

SQLAlchemy | ConVirt | Fabric | Azure | Raspberry Pi

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

SYSTEM ADMINISTRATION

FEBRUARY 2013 | ISSUE 226 | www.linuxjournal.com

FREE TO SUBSCRIBERS
Epub, Kindle, Android, iPhone & iPad editions

Manage Your Linux Deployment with **ConVirt**
Play with Your Linux Boxes

A Look at **SQLAlchemy's Features**

HOW TO:
Handling R Packages

Manage Your Virtual Deployment with **ConVirt**

Spin up Linux VMs on **Azure**

Use **Fabric** for Sysadmin Tasks on Remote Machines

PLUS: Use a Raspberry Pi as a Colocated Server





**THIS TIME
WE'RE TURNING IT UP
TO 11!**

The 11th Annual
Southern California Linux Expo

Featuring Performances by:

- * Matthew Garrett
- * 70 other Open Source Acts!

With Special Events by:

- * PostgreSQL and MySQL Community Days
- * DevOps Day LA
- * UbuCon
- * SCALE U
and more...



SCALE 11x

February 22-24, 2013
Hilton Hotel @ LAX
Los Angeles, CA

<http://www.socallinuxexpo.org>
Use Promo Code LJ11X for a 30%
discount on admission to SCALE

SILICON MECHANICS



visit us at www.siliconmechanics.com or call us toll free at 888-352-1173

RACKMOUNT SERVERS STORAGE SOLUTIONS HIGH-PERFORMANCE COMPUTING

**“Just because
it’s badass,
doesn’t mean
it’s a game.”**

Pierre, our new Operations Manager, is always looking for the right tools to get more work done in less time. That’s why he respects NVIDIA® Tesla® GPUs: he sees customers return again and again for more server products featuring hybrid CPU / GPU computing, like the Silicon Mechanics Hyperform HPCg R2504.v3.

We start with your choice of two state-of-the-art processors, for fast, reliable, energy-efficient processing. Then we add four NVIDIA® Tesla® GPUs, to dramatically accelerate parallel processing for applications like ray tracing and finite element analysis. Load it up with DDR3 memory, and you have herculean capabilities and an 80 PLUS Platinum Certified power supply, all in the space of a 4U server.



When you partner with Silicon Mechanics, you get more than stellar technology - you get an Expert like Pierre.

Expert included.

SYSTEM ADMINISTRATION

FEATURES

68 **ConVirt: the New Tool in Your Virtual Toolbox**

Take a look at a great, open-source tool for managing multiple types of hypervisors.

Jeramiah Bowling

82 **Linux on Azure—a Strange Place to Find a Penguin**

On the heels of Amazon and Google, Microsoft has jumped into the Infrastructure-as-a-Service world, offering cloud-based Linux virtual machines.

Andrew Fabbro

94 **Fabric: a System Administrator's Best Friend**

No sysadmin should be unaware of this tool for managing machines simultaneously.

Adrian Hannah



COVER IMAGE: © Can Stock Photo Inc. / buchachon

INDEPTH

104 Making Linux and Android Get Along (It's Not as Hard as It Sounds)

How to make Android devices play with your Linux systems.

Aaron Peters

COLUMNS

36 Reuven M. Lerner's At the Forge

SQLAlchemy

44 Dave Taylor's Work the Shell

Cribbage: Sorting Your Hand

48 Kyle Rankin's Hack and /

Raspberry Strudel: My Raspberry Pi in Austria

54 Shawn Powers' The Open-Source Classroom

Trying to Tame the Tablet

120 Doc Searls' EOF

Leaving the Land of the Giants

IN EVERY ISSUE

8 Current_Issue.tar.gz

10 Letters

16 UPFRONT

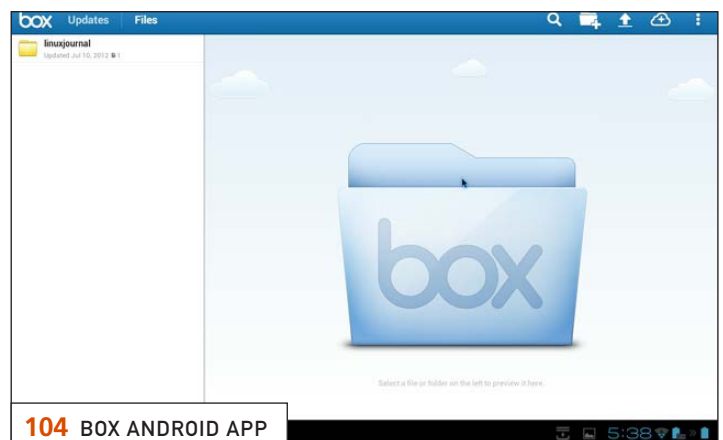
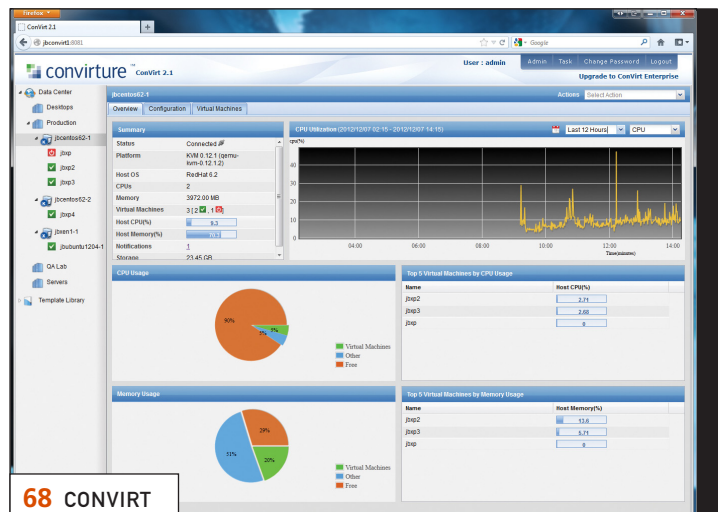
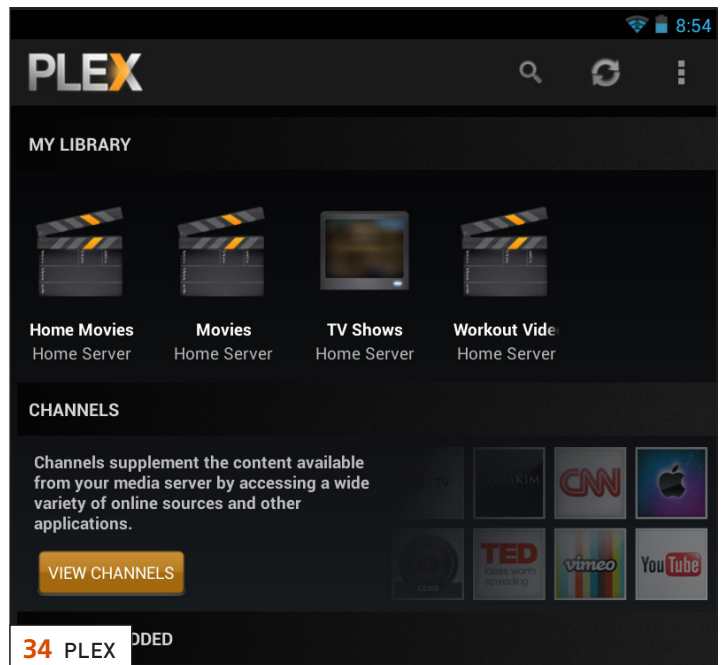
34 Editors' Choice

64 New Products

129 Advertisers Index

ON THE COVER

- Make Your Android Device Play with Your Linux Box, p. 104
- A Look at SQLAlchemy's Features, p. 36
- Manage Your Virtual Deployment with ConVirt, p. 68
- Spin up Linux VMs on Azure, p. 82
- Use Fabric for Sysadmin Tasks on Remote Machines, p. 94
- How To: Handling R Packages, p. 24
- Plus: Use a Raspberry Pi as a Colocated Server, p. 48



LINUX JOURNAL™

Subscribe to
Linux Journal
Digital Edition
for only
\$2.45 an issue.



ENJOY:

- Timely delivery
- Off-line reading
- Easy navigation
- Phrase search and highlighting
- Ability to save, clip and share articles
- Embedded videos
- Android & iOS apps, desktop and e-Reader versions

SUBSCRIBE TODAY!

LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors

Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan

Publisher	Carlie Fairchild publisher@linuxjournal.com
Director of Sales	John Grogan john@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

**Linux Journal is published by, and is a registered trade name of,
Belltown Media, Inc.**

PO Box 980985, Houston, TX 77098 USA

Editorial Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruiuzenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

Advertising

E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 980985, Houston, TX 77098 USA

LINUX is a registered trademark of Linus Torvalds.

TrueNAS™ Storage Appliances Harness the Cloud



Unified. Scalable. Flexible.

Thanks to the Intel® Xeon® Processor 5600 series and high-performance flash, every TrueNAS Storage appliance delivers the utmost in throughput and IOPS.

As IT infrastructure becomes increasingly virtualized, effective storage has become a critical requirement. iXsystems' TrueNAS Storage appliances offer high-throughput, low-latency backing for popular virtualization programs such as Hyper-V, VMWare®, and Xen®. TrueNAS hybrid storage technology combines memory, NAND flash, and traditional hard disks to dramatically reduce the cost of operating a high performance storage infrastructure. Each TrueNAS appliance can also serve multiple types of clients simultaneously over both iSCSI and NFS, making TrueNAS a flexible solution for your enterprise needs.

For growing businesses that are consolidating infrastructure, the **TrueNAS Pro** is a powerful, flexible entry-level storage appliance. iXsystems also offers the **TrueNAS Enterprise**, which provides increased bandwidth, IOPS and storage capacity for resource-intensive applications.

Call **1-855-GREP-4-IX**, or go to **www.iXsystems.com**

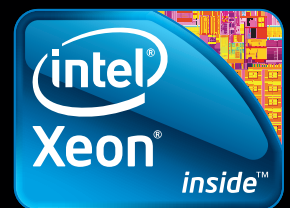
- ✓ Supports iSCSI and NFS exports simultaneously
- ✓ Compatible with popular Virtualization programs such as Hyper-V, VMware, and Xen
- ✓ 128-bit ZFS file system with up to triple parity software RAID

TrueNAS Pro Features

- One Six-Core Intel® Xeon® Processor 5600 Series
- High Performance Write Cache
- Up to 480GB MLC SSD Cache
- Up to 220 TB SATA Capacity
- Quad Gigabit Ethernet
- 48GB ECC Memory

TrueNAS Enterprise Features

- Two Six-Core Intel® Xeon® Processors 5600 Series
- Extreme Performance Write Cache
- Up to 1.2TB High Performance ioMemory
- Up to 500TB SATA or 320TB SAS Capacity
- Dual Ten Gigabit Ethernet
- 96GB ECC Memory





SHAWN POWERS

Digital Duct Tape

I've had enough system administration jobs to know that companies tend to take drastically different approaches to how they handle technology. Some companies budget extensively for their server infrastructures, and others have old workstations with box fans cooling them for servers. Whatever the server room looks like, things inevitably go wrong, and it's the job of the sysadmin to save the day. Sometimes that means a quick hack to get things going temporarily, and sometimes it means elaborate planning and scheduling for maintenance and replacement. That's the thing about system administration—you have to think on your feet and come up with solutions on the fly. It can be exciting, terrifying, stressful and rewarding, all at the same time.

This is our system administration issue, which always is one of my favorites. Rather than diving right into the sysadmin stuff though, Reuven M. Lerner starts things off with SQLAlchemy, which acts as a bridge for your Python objects to “talk” to an SQL database. It's a powerful

Python module, and if you're using Python with an SQL back end, you'll want to check it out. Dave Taylor, on the other hand, continues his series on creating a shell script to play *Cribbage*. Dave has a great way of tricking us all into learning things by using fun objectives. We certainly don't mind.

Remember when I said that Kyle Rankin got me started with Raspberry Pi hacking? This month he covers setting up the smallest collocated server you'll probably ever see. Kyle has a Raspberry Pi sitting in a data center rack in Austria, and he walks through preparing the little server for remote-only administration. Because the RPi lacks many of the features server-class machines usually have, a lot of planning goes into the preparation. Even if you don't plan to set up a Raspberry Pi server, it's a great article.

My column this issue hits much closer to home. If you have a fancy new Android tablet, but you're struggling to use it as much as you'd like, you're not alone. This month I tackle my Nexus 7. At first I struggled to do much more

than play *Angry Birds* with mine, but after a lot of effort, my tablet is a useful tool at work as well as a fancy toy. Those of you struggling to find your tablet's niche may benefit from my experiences.

Jeremiah Bowling addresses virtualization this month. Using ConVirt, he shows how to manage multiple virtualization architectures with a single tool. If you want to manage Xen and KVM side by side, it's worth checking out. In fact, with the paid version of the program, it's even possible to manage VMware hosts! Having the ability to look at different virtualization back ends with the same client makes comparing performance much easier.

System administrators like things to be easier, and so this month, Adrian Hannah teaches how to use Fabric, which is a tool for administering dozens of machines simultaneously. Whether you want to remove files from your entire server farm or install a package with its dependencies on a whole rack of servers, Fabric can make it a one-step process.

Andrew Fabbro and Aaron Peters both describe how to make Linux play well with others. They have drastically different takes on the subject, however. Andrew walks through the steps of getting Linux up and running in Microsoft's Azure cloud. Why would a person want to do that? Well, for the

same reason geeks do many things: because they can. Aaron, however, solves a problem we're all a little more familiar with, and that is how to connect your Android tablet with your Linux system. Although many tablets are unable to plug in to a system with USB for file access, there are many, many ways to connect with Android, and Aaron explores a bunch of them.

We tried to cover a wide variety of system administration topics this month, not just the traditional geek-in-the-server-room scenarios. As technology infiltrates every aspect of our lives, even those folks without the slightest desire to manage a data center must have at least rudimentary administration skills in order to function. For those hard-core sysadmins out there, if you're anything like me, your bag of tricks is like Mary Poppins' bag—there's always room for more. We hope this issue will be as useful for you as it has been enjoyable for us to create. For now, I need to leave—there's a server somewhere that needs to be turned off and turned back on....■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the [#linuxjournal](https://www.freenode.net) IRC channel on Freenode.net.

letters



Spaces in F77

Fred and Dave Taylor are both right (see the December 2012 issue's Letters section). The F77 standard—ANSI X3.9-1978

ISO 1539-1980 (E)—says 1–6 characters using a–z and 0–9; however, it also says that “Blanks are used to improve readability, but unless otherwise noted have no significance.” In other words, it is generally legal F77 to use spaces in names and keywords to decrease readability! Try `p r i n t * , 6 6` (yes, with spaces liberally interspersed)!
—dandeljx

Notebook Installation Articles

One topic that would be very helpful if it were discussed in an article is the installation procedure in modern notebooks. The new UEFI stuff and Windows 8 are making it very difficult to install a dual-boot on a notebook with Windows pre-installed. There are no clear or detailed solutions on the Web, so a good article explaining this would help us a lot.

Thanks, and keep up the good work!
—Toshiro

Great suggestion Toshiro, thanks! We'll see what we can do.—Ed.

Digital Version

I liked the print version but was forced to change. The NOOK 7" display was too small, and I didn't want to spend an outrageous amount of money for a 10" tablet. But, I found a cheap Android tablet from China on eBay, and it works great. I have changed all of my magazine subscriptions to digital. The only disadvantage is not being able to tear out the pages of interesting articles. Now I am waiting for an issue on how to hack it and convert it to a Linux distro of my liking.

—Jon GrosJean

Great to hear Jon. I find the PDF version a bit too small on my 7" tablet too. I might have to look for an inexpensive 10" model for the same reason. Thanks for the idea!—Ed.

About Fortran Variables

In the Letters section of the December 2012 issue, Dave Taylor and reader Fred comment on

FORTRAN and F77 variable names. Although what they write may have been correct for FORTRAN 77 or F77, the present Fortran standard (Fortran 2003) is much more flexible. The name of an entity may consist of between 1 and 31 alphanumeric characters (letters, underscores and numerals) of which the first must be a letter. For example, `time_of_flight` is a valid name. This is just one of the many enhancements that make the present Fortran standard a modern language. As a final comment, since the 95 standard, the official name is with a capital (F) followed by lowercase letters (ortran).

—Nuno Pinhao

Dave Taylor replies: *Indeed. I'll have to brush up my Algol-68 too, at this rate.*

Discouraging

I find it discouraging to have several great options for Linux in public school classrooms only to have it dashed by both sides of the aisles. I have long advocated for Linux to come into play in the States and help build repair facilities to facilitate incomes. More and more, I am finding

it increasingly stressful in Maine to watch as we spend millions repairing Windows and Macintosh systems along with iPad tablets. It is truthfully frustrating to advocate systems I've been using for the past six years—Ubuntu, OpenSUSE, VectorLinux and countless others—only to have them waived off as “not being user-friendly”. I'm not a coder. I'm a poet and a writer. So, if I can use these

Embedded Server

Standard SIB

- Fanless x86 1GHz CPU
- 1 GB DDR2 RAM On Board
- 4 GB Compact Flash Disk
- 10/100/1000 Base-T Ethernet
- Two RS-232 Ports
- Four USB 2.0 Ports
- Mini-PCIE
- Audio In / Out
- Power Supply Included
- Analog SVGA 3D Video
- VESA Hole Pattern
- Optional Wireless LAN
- Locked Compact Flash Access
- No Moving Parts
- XPE or Linux with Eclipse IDE
- Dimensions: 4.9" x 4.7" x 1.7" (125 x 120 x 44mm)




2.6 KERNEL

The EMAC Server-In-a-Box (SIB) is a low cost, small footprint, yet powerful server. Like all EMAC SIBs, the Standard SIB has no moving parts and features a rugged enclosure design making it an ideal choice for most industrial applications. The Standard SIB has a secure locking cover for securing the flash media, while still offering easy removal for updates and backing up the system.

http://www.emacinc.com/servers/Standard_sib.htm

Since 1985

OVER

28

YEARS OF

SINGLE BOARD

SOLUTIONS



EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

systems, a school can use these systems as well. They can assign different users, different groups and different networking settings. I've also been witnessing the ignorance of my own party it develops a draconian philosophy of removing computers completely.

It's the brush off that is the most stressful, however—to advocate, voice it and e-mail the governor of my state and be given blank responses. There is truthfully nothing that gets done as far as bringing Linux to the classroom. People look at the budgets, I look at the budgets, and we see a magic line of slashing budgets.

I say, instead of the magic line of slashing budgets, advocate for Linux companies to come to the state or area where you live. People say Linux is too hard to use or to use systems from the Windows 98-era. That is not how it works. What we are doing with Apple and Microsoft is hamstringing and confining parents, teachers and our state budgets to a massive monopoly. The game board is rigged, and ever since NVIDIA became a silent partner to Microsoft, the rules have been changing continually for

users on a budget. I want people to realize that they can speak up, that they can bring the change, and that they can bring jobs using Linux.

—**Joseph Ziehmer**

Joseph, as someone who has worked in education for almost 20 years now, I feel your pain. Thankfully, in my last position, I was able to use LTSP and Linux thin clients to save significant money while providing a user-friendly experience for our students. Sadly, that's the exception rather than the rule. I think as a community we need to continue touting the benefits while at the same time avoiding "trash talking" the opposition. I've found the negative campaign method seems to make people defensive and less likely to try Linux at all. Good luck, and keep fighting the good fight.—Ed.

Wunderlist and Wunderlist 2

I was looking at implementing it, but I see that Wunderlist2 does not have a native application for Linux, so that goes in the round can. It's funny that they can do it for iOS and Android, both of which descend from either BSD or Linux, but they cannot do a new one for Linux. Oh yes, they have

a Windows app too. If the Web app is so good, how come they need native apps for the other platforms?

—Chuck Hast

This seems to be how things go for me. I recently wrote about Wunderlist and its native Linux client, and then they release version 2 with no Linux client. My only hope is that the Linux version eventually will come out. As it is now, I have significant egg on my face.—Ed.

Backup Software Fully Cross-Platform

Regarding Doc Searls' article "Heavy Backup Weather" in the October 2012 issue, I've been using CrashPlan (<http://www.crashplan.com>) for the past three years, both for onsite and offsite backups. Aside from its Java requirement, it's been great.

—Gerry Normandin

Doc Searls replies: Sounds good. I'll give it a try.

Powerful: Rhino



Rhino M4700/M6700

- Dell Precision M4700/M6700 w/ Core i7 Quad (8 core)
- 15.6"-17.3" FHD LED w/ X@1920x1080
- NVidia Quadro K5000M
- 750 GB - 1 TB hard drive
- Up to 32 GB RAM (1866 MHz)
- DVD±RW or Blu-ray
- 802.11a/b/g/n
- Starts at \$1190
- E6230, E6330, E6430, E6530 also available

- High performance NVidia 3-D on an FHD RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven



Raven X230/X230 Tablet

- ThinkPad X230/X230 tablet by Lenovo
- 12.1" HD LED w/ X@1366x768
- 2.6-2.9 GHz Core i7
- Up to 16 GB RAM
- 750 GB hard drive / 180 GB SSD
- Pen/finger input to screen, rotation
- Starts at \$2050
- T430, T530, W530 also available

Rugged: Tarantula



Tarantula CF-31

- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA TouchScreen
- 2.4-2.53 GHz Core i5
- Up to 8 GB RAM
- 320-750 GB hard drive / 512 GB SSD
- CF-19, CF-52, CF-H2 also available

EmperorLinux
...where Linux & laptops converge

www.EmperorLinux.com
1-888-651-6686



Shawn Powers replies: *I completely agree! I don't even mind the Java-based front end, but I certainly wish the dæmon itself was running something other than Java. It's one of those programs that works so well, I tolerate Java.*

Advanced Articles

This is in response to Doug's letter in the January 2013 issue's Letters section titled "More-Advanced Articles".

First, I echo Doug's praise. *Linux Journal* keeps me up to date.

Second, perhaps instead of trying to balance your articles between beginner or novice-level articles and more-advanced articles in one magazine, you could have a second magazine. "Advanced Linux Journal" sounds good to me. I would pay for a subscription to a second magazine.

—harleypig

It's definitely something to consider. If the demand is high enough, perhaps it could happen someday!—Ed.

WRITE LJ A LETTER We love hearing from our readers. Please send us your comments and feedback via <http://www.linuxjournal.com/contact>.

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

LINUX JOURNAL

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: <http://www.linuxjournal.com/subs>. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE:

Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at <http://www.linuxjournal.com/digital>.

LETTERS TO THE EDITOR:

We welcome your letters and encourage you to submit them at <http://www.linuxjournal.com/contact> or mail them to *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Letters may be edited for space and clarity.

WRITING FOR US:

We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line: <http://www.linuxjournal.com/author>.

FREE e-NEWSLETTERS:

Linux Journal editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/emailsletters>.

ADVERTISING:

Linux Journal is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line: <http://www.linuxjournal.com/advertising>. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

1&1 DEDICATED SERVERS

1&1

YEARS

25

1&1 is celebrating its 25th anniversary.

Over the past 25 years 1&1 has grown to become one of the world's leading web hosts. Today, with 12 million customer contracts and 5000 employees 1&1 provides superior web hosting and server solutions to support your business. In celebration here is a gift from us to you.



Our data centers offer top security, Cisco firewall protection and maximum uptime.

- ✓ Unlimited traffic with no extra cost
- ✓ Parallels Plesk® Panel 11 for unlimited domains
- ✓ Exclusive to 1&1: Optional SUSE Linux Enterprise Server
- ✓ Mobile Server Monitoring for Android/iOS devices



1and1.com

Save \$360

Server XL 6

AMD Hexa-Core

6 Cores x 2,8 GHz
(3.3 GHz Turbo Core)

16 GB RAM DDR3 ECC

1,000 GB (2 x 1,000 SATA)

Software RAID 1

Free choice of CentOS, Debian, Ubuntu, or openSUSE.

Unlimited Traffic (100 Mbit/s)

~~\$129.99~~ **\$99.99** per month*

**OFFER ENDS
02/28/13**



* Offer valid for a limited time only. First year save \$360 off regular price of Dedicated Server XL 6. Other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are the property of their respective owners. © 2013 1&1 Internet. All rights reserved.

Intel, the Intel logo, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and/or other countries.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

The **udev** project appears to be in crisis. **Kay Sievers** has come under fire for failing to fix problems that have cropped up in the system, and it looks as though top kernel folks like **Al Viro**, not to mention **Linus Torvalds**, have been calling for someone else to take over the project.

The main issue is that user systems have been hanging. According to Kay, this is partly due to udev having a mysterious slowdown that he hasn't been able to fix yet. The slowdown results in certain driver requests being delayed until they time out, which apparently causes the appearance of a crash.

But Kay feels that the real problem is with the kernel's behavior, not with udev, and that the main kernel code should deal with it. Al and Linus (and the rest of the people complaining) argue that udev previously had been working, and that it was a patch to udev that resulted in the system crashes; therefore, udev either needed to fix the issue or revert

the patch.

This harkens back to the days when kernel folks blamed **GCC** for producing bad machine code, while the GCC folks blamed the kernel for using bad C code. One key difference is that unlike GCC, the udev code is actually part of the kernel and isn't an independent project.

It seems clear that if Kay can't fix the problem, or at least adopt better development practices, someone else will be asked to maintain udev. **Greg Kroah-Hartman**, one of the original udev authors, would be an obvious candidate, at least for the short term. But, he's pretty busy these days doing tons of other kernel work.

Recently, Linus Torvalds decided to simplify the **cryptographic signature code** for kernel modules. His initial motivation was to speed things up by migrating some of the time-consuming signing issues from compile time to install time where they would end up being faster.

This turned out to be slightly controversial. **David Howells** suggested that Linus should go even further and take out all the module-signing code and just let users do it manually. But, this ended up causing some unexpected blowback from Linus.

The issue Linus is concerned with is the ordinary user who wants to protect the system from root kits and other attacks. Requiring modules to be signed by a secure key is a good way to address that. But, he felt that David was concerned with allowing distribution vendors to keep a cryptographic stranglehold over what kind of software ordinary users could run on their systems.

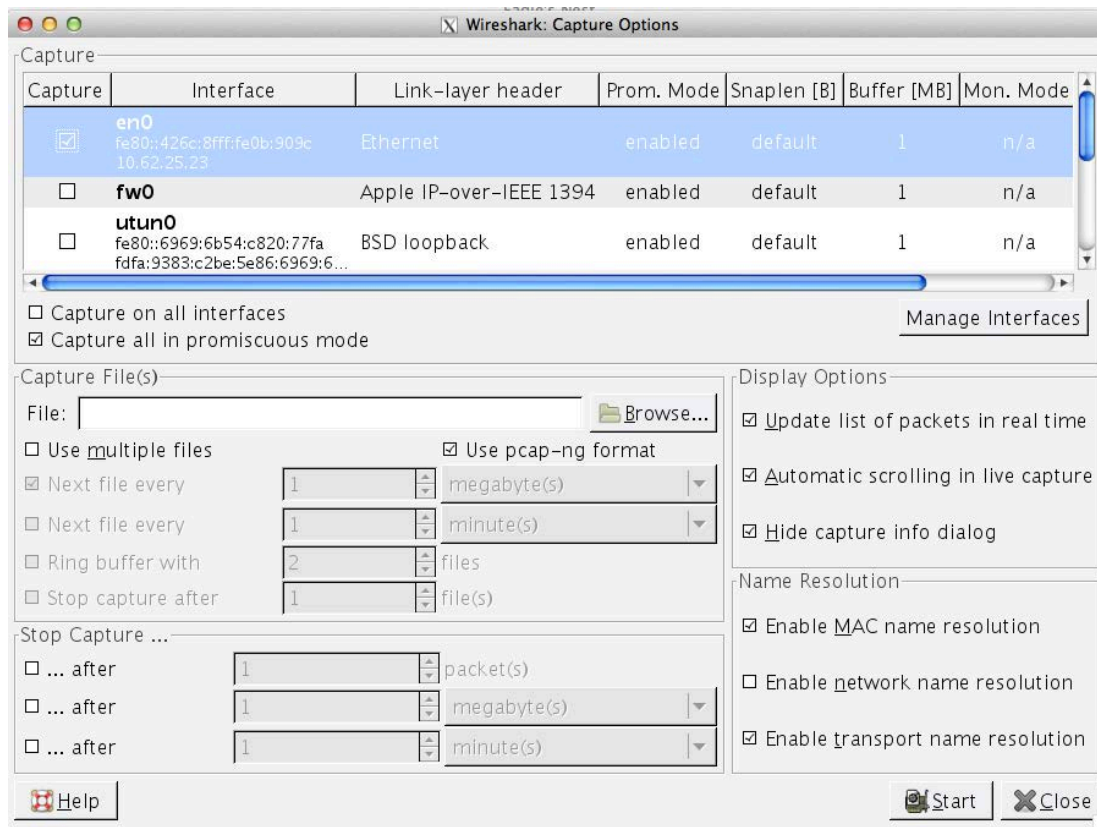
There was a brief attempt recently to change the way **“signed-off-by” reviews** are submitted. Typically, whenever a patch gets sent into the kernel, it passes through a gauntlet of reviewers who confirm that the patch looks good, contains no proprietary code and so on. But, Al Viro pointed out that in a lot of cases, reviews show up in the mailing list, after the patch already has been accepted into the kernel.

In that case, the sign-off doesn't get included. Al felt this was lost data, and he suggested changing the process, so that sign-offs could be added after the fact.

There actually was quite a bit of support for this idea, and it turned out that the latest versions of **git** already support it, via the `git notes add` command. But, although Linus Torvalds is fine with people using that sort of thing for local development, he said he wouldn't include after-the-fact sign-offs in the main tree. He just felt it wasn't that important. As long as someone signs off on the code, especially the author of the given patch, he's fine with not having the maximum number of sign-offs that he could get.

Considering that the signed-off-by process was created in direct response to the **SCO lawsuits** (http://en.wikipedia.org/wiki/SCO%E2%80%93Linux_controversies), he must be pretty confident that it's not an important issue. I believe at the time Linus was particularly inconvenienced, having to account for the origins and licenses of many kernel patches.—**ZACK BROWN**

Non-Linux FOSS: Dive Deep with Wireshark



Before you say anything, yes, I know Wireshark is available for Linux. This time, however, Windows and OS X users get to play too. Wireshark is an open-source network analysis tool that is really amazing for troubleshooting a network.

Running Wireshark on OS X does require an X11 server (see my Non-Linux FOSS article in the December 2012 issue of *LJ* on XQuartz). It also looks a bit dated once it's up and running, but rest assured, the latest version is functioning behind the scenes. If you're thinking

this program looks a lot like Ethereal, you're absolutely correct. It's the same program, but six or so years ago the name changed.

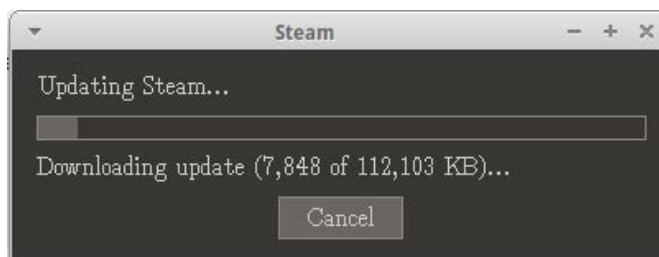
Wireshark is strictly a wired-ethernet inspection tool, but if you're trying to solve a network issue, it's the de facto standard tool. It's not a new tool by any means, but if you're on a foreign operating system (that is, not Linux), it's nice to know some old standbys are available. Check it out today at <http://www.wireshark.org>.

—SHAWN POWERS

It's Getting Steamy in Here!

After months of me promising Steam would be coming to Linux, the public beta is finally here. The early verdict: it's pretty great! The installer is a simple pre-packaged .deb file for Ubuntu (or Xubuntu in my case), and the user portion of the install looks very much like Windows or Macintosh. In my limited testing, I've found the Steam beta to be at least as stable as Desura. I also was impressed with the large number of my Steam games that have Linux versions ready to download and play.

If you were under the impression that Steam was going to be the next



Duke Nukem Forever, I'm happy to say that you (and I) were wrong. Steam is finally coming to Linux, which has the potential to change the way Linux users play games. It also means fewer reboots into Windows just to shoot a few zombies! Check it out at <http://www.steamforlinux.com>.

—SHAWN POWERS



System Administration Poll

System Administration is one of the most popular topics at LinuxJournal.com, and many of our readers have loads of experience in the field. We recently polled our on-line readers about their system administration habits, and we received some interesting answers, as usual.

We were surprised to learn that an almost equal number of you use a GUI or Web-based tool versus the command line, with 51% using the latter. And, on the command line, your preferred protocol is SSH by a wide margin with 87%. Telnet and remote serial console each received 6%, with 1% of you using something else entirely. 45% of you manage one server, while 15% manage more than 20, and more than a few of you are employed by hosting companies or companies with similar needs, so those numbers get pretty high.

We were not very surprised to learn that vim was your favorite command-line text editor by far, with 74% of the votes, compared to nano/pico with 14% and emacs with 8%. The remaining 4% of you use something else, and among the other options was naturally "all of the above".

61% of you are mostly running Ubuntu or Debian-based servers,

and Red Hat is your second favorite (24%), while 7% are running Windows. The other 8% of you are running a variety of other operating systems including other flavors of Linux, Solaris, AIX or FreeBSD.

Security updates are a regular and necessary process, and 43% of you do them at least annually, while 12% apply security updates daily. We're relieved to know so many of you are on top of things. Non-security updates are also frequent with the majority of readers updating at least quarterly.

The full survey results are listed below for your perusal. Thanks again for always being willing to share with the class!

1) Do you do the majority of your system administration work from:

- the command line: 51%
- a GUI/Web-based tool: 49%

2) When accessing your servers via command line, do you use:

- SSH: 87%
- Telnet: 6%
- remote serial console: 6%
- other: 1%

3) How many servers do you manage?

- 1: 45%
- 2-5: 20%

- 6–10: 10%
- 11–20: 10%
- more than 20: 15%

4) Which command-line text editor is best?

- vim: 74%
- nano/pico: 14%
- emacs: 8%
- other: 4%

5) Do you use a configuration management tool like puppet?

- yes: 16%
- no: 84%

6) Are most of your servers:

- Ubuntu-/Debian-based: 61%
- Red Hat-based: 24%
- Windows: 7%
- other: 8%

7) How often do you apply security updates to your systems?

- daily: 12%
- weekly: 21%
- monthly: 15%
- quarterly: 9%
- annually: 43%

8) How often do you apply non-security updates to your system?

- daily: 7%
- weekly: 18%
- monthly: 17%
- quarterly: 12%

- annually: 46%

9) Have you ever delayed a kernel update in order to preserve your coveted uptime?

- yes: 30%
- no: 70%

10) Do you work on your server farm from home?

