIS server.

- ypbind

Starts the NIS client.

- /etc/init.d/nfsserver

Starts the NFS server.

- /etc/init.d/postfix

Controls the postfix process.

2.3. Dhclient

Home and client Linux desktops often have /sbin/dhclient running. This is a daemon that enables a network interface to lease an ip configuration from a dhcp server. When your adapter is configured for dhcp or bootp, then /sbin/ifup will start the dhclient daemon.

When a lease is renewed, dhclient will override your ifconfig set ip address!

2.4. Arp

The ip to mac resolution is handled by the layer two broadcast protocol arp. The arp table can be displayed with the arp tool. The screenshot below shows the list of computers that this computer recently communicated with.

```
linsrv1:~ # arp -a
? (192.168.0.1) at c4:04:15:13:31:25 [ether] on eth1
? (192.168.0.20) at 00:19:99:ce:8e:71 [ether] on eth1
? (192.168.0.22) at 6c:62:6d:d4:3d:77 [ether] on eth1
? (192.168.0.10) at 00:18:4d:ff:ff:07 [ether] on eth1
```

2.5. Optional: ethtool

```
linsrv1:~ # ethtool eth1
Settings for eth1:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: on (auto)
        Supports Wake-on: pumbg
        Wake-on: g
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```

2.6. Practice: Interface configuration

Verify whether dhclient is running.

Display your current ip address(es).

Display the configuration file where this ip address is defined.

Follow the nic configuration in the book to change your ip address from dhcp client to fixed. Keep the same ip address to avoid conflicts!

Did you also configure the correct gateway in the previous question? If not, then do this now.
Verify that you have a gateway.

Verify that you can connect to the gateway, that it is alive.

Change the last two digits of your mac address.

Which ports are used by http, pop3, ssh, telnet, nntp and ftp?

Explain why e-mail and websites are sent over tcp and not udp.

Display the hostname of your computer.

Which ip-addresses did your computer recently have contact with?

1. Network Sniffing

2. Binding and Bonding

3. SSH Client and Server

4. Sharing File Systems with NFS

