HAAGA-HELIA
University of Applied Sciences

Automated backup solution for Uponor North America

Antti Tirronen

Bachelor's Thesis

Degree Programme in Business

Information Technology

May 2014

HAAGA-HELIA
University of Applied Sciences

Abstract

May 22 2014

Degree programme

Author or authors Antti Tirronen	**Group or year of entry** 2010
Title of report Automated backup solution for Uponor North America	**Number of report pages and attachment pages** 22+3
Teacher(s) or supervisor(s) Petri Hirvonen	

The goal of the thesis project was to create an automated disk imaging system for Uponor North America.

The thesis describes how to create an inexpensive, open-source disk cloning system by using Debian Linux and Clonezilla Server Edition in combination with scripting to automate the backup process.

The project ended in fall 2013 and the system has been successfully used for weekly automated backups of high-priority computer equipment since fall 2014.

Keywords
Cloning, imaging, backup, Linux, Clonezilla, DRBL

Table of contents

1 Introduction 1

2 Uponor North America 2

3 Core concepts 3

4. Available software 4

4.1 Commerial software 4

4.2 Open-source software 5

4.3 Conclusion 5

5 System design 6

6 Preparing the server 8

6.1 Prerequisites 8

6.2 Hardware 8

6.3 Operating system 8

6.4 Remote access 9

6.5 Network configuration 9

6.6 DRBL 9

6.6.1 Installing DRBL 9

6.6.2 Configuring DRBL 10

6.7 Mail 12

7 Preparing the client 13

7.1 Hardware 13

7.2 BIOS settings 13

8 Manual disk imaging 14

8.1 Creating a system image 1

8.2 Restoring a system image 1

8.3 Cancelling a task 1

9 Automating the backup process 1

9.1 Server-side scripting 1

9.2 Client-side scripting 1

10 Final testing 1

10.1 Creating a system image 1

10.2 Restoring a system image 1

11 Optional features 1

11.1 Expanding the system 1

12 Conclusion 1

References 1

Attachments 1

1 Introduction

The project was started upon request by the author's employer, Uponor North America. The goal of the thesis project was to build an inexpensive disk cloning system for running scheduled disk backups of a computer located in the lobby of the company's Apple Valley headquarters. The computer is used by Uponor's receptionists for receiving incoming calls and forwarding them to departments and employees in the building. Due to the nature of the computer and its importance to the company's telephone system, an automated disk backup system that could create a system image the hard drive and restore it to other disks or computers, was needed. Although one of the goals was to save money, commercial products were also considered and a short description of two of them is included in this report.

The project's purpose was also to serve as a learning experience to the author, who started working in the company in May 2013 as an IT intern and is currently (May 2014) employed as a help desk technician. The author has used similar software in past projects, and this thesis offered a great opportunity to further improve already acquired skills. His responsibilities in the company include desktop support, administering Active Directory users and computers, installing new equipment and various other tasks. Beginning from spring 2014, his responsibilities also include daily server backups using Symantec NetBackup software and Quantum Scalar hardware.

Initial discussions regarding the project were held in May and June 2013, the project started in June 2013 and the server was finally taken into use in September 2013. The final report was prepared during winter 2013 and spring 2014.

2 Uponor North America

Uponor North America (UNA) is the North American branch of Uponor Oyj, a Finland-based company that designs, manufactures and sells fire sprinkling, HVAC (heating, ventilation and air conditioning) and plumbing solutions. UNA is specialized in cross-linked polyethylene (PXE) piping and currently holds the highest market share in North America. (Uponor USA 2014)

Uponor North America has branch offices in both the United States and Canada, its headquarters being in Apple Valley, Minnesota. The headquarters employs approximately 380 people, including the company's Information Technology department. The IT department is responsible for IT infrastructure in all US and Canadian locations. (Uponor USA 2014)

The company has a diverse IT infrastructure that utilizes a wide variety of hardware and software, ranging from office and design software to manufacturing and warehouse management software.

3 Core concepts

Unicasting

Unicasting is a type of network communication, where there is only one sender and one receiver. Unicasting is the most common type of communications in networks around the world. (Fairhurst 2009)

Multicasting

In multicasting, data is sent from one or more sender to a set of receivers. (Fairhurst 2009)

Broadcasting

Broadcasting means communication, where one point sends data to all other points. As an example, ARP (Address Resolution Protocol) uses broadcasting to send resolution queries to computers in the network. (Fairhurst 2009)

4 Available software

Four popular cloning software suites were brought up in meetings with the IT Manager James Quinn. A short summary of the software and their features follows.

4.1 Commercial software

The two commercial backup products that were considered were Norton Ghost from Symantec Corporation and Acronis TrueImage from Acronis International GmbH. According to the companies' websites, Symantec is based in the Mountain View, California and Acronis in Schaffhausen, Switzerland.

Norton Ghost is familiar to the author because it is used by HAAGA-HELIA University of Applied Sciences for both education and internal IT purposes. However, a quick look at Norton's website revealed that the software had been discontinued in spring 2013 and is no longer available for purchase.

