Create a ram disk for pxe clients to load

Don't lose that backup file

vi /etc/initramfs-tools/initramfs.conf
# initramfs.conf
# Configuration file for mkinitramfs(8). See initramfs.conf(5).
#
# Note that configuration options from this file can be overridden
# by config files in the /etc/initramfs-tools/conf.d directory.
#
# MODULES: [ most | netboot | dep | list ]
# most - Add most filesystem and all harddrive drivers.
# dep - Try and guess which modules to load.
# netboot - Add the base modules, network modules, but skip block devices.
# list - Only include modules from the 'additional modules' list
#
MODULES=netboot
BOOT=nfs

# BUSYBOX: [ y | n | auto ]
# Use busybox shell and utilities. If set to n, klibc utilities will be used.
# If set to auto (or unset), busybox will be used if installed and klibc will
# be used otherwise.
COMPCACHE_SIZE=""

# # COMPRESS: [ gzip | bzip2 | lzma | lzop | xz ]
#

COMPRESS=gzip

# # NFS Section of the config.
#

# DEVICE: ...
# Specify a specific network interface, like eth0
# Overridden by optional ip= bootarg
#

# see what your network device is called before editing this
DEVICE=ens18

# # NFSROOT: [ auto | HOST:MOUNT ]
#

NFSROOT=auto

"/etc/initramfs-tools/initramfs.conf" 75 lines, 1723 characters written
root@dhcp-it3110:~# mkdir /var/lib/tftpboot/diskless1
root@dhcp-it3110:~# mknitramfs -o /var/lib/tftpboot/diskless1/initrd.img
root@dhcp-it3110:~# echo "That created our initial ram disk"
root@dhcp-it3110:~# cp /boot/vmlinux-`uname -r` /var/lib/tftpboot/diskless1/vmlinux
make all required system files available via nfs
To do this, connect to nfs with a client, copy all the client files onto the share
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients. See exports(5).

# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/joeshare        144.38.193.193/26(rw,sync,no_subtree_check)
/classshare       10.150.0.0/16(ro)
/classshare       144.38.193.193/26(ro)

# added for diskless boot
/diskless        144.38.193.0/24(rw,sync,no_subtree_check,no_root_squash)

"/etc/exports" 17 lines, 621 characters written
joe@nfs:~$ sudo mkdir /diskless
joe@nfs:~$ sudo service nfs-kernel-server restart
joe@nfs:~$
You can pick ANY machine to be your NFS client

$ sudo apt install nfs-common
Reading package lists...  Done
Building dependency tree
Reading state information...  Done
The following additional packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-2.1-6 libnfsidmap2 libtirpc1 nfs-common rpcbind
0 upgraded, 6 newly installed, 0 to remove and 81 not upgraded.
Need to get 530 kB of archives.
After this operation, 1,737 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
joe@rootkit:~$ showmount -e nfs.theugmibear.com
Export list for nfs.theugmibear.com:
/diskless  144.38.193.0/24
/classshare 144.38.193.193/26,10.150.0.0/16
/joeshare  144.38.193.193/26
joe@rootkit:~$
joe@rootkit:~$ showmount -e nfs.the gum mibear.com
Export list for nfs.the gum mibear.com:
/diskless 144.38.193.0/24
/classshare 144.38.193.193/26, 10.150.0.0/16
/joeshare 144.38.193.193/26
joe@rootkit:~$ echo "Another test to see if my share is exported"
Another test to see if my share is exported
joe@rootkit:~$
joe@rootkit:$ sudo mkdir /tempmount
joe@rootkit:$ mount -t nfs nfs.thegummibear.com:/diskless /tempmount/
mount: only root can use "--types" option
joe@rootkit:$ sudo mount -t nfs nfs.thegummibear.com:/diskless /tempmount/
joe@rootkit:$
joe@rootkit:~$ mount | grep nfs
nfs.thegummibear.com:/diskless on /tempmount type nfs4 (rw, relatime, vers=4.2, rsize=524288, wsize=524288, namlens=255, hard, proto=tcp, timeo=600, retrans=2, sec=sys, clientaddr=144.38.193.206, local_lock=none, addr=144.38.193.201)
joe@rootkit:~$ echo "Looks like it is mounted"
Looks like it is mounted
joe@rootkit:~$
joe@rootkit:~$ echo "Now going to copy all files from this machine to nfs share"
Now going to copy all files from this machine to nfs share
joe@rootkit:~$ sudo bash
root@rootkit:~# cp -ax /tempmount/
Go back and put your initram file back in correct location

```bash
root@dhcp-it3110:~ # cp /etc/initramfs-tools/initramfs.conf.0 /etc/initramfs-tools/initramfs.conf
```