- yes: 44%
- no: 56%

11) If so, do you use a VPN?

- yes: 65%
- no: 35%

12) Does your server infrastructure include a DMZ?

- yes: 52%
- no: 48%

13) What percentage of your servers are virtualized?

- 0–25%: 43%
- 26–50%: 20%
- 51–75%: 17%
- 76–100%: 20%

14) If you use virtualization, what is your host environment?

- VMware: 42%
- Xen: 13%
- KVM: 18%
- Hyper-V: 3%
- n/a: 12%
- other: 12%

[UPFRONT]

15) Do you host e-mail:

- locally: 55%
- with a cloud host: 19%
- we don't provide e-mail: 26%

16) Do you allow users VPN access into your network?

- yes: 54%
- no: 46%

17) Do you have Wi-Fi coverage at your workplace?

- yes: 84%
- no: 16%

18) If yes, do you allow guest access to Wi-Fi?

- yes: 40%
- no: 49%
- n/a: 11%

19) Is your network and server layout well-documented?

- yes: 57%
- no: 43%

20) Are you the lone system administrator at your workplace?

- yes: 46%
- no: 54%

21) Do you have to support platforms other than Linux?

- yes: 71%
- no: 29%

22) Have you ever had a system compromised?

- yes: 37%
- no: 63%

23) Do you use:

- a router/firewall appliance (Cisco, etc.): 62%
- a software-based router/firewall solution: 38%

24) Does your husband/wife/significant other know your password(s)?

- yes: 7%
- no: 93%

25) Do you use a password program like LastPass or KeePassX?

- yes: 37%
- no: 63%

26) How often do you change your passwords?

- daily: 1%
- weekly: 3%
- monthly: 19%
- quarterly: 31%
- rarely: 46%

27) Do you force your users to change their passwords?

- yes: 50%
- no: 50%

—**KATHERINE DRUCKMAN**
and **SHAWN POWERS**

LINUX JOURNAL ARCHIVE DVD



NOW AVAILABLE

Save \$10.00 by using discount code DVDFEB at checkout.

Coupon code expires 2/16/2013

www.linuxjournal.com/dvd

Handling R Packages

One of the R statistics program's great features is its modular nature. As people develop new functionality, R is designed so that it's relatively easy to package up the new functionality and share it with other R users. In fact, there is an entire repository of such packages, offering all sorts of goodies for your statistical needs. In this article, I look at how to find out what libraries already are installed, how to install new ones

and how to keep them up to date. Then, I finish with a quick look at how to create your own.

The first step is to check and see what libraries already are installed on your system (Figure 1). You can do this by running `library()` from within R. This provides a list of all the libraries installed in the various locations visible to R. If you find the library you're interested in, your work is almost done.

```

File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x
Packages in library '/usr/lib/R/site-library':
colorspace      Color Space Manipulation
date            Functions for handling dates
doSNOW         Foreach parallel adaptor for the snow package
effects        Effect Displays for Linear, Generalized Linear,
               Multinomial-Logit, Proportional-Odds Logit
               Models and Mixed-Effects Models
foreach        Foreach looping construct for R
iterators      Iterator construct for R
multicore      Parallel processing of R code on machines with
               multiple cores or CPUs
scatterplot3d  3D Scatter Plot

Packages in library '/usr/lib/R/library':
base           The R Base Package
boot          Bootstrap Functions (originally by Angelo Canty
               for S)
class         Functions for Classification
cluster       Cluster Analysis Extended Rousseeuw et al.
codetools     Code Analysis Tools for R
compiler      The R Compiler Package
:

```

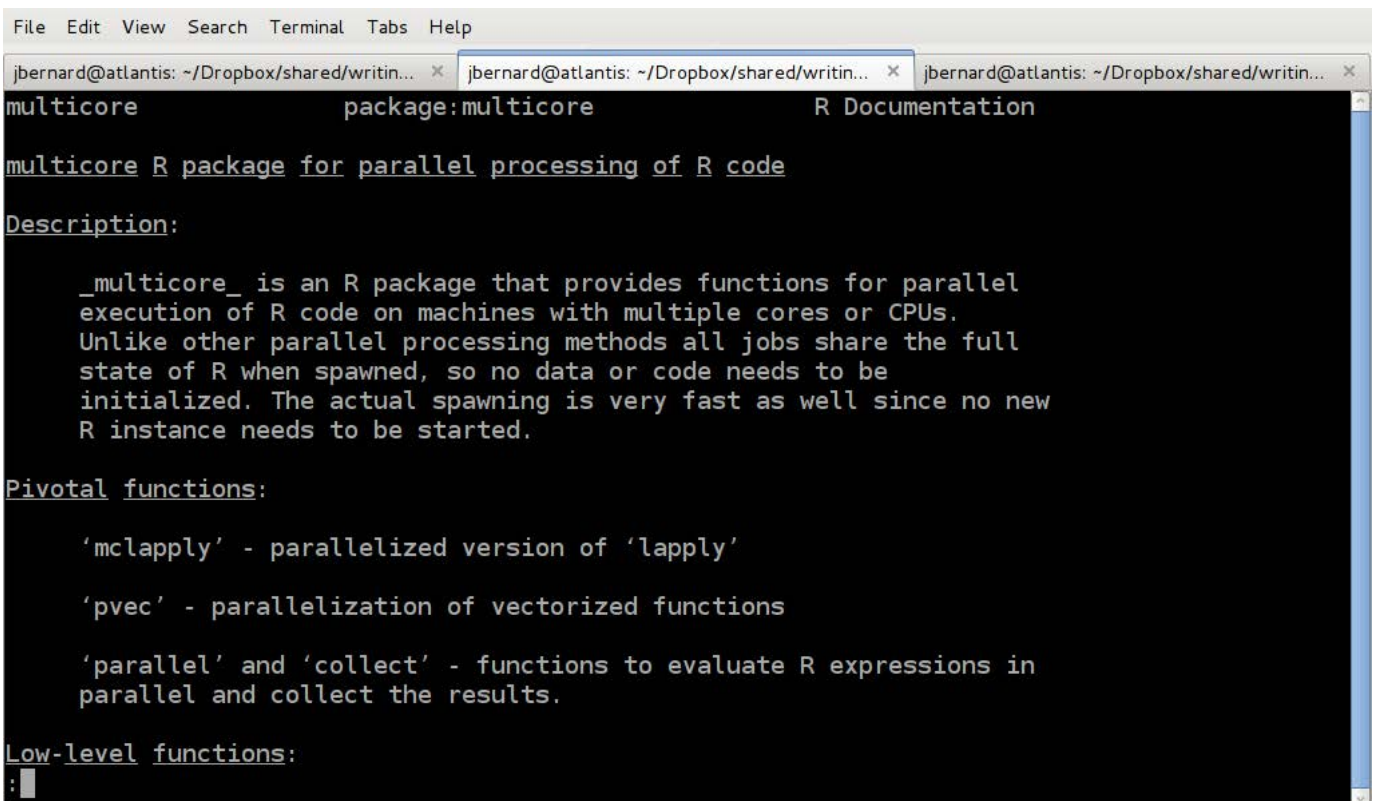
Figure 1. The `library()` command gives a list of currently installed libraries.

In order to make R load the library of interest into your workspace, you need to call `library` with the name of the library in brackets. Let's say you want to do parallel code with the multicore library. You would call `library("multicore")`.

If you want to learn more about a library, R includes a help system that is modeled after the man page system used in Linux. There are two ways to access it. The first is to use the `help()` command. So in this case, you would run `help("multicore")` (Figure 2). The shortest way to get help

is to use the special character `?`. For example, you could type `?multicore` to get the same result. A related command that is good to know is `??`. It does a search through the library names and descriptions based on the text given. For example, `??plot` pulls up entries related to the word plot (Figure 3).

But, what if the library you are interested in isn't already on your system? Then you need to install it somehow. Luckily, R has a full package management system built in. Installing a package is as easy



```
File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~/Dropbox/shared/writin... x jbernard@atlantis: ~/Dropbox/shared/writin... x jbernard@atlantis: ~/Dropbox/shared/writin... x
multicore package:multicore R Documentation
multicore R package for parallel processing of R code
Description:
  _multicore_ is an R package that provides functions for parallel
  execution of R code on machines with multiple cores or CPUs.
  Unlike other parallel processing methods all jobs share the full
  state of R when spawned, so no data or code needs to be
  initialized. The actual spawning is very fast as well since no new
  R instance needs to be started.
Pivotal functions:
  'mclapply' - parallelized version of 'lapply'
  'pvec' - parallelization of vectorized functions
  'parallel' and 'collect' - functions to evaluate R expressions in
  parallel and collect the results.
Low-level functions:
:█
```

Figure 2. Getting Help on a Library

as running `install.packages()`, where you hand in a list of package names. But, how do you know what packages are available for installation? The R project has a full repository of packages ready for you to use. You can find them at <http://cran.r-project.org>. On the left-hand menu, you will see an entry called "Packages", which will bring you to list of packages. You can search alphabetically by name or by category.

Say you're interested in doing linear programming. On CRAN, you will find the `linprog` package, which you can install with `install.packages("linprog")`. When you first run this command, it should come back with an error (Figure 4). By default, R tries to install packages into the system library location. But, unless you are running as root (and you aren't doing that, right?), you won't have the proper permissions to do so. Therefore, R will ask if you want to install the new package into a personal library storage location in your home directory. After you agree to this, it will go ahead and try to download the source for this package. If this is the first time you have installed a package, R will ask you to select

a CRAN mirror for downloading the package. This mirror will be used for all future downloads. By default, R also will download and install any dependencies the requested package needs. So in this sense, it really is a proper package management system.

For many packages, all that is involved is strictly R code. But in some cases, the author may have written part of the code in some other language, like C or FORTRAN, and wrapped it in R code. In those types of packages, the other code needs to be compiled into binary code before it can be used. How can you do that? Well, the R package system actually can handle compiling external code as part of the installation process. In some cases, this external code may need other third-party libraries in order to be compiled. To hand in locations for those, you need to add some options to the `install.packages` function call. Checking the help page (with `?install.packages`) shows that you can include installation options as `INSTALL_opts`.

Now that you have your collection of packages all installed and configured on your system, what

```

jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs
File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install
installed.packages install.packages
> install.packages
install.packages
> up
update update.formula update.packageStatus upper.tri
update.default update.packages upgrade
> ?upgrade
> pac
pacf packageEvent packageSlot packageStatus packBits
package.contents packageHasNamespace packageSlot<- packageVersion
packageDescription package.skeleton packageStartupMessage package_version
> packageStatus()
-- Please select a CRAN mirror for use in this session ---
Loading Tcl/Tk interface ... done
Number of installed packages:

                ok upgrade unavailable
/home/jbernard/R/x86_64-pc-linux-gnu-library/2.15  0      0      0
/usr/local/lib/R/site-library                    0      0      0
/usr/lib/R/site-library                          5      3      0
/usr/lib/R/library                               17     12      0

Number of available packages (each package counted only once):

                installed not installed
http://mirror.its.dal.ca/cran/src/contrib        23     4137
> █

```

Figure 5. Checking Whether Any Packages Have Updates Available

do you do if a bug gets fixed in one of them? Or, what happens if a new version comes out with a better algorithm? Well, R's package management system can handle this rather well. You can check to see whether any packages need to be updated by running `packageStatus()` (Figure 5). If you see that updates are available, you can install the updates by using the command `update.packages()`.

This command goes through each available update and asks you whether you want to install the new version.

Many packages include either demos, data files or both. The demos walk you through some examples of how to use the functions provided by the package in question. To see what demos are available, you can call `demo()` (Figure 6). To run a particular demo, for example, the `nlm`

```

jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs
File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_libs x
Demos in package 'base':
error.catching      More examples on catching and handling errors
is.things           Explore some properties of R objects and
                    is.FOO() functions. Not for newbies!
recursion           Using recursion for adaptive integration
scoping             An illustration of lexical scoping.

Demos in package 'graphics':
Hershey             Tables of the characters in the Hershey vector
                    fonts
Japanese            Tables of the Japanese characters in the
                    Hershey vector fonts
graphics            A show of some of R's graphics capabilities
image              The image-like graphics builtins of R
persp              Extended persp() examples
plotmath           Examples of the use of mathematics annotation

Demos in package 'stats':
glm.vr             Some glm() examples from V&R with several
                    predictors
lm.glm            Some linear and generalized linear modelling
                    examples from `An Introduction to Statistical
                    Modelling' by Annette Dobson
nlm               Nonlinear least-squares using nlm()
smooth            `Visualize' steps in Tukey's smoothers

Use 'demo(package = ,packages(all.available = TRUE))'
to list the demos in all *available* packages.

(END)

```

Figure 6. The List of Demos Available in R

demo, you would run `demo(nlm)`.

Many packages also include sample data files that you can use when you are learning to use the new functions. To see what data files are available, you would call `data()` (Figure 7). To load a particular data file, you need to call `data` with the data file you are interested in. For example, if you want to play with water levels in Lake Huron, you would call `data(LakeHuron)`. You

can get more information on the data, including a description and a list of the variables available, by running `?LakeHuron` (Figure 8).

So far, I've been looking at dealing with individual packages, but sometimes you need functions provided by several different packages. In R parlance, this is called task views. These are groups of packages that are all useful for a particular area of research. If you are

```

jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_Libs
File Edit View Search Terminal Tabs Help
jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_Libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_Libs x jbernard@atlantis: ~/Dropbox/shared/writings/lj/science/R_Libs x
Data sets in package 'datasets':
AirPassengers      Monthly Airline Passenger Numbers 1949-1960
BJsales            Sales Data with Leading Indicator
BJsales.lead (BJsales) Sales Data with Leading Indicator
BOD                Biochemical Oxygen Demand
CO2                Carbon Dioxide Uptake in Grass Plants
ChickWeight        Weight versus age of chicks on different diets
DNase              Elisa assay of DNase
EuStockMarkets     Daily Closing Prices of Major European Stock
Indices, 1991-1998
Formaldehyde        Determination of Formaldehyde
HairEyeColor        Hair and Eye Color of Statistics Students
Harman23.cor        Harman Example 2.3
Harman74.cor        Harman Example 7.4
Indometh            Pharmacokinetics of Indomethacin
InsectSprays        Effectiveness of Insect Sprays
JohnsonJohnson    Quarterly Earnings per Johnson & Johnson Share
LakeHuron           Level of Lake Huron 1875-1972
LifeCycleSavings    Intercountry Life-Cycle Savings Data
Loblolly            Growth of Loblolly pine trees
Nile                Flow of the River Nile
Orange              Growth of Orange Trees
OrchardSprays        Potency of Orchard Sprays
PlantGrowth         Results from an Experiment on Plant Growth
Puromycin           Reaction Velocity of an Enzymatic Reaction
Seatbelts           Road Casualties in Great Britain 1969-84
Theoph              Pharmacokinetics of Theophylline
Titanic             Survival of passengers on the Titanic
ToothGrowth         The Effect of Vitamin C on Tooth Growth in
Guinea Pigs
UCBAdmissions       Student Admissions at UC Berkeley
UKDriverDeaths      Road Casualties in Great Britain 1969-84
UKgas               UK Quarterly Gas Consumption
USAccDeaths         Accidental Deaths in the US 1973-1978
USArrests           Violent Crime Rates by US State
USJudgeRatings      Lawyers' Ratings of State Judges in the US
Superior Court
USPersonalExpenditure Personal Expenditure Data
VADeaths            Death Rates in Virginia (1940)
WWWusage            Internet Usage per Minute
:

```

Figure 7. The List of Data Files Available in R

interested in using task views, start by installing the `ctv` package. In R, run `install.packages("ctv")` to install the main task view package.

Once that's done, you can load the library with `library("ctv")`. Now, you will have new functions included in the install and update packages. To install a view, like the Graphics view, you simply can run `install.views("Graphics")`. You can update these views as a

whole with the `update.views()` command. These task views, like all of the packages, are written and maintained by other users like yourself. So, if you have some area of research that isn't being served right now, you can step in and organize a new view yourself.

Up to this point, I've been discussing how to use packages that have been written and provided by other people. But, if you are doing

Once you have all of your code and data files written and packaged, you can go ahead and run a check on your new package by running the command `R CMD check /path/to/package` on the command line. This runs through some standard checks to make sure everything is where R expects things. Once your package passes the checks, you can run `R CMD build /path/to/package` to see if R can build your package properly. This is especially important if you have external code in another programming language. Once your package passes the checks and builds correctly, you can bundle it up as a tarball and send it up to <http://CRAN.R-project.org/incoming> as anonymous, and then send an e-mail to `CRAN@R-project.org`. Once your package has been checked by someone at CRAN to verify that it builds correctly, your newly created package will be added to the repository. Fame and fortune will be soon to follow.

Hopefully this article has provided enough information to help you get even more work done in R. And remember, we all progress when we share, so don't hesitate to add to the functionality available to the whole community.

—**JOEY BERNARD**

They Said It

It does not do to leave a live dragon out of your calculations, if you live near him.—*J. R. R. Tolkien, The Hobbit*

A goal without a plan is just a wish.—*Antoine de Saint-Exupéry*

In preparing for battle I have always found that plans are useless, but planning is indispensable.—*Dwight D. Eisenhower*

Someone's sitting in the shade today because someone planted a tree a long time ago.—*Warren Buffett*

Everybody has a plan until they get punched in the face.—*Mike Tyson*

20
13

ATTEND FLOURISH!
INSPIRING TALKS,
ILLUMINATING WORKSHOPS

OPEN SOURCE, OPEN FUTURE.

MARCH

1-2

UIC FORUM
725 W ROOSEVELT RD.
(MC 126)
CHICAGO, IL



FLOURISH!

A CONFERENCE PROMOTING
THE ADOPTION OF FREE,
LIBRE, OPEN SOURCE
SOFTWARE IN THE MIDWEST.

sponsored by:

github
SOCIAL CODING

GET UPDATES AT [HTTP://FLOURISHCONF.COM/](http://flourishconf.com/)

Android Candy: Plex

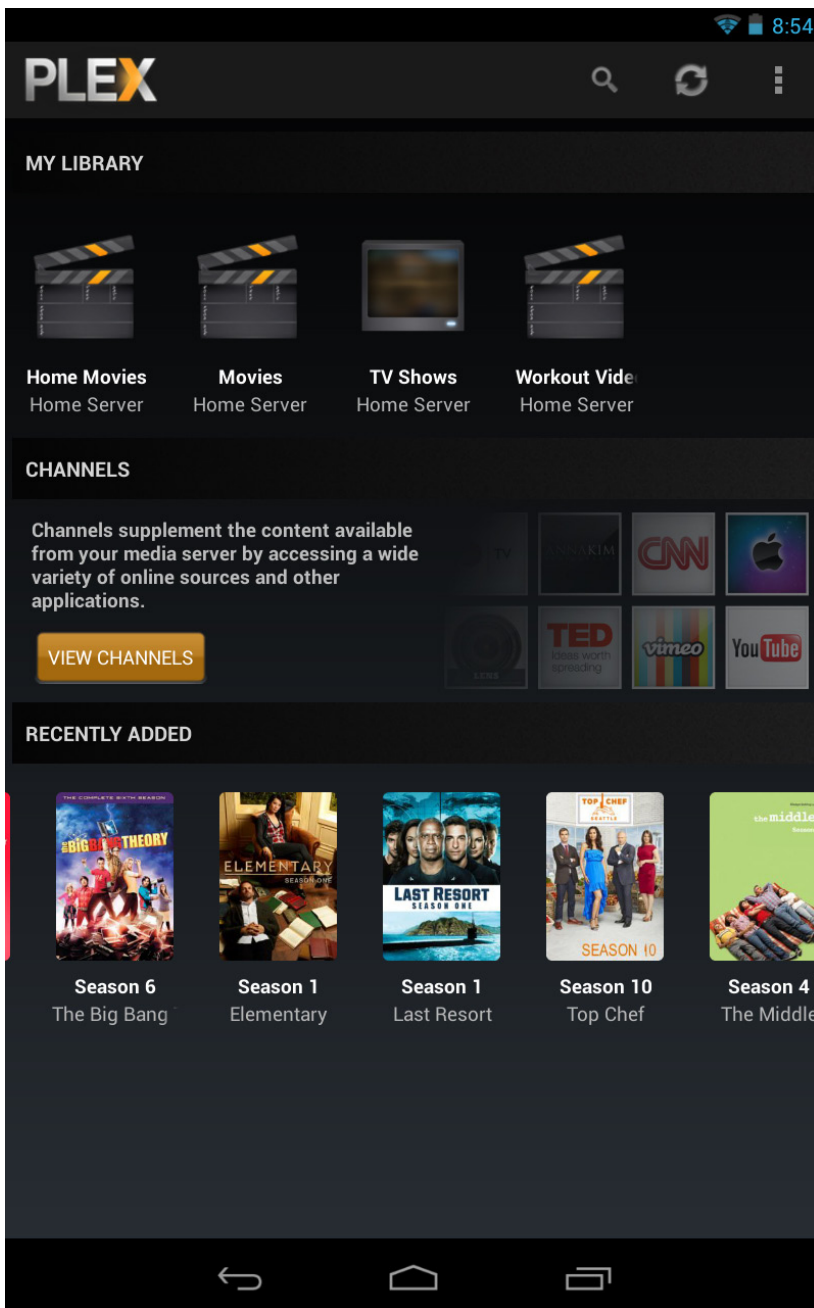


Figure 1. Plex shows your home video collection in much the same way as Hulu or Netflix.

Anyone with an iPhone probably is familiar with the AirVideo application. Basically, it's the combination of a server app that runs on your Windows or OS X machine, and it serves video over the network to an AirVideo application on your phone. It's extremely popular, and for a good reason—it works amazingly well.

For a long time, there wasn't a good solution for the Android world, largely due to the way Android streamed video. Now, however, there is an incredible application for doing the exact same thing iOS users do with AirVideo. You've probably heard of Plex, but you may not know about the server/client combination it can do with Android.

Once you install the server application,

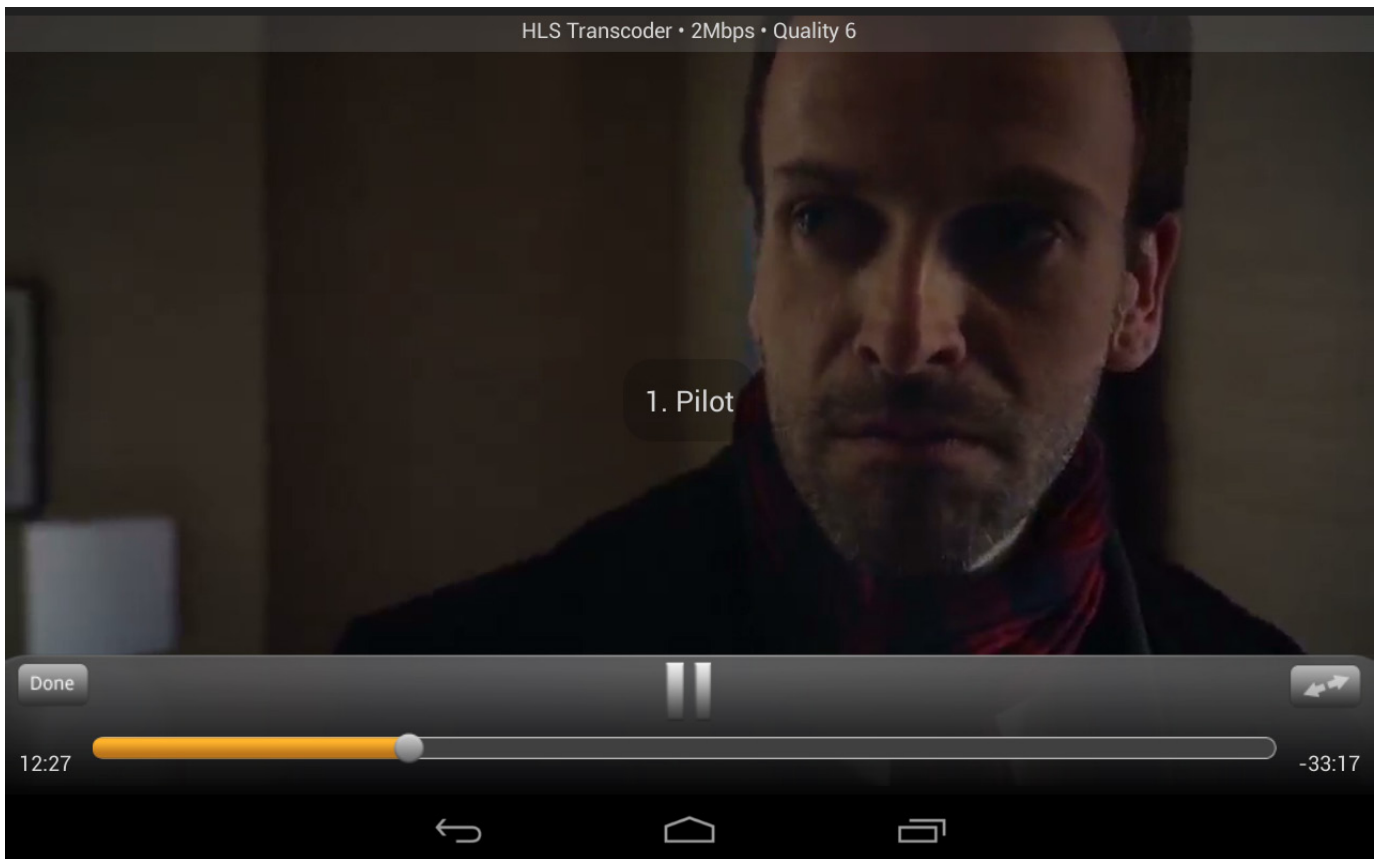


Figure 2. The video quality adjusts for your current bandwidth and renders crisp video even on a large tablet display.

which runs perfectly fine on a Linux server, you install the Plex application from the Google Play store, and your video collection follows you anywhere you have connectivity. The content is, of course, dependent on the content you have on your server, but the format in which your content is stored doesn't matter very much. Plex's server application does a great job of streaming most video

formats and converting to an appropriate bandwidth on the fly.

Plex may have started out as a Macintosh-compatible competitor to XBMC, but it's evolved into an incredible video-streaming system. With Plex, you can become your own Netflix! Due to its Linux compatibility and incredible video streaming ability, Plex is this month's Editors' Choice!

—SHAWN POWERS

SQLAlchemy



REUVEN M.
LERNER

Talk to your SQL database via Python objects with SQLAlchemy.

Although it sometimes might seem as if relational databases have gone the way of the dinosaur, making way for non-relational (NoSQL) databases, such as MongoDB and Cassandra, a very large number of systems still depend on a relational database. And, although there is no requirement that a relational database use SQL as its query language, it's a rare database product that does not do so.

The good news is that SQL is relatively easy to work with, particularly when the queries are straightforward. It's fast and easy to create tables, insert data into them, update that data and write queries that retrieve some or all parts of the data. SQL also makes it fairly easy to combine ("join") information from multiple tables, letting you normalize the data, while keeping speed and flexibility at a maximum.

SQL might not be difficult to work with on its own, but you rarely work with it in a vacuum. Usually, your SQL statements reside within a

program you have written. The SQL is kept as a text string within the application and is then sent, via a network socket, to the server.

There are several problems with this. First, it means you have to mix two different languages within the same program. Inside your Web application, which you've worked hard to write, and which you try to ensure is maintainable, you have code in a totally separate language, inside strings, which you cannot test or maintain directly.

Even if the SQL queries weren't written inside strings, you still would be faced with the fact that the majority of your Web application is written in one language, but your data-manipulation routines are written in another language. A Web application contains, no matter how you slice it, components in HTML, CSS and JavaScript, as well as whatever server-side language you're using. Adding SQL to this can only complicate things further.

Even if SQL and a typical

server-side language were on equal footing in terms of the syntax of a Web application, there's a fundamental mismatch between the ways in which they handle data. SQL operates with rows and columns within tables; everything in a relational database has to fit into this table-centric view of the world. By contrast, modern programming languages have a rich variety of data structures and typically are object-oriented to some degree or another.

Libraries that bridge the gap between procedural code and SQL are known as object-relational mappers, or ORMs. ORMs typically represent database records as instances of a particular class. In order to represent 50 records, you would need 50 instances of a class, with the state of each instance reflecting the names, types and values of the columns in that record.

There are two basic paradigms for passing data between the object-oriented data structures and the database, both of which were described by Martin Fowler. In the first paradigm, known as Active Record, each instance is tied directly to a row in the database, and the class itself (as well as each object)

is responsible for ensuring that the data is saved to the database. In other words, Active Record requires that you create a single class, and that it handles both sides of the object-relational divide. The Active Record class in Ruby on Rails is (not surprisingly) an implementation of this paradigm and provides a great deal of power and flexibility.

A second paradigm is known as Data Mapper, and it requires the use of three different object classes: a class that represents the data itself at the object level, a class that represents the database table and a "mapper" object that acts as a go-between, ensuring that the object and relational parts of the system are appropriately synchronized.

An excellent and popular example of the Data Mapper paradigm can be found in the SQLAlchemy project. SQLAlchemy has been around for a number of years already, and makes it possible to work with relational databases flexibly from within your Python program, without having to write any SQL.

In this article, I take a look at SQLAlchemy, exploring a number of its options and features, and considering how it can be used in Web and other applications.

Connecting to a Database

Installing SQLAlchemy should be straightforward to anyone who has installed Python packages before. You can get SQLAlchemy from PyPi, the Python Package Index, either by downloading it from <http://pypi.python.org> or by using the `easy_install` or `pip` programs to retrieve and install it. I was able to install it with:

```
pip install sqlalchemy
```

You might need to install SQLAlchemy as root. Or, you can install it into a virtual machine, using the popular `virtualenv` package for Python, which gives you nonroot control and permissions over a Python environment.

You also will need to install a driver for the database you intend to use. My favorite relational database is PostgreSQL, and I use the `psycopg` Python driver, also available on PyPi and (by extension) via `pip`.

I should note that although I know SQLAlchemy works with Python 3, much of the work I do nowadays is still in Python 2, mostly because that's what my clients are using. My examples, thus, also will be in Python 2, although I believe they will work in Python 3 with little or no changes.

Let's assume you have a database

table, `People`:

```
CREATE TABLE People (id SERIAL PRIMARY KEY,
                      first_name TEXT,
                      last_name TEXT,
                      email TEXT,
                      birthday DATE);
```

Let's also add some initial records:

```
INSERT INTO People (first_name, last_name, email, birthday)
VALUES ('Reuven', 'Lerner', 'reuven@lerner.co.il', '1970-jul-14'),
       ('Foo', 'Bar', 'foobar@example.com', '1970-jan-1');
```

In order to access this table using SQLAlchemy's ORM, you first need to create a database session object, which itself must be created using an "engine". Each database driver has its own style of URL. In the case of PostgreSQL accessed via the `psycopg2` driver, you would use something like this:

```
dburl = 'postgresql+psycopg2://reuven:reuven@localhost/atf'
```

This URL indicates not only the database and driver type, but also my user name and password ("reuven" in both cases), the hostname (`localhost`) and the name of the database I'll be accessing ("atf"). If the database is not available at the default PostgreSQL port of 5432, you can specify that as well in the URL.

You then tell Python to create a new engine based on this URL:

```
from sqlalchemy import create_engine
engine = create_engine(dburl)
```

Now that you have the engine defined, you can create a session based on this engine. Doing so requires two steps: first you create a new, custom Session class for this engine, and then you create an instance of the Session class that you will use to access the database:

```
from sqlalchemy.orm import sessionmaker # import sessionmaker class
Session = sessionmaker(bind=engine)    # make custom session type
session = Session()                    # make instance of session
```

You're now connected to the database! But, that's not quite enough. If you want to map your database table to one or more Python objects, you need to define a class. You do this by defining a normal Python class, with a few subtle changes:

- The class must inherit from Base, a class returned from the declarative_base function provided by SQLAlchemy.
- The database columns must be defined as class attributes, as instances

of the SQLAlchemy-provided Column class.

- You connect the class with your database table by defining the `__tablename__` class-level attribute.

For example, the following Python class provides a mapping to the People database table:

```
from sqlalchemy import Column, Integer, String, DateTime
from sqlalchemy.ext.declarative import declarative_base
Base = declarative_base()

class Person(Base):

    __tablename__ = 'people'
    id = Column(Integer, primary_key=True)
    first_name = Column(String)
    last_name = Column(String)
    email = Column(String)
    birthday = Column(DateTime)

    def __init__(self, firstname, lastname, email, birthday):
        self.first_name = firstname
        self.last_name = lastname
        self.email = email
```

It might not be obvious at first glance, but this class implements the Data Mapper design pattern. The class attributes that you have defined describe the columns in the database table and can contain a great deal of detail, including indexes, uniqueness

requirements and even custom integrity constraints, such as those provided by PostgreSQL. The class itself is a standard Python class.

But behind the scenes, because this class inherits from `Base`, you get a number of other attributes, including `__mapper__`, which indicates how your Python class will be mapped to the database table. You can see this in an interactive Python shell by asking to see the printed representation of `Person.__mapper__`:

```
Person.__mapper__
<Mapper at 0x10af5ef90; Person>
```

You now have a session that connects to the database and a table in the database that has been described in Python. You now can execute a query against your table:

```
for p in session.query(Person):
    print p.first_name
```

That gives the following:

```
Reuven
Foo
```

In other words, `session.query` is executing a query against the database, without you having to specify the SQL. You also can restrict

the records you'll get, by chaining the `filter_by` method to your query:

```
for p in session.query(Person).filter_by(id=1):
    print p.first_name
```

That gives the following:

```
Reuven
```

Note that the `filter_by` method is not acting on the results of `session.query`. Rather, it is changing the SQL that eventually is sent to the database. You can see this by assigning printing to the query object without executing it or putting it in an iteration context:

```
print session.query(Person).filter_by(id=1)

SELECT people.id AS people_id,
       people.first_name AS people_first_name,
       people.last_name AS people_last_name,
       people.email AS people_email,
       people.birthday AS people_birthday
FROM people
WHERE people.id = :id_1
```

You also can see from this query that SQLAlchemy binds parameters to variables inside your query, rather than directly placing your values. Not only does this allow you to re-run queries later with different variable values, but it reduces the possibility that you will

suffer from an SQL injection attack, which still is surprisingly common.

