Mandriva Linux: HOWTO Clone a system using KA method

Antoine Ginies

Mandriva Linux: HOWTO Clone a system using KA method

by Antoine Ginies

Published 2007

Revision History

Revision 0.3 May 2010 Revised by: ag
update/fix
Revision 0.2 Fev 2010 Revised by: ag
update
Revision 0.1 OCT 2007 Revised by: ag
update

Table of Contents

.....1

CLONING WILL ERASE ALL CLIENT NODES DATA ! 1
Clone a computer over the network 1
KA method 1
HOW it works 2
Steps 2
Needed files 2
Step 1: PXE, TFTP, DHCPD services 2
PXE parameters on server 2
TFTP server 3
PXE configuration 4
DHCPD configuration 4
Setup a node as a golden node 6
The rescue.sqfs file 6
ka-d.sh 7
replication.conf 7
fdisk_to_desc 7
gen_modprobe_conf.pl 7
ka-d-client 7
ka-d-server 8
ka_replication.sh 8
store_log.sh 8
bootable_flag.sh 8
make_initrd_grub 8
make_initrd_lilo 8
prepare_node.sh 8
send_status.pl 8
status_node.pl 8
The golden node, KA server 9
KA client node 10
PXE server (kamethod) 10
Stage1 KA method, node waiting stage2 10
Stage2, the duplication process 10
Prepare the node 10
PXE server to local boot 11
Step by step from scratch KA duplication 11
Golden node side 11
KA client side 23
Post duplication process 28

Clone a node/computer using KA method

CLONING WILL ERASE ALL CLIENT NODES DATA !

!! USE WITH CARE !!

Clone a computer over the network

Goal of duplication is to easily deploy a computer over network without taking care of numbers of computer. In this documentation, we call golden node the node we want to clone. We can duplicate SCSI or IDE hard drive, and duplication support multiple filesystem (reiserfs, ext2, ext3, ext4, xfs, jfs). This method came from a very old project called CLIC, and was used under IGGI project, all Mandrake Clustering products, and now it is used under XtreemOS project. Now it should be available in 2010 spring, and all futur product.

WARNING: all data on client nodes will be ERASED ! We duplicate partitions of HDD's golden node, and the process will do an fdisk command on the client node, so ALL YOUR DATA will be erased on client nodes.

KA method

With KA method you can quickly duplicate a node using a **desc** file describing partitions. KA method only duplicate data on partitions, so if you have 80go HDD disk, and only 10go on it, KA only duplicates 10go, and not the whole disk. KA method doesn't not support RAID software.

Drawbacks:

- KA method doesn't support RAID software (use dolly to do that)
- all data on client nodes are erased
- you need a PXE, DHDCP and TFTP server
- you must re-create same partition table as the golden node (even if size can differ)
- even if it has been tested, it's still an experimental method
- cloning script are old, and need a full rewrite
- now it's only works with the Mandriva installer (need to patch it to support a KA method)
- if a node crash while doing a duplication, the duplication process stop (or became very unstable)
- using fdisk to erase and re-format the HDD is not a good way to proceed
- UUID support is not really done (fstab use old /dec/sdX)

- you can only clone Linux filesystems (if you want to duplicate another kinf of FS, it's up to you to modify the scripts)
- of course various other things !

HOW it works

Steps

The clone process works in three steps

- **PXE boot to retrieve stage1**: the computer boot on PXE mode, retrieve **vmlinuz** and an **initrd** image. The computer is in **stage1** mode, and is able to get the stage2 throug KA. Network is up.
- **get stage2**: the computer gets the stage2 with the KA method. The **stage2** contains all necessary tools to recognize your hardware (the most important things is to detect your HDD and your network card), and all necessary tools/scripts to finalize the cloning process.
- **Duplication process**: the computer auto-probes needed modules to be able to access the HDD. A basic log server is launched on the client node to be able to run command and get status of the KA duplication process. The computer reconfigure the modprobe.conf and restore the booloader (grub or lilo)

Needed files

All needed files are available in Mandriva Linux cooker.

- **install/stage2/rescue.sqfs**: this is the stage2 file with all needed files to detect and probe modules, and launch the third step of the duplication process. This file will be used on the golden node.
- **isolinux/alt0/vmlinuz**: linux kernel, needed in the **/var/lib/tftpboot/X86PC/linux/images/** directory of the PXE server
- **isolinux/alt0/all.rdz**: stage1 and all needed modules and tools.

Step 1: PXE, TFTP, DHCPD services

To easily clone a computer node, we use PXE technology to boot a **kernel**, and an **initrd** image wich contains all needed modules for network and media storage. Documentation about PXE can be found here: PXE doc (http://people.mandriva.com/~aginies/doc/pxe/). Please, keep in mind setting such services can **DISTURB** your current network architecture.

PXE parameters on server

Mandriva Linux installer supports various methods to install a computer. With PXE configuration file you can specify wich method you want to use to install your node, or add a specific option at boot prompt. Edit your default PXE configuration file to add your custom entry (**/var/lib/tftpboot/X86PC/linux/pxelinux.cfg/default**).

```
PROMPT 1
DEFAULT local
DISPLAY messages
TIMEOUT 50
F1 help.txt

label local
    LOCALBOOT 0

label kamethod
    KERNEL images/vmlinuz
    APPEND initrd=images/all.rdz ramdisk_size=64000 vga=788 \
       automatic=method:ka,interface:eth0,network:dhcp root=/dev/ram3 rw kamethod
```

At boot prompt no you can boot:

- **DEFAULT local**: default boot will be local one, change it with the name of a **LABEL**
- **local**: boot local
- **kamethod**: automatic mode, get stage2 through **KA**. Network interface is set to eth0. Auto setup the network with DHCP, and use the KA technology to launch the replication method.

TFTP server

TFTP server should be activated in **/etc/xinetd.d/tftp** file, and the **xinetd** service started.

```
service tftp
{
     disable= no
     socket_type= dgram
     protocol= udp
     wait= yes
     user= root
     server= /usr/sbin/in.tftpd
     server_args = -s /var/lib/tftpboot
     per_source= 11
     cps= 100 2
     flags= IPv4
}
```

PXE configuration

```
# which interface to use
interface=eth0
default_address=IPADDR_PXE

# the multicast ip address to listen on
multicast_address=224.0.1.2

# mtftp info
mtftp_address=IPADDR_TFTP
mtftp_client_port=1758
mtftp_server_port=1759

# the port to listen on
listen_port=4011

# enable multicast?
use_multicast=1

# enable broadcast?
use_broadcast=0

# user prompt
prompt=Press F8 to view menu ...
prompt_timeout=2

# what services to provide, priority in ordering
# CSA = Client System Architecture
# service=<CSA>,<min layer>,<max layer>,<basename>,<menu entry>
service=X86PC,0,2,linux,Mandriva Linux x86
service=IA64PC,0,2,linux,Mandriva Linux IA64
service=X86PC,0,0,local,Local boot

# tftpd base dir
tftpdbase=/

# domain=guibland.com
domain=
```

DHCPD configuration

IE of an **/etc/dhcpd.conf** configuration file. Change **IPADDR_TFTP** with the IP address of the TFTP serrver, and the **NET** value. Don't forget to adjust the **domain-name** and the **domain-name-servers**.

```
ddns-update-style none;
```

```
allow booting;
allow bootp;

authoritative;

# Definition of PXE-specific options
# Code 1: Multicast IP address of bootfile
# Code 2: UDP port that client should monitor for MTFTP responses
# Code 3: UDP port that MTFTP servers are using to listen for MTFTP requests
# Code 4: Number of secondes a client must listen for activity before trying
#         to start a new MTFTP transfer
# Code 5: Number of secondes a client must listen before trying to restart
#         a MTFTP transfer

# define Option for the PXE class
option space PXE;
option PXE.mtftp-ip code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr code 7 = ip-address;

#Define options for pxelinux
option space pxelinux;
option pxelinux.magic code 208 = string;
option pxelinux.configfile code 209 = text;
option pxelinux.pathprefix code 210 = text;
option pxelinux.reboottime code 211 = unsigned integer 32;
site-option-space "pxelinux";

option pxelinux.magic f1:00:74:7e;
option pxelinux.reboottime 30;

#Class that determine the options for Etherboot 5.x requests
class "Etherboot" {
#if The vendor-class-identifier equal Etherboot-5.0
match if substring (option vendor-class-identifier, 0, 13) = "Etherboot-5.0";
# filename define the file retrieve by the client, there nbgrub
# our tftp is chrooted so is just the path to the file
filename "/etherboot/nbgrub";
#Used by etherboot to detect a valid pxe dhcp server
option vendor-encapsulated-options 3c:09:45:74:68:65:72:62:6f:6f:74:ff;
# Set the  "vendor-class-identifier" field to "PXEClient" in dhcp answer
# if this field is not set the pxe client will ignore the answer !
option vendor-class-identifier "Etherboot-5.0";
vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;
# IP of you TFTP server
next-server IPADDR_TFTP;
}
```

```
# create the Class PXE
class "PXE" {
# if the "vendor-class-identifier" is set to "PXEClient" in the client dhcp request
match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";
filename "/X86PC/linux/linux.0";
option vendor-class-identifier "PXEClient";
vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;
next-server IPADDR_TFTP;
}

#host node20 {
#    hardware ethernet 00:40:CA:8C:B6:E9;
#    fixed-address node20;
#}

subnet NET.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option routers IPADDR_GW;
  default-lease-time 288000;
  max-lease-time 864000;
  option domain-name "guibland.com";
  option domain-name-servers IPADDR_DNS;
  next-server IPADDR_TFTP;
  pool {
  range NET.30 NET.40;
  }
}
```