Acronis True Image is an inexpensive disk cloning and backup solution for home and business users. At the time of writing this, Acronis' website advertised Acronis True Image 2014 Premium for a price of $99.9.

Currently, Acronis comes in two versions: Acronis True Image 2014 and Acronis True Image 2014 Premium. True Image can back up and restore files and disk images, automate tasks and synchronize files between computers and mobile devices. Acronis also provides 5 gigabytes of cloud space for its users. The Premium edition adds some extra features, namely bare-metal imaging of multiple computers, including computers of different make or model. (Acronis International GmbH 2014)

As proprietary software suites, Symantec and Acronis do not give their users access to their source code and while they offer a wide variety of easy-to-use features, they do not offer the same level of adaptability as their open source rivals.

4.2 Open-source software

The first open-source cloning that was considered was Clonezilla Server Edition. Clonezilla's website advertises Clonezilla as a "Free and Open Source Software for Disk Imaging and cloning". The software is developed by the NCHC Free Software Labs and is licensed under the GNU General Public License 2 (GNU GPL2) license and is therefore free to copy, modify and distribute (

Clonezilla comes in two flavors: Clonezilla Live and Clonezilla Server Edition (SE). The former can be used from an optical disk or a USB flash drive without installing it on the host computer. Clonezilla Live supports unicasting mode only. Clonezilla Server Edition (Clonezilla SE) is the server version and supports unicasting, multicasting and broadcasting modes. Clonezilla SE comes as a part of a larger metapackage called DRBL. DRBL includes a DHCP server, PXE server and other features necessary in order to unleash the full potential of Clonezilla. (DRBL)

Fog is another GNU/Linux based cloning solution. Fog Project's website lists several features that Clonezilla does not have: a PHP-based control interface, tools for Windows-integration and an ability to clone disk images to hard disks that are smaller in capacity than the source disk. (FOG Project)

4.3 Conclusion

Clonezilla Server Edition was chosen because of its open-source license and familiarity to the author of the project. The author has done a multicast cloning project as part of HAAGA-HELIA course Linux-projekti ICT4TN018-1. The project used multicasting to successfully clone the contents of one hard drive to all twenty-five computers of a computer lab and remotely manage them with Puppet, an open-source configuration management tool.

5 System design

The latest stable version of the Debian operating system, Debian 7 Wheezy, was chosen as the basis for the Clonezilla server. Debian is an open-source operating system that utilizes the Linux kernel. Debian was chosen because of its popularity as a server operating system and because the author has utilized Ubuntu, a Debian derivative, in the project mentioned in chapter 3.3. According to W3Techs Web Technology Surveys, Debian is currently the most popular web server operating system. The 64-bit version was chosen because it offers better performance and features than the 32-bit version.

The server is a Hewlett-Packard XW4600 workstation with an Intel Core 2 Duo E6550 processor, four gigabytes of random-access memory and a 250 gigabyte hard drive. The receptionist's computer is a HP Compaq DC7900 workstation with an Intel Core 2 Quad Q8300 processor, four gigabytes of random-access memory and a 250 gigabyte hard drive.

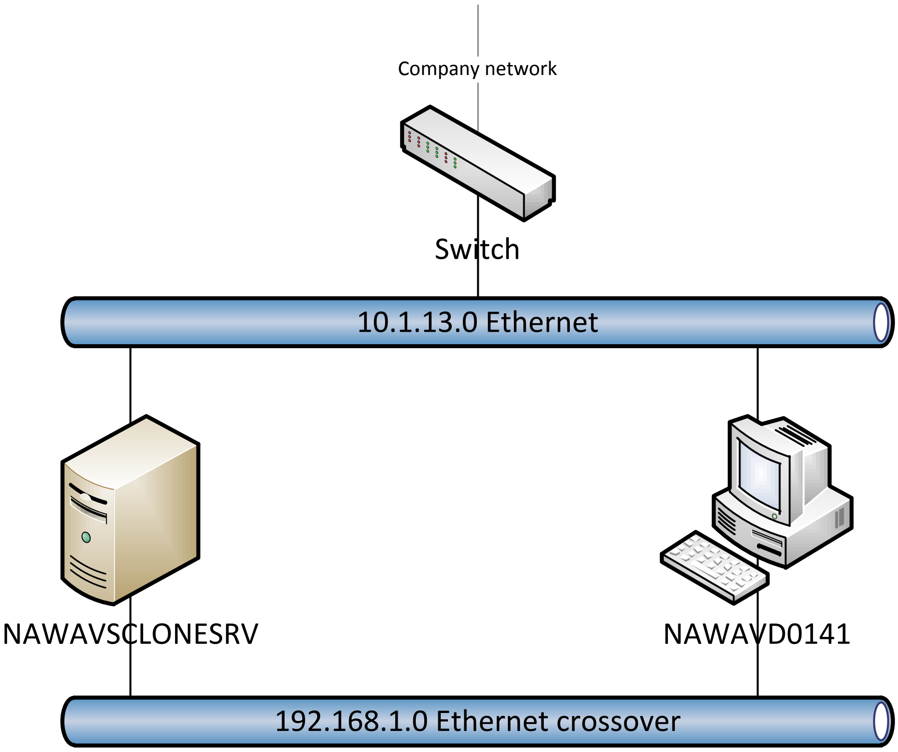


Figure 1. System design.