You also can order the results:

```
for person in session.query(Person).order_by('first_name'):
    print person.first_name
```

```
Foo
```

```
Reuven
```

And, you also can do all of the basic “CRUD” activities that you would expect. For example, you can create a new instance of your Person class and then save it to the database:

```
p = Person('newfirst', 'newlast', 'new@example.com', '1-jan-2012');
session.add(p)
session.commit()
```

Notice how I can handle multiple inserts (or other actions) inside a single transaction by only issuing `session.commit()` after adding several objects. Similarly, I can update the object and the corresponding row in the database:

```
p.first_name = '!!!!'
session.add(p)
session.commit()
```

I also can delete the object:

```
session.delete(p)
session.commit()
```

Relationships

If SQLAlchemy could only do this, it still would be a nice library, simplifying your queries. But the real power of SQLAlchemy occurs when you define relationships between tables. For example, let’s assume that I have an Appointments table, indicating when I’m meeting with various people:

```
CREATE TABLE Appointments (
    id SERIAL PRIMARY KEY,
    person_id INTEGER NOT NULL REFERENCES People,
    meeting_at TIMESTAMP NOT NULL,
    notes TEXT
);
```

Let’s also add some appointments:

```
INSERT INTO Appointments (person_id, meeting_at, notes)
VALUES (2, '1-jan-2013', 'New Year meeting'),
(2, '1-feb-2013', 'Monthly update');
```

Now I need to create a Python class that represents appointments:

```
class Appointment(Base):
    __tablename__ = 'appointments'
    id = Column(Integer, primary_key=True)
    person_id = Column(Integer)
    meeting_at = Column(DateTime)
    notes = Column(String)
```

Now, this class will work just fine. However, there’s no relationship,

according to Python, between the Person class and the Appointment class. To make this work, you'll need to change each of these table definitions. In the case of Appointment, you'll need to indicate that the person_id column doesn't just contain an integer, but that it is a foreign key that points to the "id" column on the People table:

```
person_id = Column(Integer, ForeignKey('people.id'))
```

On the Person table, you'll need to add a line to the class attributes, after describing all of the columns:

```
appointments = relationship("Appointment", backref="person")
```

Thanks to these two lines, you get an "appointments" attribute on your Person model. But thanks to the "backref" parameter, you also get a "person" reference on the appointment. This means you can do something like this:

```
for a in session.query(Appointment):  
    print a.person
```

```
for p in session.query(Person):  
    print p.appointments
```

Note that the assumption is that you'll have multiple appointments

per person, representing a one-to-many relationship.

Let's say, however, that you want to have a many-to-many relationship between people and appointments, such that you can meet with more than one person at a time, and you can have more than one appointment with a particular person. In order to do that, you need to modify your database table and code somewhat, adding a third (association) table. SQLAlchemy makes it easy to do that. Although I don't have space to show it here, the basic idea is that you create the third table, and you use the relationship() function to indicate that there is a secondary relationship between the class and the join table.

Conclusion

SQLAlchemy is packed with features. In addition to the introductory examples I showed here, it handles everything from joins to connection pooling, to dynamically calculated column values, to creating Python classes based on an existing database table. There is no doubt that it's a powerful system, one that I expect to use in some of the Python projects on which I work.

That said, I found SQLAlchemy to be a bit overwhelming for the newcomer. Perhaps it's because I have long used the Active Record model in Ruby,

which has minimal configuration and syntax, but I found the syntax for SQLAlchemy to be a bit overly verbose. Then again, Python has long preferred things be explicit, and there's no doubt that SQLAlchemy provides a clear and explicit ORM, without much magic and with obvious ramifications for every function call and parameter.

The other thing that might throw off newcomers to SQLAlchemy is that the documentation is complete, but not particularly friendly. Once you start to use the system, I expect that you (like me) will be able to understand the documentation and make good use of it. But I found that even the tutorial documents were a

bit formal, trying to tell you too much before moving ahead with actual code. Hopefully, this article can help some more people become interested in SQLAlchemy.

In conclusion, SQLAlchemy is a great Python module, one that deserves its sterling reputation and broad popularity. If you're interested in working with databases from Python programs, you definitely should take a look at SQLAlchemy. ■

Reuven M. Lerner is a longtime Web developer, consultant and trainer. He is also finishing a PhD in learning sciences at Northwestern University. His latest project, SaveMyWebApp.com, went live last spring. Reuven lives with his wife and children in Modi'in, Israel. You can reach him at reuven@lerner.co.il.

Resources

The home page for SQLAlchemy is <http://sqlalchemy.org>, and the documentation is at <http://docs.sqlalchemy.org>. The Python language is at <http://python.org>. I suggest that you read through the introductory section and then the ORM documentation, rather than look at the document sequentially.

There are a number of on-line tutorials for SQLAlchemy. Two that I enjoyed, which are freely available to the public, are <https://www.youtube.com/watch?v=399c-ycBvo4> and <https://www.youtube.com/watch?v=PKAdehPHOMo>.

Finally, Rick Copeland's book, *Essential SQLAlchemy*, published by O'Reilly in 2008, is a good introduction, particularly if you look at the ORM section. The rest is a bit dry and technical, even if the examples are well written. This book is not completely up to date, and there are several items in it that reflect the fact that it was published several years ago. Nevertheless, having an additional reference can be quite handy and can provide examples for certain features that aren't otherwise obvious.



DAVE TAYLOR

Cribbage: Sorting Your Hand

Continuing our development of a *Cribbage* game, this month Dave tackles the tricky task of sorting a hand by rank value.

We've been working on writing code for the game *Cribbage*, and last month, I ended this column by creating the code needed to pick a random subset of six cards out of a "deck" and display them in an attractive format—like this:

```
$ sh cribbage.sh  
Card 0: 7C  
Card 1: 5H  
Card 2: 9H  
Card 3: 10S  
Card 4: 5D  
Card 5: AS
```

The primary task on the agenda this month is to sort the cards after they've been dealt. This means we're going to have to sort the cards by rank while ignoring the suit, then slot them back into the "hand" array. Is there an easy way to do that? Actually, we'll use the `sort` function.

We can prototype this by using the command line to see what result we get:

```
$ sh cribbage.sh | sort -n  
Card 0: 4S  
Card 1: 7C  
Card 2: 9S  
Card 3: JC  
Card 4: 7H  
Card 5: 8C
```

What the heck? Oh! You can see the problem, right? By telling `sort` to order things numerically, it properly ignores "Card" but then sees the ordinal value of the card and sorts based on that, rather than on the actual card value itself.

Even if we fix this, however, we still have the problem that face cards will sort before numeric value cards, which isn't what we want. In fact, we want aces to sort as lower than 2s, while jacks, queens and kings sort as higher than 10s.

If you wanted to have aces “high”, the easiest way to do that would be to change the display routine, of course: 1 = a deuce, 2 = a three, 12 = king and 13 = ace. Poof. Everything sorts ace-high. That’s just not how Cribbage scores them.

To accomplish *Cribbage*-rank sorting, we’ll need to change the output to push out two values: the rank and the total card value. It’s going to look ugly, but it’s just an interim result.

Here’s how I tweak the code to display these values:

```
showcard()
{
  # given a card value of 0..51 show the suit and rank
  suit=$(( $1 / 13 ))
  rank=$(( ( $1 % 13 ) + 1 ))
  case $rank in
    1) orank="A" ;;
    11) orank="J" ;;
    12) orank="Q" ;;
    13) orank="K" ;;
    *) orank=$rank ;;
  esac
  showcardvalue=$orank${suits[$suit]}
}
```

If you compare it to the version we built last month, the main difference is that instead of calculating the rank of the card and then overwriting it with “A”, “J”, “Q” or “K” as appropriate,

we’re using a new variable, `orank`, to store the corrected value. Why? Because now in the main section of the script we also can access the `$rank` of the card as desired:

```
showcard ${hand[$card]}
echo "$rank ${hand[$card]}"
```

For each card chosen, the script has an interim output of rank followed by the numeric value of the card, with no fancy display (even though we’re still tapping the `showcard` function for simplicity). The result:

```
$ sh cribbage.sh
13 38
6 31
8 33
10 35
5 30
12 24
```

Ugly? Definitely. But now we can sort it and get useful results, even if they might not look like it quite yet:

```
$ sh cribbage.sh | sort -n
1 26
2 14
2 40
3 2
7 45
10 22
```

It still looks confusing, but you can see that it's in rank order.

So, how do we get that back into the "hand" array now that we know how to sort it? That's actually rather tricky because of variable scoping issues, as you'll see.

Before we go there, however, I've written a new "showhand" function that displays all the cards in the hand on a single line, with the help of /bin/echo for echoes without a trailing line break:

```
showhand()
{
    # show our hand neatly
    /bin/echo -n "Hand: "
    for card in {0..4}
    do
        showcard ${hand[$card]}
        /bin/echo -n "$showcardvalue, "
    done
    showcard ${hand[5]}
    echo "$showcardvalue."
}
```

With that available, our main code starts to look nice and clean:

```
dealhand;
showhand; # for testing sorthand only
sorthand;
showhand;
```

For debugging purposes, I'm going

to display the hand before and after we've sorted by rank. Eventually, the first "showhand" would just be axed, of course.

Now, let's get back to the code needed to sort the cards in our hand (a feature that a lot of iOS *Cribbage* games seem to omit, as far as I can tell).

My first stab at writing "sorthand" took advantage of a very slick feature in the Bourne shell that lets you tie the output of one loop to the input of another with a pipe. For example:

```
for card in {0..5}
do
    showcard ${hand[$card]}
    echo "$rank ${hand[$card]}"
done | sort -n | while read rank value
do
    hand[$index]=$value
    index=$(( $index + 1 ))
done
```

The problem is that the shell's pipe implementation pushes the second loop into a subshell without any easy way to get the changed values back up to the parent shell. The result: by the line immediately after the last done statement, all the new values have been lost.

That's too bad, because it definitely was more elegant. But then again, it's not about elegant, it's about

functional, right?

Here's how I actually solved it, by using a temporary file to store the intermediate results instead. It's considerably less elegant, for sure:

```
sorthand()
{
    # hand is dealt, now sort it by card rank...
    index=0
    tempfile="/tmp/.deleteme"
    for card in {0..5}
    do
        showcard ${hand[$card]}
        echo "$rank ${hand[$card]}"
    done | sort -n > $tempfile

    while read rank value
    do
        hand[$index]=$value
        index=$(( $index + 1 ))
    done < $tempfile
    rm -f $tempfile
}
```

Note that to get the input of the temporary file as the input for the `while` loop, I simply redirect `stdin` for the loop at the very *end* of the loop: `done < $tempfile`.

Let's test it by dealing a few hands and then showing them immediately post-deal and then after they've been rearranged with the `sorthand` function:

```
$ sh cribbage.sh
Hand: 9H, 6D, KC, AH, 9S, JH.
Hand: AH, 6D, 9S, 9H, JH, KC.
$ sh cribbage.sh
Hand: 4D, QS, AC, 9H, 10C, JS.
Hand: AC, 4D, 9H, 10C, JS, QS.
$ sh cribbage.sh
Hand: 9H, 10C, 7C, 7H, 5H, AS.
Hand: AS, 5H, 7C, 7H, 9H, 10C.
```

It looks like it's working exactly as we'd hope. Yeee-ha!

Yes, there are undoubtedly more efficient ways to write this code, and you can quite reasonably ask if a shell script is the optimal development environment for this sort of project, but, seriously, lighten up. Let's enjoy this project, not flagellate ourselves over punctuation!

And on that note, let's wrap up this month's column and start thinking about a considerably harder challenge we'll face starting next month: how to evaluate the *value* of the hand so that we can recommend which four of the six cards dealt should be kept to optimize the *Cribbage* hand.

You are learning *Cribbage* as we go, right? You'll want it for next month's installment, for sure. ■

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at <http://www.DaveTaylorOnline.com>.



KYLE RANKIN

Raspberry Strudel: My Raspberry Pi in Austria

A \$35 Raspberry Pi is now as powerful as my first collocated server. Find out how I tweaked the OS and racked it in a data center.

I remember my first collocated server rather fondly. It was a 1U Supermicro that had been decommissioned from my employer after a few years' service. Although it was too old and slow for my company, the 800MHz CPU, 1GB RAM and 36GB SCSI storage was perfect for my needs back in 2005. A friend was kind enough to allow me to collocate the server at his facility for free. So, after a lot of planning, I installed and configured Debian, generated SSH keys and set IPs so I could manage this machine remotely. Once it was collocated, it became my primary server for Web, DNS, SMTP and my perpetual

Irssi-in-a-screen session. The machine served me for more than five years until I ultimately replaced it with newer hardware.

Fast-forward to today, and although my primary server has significantly more resources, I just finished collocating a new server, again for free, and again with similar resources as my old Supermicro: 900MHz CPU, 256MB RAM and 40GB Flash storage. This time though, the server is a Raspberry Pi, and the facility is located in Austria. In this article, I explain how I was able to collocate a Raspberry Pi and the steps I went through to prepare it for remote management.

Although I certainly prefer servers that provide remote lights-out management, beggars can't be choosers, so many of my personal servers have had to make do with SSH and the ability to have someone cycle the power.

The Deal

When I first saw the deal advertised, I couldn't believe my eyes. An Austrian colocation facility, EDIS GmbH, was offering free Raspberry Pi colocation. Although I was a bit skeptical, I carefully read through the fine print, but it was pretty clear. If you set up an account, the EDIS GmbH folks would send you network information for your Raspberry Pi. Once you configured the network settings, you would send the Raspberry Pi along with the SD card, USB cable and optionally a small USB thumbdrive and they would rack it and provide 100Gb/month of traffic on a 100Mb connection. They even offered free remote power cycling of the server as long as you were fine with waiting 24–48 hours. I figured the worst that could happen is that I'm out a \$35 Raspberry Pi and some Flash storage, so I signed up and set aside a Raspberry Pi, 8GB SD card and 32GB thumbdrive while I waited

for my IP information.

The Setup

I'm no stranger to colocating servers without remote management. Although I certainly prefer servers that provide remote lights-out management, beggars can't be choosers, so many of my personal servers have had to make do with SSH and the ability to have someone cycle the power. Although I wasn't sure how I would use the server, I did know I wanted to keep the OS relatively lightweight. I also didn't want to take too many chances with a machine I would have little access to, so I went with the standard Raspbian "wheezy" Debian distribution linked to on the Raspberry Pi download page. There already is plenty of documentation on how to set up Raspbian, so I don't go into that here. Instead, I focus on the changes I made to the distribution before I shipped it off.

Because Raspbian assumes you will

run a desktop, it splits the available RAM with the GPU. Of course, on a server, you need the RAM for your services, so the next thing I did was run `sudo raspi-config` to launch an ncurses interface that let me tweak some of the hardware defaults. I ended up allocating only a little bit of RAM to the GPU, and while I was in the interface, I figured it wouldn't hurt to expand the root filesystem to fill my SD card, overclock the Raspberry Pi to 900MHz, change my locale and time zone, and change the boot behavior so the desktop didn't start at boot.

The default Raspbian image ships with a standard user and known password. I didn't want anyone to log in to my server except for me, so the next thing I did was add my own user account:

```
$ sudo adduser greenfly
```

Then, I edited the `/etc/group` file as root, and anywhere I saw the pi user, I added my greenfly user to the list. In particular, you would want to add your new user to the sudo group, because the default sudoers file on the distribution gives any members of that group full sudo privileges. At this point, I also used `ssh-copy-id` to copy my public SSH key to this server

so I could ssh in to it.

After I confirmed that I could log in as my user and sudo to root, I modified `/etc/ssh/sshd_config` and changed `PasswordAuthentication` to no, so I wouldn't have to worry about SSH brute-force attacks. Then, once I confirmed I could still ssh in, I deleted the pi user and removed its home directory:

```
$ sudo deluser --remove-home --group pi
```

Now that my user was set up, the next step was to remove all the desktop packages I would no longer need so I would have extra space for any services I wanted to install. There wasn't really a science to this; I just tried to pick base desktop packages I thought would have a lot of other desktop dependencies to remove:

```
$ sudo apt-get remove x11-common openbox-lxde omxplayer  
↳openbox libgtk2.0-common lxde-common xarchiver
```

Configure Bulk Storage

The base OS for the Raspberry Pi was on an 8GB SD card. I wanted the option to have more storage, and the folks at colo facility stated they would allow external USB drives as long as they were less than 4cm long. I had a 32GB USB stick that fit that profile and that showed

up as /dev/sda when plugged in, so I then proceeded to partition it and format it:

```
$ sudo fdisk /dev/sda
$ sudo mkfs -t ext4 /dev/sda1
```

One thing I didn't do was add the disk to my /etc/fstab. I didn't want to risk the server stalling in the boot process either because the USB drive was unplugged or had failed, so I decided to add the mount statement to the end of /etc/rc.local.

While I'm are talking about /etc/fstab, I also decided I should set up some swap storage for the device. I don't plan on needing swap, but I didn't want an out-of-memory issue crashing the server. Unlike with a traditional server, on Raspbian, the recommendation is to use `dphys-swapfile` to create a swap file that it takes care of mounting for you:

```
$ sudo dphys-swapfile setup
```

By default, it picks a swap file size it feels is optimal for your system, but you always can edit /etc/dphys-swapfile and change the size.

The IP Change

While I could set up a local network

to test the colocated network settings truly, I didn't want to go to the trouble, so the very last change I made to the system was the network settings. Before that point, I rebooted and updated the Raspberry Pi a few times and made sure I was still able to log in. Once I was ready, I edited /etc/network/interfaces and changed my eth0 network configuration from dhcp to static (IPs changed to protect, well, me):

```
auto eth0
iface eth0 inet static
    address 151.236.x.x
    netmask 255.255.255.0
    gateway 151.236.x.1
```

What I Wish I Would Have Done

It turned out it cost me only around \$5 to ship the Raspberry Pi from California to Austria with the US postal service. Of course, the moment I dropped it off, I started thinking about all the things I should have done. In particular, there are two things I wish I would have done.

First, I wish I would have set up a system so that the Raspberry Pi automatically e-mails me whenever it boots. That would have gone a long way toward helping with my impatience while I waited for the server to be racked. Instead,

all I could do was ping the IP and hope I set up the network right. To set it up to e-mail me, all I would have had to do is install the postfix package and during install time, configure it to be a standalone Internet mail server. Then I could install the mailutils package and add a mail command near the end of my `/etc/rc.local` file:

```
echo "I'm alive!" | mail -s "I'm alive!" me@mydomain.com
```

Then before I changed the network settings at home, I could have rebooted the system a few times and confirmed that the mail was sent.

The second thing I wish I would have done is *pay extra for a tracking number!* I was actually pretty angry with myself for forgetting to do this. Not only could I have known where the Raspberry Pi was while it was shipped, I also would have known when it arrived at the colo. Furthermore, without any tracking,

any dishonest person along the way could have pocketed the Raspberry Pi and said it was lost in shipping.

You may be wondering what I'm going to use this Raspberry Pi for after all. Since there isn't much redundancy, I'm not going to host anything critical on it; however, I'm considering what kind of redundancy I could get if I partnered up and shared resources with a fellow Raspberry Pi colo customer. At the moment, I'm just using it to provide a network sanity check so I can perform network troubleshooting from outside the US. Beyond that, I have set up postfix and nginx on it and plan to run some sort of rudimentary Web service and possibly backup DNS. Keep an eye on this column for updates as I start to add services to it. ■

Kyle Rankin is a Sr. Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Resources

The EDIS GmbH Raspberry Pi Colo Product Page:
<https://manage.edis.at/whmcs/cart.php?gid=6>

Raspberry Pi OS Download Page: <http://www.raspberrypi.org/downloads>

The Official Conference, Trainings, and Expo of The Apache Software Foundation



APACHECON

NORTH AMERICA 2013



Open Source Community Leadership
Drives Enterprise-Grade Innovation

Join us!

February 24th - March 2nd 2013

Portland, Oregon

Conference Website: na.apachecon.com Sponsorship inquiries send to:
ACNASponsors@theopenbastion.com

10% Discount with PROMO CODE LNXJ

Presented for
The Apache Software Foundation by:
The Open Bastion





SHAWN POWERS

Trying to Tame the Tablet

Don't let your fancy new tablet collect dust!

Like many folks, I received a shiny new Nexus 7 tablet for Christmas. This brought me great joy and excitement as I began to plot my future paperless life. For most of the evening and an hour or so the next day, I was sure the new Android tablet would change my life forever. Sadly, it wasn't that easy. This month, I want to dive head first into the tablet lifestyle, but I'm not sure if it's really the lifestyle for me. I'll try to keep everyone posted during the next few months (most likely in the Upfront section of *LJ*). And please, *please* don't hesitate to send me messages about the ways you find your Android tablet useful at work/home/play.

At Work

The main reason I decided on the Nexus 7 was because with the leather case I bought for it (Figure 1), it was small enough to carry to meetings easily, yet big enough to view full-size documents. I figured

with a tablet computer, I might be able to do away with most of the paper in my life. I have cabinets *full* of filed papers that I never use. I do, however, search my e-mail on a regular basis for communications sent or received years ago. I want that same accessibility for items that exist only in paper form now.

Paperless: Evernote or Dropbox

I've been trying to go paperless since long before I got a tablet computer. There seems to be two schools of thought in the paperless department. There are the Evernote people, and there are the "every-other-kind" of people. I have Evernote on every electronic device I own (which is a significant number), and I have to admit, for raw information, Evernote is amazing. The problem comes with documents. Granted, documents can be added to an Evernote note, but they are like e-mail attachments, and they can't be modified once



Figure 1.
My case
doubles
as a stand.

attached. This means, at least for me, that the only documents I ever attach are “complete” documents that are printed as PDF files.

I don’t have a good solution for how to handle Word/LibreOffice documents in Evernote. So, that

means I have an inconvenient combination of Evernote for unformatted information and Dropbox for documents. Thankfully, both applications run very well on Android, so although I don’t have a central repository for all my information, at

After buying a stylus, coming up with a note-taking application proved to be difficult.

least I can access all the information from my tablet.

Getting Data In

Evernote includes a really nice mechanism for using a device's camera for importing digital snapshots of documents, notes, whiteboards and so forth. Unfortunately, the Nexus 7 doesn't have a rear camera. Thankfully, my cell phone has a really nice camera, and it also has Evernote installed. Because I never intended my tablet to replace my cell phone,

this isn't a big issue for me. I just whip out my phone if I need to import something optically into Evernote.

My biggest hope with the Nexus 7 was that I could avoid toting around legal pads and pens to meetings. I tend to take "doodle" notes, so a laptop really isn't ideal for me at a meeting. (Plus, I tend to become distracted with a laptop and multitask my way into trouble quite often.) I researched capacitive styli and found the New Trent IMP62B to be just about the best option (Figure 2). It's

Figure 2. This stylus is remarkably precise given the size of its tip.



less than \$10, and it's remarkably precise for a stylus with a rather bulbous tip.

After buying a stylus, coming up with a note-taking application proved to be difficult. I almost can get there with a couple apps, but nothing has been the ideal option for me. The closest I've come to perfection is Lecture Notes, which has some critical features:

- Importing PDF files from Dropbox for annotation during a meeting (for example, an agenda).
- Exporting directly to Evernote.
- Very fine lines when writing.
- Simple interface for changing pens, erasing and so on.

I'll admit, it's still not as fast as writing on paper, but for some quick doodles on a PDF agenda, Lecture Notes does a nice job (Figure 3).

My wife actually likes to type on her tablet (an iPad Mini) with the onboard keyboard. If she's taking notes, she'll just open up Google Docs and type on the screen. For me, typing on any screen is awkward and slow. If I have to do any real typing on my tablet, I'll use a Bluetooth keyboard. At that

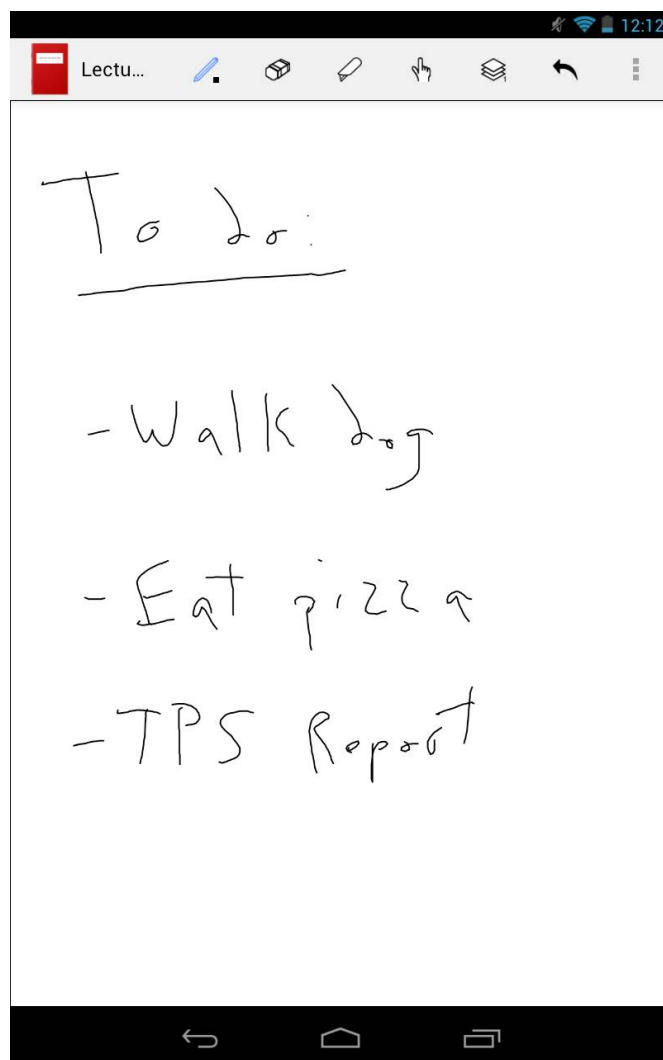


Figure 3. Lecture Notes is a great application if you want to take notes with a stylus.

point, however, I might as well just use a laptop. In a pinch, it's certainly possible to type a few notes with the on-screen keyboard, and if you don't have a laptop, a Bluetooth keyboard will help manage some serious typing. Still, I don't recommend it. Any Nexus-size keyboards are too small to type well with, and any

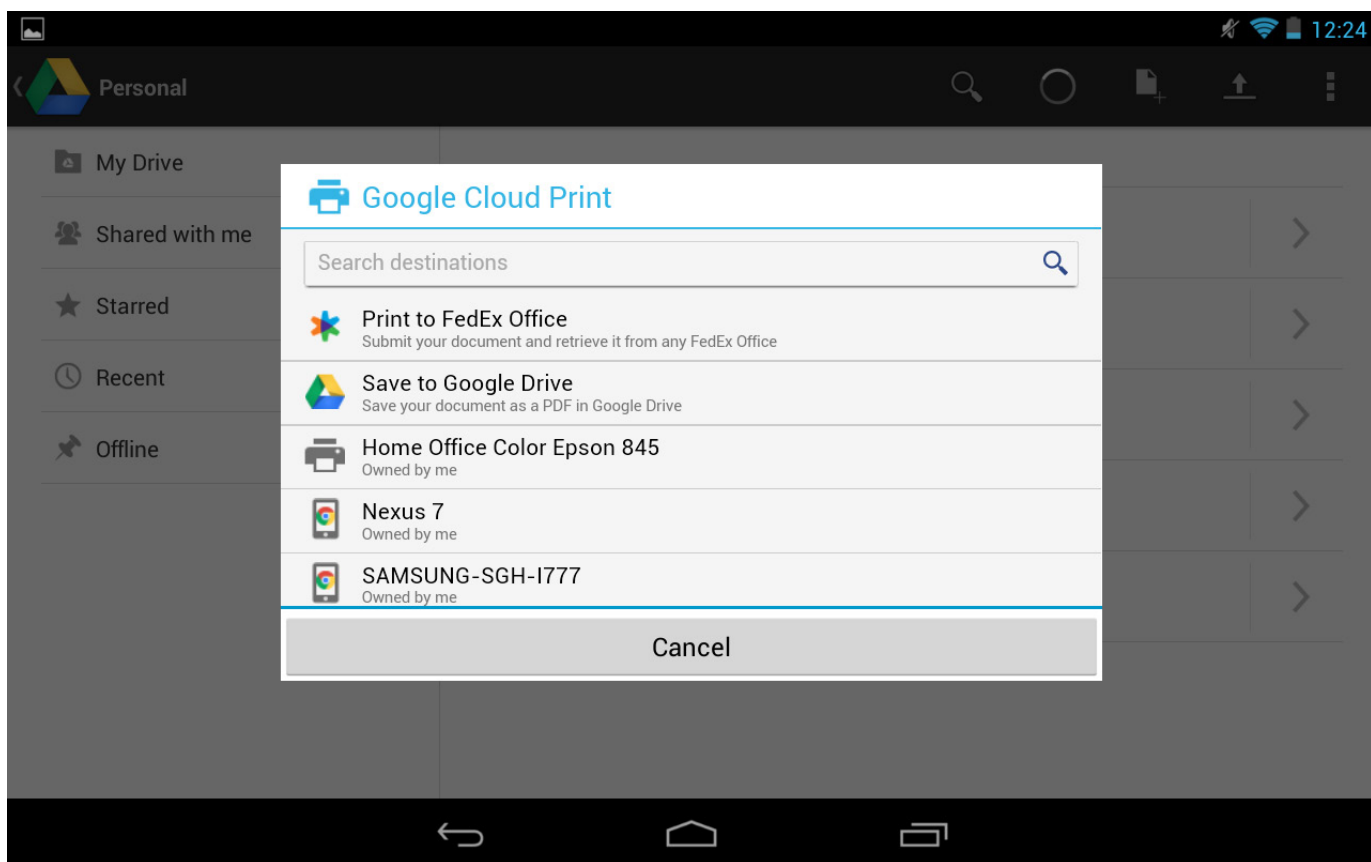


Figure 4. My printer has native Google Print support, but it's possible to set up a traditional printer.

full-size Bluetooth keyboards are cumbersome to carry around.

Printing and Viewing

Just a couple years ago, it was absurd to think about printing from a phone or tablet. Now, it's easy to set up network printing for Android devices, and Linux users easily can share printers with iOS devices as well. So printing, interestingly enough, is fairly ubiquitous. Figure 4 shows an example of printing from

Google Drive.

Speaking of Google Drive, the native Google application does a decent job of creating Microsoft-compatible Office files. The newest version of Drive even allows editing and creating spreadsheet files! When combined with Android's built-in file viewer, it's difficult to find a document Android can't read. I've never been stuck in a meeting unable to view an e-mail attachment, which would be a real showstopper for me at work.

Geeking It Up

If you're stuck wearing a tie and attending meetings all day, the above information might be all you're interested in. For me, although I attend more meetings than I care for, I also have the opportunity to be a geek. A tablet computer offers some really great apps for system administrators or just geeks on their lunch break. Here are some of my favorites:

- **ConnectBot:** this is the de facto standard SSH client for accessing remote servers. As with typing long documents, the on-screen keyboard can be frustrating for more than a few quick server tweaks, but the program itself is awesome. If you've ever SSH'd into a server on a cell-phone screen, the 7" of real estate on the tablet will be a godsend. No geek is complete without a command-line interface, and ConnectBot provides remote access to one.
- **WiFi Analyzer:** I've mentioned this app before in *Linux Journal*, and rightly so. It does exactly what's on the tin: it analyzes the Wi-Fi networks in your area. Whether you want to find an open channel or check signal strength in different

areas of your building, WiFi Analyzer is amazing.

- **WiFi Map Maker:** I had never heard of this application, but a reader (Roman, I won't mention his last name out of respect for his privacy) sent me information on it. If you need to make a quick-and-easy map of Wi-Fi hotspots, this is hard to beat. It uses the built-in GPS on your tablet to create a thermal map of Wi-Fi coverage in real time.
- **SplashTop:** now that SplashTop supports controlling Linux workstations along with Windows and OS X, it's become a whole lot more usable for me. Using its custom application installed on your computer, SplashTop allows remote control of workstations with incredible responsiveness. It's a bit like VNC simplified and on steroids. Heck, it's even possible to play PC games over the connection! (Not that you'd ever do that at work.)

At Home: a Boy and His Recliner—and Tablet

I don't think I've watched a television show or movie at my house in the past decade without a notebook computer sitting on my lap. Whether it's to look up an actor on IMDB or to catch up on

RSS feeds during the boring scenes, an on-line connection has become a requirement for me in my recliner. In this case, I've found the Nexus 7 to be a decent replacement for a full-blown laptop. Not only can I do all the things I normally do with my laptop, but I also can use an XBMC remote application to control the TV. If I happen across a cool on-line video, I can send it to my XBMC unit quickly with iMediaShare, which uses Apple's AirPlay technology to stream video directly to the TV. It gives me a certain level of satisfaction to stream video from an Android device to my Linux nettop running XBMC using an Apple protocol, yet having no Apple hardware or software in the mix. Truth be told, it works a lot more consistently than the Apple TV and actual AirPlay does. iMediaShare has both a free and paid version, which are available on the Google Play store.

One thing I never do on my laptop is read books. Even though I can read countless Web articles on the computer, for some reason, I can't bring myself to read actual book-length material. With the tablet on my lap instead of a laptop, flipping open the Kindle app allows me to read a few pages of a book if there's nothing interesting on TV. Why the Kindle app? I'm glad you asked. As it turns out, even though it has the absolute

worst interface for finding a book in your collection, it has some features that I find indispensable:

- With the "Personal Documents" feature Amazon offers, any DRM-free ebook can be e-mailed and stored on Amazon Cloud. They can be retrieved from any Kindle device or app (excluding the Cloud Reader, but I don't read books on my computer screen anyway).
- WhisperSync used to work only on Amazon-purchased materials, but now it works on Personal Documents too. This means I can pick up my cell phone to read a few pages at the doctor's office, and then pick up my tablet later and automatically be right where I left off. Because this works across platforms, it makes the Kindle reader my go-to app.
- I keep my DRM-free e-book collection at home on Calibre. With Calibre's export feature, sending a book to a specific Kindle device's e-mail address is a single click away.

I really do wish Amazon would improve the browsing interface for Android devices. I suspect Amazon is trying to push people into buying

a Kindle Fire, however, since it also won't release the Amazon Prime streaming app. Oh well, the WhisperSync feature makes all the difference for me, and I'm willing to suffer a cruddy interface when opening a book.

Pure, Down-Home Entertainment

The tablet size and touchscreen really do make it a perfect device for simple gaming. Whether you want to sling *Angry Birds* at a bunch of pigs or use the tablet like a steering wheel to drive your 4x4 across rough terrain,

the Nexus 7 is awesome. I'm not much of a gamer, but as it happens, that's exactly the type of person tablet games are made for! If I want to play a quick game of *Solitaire* or even shoot a couple zombies, the tablet interface is perfect.

Entertainment doesn't stop with games, however. I've mentioned Plex in recent issues of *Linux Journal*, but it bears mentioning again. If you have a collection of videos on your home server, Plex will transcode and stream them to you anywhere. It works at least as well as the AirVideo

LINUX JOURNAL

on your
Android device

Download app now in
the **Android Marketplace**

www.linuxjournal.com/android



For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.

application on iOS, and the server component works excellently on a headless Linux server. When you add Netflix, Hulu Plus, Smart Audiobook Player, Pandora, Google Music, Amazon MP3 and the ability to store local media, it's hard to beat the Nexus 7 for media consumption.