Setup a node as a golden node

The rescue.sqfs file

You need the rescue disk (wich contains the **/ka** directory), Just extract this file, and copy all directory in **/mnt/ka**.

```
[root@guibpiv ~]# mkdir /mnt/ka
[root@guibpiv ~]# cd /mnt/ka/
[root@guibpiv ka]# unsquashfs rescue.sqfs
[root@guibpiv ka]# mv squashfs-root/* .
[root@guibpiv ka]# ls
bin/  dev/  etc/  ka/  lib/  modules/  proc/  sbin/  squashfs-root/  tmp/  usr/  var/
```

Go in the **/mnt/ka/ka** directory, and see all new files available. All those files are needed to do a **KA** duplication process. We will explain now the rule of each of them. You can modify all them, those files will be copied in the directory **/tmp/stage2** of the client node of the duplication process (second step).

ka-d.sh

This is the master script to declare a node as a golden node. This script takes a lot of arguments. This script should be run on the host wich have the **/mnt/ka** directory.

```
-h, --help : display this message
-n num : specify the number of (destination) nodes
-x dir : exclude directory
-X sdb|sdc : exclude sdb for the replication
-m drive : copy the master boot record (for windows) of this drive (not really tested y
-M drive file : use 'file' as master boot record (must be 446 bytes long) for the speci
-D partition : also copy partition 'partition'
-p drive pdesc : use 'pdesc' file as partition scheme (see doc) for the specified drive
-d delay : delay beteween the release of 2 clients (1/10 second)
-r 'grub|lilo' : choose the bootloader (you can add mkinitrd options)

ie: ka-d.sh -n 3 -p sda /tmp/desc -X 'sdb|sdc' -r 'grub --with=ata_piix --with=piix'
```

replication.conf

This file contains all variables needed by other scripts. It also tries to get information like IP address.

fdisk_to_desc

This script generate the description table of the hard drive disk in the **/tmp/desc** file. This file must follow some rules: one line per partition, with two fields : type of partition and size in megabytes. The type can be linux, swap, extended. Other types can be obtained by appending their hexadecimal number to 'type'. For example linux is the same as type83. The size is either a number of megabytes, or the keyword fill (to take all available space). The logical partitions must have the logical keyword. Do a **man ka-d** for more help.

gen_modprobe_conf.pl

This script creates a basic output like the content of the**/etc/modprobe.conf** file. Drawbacks this file must be updated for each new modules available in the kernel (based on the **kernel/list_modules.pm** file).

ka-d-client

The **ka-d-client** binary file is used to get stage2 with the **KA** method, and after get the whole system. The important argument is the **-s** session name. A **KA** can only connect to a specific session (getstage2, kainstall ...). The code source is available in the ka-deploy SRPM.

ka-d-server

The **ka-d-server** binary file is used to be a **KA** golden node server. Like the **ka-d-client** the session arguments is an important parameter (**-s session_name**). The session name will be **getstage2** to retrieve the stage2 (after the PXE boot) and will be **kainstall1** at duplication process step. If you want to do more than one duplication process of nodes at the same time, you should synchronize the ka_sesion name between the server and the client. The code source is available in the ka-deploy SRPM.

ka_replication.sh

Script launched on the **KA** client (after getting stage2 and probing modules), to do the full process of the **Ka** duplication. This script call other scripts to prepare the node (prepare_node.sh), configure the bootloader (make_initrd_grub or make_initrd_lilo).

store_log.sh

Basic script to store the log of the **KA** duplication process on an FTP server. Adjust to feet your need, and uncomment the line **#store_log.sh** in the **/mnt/ka/ka/ka_replication.sh** file.

bootable_flag.sh

Script to set bootable an HDD using fdisk. First arg must be the HDD device.

make_initrd_grub

Restore and reload the Grub bootloader in the **/mnt/disk** directory. It's a very basic script, and perhaps use the **restore_bootloader** of the Mandriva Linux Rescue should be a better idea.

make_initrd_lilo

Restore and reload the lilo bootloader in the **/mnt/disk** directory. Again it's a very basic script, perhaps we should use the **restore_bootloader** of the Mandriva Linux Rescue.

prepare_node.sh

This script remove in the futur system the old network's udev rules, old dhcp cache files, launch the script **gen_modprobe_conf.pl** to regenerate an up to date **/etc/modprobe.conf** in the new system, and launch the script to restore the bootloader. If you want to do more action on the installed, system, you can modify this script.

send_status.pl

Very basic perl script to open the port 12345, and paste the content of the **/tmp/ka*** file. It also permit the execution of commands on node, if user send a message from the golden node with the **exec** prefix.

status_node.pl

Script to connect to a client node, first arg must be the IP address of the node. You can run command on the node with the **exec** prefix.

The golden node, KA server

Now, it is time to build a description of the node partitions. You can use the script **/mnt/ka/ka/fdisk_to_desc** as root user, or your favorite text editor, you can write a file like this one:

```
linux 3500
extended fill
logical swap 500
logical linux fill
```

This file describes your partition table and the sample above can be considered as a default one for a recommended installation. There is a 3.5GB **/** partition, a 500 MB swap partition, and **/var** fills the rest, of course you can adjust sizes accoding to your system.

Type the following to start the ka replication server as root user on the golden node:

```
      <screen>
[root@node40 ka]# ./ka-d.sh -n 1 -p sda /root/desc -X sdb -r "grub --with=jfs --with=ata_pi
takembr =
desc = sda /root/desc
+ Mount points :
     /dev/sda5 / ext3
     /dev/sda1 swap swap
+ Hard drives :
     sda
+ Reading partition table description for sda
    Added partition 1 : type 82
    Added partition 5 : type 83
+ Included mount points : /
+ Bootloader is: grub --with=jfs --with=ata_piix
+++ Sending Stage2 +++
Compiled : Aug 23 2007 12:58:29
ARGS=+ka-d-server+-s+getstage2+-n+1+-e+(cd /mnt/ka; tar --create --one-file-system --sparse
Server IP = 10.0.1.40
command = (cd /mnt/ka; tar --create --one-file-system --sparse  . )
I want 1 clients
Socket 4 on port 30765 on node40.guibland.com ready.
Socket 5 on port 30764 on node40.guibland.com ready.
```

- **-r "grub --with=jfs --with=ata_piix"**: use grub bootloader and **--with=jfs --with=piix** mkinitrd option in the chrooted system after the **KA** deploiement
- **-n nb_nodes**: specify how many nodes are clients
- **-p sda desc**: specify the name of the hdd
- **-x /tmp**: exclude **/tmp** directory
- **-X sdb**: exclude **sdb** hdd for the duplication

Now the golden node is waiting for clients nodes to start replication.

KA client node

PXE server (kamethod)

We have to configure the PXE to boot by default on **kamethod**. To do this just edit **/var/lib/tftpboot/X86PC/linux/pxelinux.cfg/default** and set **DEFAULT** to kamethod:

```
DEFAULT kamethod
```

So, next time a node boots, the PXE server will force the node to boot using the kamethod entry.

Stage1 KA method, node waiting stage2

Now, you boot all remaining nodes. The replication process will start once all nodes are up and waiting on the **KA** screen.

If the nodes can't reach the golden node, running the **KA** server the message **Can't reach a valid KA server** will appear. Each node will try five times to reach the **KA** server, after that the node will reboot. As the node boots on **kamethod**, it will retry until it finds it.

