

LINUX™ JOURNAL

Since 1994: The Original Magazine of the Linux Community

Test Security with an

Internal Phishing Campaign



Build Your
Own Cluster:
Installation

Update Your
Ticketing System from
the Command Line

Cool Project: Set Up
YouTube Live Streams

JUNE 2017 | ISSUE 278
<http://www.linuxjournal.com>

EOF:
Open Source
Comes
of Age



WATCH:
ISSUE
OVERVIEW



**Practical books
for the most technical
people on the planet.**

GEEK GUIDES



**Download books for free with a
simple one-time registration.**

<http://geekguide.linuxjournal.com>

NEW!



Harnessing the Power of the Cloud with SUSE

Author:
Petros Koutoupis
Sponsor:
SUSE

NEW!



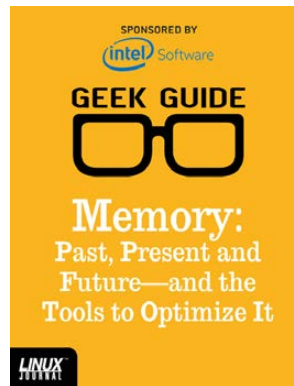
DevOps for the Rest of Us

Author:
John S. Tonello
Sponsor:
Puppet



An Architect's Guide: Linux for Enterprise IT

Author:
Sol Lederman
Sponsor:
SUSE



Memory: Past, Present and Future—and the Tools to Optimize It

Author:
Petros Koutoupis
Sponsor:
Intel



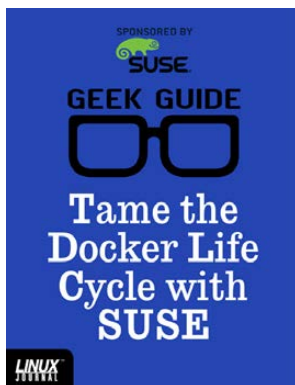
Cloud-Scale Automation with Puppet

Author:
John S. Tonello
Sponsor:
Puppet



Why Innovative App Developers Love High-Speed OSDBMS

Author:
Ted Schmidt
Sponsor:
IBM



Tame the Docker Life Cycle with SUSE

Author:
John S. Tonello
Sponsor:
SUSE



SUSE Enterprise Storage 4

Author:
Ted Schmidt
Sponsor:
SUSE

CONTENTS

JUNE 2017
ISSUE 278

FEATURES

74 BYOC: Build Your Own Cluster, Part II—Installation

Install Linux on an arbitrarily large number of computers with a push of a button.

Nathan R. Vance,
Michael L. Poublon
and William F. Polik

96 Testing the Waters: How to Perform Internal Phishing Campaigns

Are your users the weakest link in your anti-phishing strategies? Try Gophish and find out.

Jeremiah Bowling



COLUMNS

30 Reuven M. Lerner's At the Forge

Learning Data Science

36 Dave Taylor's Work the Shell

Analyzing Song Lyrics

42 Kyle Rankin's Hack and /

Update Tickets from the
Command Line

52 Shawn Powers' The Open-Source Classroom

Live Stream Your Pets
with Linux and YouTube!

116 Doc Searls' EOF

Open Source Comes of Age

IN EVERY ISSUE

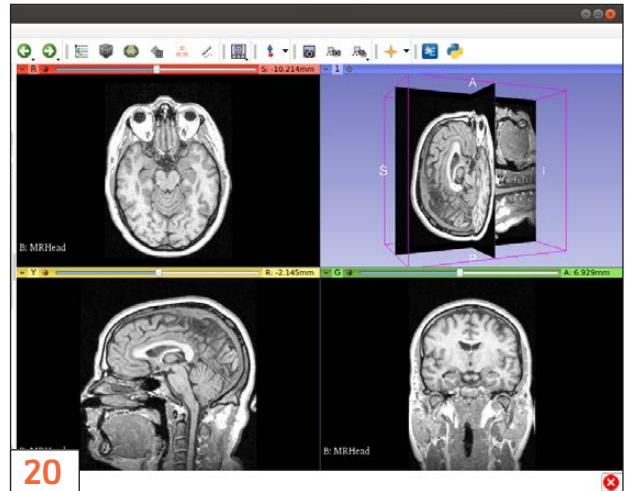
8 Current_Issue.tar.gz

10 UPFRONT

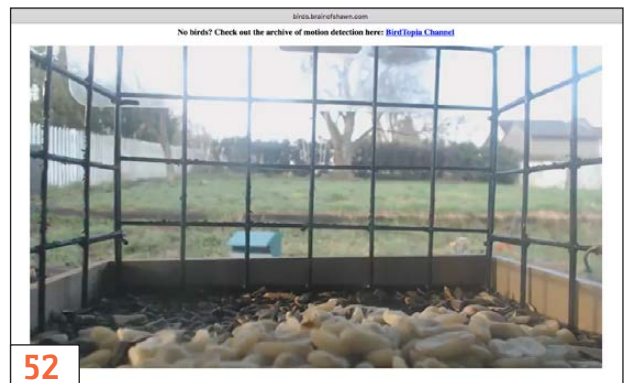
28 Editors' Choice

66 New Products

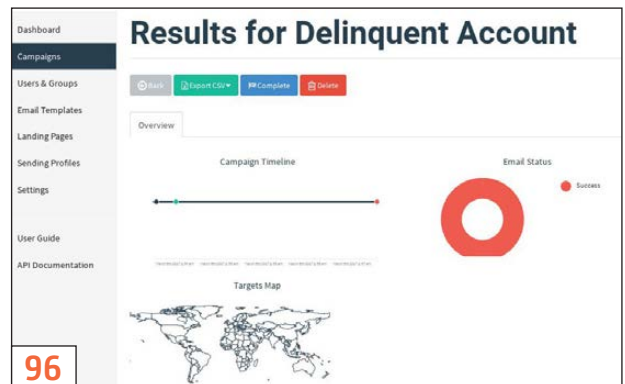
121 Advertisers Index



20



52



96

ON THE COVER

- Test Security with an Internal Phishing Campaign, p. 96
- Build Your Own Cluster: Installation, p. 74
- Update Your Ticketing System from the Command Line, p. 42
- Cool Project: Set Up YouTube Live Streams, p. 52
- EOF: Open Source Comes of Age, p. 116

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 • Adam Monsen

President Carlie Fairchild
publisher@linuxjournal.com

Publisher Mark Irgang
mark@linuxjournal.com

Associate Publisher John Grogan
john@linuxjournal.com

Director of Digital Experience 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

Nick Baronian
Kalyana Krishna Chadalavada
Brian Conner • Keir Davis
Michael Eager • Victor Gregorio
David A. Lane • Steve Marquez
Dave McAllister • Thomas Quinlan
Chris D. Stark • Patrick Swartz

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.

**STORAGE
REDEFINED:**

**You
cannot
keep up
with data
explosion.**

Manage data expansion with SUSE Enterprise Storage.

SUSE Enterprise Storage, the leading open source storage solution, is highly scalable and resilient, enabling high-end functionality at a fraction of the cost.

suse.com/storage



The Care and Maintenance of Penguins

I love classic Volkswagen Beetles. In fact, I own more than one. (It only makes sense to have one convertible and one sedan!) One of the best things about owning classic Volkswagens is that fixing and maintaining them is really simple. Kyle Rankin would agree with me on that, because he's also a Volkswagen fan, although his VW of choice is a Ghia. Nevertheless, when it comes to maintenance, those old rear-engine vehicles are simple and fun. Linux is a bit like that. It's not the sort of OS everyone uses; there are many friendly people online willing to help you fix things, and everyone who sees you use it will be jealous!

This month, Reuven M. Lerner starts things off by talking about Data Science. Technology and computers have become such an integral part of our lives, that data itself has become invaluable. Reuven explains what Data Science is all about, how it affects our world and provides many resources to help you learn more about it.

Dave Taylor follows up with a nifty article on extracting information about song lyrics using scripts. In true Dave Taylor fashion, he teaches how to pull data from an online source and analyze the



**SHAWN
POWERS**

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 email at shawn@linuxjournal.com. Or, swing by the [#linuxjournal](https://freenode.net) IRC channel on Freenode.net.



VIDEO:
Shawn Powers runs through the latest issue.

data while having fun along the way. Even if you're not interested in statistical information about 1960s artists' lyrics, you'll want to read his column this issue for the scripting tricks.

Kyle Rankin talks about the Jira ticketing system again this month, but rather than learning about the system itself, Kyle describes how to interact with the system via the command line. A GUI is nice, but if you're trying to update tickets while you're knee deep in terminal windows, it's nice to make a change without opening a web browser.

I go the opposite way in my column this month and explain how I stream my bird feeder cam to YouTube in order to provide a live stream that doesn't stress my bandwidth. I've talked about most of the tools separately, but in this article, I explain how to automate the live stream.

We pick up where we left off last month with Nathan R. Vance, Michael L. Poublon and William F. Polik teaching how to create your own computer cluster. Their last article covered setting up the hardware, and this month, they describe how to automate the installation process for individual nodes. Making the process fully automated requires a bit of work up front, but the authors walk through the process here step by step.