Distributing and sharing file systems over a network is a common task in corporate environments. The well-proven network file system (NFS) works with NIS, the yellow page's protocol. For a more secure protocol that works with LDAP and Kerberos, check NFSv4 (default). Combined with pNFS, you can eliminate performance bottlenecks.

NFS with NIS makes a network transparent to the user. With NFS, it is possible to distribute arbitrary file systems over the network. With an appropriate setup, users always find themselves in the same environment regardless of the terminal they currently use.

IMPORTANT: Need for DNS

In principle, all exports can be made using IP addresses only. To avoid time-outs, you need a working DNS system. DNS is necessary at least for logging purposes, because the mountd daemon does reverse lookups.

4.1. NFS Protocol Versions

The following are terms used in the YaST module.

- Exports

A directory exported by an NFS server, which clients can integrate it into their system.

- NFS Client

The NFS client is a system that uses NFS services from an NFS server over the Network File System protocol. The TCP/IP protocol is already integrated into the Linux kernel; there is no need to install any additional software.

- NFS Server

The NFS server provides NFS services to clients. A running server depends on the following daemons: nfsd (worker), idmapd (user and group name mappings to IDs and vice versa), statd (file locking), and mountd (mount requests).

- NFSv3

NFSv3 is the version 3 implementation, the old stateless NFS that supports client authentication.

- NFSv4

NFSv4 is the new version 4 implementation that supports secure user authentication via kerberos. NFSv4 requires one single port only and thus is better suited for environments behind a firewall than NFSv3.
The protocol is specified as http://tools.ietf.org/html/rfc3530.

- pNFS

Parallel NFS, a protocol extension of NFSv4. Any pNFS clients can directly access the data on an NFS server.

4.2. RPCinfo

4.3. Server Configuration

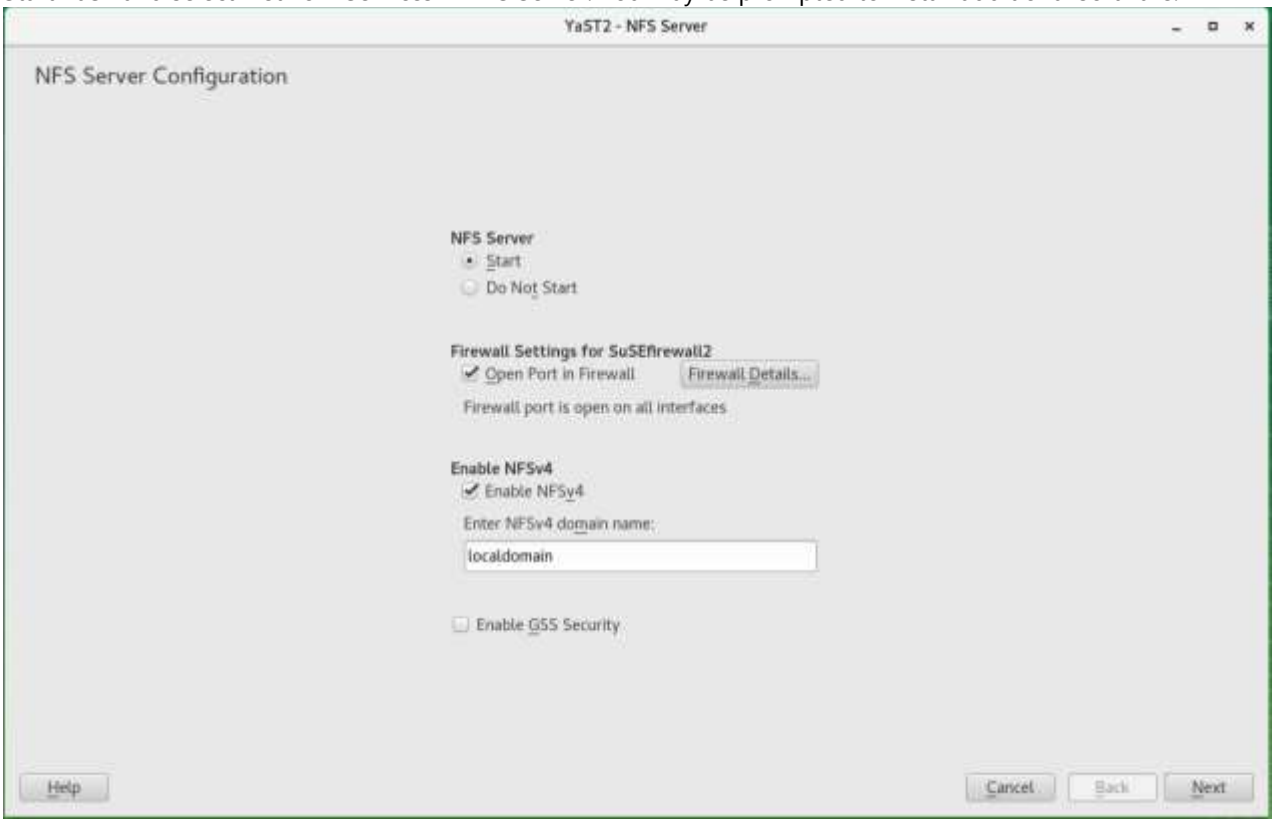
Configuring an NFS server can be done either through YaST or manually.
For authentication, NFS can also be combined with Kerberos.

4.4. Exporting File Systems with YaST