Stage2, the duplication process

Once all the nodes have found the **KA** server, the first duplication process will start. This step duplicates the **stage2** from the **/mnt/ka** directory of the golden node, in the client's nodes memory (**/dev/ram3** formated as ext2). Then, nodes chroot their memories (the **/tmp/stage2** directory), and launch the **drvinst** command from the stage2, to probe all needed their modules (drivers). Then, the second step of the duplication starts.

The duplication process will clone your drives following the description you have made (**/tmp/desc** of the golden node). Nodes will rewrite their partition table, then format their filesystems (ReiserFs, XFS, ext2/3/4, JFS). All new partitions will be mounted in the **/mnt/disk** directory. Then, the drive duplication process will begin. On a fast Ethernet switch you can reach speeds of 10MBytes/sec.

Prepare the node

At the end of the duplication process, each node will chroot its partitions and rebuild its **/boot/initrd.img**, and **/etc/modprobe.conf** files. This step ensures that your node will reboot using its potential SCSI drives and adjusting its network card driver. Before rebooting, each node reinstalls lilo/grub. All your node are now ready, and are clone of master node.

PXE server to local boot

Don't forget to change the default PXE boot to **local** so node after replication will boot localy.

Step by step from scratch KA duplication

We will use a PIV 3gz box as golden node, with a SATA hard drive, and an Intel 82540EM Gigabit Ethernet Controller card. This golden box will be the: PXE, DHCPD, TFTP server. Client nodes are

- basic PIV 2.8gz, with a Realtek Semiconductor 8139 network card, and a IDE hard drive disk
- PE2650 dual XEON 2.4gz, SCSI Hard Drive disk, and NetXtreme BCM5701 Gigabit Ethernet cards

Both nodes are configured to boot on their network card.

Golden node side

Prepapre the golden node, install all needed tools.

```
[root@localhost ~]# urpmi ka-deploy-source-node
    http://192.168.1.253/cooker/i586/media/main/release/ka-deploy-source-node-0.94.1-1mdv20
installing ka-deploy-source-node-0.94.1-1mdv2010.1.i586.rpm from /var/cache/urpmi/rpms
Preparing...                ########################################################
      1/1: ka-deploy-source-node ########################################################

[root@localhost ~]# rpm -ql ka-deploy-source-node
/etc/ka
/etc/ka/replication.conf
/usr/bin/bootable_flag.sh
/usr/bin/fdisk_to_desc
/usr/bin/gen_modprobe_conf.pl
/usr/bin/ka-d-client
/usr/bin/ka-d-server
/usr/bin/ka-d.sh
/usr/bin/ka_replication.sh
/usr/bin/make_initrd_grub
/usr/bin/make_initrd_lilo
/usr/bin/prepare_node.sh
```

```
/usr/bin/send_status.pl
/usr/bin/status_node.pl
/usr/bin/store_log.sh
/usr/bin/udev_creation.sh
/usr/share/ka-deploy-0.94.1
/usr/share/man/man1/ka-d-client.1.lzma
/usr/share/man/man1/ka-d-server.1.lzma
/usr/share/man/man1/ka-d.1.lzma
/usr/share/man/man1/ka-d.sh.1.lzma
/usr/share/man/man1/ka-deploy.1.lzma
```

Create the /mnt/ka directory, and put all stuff in it (this directory will be sent to all client nodes and use to finish the duplication process)

```
[root@localhost ~]# mkdir /mnt/ka
lftp ftp.proxad.net:~> cd pub/Distributions_Linux/MandrivaLinux/devel/cooker/i586/install/s
lftp ftp.proxad.net:/pub/Distributions_Linux/MandrivaLinux/devel/cooker/i586/install/stage2
19132416 bytes transferred in 78 seconds (241.1K/s)

[root@localhost ~]# urpmi squashfs-tools
    http://192.168.1.253/cooker/i586/media/main/release/squashfs-tools-4.0-3.20091221.1mdv2
installing squashfs-tools-4.0-3.20091221.1mdv2010.1.i586.rpm from /var/cache/urpmi/rpms
Preparing...                ###########################################################
      1/1: squashfs-tools   ###########################################################

[root@localhost ~]# unsquashfs rescue.sqfs
Parallel unsquashfs: Using 2 processors
988 inodes (1222 blocks) to write
[=============================================================================================
created 550 files
created 93 directories
created 60 symlinks
created 371 devices
created 1 fifos

[root@localhost ~]# cd squashfs-root/
[root@localhost squashfs-root]# ls
bin/  dev/  etc/  ka/  lib/  modules/  proc/  sbin/  tmp/  usr/  var/
[root@localhost squashfs-root]# mv * /mnt/ka/
```

Install all needed packages to be able to be a PXE, DHCPD and TFTP server

```
[root@localhost ka]# ka-d.sh -h
/usr/bin/ka-d.sh : clone this machine
Usage:
 -h, --help : display this message
 -n num : specify the number of (destination) nodes
 -x 'dir|dir2' : exclude directory
        -X 'sdb|sdc' : exclude sdb for the replication
```

```
       -m drive : copy the master boot record (for windows) of this drive
 -M drive file : use 'file' as master boot record (must be 446 bytes long) for the specifie
 -D partition : also copy partition 'partition'
 -p drive pdesc : use 'pdesc' file as partition scheme (see doc) for the specified drive
 -d delay : delay beteween the release of 2 clients (1/10 second)
 -r 'grub|lilo' : choose the bootloader (you can add mkinitrd options)

 ie: ka-d.sh -n 3 -p sda /tmp/desc -X sdb -r 'grub --with=ata_piix --with=piix'

[root@localhost ka]# urpmi ka-deploy-server
To satisfy dependencies, the following packages are going to be installed:
   Package                          Version      Release       Arch
(medium "Main")
  bind-utils                       9.7.0        4mdv2010.1    i586
  clusterscripts-common            3.5          1mdv2010.1    noarch
  clusterscripts-server-conf       3.5          1mdv2010.1    noarch
  clusterscripts-server-pxe        3.5          1mdv2010.1    noarch
  dhcp-server                      4.1.1        5mdv2010.1    i586
  ka-deploy-server                 0.94.1       1mdv2010.1    i586
  perl-Crypt-PasswdMD5             1.300.0      1mdv2010.1    noarch
  pxe                              1.4.2        19mdv2010.1   i586
  pxelinux                         3.83         1mdv2010.1    i586
  syslinux                         3.83         1mdv2010.1    i586
  tftp-server                      5.0          4mdv2010.1    i586
  xinetd                           2.3.14       11mdv2010.1   i586
12MB of additional disk space will be used.
2.5MB of packages will be retrieved.
Proceed with the installation of the 12 packages? (Y/n)
```