Last but not least, Jeremiah Bowling has a fascinating article on phishing your own users. No, he doesn't teach you how to steal your users' credit-card numbers, but rather he explains how to use the open-source Gophish package to test your users. Obviously, the process must be coupled with education in order for it to work, but coming up with phishing scenarios that are both safe and effective for testing is hard! I didn't know such a thing existed, but sure enough, it does, and Jeremiah explains how to use it.

We also have new product announcements, tech tips, cool apps and all the usual things you've come to expect in every issue of *Linux Journal*. So whether you're a fan of maintaining your Linux machines or just want to try the latest and greatest toys, we've got you covered. This is a fun issue, and we hope you enjoy it as much as we enjoyed putting it together! ■

[RETURN TO CONTENTS](#)



PREVIOUS
Current_Issue.tar.gz

NEXT
Editors' Choice



diff -u

What's New in Kernel Development

Some **PCI** devices include their own RAM, and **Logan Gunthorpe** wanted to make it available to the system as general-purpose memory. He understood that there could be a slowdown when using RAM from those devices relative to the RAM chips on the motherboard, but he figured that in cases of heavy load, it could be worth it. Sometimes what you really need is every last drop of memory, regardless of any other consideration.

He posted a patch to implement **p2pmem**, a peer-to-peer memory driver for PCI devices. But to avoid too much slowdown, he constrained his code to linking memory only from devices that all sat behind the same PCI switch.

But **Sinan Kaya** didn't like this, saying it wasn't a portable solution. He wanted Logan to remove any such restrictions and let users decide for themselves if the performance hit was too terrible, or if the code wouldn't work at all with a given device. That way, Logan's patch would work the same on all architectures.

They went back and forth about this. Logan felt it was important to ensure good performance, which required the code to include a certain amount of understanding of the hardware it controlled. The simplest

At Your Service

approach was to support PCI devices that were all behind the same switch; anything more generic than that, he said, risked exploding the complexity of the code, as well as the need to list tons of specific devices and their compatibility issues.

But ultimately, Sinan made the point that Logan's code simply could be generic and allow the users to shoot themselves in the foot if they so desired. Logan's patch would be off by default anyway, so there was no harm in letting users make the final call based on their own knowledge of the hardware on their systems.

This actually had been Logan's inclination from the start, but he'd received push-back from the **LSF** (load sharing facility) folks, who preferred things to be simple and functional. But with Sinan's argument about portability, Logan said it made more sense to remove the requirement that all shared memory devices be behind the same PCI switch, and just let users make the decision themselves.

The discussion ended there, but presumably the LSF folks will have their own objections, and the whole patch will have to go through several more iterations before everyone is fully satisfied, especially the kernel maintainers themselves.

Sometimes it's useful to have a whole separate thread running a particular kernel operation. If something is complicated and can take an arbitrary amount of time, giving it its own thread can fix latency issues by adding it to the normal process rotation in the scheduler. Then it can take however long it wants, without inconveniencing other parts of the kernel.

The **printk()** function is a good example of something that would benefit from having its

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an online digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your email address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly online: <http://www.linuxjournal.com/subs>. Email 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 online: <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/enewsletters>.

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.

own **kthread**. The `printk()` function sends messages to the console, to keep users informed of any problems with the system. Unfortunately, `printk()` works only in certain contexts and can take an arbitrary amount of time to execute. For people writing code in any given part of the kernel, it can be annoying to keep track of where and how to call `printk()` such that it will work. Putting `printk()` in its own kernel thread would simplify that whole question greatly.

Sergey Senozhatsky recently posted some code to do this, and although it did receive some initial interest, folks like **Peter Zijlstra** objected.

The problem with putting something into its own kernel thread is that each thread is a permanent drain on overall performance. The scheduler must cycle through every single thread, many times per second. As the number of threads on a system goes up, the system performance can become choppier and choppier. As a result, any new feature that requires a new kernel thread typically must have something really stupendous to offer. Maybe it resolves a security issue or makes people's lives much easier than they were before, or maybe it's something that just naturally belongs in a separate thread.

The `printk()` function may turn out to have a valid need for its own thread. But since `printk()` works fairly well as currently implemented, there are bound to be plenty of other folks like Peter who need to be convinced.

The **AVR32 Architecture** is being removed from the kernel. It's an old system-on-a-chip that came out of **Atmel corporation**. But it hasn't been supported in quite some time, and it's become a drag on other architectures that share drivers with it, such as the **Atmel ARM SoC**.

Hans-Christian Noren Egtvedt posted a patch to get rid of it, and a bunch of people cheered out loud. Various folks also suggested additional parts of the kernel source that could be included in the AVR32 wipeout.

In some ways, it's sad to lose an architecture like this. Sometimes it's fun to think about running a modern version of Linux on an old **TRS-80** color computer or whatnot. But, the kernel is a living project. And it's somewhat uplifting to remember that, aside from its mysterious absence from desktop systems, Linux does indeed essentially run the entire internet—everything connected to it, and

everything not connected to it. So I guess we can do without the AVR32 architecture.

Some things we don't like, and we can fix. Other things we don't like, but the fix is worse. Recently, **Christoph Hellwig** railed against the fact that system calls would accept any input flag at all and just ignore the ones that weren't supported. The reason the kernel does this is so that user code can run on older kernels without making the system calls choke on unknown input.

The bad part, unfortunately, is that it makes it impossible for user code to probe a system call to see whether a given feature is supported. And it turns out that some user code really would benefit from being able to do things like that—for example, atomic input/output.

One problem with fixing the system calls to reject unsupported input flags is that it would break existing binaries running in the world. All such binaries would need to be recompiled, which would be a problem if the binaries are very old and the source code is no longer available—which is actually a significant possibility in some cases.

Breaking existing binaries is called an **ABI** (application binary interface) change, and it's allowed only under very extreme circumstances—for example, if it's the only way to plug a given security hole.

But, **Linus Torvalds** didn't like Christoph's idea for another reason entirely. He said, "probing for flags is why we **could** add things like O_NOATIME etc - exactly because it 'just worked' with old kernels, and people could just use the new flags knowing that it was a no-op on old kernels."

So even though some users would benefit from being able to probe for features, even more users benefit from not having to worry about a given feature failing to do anything at all.—Zack Brown

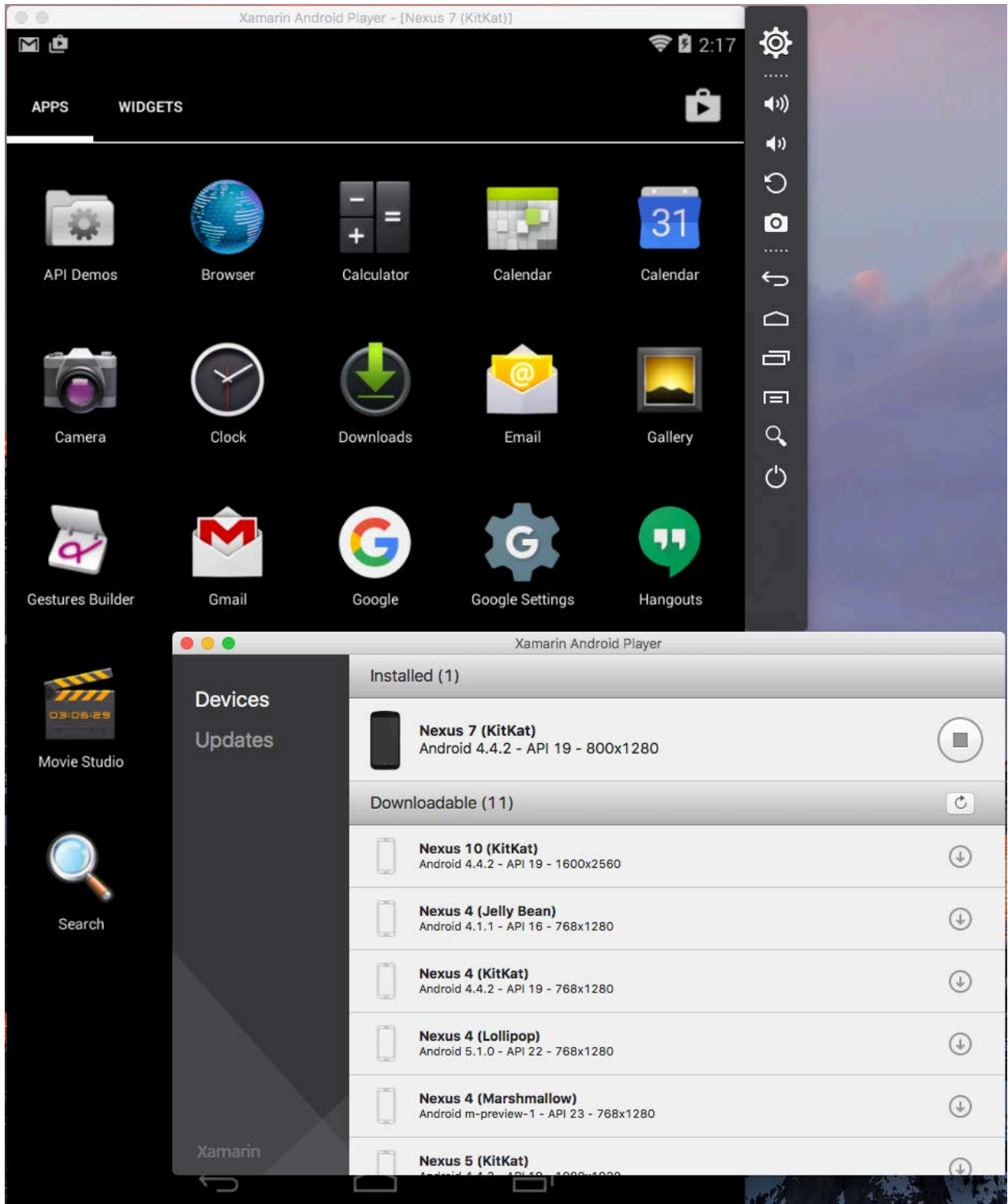
Android Candy: A Virtual Android

My phone is dead. I'm not exactly sure what happened to it, but for some reason, my beloved Sony Xperia Z5 Compact no longer turns on. Granted, it's not my main work phone, but it's my personal phone and also my audiobook player. The biggest problem is that when I'm exploring new Android apps, the Sony is the device I use for testing. Thankfully, there are other options.

