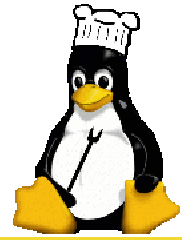# The Open Pitt

What's cooking in Linux and Open Source in Western Pennsylvania

Book Review: *Practical Virtualization Solutions* by Beth Lynn Eicher

Authors: Kenneth Hess, Amy Newman
Publisher: Prentice Hall
ISBN: 0-13-714297-8
$39.99, 336 pages, 2010

The Western Pennsylvania Linux Users Group uses virtualization to host our web site, `www.wplug.org`. It is housed on a cloud-based Xen-powered virtual private server on a physical host run by a company called Linode. Instead of worrying about the expense of maintaining our own hardware, paying for a network connection, and leasing server room rack space, we pay Linode to rent a portion of a server. Over the web, the WPLUG Internet committee can reboot, reinstall the operating system, and configure services with full root access to our own virtual server. We virtually have our own system residing in one of Linode's data centers across the United States and Europe.

"Virtualization" and "cloud computing" are the hottest buzzwords in technology today; however, few understand what these terms mean. To make matters even more confusing, there is a sea of products for virtualization solutions, hardware vendors who sell servers that are virtualization ready, and cloud vendors that lease or timeshare virtualized hosts. This book attempts to explain the many virtualization options for a system administrator audience.

Not everyone uses virtualization like WPLUG does. Some use virtualization for temporary projects such as trying new operating systems, demonstrating a technology for educational reasons, simulating networks for future studies, or development on emulated hardware platforms. The focus of this book is building a virtualization infrastructure for a company. If is not your job to decide how to manage your servers, chapter 8 describes VirtualBox, a free and open source virtualization solution for desktop users. This book is primarily targeted toward system administrators who want to use virtualization for their business needs.

The authors compare the different types of virtualization, discussing how to choose the virtualization solution that would work best for your company. They also cover making the decision between outsourcing your virtualization to a cloud-based vendor or keeping things in house. An important non-technical point in the book is selling management on virtualization. Case studies of successful virtualization deployments round out the book and provide useful real-world examples.

Rolling out virtualization in a corporate infrasructure is a huge project. This book overall does a good job explaining the reasons why a commercial virtualization solution matches an enterprise's needs. Unfortunately, it was published too soon to discuss the new Red Hat Enterprise Virtualization product. It would have been nice if the book discussed the differences between the XenWorks product by Citrix and Xen.org, an open-source project which is heavily sponsored by Oracle and Novell. None the less, this book is the most complete and up-to-date resource for evaluating virtualization solutions for an enterprise environment.

Beth Lynn Eicher is a former board member of the Western Pennsylvania Linux Users Group. She also writes for `whatwillweuse.com`, *a technology blog which watches Microsoft's market share.*

April/May Roundup

Apr. 10 General User Meeting: **Beth Lynn Eicher** spoke about the social benefits of Linux and other Free and Open Source software. After covering the basic philosophy behind it, she discussed what advantages it may have for home use, in business, and in education. She concluded that Free and Open Source software is a unique and effective way to empower nearly any community.

May 8 General User Meeting: **Ted Rodgers** discussed deploying and managing Linux clusters in a high-performance computing environment and the challenges of scaling normal administration methods to dozens or hundreds of servers. He described the network (PXE) booting process and went through the details of setting up the necessary services and network configuration. Also covered were methods for quickly cloning systems over the network, and a comparison with more traditional automated installation tools.

UNIX Curio

This is the first column in a series dedicated to exploring little-known—and occasionally use-ful—trinkets lurking in the dusty corners of UNIX-like operating systems.

This month's column was inspired by an article on the Linux Journal web site <`http://www.linuxjournal.com/content/add-binary-payload-your-shell-scripts`> describing a custom-built script that would contain a binary tar archive and, when run, would extract the contents onto the user's system. Upon reading this, memories immediately came rushing back of the days of Usenet, before MIME-encoded e-mail made sending file attachments standard, and where we walked ten miles each way to school (uphill both ways!) in three feet of snow.