Configure all services

```
[root@localhost ~]# hostname
node42.guibland.com
[root@localhost ~]# domainname
guibland.com

[root@localhost ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 100
    link/ether 00:17:31:19:a0:78 brd ff:ff:ff:ff:ff:ff
    inet 10.0.1.42/24 brd 10.0.1.255 scope global eth0
    inet6 fe80::217:31ff:fe19:a078/64 scope link
       valid_lft forever preferred_lft forever

[root@localhost ~]# vi /etc/pxe.conf
-----------
# which interface to use
interface=eth0
default_address=10.0.1.42

# the multicast ip address to listen on
multicast_address=224.0.1.2
```

```
# mtftp info
mtftp_address=10.0.1.42
mtftp_client_port=1758
mtftp_server_port=1759

# the port to listen on
listen_port=4011

# enable multicast?
use_multicast=1

# enable broadcast?
use_broadcast=0

# user prompt
prompt=Press F8 to view menu ...
prompt_timeout=2

# what services to provide, priority in ordering
# CSA = Client System Architecture
# service=<CSA>,<min layer>,<max layer>,<basename>,<menu entry>
service=X86PC,0,2,linux,Mandriva Linux x86
service=IA64PC,0,2,linux,Mandriva Linux IA64
service=X86PC,0,0,local,Local boot

# tftpd base dir
tftpdbase=/

# domain name
domain=guibland.com
---------------

[root@localhost ~]# vi /etc/xinetd.d/tftp
service tftp
{
        disable = no
        socket_type             = dgram
        protocol                = udp
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -s /var/lib/tftpboot
        per_source              = 11
        cps                             = 100 2
        flags                   = IPv4
}


[root@localhost ~]# cp /etc/dhcpd.conf.pxe.single /etc/dhcpd.conf
cp: overwrite ‘/etc/dhcpd.conf’? y

[root@localhost ~]# cat /etc/resolv.conf
```

```
nameserver 10.0.1.253
search guibland.com

[root@localhost ~]# cat /etc/dhcpd.conf
# for explanation in french go to : http://www.delafond.org/traducmanfr/man/man5/dhcpd.conf
ddns-update-style none;
allow booting;
allow bootp;

# Your dhcp server is not master on your network !
#not authoritative;
# Your dhcpd server is master on your network !
#authoritative;
authoritative;

#Interface where dhcpd is active
#DHCPD_INTERFACE = "eth0";

# Definition of PXE-specific options
# Code 1: Multicast IP address of bootfile
# Code 2: UDP port that client should monitor for MTFTP responses
# Code 3: UDP port that MTFTP servers are using to listen for MTFTP requests
# Code 4: Number of secondes a client must listen for activity before trying
#         to start a new MTFTP transfer
# Code 5: Number of secondes a client must listen before trying to restart
#         a MTFTP transfer

# define Option for the PXE class
option space PXE;
option PXE.mtftp-ip code 1 = ip-address;
option PXE.mtftp-cport code 2 = unsigned integer 16;
option PXE.mtftp-sport code 3 = unsigned integer 16;
option PXE.mtftp-tmout code 4 = unsigned integer 8;
option PXE.mtftp-delay code 5 = unsigned integer 8;
option PXE.discovery-control code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr code 7 = ip-address;

#Define options for pxelinux
option space pxelinux;
option pxelinux.magic      code 208 = string;
option pxelinux.configfile code 209 = text;
option pxelinux.pathprefix code 210 = text;
option pxelinux.reboottime code 211 = unsigned integer 32;
site-option-space "pxelinux";
# These lines should be customized to your setup
#option pxelinux.configfile "configs/common";
#option pxelinux.pathprefix "/pxelinux/files/";
#filename "/pxelinux/pxelinux.bin";

option pxelinux.magic f1:00:74:7e;
option pxelinux.reboottime 30;
#if exists dhcp-parameter-request-list {
# Always send the PXELINUX options
```

```
# append dhcp-parameter-request-list 208, 209, 210, 211;
# append dhcp-parameter-request-list 208,211;
#     }

#Class that determine the options for Etherboot 5.x requests
class "Etherboot" {

#if The vendor-class-identifier equal Etherboot-5.0
match if substring (option vendor-class-identifier, 0, 13) = "Etherboot-5.0";

# filename define the file retrieve by the client, there nbgrub
# our tftp is chrooted so is just the path to the file
filename "/etherboot/nbgrub";

#Used by etherboot to detect a valid pxe dhcp server
option vendor-encapsulated-options 3c:09:45:74:68:65:72:62:6f:6f:74:ff;

# Set the  "vendor-class-identifier" field to "PXEClient" in dhcp answer
# if this field is not set the pxe client will ignore the answer !
option vendor-class-identifier "Etherboot-5.0";

vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;

# IP of you TFTP server
next-server 10.0.1.42;
}


# create the Class PXE
class "PXE" {
# if the "vendor-class-identifier" is set to "PXEClient" in the client dhcp request
match if substring(option vendor-class-identifier, 0, 9) = "PXEClient";

# filename define the file retrieve by the client, there pxelinux.0
# our tftp is chrooted so is just the path to the file
# If you prefer use grub, use pxegrub compiled for your ethernet card.
#filename "/PXEClient/pxegrub";
filename "/X86PC/linux/linux.0";

# Set the  "vendor-class-identifier" field to "PXEClient" in dhcp answer
# if this field is not set the pxe client will ignore the answer !
option vendor-class-identifier "PXEClient";


vendor-option-space PXE;
option PXE.mtftp-ip 0.0.0.0;

# IP of you TFTP server
next-server 10.0.1.42;
}

# Tags uses by dhcpnode and setup_add_nodes_to_dhcp
```

```
# TAG: NODE_LIST_ADMIN_BEGIN

# TAG: NODE_LIST_ADMIN_END

# TAG: MY_ADMIN_BEGIN
subnet 10.0.1.0 netmask 255.255.255.0 {
  option subnet-mask 255.255.255.0;
  option routers 10.0.1.253;
  default-lease-time 288000;
  max-lease-time 864000;
  option domain-name "guibland.com";
  option domain-name-servers  10.0.1.253;
  next-server 10.0.1.42;

  pool {
    range 10.0.1.110 10.0.1.120;
  }
}

# TAG: MY_ADMIN_END


[root@localhost ~]# service xinetd restart
Stopping xinetd
Starting xinetd
[root@localhost ~]# service pxe restart
Stopping PXE server
Dhcp server is not running on this machine !
Be sure that a valid PXE Dhcp server is running on your network
Starting PXE server
[root@localhost ~]# service dhcpd restart
Shutting down dhcpd:
Starting dhcpd:
```

KA listen only listen on eth0, and need a FQDN. So if it is not the case, ka-d-server will try to open a port on 0.0.0.0 IP address, wich cause an error. You can fix it easely setting an valid hostname in /etc/hosts file. Don't forget to kill ka-d-server with crtl+C key, after testing it will open a port on a valid IP address.

```
[root@node42 ~]# ka-d-server
Compiled : May  4 2010 20:33:07
ARGS=+ka-d-server+
Server IP = 0.0.0.0
command = (cd /; tar  --create --one-file-system --sparse /)
I want 1 clients
ka-d-server: server.c:1987: main: Assertion 'socket_server >=0' failed.
Aborted


[root@node42 ~]# cat /etc/hosts
127.0.0.1 localhost.localdomain localhost
10.0.1.42 node42.guibland.com
```

```
[root@node42 ~]# ka-d-server
Compiled : May  4 2010 20:33:07
ARGS=+ka-d-server+
Server IP = 10.0.1.42
command = (cd /; tar  --create --one-file-system --sparse /)
I want 1 clients
Socket 3 on port 30765 on node42.guibland.com ready.
Socket 4 on port 30764 on node42.guibland.com ready.
[root@node42 ~]# ^C
```

We need to describe the partition table of our golden node, to send it to client nodes.

```
[root@node42 ~]# fdisk -l

Disk /dev/sda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd9b576f2

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        1019     8185086   82  Linux swap / Solaris
/dev/sda2            1020        4843    30716280   83  Linux
/dev/sda3            4844        9729    39246795    5  Extended
/dev/sda5            4844        9729    39246763+  83  Linux


[root@node42 ~]# fdisk_to_desc
 -devices: sda1 -size en Mo: 7993 -filesystem: Linux
Use of uninitialized value $e in concatenation (.) or string at /usr/bin/fdisk_to_desc line
 -devices: sda2 -size en Mo: 29996 -filesystem: Linux
Use of uninitialized value $e in concatenation (.) or string at /usr/bin/fdisk_to_desc line
 -devices: sda3 -size en Mo: 38326 -filesystem: Extended
 -devices: sda5 -size en Mo: 38326 -filesystem: Linux
Desc file is /tmp/desc
[root@node42 ~]# cat /tmp/d
ddebug.log  desc
[root@node42 ~]# cat /tmp/desc
swap 7993
linux 29996
extended 38326
logical linux 38326

[root@node42 ~]# cat /tmp/desc
swap 7993
linux 29996
extended fill
logical linux fill
```

Set default PXE boot to kamethod

```
[root@node42 ~]# cat /var/lib/tftpboot/X86PC/linux/pxelinux.cfg/default
PROMPT 1
DEFAULT kamethod
DISPLAY messages
TIMEOUT 50

label local
 LOCALBOOT 0

label kamethod
    KERNEL images/vmlinuz
    APPEND initrd=images/all.rdz ramdisk_size=64000 vga=788 automatic=method:ka,interface:e


[root@node42 ~]# cd /var/lib/tftpboot/X86PC/linux/
[root@node42 linux]# ls
help.txt  linux.0  memdisk  messages  pxelinux.cfg/
[root@node42 linux]# mkdir images

lftp ftp.proxad.net:/pub/Distributions_Linux/MandrivaLinux/devel/cooker/i586/isolinux/alt0>
-rw-r--r--    1 ftp      ftp      15613654 Apr 23 17:26 all.rdz
-rw-r--r--    1 ftp      ftp       2279584 Apr 23 17:26 vmlinuz
lftp ftp.proxad.net:/pub/Distributions_Linux/MandrivaLinux/devel/cooker/i586/isolinux/alt0>
17893238 bytes transferred in 77 seconds (227.9K/s)
Total 2 files transferred
lftp ftp.proxad.net:/pub/Distributions_Linux/MandrivaLinux/devel/cooker/i586/isolinux/alt0>

[root@node42 images]# pwd
/var/lib/tftpboot/X86PC/linux/images
[root@node42 images]# ls
all.rdz  vmlinuz
```