And in between Work and Home

One place I never expected to use my tablet was in my car. No, I don't read books or watch videos during the daily commute, but I certainly enjoy listening to audiobooks. With its built-in Bluetooth connection, I happily can

stream a book through my car's audio system. I find traffic jams much more palatable now that it means more time for "reading".

I've also found Google Map's ability to download maps for off-line use to be awesome. I opted to get the Wi-Fi-only model of the Nexus 7, so when I'm in the car, I don't have Internet connectivity. My car doesn't have a navigation system, so the 7" screen and off-line maps make for an incredible GPS system. Google's turn-by-turn navigation is amazing, and the nice big screen means it's more useful

Resources

Dropbox: <http://www.dropbox.com>

Evernote: <http://www.evernote.com>

New Trent IMP62B Stylus: <http://www.newtrent.com/stylus-pen-imp62b.html>

Google Drive: <http://drive.google.com>

ConnectBot: <http://code.google.com/p/connectbot>

WiFi Analyzer: <https://sites.google.com/site/farproc/wifi-analyzer>

Dave's Apps: <http://www.davekb.com/apps>

SplashTop: <http://www.splashtop.com>

iMediaShare: <http://www.imediashare.tv>

XBMC: <http://www.xbmc.org>

than my phone's GPS. I don't have a great way to mount the tablet in my car yet, but I suspect with a bit of Velcro, it won't be a big problem.

Where to Go from Here?

I've given you a glimpse at how I use my tablet on a day-to-day basis. I hesitated to do this though, because I don't feel I'm really using the Nexus 7 to its fullest potential. Based on a few conversations I've had with fellow readers, however, I don't think I'm alone. I don't think tablet computers will replace desktops or even laptops

any time soon, but I do think they have a place in our daily lives. Hopefully this article gets you started with integrating a tablet computer into your everyday life. I look forward to hearing about and sharing your experiences, so please write me at shawn@linuxjournal.com. ■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

LINUX JOURNAL

now available
for the **iPad** and
iPhone at the
App Store.



linuxjournal.com/ios

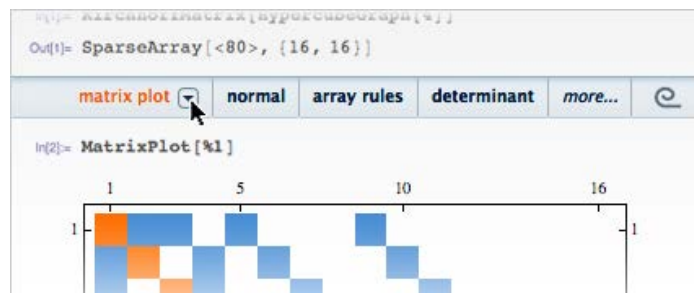


For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.

AdaCore and Altran Praxis' SPARK Pro



SPARK Pro is a product jointly developed by Altran Praxis and AdaCore that provides the language, toolset and design discipline for engineering high-assurance software. The developers say that the new version 11 of SPARK Pro offers many enhancements related to the way that functions and proof functions are handled. These changes are said to improve project efficiency by eliminating the vast majority of rules that previously were encoded manually. The main changes include a more powerful language for specifying proof functions and the ability to use the functions in any proof context. This greatly simplifies the task of writing and maintaining functional contracts for critical software, providing high assurance at lower cost. SPARK Pro combines Altran Praxis' SPARK language and verification tools with AdaCore's GNAT Programming Studio and GNATbench Integrated Development Environments. There are SPARK versions based on Ada 83, Ada 95 and Ada 2005, so all standard Ada compilers and tools work out of the box with SPARK, say the companies. <http://www.altran-praxis.com> and <http://adacore.com>



Wolfram Research Mathematica

Wolfram Research calls its Mathematica application, recently upgraded to version 9, "the broadest, deepest computation system in the world". Mathematica

9 adds more than 400 functions in new and expanding application areas and also introduces the Wolfram Predictive Interface. The latter, intended to help users fully utilize Mathematica's vast scope and depth, is a suite of features that intelligently suggests what to try next based on sophisticated heuristics and data from millions of queries from the Wolfram|Alpha site. Other new features include highly integrated units support; major new data science, probability and statistics functionality; full R integration into the Mathematica workflow; 3-D volumetric image processing and others. Supported platforms include Linux x86, Windows and Mac OS X.

<http://www.wolfram.com>



Opengear's ACM5504-5-G-W-I Gateway

Improving the management of critical infrastructure in remote locations is what the new Opengear ACM5504-5-G-W-I Remote Infrastructure Management (RIM) Gateway is all about. A new member of Opengear's ACM5500 product family, this sibling offers IT managers a wide range of connectivity options with its new integrated wireless access point (Wi-Fi 802.11 b/g/n) to complement the cellular, wired and wireless access already present in other gateways. Besides enabling direct management of Wi-Fi-enabled devices, the solution provides the option of convenient wireless access to the management network using mobile devices, such as tablets and smartphones. All RIM gateways in the ACM5500 product family provide serial console-port connectivity, environmental monitoring, power management and monitoring and remote site storage of off-line logs and running configuration files.

<http://www.opengear.com>

Jon William Toigo's *Office Automation 2.0* (Apress)

Those in our midst who look after enterprise-wide IT planning should sneak a peek into Jon William Toigo's new book *Office Automation 2.0*. The Apress title is an essential guide to office automation in the post-PC era and helps businesses assess such technologies as virtual desktop infrastructure, mobile clients and cloud services in terms of their practical applications to streamlining workload. Toigo emphasizes that rollouts of the latest enterprise-class technologies cannot produce business value unless management ensures that the front office is trained to use them correctly, and that end-user practices and IT processes are dynamically and efficiently coupled in the organizational culture. Toigo also provides practical guidance for innovative managers who are seeking to make every automation investment dollar count toward the three key metrics of business value: cost-containment, risk reduction and improved productivity.

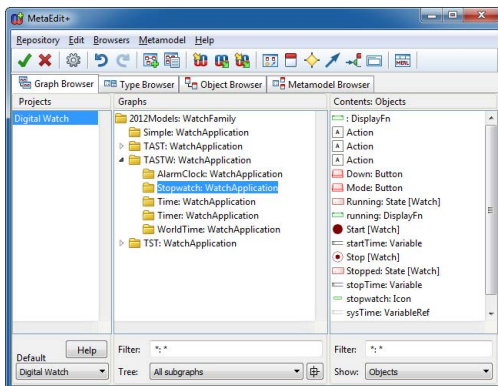
<http://www.apress.com>



Philatron Wire and Cable's Flexy CordT

Traditional extension cords can waste you a lot of time, either by having to unwind and rewind them repeatedly, or worse, when you struggle trying to untangle a mess of wire. The answer to this struggle, says Philatron Wire and Cable, is not in the numerous cord-reeling and -wrapping devices on the market, but rather in the company's new Flexy CordT, which it bills as the "21st Century extension cord". Flexy CordT line of extension cords are designed with a reduced cord and increased coil diameters, which gives them suppleness properties similar to the Slinky toys. In addition, Flexy Cords are developed with specially engineered materials with "memory" (so they "remember" their original retracted length) and are tangle-proof and kink-proof. Flexy Cords are available in different compact lengths: 4 inches (extending to 8 feet), 5 inches (extending to 10 feet), 10 inches (extending to 20 feet) and 20 inches (extending to 45 feet).

<http://www.flexycord.com>

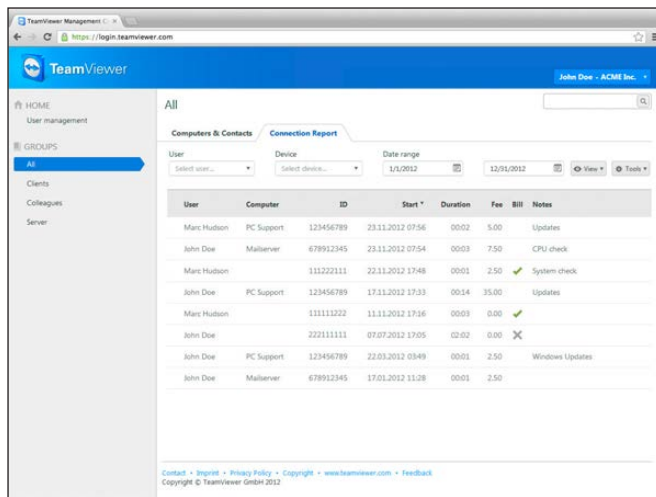


MetaCase's MetaEdit+

The new v5.0 release of MetaEdit+ from MetaCase adds a wide range of features to the company's flagship software development tool. MetaEdit+ is aimed at expert developers who seek to create graphical domain-specific languages and code generators rapidly. MetaCase states that the rich graphical notations go beyond plain icons and links;

they can change on the fly depending on model data, be nested to unlimited depth, be retrieved from libraries, and they have a fixed or dynamically varying number of ports to which to connect. These new features allow domain-specific models to mimic closely the problem domains they describe. The new version integrates into programming tools like Visual Studio and Eclipse. Software developers get one-click access from their IDE to MetaEdit+ models, can integrate generated code with hand-written code and libraries, and automate their build process. Versions for Linux, Windows and Mac OS X are available.

<http://www.metacase.com>



TeamViewer

TeamViewer is a popular remote-control and on-line meetings application for Linux, Windows and Mac OS X. The most notable feature in the new version 8 is the TeamViewer Management Console, a cloud-based administration tool that offers a wide range of capabilities that addresses the needs of corporate environments, with emphasis on

accountability, stricter security guidelines and the need for central control of user accounts. Other major new capabilities of TeamViewer 8 include connection reporting of all sessions and browser-based single-click connections. New features that reflect the latest demands of telecommuters include session handover, remote printing, deeper Microsoft Outlook integration, transmission of remote sound and video and enhanced session recordings.

<http://www.teamviewer.com>

Investintech.com's Able2Extract PDF Converter

You might call Investintech.com's Able2Extract PDF Converter the Swiss army knife of PDF converters. Not only is Able2Extract able to convert PDFs to a wide range of formats accurately, but it also features the unique ability to work across Linux (Ubuntu and Fedora), Mac OS X and Windows platforms. Investintech.com boasts that Able2Extract keeps intact all aspects—images, colors, formatting and fonts—regardless of file format. Supported formats include converting PDF to OpenOffice.org, MS-Office, AutoCAD and commonly used image formats. In addition, users can focus on the content they need by selecting conversion down to a single sentence.

<http://www.investintech.com>



Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o Linux Journal, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

ConVirt

the New Tool in Your Virtual Toolbox

Can you manage different hypervisor platforms from a single pane of glass? Yes, you can.

JERAMIAH BOWLING

Virtualization is now a staple of the modern enterprise. As more and more shops switch to the virtual paradigm, managing those new virtual resources is a critical part of any deployment. For admins using Microsoft- or VMware-based hypervisors, powerful management tools are available to keep their virtual houses in order. Unfortunately, those products and their accompanying tools come with a hefty price tag. The good

news is that inexpensive open-source virtualization is on the rise, driven in large part due to its low performance overhead. However, one of the primary obstacles to large-scale open-source virtualization adoption has been the lack of robust management tools. virt-manager is the most well known and used, and although it's a great tool, it does not hold a candle to the enterprise tools put out by the big vendors. That's where ConVirt comes in.

ConVirt is an open-source tool capable of managing multiple types of hypervisors including Xen, KVM and now VMware from a single pane of glass. When evaluating ConVirt for your needs, it is best to think of it as a front end to the native tools of the hypervisors that provides extended features not available in a standalone hypervisor. Although there is some overlap with virt-manager, ConVirt adds an additional level of enterprise manageability. ConVirt is currently offered in three tiers: Open Source, Enterprise and Enterprise Cloud. This article focuses on the open-source version. The open-source version does not include the ability to manage VMware items, so the testing environment for this article contains only Xen and KVM servers. Even though I don't cover it here, the ability to manage VMware hosts along with KVM and Xen hosts from the same pane of glass should peak the interests of many admins.

Let's get started by installing the ConVirt Management Server or CMS. ConVirt can be installed on most flavors of Linux or as a pre-configured virtual appliance that can be imported into a KVM or Xen server. I chose to deploy my CMS on a physical server running CentOS 6.2 to allow plenty of storage space (the virtual appliance

is roughly 2–3GB in size), although the appliance probably will get you up and running faster. Make sure that whichever installation method you select, that you open all the necessary ports on your CMS and on your managed servers/hosts that you want to manage through the console (TCP 8081, 8006, VNC ports and SSH).

The term "managed server" refers to those hosts running a hypervisor that is managed by ConVirt and can be used interchangeably with the term "host". Follow the installation procedures available on the Convirture Wiki site to perform the installation of the CMS. Most of the install is handled by a script that pulls down the dependencies and installs MySQL. I won't go into finer detail on the server install, as it is well documented on the site and I would just be repeating the information here.

After the CMS install is complete, access your management page at <http://youripaddress:8081> (Figure 1). Use the default login of "admin/admin" to bring up the main console. For those used to VMware's vSphere, this interface will feel familiar. The layout is broken into three main panels: a navigation panel on the left, a display panel for selected items in the middle of the page and a panel at the bottom for displaying

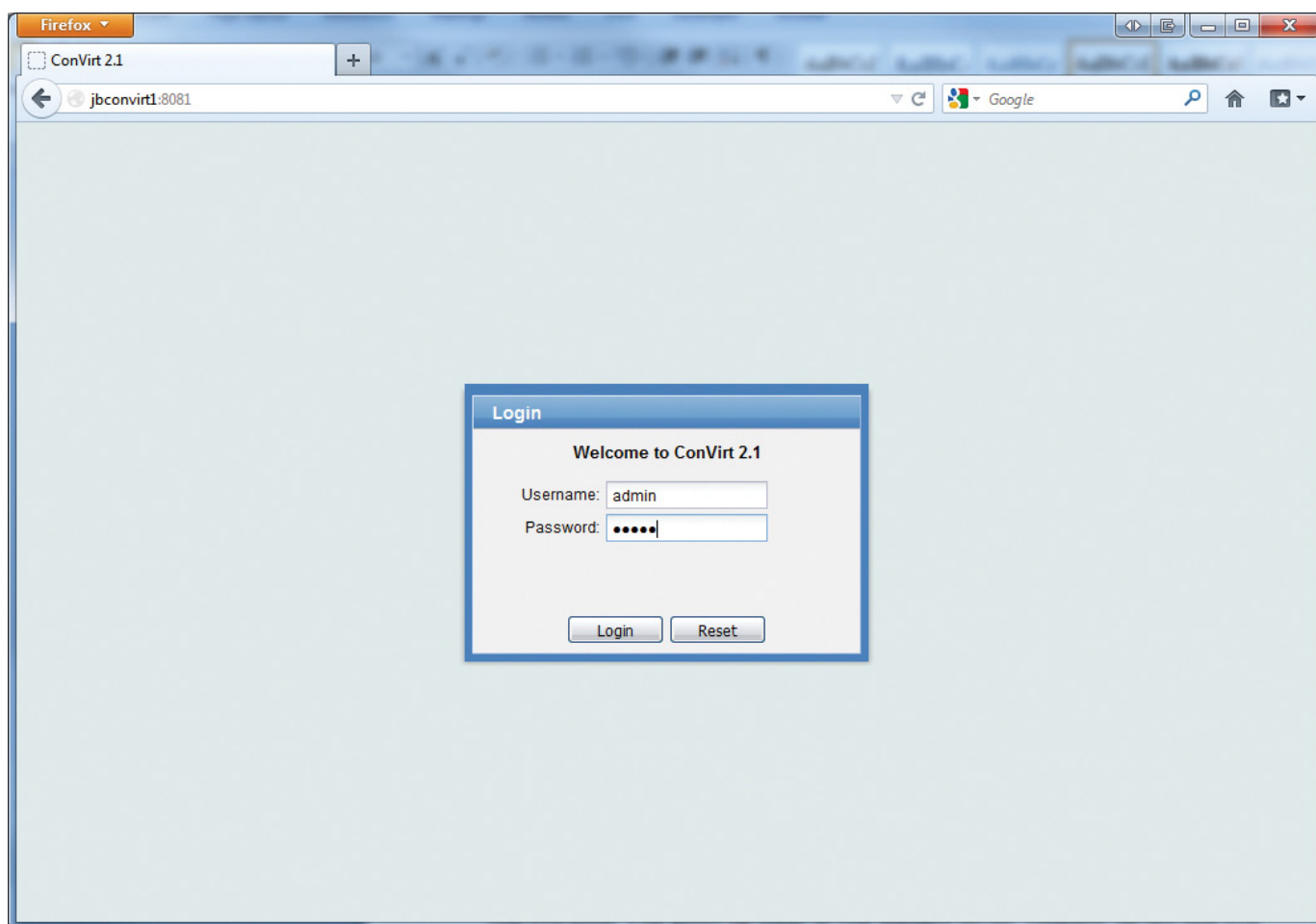


Figure 1. The Main Login Screen

task results (Figure 2).

The navigation pane is logically divided into a tree with your Data Center at the top with Server Pools and Templates listed underneath it. This outline reflects how resources are organized in ConVirt: Data Center→Server Pool→Managed Server (host)→Guest. Your Data Center is the top-most delineation of your virtual environment. It could be a site or an organizational unit. Under the Data Center are Server

Pools that group together like managed servers that share common items like storage and virtual network configurations. Managed servers are placed in the server pools along with any guests/VMs that reside on them. Templates fall into their own category, but also are available from the navigation pane. Templates are pre-configured groups of settings used at provisioning time to carve up/define the virtual resources available to new guests (processors, memory,

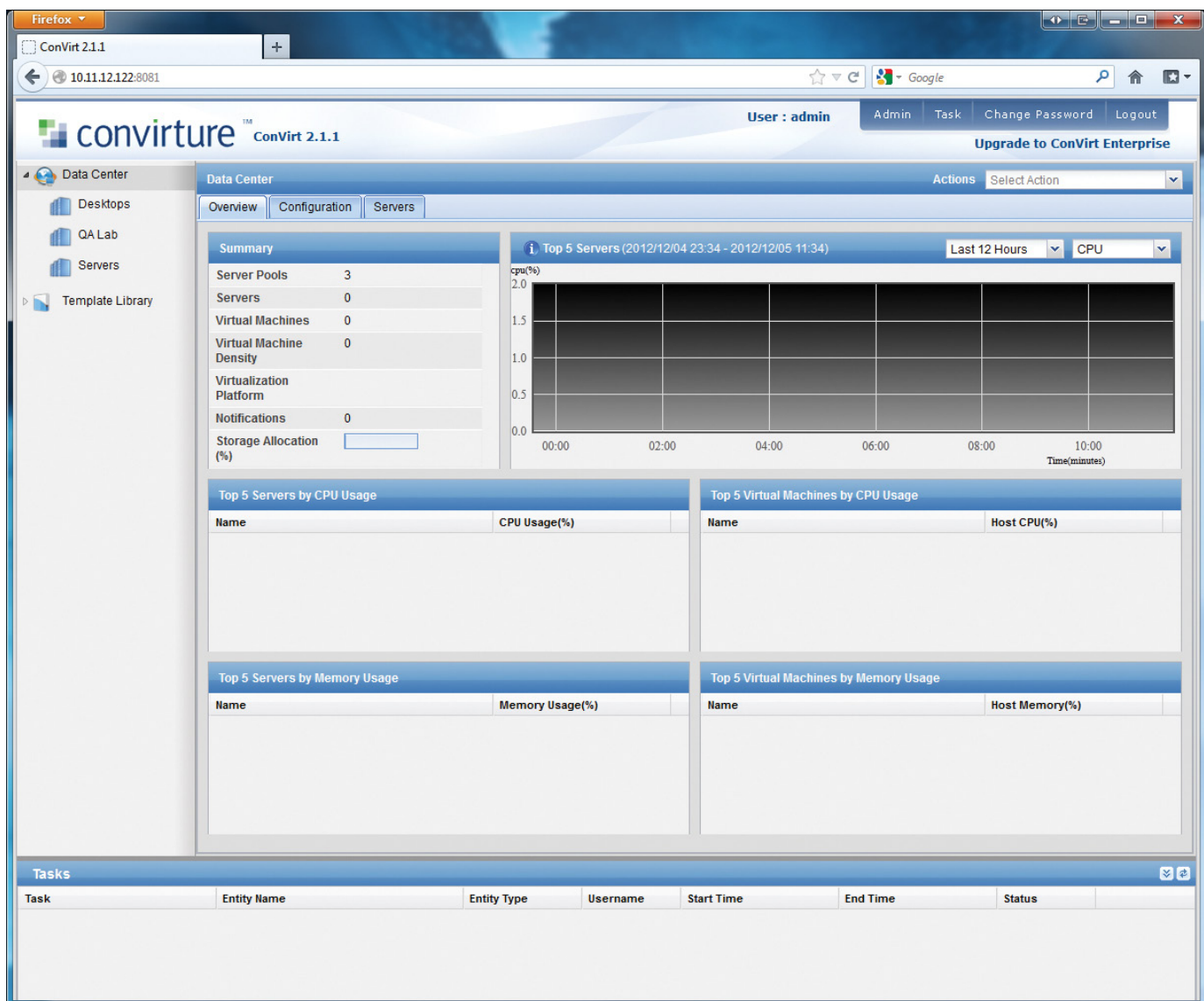


Figure 2. First View of the Data Center

storage and NICs).

The next step in your deployment is to prepare your hosts to become managed servers. Specific hypervisors have individual requirements before being added to the CMS, but the process for preparing each host is roughly the same for each. Create a network bridge on each host,

download the ConVirt tool from the site and install any dependencies. Then configure SSH on each managed/server host for root access, and finally, run the `convirt-tool setup` command. Debian/Ubuntu users should note that you will need to set a password on the root account manually in order to manage

any hypervisor from the CMS. I also suggest that you name any bridges you create with identical names (for example, KVM=br0, Xen=Xenbr0), as this helps standardize your guests' networking options. For this article, I created two KVM servers and one Xen server to manage with ConVirt.

With the hosts prepared, you now can add them to the CMS. This starts by adding hosts to a server pool. You can use the pre-configured Server Pools (Desktop, Server, QA Lab) or create your own. I created an additional pool to play with that I named "Production", and in case I messed anything up, it wouldn't affect the default pools. When you have your pool selected, right-click on it and select Add Server. On the resulting screen, select your platform, either Xen or KVM, and fill in the hostname or IP address.

If you have not configured SSH for root access on the host, the server will fail. If the server is added successfully, it now should display under the server pool you chose with a little K (KVM) or X (Xen) icon (Figure 3). Click on the newly added server to see performance information about your host displayed in the center pane (Figure 4). From this display, you also can view the number, type and status of the guest running on the host.

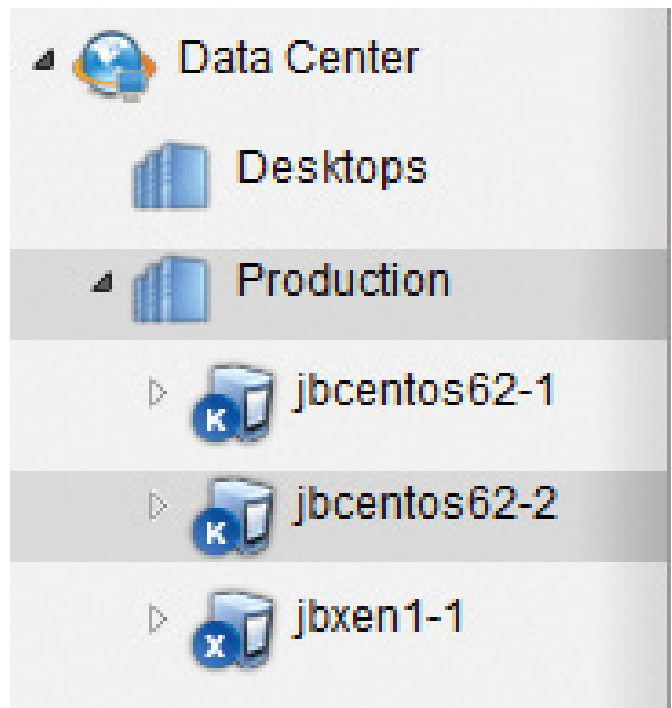


Figure 3. Our New Server Group

Continue adding all of your hosts as managed servers to the console until they have all been added. You then can import any pre-existing VMs on your hosts by right-clicking the managed server and selecting Import Virtual Machine Config Files. You also might notice from this same menu context that you can move servers between pools. This feature is useful during organizational changes or when moving test servers into a production environment. Be aware that moving a server between pools also moves any that reside on it, so be aware of any configuration changes that might be applied by moving your server/guests into the

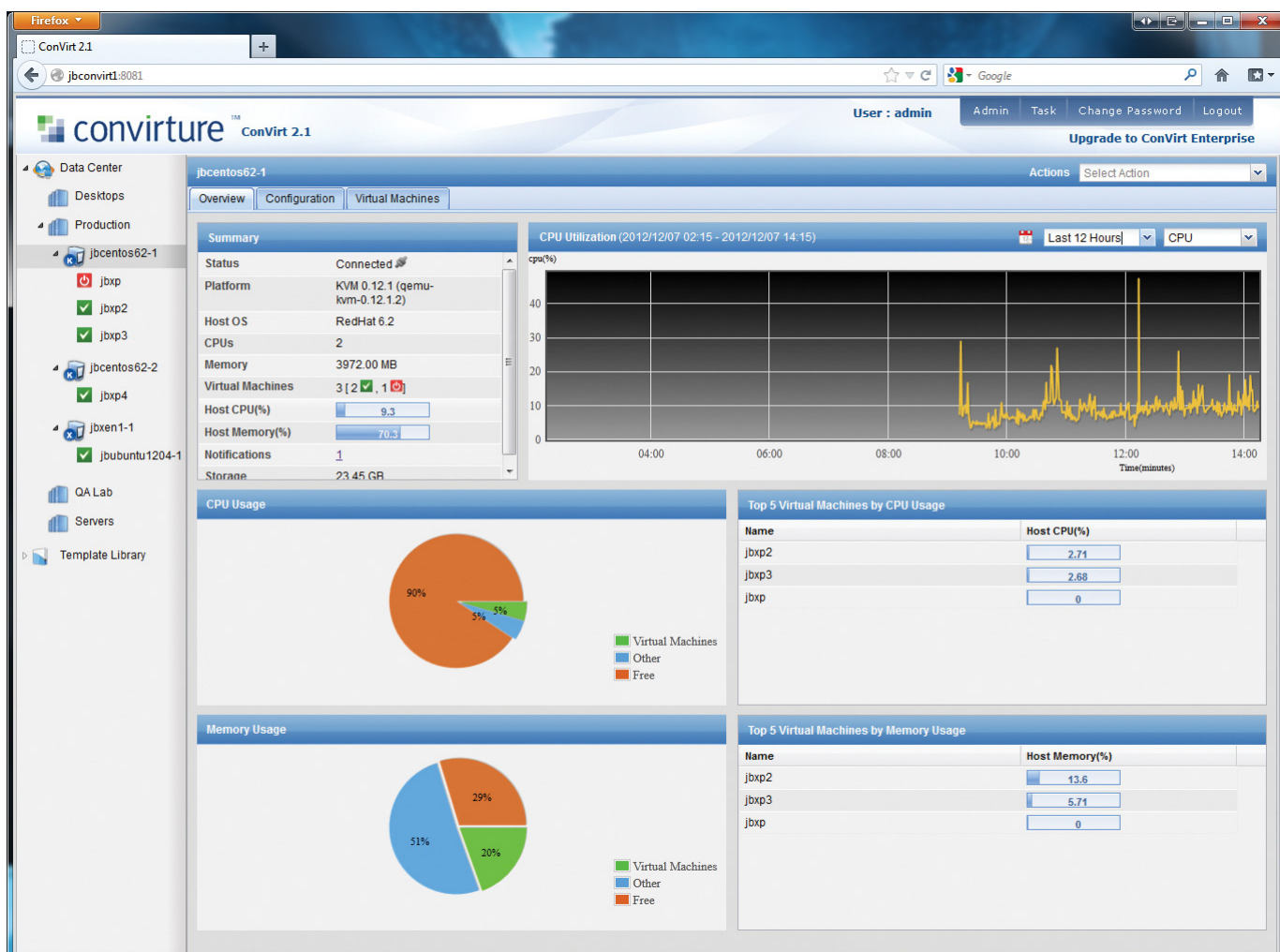


Figure 4. Real-Time Performance Stats on One of Our KVM Servers

new pool. You also are required to power down any running guests before moving the server.

Because I already have covered how to add existing guests to managed servers, let's create a new guest from a template (this also is called provisioning). To get a feel for all of your options, let's provision a guest VM from CD as well as clone a guest from a golden image using a reference disk.

Out of the box, ConVirt has two pre-configured templates for use with provisioning. These templates contain common configuration settings for a specific OS installed from a CD. Provisioning from the built-in templates is easy. Simply right-click a template, and select Provision to create a guest on your selected managed server.

For this example setup, let's create a Linux desktop from the existing

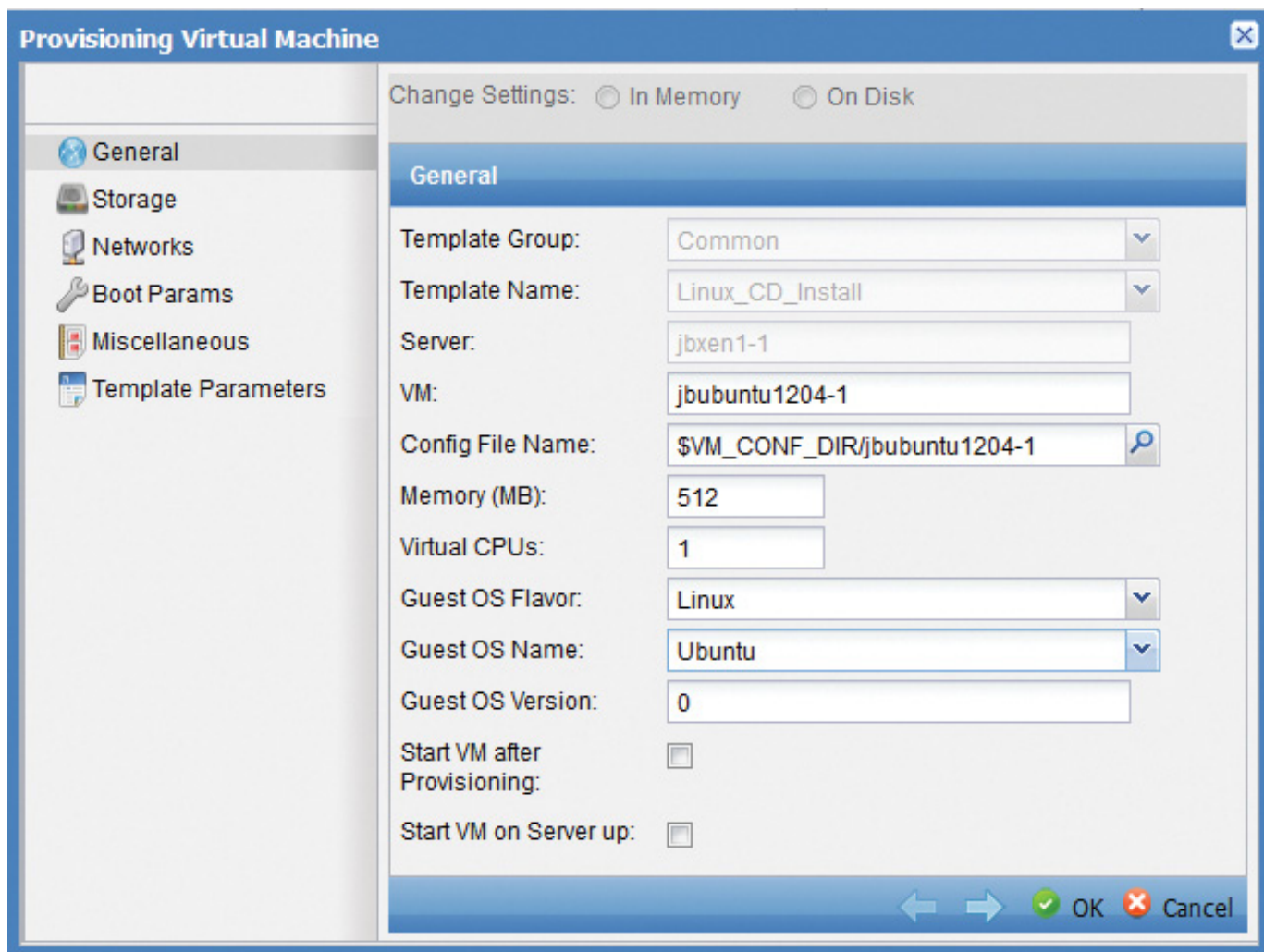


Figure 5. Provisioning a Guest from the Linux Template

Linux CD template. After clicking on Provision, you are asked on which server to place the new guest VM, and then you're prompted to provide a name for it (Figure 5). ConVirt then creates a guest based on your name and creates a 10GB virtual hard drive and maps the guest to the physical CD/DVD of the host on which it's provisioned.

Next, insert your physical install media on the host's physical drive.

Once the guest VM appears under the host, power it up by right-clicking on the new guest and clicking Start.

If you do not want to use CDs, you also have the option to boot from an ISO file. To do so, change the path of your /dev/cdrom to an accessible ISO file (Figure 6) in the settings of a template or the guest itself. Once the VM has been started, right-click on it and select View Console. If you have a Java-enabled browser, you can access



Figure 6. Mounting an ISO to the Guest CD-ROM

the new VM's desktop via the Web console, or if you choose another VNC client, ConVirt will display the IP and port required to access the VM. If you prefer to administer your host via SSH, you also can launch a session from the guest's right-click context menu.