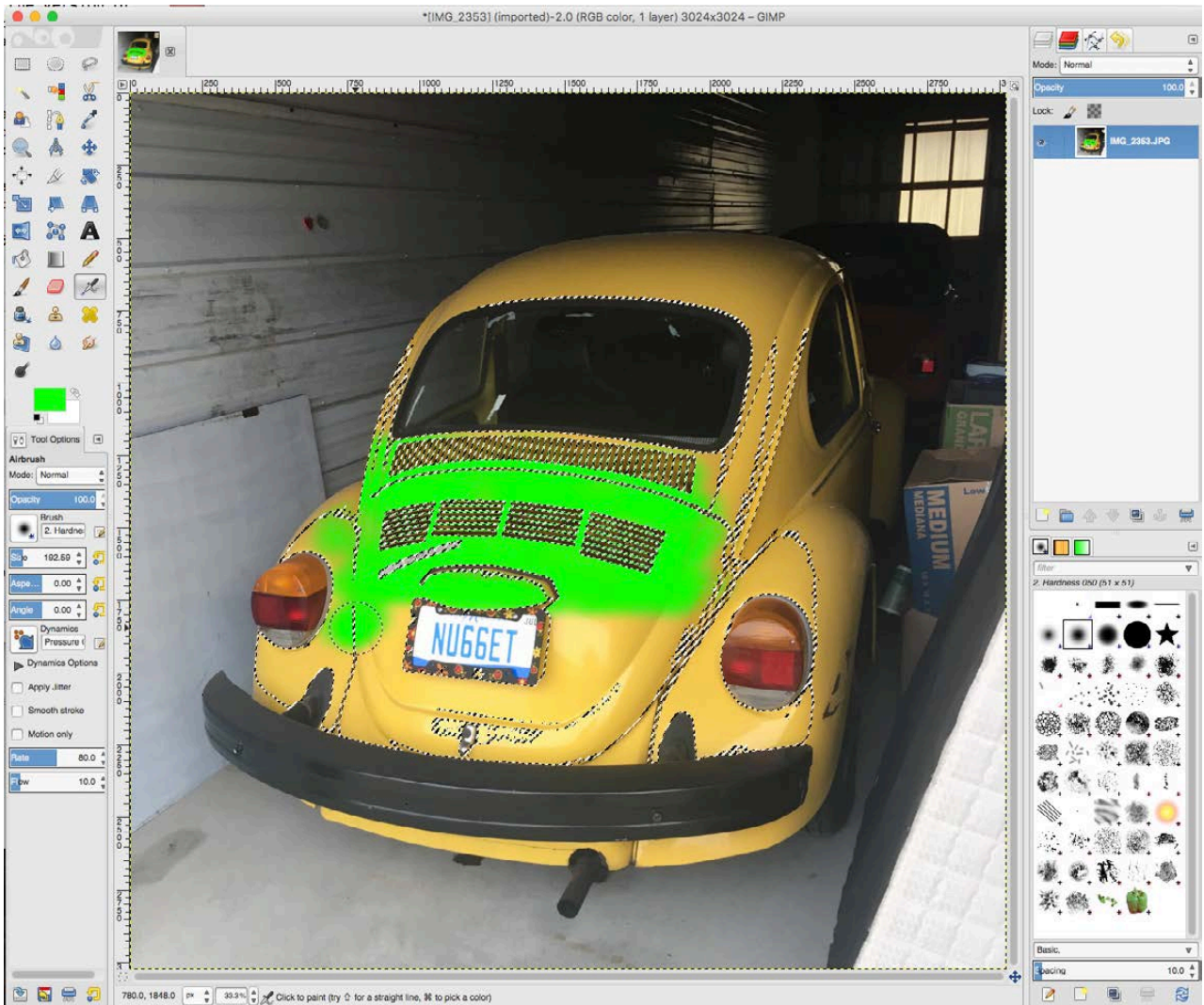
I really wanted a simple way to demo Android apps, and I figured an emulator would be the way to go. The problem is, running Android isn't as simple as booting an Ubuntu ISO in VirtualBox. Thankfully, it's actually not much more difficult than that either! There are several different ways to emulate Android on a computer, at least one of which actually does use VirtualBox. Since I'm currently sitting in front of an OS X machine, I needed something I could install easily on an Apple. Enter: Xamarin's Android Player. Only after I started using it did I discover that Xamarin is a Microsoft-owned company. For some reason, emulating Android on a Macintosh using Microsoft software made me giggle.

Anyway, if you want to run Android on Windows or OS X, head over to <https://developer.xamarin.com/releases/android/android-player> and grab a copy of Xamarin Android Player. The interface makes it easy to get the version of Android you want, and you even get a nice GUI for choosing the form factor of the Android device being emulated. Unfortunately, the Google Play Store doesn't come by default and must be added after installing Android.

Thankfully, the Xamarin folks made installing Google's apps easy. Just grab the proper version of gapps from <http://www.teamandroid.com/gapps> and drag the .zip file to the running Android emulator. The emulator will reboot, and Google Apps will be installed! It works great for testing and really well for demonstration too. I love that the interface allows for rotating the device as well, just like a phone or tablet would do. If you



need Android, but your phone has mysteriously stopped working, or if you just want to experiment with an Android emulator, Xamarin works well and is easy to install.—Shawn Powers



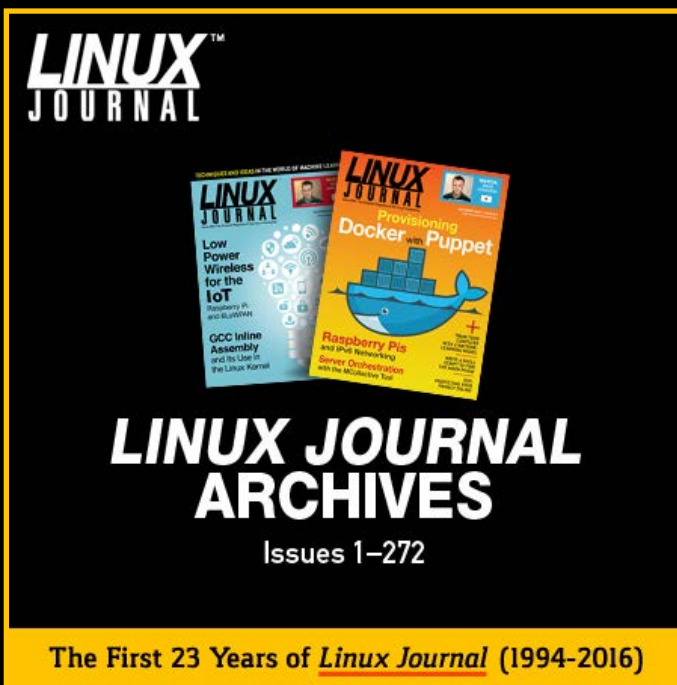
Non-Linux FOSS: GIMP, More Awesome Than I Remember

For what seems like decades, GIMP (Graphic Image Manipulation Program) has been the de facto standard image editor for Linux. It works well, has many features, and it even supports scripting. I always have found it a bit clumsy, however, and I preferred using something else for day-to-day work. I recently had the pleasure of sitting at a computer without an image editor though, so I figured I'd give GIMP another try on a

non-Linux operating system. See, the last time I tried to use GIMP on OS X, it required non-standard libraries and home-brew adding. Now, if you head over to <https://gimp.org>, you can download a fully native version of GIMP for Windows, OS X and Linux.

I'll be honest, just being able to install GIMP with a simple drag and drop on OS X was an improvement worth noting. When I actually started using it, however, I found that it's truly as powerful as Photoshop for the things I do. Granted, Photoshop likely has features advanced users might need that GIMP doesn't have, but everything I do with images was fully supported without exception. In the screenshot (from an OS X machine), you can see an example of me "painting" my Volkswagen using nothing more than the auto-selection tool and a virtual airbrush.

If you're looking for a great graphic editor and want to be consistent across platforms, GIMP is truly hard to beat. And if you tried it years ago but didn't really care for it, I urge you to give it another try. I'm shocked at how well it works, even on platforms other than Linux!—Shawn Powers



LINUX
JOURNAL

LINUX JOURNAL ARCHIVES
Issues 1–272

The First 23 Years of *Linux Journal* (1994–2016)

**Archive
1994–2016**
**NOW
AVAILABLE!**

SAVE \$10.00
by using
discount code
2017ARCH
at checkout.