Now it's time to launch the duplication process

```
[root@node42 ka]# ka-d.sh -n 2 -p sda /tmp/desc -r grub
takembr =
desc = sda /tmp/desc
'/etc/fstab' -> '/tmp/ka-d3156/pfstab.tmp'
+ Mount points :
     /dev/sda2 / ext3 relatime 1 1
     /dev/sda5 /home ext4 relatime 1 2
     /dev/sda1 swap swap defaults 0 0
+ Hard drives :
     sda
+ Reading partition table description for sda
    Added partition 1 : type 82
    Added partition 2 : type 83
```

```
    Added partition 5 : type 83
+ Included mount points : / /home
+ Bootloader is: grub
+++ Sending Stage2 +++
Compiled : May  4 2010 20:33:07
ARGS=+ka-d-server+-s+getstage2+-n+2+-e+(cd /mnt/ka; tar --create --one-file-system --sparse
Server IP = 10.0.1.42
command = (cd /mnt/ka; tar --create --one-file-system --sparse  . )
I want 2 clients
Socket 4 on port 30765 on node42.guibland.com ready.
Socket 5 on port 30764 on node42.guibland.com ready.
got UDP packet from 10.0.1.111
Session name matches
Sending UDP reply to 10.0.1.111
Accepting connection from 10.0.1.111
Clients : want_data 0  / connected 0
client says hello !
Client sends options
Client accepts data
got UDP packet from 10.0.1.110
Session name matches
Sending UDP reply to 10.0.1.110
Accepting connection from 10.0.1.110
Clients : want_data 1  / connected 0
client says hello !
Client sends options
Client accepts data
Added client 10.0.1.110, daddy = 10.0.1.42
Added client 10.0.1.111, daddy = 10.0.1.110
Accepting connection from 10.0.1.110
checking connection auth10.0.1.42 reports 10.0.1.110 has opened data connection
Client 10.0.1.110 reports data position : 0
10.0.1.42 reports 10.0.1.110 has been accepted
Welcome son, you are number 1 (MAX 4)
Client got client
10.0.1.110 reports 10.0.1.111 has opened data connection
Client 10.0.1.111 reports data position : 0
sending auth for 10.0.1.111 to 10.0.1.110
Client got client
10.0.1.110 reports 10.0.1.111 has been accepted
Let's go!
Total data read = 43 Megs, BUF: 34M  FREE = 0M  startpos = 8MM
End of data flow
Dropping children
Dropping child 10.0.1.110
All children dropped
Client says dad disconnected
Client says he has finished
Client has finished transfer
Busy clients: 1 -- connected : 2
Client says dad disconnected
Client says he has finished
Client has finished transfer
```

```
Busy clients: 0 -- connected : 2
Peer closed connection on socket 7
close_connection(7)
Busy clients: 0 -- connected : 1
Peer closed connection on socket 6
close_connection(6)
Busy clients: 0 -- connected : 0
All clients left, I quit
Total data sent = 44 Megs, in 1543 packets
Transfer time = 0.858 seconds, throughput = 51.307 Mbytes/second
The pipeline was emptied in 3.250 seconds
 - Sending partition/filesystem/mount points informations...
 +++ Running ka-deploy +++
Compiled : May  4 2010 20:33:07
ARGS=+ka-d-server+-s+kainstall1+-n+2+-e+(cd /tmp/ka-d3156 && tar c *)+
Server IP = 10.0.1.42
command = (cd /tmp/ka-d3156 && tar c *)
I want 2 clients
Socket 4 on port 30765 on node42.guibland.com ready.
Socket 5 on port 30764 on node42.guibland.com ready.
got UDP packet from 10.0.1.110
Session name matches
Sending UDP reply to 10.0.1.110
Accepting connection from 10.0.1.110
Clients : want_data 0  / connected 0
client says hello !
Client sends options
Client accepts data
got UDP packet from 10.0.1.111
Session name matches
Sending UDP reply to 10.0.1.111
Accepting connection from 10.0.1.111
Clients : want_data 1  / connected 0
client says hello !
Client sends options
Client accepts data
Added client 10.0.1.110, daddy = 10.0.1.42
Added client 10.0.1.111, daddy = 10.0.1.110
Accepting connection from 10.0.1.110
checking connection auth10.0.1.42 reports 10.0.1.110 has opened data connection
Client 10.0.1.110 reports data position : 0
10.0.1.42 reports 10.0.1.110 has been accepted
Welcome son, you are number 1 (MAX 4)
Client got client
10.0.1.110 reports 10.0.1.111 has opened data connection
Client 10.0.1.111 reports data position : 0
sending auth for 10.0.1.111 to 10.0.1.110
Client got client
10.0.1.110 reports 10.0.1.111 has been accepted
Let's go!
Total data read = 0 Megs, BUF: 0M  FREE = 34M  startpos = 0M
End of data flow
Dropping children
```

```
Dropping child 10.0.1.110
All children dropped
Client says dad disconnected
Client says he has finished
Client has finished transfer
Busy clients: 1 -- connected : 2
Client says dad disconnected
Client says he has finished
Client has finished transfer
Busy clients: 0 -- connected : 2
Peer closed connection on socket 6
close_connection(6)
Busy clients: 0 -- connected : 1
Peer closed connection on socket 7
close_connection(7)
Busy clients: 0 -- connected : 0
All clients left, I quit
Total data sent = 0 Megs, in 1 packets
Transfer time = 0.006 seconds, throughput = 1.698 Mbytes/second
The pipeline was emptied in 0.038 seconds
 WAITING node (partition/format)
 - Sending Linux filesystem...
 +++ Running ka-deploy +++
Compiled : May  4 2010 20:33:07
ARGS=+ka-d-server+-s+kainstall2+-n+2+-e+(cd /; tar --create --one-file-system --sparse  / /
Server IP = 10.0.1.42
command = (cd /; tar --create --one-file-system --sparse  / /home)
I want 2 clients
Socket 4 on port 30765 on node42.guibland.com ready.
Socket 5 on port 30764 on node42.guibland.com ready.
got UDP packet from 10.0.1.110
Session name matches
Sending UDP reply to 10.0.1.110
Accepting connection from 10.0.1.110
Clients : want_data 0  / connected 0
client says hello !
Client sends options
Client accepts data
got UDP packet from 10.0.1.111
Session name matches
Sending UDP reply to 10.0.1.111
Accepting connection from 10.0.1.111
Clients : want_data 1  / connected 0
client says hello !
Client sends options
Client accepts data
Added client 10.0.1.110, daddy = 10.0.1.42
Added client 10.0.1.111, daddy = 10.0.1.110
Accepting connection from 10.0.1.110
checking connection auth10.0.1.42 reports 10.0.1.110 has opened data connection
Client got client
10.0.1.110 reports 10.0.1.111 has opened data connection
Client 10.0.1.110 reports data position : 0
```

```
10.0.1.42 reports 10.0.1.110 has been accepted
Welcome son, you are number 1 (MAX 4)
Client 10.0.1.111 reports data position : 0
sending auth for 10.0.1.111 to 10.0.1.110
Client got client
10.0.1.110 reports 10.0.1.111 has been accepted
Let's go!
Total data read = 789 Megs, BUF: 34M  FREE = 0M  startpos = 754M
End of data flow
Dropping children
Dropping child 10.0.1.110
All children dropped
Client says dad disconnected
Client says he has finished
Client has finished transfer
Busy clients: 1 -- connected : 2
Client says dad disconnected
Client says he has finished
Client has finished transfer
Busy clients: 0 -- connected : 2
Peer closed connection on socket 7
close_connection(7)
Busy clients: 0 -- connected : 1
Peer closed connection on socket 6
close_connection(6)
Busy clients: 0 -- connected : 0
All clients left, I quit
Total data sent = 792 Megs, in 25445 packets
Transfer time = 69.904 seconds, throughput = 11.343 Mbytes/second
The pipeline was emptied in 4.002 seconds
```