With YaST, turn a host in your network into an NFS server—a server that exports directories and files to all hosts granted access to it or to all members of a group. Thus, the server can also provide applications without installing the applications locally on every host.

To set up such a server, proceed as follows:

Start YaST and select Network Services > NFS Server. You may be prompted to install additional software.



1. Activate the Start radio button.

2. If a firewall is active on your system (SuSEFirewall2), check Open Ports in Firewall. YaST adapts its configuration for the NFS server by enabling the nfs service.

3. Check whether you want to Enable NFSv4. If you deactivate NFSv4, YaST will only support NFSv3. For information about enabling NFSv2, see NFSv2.

   3.1. If NFSv4 is selected, additionally enter the appropriate NFSv4 domain name.

   3.2. Make sure the name is the same as the one in the /etc/idmapd.conf file of any NFSv4 client that accesses this particular server. This parameter is for the idmapddaemon that is required for NFSv4 support (on both server and client). Leave it as localdomain (the default) if you do not have any special requirements.

4. Click Enable GSS Security if you need secure access to the server. A prerequisite for this is to have Kerberos installed on your domain and to have both the server and the clients kerberized. Click Next to proceed with the next configuration dialog.

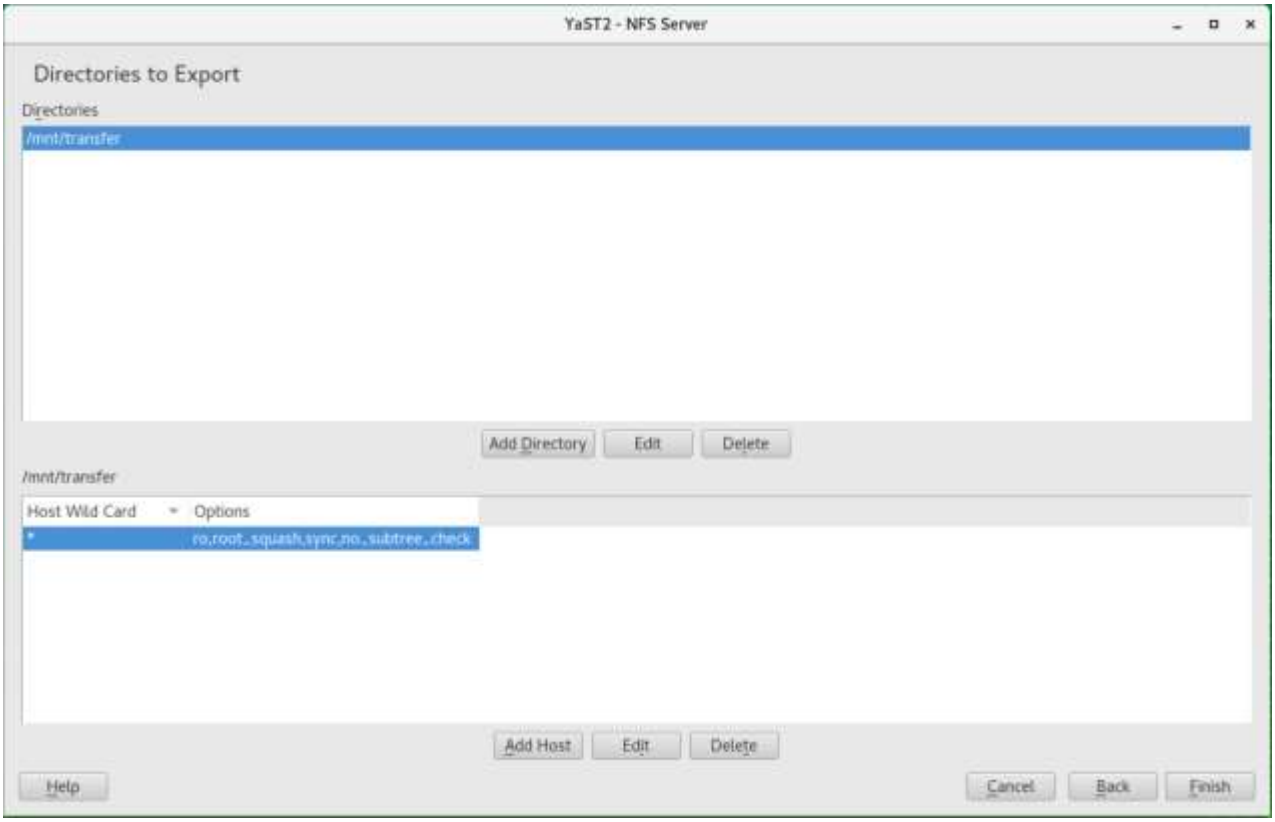
5. Click Add Directory in the upper half of the dialog to export your directory

6. If you have not configured the allowed hosts already, another dialog for entering the client information and options pops up automatically. Enter the host wild card (usually you can leave the default settings as they are).

   There are four possible types of host wild cards that can be set for each host: a single host (name or IP address), netgroups, wild cards (such as * indicating all machines can access the server), and IP networks.

   For more information about these options, see the exports man page.

7. Click Finish to complete the configuration.



4.5. /etc/exports

4.6. Exportfs

4.7. Client Configuration

4.8. Practice: Introduction to NFS

5. Introduction to Networking