Coupon code expires 6/28/2017

www.linuxjournal.com/archive

PoE, PoE+ and Passive POE

I've been installing a lot of POE devices recently, and the different methods for providing power over Ethernet cables can be very confusing. There are a few standards in place, and then there's a



This AP requires a Passive PoE 24v supply. It can be confusing, because even though it says it's PoE, it won't power on using a standard 802.3af switch.

method that isn't a standard, but is widely used.

802.3af or Active PoE: This is the oldest standard for providing power over Ethernet cables. It allows a maximum of 15.4 watts of power to be transmitted, and the devices (switch and peripheral) negotiate the amount of power and the wires on which the power is transmitted. If a device says it is PoE-compliant, that compliance is usually referring to 802.3af.

802.3at or PoE+: The main difference between PoE and PoE+ is the amount of power that can be transmitted. There is still negotiation to determine the amount of power and what wires it's transmitted on, but PoE+ supports up to 25.5 watts of power. Often, access points with multiple radios or higher-powered antennas require more power than 802.3af can supply.

Passive PoE: This provides power over the Ethernet lines, but it doesn't negotiate the amount of power or the wires on which the power is sent. Many devices use Passive PoE (notably, the Ubiquiti line of network hardware often uses 24v Passive PoE) to provide power to remote devices. With Passive PoE, the proprietary nature of the power specifics means that it's often wise to use only power injectors or switches specifically designed for the devices that require Passive PoE. The power is "always on", so it's possible to burn out devices if they're not prepared for electrified Ethernet wires, or if the CAT5 cabling is wired incorrectly.

The best practice for using power over Ethernet is either to use equipment that adheres to the 802.3af/at standards or to use the power injectors or switches specifically designed for the hardware. Usually, the standard-based PoE devices are more expensive, but the ability to use any brand PoE switch and device often makes the extra expense worthwhile. That said, there's nothing wrong with Passive PoE, as long as the correct power is given to the correct devices.—Shawn Powers

Slicing Scientific Data

In previous articles, I've covered scientific software that either analyzes image information or actually generates image data for further analysis. In this article, I introduce a tool that you can use to analyze images generated as part of medical diagnostic work.

In several diagnostic medical tests, complex three-dimensional images are generated that need to be visualized and analyzed. This is where 3D Slicer steps into the workflow. 3D Slicer is a very powerful tool for dissecting, analyzing and visualizing this type of complex 3D imaging data. It is fully open source, and it's available not only on Linux, but also on Windows and Mac OS X.

It's also built as a core program with a plugin architecture. This means you can add extra functionality to do completely new analysis.

Although 3D Slicer was written to handle medical images, the development team has been very careful to say that the software has not been approved for clinical use and shouldn't be used for diagnostic work. It's intended to be a research tool—hence its open-source license and plugin architecture, which aid in working with newly created algorithms and developing the next-generation tools that will be incorporated into diagnostic software.

Installation involves downloading a file directly from the project website (<https://www.slicer.org>). For Linux, this file is a gzipped tarball. You can select between a stable release or a nightly release. Once you download the tarball, you can unpack it with the command:

```
tar xvzf Slicer-4.6.2-linux.amd64.tar.gz
```

This unpacks everything into a subdirectory named Slicer-4.6.2-linux-amd64. Of course, the 4.6.2 portion will be different if you download a different version.

Once you have everything untarred, you can run it with:

```
./Slicer-4.6.2-linux-amd64/Slicer
```

When it starts, you end up with an empty project (Figure 1).

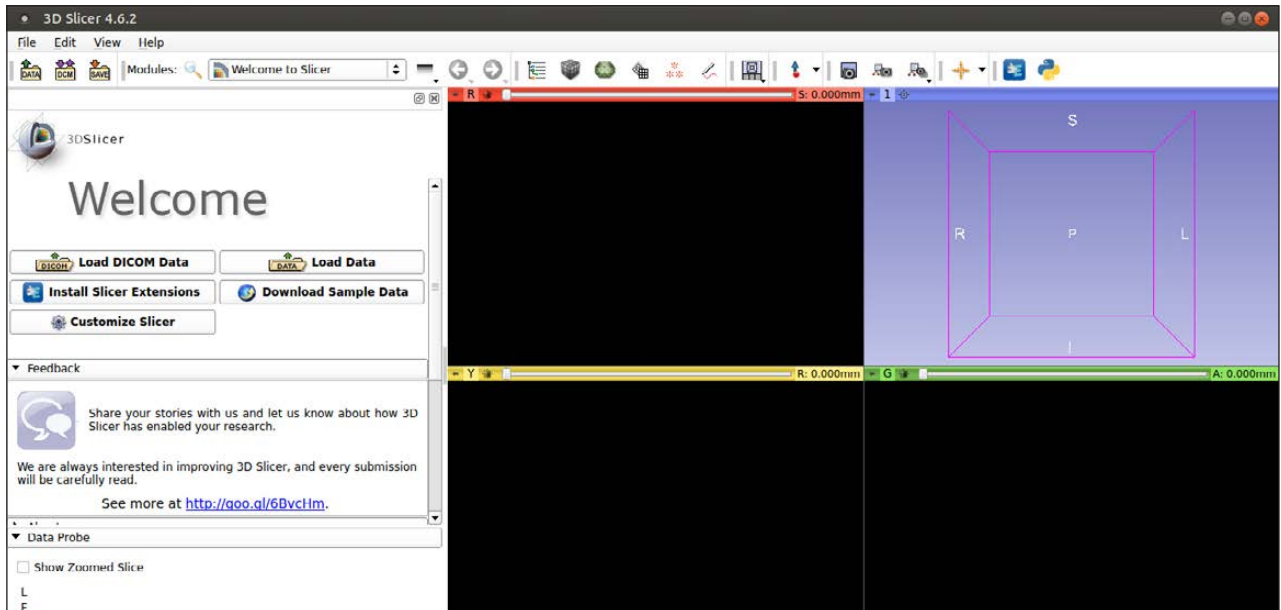


Figure 1. When you first start 3D Slicer, you get a display of an empty project, ready to start your work.

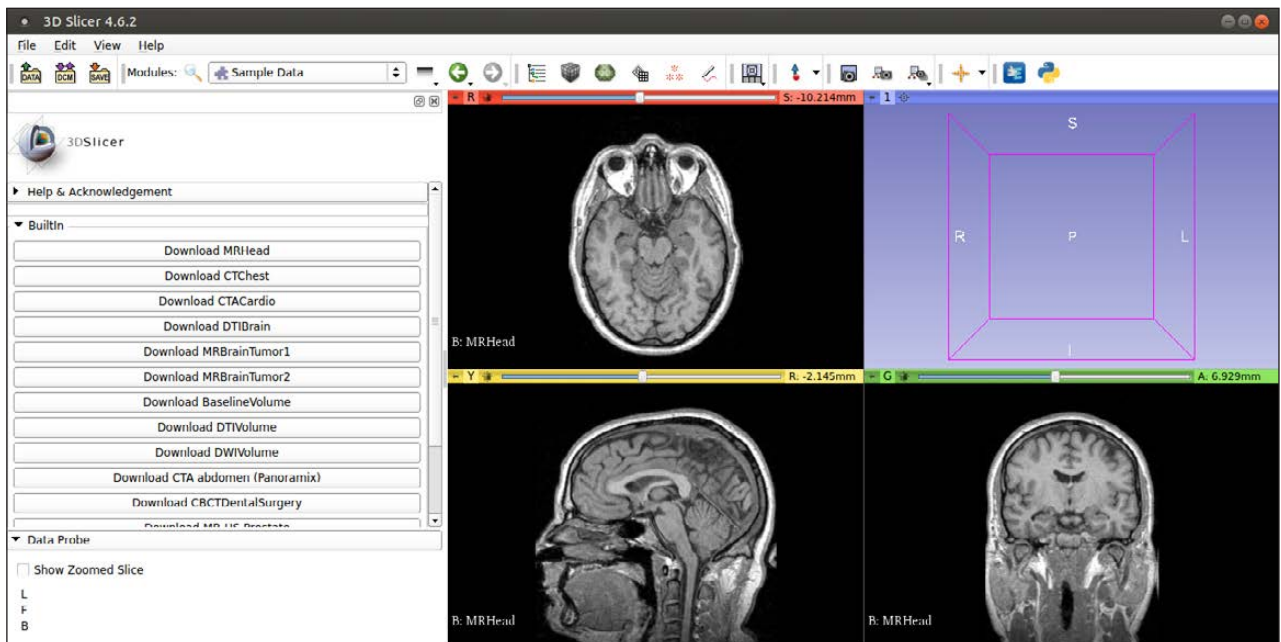


Figure 2. Several sample data sets are available for you to learn with, such as this head MRI data.

If you're trying to learn how to use 3D Slicer, you may not have any data to work with at first. Luckily, there is a button on the main window that allows you to download sample data. When you click it, you get a list of potential sample data sets available for download.

For this article's example, click the Download MRHead button and use the related data set (Figure 2).