KA client side

To get the log of the client node, launch **/mnt/ka/ka/status_node.pl IPADD** on the golden node.

```
[root@node42 ka]# status_node.pl 10.0.1.111

 Get the status of the Ka duplication process
 If you want to execute a command on node, just use the 'exec' prefix
10.0.1.111> ------| Ka |---- Install starting...
10.0.1.111> Current session is -s kainstall1
10.0.1.111>  Receiving partitions information...OK
10.0.1.111>  Cleaning hard drive...
10.0.1.111> ==> /tmp/kacmd <==
10.0.1.111> Starting log server..
10.0.1.111>
10.0.1.111> ==> /tmp/ka_log-10.0.1.111-20100507-10h32 <==
10.0.1.111> OK
10.0.1.111> Device contains neither a valid DOS partition table, nor Sun, SGI or OSF diskla
10.0.1.111> Building a new DOS disklabel with disk identifier 0x59be1427.
```

```
10.0.1.111> Changes will remain in memory only, until you decide to write them.
10.0.1.111> After that, of course, the previous content won't be recoverable.
10.0.1.111>
10.0.1.111>
10.0.1.111> The number of cylinders for this disk is set to 1116.
10.0.1.111> There is nothing wrong with that, but this is larger than 1024,
10.0.1.111> and could in certain setups cause problems with:
10.0.1.111> 1) software that runs at boot time (e.g., old versions of LILO)
10.0.1.111> 2) booting and partitioning software from other OSs
10.0.1.111>    (e.g., DOS FDISK, OS/2 FDISK)
10.0.1.111> Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
10.0.1.111>
10.0.1.111> Command (m for help): The partition table has been altered!
10.0.1.111>
10.0.1.111> Calling ioctl() to re-read partition table.
10.0.1.111> Syncing disks.
10.0.1.111>  Writing partition table for sda using fdisk...OK
10.0.1.111>  Formatting /dev/sda2 as ext3...OK
10.0.1.111>  Formatting /dev/sda5 as ext4...OK
10.0.1.111>  Formatting /dev/sda1 as swap...OK
10.0.1.111> - Mounting /dev/sda2 as /mnt/disk/ ......OK
10.0.1.111> - Mounting /dev/sda5 as /mnt/disk/home ......OK
10.0.1.111> ++++++++++++++++++++++++++
10.0.1.111> none on /sys type sysfs (rw,relatime)
10.0.1.111> none on /proc/bus/usb type usbfs (rw,relatime)
10.0.1.111> /dev/ram3 on /tmp/stage2 type ext2 (rw,relatime,errors=continue)
10.0.1.111> /dev/sda2 on /mnt/disk type ext3 (rw,relatime,errors=continue,data=ordered)
10.0.1.111> /dev/sda5 on /mnt/disk/home type ext4 (rw,relatime,barrier=1,data=ordered)
10.0.1.111> ++++++++++++++++++++++++++
10.0.1.111> Linux copy is about to start
10.0.1.111> Server IP is 10.0.1.42
10.0.1.111> Buffers names :pipe
Total data received = 21 Megs (10.830 Mbytes/sec); BUF :0M
10.0.1.111> ==> /tmp/kacmd <==
10.0.1.111> Starting log server..
10.0.1.111>
10.0.1.111> ==> /tmp/ka_log-localhost-20100507-11h17 <==
Total data received = 788 Megs (10.796 Mbytes/sec); BUF :0M End of data flow
10.0.1.111> Flushing buffers
10.0.1.111> Total data received = 792 Megs, in 574479 packets
10.0.1.111> Elapsed time = 73.413 seconds, throughput = 10.801 Mbytes/second
10.0.1.111>  Syncing disks...OK
10.0.1.111> Linux copy done.
10.0.1.111> Creating excluded directories
10.0.1.111> - bootloader is grub (user choice)
10.0.1.111> '/tmp/partfiles/pfstab' -> '/mnt/disk/etc/fstab'
10.0.1.111> - Removing duplicated dhcp cache
10.0.1.111> - Writing modprobe.conf
10.0.1.111> ********************
10.0.1.111> install scsi_hostadapter /sbin/modprobe aic7xxx; /bin/true
10.0.1.111> install scsi_hostadapter /sbin/modprobe pata_serverworks; /bin/true
10.0.1.111> alias eth0 tg3
10.0.1.111> alias eth1 tg3
```

```
10.0.1.111> ********************
10.0.1.111> - Remove persistent udev rules
10.0.1.111> removed `/mnt/disk/etc/udev/rules.d/70-persistent-net.rules'
10.0.1.111> - Fix /dev in /mnt/disk
10.0.1.111> `/ka2/udev_creation.sh' -> `/mnt/disk/sbin/udev_creation.sh'
10.0.1.111> Starting udev:                                                 [  OK
10.0.1.111> umount: /mnt/disk/dev: device is busy.
10.0.1.111>         (In some cases useful info about processes that use
10.0.1.111>          the device is found by lsof(8) or fuser(1))
10.0.1.111> - Running mkinitrd
10.0.1.111> - Looking for default grub menu
10.0.1.111> - erase old initrd.img link
10.0.1.111> removed `/mnt/disk/boot/initrd.img'
10.0.1.111>  initrd will be : /boot/initrd-2.6.33.3-desktop-1mnb.img
10.0.1.111>  running: chroot /mnt/disk /sbin/mkinitrd -v -f  /boot/initrd-2.6.33.3-desktop-
10.0.1.111> Creating initramfs
10.0.1.111> Looking for driver for /dev/sda2 in /sys/block/sda/sda2
10.0.1.111> Looking for deps of module scsi:t-0x00: crc-t10dif scsi_mod sd_mod
10.0.1.111> Looking for deps of module pci:v00009005d000000CFsv00001028sd00000121bc01sc00i0
10.0.1.111> Looking for deps of module pci:v00008086d00000309sv00000000sd00000000bc06sc04i0
10.0.1.111> Looking for driver for /dev/sda1 in /sys/block/sda/sda1
10.0.1.111> Using modules:  usbhid ehci-hcd ohci-hcd uhci-hcd ext3 crc-t10dif scsi_mod sd_m
10.0.1.111> Building initrd in /tmp/initrd.uuIIkZ
10.0.1.111> /sbin/nash -> /tmp/initrd.uuIIkZ/bin/nash
10.0.1.111>   /usr/lib/libnash.so.6.0.93 -> /tmp/initrd.uuIIkZ/usr/lib/libnash.so.6.0.93
10.0.1.111>     /lib/libdevmapper.so.1.02 -> /tmp/initrd.uuIIkZ/lib/libdevmapper.so.1.02
10.0.1.111>       /lib/libreadline.so.6 -> /tmp/initrd.uuIIkZ/lib/libreadline.so.6
10.0.1.111>         /lib/libreadline.so.6.1 -> /tmp/initrd.uuIIkZ/lib/libreadline.so.6.1
10.0.1.111>           /lib/libncurses.so.5 -> /tmp/initrd.uuIIkZ/lib/libncurses.so.5
10.0.1.111>             /lib/libncurses.so.5.7 -> /tmp/initrd.uuIIkZ/lib/libncurses.so.5.7
10.0.1.111>               /lib/libc.so.6 -> /tmp/initrd.uuIIkZ/lib/libc.so.6
10.0.1.111>                 /lib/libc-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/libc-2.11.1.so
10.0.1.111>                   /lib/ld-linux.so.2 -> /tmp/initrd.uuIIkZ/lib/ld-linux.so.2
10.0.1.111>                     /lib/ld-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/ld-2.11.1.so
10.0.1.111>               /lib/libdl.so.2 -> /tmp/initrd.uuIIkZ/lib/libdl.so.2
10.0.1.111>                 /lib/libdl-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/libdl-2.11.1.so
10.0.1.111>       /lib/libudev.so.0 -> /tmp/initrd.uuIIkZ/lib/libudev.so.0
10.0.1.111>         /lib/libudev.so.0.7.0 -> /tmp/initrd.uuIIkZ/lib/libudev.so.0.7.0
10.0.1.111>     /usr/lib/libparted.so.0 -> /tmp/initrd.uuIIkZ/usr/lib/libparted.so.0
10.0.1.111>       /usr/lib/libparted.so.0.0.1 -> /tmp/initrd.uuIIkZ/usr/lib/libparted.so.0.
10.0.1.111>         /lib/libuuid.so.1 -> /tmp/initrd.uuIIkZ/lib/libuuid.so.1
10.0.1.111>           /lib/libuuid.so.1.3.0 -> /tmp/initrd.uuIIkZ/lib/libuuid.so.1.3.0
10.0.1.111>     /lib/libblkid.so.1 -> /tmp/initrd.uuIIkZ/lib/libblkid.so.1
10.0.1.111>       /lib/libblkid.so.1.1.0 -> /tmp/initrd.uuIIkZ/lib/libblkid.so.1.1.0
10.0.1.111>     /lib/libpopt.so.0 -> /tmp/initrd.uuIIkZ/lib/libpopt.so.0
10.0.1.111>       /lib/libpopt.so.0.0.0 -> /tmp/initrd.uuIIkZ/lib/libpopt.so.0.0.0
10.0.1.111>     /lib/libresolv.so.2 -> /tmp/initrd.uuIIkZ/lib/libresolv.so.2
10.0.1.111>       /lib/libresolv-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/libresolv-2.11.1.so
10.0.1.111>     /usr/lib/libelf.so.1 -> /tmp/initrd.uuIIkZ/usr/lib/libelf.so.1
10.0.1.111>       /usr/lib/libelf-0.146.so -> /tmp/initrd.uuIIkZ/usr/lib/libelf-0.146.so
10.0.1.111>     /lib/libm.so.6 -> /tmp/initrd.uuIIkZ/lib/libm.so.6