Each computer has two network adapters: one integrated adapter and one Intel add-on adapter installed to the PCI port. The phone system software installed on the receptionist's computer has a license key that's tied to the MAC address of the computer's network card. Because of this and as a failsafe in case of a need to transfer the license to another computer, the computer has two network adapters: one integrated on the motherboard and one that is plugged into a PCI port. The integrated adapter is connected to the cloning server with a crossover cable and the PCI network adapter connects the computer to the company network. A crossover cable was used in order to minimize changes to the company's network infrastructure and configuration (figure 1).

6 Preparing the server

6.1 Prerequisites

The latest 64-bit Debian 7 image was downloaded from Debian's website. This image file was used for installing the operating system on the server. The ISO image is a network install or "netinst" CD, which includes only the most essentials components of the operating system. Other packages are downloaded from the Internet during installation.

The image was written to a flash drive by using Universal USB Installer, which is available from Pendrivelinux.com.

6.2 Hardware

An Intel Pro 1000 network adapter was installed in one of the PCI slots in order to have two separate physical network interfaces. The computer was thoroughly inspected to make sure it suits the project's requirements. No other hardware changes were required.

6.3 Operating system

In order to boot from the flash drive, the boot order settings had to be changed on the server. This was done by starting the computer and pressing the F10 key when the HP splash screen appeared on the screen. The settings were found from under the Boot options.

After rebooting the computer it was booted from the flash drive by pressing the F9 key during start up and selecting the flash drive as the boot source. When the Debian installer screen appeared, "Install" was selected as the boot option. English, United States and American English were selected as the Language, locale and keymap settings, respectively. Hostname was set as nawavsclonesrv and domain name as uponor.local. A root password was set up for administrative purposes and a normal user account, uponor, was created for daily use. Time zone was set as Central. Partitioning method was default (Guided - use entire disk) and since the computer has only one hard drive, there was only one disk partition to select from.

6.4 Remote access

In order to allow remote management of the server, an SSH server had to be installed. The package that was installed is called openssh-server and is available from the default Debian software repositories. For security reasons, file /etc/ssh/sshd_config was edited to deactivate root access. This was done by changing the PermitRootLogin setting to no.

A VNC (Virtual Network Computing) server called vnc4server was installed to allow graphical remote access. File /home/uponor/.vnc/xstartup had to be edited to include the line gnome-session &, otherwise the Gnome desktop environment failed to load when viewed from VNC. Gnome 3 interface still fails to load but the Classic interface works fine in VNC. The VNC server was configured to accept the same password as the Windows computers used in the company.

6.5 Network configuration

DRBL's online documentation states that two separate network interfaces is required in order to achieve best results when using multicasting mode. This can be achieved by configuring a virtual interface in /etc/network/interfaces, or, as DRBL's official documentation recommends, by using two physical network interfaces. This project does not utilize multicasting, but in order to support better scalability after possible future upgrades, an additional network card was installed in one of the server's PCI ports.

6.6 DRBL

6.6.1 Installing DRBL

Installation began by adding the DRBL package repository to the sources list:

```
nano /etc/apt/sources.list.d/drbl.list
```

The following lines were added to the file:

```
# Diskless Remote Boot in Linux
deb http://drbl.sourceforge.net/drbl-core drbl stable
```

The repository's public encryption key was added to the system:

```
wget http://drbl.sourceforge.net/GPG-KEY-DRBL
```

Package list was updated:

```
apt-get update
```

Finally, the drbl metapackage was installed:

```
apt-get install drbl
```

The DRBL installation utility was started with the drblsrv command accompanied by the parameter that tells it to start installing the software:

```
drblsrv –i
```

The setup asked a series of questions about the DRBL server. All default settings were suitable for our needs so they were all accepted by pressing the Enter key.

6.6.2 Configuring DRBL

The next step in the setup process was to configure the DRBL server. This was done with the following command:

```
drblpush –i
```

The default settings were accepted except for a few exceptions:

```
Now we can collect the MAC address of clients!
If you want to let the DHCP service in DRBL server offer same IP address to client every time when client boot, and you never did this procedure, you should do it now!
If you already have those MAC addresses of clients, you can put them into different group files (These files number is the same number of networks cards for DRBL service). In this case, you can skip this step.
This step helps you to record the MAC addresses of clients, then divide them into different groups. It will save your time and reduce the typos.
The MAC addresses will be recorded turn by turn according to the boot of clients,
and they will be put into different files according to the network card in server, file name will be like macadr-eth1.txt, macadr-eth2.txt... You can find them in directory /etc/drbl.
Please boot the clients by order, make sure they boot from etherboot or PXE!
Do you want to collect them ?
[y/N]
```

The setup utility asked if the user wants to collect the MAC addresses of the clients. This makes it possible for the built-in DHCP server to offer the same IP to the clients every time they boot. The default answer to this option is no (N) but this was changed to yes (y). After changing the setting, the client computer was turned on. Since network boot was already set as the default boot option, no other client-side actions were necessary at this point. When

The setup utility also asked if the user wants to use static IPs for its clients:

> Do you want to let the DHCP service in DRBL server offer same IP address to the client every time when client boots (If you want this function, you have to collect the MAC addresses of clients, and save them in file(s) (as in the previous procedure)). This is for the clients connected to DRBL server's ethernet network interface eth1 ?
> [y/N]

The default setting is no and this was changed to yes.