# D-I config version 2.0
# search path for the c32 support libraries (libcom32, libutil etc.)
path ubuntu-installer/amd64/boot-screens/
include ubuntu-installer/amd64/boot-screens/menu.cfg
default ubuntu-installer/amd64/boot-screens/vesamenu.c32

#prompt 0
timeout 10

label ubuntu
  menu default
  menu label ubuntu
  kernel ubuntu-installer/amd64/linux
  append ks=https://gist.githubusercontent.com/funzoneeq/d77369203ea447dc3cc2/raw/9c4949b0ca87763be3db7643fa5934f1aeed55be8/ubuntu.ks vga=normal initrd=ubuntu-installer/amd64/initrd.gz ramdisk_size=16432 root=/dev/rd/0 rw

label debian
  menu default
  menu label debian
  kernel debian-installer/amd64/linux
  append vga=normal initrd=debian-installer/amd64/initrd.gz ramdisk_size=16432 root=/dev/rd/0 rw
# D-I config version 2.0
# search path for the c32 support libraries (libcom32, libutil etc.)
path ubuntu-installer/amd64/boot-screens/
include ubuntu-installer/amd64/boot-screens/menu.cfg
default ubuntu-installer/amd64/boot-screens/vesamenu.c32
prompt 0
timeout 10
label ubuntu
    menu default
    menu label ubuntu
    kernel ubuntu-installer/amd64/linux
    append ks=https://gist.githubusercontent.com/funzoneq/d77369203ea447dc3cc2/raw/9c494980ca877683e3db7643fa5934f1aed55be8/ubuntu.ks vga=normal initrd=ubuntu-installer/amd64/initrd.gz ramdisk_size=16432 root=/dev/rd/0 rw --label debian
    menu default
    menu label debian
    kernel debian-installer/amd64/linux
    append vga=normal initrd=debian-installer/amd64/initrd.gz ramdisk_size=16432 root=/dev/rd/0 rw --label diskless
    kernel diskless1/vmlinux
    append root=/dev/nfs initrd=diskless1/initrd.img nfsroot=144.38.193.201:/diskless ip= dhcp rw

"pxelinux.cfg/default" 20 lines, 899 characters
root@dhcp-it3110:~/var/lib/tftpboot/diskless1# cd diskless1/
root@dhcp-it3110:~/var/lib/tftpboot/diskless1# ls
initrd.img  vmlinuz
root@dhcp-it3110:~/var/lib/tftpboot/diskless1# ls -la
total 37424
drwxr-xr-x 2 root root 4096 Jan 27 11:29 .
drwxrwxr-x 4 root root 4096 Jan 27 11:28 ..
-rw-r--r-- 1 root root 30008670 Jan 27 11:29 initrd.img
-rw------- 1 root root 8302232 Jan 27 11:29 vmlinuz
root@dhcp-it3110:~/var/lib/tftpboot/diskless1# echo "Maybe fix some permissions"
Maybe fix some permissions
root@dhcp-it3110:~/var/lib/tftpboot/diskless1# chmod 644 *
root@dhcp-it3110:~/var/lib/tftpboot/diskless1#
cp: cannot create directory '/tempmount/boot': Permission denied
joe@rootkit:~$

joe@rootkit:~$ echo "Now going to copy all files from this machine to nfs share"
Now going to copy all files from this machine to nfs share
joe@rootkit:~$ sudo bash
root@rootkit:~# cp -ax /tempmount/
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018/system@5b0490c4355540a1a9b1033a552e4c0a-0000000000002e01-0005902a79a8571.journal': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018/system@5b0490c4355540a1a9b1033a552e4c0a-000000000000005cc-00059ca905912418.journal': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018/user-1000@24d3bef2ad29b4162adee5652de8119a-000000000000002e-0005902a7a91d658.journal': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018/system.journal': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018/user-1000.journal': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/5222d50469bb4fa996c3085874a87018': Operation not supported
cp: preserving permissions for '/tempmount/var/log/journal/': Operation not supported