10.0.1.111>       /lib/libm-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/libm-2.11.1.so
10.0.1.111>     /lib/libgcc_s.so.1 -> /tmp/initrd.uuIIkZ/lib/libgcc_s.so.1
```

```
10.0.1.111>       /lib/libgcc_s-4.4.3.so.1 -> /tmp/initrd.uuIIkZ/lib/libgcc_s-4.4.3.so.1
10.0.1.111>   /usr/lib/libbdevid.so.6.0.93 -> /tmp/initrd.uuIIkZ/usr/lib/libbdevid.so.6.0.9
10.0.1.111> /sbin/modprobe -> /tmp/initrd.uuIIkZ/bin/modprobe
10.0.1.111>   /lib/libmodprobe.so.0 -> /tmp/initrd.uuIIkZ/lib/libmodprobe.so.0
10.0.1.111>     /lib/libmodprobe.so.0.0.0 -> /tmp/initrd.uuIIkZ/lib/libmodprobe.so.0.0.0
10.0.1.111>       /lib/libz.so.1 -> /tmp/initrd.uuIIkZ/lib/libz.so.1
10.0.1.111>         /lib/libz.so.1.2.3 -> /tmp/initrd.uuIIkZ/lib/libz.so.1.2.3
10.0.1.111> /sbin/rmmod -> /tmp/initrd.uuIIkZ/bin/rmmod
10.0.1.111> /bin/ln -> /tmp/initrd.uuIIkZ/bin/ln
10.0.1.111> resolving for MODULES
10.0.1.111> and that has items of usbhid ehci-hcd ohci-hcd uhci-hcd ext3 crc-t10dif scsi_mo
10.0.1.111> Looking for deps of module usbhid: usbcore hid
10.0.1.111> Looking for deps of module ehci-hcd: usbcore
10.0.1.111> Looking for deps of module ohci-hcd: usbcore
10.0.1.111> Looking for deps of module uhci-hcd: usbcore
10.0.1.111> Looking for deps of module ext3: jbd
10.0.1.111> Looking for deps of module crc-t10dif
10.0.1.111> Looking for deps of module scsi_mod
10.0.1.111> Looking for deps of module sd_mod: crc-t10dif scsi_mod
10.0.1.111> Looking for deps of module scsi_transport_spi: scsi_mod
10.0.1.111> Looking for deps of module aic7xxx: scsi_mod scsi_transport_spi
10.0.1.111> Looking for deps of module pci_hotplug
10.0.1.111> Looking for deps of module shpchp: pci_hotplug
10.0.1.111> Looking for deps of module pata_serverworks: scsi_mod libata
10.0.1.111> Looking for deps of module aic7xxx: scsi_mod scsi_transport_spi
10.0.1.111> Looking for deps of module crc-t10dif
10.0.1.111> Looking for deps of module ehci-hcd: usbcore
10.0.1.111> Looking for deps of module ext3: jbd
10.0.1.111> Looking for deps of module hid
10.0.1.111> Looking for deps of module jbd
10.0.1.111> Looking for deps of module libata: scsi_mod
10.0.1.111> Looking for deps of module ohci-hcd: usbcore
10.0.1.111> Looking for deps of module pata_serverworks: scsi_mod libata
10.0.1.111> Looking for deps of module pci_hotplug
10.0.1.111> Looking for deps of module scsi_mod
10.0.1.111> Looking for deps of module scsi_transport_spi: scsi_mod
10.0.1.111> Looking for deps of module sd_mod: crc-t10dif scsi_mod
10.0.1.111> Looking for deps of module shpchp: pci_hotplug
10.0.1.111> Looking for deps of module uhci-hcd: usbcore
10.0.1.111> Looking for deps of module usbcore
10.0.1.111> Looking for deps of module usbhid: usbcore hid
10.0.1.111> resolving for availmodules
10.0.1.111> and that has items of
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/scsi/aic7xxx/aic7xxx.ko.gz ->
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/lib/crc-t10dif.ko.gz -> /tmp/initrd.u
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/usb/host/ehci-hcd.ko.gz -> /t
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/fs/ext3/ext3.ko.gz -> /tmp/initrd.uuI
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/hid/hid.ko.gz -> /tmp/initrd.
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/fs/jbd/jbd.ko.gz -> /tmp/initrd.uuIIk
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/ata/libata.ko.gz -> /tmp/init
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/usb/host/ohci-hcd.ko.gz -> /t
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/ata/pata_serverworks.ko.gz ->
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/pci/hotplug/pci_hotplug.ko.gz
```

```
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/scsi/scsi_mod.ko.gz -> /tmp/i
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/scsi/scsi_transport_spi.ko.gz
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/scsi/sd_mod.ko.gz -> /tmp/ini
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/pci/hotplug/shpchp.ko.gz -> /
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/usb/host/uhci-hcd.ko.gz -> /t
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/usb/core/usbcore.ko.gz -> /tm
10.0.1.111> /lib/modules/2.6.33.3-desktop-1mnb/kernel/drivers/hid/usbhid/usbhid.ko.gz -> /t
10.0.1.111> /etc/sysconfig/keyboard -> /tmp/initrd.uuIIkZ/etc/sysconfig/keyboard
10.0.1.111> /bin/loadkeys -> /tmp/initrd.uuIIkZ/bin/loadkeys
10.0.1.111> /etc/sysconfig/console/default.kmap -> /tmp/initrd.uuIIkZ/etc/sysconfig/console
10.0.1.111> /etc/sysconfig/i18n -> /tmp/initrd.uuIIkZ/etc/sysconfig/i18n
10.0.1.111> /bin/setfont -> /tmp/initrd.uuIIkZ/bin/setfont
10.0.1.111> /usr/lib/kbd/consolefonts/lat0-16.psfu.gz -> /tmp/initrd.uuIIkZ/usr/lib/kbd/con
10.0.1.111> /lib/udev/console_init -> /tmp/initrd.uuIIkZ/lib/udev/console_init
10.0.1.111> probing for drm modules for pci device /sys/bus/pci/devices/0000:00:0e.0
10.0.1.111> Adding graphics device /sys/bus/pci/devices/0000:00:0e.0
10.0.1.111> Looking for deps of module pci:v00001002d00004752sv00001028sd00000121bc03sc00i0
10.0.1.111> resolving for GRAPHICSMODS
10.0.1.111> and that has items of
10.0.1.111> Adding module usbhid
10.0.1.111> Adding module ehci-hcd
10.0.1.111> Adding module ohci-hcd
10.0.1.111> Adding module uhci-hcd
10.0.1.111> Adding module ext3
10.0.1.111> Adding module crc-t10dif
10.0.1.111> Adding module scsi_mod
10.0.1.111> Adding module sd_mod
10.0.1.111> Adding module scsi_transport_spi
10.0.1.111> Adding module aic7xxx
10.0.1.111> Adding module pci_hotplug
10.0.1.111> Adding module shpchp
10.0.1.111> Adding module pata_serverworks
10.0.1.111> /usr/sbin/resume -> /tmp/initrd.uuIIkZ/bin/resume
10.0.1.111>   /usr/lib/suspend/resume -> /tmp/initrd.uuIIkZ/usr/lib/suspend/resume
10.0.1.111>     /usr/lib/liblzo2.so.2 -> /tmp/initrd.uuIIkZ/usr/lib/liblzo2.so.2
10.0.1.111>       /usr/lib/liblzo2.so.2.0.0 -> /tmp/initrd.uuIIkZ/usr/lib/liblzo2.so.2.0.0
10.0.1.111>     /lib/libpthread.so.0 -> /tmp/initrd.uuIIkZ/lib/libpthread.so.0
10.0.1.111>       /lib/libpthread-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/libpthread-2.11.1.so
10.0.1.111>     /lib/libply.so.2 -> /tmp/initrd.uuIIkZ/lib/libply.so.2
10.0.1.111>       /lib/libply.so.2.0.0 -> /tmp/initrd.uuIIkZ/lib/libply.so.2.0.0
10.0.1.111>         /lib/librt.so.1 -> /tmp/initrd.uuIIkZ/lib/librt.so.1
10.0.1.111>           /lib/librt-2.11.1.so -> /tmp/initrd.uuIIkZ/lib/librt-2.11.1.so
10.0.1.111>     /lib/libply-splash-core.so.2 -> /tmp/initrd.uuIIkZ/lib/libply-splash-core.s
10.0.1.111>       /lib/libply-splash-core.so.2.0.0 -> /tmp/initrd.uuIIkZ/lib/libply-splash-
10.0.1.111> /etc/suspend.conf -> /tmp/initrd.uuIIkZ/etc/suspend.conf
10.0.1.111> This initrd uses dynamic shared objects.
10.0.1.111> Adding dynamic linker configuration files.
10.0.1.111> /etc/ld.so.conf -> /tmp/initrd.uuIIkZ/etc/ld.so.conf
10.0.1.111> Running ldconfig
10.0.1.111> Installation finished. No error reported.
10.0.1.111> This is the contents of the device map /boot/grub/device.map.
10.0.1.111> Check if this is correct or not. If any of the lines is incorrect,
10.0.1.111> fix it and re-run the script ‘grub-install’.
```

```
10.0.1.111>
10.0.1.111> (hd0) /dev/sda
10.0.1.111>  Umounting /dev/sda5...OK
10.0.1.111>  Umounting /dev/sda2...OK
10.0.1.111> ftp: connect: Connection refused
10.0.1.111> Local directory now /tmp
10.0.1.111> Not connected.
```