Provisioning from CD is nice for

custom machines or one-off builds, but if you have to spin up multiple guests at once, it is a very inefficient method. It is much more efficient to create a single VM and clone it over and over again, which is possible in ConVirt. To demonstrate this method of provisioning, I created a pristine

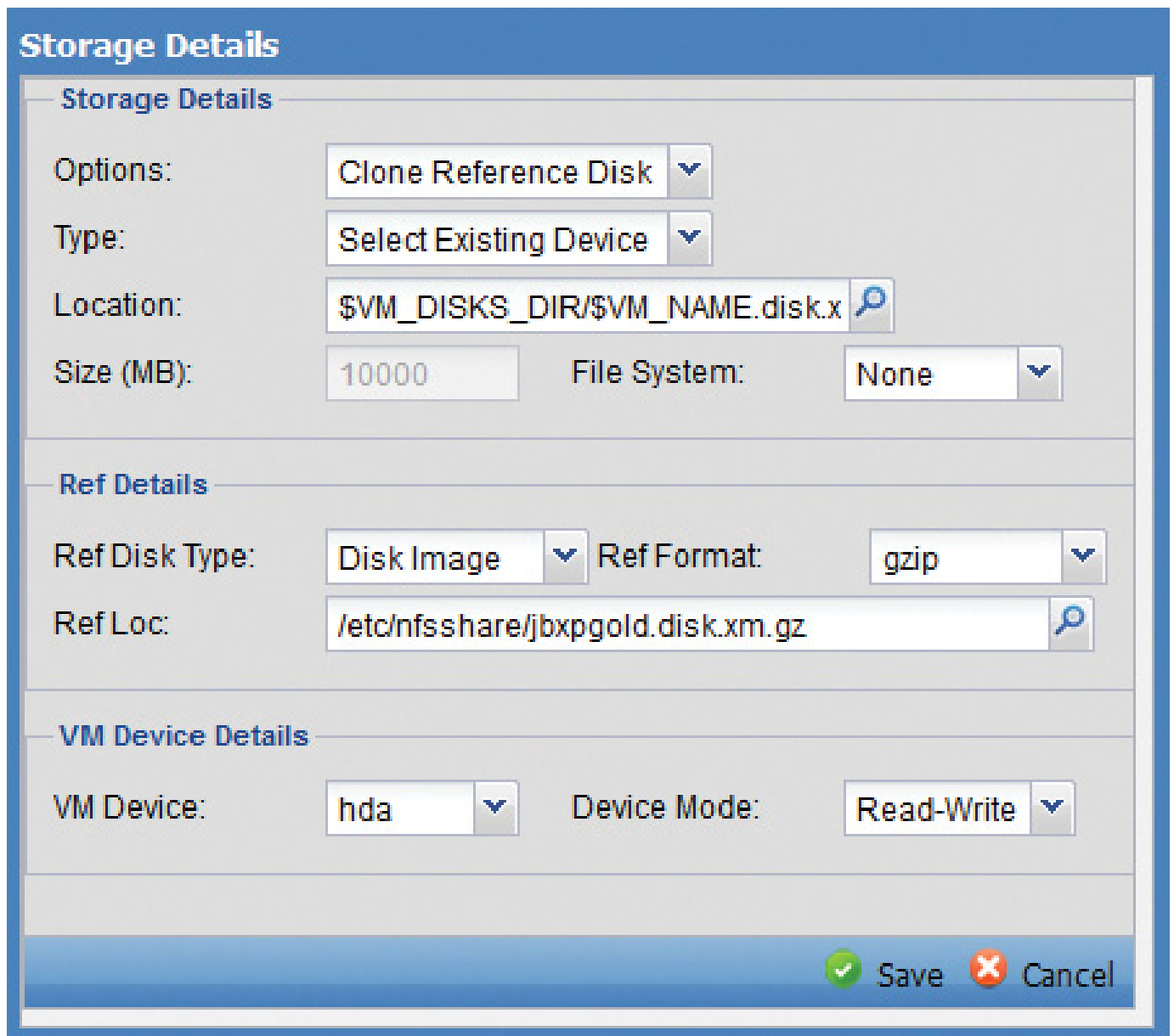


Figure 7. Provisioning Settings to Clone the Golden Image

(or “golden”) image of a Windows XP machine. This VM contains all of the settings and software needed so that I don’t need to make changes to each new VM that is spun up. After the golden image is ready, power it down in the hypervisor or ConVirt, and copy the guest’s .xm file to a separate

location. In my case, I copied it to an NFS share mounted on the ConVirt and all of my managed servers. You then need to gzip your .xm image in its new location to give it a .gz extension. Next, copy the Windows CD template by right-clicking it in the templates section and clicking on the

Create Like option.

You could create a template from scratch, but copying and modifying a built-in one is just as quick. If you have very custom settings that differ greatly from those found in the pre-built templates, that may be the way to go.

When prompted, give your template a new name. Once the new template appears in the list, right-click on it and select the Edit Settings option. Click on the Storage option and remove the existing storage defined for hda. Click on the New button at the top of the window. On the resulting window, set the Option field to Clone Reference Disk. Change the Ref. Disk Type to Disk Image and the Ref. Format to .gzip. In the Ref Loc. field, browse or enter the path to your ISO file. Change the VM Device: field to "hda". Your settings should resemble those shown in Figure 7.

To deploy a new cloned VM from this template, right-click it and select Provision. With the reference disk method, ConVirt will copy the .gz file to its destination and expand it to the desired size of the new VM. What is really nice is that you can specify a larger disk size than the one inside your golden image. On my XP VMs, the space automatically was added to the guest partition (not usually an easy task). It is a common

best practice to keep your golden image as small as possible to fit as many different size drives (virtual or physical) that you will deploy it to.

After your deployment is in place, you may find that you need to move guests to another host to balance loads between servers, to move a VM from one network site or segment to another or to perform maintenance on a host with zero downtime to running guests. VMware dominated the market for years with its vMotion feature that performs this task well. ConVirt provides this same operation.

Note that in order to migrate running guests between hosts, both hosts must have access to the same shared storage. You may run into other limitations when migrating guests, such as both hosts must have the same processor type and/or must be on the same hypervisor platform (like KVM or Xen), so plan accordingly. I was unable to determine whether this was a technical limitation or an unlocked feature in the Enterprise version of ConVirt. Either way, there are some native tools in the hypervisors that can convert foreign disk/VM types for importation into their native platform. After you have met all the prerequisites, migrating is as simple as right-clicking the guest and selecting your destination server.

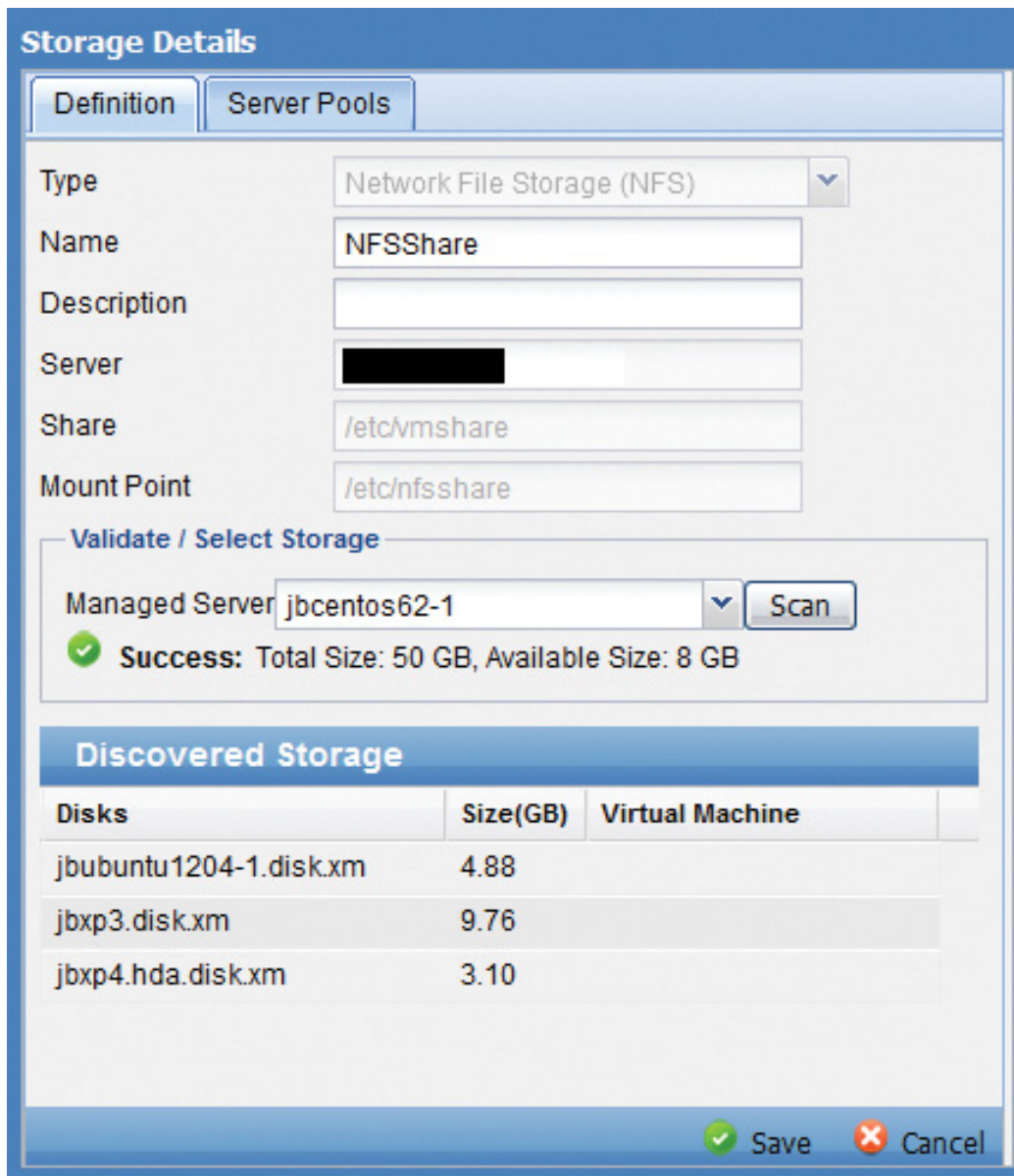


Figure 8. Shared Storage Details

You can monitor your migration task in the bottom pane of the console.

One last feature I want to mention is ConVirt’s management of shared storage, because I think it is useful (Figures 8 and 9). With the designer’s tree-based approach to organizing virtual resources, you set shared

storage at the Data Center-level and then attach it to Server Pools, which gives you the ability to mix and match your storage among the pools. Be aware that for all servers in the pool to use the storage, they must connect to the storage using the same logical path (like migration). I found this feature incredibly useful as it really simplifies assignment of any networked storage resources you have in

your environment (SAN, iSCSI or NFS). You also can set certain provisioning settings at the pool level that override those in a template. This means you can provision the same template with multiple storage options. This would be very handy if you have Server Pools in different sites or different departments,

each that should use their own storage resources.

In this article, I've touched on many of the nicer features in ConVirt, but now let me talk about some things that are missing. Before doing so, you should recognize that I am comparing apples and oranges when I talk about ConVirt and vendor-produced management tools. Even comparing the Enterprise version of ConVirt is not

wholly accurate, as ConVirt is meant to manage a heterogeneous virtual environment, whereas Microsoft and VMware are tuned to their own homogeneous platforms.

That being said, I still had a few gripes with ConVirt. The first is that it requires root access to managed

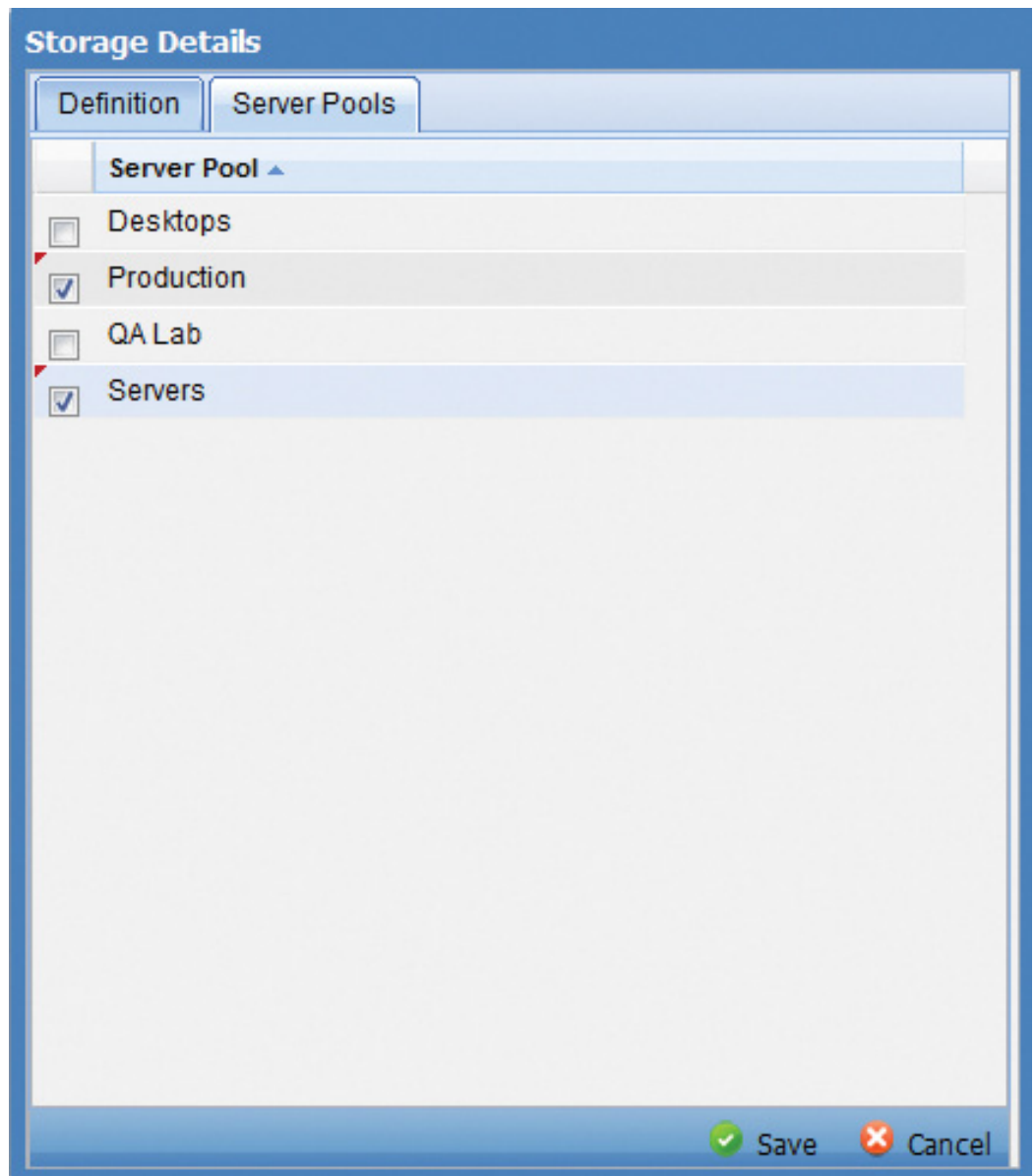


Figure 9. Server Pools That Can Use This Storage

servers to communicate with the CMS, which I am sure most admins won't be crazy about. Snapshot support also is noticeably missing from the open-source version. There is an option available for the VMs called Hibernate, but that takes a snapshot only of the running memory not the underlying

disk. The lack of snapshots bothered me only for half-a-second when I realized it is available in the Enterprise version. The last item missing from ConVirt is administrative roles. You do have the ability to create users and groups in the console, but as far as I can tell, the only thing that gets you is auditing of the tasks that take place on the CMS server. It felt like this was added into the product in its most basic form, but never fully developed.

In the end, these are minor complaints, as ConVirt provides far more utility than the few features it lacks. The software really gives you a lot of flexibility, especially with KVM, and you can't beat the price point. I'm sure those features unlocked in the Enterprise version (snapshots, high availability and spanned virtual networks) are worth the money and bring it more in line with the vendor-produced management offerings. I know how much VMware costs, and

I am sure ConVirt comes in under that. I will say that you really need to know your chops when managing different hypervisors at the same time. I am one of those admins who works with vSphere daily, and I have become accustomed to a homogeneous environment, so I really had to get under the hood of both KVM and Xen to make things go smoothly. That being said, once it is in place, I believe it is easier to administer by non-Linux IT pros or admins who need to perform day-to-day tasks in their virtual environment than virt-manager or command-line tools. Add in the ability to manage a multiplatform hypervisor environment, and the value of ConVirt is apparent. ■

Jeramiah Bowling has been a systems administrator and network engineer for more than ten years. He works for a regional accounting and auditing firm in Hunt Valley, Maryland, and holds numerous industry certifications, including the CISSP. Your comments are welcome at jb50c@yahoo.com.

Resources

Convirture's Main Site: <http://www.convirture.com>

Installation Guide/Wiki: http://www.convirture.com/wiki/index.php?title=Convirt2_Installation

KVM: <http://www.linux-kvm.org>

Xen: <http://www.xen.org>



ACHIEVEMENT UNLOCKED! MASTER OF THE WEB

1-UP YOUR ABILITIES

PHP - PYTHON - RUBY - JAVA - .NET
HTML5 - JAVASCRIPT - MOBILE

EXPAND

YOUR SKILLS WITH EXPERTS
FROM ACROSS THE GLOBE.

EXPLORE

DIVERSE TECHNOLOGIES WITH
160 PRESENTATIONS.

EXPERIENCE

THE BEST OF WEB COMMUNITY AND
CULTURE.

- **START**
REGISTER ON CONFOO.CA
BEFORE JAN. 20
FOR A DISCOUNT

FOLLOW
@CONFOOCA

SPONSORED BY:



FooLab



Linux on Azure: a Strange Place to Find a Penguin

Running Linux on Azure might sound like a penguin pitching his tent in the depths of Mordor, but this overview shows that it's a pleasant computing environment.

Andrew Fabbro

Linux enthusiasts might think the idea of running a Linux virtual machine on Microsoft's Azure service is like finding a penguin sun tanning in the Sahara. Linux in the heart of the Microsoft cloud? Isn't that just wrong on so many levels?

Why would anyone want to run Linux on Microsoft servers? For the

hobbyist, I suppose for the same reason people climb Mount Everest: because it's there. For the business user, the prospect of spinning up Linux VMs in Microsoft's fabric offers new options for collocating open-source technologies with existing Microsoft Azure services. For the cloud market in general, more competition is good news for consumers.

The Cloud Marketplace

Virtual machines in the form of virtual private servers (VPSes) have been offered for nearly a decade from a galaxy of providers, using virtualization technologies, such as Xen, Virtuozzo/OpenVZ and KVM. These providers subdivide a physical server into multiple small virtual servers. Users typically subscribe on a monthly basis, with an allotment of memory, disk and network bandwidth.

Later vendors, such as Amazon, Rackspace and now Microsoft, offer the same service with a finer-grained commitment. Users can spin up a VM (or a hundred) by the hour, pay for bandwidth by the gigabyte and utilize more advanced features, such as private networks, SAN-like storage features, offloaded database engines and so on.

All of this diversity is good news for end users. In 2002, a VPS with 128MB cost nearly \$100/month. In 2006, you could get a VPS with 512MB of RAM for \$40/month. Today, such VPSes can be found for less than \$5/month in the VPS market or for pennies per hour from cloud providers.

Microsoft Enters the Market

Amazon enjoyed early success with its Elastic Compute Cloud, and other vendors, such as Rackspace, soon

followed suit. Microsoft originally opted for a different, more complex cloud strategy. Azure was built as a “platform as a service” offering (see the Cloud Flavors sidebar) in which developers could write applications that ran in various roles and talked to Azure APIs. In theory, this allowed developers to concentrate on code and not worry about the abstracted hardware underneath.

In practice, developers were forced to write Azure-centric applications and adoption was slow. Many enterprises with mixed Windows/Linux environments found that hosting their own self-managed servers on Amazon and other cloud environments was more attractive than spending time porting and debugging their applications.

In 2012, Microsoft added “infrastructure as a service” (virtual machines) offerings to its lineup, allowing users to run and administer Windows and Linux virtual machines they directly control.

Azure virtual machines are still in “Community Preview”, which is Microsoft lingo for “Beta”. Support is limited to forums, and as you’ll see, some sandpapering of the offering still is needed. However, after using the service for a couple months, I find Linux on Azure to be stable and easy

Cloud Flavors

Cloud-based services come in several different forms, depending on what's being abstracted and provisioned.

- **Software as a Service (SaaS):** the provider runs an application and exposes an interface to subscribers. This is simply using a Web-based application. Examples include Salesforce.com, NetSuite and 37 Signals' Basecamp.
- **Platform as a Service (PaaS):** the provider offers a platform that users can use to build applications. Subscribers write and provide code, which runs on abstracted hardware and software services. Examples include Microsoft's Azure original offerings, Oracle's Fusion cloud or Google's App Engine.
- **Infrastructure as a Service (IaaS):** the provider delivers virtual machines and other infrastructure pieces that users can configure as they like. Examples include Amazon's EC2, Rackspace Cloud, Google Compute and Microsoft's Azure. Virtual Private Servers also are IaaS offerings with a different financial model.

to use, and it performs well. At the time of this writing, Microsoft has not set a date for General Availability.

Comparing Azure to Amazon EC2

Azure's chief competition is Amazon EC2, and it's not hard to see that Microsoft patterned its IaaS offering after its rival's success.

Like EC2, Azure is priced by the hour, and the rates are similar. While

in Community Preview, pricing is slightly discounted compared to expected General Availability pricing. Actual price comparisons for hourly VMs depend on how long of a term commitment you make. For example, Amazon offers both spot instances and prepaid reserved instances, while Microsoft also discounts longer-term commitments. Storage and bandwidth pricing are very similar. In general,

running a VM on Azure will cost you about the same as EC2, which is probably just what Microsoft planned.

EC2 offers features Microsoft has not caught up to yet. For example, the underlying storage virtual machine disk (Elastic Block Storage) can be snapshotted to S3 storage. However, many Amazon services have parallels in the Azure world. For example, Amazon's SimpleDB is analogous to Azure's Tables. Both vendors offer complex networking features, caching, monitoring and Content Delivery Network options. In either environment, a VM can be sited in the Americas, Europe or Asia, with global CDN nodes.

The Azure value proposition is not "we are a better cloud" but rather "you can do EC2-like things here alongside your Azure platforms". For shops that have deep Microsoft deployments, were early adopters of Azure or want to develop applications that move into and out of Azure, the new IaaS offerings will be appealing.

Taking Azure for a Spin

Using Microsoft Azure requires a free Windows Live account, as well as a credit card to open a charge account. If your employer participates in the Microsoft Developer Network (MSDN) program,

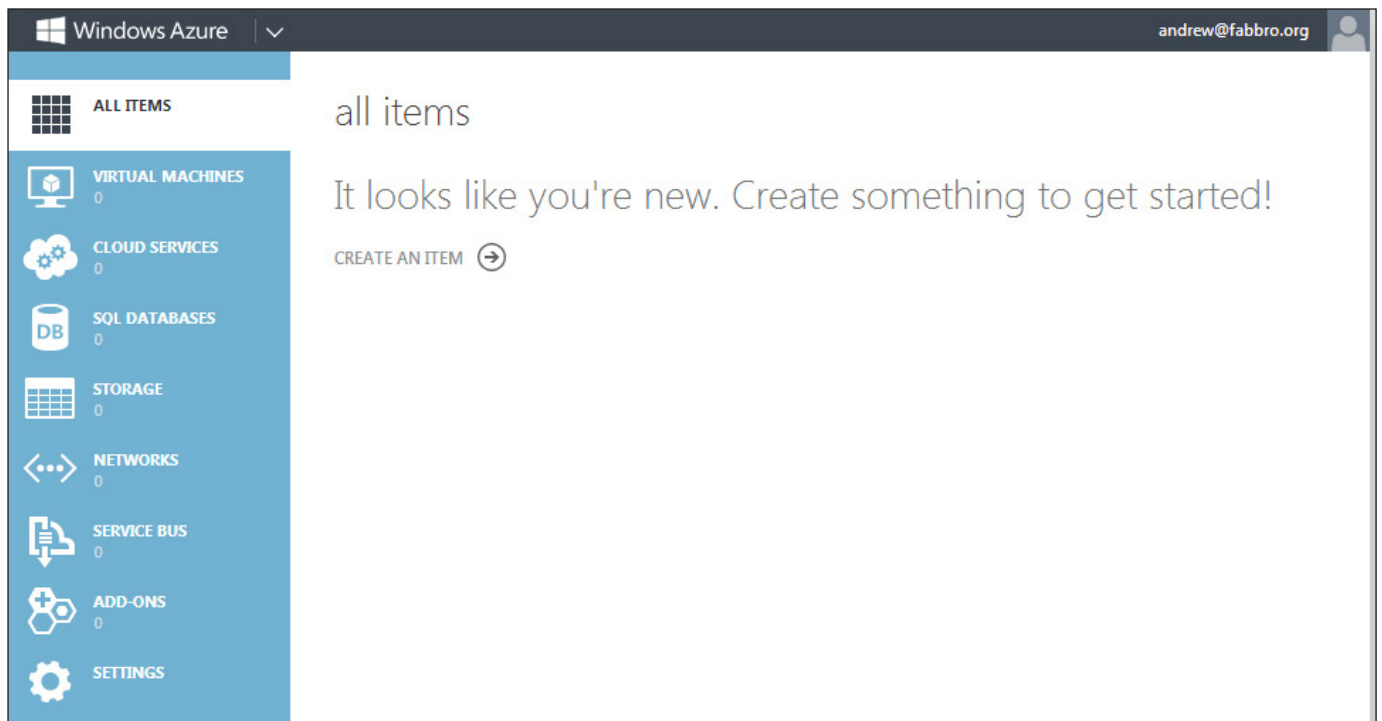


Figure 1. The Azure management portal is easy to use and attractively designed.

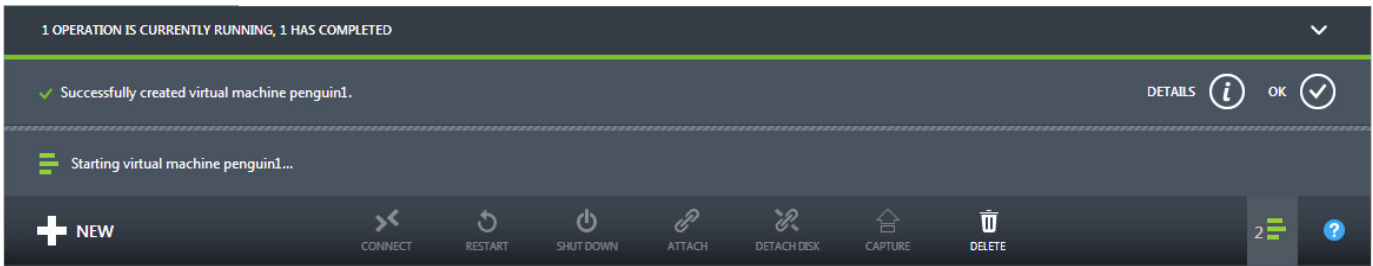


Figure 2. The Azure portal displays task message and status.

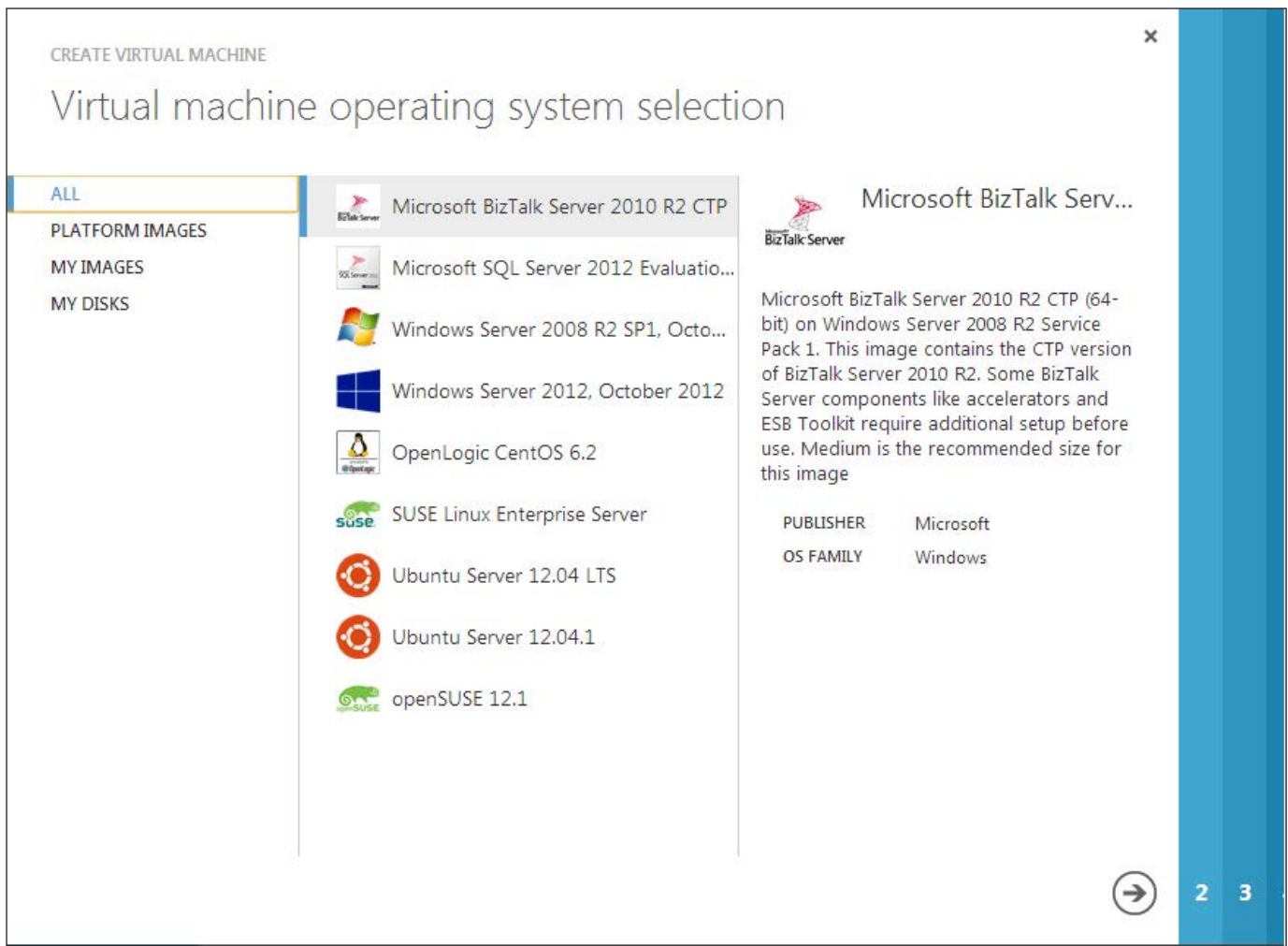


Figure 3. Five different Linux options are offered when creating a VM.

you already may be entitled to a free quantity of Azure services every month. Once your account is set up, you can head to the management

portal and start adding services.

The Azure control panel is, quite simply, gorgeous. Perhaps pretty controls are not a big selling point

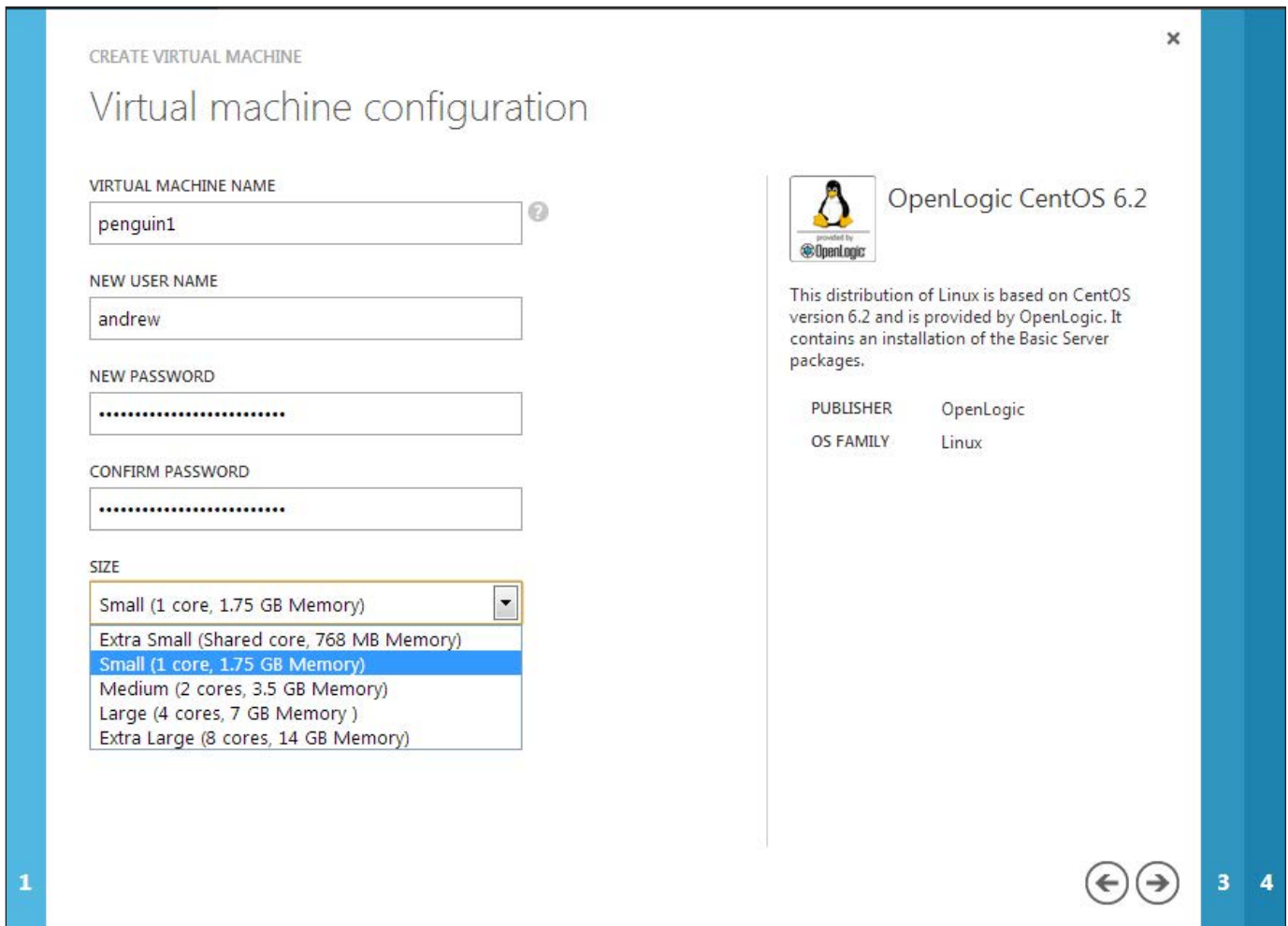


Figure 4. Setting Up penguin1 in Azure

for a cloud service, but the Azure interface is marvelously interactive. As you set up services, messages on the status of operations appear at the bottom of the screen asynchronously. Performance graphs and history are integrated into the display, and the panel feels much more like a desktop app than a “click-submit-and-wait” Web interface.

Azure offers several flavors of Linux: CentOS 6.2, Ubuntu 12.04,

SUSE Linux Enterprise Server and OpenSUSE 12.1. It’s possible to roll your own image and upload it, but this requires working with Microsoft’s Hyper-V server product, which is something the average Linux user is unlikely to have handy.