Once downloaded, it's loaded automatically, and you can see the results in the three 2D slice viewing windows. A fourth window is used for 3D rendering, however. In order to get an image rendered there, you need to hover over the pin icon in the top left-hand corner of one of the 2D panes. Once you do, a small popup window appears where you can select a link icon to tell 3D Slicer to link all three slices together. You then can click the eye icon beside the link icon to tell 3D Slicer to render the 3D view of the image data (Figure 3).

You can manipulate this rendered image with your mouse, which allows you to rotate it or change the zoom level. There also are several built-in visualization options, which are available by clicking the pin icon at the top left-hand corner of the 3D pane. Doing so pops up a new window where you can manipulate the 3D image, including setting the zoom level and what labels are displayed, and you even can make the image rotate automatically (Figure 4).

Simply viewing the image data is not the only thing you likely will want to do as far as analyzing your data. This is where 3D Slicer's plugin

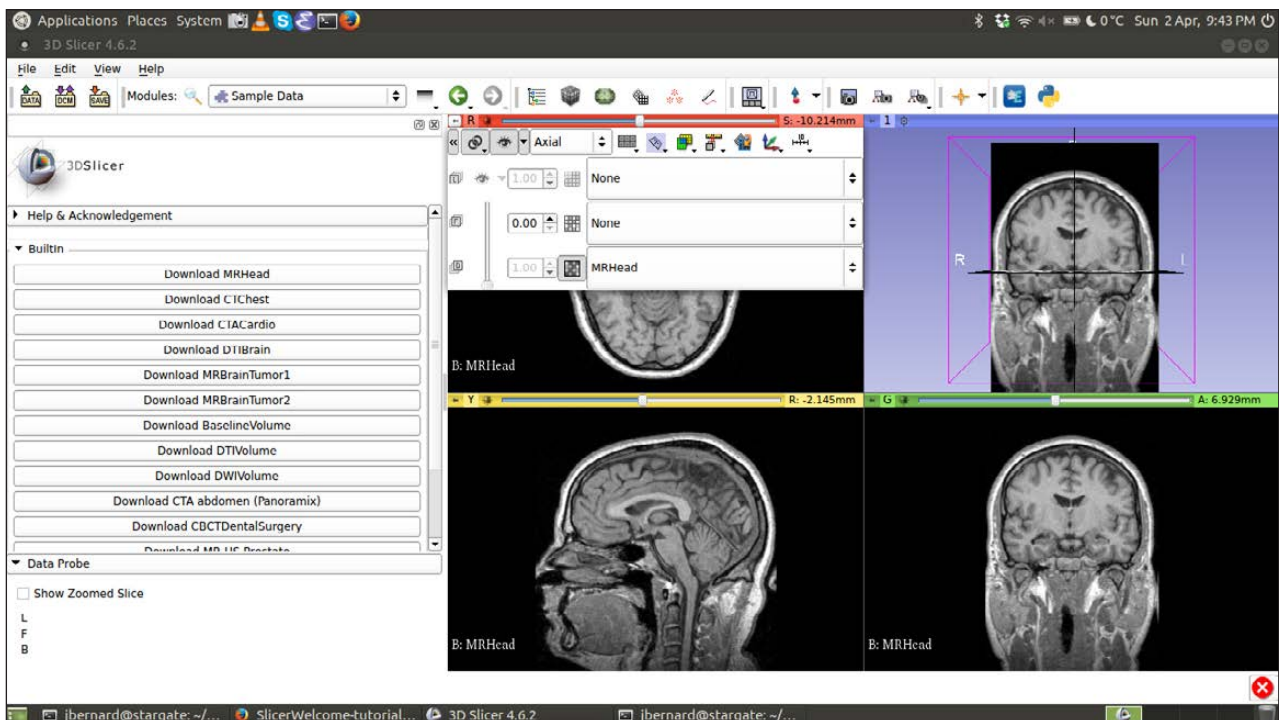


Figure 3. You can get a 3-D rendering of the imaging data for alternative analysis options.

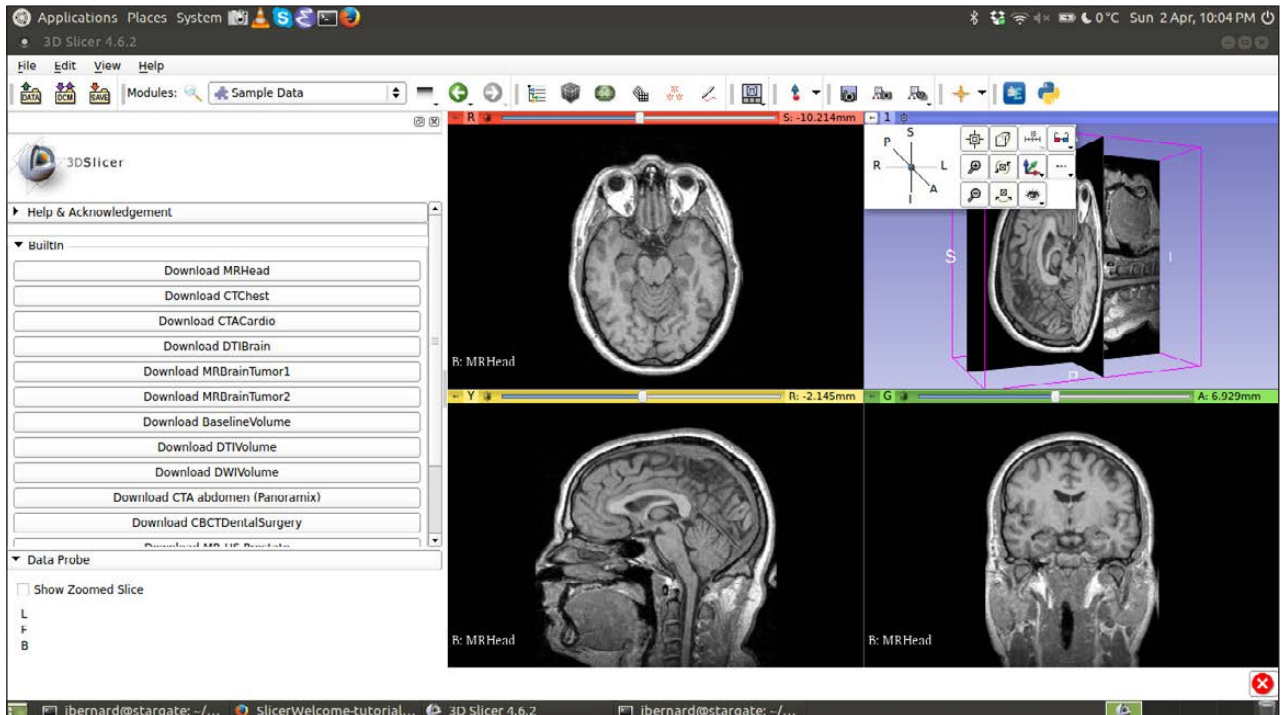


Figure 4. You can use several built-in functions to manipulate the 3D rendering of your imaging data.

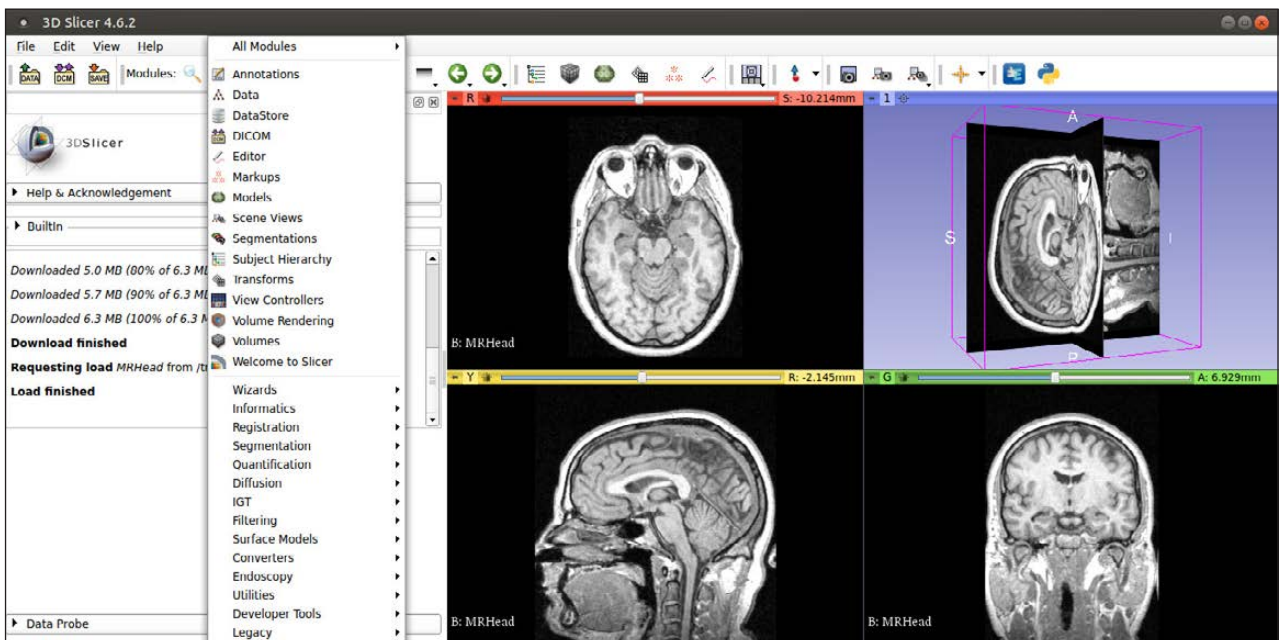


Figure 5. A rather large collection of modules is available for data analysis.

architecture really shines. More than 100 modules are available in 36 different categories. You can find them in the dropdown box in the center at the top of the window (Figure 5).