Post duplication process

Now client nodes reboots, so we have to switch PXE to a local boot.

```
[root@node42 ka]# vi /var/lib/tftpboot/X86PC/linux/pxelinux.cfg/default
DEFAULT local
```

Nodes hsould be up, we can to see their hardware.

```
[root@node42 ka]# ssh render@10.0.1.110
[render@linux ~]$ lspcidrake
8139too         : Realtek Semiconductor Co., Ltd.|RTL-8139/8139C/8139C+ [NETWORK_ETHERNET]
snd_intel8x0    : Intel Corporation|82801EB/ER (ICH5/ICH5R) AC'97 Audio Controller [MULTIME
i2c_i801        : Intel Corporation|82801EB/ER (ICH5/ICH5R) SMBus Controller [SERIAL_SMBUS]
ata_piix        : Intel Corporation|82801EB/ER (ICH5/ICH5R) IDE Controller [STORAGE_IDE] (r
iTCO_wdt        : Intel Corporation|82801EB/ER (ICH5/ICH5R) LPC Interface Bridge [BRIDGE_IS
shpchp          : Intel Corporation|82801 PCI Bridge [BRIDGE_PCI] (rev: c2)
ehci_hcd        : Intel Corporation|82801EB/ER (ICH5/ICH5R) USB2 EHCI Controller [SERIAL_US
uhci_hcd        : Intel Corporation|82801EB/ER (ICH5/ICH5R) USB UHCI Controller #4 [SERIAL_
uhci_hcd        : Intel Corporation|82801EB/ER (ICH5/ICH5R) USB UHCI Controller #3 [SERIAL_
uhci_hcd        : Intel Corporation|82801EB/ER (ICH5/ICH5R) USB UHCI Controller #2 [SERIAL_
uhci_hcd        : Intel Corporation|82801EB/ER (ICH5/ICH5R) USB UHCI Controller #1 [SERIAL_
unknown         : Intel Corporation|82865G/PE/P Processor to I/O Memory Interface [SYSTEM_O
Card:Intel 810 and later: Intel Corporation|82865G Integrated Graphics Controller [DISPLAY_
unknown         : Intel Corporation|82865G/PE/P DRAM Controller/Host-Hub Interface [BRIDGE_
hub             : Linux 2.6.33.3-desktop-1mnb uhci_hcd|UHCI Host Controller [Hub|Unused|Ful
hub             : Linux 2.6.33.3-desktop-1mnb uhci_hcd|UHCI Host Controller [Hub|Unused|Ful
hub             : Linux 2.6.33.3-desktop-1mnb uhci_hcd|UHCI Host Controller [Hub|Unused|Ful
usbhid          :  |SCISSORS Keyboard [Human Interface Device|Boot Interface Subclass|Keybo
hub             : Linux 2.6.33.3-desktop-1mnb uhci_hcd|UHCI Host Controller [Hub|Unused|Ful
hub             : Linux 2.6.33.3-desktop-1mnb ehci_hcd|EHCI Host Controller [Hub|Unused|Ful
[render@linux ~]$ cat /etc/modprobe.conf
install scsi_hostadapter /sbin/modprobe ata_piix; /sbin/modprobe ahci; /bin/true
alias eth0 8139too
install usb-interface /sbin/modprobe ehci_hcd; /sbin/modprobe uhci_hcd; /bin/true
alias sound-slot-0 snd_intel8x0

[root@node42 ka]# ssh render@10.0.1.111
[render@localhost ~]$ lspcidrake
ath5k           : Atheros Communications Inc.|AR2413 802.11bg NIC [NETWORK_ETHERNET] (rev:
tg3             : Broadcom Corporation|NetXtreme BCM5701 Gigabit Ethernet [NETWORK_ETHERNET
```

```
tg3             : Broadcom Corporation|NetXtreme BCM5701 Gigabit Ethernet [NETWORK_ETHERNET
aic7xxx         : Adaptec|AIC-7899P U160/m [STORAGE_SCSI] (rev: 01)
aic7xxx         : Adaptec|AIC-7899P U160/m [STORAGE_SCSI] (rev: 01)
shpchp          : Intel Corporation|80303 I/O Processor PCI-to-PCI Bridge [BRIDGE_PCI] (rev
unknown         : Broadcom|CIOB-X2 PCI-X I/O Bridge [BRIDGE_HOST] (rev: 03)
unknown         : Broadcom|CIOB-X2 PCI-X I/O Bridge [BRIDGE_HOST] (rev: 03)
unknown         : Broadcom|CIOB-X2 PCI-X I/O Bridge [BRIDGE_HOST] (rev: 03)
unknown         : Broadcom|CIOB-X2 PCI-X I/O Bridge [BRIDGE_HOST] (rev: 03)
unknown         : Broadcom|CSB5 LPC bridge [BRIDGE_ISA]
ohci_hcd        : Broadcom|OSB4/CSB5 OHCI USB Controller [SERIAL_USB] (rev: 05)
pata_serverworks: Broadcom|CSB5 IDE Controller [STORAGE_IDE] (rev: 93)
i2c_piix4       : Broadcom|CSB5 South Bridge [BRIDGE_HOST] (rev: 93)
Card:ATI Mach 64-based cards (no 3D acceleration): ATI Technologies Inc|Rage XL [DISPLAY_VG
unknown         : Dell|Embedded Remote Access: BMC/SMIC device
unknown         : Dell|Remote Access Card III
unknown         : Dell|Embedded Remote Access or ERA/O
unknown         : Broadcom|CMIC-LE [BRIDGE_HOST]
unknown         : Broadcom|CMIC-WS Host Bridge (GC-LE chipset) [BRIDGE_HOST]
unknown         : Broadcom|CMIC-WS Host Bridge (GC-LE chipset) [BRIDGE_HOST] (rev: 13)
hub             : Linux 2.6.33.3-desktop-1mnb ohci_hcd|OHCI Host Controller [Hub|Unused|Ful
```

```
[render@localhost ~]$ cat /etc/modprobe.conf
install scsi_hostadapter /sbin/modprobe pata_serverworks; /sbin/modprobe aic7xxx; /bin/true
alias eth0 tg3
alias eth1 tg3
install ide-controller /sbin/modprobe ide_generic; /bin/true
install usb-interface /sbin/modprobe ohci_hcd; /bin/true
alias wlan0 ath5k
```