For this article, I create a CentOS 6.2 VM called “penguin1”. A DNS name is created automatically for the VM in the cloudapp.net domain, which then can be CNAME’d if you

ADD ENDPOINT ×

Specify the details of the endpoint

NAME

PROTOCOL

PUBLIC PORT

PRIVATE PORT

1 ← ✓

Figure 5. Adding a Network Endpoint to Set Up a Web Server

own your own domain.

VMs are not directly exposed to the Internet, but rather are given 10.x IP addresses. Inside the Azure panel, users then can configure endpoints to open firewall ports and map them as they like. For example, to set up a Web server, it's necessary to create a port 80 (and perhaps 443) endpoint, which can be mapped to any port desired on the VM.

This network firewall is a nice

security feature. By default, only port 22 (SSH) is configured. If you intend to change your default SSH port (as often is done to prevent script-kiddie scanning), you'll need to change the endpoint in the Azure management portal as well. You also have the option of changing it in the management portal and mapping it back to 22 on the VM.

Creating "From Gallery" gives you the most options for creation. After supplying basic information, such as

name and size, provisioning begins immediately and takes about ten minutes to complete in my experience.

What's the Root Password!?!?

Users of other VPS systems may wonder where they specify the root password. The answer is simple: you don't. Whatever account you specify will be given sudo authority to su to root:

```
[andrew@penguin1 ~]$ sudo su -  
[sudo] password for andrew:  
[root@penguin1 ~]#
```

From that point, you can set the root password if you want with `passwd`.

Storage

Linux virtual machines have three types of storage.

First, each is given a 30GB root volume. Storage is locally redundant and optionally can be made geographically redundant for about a 33% cost increase. Unfortunately, short of creating your own template, there is no way to modify this 30GB configuration if you prefer a different filesystem layout or want a smaller monthly storage bill. Root volume data persists across reboots and is a permanent

BLOB in Azure storage. So if you delete a VM, it's possible to retain its root volume and later mount it up on another system or use it as the root volume for a new VM.

You also can create as many other volumes as you'd like. These live as BLOBs in Azure storage and are persistent. They function much like SAN volumes, allowing you to create and attach them to one VM, then later unmount and attach to another. Unfortunately, there is no way to resize these volumes, which is a disappointing limitation. While you can create a larger volume, move data and delete the old volume, this obviously is not a scalable approach.

When you create a new disk and attach it to the VM, it appears as a new SCSI device that you can `mkfs` and mount. After creating a new 20GB disk in the Azure portal, `penguin1`'s `dmesg` shows:

```
scsi 4:0:0:0: Direct-Access Msft Virtual Disk 1.0 PQ: 0 ANSI: 4  
sd 4:0:0:0: Attached scsi generic sg3 type 0  
sd 4:0:0:0: [sd] 41943040 512-byte logical blocks: (21.4 GB/20.0 GiB)  
sd 4:0:0:0: [sd] Write Protect is off  
sd 4:0:0:0: [sd] Mode Sense: 0f 00 10 00  
sd 4:0:0:0: [sd] Write cache: enabled, read cache: enabled,  
    supports DPO and FUA  
sdc: unknown partition table  
sd 4:0:0:0: [sd] Attached SCSI disk
```

Finally, virtual machines also come with a variable amount of truly local storage. This storage does not live in the Azure cloud but rather is provisioned on the actual physical node.

Now you can fdisk, mkfs and mount:

```
[root@penguin1 ~]# fdisk /dev/sdc
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2610, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2610, default 2610):
Using default value 2610
```

```
Command (m for help): w
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
Syncing disks.
[root@penguin1 ~]# mkfs.ext4 /dev/sdc1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1310720 inodes, 5241198 blocks
```

```
262059 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
160 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736,
    1605632, 2654208, 4096000
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 21 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

```
[root@penguin1 ~]# mkdir /data
[root@penguin1 ~]# mount /dev/sdc1 /data
[root@penguin1 ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VolGroup-lv_root	28G	2.0G	24G	8%	/
tmpfs	872M	0	872M	0%	/dev/shm
/dev/sda1	485M	86M	374M	19%	/boot
/dev/sdb1	69G	180M	66G	1%	/mnt/resource
/dev/sdc1	20G	172M	19G	1%	/data

As you might expect, if I were to delete this disk in the Azure portal, the operation would fail unless I first unmounted it.

Finally, virtual machines also come with a variable amount of truly local storage. This storage does not live in the Azure cloud but rather is provisioned on the actual physical node. If there is a hardware or other fault in the Azure fabric and your VM migrates to a new host, data stored in this space is lost. This filesystem is meant to be used for state information, scratch data and other temporary files. On Linux images, it shows up as `/mnt/resource`. Small VMs receive a 70GB filesystem, and the size increases with VM size, up to 800GB for Extra Large VMs.

I have experienced a hardware fault on Azure. My VM shut down, migrated to a new hardware host and booted up on the new server. Services that I'd defined to start up came up normally, all network endpoints were moved correctly, disks I'd mounted were remounted, and the VM worked fine. However, the scratch data I had in `/mnt/resource` was lost, and the filesystem was empty, as designed.

Rough Edges

The CentOS image could use some improvement. Provided by OpenLogic, I've noted some things that make me scratch my head:

- A swap partition is configured, but does not appear in `/etc/fstab`. Because there isn't much advantage to creating a swap partition but not using it, this is presumably an oversight.
- Some default services run without justification. For example, why is CUPS needed? I'm unlikely to print in the cloud. Given that RAID redundancy is provided by the Azure storage layer and software RAID is not needed, why is `mdmonitor` set to run at boot?
- I periodically receive crash reports from `fprintd`. Removing this service is straightforward, but why the CentOS image is created to support fingerprint biometric authentication in a virtual, cloud-based environment mystifies me.
- `iptables` is enabled with a single rule to accept `bootp`, though the `INPUT` chain has a default `ACCEPT` policy anyway.

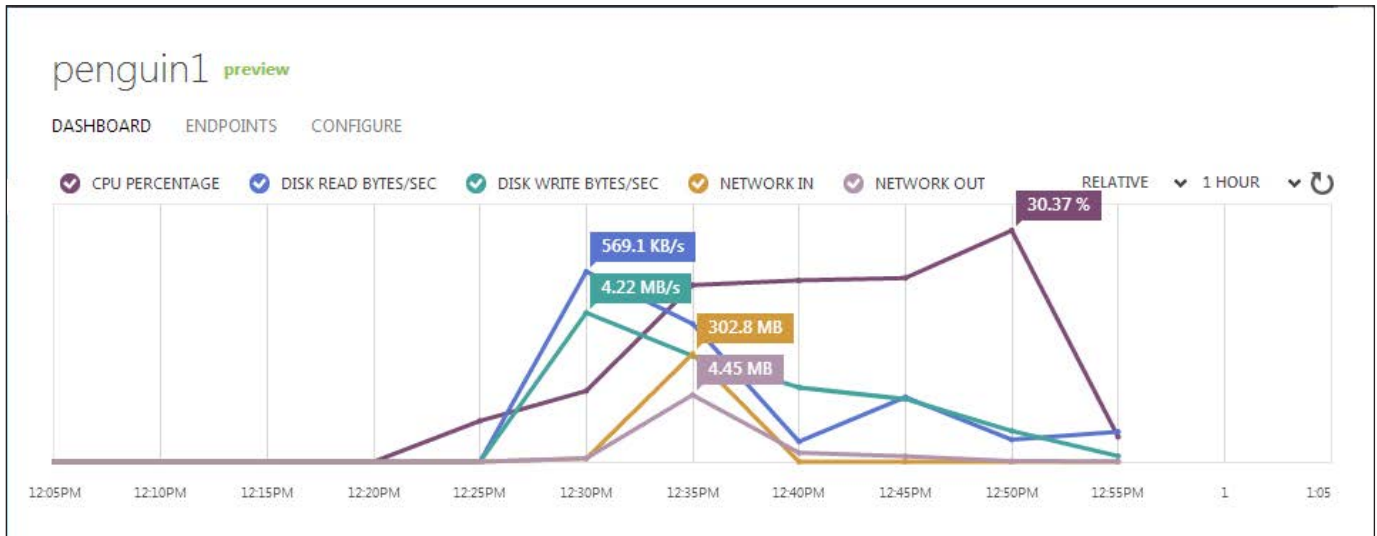


Figure 6. The Integrated Display of Performance Metrics in the Azure Portal

Hopefully, these things will be corrected as Linux settles into Azure.

Performance

In general, I/O performance is excellent. Measuring with `ioping`, I found the `/data` disk I created earlier averaged 6ms latency, while `/mnt/resource` showed a zippy 0.4ms. The root disk was a slower 18ms, but as Microsoft explains in its documentation, it optimizes the I/O performance on volumes tagged “OS Disk” differently:

The operating system disk and data disk has a host caching setting (sometimes called host-cache mode) that enables improved performance under

some circumstances. However, these settings can negatively affect performance in other circumstances, depending on the application. Host caching is OFF by default for both read operations and write operations for data disks. Host-caching is ON by default for read and write operations for operating system disks. As noted, these should work best in most cases. However, your mileage may vary. We recommend you place data intensive operations on a data disk separate from the OS disk.

Compute performance depends on the VM size you select. Under the covers, Microsoft is using AMD gear,

as evidenced by /proc/cpuinfo:

```
processor      : 1
vendor_id     : AuthenticAMD
cpu family    : 16
model         : 8
model name    : AMD Opteron(tm) Processor 4171 HE
stepping      : 1
cpu MHz       : 2094.702
cache size    : 512 KB
```

Summary

Some may say the best thing about Linux in Azure is that it gives competition to Amazon and Google. For the end

user, performance and pricing are very similar, while the enterprise administrator may find the ability to collocate Linux alongside Windows and Azure-specific deployments to be advantageous. While running Linux in the Azure cloud might seem like a penguin pitching his tent in Mordor, one more quality player in the cloud space is good news for all. ■

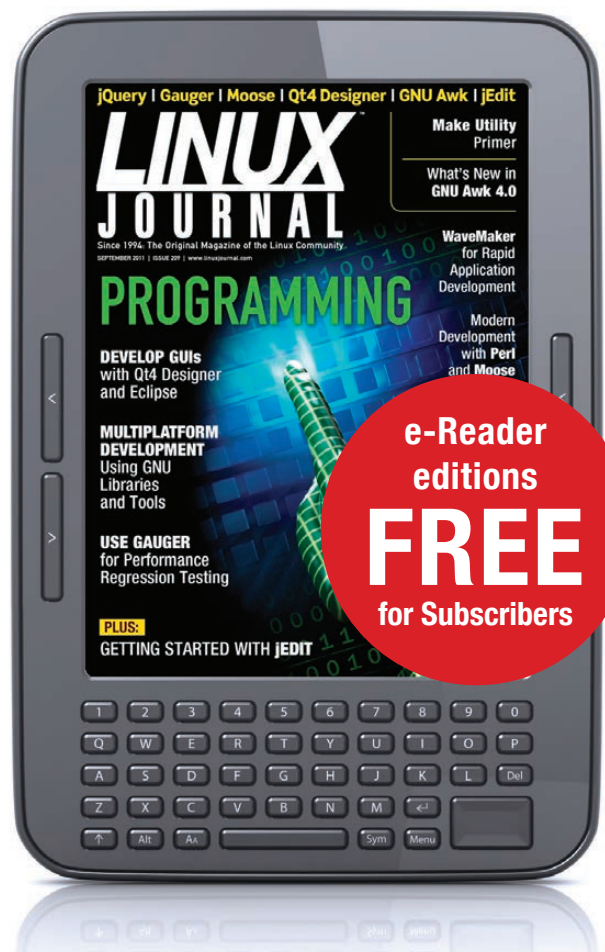
Andrew Fabbro is a senior technologist living in the Portland, Oregon, area. He's used Linux since Slackware came on floppies and presently works for Con-way, a Fortune 500 transportation company.

LINUX JOURNAL

on your
e-Reader

Customized
Kindle and Nook
editions
now available

[LEARN MORE](#)



FABRIC

a System Administrator's Best Friend

Do you routinely make changes to more than a dozen machines at a time? Read this article to find out about a tool to make that task much easier.

ADRIAN HANNAH

I'll be honest. Even though this library is fully five years old, I hadn't heard of Fabric until about six months ago. Now I can't imagine not having it in my digital tool belt. Fabric is a Python library/tool that is designed to use SSH to execute system administration and deployment tasks on one or more remote machines. No more running the same task, machine by machine, to make one change across the board. It is a simple fire-and-forget tool that will make your life so much simpler. Not only can you run simple tasks via SSH on multiple machines, but since you're using Python code

most prolific way to install Fabric is using pip (or `easy_install`). On most systems, you can use your system's package manager (`apt-get`, `install`, and so on) to install it (the package either will be `fabric` or `python-fabric`). If you're feeling froggy, you can check out the Git repository and hack away at the source code.

Once installed, you will have access to the `fab` script from the command line.

Operations

The Fabric library is composed of nine separate operations that can be used

IT IS A SIMPLE FIRE-AND-FORGET TOOL THAT WILL MAKE YOUR LIFE SO MUCH SIMPLER.

to execute items, you can combine it with any arbitrary Python code to make robust, complex, *elegant* applications for deployment or administration tasks.

Installation

Fabric requires Python 2.5 or later, the `setuptools` packaging/installation library, the `ssh` Python library, and SSH and its dependencies. For the most part, you won't have to worry about any of this, because Fabric can be installed easily through various package managers. The easiest, and

in conjunction to achieve your desired effect. Simply insert these functions into your `fabfile` and off you go:

- `get(remote_path, local_path=None)` — `get` allows you to pull files from the remote machine to your local machine. This is like using `rsync` or `scp` to copy a file or files from many machines. This is super effective for systematically collecting log files or backups in a central location. The remote path is the path of the file

on the remote machine that you are grabbing, and the local path is the path to which you want to save the file on the local machine. If the local path is omitted, Fabric assumes you are saving the file to the working directory.

- `local(command, capture=False)` — the local function allows you to take action on the local host in a similar fashion to the Python subprocess module (in fact, local is a simplistic wrapper that sits on top of the subprocess module). Simply supply the command to run and, if needed, whether you want to capture the output. If you specify `capture=True`, the output will be returned as a string from local; otherwise, it will be output to STDOUT.
- `open_shell(command=None)` — this function is mostly for debugging purposes. It opens an interactive shell on the remote end, allowing you to run any number of commands. This is particularly helpful if you are running a series of particularly complex commands and it doesn't seem to be working on some of your machines.
- `prompt(text, key=None, default='', validate=None)` — in the case when you need to supply a value, but don't want to specify it on the command line for whatever reason, `prompt` is the ideal way to do this. I have a fabfile I use to add/remove/check the status of software on all of the servers I maintain, and I use this in the script for when I forget to specify what software I'm working on. This prompt will appear for each host you specify, so make sure you account for that!
- `put(local_path, remote_path, use_sudo=False, mirror_local_mode=False, mode=None)` — this is the opposite command of `get`, although you are given more options when putting to a remote system than getting. The local path can be a relative or absolute file path, or it can be an actual file object. If either `local_path` or `remote_path` is left blank, the working directory will be used. If `use_sudo=True` is specified, Fabric will put the file in a temporary location on the remote machine, then use `sudo` to move it from the temporary location to the specified location. This is particularly handy when moving system files like `/etc/resolv.conf` or the like that

can't be moved by a standard user and you have root login turned off in SSH. If you want the file mode preserved through the copy, use `mirror_local_mode=True`; otherwise, you can set the mode using `mode`.

- `reboot(wait=120)` — `reboot` does exactly what it says: reboots the remote machine. By default, `reboot` will wait 120 seconds before attempting to reconnect to the machine to continue executing

`quiet=False, warn_only=False, stdout=None, stderr=None)` — this and `sudo` are the two most used functions in Fabric, because they actually execute commands on the remote host (which is the whole point of Fabric). With `run`, you execute the specified command as the given user. `run` returns the output from the command as a string that can be checked for a `failed`, `succeeded` and `return_code` attribute. `shell` controls whether a shell interpreter is created for the

THIS IS PARTICULARLY HANDY WHEN MOVING SYSTEM FILES LIKE `/etc/resolv.conf` OR THE LIKE THAT CAN'T BE MOVED BY A STANDARD USER AND YOU HAVE ROOT LOGIN TURNED OFF IN SSH.

any following commands.

- `require(*keys, **kwargs)` — `require` forces the specified keys to be present in the shared environment dict in order to continue execution. If these keys are not present, Fabric will abort. Optionally, you can specify `used_for` to indicate what the key is used for in this particular context.
- `run(command, shell=True, pty=True, combine_stderr=True,`

`command`. If turned off, characters will not be escaped automatically in the command. Passing `pty=False` causes a pseudo-terminal not to be created while executing this command; this can have some benefit if the command you are running has issues interacting with the pseudo-terminal, but otherwise, it will be created by default. If you want `stderr` from the command to be parsable separately from `stdout`, use `combine_stderr=False` to indicate that. `quiet=True` will

cause the command to run silently, sending no output to the screen while executing. When an error occurs in Fabric, typically the script will abort and indicate as such. You can indicate that Fabric need not abort if a particular command errors using the `warn_only` argument. Finally, you can redirect where the remote `stderr` and `stdout` redirect to on the local side. For instance, if you want the `stderr` to pipe to `stdout` on the local end, you could indicate that with `stderr=sys.stdout`.

- `sudo(command, shell=True, pty=True, combine_stderr=True, user=None, quiet=False, warn_only=False, stdout=None, stderr=None, group=None)` — `sudo` works precisely like `run`, except that it will elevate privileges prior to executing the command. It basically works the same as if you'd run the command using `run`, but prepended `sudo` to the front of command. `sudo` also takes `user` and `group` arguments, allowing you to specify which user or group to run the command as. As long as the original user has the permissions to escalate for that particular user/group and command, you are good to go.

The Basics

Now that you understand the groundwork of Fabric, you can start putting it to use. For this article, I explain how to make a simple fabfile for the purpose of installing/removing software on your machines. First, you need what is called a fabfile. The fabfile contains all of your Fabric functions. By default, it needs to be named `fabfile.py` and be in the working directory, but as mentioned previously, you can specify the fabfile from the command line if need be. So, open your fabfile and start it with `from fabric.api import *` to include all the Fabric functionality. Then define all of your functions. Let's start with installing some software:

```
def install(pkg=None):
    if pkg is not None:
        env["pkg"] = pkg
    elif pkg is None and env.get("pkg") is None:
        env["pkg"] = prompt("Which package? ")
    sudo('yum install -y %s' % env["pkg"])
```

You then can install a package via `yum` on all of your machines by running:

```
$ fab --hosts=host1,host2,host3 install
```

Big Data gets real at Big Data TechCon!

The **HOW-TO** conference for Big Data and IT Professionals



Discover how to master Big Data from real-world practitioners – instructors who work in the trenches and can teach you from real-world experience!

Come to Big Data TechCon to learn the best ways to:

- Collect, sort and store massive quantities of structured and unstructured data
- Process real-time data pouring into your organization
- Master Big Data tools and technologies like Hadoop, Map/Reduce, NoSQL databases, and more
- Learn HOW TO integrate data-collection technologies with analysis and business-analysis tools to produce the kind of workable information and reports your organization needs
- Understand HOW TO leverage Big Data to help your organization today

Over 50
how-to
practical classes
and workshops
to choose
from!

BigData TECHCON

April 8-10, 2013
Boston, MA

www.BigDataTechCon.com

Register Early and SAVE!

A BZ Media Event

Big Data TechCon™ is a trademark of BZ Media LLC.

A BRIEF WORD ON APPLICATION DEPLOYMENT

Fabric also is used in development teams to deploy new code to production. It is actually used in a fairly similar fashion to how system administrators use it (copy files, run a few commands and so on), just in a very specific manner. Because of how automated Fabric is, it's easy to incorporate it into a continuous integration cycle and even fully automate your deployment process.

Then, you'll be prompted for the package to install only once. Alternatively, since you indicated an optional parameter of `pkg`, you can indicate that from the command line so you won't be prompted on execution, like this:

```
$ fab --hosts=host1,host2,host3 install:pkg=wormux
```

or:

```
$ fab --hosts=host1,host2,host3 install:wormux
```

Also note that you are prompted for the password for both SSH and sudo only once. Fabric stores this in memory and reuses it, if possible, for every other machine. Congratulations! You've just successfully created your first Fabric script. It's as simple as that!

Tips and Tricks

I've picked up some neat tricks since I've started with Fabric. First, you generally never see a Fabric command as simple as what is above. When fully automated, it looks more like this:

```
$ fab --skip-bad-hosts -u user -p 12345 -i ~/.ssh/id_dsa --warn-only
└--hosts=host1,host2,host3,host4,host5,host6,host7,host8,host9,host10
└--parallel --pool-size=20 install:pkg=wormux
```

Who wants to type that out every time they want to run a command? No one! That's why aliasing almost all of that is so convenient and efficient. Add the following to your `.bashrc` file:

```
alias f="fab --skip-bad-hosts -u user -p 12345 -i ~/.ssh/id_dsa
└--warn-only
└--hosts=host1,host2,host3,host4,host5,host6,host7,host8,host9,host10
└--parallel"
```

Then, all you have to do each time you want to run Fabric is this:

```
$ f install:pkg=wormux
```

Even using this technique, your alias can become cumbersome if you have more than a few machines you commonly administer. A simple solution to that is to add this function to your fabfile:

```
def set_hosts():
    env.hosts = open('hosts', 'r').readlines()
```

Then, put all your hostnames in a file called hosts in the same directory as your fabfile, and modify your alias to look like this:

```
alias f="fab --skip-bad-hosts -u user -p 12345 -i ~/.ssh/id_dsa
↳--warn-only --parallel set_hosts"
```

This is particularly convenient if you have a variety of fabfiles that you use on different groups of machines, or in different contexts.

There are occasions when you need to execute certain commands from within a specific directory. Because each command is a discrete and non-persistent connection to the machine, this is not inherently simple. However, simply by enclosing the necessary

commands in a with statement, you have a solution:

```
with cd("~/gitrepo"):
    run('git add --all')
    run('git commit -m "My super awesome automated
↳commit script for `date`"')
```

More Information

There are several ways to get help with Fabric. The most effective is to use the fab-file mailing list (<http://lists.nongnu.org/mailman/listinfo/fab-user>). The developers are generally very prompt in responding. There is also a Fabric Twitter account @pyfabric (<http://www.twitter.com/pyfabric>) where Fabric news and announcements are released. You can submit and view bugs through the Fabric Github page (<https://github.com/fabric/fabric/issues>). Of course, you also can't discount the #fabric channel on Freenode, where you can connect with the community and get some quick answers. Finally, you always can browse the documentation hosted at <http://www.fabfile.org>. ■

Adrian Hannah has spent the last 15 years bashing keyboards to make computers do what he tells them. He currently works as a Senior System Administrator for a Web startup in New York City. He is a jack of all trades and a master of none. Find out more at <http://about.me/adrianhannah>.

COMMAND-LINE ARGUMENTS

- `-a, --no_agent` — sets `env.no_agent` to True, forcing your SSH layer not to talk to the SSH agent when trying to unlock private key files.
- `-A, --forward-agent` — sets `env.forward_agent` to True, enabling agent forwarding.
- `--abort-on-prompts` — sets `env.abort_on_prompts` to True, forcing Fabric to abort whenever it would prompt for input.
- `-c RCFILE, --config=RCFILE` — sets `env.rcfile` to the given file path, which Fabric will try to load on startup and use to update environment variables.
- `-d COMMAND, --display=COMMAND` — prints the entire docstring for the given task, if there is one. It does not currently print out the task's function signature, so descriptive docstrings are a good idea. (They're always a good idea, of course, just more so here.)
- `--connection-attempts=M, -n M` — sets the number of times to attempt connections. Sets `env.connection_attempts`.
- `-D, --disable-known-hosts` — sets `env.disable_known_hosts` to True, preventing Fabric from loading the user's SSH `known_hosts` file.
- `-f FABFILE, --fabfile=FABFILE` — the fabfile name pattern to search for (defaults to `fabfile.py`), or alternately an explicit file path to load as the fabfile (for example, `/path/to/my/fabfile.py`).
- `-F LIST_FORMAT, --list-format=LIST_FORMAT` — allows control over the output format of `--list.short` is equivalent to `--shortlist`; `normal` is the same as simply omitting this option entirely (the default), and `nested` prints out a nested namespace tree.
- `-g HOST, --gateway=HOST` — sets `env.gateway` to HOST host string.
- `-h, --help` — displays a standard help message with all possible options and a brief overview of what they do, then exits.
- `--hide=LEVELS` — a comma-separated list of output levels to hide by default.
- `-H HOSTS, --hosts=HOSTS` — sets `env.hosts` to the given comma-delimited list of host strings.
- `-x HOSTS, --exclude-hosts=HOSTS` — sets `env.exclude_hosts` to the given comma-delimited list of host strings to keep out of the final host list.
- `-i KEY_FILENAME` — when set to a file path, will load the given file as an SSH identity file (usually a private key). This option may be repeated multiple times. Sets (or appends to) `env.key_filename`.
- `-I, --initial-password-prompt` — forces a password prompt at the start of the session (after fabfile load and option parsing, but before executing any tasks) in order to pre-fill `env.password`. This is useful for fire-and-forget runs (especially parallel sessions, in which runtime input is not possible) when setting the password via `--password` or by setting `env.password` in your fabfile is undesirable.
- `-k` — sets `env.no_keys` to True, forcing the SSH layer not to look for SSH private key files in one's home directory.

- `--keepalive=KEEPALIVE` — sets `env.keepalive` to the given (integer) value, specifying an SSH keepalive interval.
- `--linewise` — forces output to be buffered line by line instead of byte by byte. Often useful or required for parallel execution.
- `-l, --list` — imports a fabfile as normal, but then prints a list of all discovered tasks and exits. Will also print the first line of each task's docstring, if it has one, next to it (truncating if necessary).
- `-p PASSWORD, --password=PASSWORD` — sets `env.password` to the given string; it then will be used as the default password when making SSH connections or calling the sudo program.
- `-P, --parallel` — sets `env.parallel` to True, causing tasks to run in parallel.
- `--no-pty` — sets `env.always_use_pty` to False, causing all run/sudo calls to behave as if one had specified `pty=False`.
- `-r, --reject-unknown-hosts` — sets `env.reject_unknown_hosts` to True, causing Fabric to abort when connecting to hosts not found in the user's SSH `known_hosts` file.
- `-R ROLES, --roles=ROLES` — sets `env.roles` to the given comma-separated list of role names.
- `--set KEY=VALUE,...` — allows you to set default values for arbitrary Fabric env vars. Values set this way have a low precedence. They will not override more specific env vars that also are specified on the command line.
- `-s SHELL, --shell=SHELL` — sets `env.shell` to the given string, overriding the default shell wrapper used to execute remote commands.
- `--shortlist` — similar to `--list`, but without any embellishment—just task names separated by newlines with no indentation or docstrings.
- `--show=LEVELS` — a comma-separated list of output levels to be added to those that are shown by default.
- `--ssh-config-path` — sets `env.ssh_config_path`.
- `--skip-bad-hosts` — sets `env.skip_bad_hosts`, causing Fabric to skip unavailable hosts.
- `--timeout=N, -t N` — sets connection timeout in seconds. Sets `env.timeout`.
- `-u USER, --user=USER` — sets `env.user` to the given string; it then will be used as the default user name when making SSH connections.
- `-V, --version` — displays Fabric's version number, then exits.
- `-w, --warn-only` — sets `env.warn_only` to True, causing Fabric to continue execution even when commands encounter error conditions.
- `-z, --pool-size` — sets `env.pool_size`, which specifies how many processes to run concurrently during parallel execution.

Making Linux and Android Get Along (It's Not as Hard as It Sounds)

Android devices don't come with an "installation CD" for Linux, but by installing a few tools, you won't need those coasters-to-be anyway!

AARON PETERS

Many free software fans, if they were like me, breathed a collective sigh of relief when the Android operating system hit the market. Before receiving my first smartphone (a Samsung Blackjack running Windows Mobile 5.5, I believe, that I had to update to through a torturous combination of installing Windows XP on a partition, installing the phone drivers, then running an update program), I was a steadfast "PDA-and-cell" guy who proudly carried both devices on my belt like a pair of six-shooters. But that

Blackjack showed me how nice it is to carry one device, and since receiving my first Android device (an original Droid I still use to this day), I can't imagine using a device with another mobile OS. Linux kernel, Java-based apps—these are all right up my alley.

But, like many great consumer Linux products (I'm talking to *you*, Sharp Zaurus), manufacturers assume in nearly every case that your "other" computer will run Windows. Now, it's easy enough to install Windows either on a separate partition to dual-boot or in a VM to run within Linux. But

this is a bit like killing the proverbial fly with a bazooka. Web-based applications and “the cloud” alleviate some of these difficulties, yet it’s still not an “out-of-the-box-after-a-quick-install-from-CD” process like it is for Windows users.

The good news is, with the installation or configuration of a few programs, it’s pretty easy to get your Android device (all the steps in this article are equally applicable to phones and tablets unless stated otherwise) to play nice with your

Linux boxen. In this article, I focus on files and a few approaches for making sure you always have an up-to-date copy of that spreadsheet or source file on your mobile device.

In the Cloud

The cloud computing movement has done a great deal to promote platform agnosticism, from consistent (Web-based) UIs to cross-platform APIs that allow applications to synchronize data. And with most users being

The White Paper Library

on LinuxJournal.com



www.linuxjournal.com/whitepapers

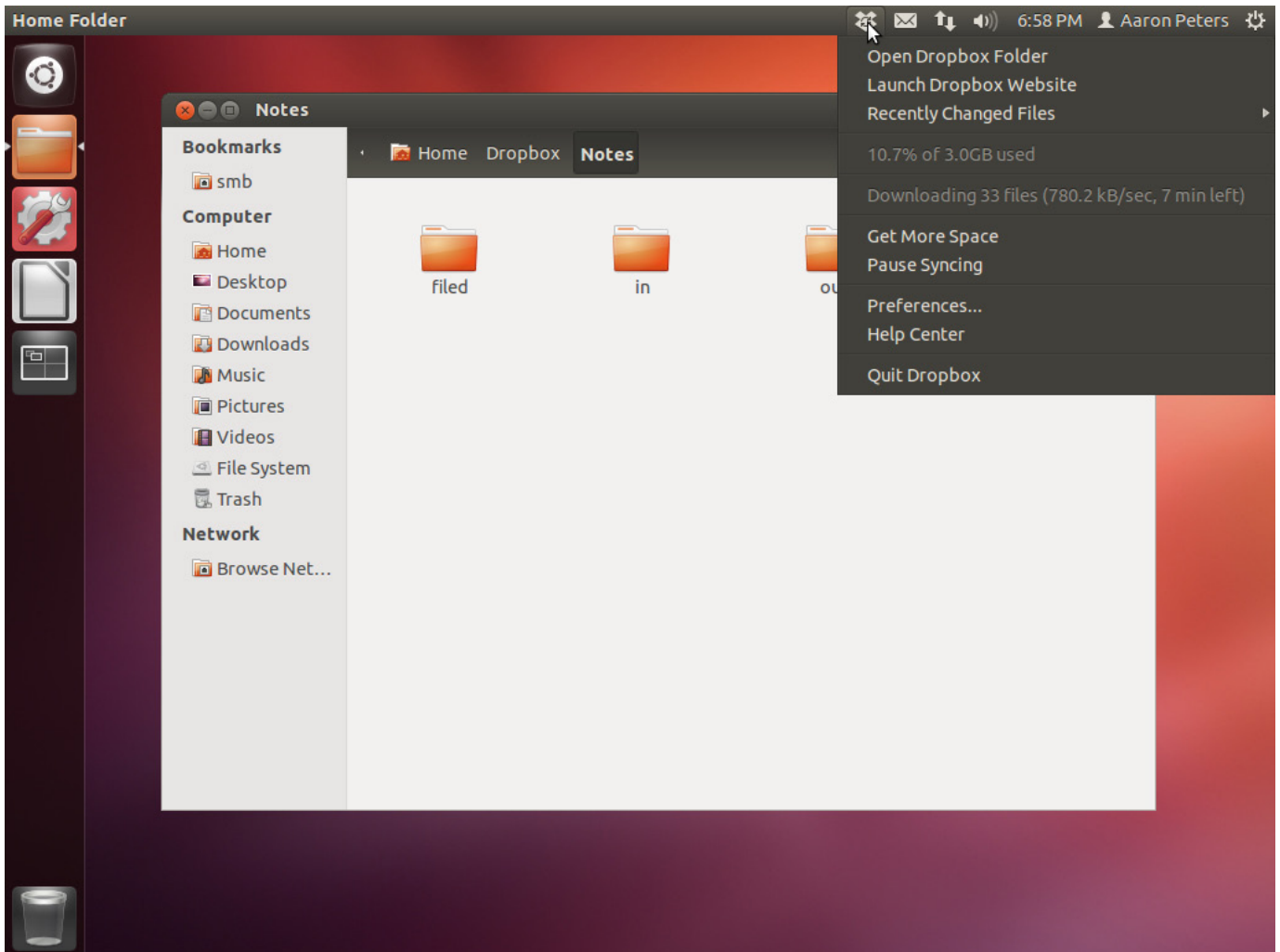


Figure 1. Nautilus Context Menu

constantly connected via 3/4G, Wi-Fi or wired networks to the Internet, cloud services have been one of the most hassle-free ways to make your data available across devices.