6.7 Mail

A mail client was needed in order to send email reports from the server to Uponor's IT help desk. Debian comes with a simple, terminal-based email application called Mail that has all the basic features needed for this task. However, some configuration changes were required in order to allow the server to send emails through the company's Exchange server. To do this, the sendmail.cf configuration file located in /etc/mail/ was modified to include the relay server's name and the sender address. Some changes were also required on the Exchange server: the server's receive connector was updated to allow the static IP of the Clonezilla server to send email.

7 Preparing the client

7.1 Hardware

The computer uses Avaya one-X Attendant software for call forwarding and the software license key is tied to the MAC address of the computer. Because of this, the computer has two network adapters: one integrated adapter and one Intel Pro 1000 adapter in one of the PCI ports of the motherboard. The PCI card allows moving the license to another PC in the event of a hardware failure. This also makes the backup process easier, as the computer was ready to use two network connections.

7.2 BIOS settings

In order to boot to the server, the target computer must have network boot enabled in the BIOS settings. This is done by pressing the F10 during startup, entering the security password and navigating to the boot settings menu.

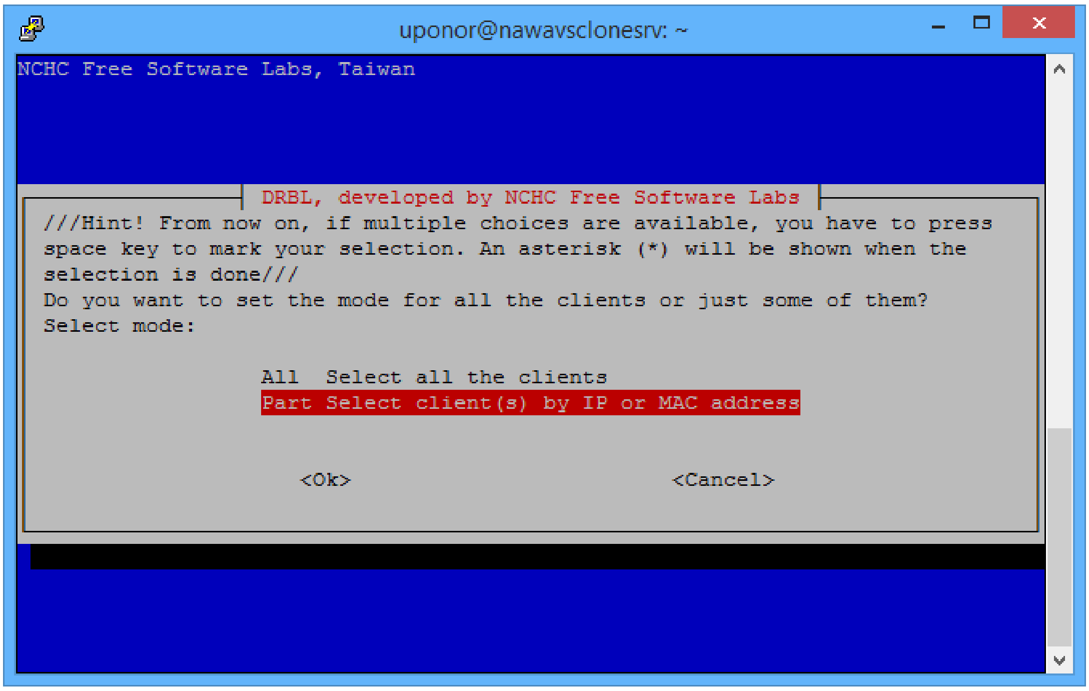
To use both WOL and PXE together, the BIOS setting that defines the remote wakeup boot source also had to be changed. The remote wakeup setting can be changed under the Advanced menu and the Power-On Options submenu by changing the Remote wakeup boot source form Local hard drive to Remote server.

The computer's power settings were also changed in the BIOS so that the computer would automatically boot immediately when power is restored.

Finally, to reboot the computer after a power outage, power management settings were also changed to power on the computer immediately after the outage is over.

8 Manual disk imaging

8.1 Creating a system image



```
NCHC Free Software Labs, Taiwan

DRBL, developed by NCHC Free Software Labs
///Hint! From now on, if multiple choices are available, you have to press
space key to mark your selection. An asterisk (*) will be shown when the
selection is done///
Do you want to set the mode for all the clients or just some of them?
Select mode:

                 All  Select all the clients
                 Part Select client(s) by IP or MAC address

                    <Ok>                        <Cancel>
```

Figure 2. Clonezilla SE interface.