^C
root@rootkit:~# echo "I hope my copy is done"
I hope my copy is done
root@rootkit:~#
root@rootkit:~# ls
root@rootkit:~# cd /tmpmount/
root@rootkit:/tmpmount# ls
bin home lib media reptile sbin sys usr vmlinuz.old
dev initrd.img lib64 mnt root srv tempmount var
etc initrd.img.old lost+found opt run swapfile tmp vmlinuz
root@rootkit:/tmpmount# echo "double check that we have what we need"
"double check that we have what we need"
root@rootkit:/tmpmount#
root@rootkit:/tempmount# ls dev/
root@rootkit:/tempmount# cp -ax /dev/* dev/
cp: preserving permissions for ‘dev/sg0’: Operation not supported
cp: preserving permissions for ‘dev/snd/timer’: Operation not supported
cp: preserving permissions for ‘dev/snd/seq’: Operation not supported
cp: preserving permissions for ‘dev/sr0’: Operation not supported
root@nfs:/diskless# vi etc/fstab
/etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=9babeace-ed9c-486e-a21d-7670e6b74046 /
    ext4 errors=remount-ro 0 1
/swapfile
    none swap sw 0 0
#deleted everything
/dev/nfs        /     nfs     defaults     1     1

"etc/fstab" 2 lines, 48 characters written
root@nfs:/diskless# vi etc/fstab
root@nfs:/diskless# echo "now testboot a vm"
now testboot a vm
root@nfs:/diskless#
Installer boot menu

Install
Command-line install
Advanced options >
Help
ubuntu
debian
diskless

Press ENTER to boot or TAB to edit a menu entry
Starting Network Name Resolution...
[FAILED] Failed to start Network Name Resolution.
See 'systemctl status systemd-resolved.service' for details.
[ OK ] Stopped Network Name Resolution.
[ OK ] Reached target Host and Network Name Lookups.
[ OK ] Reached target Network.
[FAILED] Failed to start Network Time Synchronization.
See 'systemctl status systemd-timesyncd.service' for details.
[ OK ] Stopped Network Time Synchronization.
[FAILED] Failed to start Network Time Synchronization.
See 'systemctl status systemd-timesyncd.service' for details.
[ OK ] Reached target System Initialization.
[ OK ] Listening on UUID daemon activation socket.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
Starting System Logging Service...
Starting OpenBSD Secure Shell server...
Starting LSB: Record successful boot for GRUB...
Starting Accounts Service...
Starting Dispatcher daemon for systemd-networkd...
root@nfs:/diskless# vi etc/fstab
root@nfs:/diskless# echo "now test boot a vm"

root@nfs:/diskless# ip a
1: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 42:2c:21:fc:5a:c4 brd ff:ff:ff:ff:ff:ff
   inet 144.38.193.201/26 brd 144.38.193.255 scope global ens18
     valid_lft forever preferred_lft forever
   inet6 fe80::40c2:21ff:fefc:5ac4/64 scope link
     valid_lft forever preferred_lft forever

root@nfs:/diskless# ls
bin  home  lib  media  reptile  sbin  sys  usr  vmlinuz.old
dev  initrd.img  lib64  mnt  root  srv  tempmount  var
etc  initrd.img.old  lost+found  opt  run  swapfile  tmp  vmlinuz

root@nfs:/diskless# vi etc/netplan/01-netcfg.yaml
root@nfs:/diskless# echo "Make sure the vm will boot dhcp"

Make sure the vm will boot dhcp