Dropbox

Of the free file-sharing services out there, Dropbox is arguably the most popular, perhaps because it's the simplest—no bells and whistles, no

long, complicated feature list, just good old-fashioned cloud storage. And with support for both Android (via the application in Google Play at <https://play.google.com/store/apps/details?id=com.dropbox.android&hl=en>) and Linux, either for GNOME and other GTK-centric desktops (using the the Nautilus plugin from Dropbox shown in Figure 1 and available

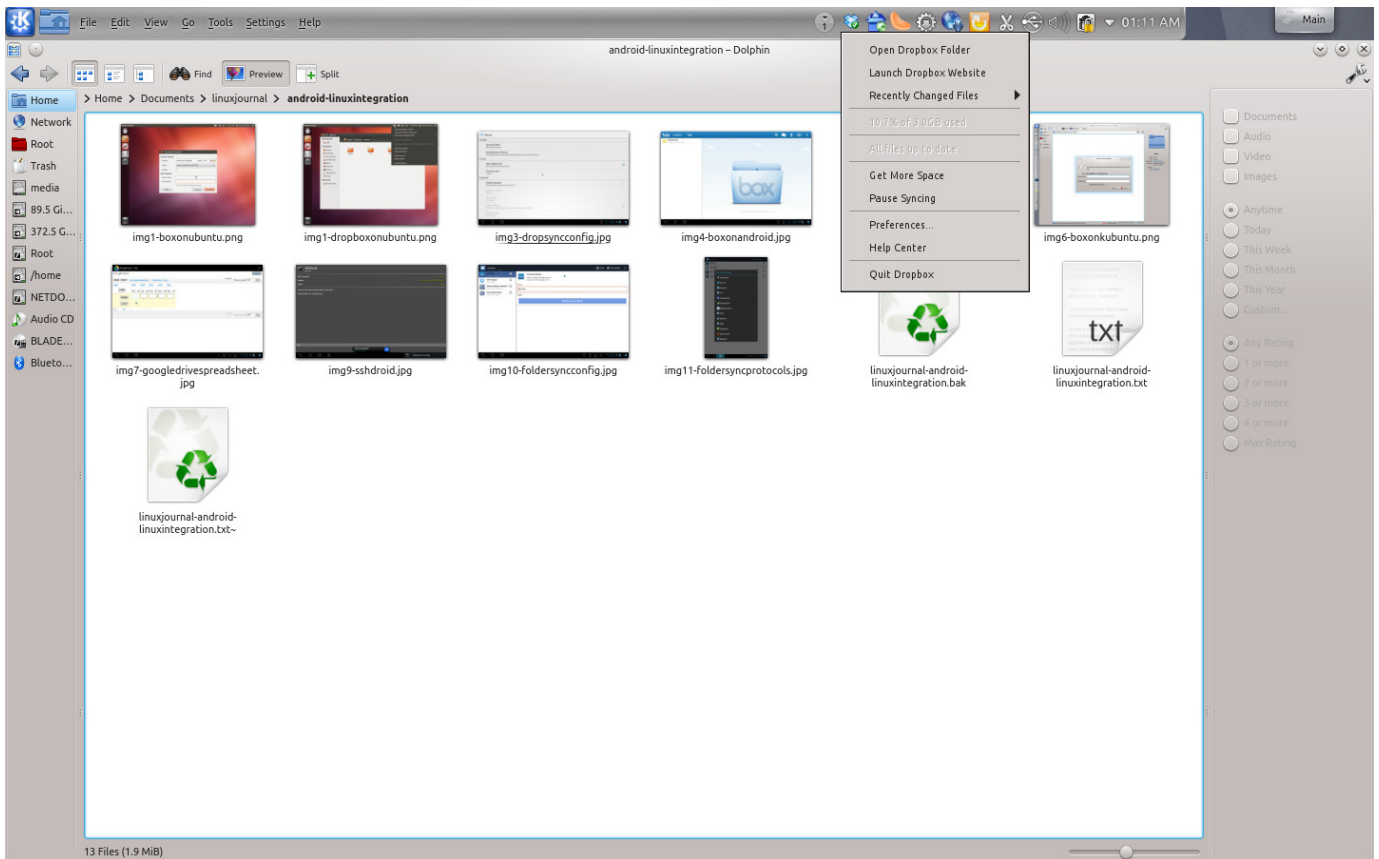


Figure 2. KFileBox Menu and Config Window

at <https://www.dropbox.com/install?os=lnx>), or KDE (via the excellent KFilebox—at the time of this writing, the project’s home page lists 0.4.7 as the most recent version, <http://kdropbox.deuteros.es>, but the SourceForge page, <http://sourceforge.net/projects/kdropbox>, lists a version 0.4.8 that works very well—shown in Figure 2).

Pointing each of the above at the same folder tree will help keep all your important folders close at hand. However, it’s important to

note the “official” Dropbox app above keeps a list of your files, but it doesn’t actually sync up the files themselves—that is, if you upload a revised file to Dropbox from your Linux box, then later go off-line with your mobile device, the Android gadget will know that file changed, but you won’t be able to view or edit it until you go back on-line. However, a free app called DropSync (<https://play.google.com/store/apps/details?id=com.ttxapps.dropsync&hl=en>)

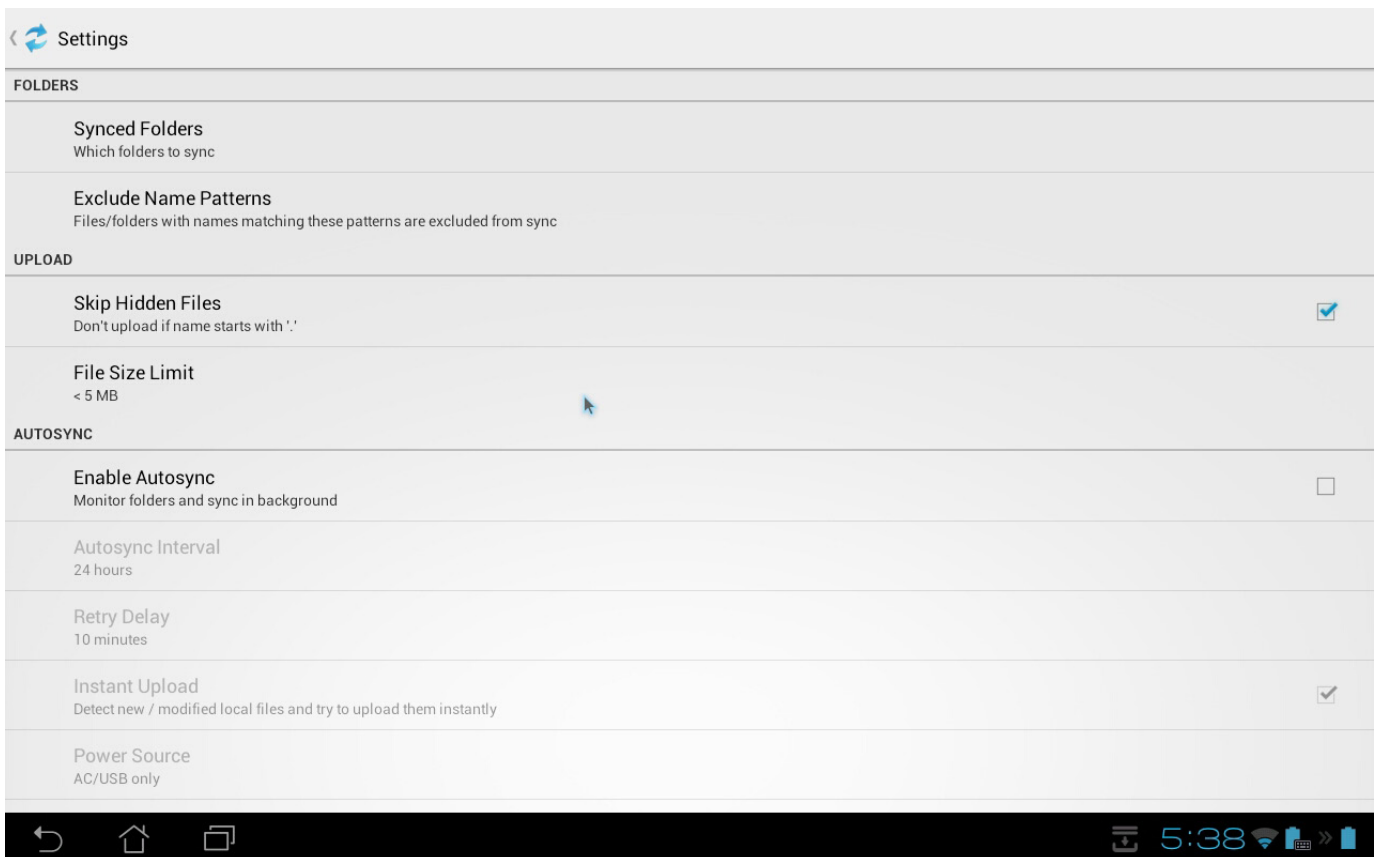


Figure 3. DropSync Config Screen

will do this for you (Figure 3). In addition, Dropbox is supported internally by a wide variety of individual Android apps, which will let you edit files directly from or save files directly to your Dropbox account. An example of this on my Transformer Prime is Epistle (<https://play.google.com/store/apps/details?id=com.kooklab.epistle&hl=en>), a very elegant Markdown editor, which automatically updates the list of files in its folder to a folder on Dropbox.

Box

Box, like Dropbox, offers users free on-line storage space accessible via a Web interface. Box also has an app in the market (<https://play.google.com/store/apps/details?id=com.box.android&hl=en>, Figure 4). One advantage of the Box app over Dropbox is that it automatically notifies you of updates to files. The Box service itself also has some nice features, some of which are available only with a premium subscription, including version management and

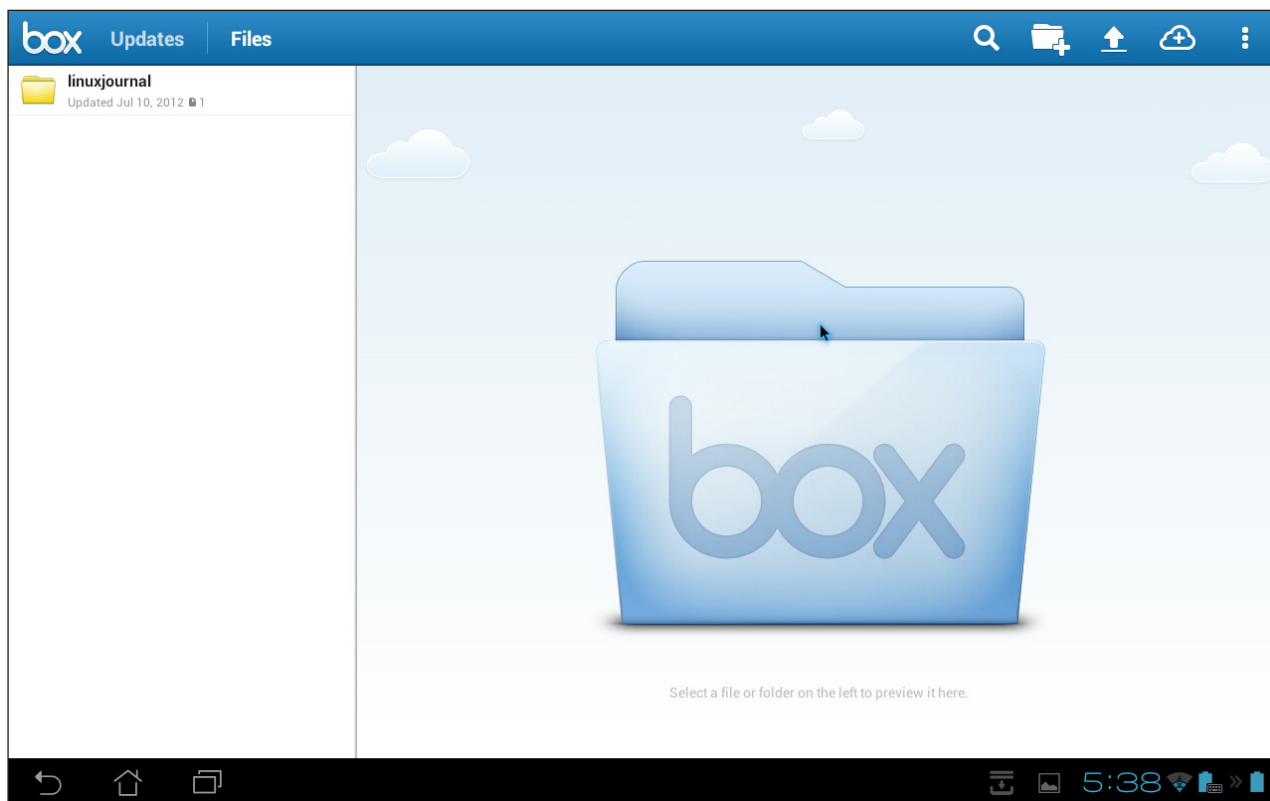


Figure 4. Box Android App

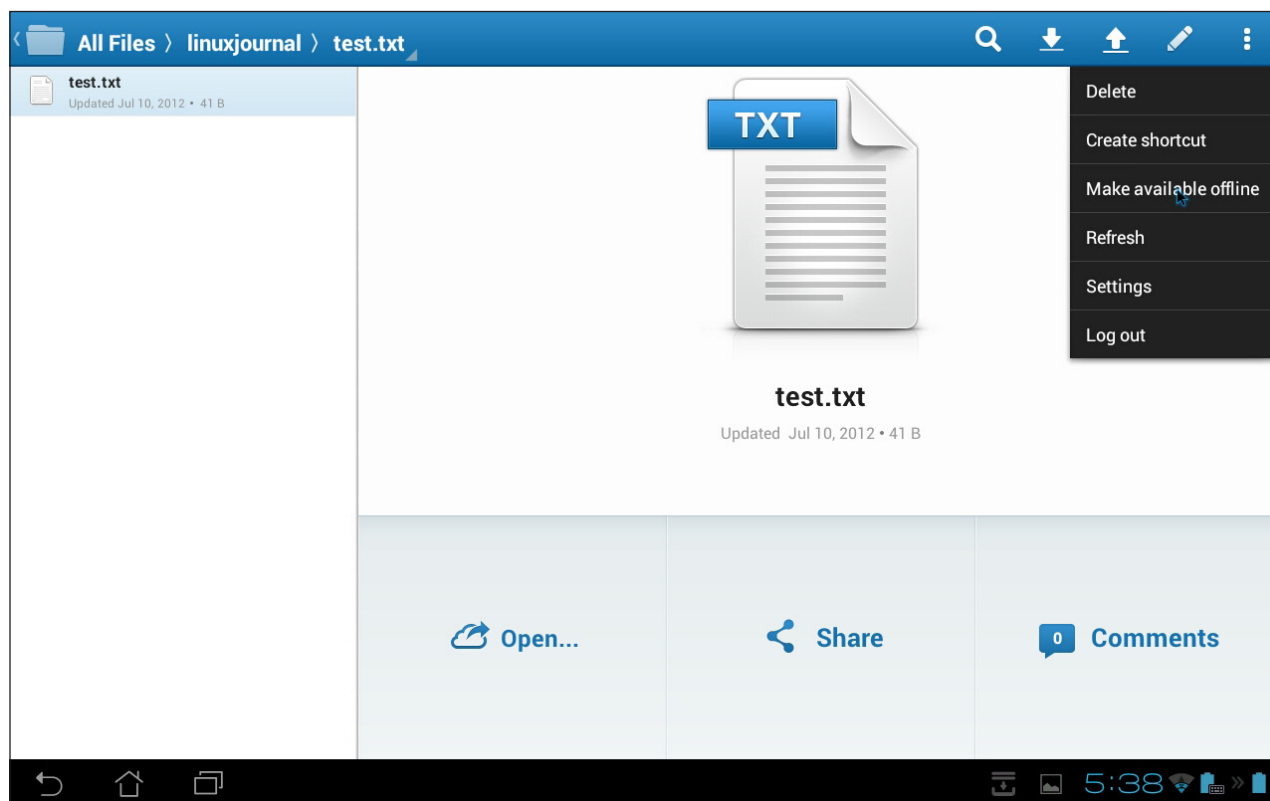


Figure 5. Box “Make Available Offline” Menu

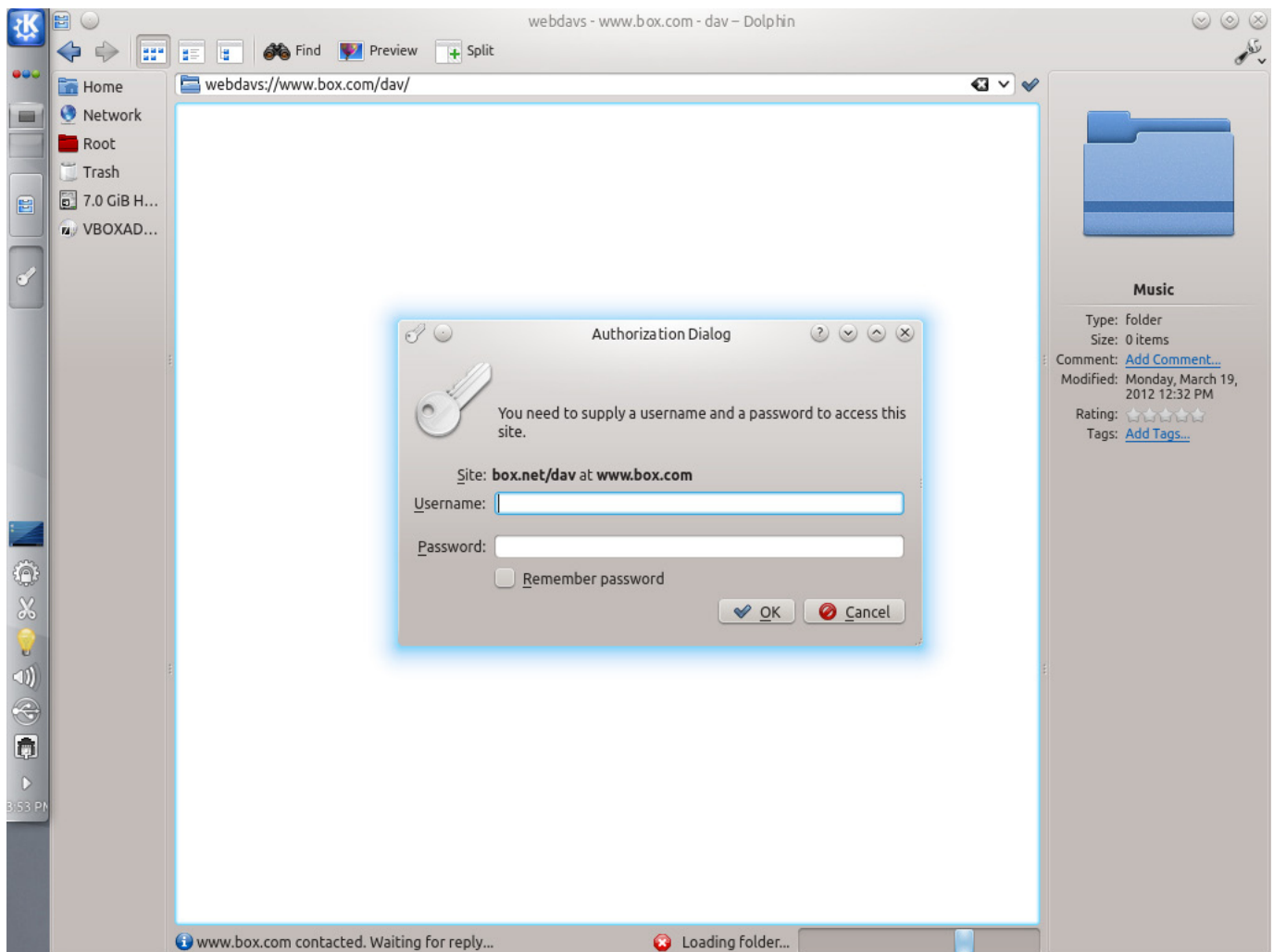


Figure 6. Box via WebDAVS

integration with other Web apps, such as LinkedIn, Salesforce, NetSuite and Basecamp. However, it suffers from one of the same weaknesses as the official Dropbox app: when the app is on-line, it updates only information on the files in your Box account, rather than caching a version of the files. Although it does have an option to mark files to "Make Available Offline" (Figure 5).

On the Linux side, although Box

doesn't have a native client program available, it does permit access to your files via WebDAVS. This means you can set up a shortcut in Nautilus (by connecting via the "Connect to Server" option to `dav://www.box.net/dav`, making sure to select "Secure WebDAV" per these instructions: <http://benjaminkerensa.com/2011/10/27/how-to-mount-box-net-securely-on-ubuntu-11-10>) or Dolphin (for some reason I could

not get the “Add Network Folder” dialog to connect, but simply typing `webdavs://www.box.net/dav` into Dolphin’s address bar prompted my credentials and worked like a charm, as I proudly display in Figure 6). In some ways, I prefer this to being forced into using a proprietary client program; on the other hand, the Dropbox client for Linux does automatically update local copies of files, while Box’s WebDAV access feature will require that you’re on-line unless you take additional steps.

Google Drive

Some heralded the re-branding of Google Docs to Google Drive as the beginning of the end for Dropbox and its brethren (perhaps some still believe this to be the case). With the built-in editing capabilities of Google Docs behind it, Google Drive is certainly a killer tool for collaboration and productivity. I’ve used shared text documents and spreadsheets with clients and colleagues, and having an on-line place both to stash this important information as well as work on it in

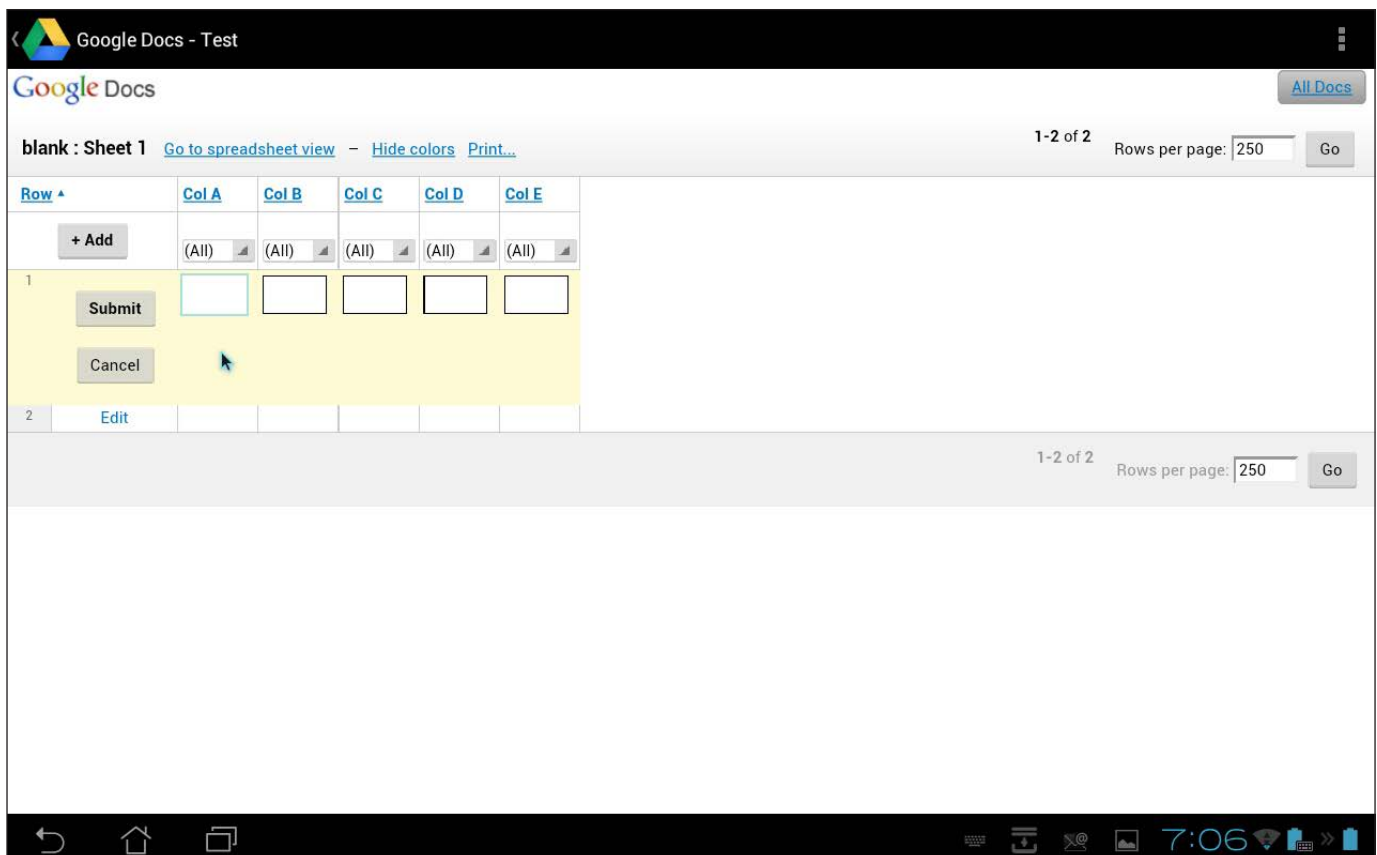


Figure 7. Google Drive Android Spreadsheet Editor

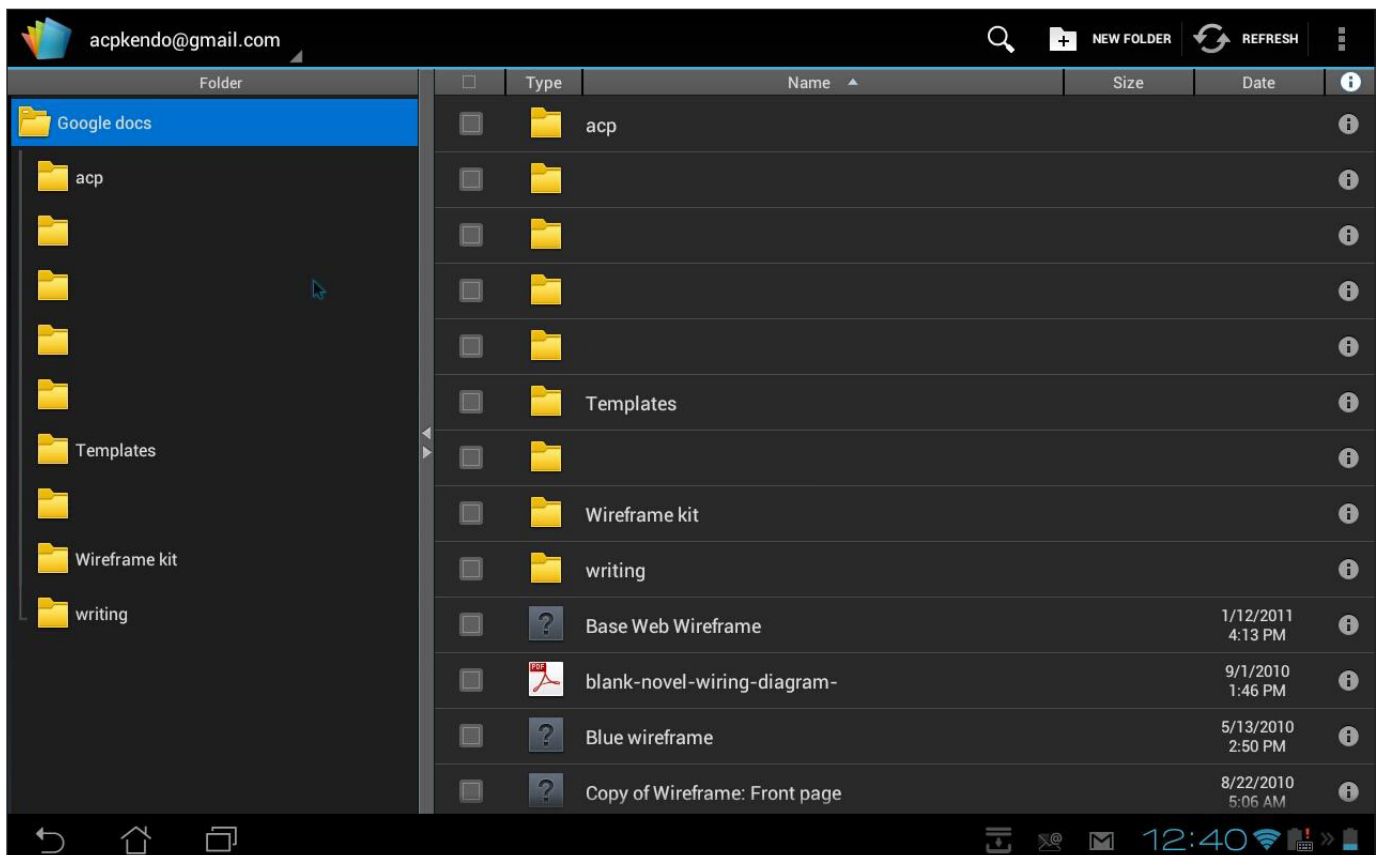


Figure 8. Polaris Office Displaying Google Drive contents

real time has been a huge time saver on more than one occasion.

But placing all your data in Google Drive isn't without its drawbacks. Google uses its own internal formats for the text documents, spreadsheets, presentations and drawings in Google Drive. While it's very non-evil about allowing you to download your files in Linux-friendly formats (even ODF for text and spreadsheets, huzzah!), it still involves conversion, which carries with it the risk of *misconversion*.

The recently updated Google Drive

app at <https://play.google.com/store/apps/details?id=com.google.android.apps.docs> (yeah, I included a link, but if you've got an Android device, you've got it already, no?) is much improved from the initial versions, in which the document editor operated through Web-based text areas. Unfortunately, the spreadsheet editor still requires you to click an Edit link at the beginning of the row to edit the values in that row (Web-based text fields), shown in Figure 7. As for file management, like Box, Google Drive will save files locally

for you to edit if you're off-line, but only if you select the Available Offline option for each file to which you'll need access. In addition, Google Drive also is supported by individual apps (like DropBox above). In addition to its own app, Google Drive is an acceptable storage place for Polaris Office (pre-installed on my Prime, shown in Figure 8) and Documents to Go.

Others

The following items also deserve special mention, and although they're not quite as widely known, accepted and/or supported across the Android community yet, each has some nice features that are worth a look.

- **Ubuntu One:** Canonical's entry into the cloud storage and Web services game, it has the benefit of a commercial supporter of the Linux client. In addition, Ubuntu One goes beyond simple file synchronization and will have the ability in the future to keep some of your more data-centric applications (such as contacts and notes) up to date as well as stream music. The support is a little patchy (for example, it will synchronize contacts, but not calendar or task data, and only on Ubuntu at present), but Ubuntu

One's promise of a "personal cloud" is certainly enticing.

- **Spideroak:** if you're nervous about entrusting all your sensitive data to a service provider's BOFH's for all you know, Spideroak may be right up your alley. The service's biggest selling point is "zero-knowledge" encryption on all your data—that is, even though the company hosts it, even it can't break into your files. It also maintains a version history on files, a feature typically only for premium customers of other services. Finally, in addition to mobile (iOS, Android and Nokia N900), the company has comprehensive Linux support, providing clients in DEB (Ubuntu/Debian), RPM (Fedora/OpenSUSE/RHEL/CentOS) and even TGZ format for Slackware users.

On the Local Network

For the paranoid among us, there are concerns about leaving all your sensitive data in the hands of corporate overlords. Fortunately, there are options for even the most anti-corporate shell jockey to connect Android and Linux over a local network.

The "Linux" Way: SSHDroid

One option is to synchronize from the

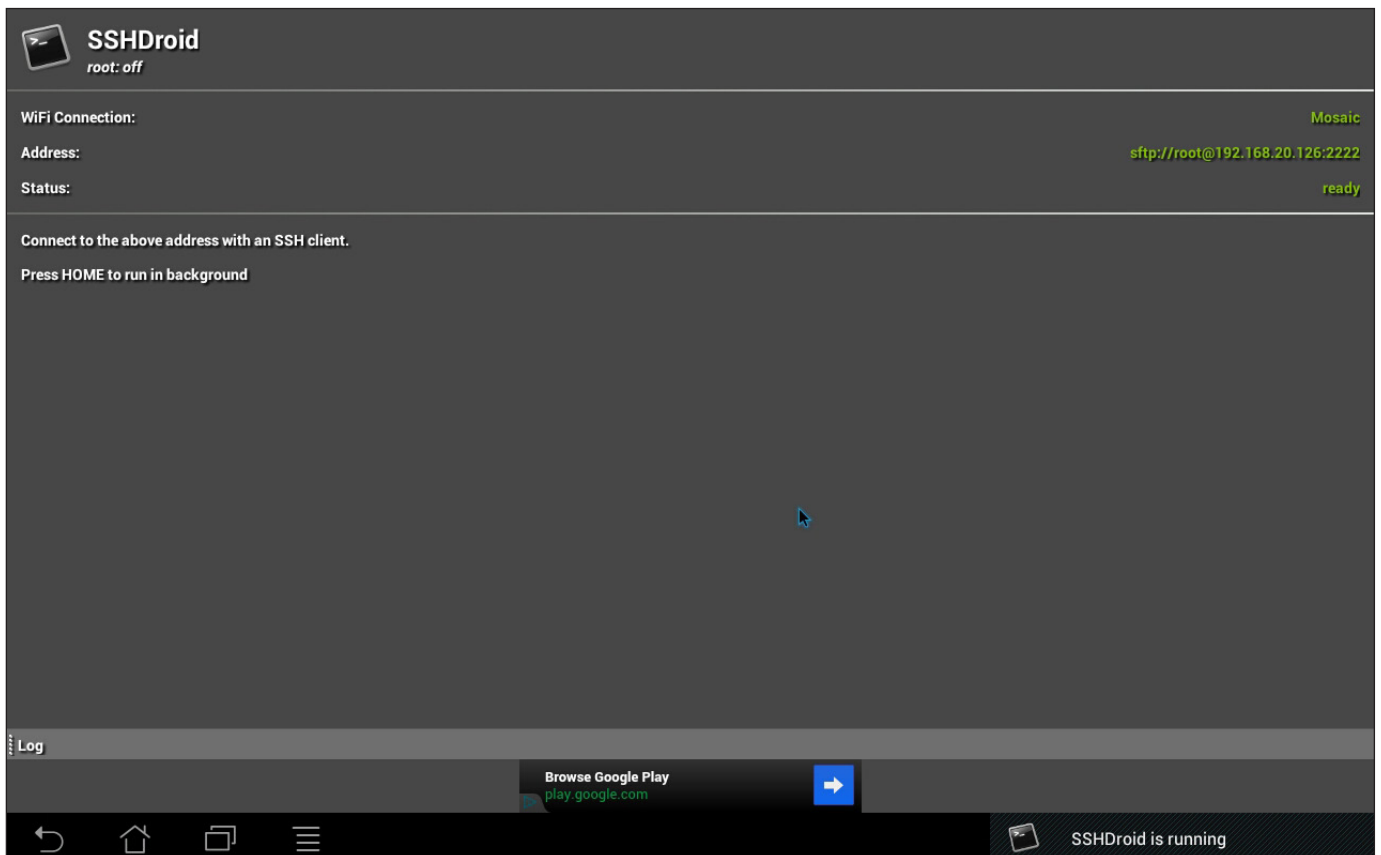


Figure 9. SSHDroid Main Screen

Linux side, meaning there needs to be a mechanism for your Linux box to see and manipulate the files on the Android device. SSHDroid (<https://play.google.com/store/apps/details?id=berserker.android.apps.sshdroid&hl=en>) provides a full SSH server for your device. As shown in Figure 9, starting this app displays a screen telling you everything you need to know, including your current IP address, the URL to connect to (it uses the SFTP protocol and defaults to port 2222) and the status of the server.

Having used this quite a bit to edit

files directly over SFTP (one reason why I love kioslaves), I can say this is probably my favorite way to use my Linux and Android machines, for a couple reasons. One, it takes the least amount of setup: you install SSHDroid, start it up and go to a URL from the Linux machine. And, Bob's your uncle. Second, it's secure. Third, while I generally use it to edit files directly over SFTP, once you're connected, you can use an application like Unison or Krusader to synchronize files. And last, the performance for large transfers is not too shabby on my Prime.