The control interface (figure 2) is accessed by entering the command dcs in the Linux terminal. The command must be entered either as the root user or by using sudo.

To manually create a disk image, the user does the following:

1. Types sudo dcs, enters the password and hits Enter
2. Uses the arrow keys to select Part Select client(s) by IP or MAC address and hits Enter
3. Selects by_MAC_addr_list Set_mode_by_clients_MAC_address_list and hits Enter
4. Uses the arrow keys to select the correct host (there should be only one), selects it by hitting the Space key and hits Enter
5. Selects clonezilla-start
6. Selects Beginner

7. Selects save-disk
8. Selects reboot
9. Selects Now_in_server. If the user selects Later_in_client, user interaction is required on the host computer when it enters the cloning interface.
10. User
11. Names the disk image.
12. Selects the target disk or partition.
13. Selects whether the user wants to check and repair the file system before saving it. If there is no reason to suspect that the file system is damaged, the checking can be skipped by hitting Enter.
14. Selects whether the saved image is to be checked or not. The default is Yes.
15. Selects the action to perform when client finishes cloning. The action can be reboot, poweroff, choose in client or do nothing.

8.2 Restoring a system image

In order to restore a disk manually, the user does the following:

1. Types sudo dcs, enters the password and hits Enter
2. Uses the arrow keys to select Part Select client(s) by IP or MAC address and hits Enter
3. Selects by_MAC_addr_list Set_mode_by_clients_MAC_address_list and hits Enter
4. Uses the arrow keys to select the correct host (there should be only one), selects it by hitting the Space key and hits Enter
5. Selects clonezilla-start
6. Selects Beginner
7. Selects restore-disk
8. Selects reboot
9. Selects the image he want to restore.
10. Selects the target disk (there should be only one, sda) with the arrow keys, hits Space and Enter. Note: the target disk has to be identical in size or bigger than the source disk or partition!

11. Selects unicast and hit Enter

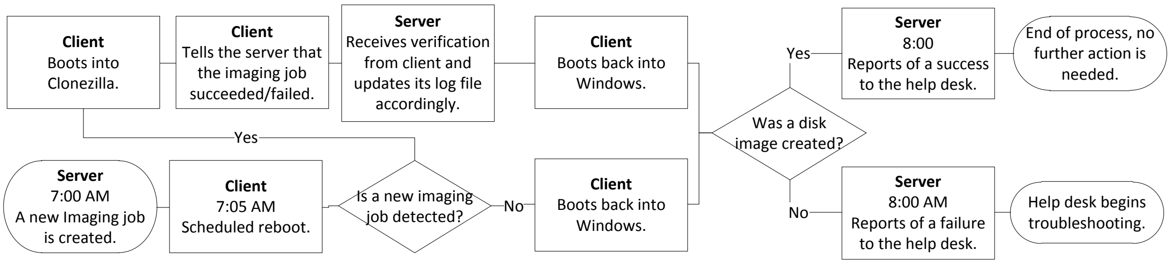
8.3 Cancelling a task

To cancel an imaging task, the user types sudo dcs to open the user interface, enters the password and does the following:

1. Uses the arrow keys to select Part Select client(s) by IP or MAC address and hits Enter
2. Selects by_MAC_addr_list Set_mode_by_clients_MAC_address_list and hits Enter
3. Uses the arrow keys to select the correct host (there should be only one), selects it by hitting the Space key and hits Enter
4. Selects clonezilla-stop. All queued tasks will be cancelled.

9 Automating the backup process

9.1 Server-side scripting



Client
Boots into Clonezilla.
Client
Tells the server that the imaging job succeeded/failed.
Server
Receives verification from client and updates its log file accordingly.
Client
Boots back into Windows.
Yes
Server
7:00 AM
A new Imaging job is created.
Client
7:05 AM
Scheduled reboot.
Is a new imaging job detected?
No
Client
Boots back into Windows.
Was a disk image created?
Yes
Server
8:00
Reports of a success to the help desk.
End of process, no further action is needed.
No
Server
8:00 AM
Reports of a failure to the help desk.
Help desk begins troubleshooting.

Figure 3. Automated backup process.