Yes, at that time, you had to put everything into the body of your message. But what if you needed to send a bunch of files to someone? There was *tar*, but the format dif-fered between systems, and e-mail and Usenet could only reliably handle 7-bit plain-text ASCII any-how. You could send separate e-mail messages (but what if one goes missing?) or put "CUT HERE" lines to designate where one file ends and another one begins (tedious for the recipient). The solution was a shell archive created by the *shar* program. This wraps all your files in a neat shell script that the recipient can just run and have the files magically pop out. All he needs is the Bourne shell and the *sed* utility, both standard on any UNIX-like system.

Suppose you had a directory named "foo" containing the files bar.c, bar.h, and bar.txt, and wanted to send these. All you'd need to do is run the following command, and your archive is on its way.

```
$ shar foo foo/* | mail -s
"Foo 1.0 files" bob@example.com
```

When the recipient runs the resulting script, it will create the foo directory and copy out the files onto his system. You can also pick and choose files; if you wanted to leave out bar.txt, you could do '`shar foo foo/bar.c foo/bar.h`' or, more simply, '`shar foo foo/bar.?`'.

Different versions of *shar* have varying capabilities. For example, the BSD and OS X editions can only really manage plain-text files. If you had a binary object file bar.o, it'd likely get mangled somewhere along the way if you tried to include it in an archive. They also require, as in the examples above, that you name a directory before naming any files inside it (the typical way is to let the *find* command do the work for you; it produces a list in the right order).

The GNU implementation is more flexible and can take just a directory name, automatically includ-ing everything underneath. It can also handle binary files by using uuencode—a method for encoding data as ASCII that predated the current base64 MIME standard. GNU *shar* rather nicely auto-detects whether the input file is text or binary and acts accordingly, and can even compress files if asked. However, unpacking encoded or compressed files from such an archive requires the recipient to have the corres-ponding decode/uncompress utility, and the documentation is littered with (now somewhat anachronistic) warnings about this.

Looking at other UNIX systems, the HP-UX version also can uuencode binary files, and as a special bonus adds logic to the script that will compile and use a simple uudecode tool if the recipient doesn't already have one. It will even han-dle device files and put the corres-ponding *mknod* commands into the script, probably making it the most full-featured implementation of all. IBM's AIX doesn't appear to come with *shar*. Neither do SunOS and Solaris, which seems quite odd as original development of the program is credited to James Gosling!

And so we bid farewell to *shar*. Next time you're considering rolling your own script for a particular purpose, consider whether such a tool might already exist, just waiting on your system for you to use it.

What is Linux?
Linux is a *kernel,* the core of a computer oper-ating system, created by Linus Torvalds. It is typically packaged as a *distribution,* which in-cludes the extra programs necessary to make a computer functional and useful. Since 1991, it has grown from a one-man project which ran on one computer to one with thousands of con-tributors running on everything from mobile phones to million-dollar supercomputers.

What are Open Source and Free Software?
Open Source and Free Software provide you, the user, with the opportunity to see the source code of the programs you use. You are free to use it, share it with others, and even make changes to it if you wish. While the Free Soft-ware and Open Source communities differ in their philosophical approach, in practical terms they share nearly identical goals. Learn more at <`http://www.opensource.org/`> and <`http://www.gnu.org/`>.

This newsletter was produced using Open Source and Free Software.

Pitch In!

The Open Pitt can only exist if people like you contribute articles. Have something to say on a topic related to Free and Open Source software? Get in contact with us at `top@wplug.org` and get your ideas out to the WPLUG community and the wider world!

Don't worry if your writing skills aren't the best. Bring your ideas, and we'll be glad to help edit your article and guide you along the process.

If you don't have a topic in mind, consider picking up a review copy of a book at one of WPLUG's meetings. We can always use more book reviews to point out useful references to the community.

In short, only you can prevent *The Open Pitt* from going on hiatus again. Give us some articles to work with, and we can keep it going strong for years to come!