

Imaging Computers using free tools

Imaging computers can save a lot of time when deploying and reinstalling windows. There are many tools that can be used to image or clone computers. I am going to focus on using simple freely available ones.

There are several open source tools that can be used to image computers. Some rely on creating a bit for bit copy of the hard drive, compressing the file and storing it. Others keep track of the partition table and the amount of used space on the drive and just store the data in the used space. Some use a network, some do not require a network. The one I have been using lately is the clonezilla live cd. I use it with a portable USB hard drive to store images to along with parted magic to resize and create partitions. I have used G4U (ghost 4 unix), G4L (Ghost for linux), udpcast, and couple of other free imaging tools.

Personal experiences:

I started out using G4U and a sftp server on the network. This worked well since it did a bit for bit copy of the hard drive. I was able to deploy the images on many different types of hard drives ranging in manufacturers and size (smaller to bigger). The downside is the time it takes to create and deploy an image in this way. It would take a couple of hours to create an image and about 30 minutes to an hour to deploy it. Not bad if you don't have a lot of machines to do or a lot of different images to keep track of. I had about 2 or 3 main images I would use, one for circ machines, one for public machines, and one for staff machines. For multiple machines I found udp cast to be very fast for copying one machine's hard drive to many others. I have used this mainly on our laptops we use for classes and the public computers when I set new ones up.

More recently I have been using clonezilla with a usb hard drive for creating images and deploying images in combination with storing a machine specific image on a linux partition on each machine. At my library I often find myself setting up one machine at a time. In order to image many staff machines at one time I would have to set them all up in a room, network them together with a separate switch, image them with a basic image, then manually set the computer names, generate new SIDS, and join them to the domain. After all that is done then I have to unhook each one, label them as to where they will go, and swap them out with existing computers. Once the new machines are

deployed occasionally I will have to reinstall or re-image one to get rid of some of the goodness that many windows nasties will give a user. If I re-image one, I would use my basic image and afterwards go through the computer specific settings. This way is definitely faster than doing a fresh install the tried and true manual way. After using Clonezilla to create images and deploy them from a usb drive it proved to be much faster than sending a bit for bit image to or from a sftp server on the network.

My Current Method:

Desktop hard drives have a lot of wasted space on them in the library I work at. Computers are coming with 80 gig hard drives. A typical windows installation takes up 15 to 20 gigs which leaves 50 or 60 gigs of unused space. Staff users save all their files to a file server. How can some of this unused space be used on desktop hard drives? I decided to use some of this space to store computer specific images on a linux partition.

- Step 1:
On the machine I use to create my master image I use a linux live cd called parted magic. Parted magic is a light weight linux distro that has some useful tools. Among the tools is one called gparted. Gparted can be used to manipulate existing partitions and create new ones. I start out by shrinking down the partition windows is installed on to make room for a 20 gig or so linux partition. I create the ext3 linux partition and format it using gparted. After this is done I proceed in booting windows up, installing software, and updates to it along with any non computer specific settings that need to be tweaked. I also copy newsid and any other utilities I will need when I perform the computer specific settings later in a folder on the C: drive.
- Step 2:
I then use Clonezilla and create an image of the entire hard drive on my USB hard drive. This takes about 15 to 40 minutes depending on hard drive size and used space.
- Step3:
I unpack a new computer and set it up at the location it is going to be in. I plug in my USB hard drive and boot the clonezilla cd up to copy the hard drive image onto the new machine. This takes about 15 minutes or so.
- Step4:
I boot into windows run newsid and tell it to rename the PC. When its done, I install additional software if needed. I then join the PC to the domain and make sure everything is working correctly.
- Step5:
I boot up the clonezilla cd and image just the windows partition to the

linux partition on the computer's internal hard drive. This provides a secure way to store the image file. When windows is booted up it can not read or write to the linux partition. This ensures the image file will not get deleted or infected by any virus that runs in windows.

Its been my experience that I will have to reinstall windows on a computer one or more times before the hard drive in the machine will fail. Having a computer specific image stored locally on the computer will allow me to quickly reinstall everything and perform windows updates afterwards in a very timely manor. I estimate the time to re-image the computer to take about 15-20 minutes add to that the time it takes to pull down and install the updates it will need. A guesstimation for this would be about a 1/2 hour or so. Total time to get the computer back in operation would be about 45 minutes. I have no images taking up space on a file server waiting to be used. Re-imaging does not effect the local network at all since the images are on locally connected devices. I can take my external hard drive to our branch library and deploy the same images there. Worst case scenerio if my portable USB hard drive dies, I have several computers I can create a new image from and tweak it accordingly to reuse.

Clonezilla also has an optional server that can be setup to use with a network. Clonezilla can make use of a windows file server, ftp server, and nfs server to store images. It can also be used to create a restore DVD/cd.

Another network based imaging solution is fog project

<http://www.fogproject.org/> Fog is a Linux-based, free and open source computer imaging solution for Windows XP and Vista that ties together a few open-source tools with a php-based web interface. Fog doesn't use any boot disks, or CDs; everything is done via TFTP and PXE. Also with fog many drivers are built into the kernel, so you don't really need to worry about drivers (unless there isn't a linux kernel module for it). Fog also supports putting an image that came from a computer with a 80GB partition onto a machine with a 40GB hard drive as long as the data is less than 40GB.

Cost:

\$0 - NewSID <http://technet.microsoft.com/en-us/sysinternals/bb897418.aspx>

\$0 - Clonezilla <http://clonezilla.org>

\$0 - Parted Magic <http://partedmagic.com>

\$120 - 500 gig USB hard drive

Priceless - Short period of time it takes to get a PC back in service.

I hope you will find this useful.

-Ed Liddle