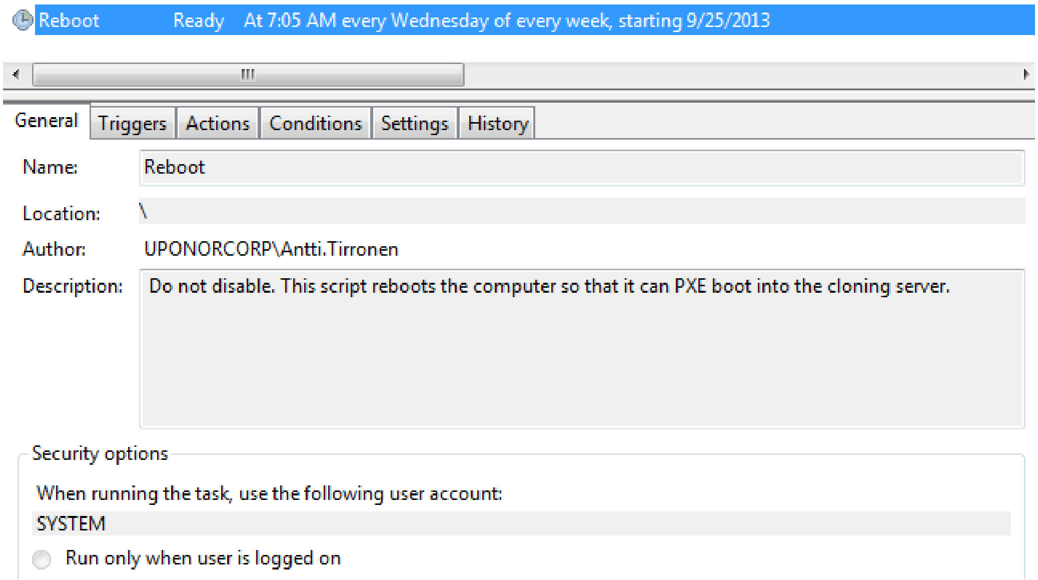
In order to fully automate disk imaging jobs, the server must do the following: start a new backup job, check its end results and send reports to the company's help desk. These tasks are handled by two scripts, drbl.sh (attachment 2) and email.sh (attachment 3). The two scripts are invoked by crontab (attachment 1).

The first script, drbl.sh, starts the imaging process by:

1. Creating a variable for naming the image. The name includes the name of the host and the date the image was created.
2. Writing a timestamp in the log file /opt/drbl.log
3. Telling the server to create a system image of the host and sends the output to the log file

The second script, email.sh , emails the results of the job to Uponor North America's IT help desk and tells if the imaging job succeeded or failed. The email report also includes a list of all disk images that were available on the server when the script was run.

9.2 Client-side scripting



Reboot Ready At 7:05 AM every Wednesday of every week, starting 9/25/2013

Name: Reboot

Location: \

Author: UPONORCORP\Antti.Tirronen

Description: Do not disable. This script reboots the computer so that it can PXE boot into the cloning server.

Security options

When running the task, use the following user account:

SYSTEM

○ Run only when user is logged on

Figure 4. Windows Task Scheduler.

A simple script called reboot.bat is run by Windows Task Scheduler every Wednesday morning (figure 4). The script reboots the computer so that it can boot into the PXE server.

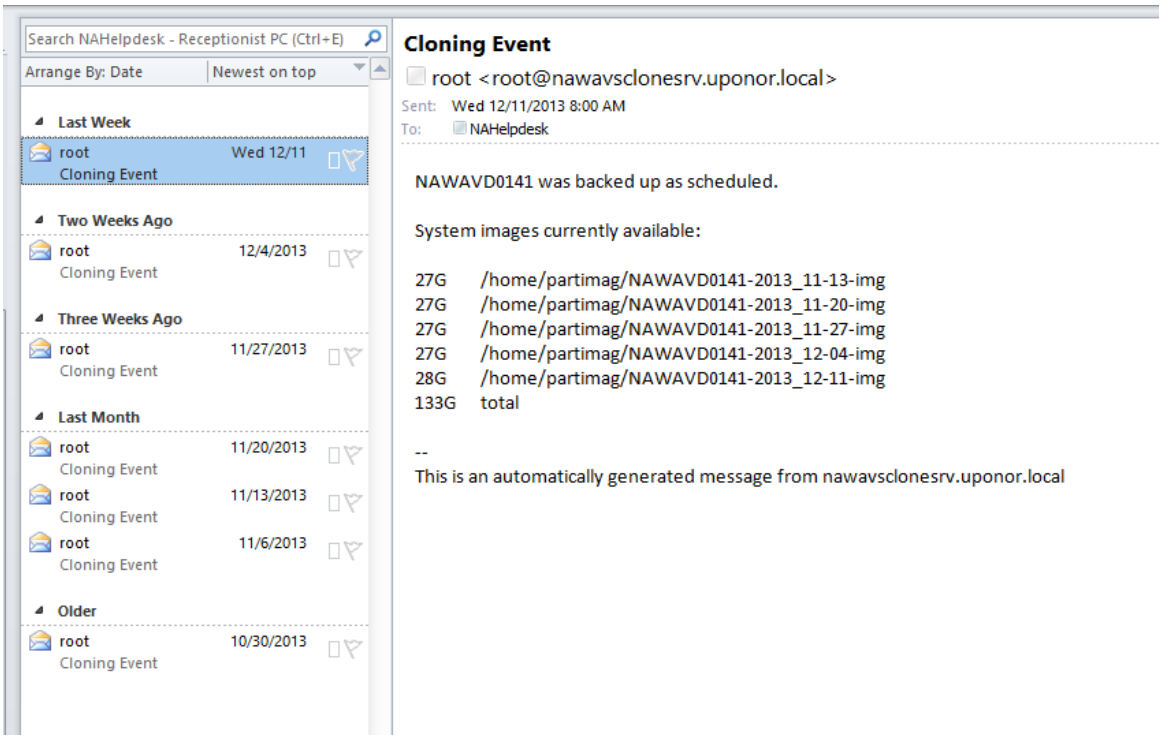
10 Final testing

A series of tests were conducted in order to evaluate the system and to find out if it works as intended. The first tests were simple imaging tests where an image was created but not restored to the disk. In the final test, a disk image was created of the hard drive, the hard drive was replaced with an identical SATA drive and then the image was restored to the drive. The test followed the schedule of a normal weekly backup.