This is where a lot of the real work gets done. As an example, say you wanted to apply an island removal filter to your image. You can select this option from the modules drop-down list, which adds a new entry within

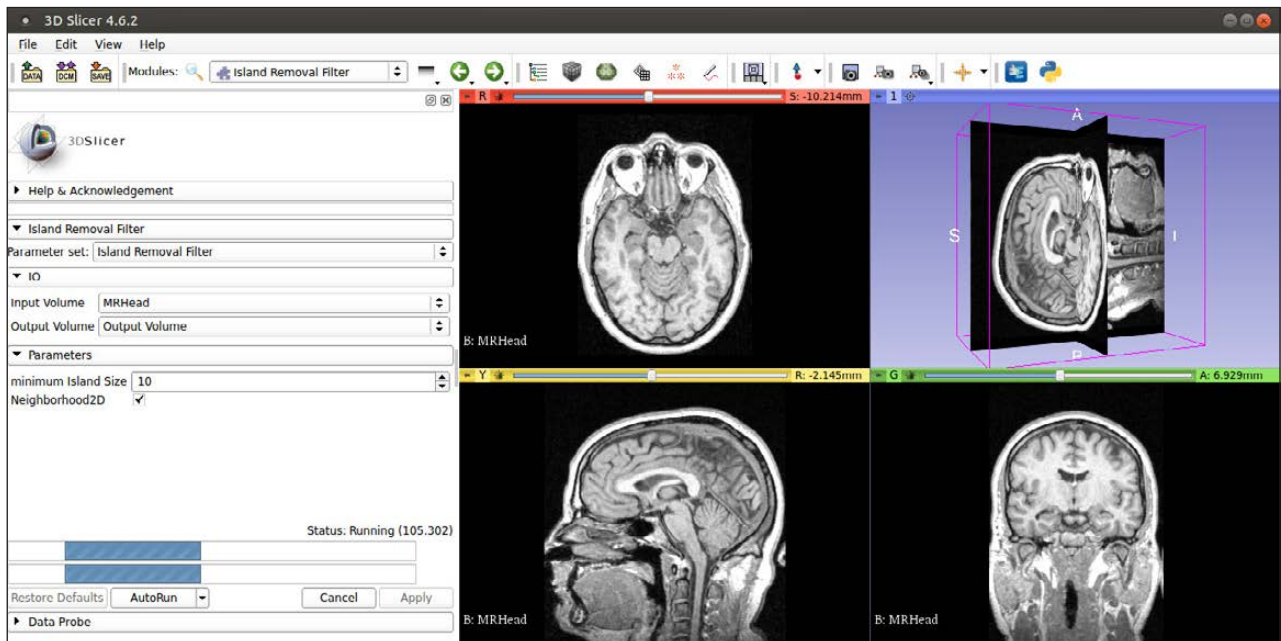


Figure 6. Activating the island removal module opens an options pane on the left-hand side.

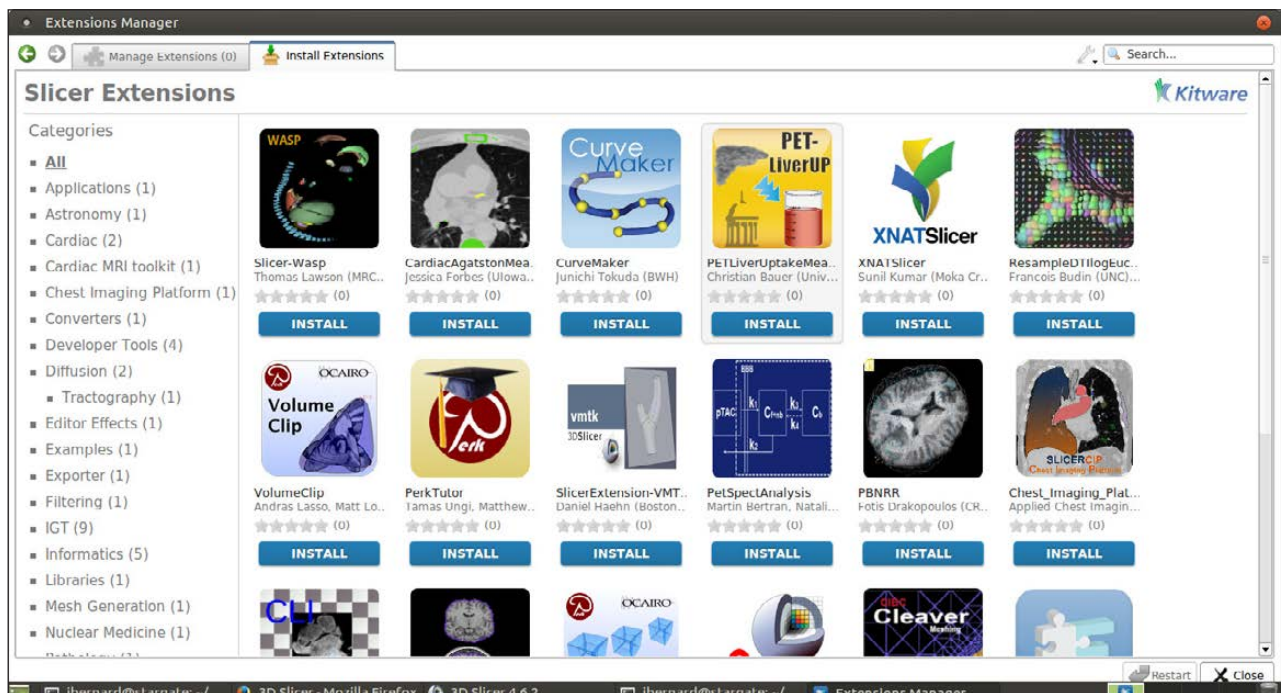


Figure 7. You can add and remove a large number of extra modules with the Extension Manager.

the left-hand pane. This is where you can select the required options, such as the input and output volumes, and the minimum island size (Figure 6). You then can click apply and let your computer run the process.

What if the modules included with the default installation don't do what you need 3D Slicer to do? Click the menu item View→Extension Manager to pop up a new management window (Figure 7).

Installing a new module is as simple as clicking the install button. Once you do, you may need to restart 3D Slicer before the new module is available to use. You can uninstall any modules no longer needed by selecting the Manage Extensions tab in the Extension Manager.

Because so much work has been put into managing and manipulating three dimensional image data, 3D Slicer's capabilities have started to be used in other problem domains. As an example, there's a module named SlicerAstro that you can use to handle astronomical image data. It includes a number of sample data sets for exploring the functions available within SlicerAstro.

Loading one of the sample data sets provides output that is very similar to that which you saw above. Selecting the module drop-down list and clicking the Astronomy→Welcome to SlicerAstro entry pops up new information within the left-hand pane (Figure 8).

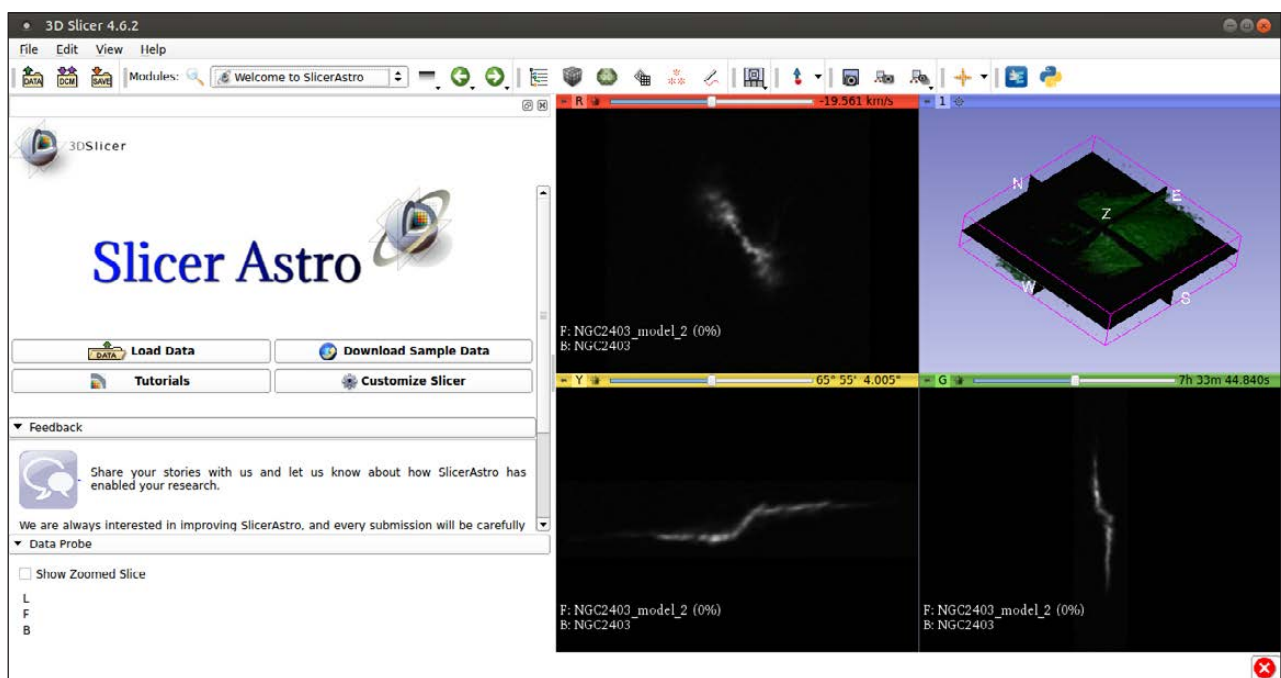


Figure 8. You can get extra information for newly installed modules, such as SlicerAstro.

