



Sun, Nov 18 2001

<http://Linux.com/enhance/newsitem.phtml?sid=1&aid=12545>

## SF-Foundries@Linux.com:: With Linux Clusters - Image is Everything!

by Rick Philbrick  
Thursday October 11th, 2001

**M**OST SYSTEM ADMINISTRATORS KEEP A SHARP LOOKOUT FOR TOOLS THAT SAVE TIME or make life easier. Tasks that have recurring steps are great candidates for time saving tools, for example, the case of building or maintaining a farm of Web servers for an Internet site or a high performance cluster. If you are a system administrator with these responsibilities, SystemImager is a tool that you should take a look at, as we will in this article.

Written by Brian Elliott Finley, and a host of other contributors, SystemImager is an open source program that will automate the installation of server machines used in a cluster. It can also be used to update the *content* of servers or deploy software upgrades to systems en masse. The idea is to create a central repository of images on an ImageServer. Cluster machines can then be installed with or synchronize themselves to an image on the ImageServer.

The ImageServer stores the image created from one of the client machines, called the GoldenClient. The other machines can then be replicated, or autoinstalled, by booting them with special boot media. The autoinstall process partitions and formats the client's hard drive, pulls the appropriate image down, and even configures unique IP address and host name information. SystemImager works with IDE and SCSI disks, and even properly replicates software RAID and hardware RAID configurations.

Think of it this way, you supply the ImageServer with a perfect template, the GoldenClient, and it will reproduce as many identical machines as you wish. No more having to tweak RPMs or having to fuss with "tarballs" on every machine in the cluster. Just get your GoldenClient working properly, pull its image to the ImageServer, and autoinstall or synchronize the rest of the clients to match that image. The `updateclient` command is used to make the synchronization bit easy.

As a bonus you don't even have to be onsite to build or update the cluster machines. Let's say you have a machine at your facility on the East coast that needs to be re-installed, but you're on the West Coast. With SystemImager, you can remotely re-install that machine from the other side of the country. You can even do this securely because SystemImager supports operation over SSH. Pretty cool stuff.

**What do you need to get started?**

First you'll need a machine designated as the ImageServer. Make sure it has enough disk space to store the images you want pulled from your GoldenClient(s). Your ImageServer needs to have a good network connection and I've found that standard 10/100 Ethernet works fine for most sites.

Download and install the SystemImager server software on your ImageServer. You can find the software at <http://systemimager.org/download/>. You may need to install one or two additional packages if they are not already on your machine, but these details are explained on the download page at the SystemImager site.

The current release, 1.5.0, is tested and known to work well on Debian and RedHat based systems. I've also had success using GentooLinux as a SystemImager client. GentooLinux is very rich in features and has built-in support for the reiserfs journaling file system. The current version of SystemImager supports the ext2, ext3, and reiserfs file systems. Future versions may include support for other filesystems, such as XFS, and the next release, expected sometime this fall, will include support for any and all Linux distributions through the use of [System Configurator](#), a tool written by Sean Dague of IBM.

## Unique IP Addresses and Hostnames

Another nice feature of SystemImager is that the ImageServer can assign unique IP addresses and hostnames to the autoinstall clients by using DHCP or a configuration file placed on the clients autoinstall media. I prefer using DHCP, so I ran `makedhcpserver` on my ImageServer, which walked me through configuring my DHCP server software. The potentially tricky process of creating a `dhcpd.conf` file is made easy by SystemImager's `makedhcpserver` command.

Here's how it goes:

```
[root@imageserver]# makedhcpserver

Welcome to the SystemImager "makedhcpserver" command. This command
will prepare this computer to be a DHCP server by creating the
following file:

/etc/dhcpd.conf

If there is an existing file, it will be backed up as:

/etc/dhcpd.conf.beforessystemimager

Continue? (y/[n]): y

Type your response or hit <Enter> to accept [defaults]. If you don't
have a response, such as no first or second DNS server, just hit
<Enter> and none will be used.

What is your domain name? [localdomain.domain]: mydomain.com
What is your network number? [192.168.1.0]:
What is your netmask? [255.255.255.0]:
What is the starting IP address for your dhcp range? [192.168.1.1]:
What is the ending IP address for your dhcp range? [192.168.1.100]: 192.168.1.99
What is the IP address of your first DNS server? []: 192.168.1.200
What is the IP address of your second DNS server? []: 192.168.1.201
What is the IP address of your third DNS server? []:
What is the IP address of your default gateway? [192.168.1.254]:
What is the IP address of imageserver? [192.168.1.254]: 192.168.1.203

Will your clients be installed over SSH? (y/[n]):

What is the air speed velocity of the common swallow? []:

Ahh, but seriously folks...
Here are the values you have chosen:

#####
DNS Domain Name: mydomain.com
network number: 192.168.1.0
netmask: 255.255.255.0
starting IP address for your dhcp range: 192.168.1.1
ending IP address for your dhcp range: 192.168.1.99
first DNS server: 192.168.1.200
second DNS server: 192.168.1.201
third DNS server:
default gateway: 192.168.1.254
imageserver: 192.168.1.203
ssh URL:
#####
```

```

Are you satisfied? (y/[n]): y

The dhcp server configuration file (/etc/dhcpd.conf) file has been
created for you. Please verify it for accuracy.

If this file does not look satisfactory, you can run this command again
to re-create it: "makedhcpserver"

WARNING!: If you have multiple physical network interfaces, be sure to
edit the init script that starts dhcpd to specify the interface that
is connected to your DHCP clients. Here's an example:

Change "/usr/sbin/dhcpd" to "/usr/sbin/dhcpd eth1".

Also, be sure to start or restart your dhcpd daemon. This can usually
be done with a command like "/etc/init.d/dhcpd restart" or similar.