10.1 Creating a system image

The backup process is described below.

1. 7:00 AM CET: bash script /opt/drbl.sh is run on the server as a scheduled task. The script creates the imaging job and tells the host to PXE boot into the server next time it restarts.
2. 7:05 AM CET: batch file C:\reboot.bat is run on the host as a scheduled task. The script reboots the computer.
3. The system image is saved on the server. Everything is automated. This takes approximately 15 minutes.
4. 8:00 AM CET: bash script /opt/email.sh is run on the server as a scheduled task. The script checks if the new system image exists on the server and sends an email to nahelpdesk@uponor.com.



Cloning Event

root <root@nawavsclonesrv.uponor.local>

Sent: Wed 12/11/2013 8:00 AM

To: NAHelpdesk

```
NAWAVD0141 was backed up as scheduled.

System images currently available:

27G    /home/partimag/NAWAVD0141-2013_11-13-img
27G    /home/partimag/NAWAVD0141-2013_11-20-img
27G    /home/partimag/NAWAVD0141-2013_11-27-img
27G    /home/partimag/NAWAVD0141-2013_12-04-img
28G    /home/partimag/NAWAVD0141-2013_12-11-img
133G   total

--
This is an automatically generated message from nawavsclonesrv.uponor.local
```

Figure 5. An example of an email received from the server.

All the scripts run once a week every Wednesday. The backup process can also be run manually by running the scripts from the terminal and restarting the target computer.

10.2 Restoring a system image

The hard drive of the computer was removed replaced with an identical SATA drive to simulate a hard drive failure. An SSH connection was opened to the server and the disk image that was created in chapter 9.1 was restored as described in chapter 7.2. The computer restarted successfully and booted into Windows 7. The primary users of the computer were able to open Avaya one-X Attendant and log in to the software. No issues were detected with the software license.

The test was successful and the author received approval from the IT department to deploy the system.

11 Optional features and future developments

11.1 Expanding the system

The system could be expanded to the company network to allow disk cloning of multiple systems. The system configuration is displayed in figure 4.

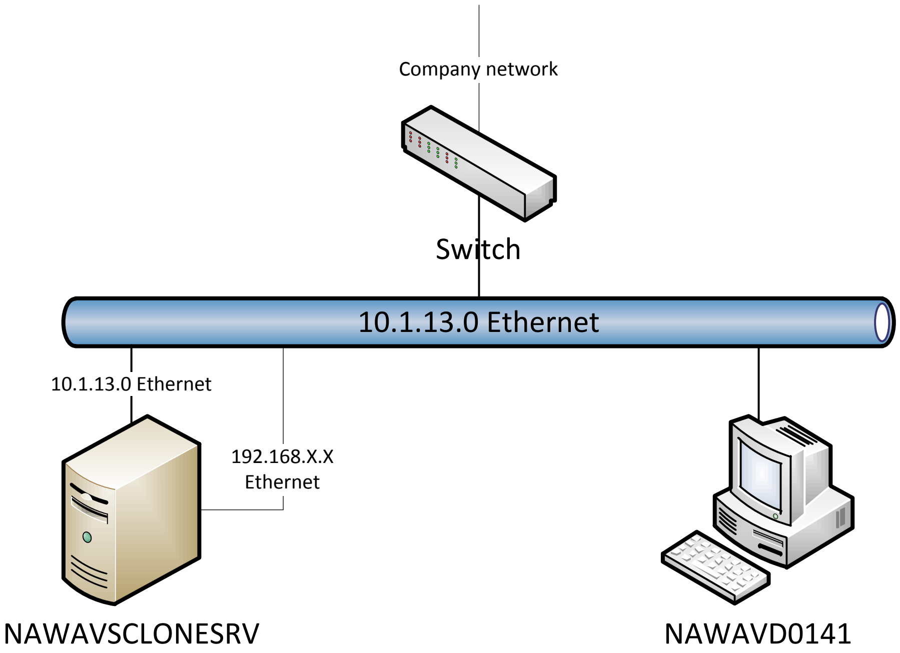


Figure 6. An optional layout for the system.

In the expanded configuration (figure 6), the network adapters that are currently connected via a crossover cable would be connected to the company network via a regular Ethernet cable like the main adapters. If the hosts cross networks, system expansion would require changes in router configuration, so that the routers know how to route traffic to the C-class network used for Clonezilla SE.

12 Conclusion

All project goals were reached and the disk cloning system fulfilled all its requirements. As of May 2014, has been in constant use for approximately eight months without any issues. It has not been used in a real catastrophe recovery situation, but has performed well in its weekly tasks.