That said, this method is best suited for those who use the Android device as a mobile extension of their desktop machine—that is, those for whom the Linux box is the boss. For those of you who do more and more computing on tablets and other mobile devices, it never hurts to have SSHDroid installed (it's free and takes up less than 1MB, rare nowadays). A more Android-centric solution is described below.

The “Android” Way: FolderSync (S/FTP, Samba, WebDAVS)

For those of you who are enjoying the

freedom of browsing the Interwebs or writing from a hammock in the back yard (which, if you haven't tried it, I highly recommend) but still want to practice good backup procedures, FolderSync (<https://play.google.com/store/apps/details?id=dk.tacit.android.foldersync.full&hl=en>) is an excellent solution. It isn't open source, or even free, but at \$2.29 for the Pro version, it's practically a no-brainer once you figure out what it can do for you.

And what is that? It will keep one folder on your device synced up with

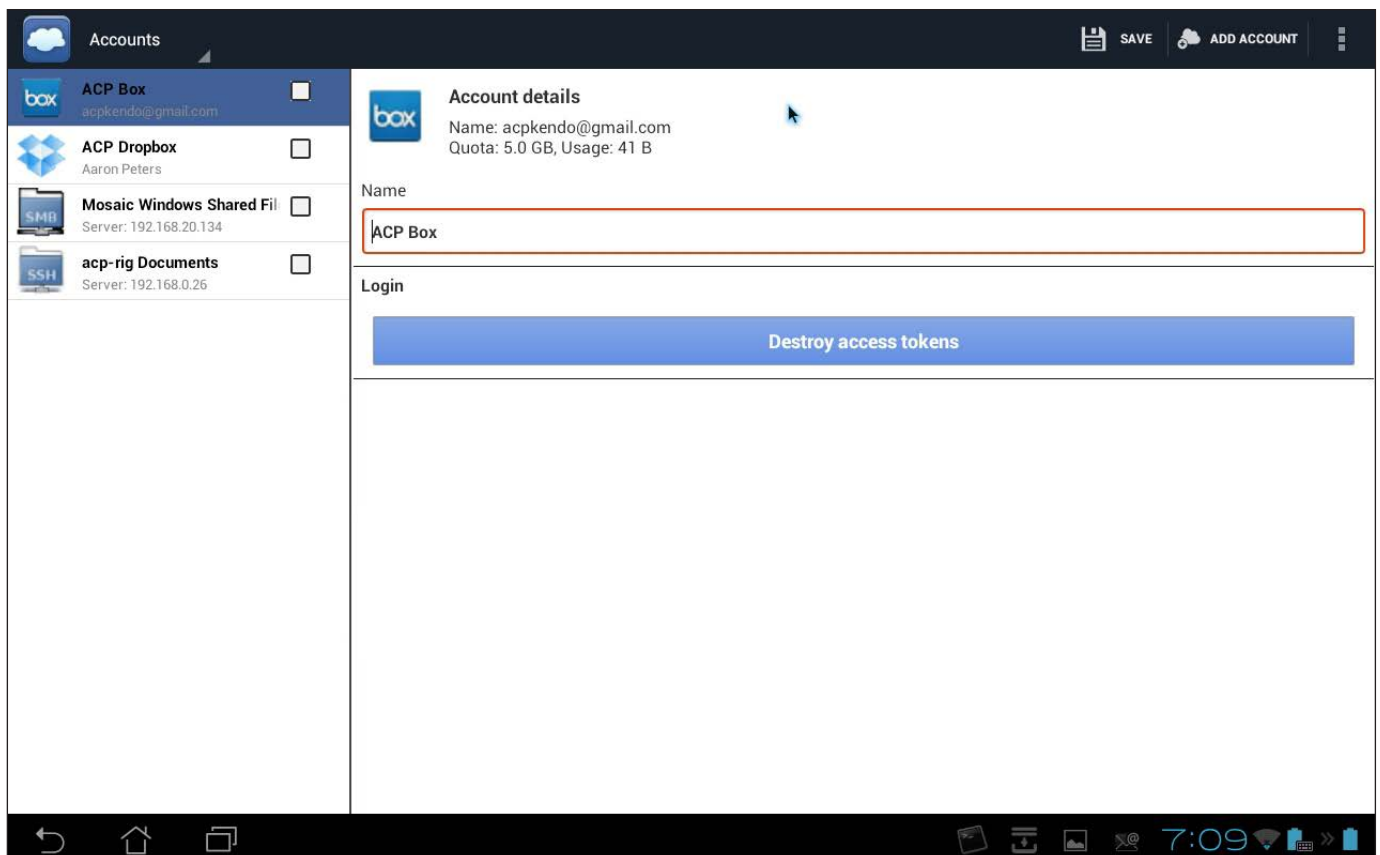


Figure 10. Configuration Options for Foldersync

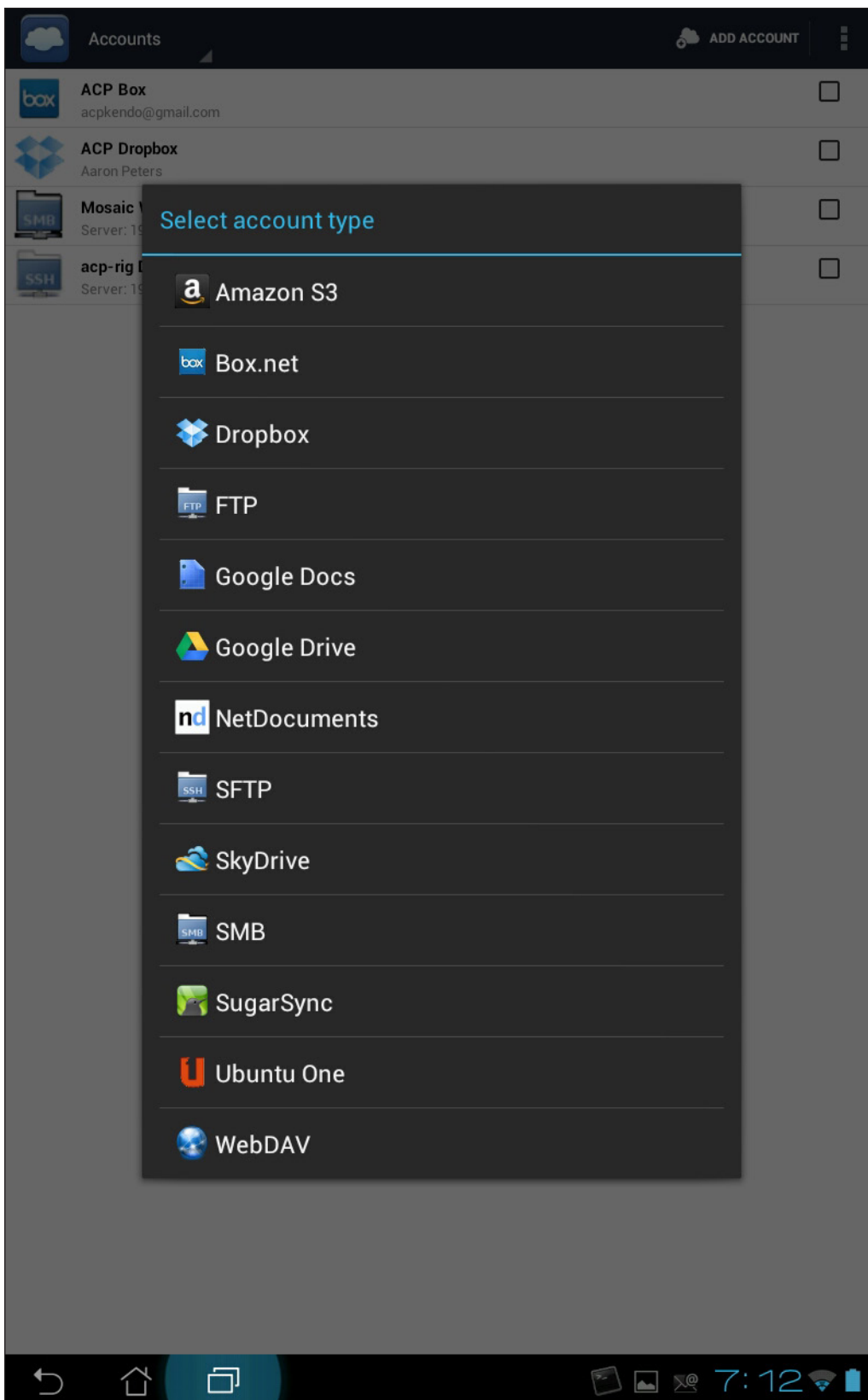


Figure 11. FolderSync-Supported Protocols

a folder on your Linux box over SFTP. You can opt to start the sync jobs manually or schedule them, with useful options, such as limiting certain jobs (called Folderpairs, as shown in Figure 10) to certain wireless networks, only synchronizing when the power is plugged in, and you can choose whether files should be updated one-way or bi-directionally.

But the great thing about FolderSync is all the different protocols it supports (a part of that selection list is shown in Figure 11). Have a file server at work that exports a Samba share?

FolderSync will link up to that, no problem. Want to do some updates to a site on your Web server? Get WebDAV(S) running on Apache, and you're set. Oh, and remember all those cloud services we talked about? Dropbox, Box and Google Drive? FolderSync does that one, that one and that one too.

The "Lite" version will allow you to sync up with one other folder on one device, so if that's all you need, you can avoid having to pony up any cash. But the Pro version will allow you to set up your Android device as a central hub for anywhere you stash files. Now if only they started making devices with 1TB Flash drives....

Direct Connection

The last, and slightly old-school way, to connect your Android device to your Linux box is via a direct USB connection. While this may evoke feelings of nostalgia for longtime gadget geeks who remember popping a Palm into a cradle and hitting the "HotSync" button, I find this to be the worst experience on newer devices, for reasons I'll explain next.

The Gingerbread (2.3.6 and below) Way

On Android devices prior to v.3.0, Google did the "right thing" to enable access to the device's

filesystem. When plugged in via a USB cable, the device appears to be just another USB drive. You could move files to and fro, access documents directly on the device, and basically treat the phone or tablet just as you would any other thumbdrive (with maybe the exception of leaving it in your pocket to go through the wash).

Like SSHDroid above, once this USB storage was mounted, you could use any Linux tool at your disposal (Unison, Krusader, rsync) to make sure they were up to date. All was well, until Google tried to be too smart for its own good.

The Honeycomb (3.0 and above) Way

From Android v3.0 and up, plugging a device in via USB no longer shows up as USB storage (that is, the "easy way"). Rather, you're required to choose in the device's settings whether, on USB connection, you'd like the device to use the MTP protocol (that is, to appear to the other machine as a media player) or the PTP protocol (that is, to appear as a camera).

Now, I've read that there's a technical reason for Google's decision to do this, mainly that all applications and data now can reside on a single filesystem (as opposed to having to choose, for example, to install apps on the "phone" or on the "SD card",

as I do on my OG Droid). All I would argue is that, for this user, those benefits do not outweigh the terrible experience of trying to use MTP on Linux (PTP actually works quite well, but only gives you access to the “DCIM” folder, so unless you want to store all your other stuff alongside the pictures taken by the built-in camera, this won’t do).

I spent the better part of a weekend combing through posts on the XDA forums (<http://forum.xda-developers.com>), which is a fantastic resource for all sorts of Android hacks, trying to find a nice, automated method of mounting the Prime’s SD card. I found a couple resources (<http://www.omgubuntu.co.uk/2011/12/how-to-connect-your-android-ice-cream-sandwich-phone-to-ubuntu-for-file-access> and <http://forum.xda-developers.com/showthread.php?t=1143044>), but eventually settled on the script and instructions provided via this YouTube video: <http://www.youtube.com/watch?v=3ehnoJn6CEk>. After all that, I sat down, ready to see the Prime as just another drive in /media, just like the old days.

Well, not only is MTP access on Linux inconvenient to use, it’s interminably slow. Once I got connected, I started copying my music collection to the Prime and left it plugged in overnight

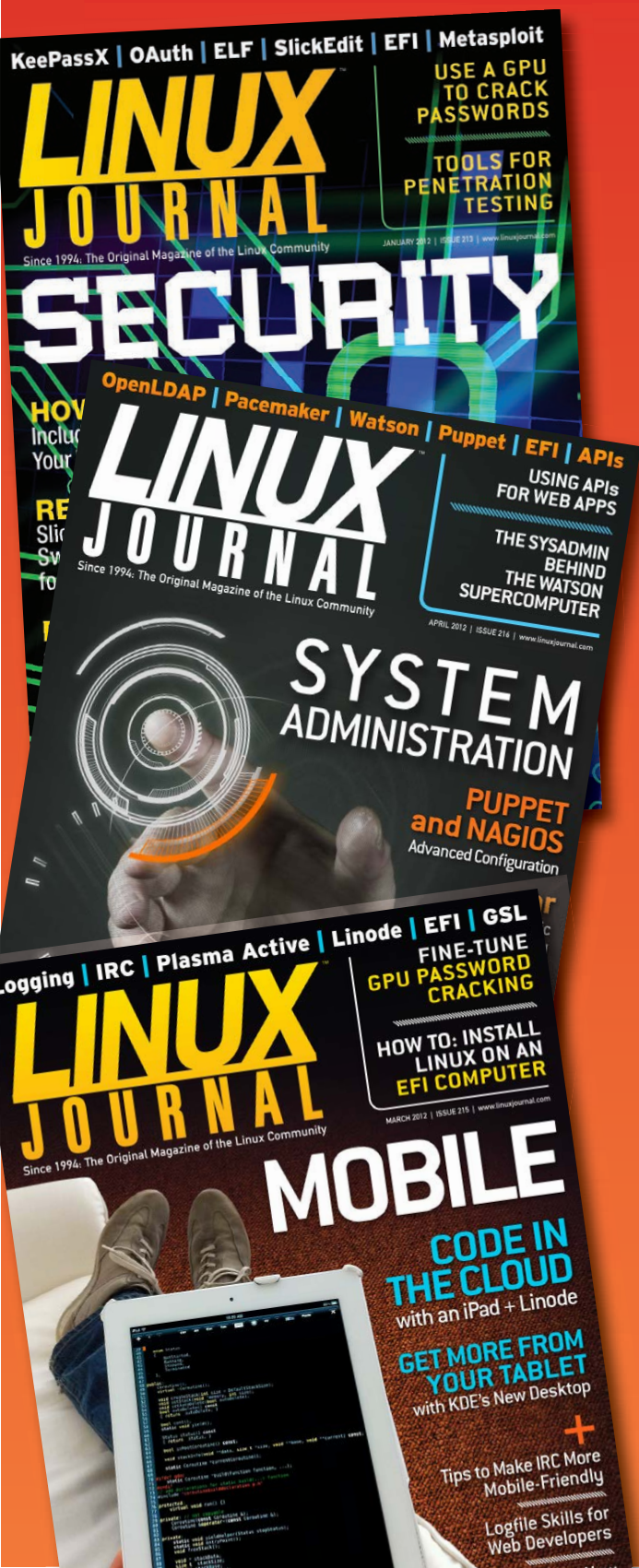
to do so. When I got up the next morning, it was approximately 5% completed. Before you start asking for transfer rates and whatnot, I don’t have them, but I was able to transfer about half that same collection within a couple hours, and over SFTP (so with en/decryption overhead) no less. So I’ve pretty much sworn off direct connection for the Prime—there are so many other ways to shuffle files and data around, who needs it?

Conclusion

One of the great things about Android is that the ecosystem is free to come up with a variety of solutions to a problem and let users sort out which one best fits their needs. It could be that no one of the above alone will suit you—I myself use both SSHDroid and FolderSync on almost a daily basis. But all of the above apps are either free, or have free trial versions, so there’s nothing stopping you from testing them out. Give it a try, and the robot and penguin will be getting along famously in no time! ■

Aaron Peters is a project manager and business analyst at a Web/mobile development firm, and he splits his free time between learning tech, writing and attacking other people with bamboo sticks. When he and his wife are not trying to corral the five animals living with them in Allentown, Pennsylvania, he sometimes answers e-mail sent to acpkendo@gmail.com.

If You Use Linux, You Should Be Reading **LINUX JOURNAL**



- » In-depth information providing a full 360-degree look at featured topics relating to Linux
- » Tools, tips and tricks you will use today as well as relevant information for the future
- » Advice and inspiration for getting the most out of your Linux system
- » Instructional how-tos will save you time and money

Subscribe now for instant access! For only \$29.50 per year—less than \$2.50 per issue—you'll have access to *Linux Journal* each month as a PDF, in ePub & Kindle formats, on-line and through our Android & iOS apps. Wherever you go, *Linux Journal* goes with you.

SUBSCRIBE NOW AT:
WWW.LINUXJOURNAL.COM/SUBSCRIBE



DOC SEARLS

Leaving the Land of the Giants

The next revolution will be personal. Just like the last three were.

The cover of the December 1st–7th 2012 issue of *The Economist* shows four giant squid battling each other (<http://www.economist.com/printedition/2012-12-01>). The headline reads, “Survival of the biggest: The internet’s warring giants”. The squid are Amazon, Apple, Facebook and Google. Inside, the story is filed under “Briefing: Technology giants at war”. The headline below the title graphic reads, “Another game of thrones” (<http://www.economist.com/news/21567361-google-apple-facebook-and-amazon-are-each-others-throats-all-sorts-ways-another-game>). The opening slug line reads “Google, Apple, Facebook and Amazon are at each other’s throats in all sorts of ways.” (Raising the metaphor count to three.)

Now here’s the question: *Is that all that’s going on? Is it not possible that, in five, ten or twenty years we’ll realize that the action that mattered in the early twenty-teens was happening in the rest of the ocean, and not just among the mollusks with the biggest tentacles?*

War stories are always interesting, and very easy to tell because the format is formulaic. Remember Linux vs. Microsoft, personalized as Linus vs. Bill? Never mind that Linux as a server OS worked from the start with countless millions (or even billions) of Windows clients. Or that both Linus and Bill had other fish to fry from the start. But personalization is cheap and easy, and there was enough antipathy on both sides to stoke the story-telling fires, so that’s what we got. Thus, today we might regard Linux as a winner and Microsoft as a loser (or

at least trending in that direction). The facts behind (or ignored by) the stories mostly say that both entities have succeeded or failed largely on their own merits.

Here's a story that illustrates how stories can both lead and mislead.

The time frame was the late 1980s and early 1990s, and the "war" was between CISC (Complex Instruction Set Computing, http://en.wikipedia.org/wiki/Complex_instruction_set_computing) and RISC (Reduced Instruction Set Computing, http://en.wikipedia.org/wiki/Reduced_instruction_set_computing). The popular CPUs at the time were CISC, and the big two CISC competitors were Intel's x86 and Motorola's 68000. Intel was winning that one, so Motorola and other chip makers pushed RISC as the Next Big Thing. Motorola had an early RISC lead with the 88000 (before later pivoting to the PowerPC).

At the time, I was working with Sun Microsystems and its allies on SPARC, Sun's RISC design, which was implemented in various ways by a raft of chip makers, including Texas Instruments, Fujitsu and Cypress Semiconductor. In spite of Sun's heft in the marketplace, we had trouble getting attention for SPARC with the

tech pubs, because they tended to see everything as an Intel vs. Motorola fight. We felt we couldn't challenge either one of those guys head-on, even if SPARC was superior on technical grounds (which Sun and its partners believed). So we decided the best strategy was for SPARC to pick a fight with another RISC upstart called MIPS.

This was pure bait for the pubs, which came over to this new fight to see what was up. I think we caught MIPS off guard at first, but it defended itself well and ended up selling years later for hundreds of \$millions to SGI, which eventually went bankrupt. SPARC is still around, running gear made by Oracle, which acquired Sun. The big winner in the CPU market remained Intel and, therefore, CISC. In fact, the x86 architecture still rules, at least on PCs and servers, but not in mobile devices, where ARM (Advanced RISC Machine) now kicks butt. And for what it's worth, MIPS is now fighting ARM in the Android market, and Motorola's chip division is the long-since-spun-off ON Semiconductor.

So, five points here:

1. Vendors use stories as marketing strategies.
2. Vendor war coverage is always to

some degree an exercise in misdirection ([http://en.wikipedia.org/wiki/Misdirection_\(magic\)](http://en.wikipedia.org/wiki/Misdirection_(magic))), even when journalistic intentions are worthy ones.

3. The real story is always much more complicated than vendor war coverage can characterize.
4. “Winners” never win forever, especially in tech.
5. “Losers” don’t always die. Often they stay alive by selling out, or they thrive by finding niches and working them.

Now back to our four squid.

The graphic above *The Economist* story is an antique-style map (http://media.economist.com/sites/default/files/cf_images/images-magazine/2012/12/01/FB/20121201_FBD000.png) of the fantasy-fiction kind, drawn by David Parkins (<http://www.davidparkins.com>). It shows a large mountainous land, with the Sea of Content to the west and the Sea of Commerce to the east. Dividing the land are four throne-doms: Applechia, Google Earth, Amazonia and Fortress Facebook. A fifth, Empire of the

Microserfs, is across the Sea of Content in the northwest corner of the map, bordered by the Cliffs of Surface. In Google Earth are Adsense-land, the Mirkwood of Regulation, the Wastes of Litigation (“Here be lawyers”), Pagerank Pinnacle (at the end of Algorithm Reach), beside which lies The Firth of Android. Appleacia has the iPhone Keep. Amazon has the Cloud Mountains and a volcano named Kindle. Between the latter and Netflix Nation (which lies above the Satrapy of Spotify) intrudes Pirate Bay. Offshore are the eBook Islands. On the opposite shore are OneClick Castle and Prime Port. Somewhere in the middle, between the Cloud Mountains and Fortress Facebook is the Lost City of MySpace. Out in the Sea of Content are small islands called RIM Rocks and Nokia. Atop the map is The Dark Offline. Floating in the Sea of Commerce is a Chinese junk flying the Samsung banner. A peninsula in the southeast corner features Secondhand City, the Bay of E and the Cape of Coin. There’s a dragon smiling out of the Sea of Commerce, named The Next Big Thing. Finally, in the center of the map, between the four thronedoms, is an un-named body of water surrounding Identity Island.

Parkins’ antique style also depicts

antique substance in the making—because all four of the thrones (or squid, take your pick) are at least as affected by their own weaknesses as by the strengths of companies they are said to be fighting. And, because so many of us are at their mercy, their weaknesses are to some degree ours as well.

So let's look at those weaknesses, and then at where the rest of the action is, because neither are getting enough attention.

First, Apple.

While it's not wise to bet against a company as successful as Apple has become, it is wise to expect failure from a company whose success is rightly attributed to a dead and irreplaceable CEO. Although it was business as usual for a while after Steve Jobs perished in September 2011, it was clear a year later that the wheels were coming off. First there was the Maps app debacle, in which Apple replaced its Google-based Maps app on iOS 6 with one based on a stew of inadequate substitutes—and then failed to improve it for months while Google took its sweet time not producing its own Maps app for the operating system. This not only hurt Apple and iOS 6, but also the new iPhone 5, which featured

the Maps value-subtract and was itself an unspectacular successor to the iPhone 4s—which wasn't all that big an improvement on the iPhone 4, which came out way back in 2010. Meanwhile, for all of Apple's continued success with the iPhone, its entire iOS smart-thing hardware market contains just three devices (iPhone, iPad and iPad Mini) made by only one company. Meanwhile, Android remains an open platform with countless hardware implementations from many companies. As I write this, the new *Consumer Reports* rates various Samsung Galaxy devices ahead of the iPhone, which had formerly topped the magazine's ratings. Countless new Android phones also will hit the market before the iPhone 6.

In 2012, Apple also continued to make fixing or improving its hardware as hard as possible for anybody not an Apple employee. Batteries, RAM and solid-state storage on new Apple hardware tends to be hard-wired or -glued. One result is the latest MacBook Pro, with its retina display, which Kyle Wiens in *Wired* calls "the least repairable, least recyclable computer I have encountered in more than a decade of disassembling

electronics” (<http://www.wired.com/opinion/2012/10/apple-and-repeat-greenwashing> and <http://www.wired.com/gadgetlab/2012/06/opinion-apple-retina-displa>).

Credit where due: Apple has been brilliant at retailing and customer support. On the latter count, nobody else is even close. Also, Apple is advantaged by a competitor—Microsoft—that seems hell-bent on sending customers anywhere else.

At this point, it’s not clear where Apple is headed. The company’s only “wow” product since Steve died was the iPad Mini, which should have come out years earlier. In the past, it was easy to assume that Apple had a “next big thing” up its sleeve. Now it’s not.

On to Google.

Last October, Google took the wraps off the biggest thing it has in the physical world: giant data centers, which it immodestly calls “Where the Internet lives” (<http://www.google.com/about/datacenters/gallery/#/>).

The photos doing the bragging are as artful as can be, considering that the subjects look like power plants: vast and stark white buildings, with glowing racks inside and huge cooling gear outside, veined by an abundance of plumbing. It makes one pause to consider how dependent we have

become on giant companies and their very earth-bound “clouds”.

By coincidence, this month is the third anniversary of a column here titled “The Google Exposure” (<http://www.linuxjournal.com/magazine/eof-google-exposure>). In it, I wrote:

I’m just worried about the way Google makes money. Nearly all of it comes from advertising. That’s what pays for all the infrastructure Google is giving to the rest of us. As our dependency on Google verges on the absolute, this should be a concern. Think of advertising as oil and Google as one big emirate. What happens when the oil runs out?...The free rides won’t go on forever. There are better ways than advertising for demand and supply to find each other...and more will be found. Google will be in the middle of that discovery process, no doubt. But it’s an open question whether Google will make the same kind of money in a post-advertising marketplace. I’m betting it won’t.

Since then, Google has continued growing at a 20+% annual rate, and diversifying a bit (for example, by acquiring Motorola Mobility). But the vulnerabilities are still there: for


Google and therefore also for the rest of us. Also, the Internet that “lives” in Google’s data centers has become an overwhelmingly commercial one, especially on the Web. The percentage of information on the Web that isn’t about selling something continues downward as more and more eyeball-routers get into the ad-based game—and game that game as well. How far can this go before the whole ad-funded system, with Google at its center, begins to fail in big and obvious ways? No way to tell, but the system we have now can’t go on forever. Trees do not grow to the sky.

Next, Facebook.

An alpha geek told me recently that the most remarkable thing about Facebook is the sturdiness of its infrastructure: it rarely if ever goes down. Compare that to Twitter, a much smaller service notorious for its familiar “fail whale”. Facebook’s infrastructure should be good for many things other than housing a locked-in “social” space where inhabitants get advertised at. What if Facebook started offering paid services to its users, turning them into actual customers? For example, it could work as a fourth-party agency (<http://blogs.law.harvard.edu/vrm/2009/04/12/vrm-and-the-four-party-system>), helping customers

actually *find* products and services, rather than merely searching for them, as they do with Google. Facebook could host personal clouds (http://www.windley.com/archives/2012/11/the_cloud_needs_an_operating_system.shtml) of data kept private for paying customers, selectively disclosing required data to potential sellers (or government agencies, or nonprofits) on a secure need-to-know basis—treating personal data the way a bank (as a fourth party) treats customers’ money. Prototype work on this kind of thing has already taken place at Innotribe (<http://innotribe.com>), the innovation arm of SWIFT (<http://www.swift.com>), the banking nonprofit that moves \$trillions around the world every day. I know, because I’ve been involved in it. But Facebook won’t go there because Facebook, like Google, sees its main business as advertising and would rather do business with businesses than with individuals. Also, like Google, it would rather sell its users to advertisers than serve as an intermediary in the far larger retail and services marketplace.

One reason Facebook won’t make that move was suggested to me by a top executive at an advertising



company a couple years ago. He told me the blinders both Facebook and Google wore were the ones that keep them focused mostly on each other. While this isn't a verbatim quote, it's close enough: "Google envies Facebook's ability to get personal with users, while Facebook envies Google's ability to put ads everywhere on the Web." Thus, we have locked tentacles rather than evolution by either squid.

Next, Microsoft.

Today in the mail came our copies of *Vanity Fair* and the *New Yorker*, both Condé Nast publications. Both looked different and confusing. Instead of the usual cover art, there were collections of squares and rectangles that called to mind the tablet app Flipboard, which organizes "social" content in the form of picture-tiles one can flip through like one would a magazine. I have Flipboard, but its lack of an outline-like organizing structure, such as a directory or a table of contents, annoys me. I thought, *This can't be real. This has to be an ad for something.* Then I saw the small print: "A sample of the new *New Yorker* experience on the Windows 8 desktop." *Oy vay.* Microsoft and Condé Nast hit into a triple-play on that one, because it made me hate the OS while

dreading at the same time having an "experience" like what it showed.

So far, I've met only one Windows user who likes Windows 8, and that's just for some deeply buried technical stuff. Everybody else either doesn't like it or hates it outright. The UI, reportedly nice on phones and tablets, is strange on anything with a keyboard and mouse or trackpad. The learning curve is more like a wall, and—well, nobody asked for all this new stuff. As for the new Surface tablet, it looks like the second coming (and going) of the Tablet PC (http://en.wikipedia.org/wiki/Microsoft_Tablet_PC). One version of the OS doesn't even run Microsoft's Office apps. Some game developers called the new OS and its Apple-like "store" for silo-ing apps a "catastrophe" (<http://www.neowin.net/news/valve-co-founder-windows-8-is-a-catastrophe>) and a "disaster" (<http://www.neowin.net/news/blizzards-rob-pardo-windows-8-is-not-awesome-for-the-company>).

On the mobile front, Microsoft teamed up with Nokia to bet the former mobile-phone giant's farm on Windows-based phones, which promptly tanked in the marketplace. Now farmland for both companies is shrinking like a puddle on a hot day.

In fact, Microsoft has some legacy advantages. It always has been far more open and supportive toward developers than Apple. Unlike Facebook and Google, its users are actually paying customers. And it has always been, at its heart, a *personal* computing company. That too should give it a kind of advantage over Facebook, Google, Twitter and everybody betting on “social” (read: advertising), “the cloud” and “big data”—all of which are corporate/enterprise plays.

Over the years, I’ve known and worked with a lot of good people inside Microsoft, all of whom have labored to open the company’s technology, make it play better with others in the marketplace, and put some truly innovative technologies to work. The company’s decision to default Do Not Track in the “on” position with the latest rev of Internet Explorer was astute, correct and perhaps even brave. It’s the kind of thing that a clued-in company would do. I’ve also seen some excellent Microsoft research on user feelings and preferences in respect to lost on-line privacy. That should energize Microsoft around some fresh opportunities, but the company seems to lack adrenal glands. Opportunities are lost every day the company fails

to win hearts and minds by standing behind users—its customers—in the fight against abuses of privacy.

Instead, Microsoft continues to fight Google straight-up with an Online Services Division that has lost \$billions over recent years.

Next, Amazon.

Amazon is strongest among *The Economist’s* four giant squid, or thrones. It succeeds, Jeff Bezos says, “by starting with the customer and moving backwards”. By 2009, Amazon already controlled more than a third of all e-commerce (<http://www.pcmag.com/article2/0,2817,2345381,00.asp>). Since then, I’ve heard numbers as high as 50%. Whatever the number, you can see the result by looking inside any UPS or Fedex delivery truck and eyeballing all the boxes labeled Amazon or Zappos (Amazon’s shoe store).

While Apple, Google and Facebook all clearly have good engineers and solid technical infrastructure, Amazon tops them all by connecting its innards directly both to individuals and to techies among business customers. It is a rare example of a geek-driven company that also understands and loves to do business with everybody it can.

Amazon’s only shortcoming is one

it shares with the rest of retailing, as well as with its big-squid competitors: it runs a big data silo where customer data goes in, but not back out to individuals. For example, I would like to have a cooperative data-sharing relationship with Amazon, in which I tell it everything I own (or feel like telling it I own), so it doesn't bother trying to sell me what I already have but didn't buy from Amazon. I would like my personal API to be one it could program against, just as I (or my fourth party) can program using its APIs. This requires a respect on Amazon's part for the fact that my life is bigger than the corner that deals with it—and that I can do more with my own data than it can. Also that this will be a Good Thing for both of us.

But there isn't any sign that this will happen, mostly because we don't yet have our own APIs, and managing our own data isn't something many of us do yet, least of all so we can deal in one consistent way with many suppliers. Mostly, we just fill up hard drives and hope whatever we have "in the cloud" is sort of safe and not going to bite us some way in the future.

Which brings us to the rest of the world.

The revolution we're in is a personal one, not a corporate one.

It is a revolution in which personal empowerment has turned out to be good for enterprises because it was good for individuals. This fact has been manifest ever since PCs appeared on Earth around the turn of the 1980s.

To MIS directors in 1983, "personal computing" was oxymoronic. Computing was a corporate thing called data processing. It was big and expensive and specialized and centralized. But those same MIS directors had to start dealing with personal computing because individuals in their organizations and out in the marketplace were getting more done with their own word processing, spreadsheets and accounting software than companies could get done with their old big-iron data-processing systems.

Likewise, IT directors in 1997 had to start dealing with personal communications (e-mail, instant messaging, personal publishing), because people in their organizations and out in the marketplace had tools of their own that stripped the gears of what the companies could do with their big old legacy systems.

IT directors in 2009 had to start dealing with iThings and Androids because that's what employees and users brought to work, and customers

brought to stores, along with zillions of apps that far exceeded what could be done with company-issued BlackBerries.

Today's "big data" bluster—all that stuff about how marketing can now know more about the customer than she knows about herself—is mainframe talk. Individuals know more about themselves than systems of any kind can guess at, no matter how much data those systems gather. Given the means to control our own lives, with their own personal platforms (not just ones on their devices, but on their own pile of data), we will be able to do far more with that data than can any other entity. We also can do it cooperatively with other entities, provided neither of us is busy trying to lock in or control the other.

In the next several years, personal data and personal operating systems for managing relationships using that data will be as revolutionary as PCs were in 1983, the Internet was in 1996 and mobile was in 2009. We can keep watching giants battle all they want. But the action that matters most won't be theirs. It will be ours. ■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

Advertiser Index

Thank you as always for supporting our advertisers by buying their products!

ADVERTISER	URL	PAGE #
1&1	http://www.1and1.com	15
APACHECON	http://na.apachecon.com	53
BIG DATA TECH CON	http://www.bigdatatechcon.com/boston2013/	99
CONFOO	http://Confoo.ca	81
EMAC, INC.	http://www.emacinc.com	11
EMPEROR LINUX	http://www.emperorlinux.com	13
FLOURISH	http://www.flourishconf.com/2012/	33
IXSYSTEMS	http://www.ixsystems.com	7
SCALE	https://www.socallinuxexpo.org/scale11x/	2
SILICON MECHANICS	http://www.siliconmechanics.com	3

ATTENTION ADVERTISERS

The *Linux Journal* brand's following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit <http://www.linuxjournal.com/advertising>.