Here you can download more sample data or get access to tutorials on how to use the SlicerAstro module. This is just one example of how you may want to extend 3D Slicer into your own problem domain of three-dimensional image analysis.

If you have complex imaging data that needs to be processed, hopefully this short introduction to 3D Slicer has provided a new option you may not have encountered before. It's heavily used in research applications, and with the ability to write your own extensions, it should be able to handle almost any work you want to throw at it. Just be aware that it has not been approved to do any diagnostic work. Also, note that a large number of tutorials are available online, covering many different problem domains. A little bit of Google-Fu should help you find examples to get started.
—Joey Bernard

[RETURN TO CONTENTS](#)

THEY SAID IT

Success is not the result of spontaneous combustion. You must set yourself on fire.
—Reggie Leach

Adversity does teach who your real friends are.
—Lois McMasters Bujold

Happiness is not a station you arrive at, but a manner of traveling.
—Margaret Lee Runbeck

Make sure to be in with your equals if you're going to fall out with your superiors.
—Jewish Proverb

Laugh at yourself first, before anyone else can.
—Elsa Maxwell



WOMEN IN TECHNOLOGY SUMMIT

JUNE 11-13, 2017 AT THE
DOUBLETREE BY HILTON IN
SAN JOSE, CA.

At the 23rd Annual Women in Technology Summit, executives, entrepreneurs, and technology thought leaders from around the world converge to **make things happen**.

FEATURED SPEAKERS

- Beena Ammanath, Vice President, Innovation, GE Digital
- Heather Furby, CEO, Creative Age Leadership
- Will Marré, Founder, SMART Power Academy
- Amanda Healy, Senior Marketing Manager and Social Evangelist, TIBCO Software
- Katie Broderick, Research Vice President, 451 Research
- Diana Kelley, Global Executive Security Advisor, IBM

Explore new business opportunities that underscore how technology is powering change; collaborate with your peers on innovative solutions to a variety of common business challenges; build and expand strong connections in a welcoming environment of women and men committed to each other's success.

 WITI.COM/SUMMIT  [@WITI](https://twitter.com/WITI)

 FACEBOOK.COM/WOMENINTECH

 YOUTUBE.COM/WOMENINTECHNOLOGY



SPECIAL OFFER

Use promo code **LFSUMMIT17** and **SAVE \$300** off registration!

◀ PREVIOUS
UpFront

NEXT ▶
Reuven M. Lerner's
At the Forge

The Wire

In the US, there has been recent concern over ISPs turning over logs to the government. During the past few years, the idea of people snooping on our private data (by governments and others) really has made encryption more popular than ever before. One of the problems with encryption, however, is that it's generally not user-friendly to add its protection to your conversations. Thankfully, messaging services are



starting to take notice of the demand. For me, I need a messaging service that works across multiple platforms, encrypts automatically, supports group messaging and ideally can handle audio/video as well. Thankfully, I found an incredible open-source package that ticks all my boxes: Wire.

There are some other great software packages for encrypting conversations. Programs like Signal do end-to-end encryption, but fall short when it comes to



audio and video. Telegram is great for sending encrypted file transfers, but it doesn't handle direct communication. Thankfully, Wire not only encrypts text, video, audio and media, but it also does end-to-end encrypted group interactions. Plus, it has clients for just about any platform imaginable.

Users have a user name that is identified much like a Twitter handle. My account, for instance, is @shawnp0wers.

And since Wire is open source, there aren't any targeted

ads, popups or banners. It's just encrypted communication done in a convenient way. In fact, thanks to ticking all the boxes in a communication client for me, Wire is receiving this month's Editors' Choice award. Check it out today at <http://wire.com>.


—Shawn Powers

Available for many platforms and devices.

iOS

[GET THE APP](#)

iOS 8.0 or above




[GET THE APP](#)

Android 4.2 or above · [APK](#) · [Details](#)

macOS


[GET THE APP](#)

macOS 10.9 or above




[GET THE APP](#)

Windows 7 or above · [Details](#)



[OPEN IN WEB](#)

Chrome, Firefox, Edge and Opera



[GET SOURCE CODE](#)

✓ [Get Binary](#)

— 2.13.2741 —

- Ubuntu (64bit)
- Ubuntu (32bit)
- Install via Debian-Repository
- AppImage (64bit)
- AppImage (32bit)

RETURN TO CONTENTS

Learning Data Science

Data science is big. If you want to learn it, where do you start?



**REUVEN M.
LERNER**

Reuven M. Lerner, a longtime Web developer, offers training and consulting services in Python, Git, PostgreSQL and data science. He has written two programming ebooks (*Practice Makes Python* and *Practice Makes Regexp*) and publishes a free weekly newsletter for programmers, at <http://lerner.co.il/> newsletter. Reuven tweets at @reuvenmlerner and lives in Modi'in, Israel, with his wife and three children.

◀ PREVIOUS
Editors' Choice

NEXT
Dave Taylor's
Work the Shell ▶

IN MY LAST FEW ARTICLES, I've written about data science and machine learning. In case my enthusiasm wasn't obvious from my writing, let me say it plainly: it has been a long time since I last encountered a technology that was so poised to revolutionize the world in which we live.

Think about it: you can download, install and use open-source data science libraries, for free. You can download rich data sets on nearly every possible topic you can imagine, for free. You can analyze that data, publish it on a blog, and get reactions from governments and companies.

I remember learning in high school that the difference between freedom of speech and freedom of the press is that not everyone has a printing press. Not only has the internet provided everyone with the equivalent of a printing press, but it has given us the

power to perform the sort of analysis that until recently was exclusively available to governments and wealthy corporations.

During the past year, I have increasingly heard that data science is the sexiest profession of the 21st century and the one that will be in greatest demand. Needless to say, those two things make for a very appealing combination! It's no surprise that I've seen a major uptick in the number of companies inviting me to teach on this subject.

The upshot is that you—yes, you, dear reader—should spend time in the coming months, weeks and years learning whatever you can about data science. This isn't because you will change jobs and become a data scientist. Rather, it's because everyone is going to become a data scientist. No matter what work you do, you'll be better at it, because you will be able to use the tools of data science to analyze past performance and make predictions based on it.

Back when I started to develop web applications, it was the norm to have a database team that created the tables and queries. Nowadays, although there certainly are places that have a full-time database staff, the assumption is that every developer has at least a passing familiarity with relationship (or even NoSQL) databases and how to work with them. In the same way that developers who understand databases are more powerful than those who don't, people in the computer field who understand data science are more powerful than those who don't.

There is a bit of bad news on this front, though. If you thought that the pace of technological change in programming and the web moved at a breakneck pace, you haven't seen anything yet! The world of data science—the tools, the algorithms, the applications—are moving at an overwhelming speed. The good news is that everyone is struggling to keep up, which means if you find yourself overwhelmed, you're probably in very good company. Just be sure to keep moving ahead, aiming to increase your understanding of the theory, algorithms, techniques and software that data scientists use.

Where should you start? In this article, I describe some of the resources I've found to be the most helpful as I've been diving deeper and deeper into data science.

Statistics

There's no way around it. If you're going to do data science, you're going to need to learn some statistics. I took a year of it in graduate school, and then I did some analysis as part of my dissertation, but there's a lot I don't know, so I've been trying to improve my understanding. Every little bit helps! Whether you're simply learning Bayes' Theorem, figuring out how linear regression works or learning how to modify your data to minimize errors, statistics is a crucial part of this.

So, where do you start? There are a number of courses, often for free or at very low cost, at edX, Udemy and Coursera. A particularly popular introduction to machine learning, which includes the basic statistical knowledge you'll need, is taught by Stanford professor Andrew Ng via Coursera. If you're looking for something more hard-core, I definitely recommend the Udemy courses by LazyProgrammer.

Two good and standard textbooks on the subject are *An Introduction to Statistical Learning* (by James, Witten, Hastie and Tibshirani) and *Elements of Statistical Learning* (by Hastie, Tibshirani and Friedman). Both books are published by Springer, and both are available in PDF form, as free downloads: <http://www.springer.com/us/book/9781461471370> (*An Introduction to Statistical Learning*) and <http://statweb.stanford.edu/~tibs/ElemStatLearn> (*Elements of Statistical Learning*). You probably should download and read those books; over time, the ideas and methods they describe will help you to reason about what you're doing.