```

Next, select a machine to be your GoldenClient. This machine's image is the one you'll be storing on the ImageServer and replicating to your other machines. Now let's download the SystemImager client software and install it on your GoldenClient. Once it's installed, simply run the `prepareclient` command. `Prepareclient` gets your GoldenClient ready to have its image pulled to the ImageServer.

## prepareclient

Here's what happens when you run `prepareclient`:

```

[root@imageserver]# prepareclient

Welcome to the SystemImager prepareclient command. This command
may modify the following files to prepare your client for having it's
image retrieved by the imageserver. It will also create the
/etc/systemimager directory and fill it with information about your
golden client, such as the disk partitioning scheme(s).

/etc/services -- add rsync line if necessary
/etc/inetd.conf -- comment out rsync line if necessary
                 (rsync will run as a daemon until shutdown)
/tmp/rsyncd.conf -- create a temporary rsyncd.conf file with a
                 [root] entry in it.

All modified files will be backed up with the .beforessystemimager
extension.

See "prepareclient -help" for command line options.

Prepare client for SystemImager? (y/[n]): y
Ok. Preparing client for SystemImager...

Creating /tmp/rsyncd.conf ...
Starting or re-starting rsync as a daemon....done!

This client is ready to have it's image retrieved.
You must now run the "getimage" command on the imageserver.

```

In order to create more machines in the likeness of the GoldenClient we'll need to get the image from the GoldenClient and store it on the ImageServer. We do this by running the `getimage` command on the ImageServer. Running the `getimage` command can be as simple as specifying the name or IP address of the GoldenClient and what you want to call the image. But there are many more options, too.

## getimage

Here is an example of me running the `getimage` command to pull an image from my GoldenClient to my ImageServer:

```

[root@imageserver]# getimage -golden my-golden-client -image web_server_image_v1

```

```

This program will get the "web_server_image_v1" system image from "my-golden-client"
making the assumption that all filesystems considered part
of the system image are using ext2, ext3, or reiserfs.

This program will not get /proc, NFS, or other filesystems
not mentioned above.

See "getimage -help" for command line options.

Continue? ([y]/n): y

Retrieving /etc/systemimager/mounted_filesystems from my-golden-client to check for mounted filesystems...
----- my-golden-client mounted_filesystems RETRIEVAL PROGRESS -----
receiving file list ... done
/var/spool/systemimager/images/web_server_image_v1/etc/systemimager/mounted_filesystems
wrote 132 bytes read 294 bytes 852.00 bytes/sec
total size is 180 speedup is 0.42
----- my-golden-client mounted_filesystems RETRIEVAL FINISHED -----

Retrieving image web_server_image_v1 from my-golden-client
----- web_server_image_v1 IMAGE RETRIEVAL PROGRESS -----
receiving file list ... done
./
bin/
bin/mt -> /etc/alternatives/mt
bin/pidof -> ../sbin/killall5
bin/rbash -> bash
bin/sh -> bash
boot/
boot/lost+found/
cdrom/
dev/
dev/MAKEDEV -> /sbin/MAKEDEV
dev/agpgart
dev/atim
dev/audio
dev/audio1
dev/audio2
dev/audio3
dev/audioc1
dev/audioc1
dev/aztcd0
dev/bpcd

[ ... etc, etc, etc ... ]

var/log/
var/log/exim/
var/log/ksymoos/
var/log/news/
var/run/
var/spool/
var/spool/cron/
var/spool/cron/atjobs/
var/state/
var/state/apt/
var/state/apt/lists/
var/state/logrotate/
var/tmp/
wrote 117490 bytes read 134577117 bytes 413808.32 bytes/sec
total size is 134072667 speedup is 1.00
----- web_server_image_v1 IMAGE RETRIEVAL FINISHED -----

Press <Enter> to continue...

IP Address Assignment
-----

There are four ways to assign IP addresses to the client systems on an
ongoing basis:

1) static_dhcp -- A DHCP server will assign the
same static address each time to clients
installed with this image. Also see the
"makedhcpstatic" command.

2) dynamic_dhcp -- A DHCP server will assign IP
addresses dynamically to clients installed
with this image. They may be assigned a
different address each time.

3) static -- The IP address the client uses
during autoinstall will be permanently
assigned to that client.

4) replicant -- Don't mess with the network
settings in this image. I'm using it as a
backup and quick restore mechanism for a
single machine.

Which method do you prefer? [1]: 3
You have chosen method 3 for assigning IP addresses.

Are you satisfied? ([y]/n): y
Would you like to run the "addclients" utility now? (y/[n]): n

