

- Partners
- Support
- Community
- Ubuntu.com

- Page History
- Login to edit

Installation/OnNFSDrive



Content Cleanup Required

This article should be cleaned-up to follow the content standards in the Wiki Guide. More info...



Style Cleanup Required

This article does not follow the style standards in the Wiki Guide. More info...

This page describes installing Ubuntu on an NFS-mounted filesystem (instead of on a local hard drive), for example so that client machines can boot the Ubuntu installation over the network using PXE.

The instructions on this page are outdated and incoherent, I advise not following them unless you really know what you are doing.

This configuration has these requirements:

- A pre-configured NFS-server. GNU/Linux on a 500MHz/512MB computer will do fine.
- 100Mbit/s network. The system will work very slowly with 10MBit/s.
- Client with at least 256MB RAM, preferably 512MB, or a hard disk for a swap partition.
- Some Linux experience and ability to compile your own kernel.

Set up the install directory

Create a directory for Ubuntu on the NFS server. This directory will be the root for storing all files needed by the clients, eg:

```
$ sudo mkdir /ubuntu
```

Get debootstrap scripts

Download and install debootstrap using apt-get:

```
# apt-get install debootstrap
```

Alternatively, you may compile from source:

1. Grab the latest debootstrap_*.tar.gz from <http://archive.ubuntulinux.org/ubuntu/pool/main/d/debootstrap/> Save the archive into the /tmp directory.
2. Uncompress and extract the archive, then cd into the newly created directory and build the program:

```
# cd /tmp
# tar xzvf debootstrap_0.2.39ubuntu22.tar.gz
# cd debootstrap-0.2.39ubuntu22
# make
```

Run debootstrap

Tell debootstrap to download a base system into your sub-directory using the following syntax:

```
# debootstrap --arch <architecture> <release> <directory> [<mirror>]
```

For the purpose of this tutorial, the command will be as follows:

```
# debootstrap --arch i386 hoary /ubuntu http://archive.ubuntu.com/ubuntu
```

If you compiled from source, use the following invocation:

```
# DEBOOTSTRAP_DIR="`pwd`"
# ./debootstrap --arch i386 hoary /ubuntu http://archive.ubuntulinux.org/ubuntu
```

debootstrap will now download the base system. If your network is unreliable, you may need to run debootstrap multiple times to get all the files; it resumes where it left off.

Near the end of debootstrap's copious output, you should see "Base system installed successfully". If not, remove everything in /ubuntu apart from the lost+found directory and try again.

Set up the host name

You should replace HOSTNAME in the command below with the desired name for your machine.

```
# echo HOSTNAME > /ubuntu/etc/hostname
```

Set up fstab

Create /ubuntu/etc/fstab. It should include procfs, sysfs and possibly cdrom and fd0. Root is not included.

Here is a decent fstab to get started with:

```
/proc      /proc      proc       defaults   0 0
/sys       /sys       sysfs      defaults   0 0

/dev/fd0    /mnt/floppy auto user,noauto,exec,umask=000 0 0
/dev/cdrom  /mnt/cdrom auto user,noauto,exec,ro    0 0
```

Enter the install environment

```
# chroot /ubuntu
# mount /proc
```

Set the root password:

```
# passwd
Enter new UNIX password:
Retype new UNIX password:
```

Compile kernel for your system

This step is not described fully here. I used kernel 2.6.10 with gentoo patches, just because I had it compiled for my system already. There are some important settings you have to make:

1. Compile your network card driver in the kernel, not as a module.
2. Enable "IP: kernel level autoconfiguration" in Device Drivers -> Networking support -> Networking options
3. Enable NFS in Filesystems -> Network file systems: Enable "NFS File system support" and "Root file system on NFS"

If you get "Warning: Unable to open initial console" on the boot and it hangs, try to also:

1. Enable "Preliminary device filesystem" in Filesystems -> Pseudo filesystems
2. Enable "Mount devfs automatically on boot"

kernel UPDATE: dapper & later users

Since dapper, it is not necessary to recompile the kernel, since initramfs is used on the initial ramdisk. Go to /etc/mkinitramfs/initramfs.conf (in the chroot) and change "BOOT=local" to "BOOT=nfs". Then install any linux-image, the initrd generated will be nfs-bootable. Alternatively, call update-initrd to regenerate initrd if you have kernel already

installed.

Make sure to send **both** kernel image ("/boot/vmlinuz-VERSION") **and** initrd ("/boot/initrd.img-VERSION") when netbooting, in the next step.

Getting out of chroot

In chrooted environment umount /proc and then exit:

```
# umount /proc
# exit
```

Get PXELinux, dhcpd and tftpd

See Installation/LocalNet for more up-to-date details. Old info follows.

For PXELinux, download the Netboot image from the Ubuntu archives, extract it, copy the pxelinux.0 file and the pxelinux.cfg directory to your tftpd root (mine is /ubuntu) and edit the pxelinux.cfg/default file as described as follows.

Configure PXELinux. A decent configuration file would be like:

```
TIMEOUT 50

DEFAULT kernel-2.6.10
APPEND root=/dev/nfs nfsroot=192.168.0.150:/ubuntu ip=dhcp
```

You should at least replace the server address. It is the address of the NFS server. Put pxelinux.0 and kernel-2.6.10 in the tftp server root directory. Put the configuration file "default" in directory "pxelinux.cfg".

Configure dhcpd to provide the boot image. A decent configuration would be like:

```
host netbooted {
    hardware ethernet 00:0C:F1:98:1B:A6;
    fixed-address 192.168.0.204;
    filename "/pxelinux.0";
}
```

I give a fixed address for this to set NFS exports more securely. You should replace the IP and the MAC address of the client. The MAC address can be seen with "ip addr" on computers running Linux. Restart dhcp server to apply the changes.

Configure NFS exports

Add a suitable line in /etc/exports:

```
/ubuntu          10.0.0.0/16(rw,sync,no_root_squash) 10.1.0.1/16(rw,sync,no_root_squash) 192.168.0.0/16(rw,sync,no_root_squash)
```

Restart NFS-server to apply the changes.

Boot

Configure the client to "Boot from network". It should load PXELinux and in 5 seconds start loading Ubuntu.

(ToDo !!) - How To set up the Ubuntu Desktop on this System

How to set up the above system to be exactly like a Ubuntu Desktop Installation on the Client machine, as the above only installs a similar system without the xserver/gnome/etc, and apt-get install ubuntu-desktop does not work out of the box.

A workaround for the time being :

- Setup NFS Server create the directory /ubuntu and export it described as above
- Install Ubuntu Desktop via NetbootImage/CD on Client machine
- Boot into it mount the /ubuntu directory via NFS
- Get a root shell and copy the entire / to the mounted NFS directory
- chroot into the NFS dir, mount proc and uninstall all linux-image-...
- Edit /etc/mkinitramfs/initramfs.conf and /etc/fstab as described above
- Install linux-image
- Configure pxelinux
- Reboot via pxe

CategoryInstallation CategoryNetworking

Installation/OnNFSDrive (zuletzt geändert am 2015-10-20 23:27:21 durch wtanaka @
cpe-66-91-18-153.hawaii.res.rr.com[66.91.18.153];wtanaka)