The total cost of the system was zero dollars, as all the hardware used was taken from old computers and all the software is free and open-source.

The system is scalable and can, with only minor changes to the system architecture, be used with multiple hosts. This would require changes to network infrastructure and cabling, but changes to the server itself would be quite small. Network interfaces and DRBL would have to be reconfigured but other changes to the server would be quite small. The same scripts can, with small changes, be used with multiple hosts.

As another possible future upgrade, the client-side script (reboot.bat) could be eliminated by using Winexe, a remote execution tool very similar to PsExec, a Microsoft tool used for executing commands on remote Windows computers. Psexec would allow restarting the computer and saving disk images without having to run any scripts or other tasks outside the server.

The project turned out to be a great learning experience for the author. There isn't much information available regarding automating Clonezilla, and the project offered a great opportunity to find the right way to do it.

References

Uponor 2014. URL: http://www.uponor.com/en/company/history.aspx. Accessed: Jan 5 2014.

Uponor 2014. URL: http://investors.uponor.com/financials/fact-sheet. Accessed: Jan 5 2014.

Uponor USA 2014. http://www.uponor-usa.com/Company/Uponor-North-America.aspx. Accessed: Jan 5 2014.

Fairhurst, G. University of Aberdeen. Local Area Networks. 2009. URL: http://www.erg.abdn.ac.uk/~gorry/course/intro-pages/uni-b-mcast.html. Accessed April 20 2014.

Symantec. 2013. URL: http://www.ghost.com/. Accessed: June 5 2013.

W3Techs - World Wide Web Technology Surveys. 2014. URL: http://w3techs.com/technologies/details/os-linux/all/all. Accessed: Apr 20 2014.

HP 2005. HP support forum - PXE And WOL not working as aspected. URL: http://h30499.www3.hp.com/t5/Business-PCs-Compaq-Elite-Pro/PXE-And-WOL-not-working-as-aspected/td-p/540290#.Ug5gepLVDIY. Accessed: Sep 10 2013.

DRBL. About DRBL. URL: http://drbl.org/about/. Accessed: Sep 10 2013.

DRBL. FAQ. URL: http://drbl.org/faq/. Accessed: Sep 10 2013.

Clonezilla. How to setup a Clonezilla server? URL: http://clonezilla.org/clonezilla-SE/. Accessed: Sep 5 2013.

FOG Project. Overview – What is FOG? 2014. URL: http://www.fogproject.org/?q=node/1. Accessed: Feb 3 2014.

Software in the Public Interest, Inc. 2013. http://www.debian.org/CD/netinst/. Accessed: Sep 5 2013.

Acronis International GmbH. Acronis True Image Comparison. http://www.acronis.com/en-us/personal/true-image-comparison. Accessed: Apr 2 2014.

Attachments

Attachment 1. Contents of crontab-e.

```
GNU nano 2.2.6        File: /tmp/crontab.LBvfHR/crontab

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
00 07 * * wed /bin/sh /opt/drbl.sh
00 08 * * wed /bin/sh /opt/email.sh
```

Attachment 2. Batch script drbl.sh.

```
#!/bin/bash

# Copyright 2013 Antti Tirronen
# License: GNU General Public License, version 2 or later

# Description:
# Creates a system image of NAWAVD0141

# Variable for naming the image
NAME="NAWAVD0141-"$(date +%Y_%m-%d)"-img"

# Writes a timestamp in the log file
(/usr/bin/printf
"*****************************************************.\n**********************
*******************************." && /usr/bin/printf "\n" && /bin/date) >>
/opt/drbl.log &&

# Tells the server to create system image of the host and sends the output to a log file
/usr/sbin/drbl-ocs -b -q2 -j2 -sc -p reboot -z1p -i 1000000 -h " 192.168.1.1" -l
en_US.UTF-8 startdisk save $NAME sda >> /opt/drbl.log 2>&1
```

Attachment 3. Batch script email.sh

```
#!/bin/bash

# Copyright 2013 Antti Tirronen
# License: GNU General Public License, version 2 or later

# Description:
# Checks if NAWAVD0141 system backup finished properly and sends an email to nahelpdesk@uponor.com


# If an image newer than 24 hours is found from /home/partimag, the server sends an email to nahelpdesk@uponor.com and deletes all images that are older than two weeks
if [ -n "$(find /home/partimag -name "*img*" -type d -ctime -1)" ]; then
        find /home/partimag -type d -ctime +30 -exec rm -rf {} + &&
        (printf "NAWAVD0141 was backed up as scheduled.\n\n" && printf "System images currently available:\n\n" && du -ch /home/partimag/*img* && printf "\n--\nThis is an automatically generated message fro$
# If the image is not found, the server sends an email to nahelpdesk@uponor.com
else
       (printf "ERROR: NAWAVD0141 was not backed up.\n\n" && printf "System images currently available:\n\n" && du -ch /home/partimag/*img* && printf "\n--\nThis is an automatically generated message from n$
fi
```