```

## addclients

So at this point we have an ImageServer, a GoldenClient, and the image of the GoldenClient stored on the ImageServer. You can store many different images on your ImageServer, so our next step is to tell

the ImageServer about the clients we want to autoinstall and which image they are to receive. This is done by running the `addclients` command. You don't have to give it any options. It will ask you all the things it needs to know in order to configure the ImageServer for you.

```
[root@imageserver]# addclients

Welcome to the SystemImager "addclients" utility
-----

This utility has 3 sections.

"Section 1" will ask you for your hostname information.

"Section 2" will allow you to create softlinks from each
client hostname to your "master" script in the
"/tftpboot/systemimager/" directory.

Example: www297.sh -> web_server_image_v1.master

"Section 3" will ask you for IP address information that will
be combined with the hostname information provided in Section 1
to create entries in "/etc/hosts" for each of these same clients.
New entries will be appended to the end of "/etc/hosts". If you
specify new hostnames for existing IP addresses, those entries
will be re-written in place to reflect the new host names.

Continue? ([y]/n):

addclients -- Section 1 (hostname information)
-----

The next series of questions will be used to create a range
of hostnames. You will be asked for your domain name, the
base host name, a beginning number, and an ending number.

For example, if you answer:
domain name = systemimager.org
base host name = www
starting number = 7
ending number = 11

Then the result will be a series of hostnames that looks like this:
www7.systemimager.org
www8.systemimager.org
www9.systemimager.org
www10.systemimager.org
www11.systemimager.org

What is your domain name? []: mydomain.com
What is the base host name that you want me to use? []: server
What number should I begin with? []: 1
What number should I end with? []: 99

I will work with hostnames: server1 through server99
in the domain: mydomain.com

Are you satisfied? (y/[n]): y

addclients -- Section 2 (soft links to master script)
-----

Would you like me to create soft links to a "master" script so
that hosts:

server1 through server99

can be autoinstalled with that image? ([y]/n):

Here is a list of available images:

debian_ide_2.2
debian_hwraid_ext2
web_server_image_v1

Which image would you like these hosts to receive? [web_server_image_v1]:

Your soft links have been created.

Press <Enter> to continue...

addclients -- Section 3 (adding or modifying /etc/hosts entries)
-----

It is necessary to have an entry for each client in "/etc/hosts".

I will ask you for your clients' IP addresses one subnet at a time.

Would you like me to make these entries for you? ([y]/n):

addclients -- Section 3 (adding or modifying /etc/hosts entries -- continued...)
-----
subnet 1

The first host in subnet 1 will be: server1
```

```
What is the starting IP address for subnet 1? []: 192.168.1.1
What is the ending IP address? []: 192.168.1.99
I will work with IP addresses: 192.168.1.1 through 192.168.1.99

Are you satisfied? (y/[n]): y

These entries have been added to /etc/hosts.

Press <Enter> to continue...
```

Now we need to create the boot media for your autoinstall clients. You can choose between CD-ROM, diskette, network, or hard drive (to re-install an already running client). We'll go with the floppy option. Let's take a blank floppy and place it in the diskette drive on the ImageServer. Make sure it's unlocked so you can write to it. Now we're going to use a program called `makeautoinstalldiskette`, no joke, that is part of the SystemImager server software. Simply run the command on the ImageServer and it will produce a bootable floppy that can be used to perform an autoinstall. The autoinstall media is not machine specific, so it can be used on any of your client machines.

And now for the fun part! Boot one of your client machines from the floppy, and watch as the hard drives are partitioned and see the image fly over the network. In just minutes you'll have installed a new machine in the likeness of the GoldenClient, with its own unique IP address and hostname!

Unless you tell `getimage` otherwise, your machine will "beep incessantly" (the default) to let you know when it's finished installing. Just take out the diskette and reboot. Your newly minted server machine is ready to join the cluster.

**For more information on SystemImager, please see:**

<http://systemimager.org/>. There you can find the software, documentation, and email lists where you can ask questions of other users.

#### **Acknowledgments:**

Special thanks Brian Elliott Finley for writing SystemImager and to Greg Pratt who contributed the templates for the Listings.

*Rick Philbrick (rickphilbrick@hotmail.com) is the Sr. Network Operations Engineer for Yozons Inc. and lives in Bellevue Washington. He has been working on Linux since 1996, being introduced to it while on the IT staff at E\*Trade.*

[ [return to enhance](#) ]