I also want to recommend the various books and courses offered by Jason Brownlee at his site <http://machinelearningmastery.com>. His writing is clear, and he tries to be very practical about what he shows you. Especially if you're using Python for machine learning, his books are a great way to get started and improve your understanding.

Note that you definitely should not wait until you have read through books, watched lectures and taken courses to start playing with machine learning. That would be akin to saying you should try to learn a language only after you have mastered its grammar. As with language, you should be trying to use it at the same time that you're learning how it works.

Along with understanding the math, it's also important to have a good

skeptical, statistical look at the world. Jake VanderPlas has a talk called “Statistics for Hackers” that not only translates the mathematical ideas into code, but it also concentrates on the aspects that are most likely to be of interest in data science.

Two other books worth mentioning are *The Cartoon Guide to Statistics* (by Larry Gotnick and Woollcott Smith) and *Statistics Done Wrong* (by Alex Reinhart). Both books are good for getting you to think in this way—by which I mean, when someone presents you with data, or if you are about to present others with data, you’ll at least find some of the holes in the argument or alternative explanations to yours.

Data Science Theory

Although statistics certainly is an important part of data science, it’s not the only part. Indeed, there are a number of model types that aren’t statistical, such as K Nearest Neighbors.

Knowing the different types of algorithms that are available, when each is appropriate and how to tweak them will be invaluable. In many cases, you’ll just want to throw a bunch of algorithms at the problem—and if your data set is small and/or easy to understand, that’ll be just fine. But if it takes a long time to train your model, trying a dozen different algorithms is neither smart nor effective. Just as an expert cook knows which knife to use, and a good programmer should know which language is appropriate for a given task, someone building machine learning models should know which algorithms are more likely to be useful. (It’s not always 100% obvious, but you do want to narrow down your starting set.)

In addition to the books I mentioned above, some others are well worth reading and reviewing. *Doing Data Science* by Cathy O’Neil and Rachel Schutt, as well as the *Python Data Science Handbook* by Jake VanderPlas, introduce the ideas behind data science, but they also include working code and examples that you can and should play with.

A phenomenal resource is the Analysis Vidhya site (<http://analyticsvidhya.com>) that summarizes, describes and instructs in a truly staggering number of technologies, algorithms and theories. Daily email messages from this site always are interesting and useful—and, quite frankly, overwhelming in their number and scope.

Data Science Hacking

Although statisticians have been using software for many years, one of the key differences between statistics and data science is that the latter requires programming knowledge. It's no surprise, given its shallow learning curve and huge, friendly community, that Python has become the leading language for data science. If you choose to use Python (which I definitely recommend), you'll need to learn a number of libraries that don't always adhere to the standard Python way of doing things: NumPy and Pandas provide data structures, and then there's also scikit-learn, which provides the algorithms and supports for machine learning.

The websites for each of these packages, but especially scikit-learn, are huge, and they likely will make you think you never can learn it all. And indeed, no one is expecting you to know everything that those packages can do by heart. But over time, you will be expected to understand more and more algorithms and ideas, and also how to implement them.

If you're using Python, the the Jupyter notebook is likely to be your day-to-day tool of choice. Jupyter (<http://jupyter.org>) continues to expand in impressive functionality, with new versions released every few weeks. If you're new to Python or to dynamic languages in general, Jupyter can feel a bit odd, but it quickly grows on you and will become a fluid part of your day-to-day work.

As you can see, it's important to practice. I often say that programming languages are like human (natural) languages, in that you need to practice using them to gain true fluency. Data science is the same, but it's also different, in that you need fluency in several related disciplines in order to succeed.

Fortunately, the world of data science is large and growing, providing a lot of interesting data sets for people to analyze, both for fun and practice, and also for serious use. "I Quant NY" (<http://iquantny.tumblr.com>) is a blog that not only provides interesting information about New York City from city-supplied data sets, but it also shows how data scientists can ask questions and provide answers that affect many people. If you're looking for data sets, it's hard to know just where to start or what sort of analysis might be most appropriate. The weekly newsletter "Data is Plural" by Jeremy Singer-Vine (<https://tinyletter.com/data-is-plural>), the "data sets"

subreddit (<https://www.reddit.com/r/datasets>) and the new website Data.World (<http://data.world>) all offer a staggering number of data sets on a variety of topics. Choose something that's of interest to you, and see what questions you can ask and answer.

I would be remiss if I didn't mention a few of the podcasts to which I listen. Not only do they provide me with the latest news, information, anecdotes and updates from the world of data science, they also allow me to understand the trends better—for example, in favor of neural networks and deep learning. “Partially Derivative” (<http://partiallyderivative.com>) and “Linear Digressions” (<http://lineardigressions.com>) are my two favorites, but there are some others, such as “Data Science at Home” (<http://worldofpiggy.com/podcast>) and “Data Skeptic” (<https://www.dataskeptic.com/>). Podcasts aren't going to help you to code better; only more coding can really do that. But they will give you perspective and understanding that make the code more obvious.

Finally, although I believe that data science is changing our world for the better, we do need to be on the lookout for potential issues. Cathy O'Neil's book, “Weapons of Math Destruction”, is a must-read for anyone entering this world. Even if you aren't writing algorithms that will affect millions of people, awareness of our biases as humans, and of our need to be transparent when implementing policy via machine, is an important one. This easily is one of the best books I've read in the last few years.

I'll definitely return to data science topics in the future, given its importance to developers. But for my next article, I plan to return to the world of web applications and databases, looking at the languages, libraries and packages we use to create modern applications. ■

Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

[RETURN TO CONTENTS](#)

Analyzing Song Lyrics

How many times did The Beatles use the word love?



DAVE TAYLOR

Dave Taylor has been hacking shell scripts on UNIX and Linux systems for a really long time. He's the author of *Learning Unix for Mac OS X* and *Wicked Cool Shell Scripts*. You can find him on Twitter as @DaveTaylor, or reach him through his tech Q&A site: <http://www.AskDaveTaylor.com>.

PREVIOUS

◀ Reuven M. Lerner's
At the Forge

NEXT

Kyle Rankin's
Hack and / ▶

I WAS READING ABOUT THE HISTORY OF THE BEATLES A FEW DAYS AGO AND BUMPED INTO AN INTERESTING FACT.

According to the author, The Beatles used the word "love" in their songs more than 160 times. At first I thought, "cool", but the more I thought about it, the more I became skeptical about the figure. In fact, I suspect that the word "love" shows up considerably more than 160 times.

And, this leads to the question: how do you actually figure out something like that? The answer, of course, is with a shell script! So let's jump in, shall we?

Download Lyrics by Artist

The first challenge, and really most of the work, is figuring out where to download the lyrics for every song by an artist, performer or band. There are lots of online archives, but are they complete?

WORK THE SHELL

One source is MLDB, the Music Lyrics Database (modeled after the Internet Movie Database, one presumes). An easy test is this: how many songs does the site list for The Beatles?

Working backward from an interactive session in a web browser, an artist search for “the beatles” produces eight pages of matches, 30 matches per page. That’s 240 songs. Wikipedia says that there are 237 original compositions for the band, and BeatlesBible.com shows 302 original songs. Confusing!

Of course, some of the songs recorded by The Beatles didn’t have lyrics. For example, on the *Magical Mystery Tour* album, there’s a track called “Flying”. Given that Paul McCartney and John Lennon were such brilliant lyricists, however, the vast, vast majority of songs recorded have at least some lyrics—even “The End”.

So let’s go with MLDB and trust that its 240 songs are close enough for this task. Now the challenge is to get a list of all the songs, and then to download the lyrics for each and every song that matches.

Fortunately, that can be done by reverse-engineering the search URLs. The second page of results for an exact-phrase artist search for “the beatles” sorted by rating produces this particular URL: <http://www.mldb.org/search?mq=the+beatles&mm=2&si=1&ob=2&from=30>.

You can experimentally verify that this produces the second page of results, but hey, let’s just run with it! Since the second page has a “from=30”, you can conclude that there are 30 entries per page (as mentioned earlier) and that from=60 gets page three, from=90 page four, and so on.

Each page can be downloaded in HTML form using `GET` or `curl`, with my preference being to use the latter—it’s more sophisticated and has oodles of options. A quick glance shows that “Yellow Submarine” shows up on the first page, so here’s a quick test, with `url` set to the value shown above:

```
$ curl -s "$url" | grep "Yellow Submarine"
<table id="thelist" cellspacing="0"><tr><th>Artist(s)</th><th>Song</th>
<th width="20">Rating</th></tr><tr class="h"><td class="fa"><a
href='artist-39-the-beatles.html'>The Beatles</a></td><td class="ft"><a
href="song-32476-i-am-the-walrus.html">I Am The Walrus</a></td><td
align="right">6</td></tr><tr class="n"><td class="fa"><a
```

WORK THE SHELL

```
href='artist-39-the-beatles.html'>The Beatles</a></td><td class="ft"><a
href="song-32461-yellow-submarine.html">Yellow Submarine</a></td><td
align="right">6</td></tr><tr class="h"><td class="fa"><a
href='artist-39-the-beatles.html'>The Beatles</a></td><td class="ft"><a
href="song-32585-day-tripper.h...
```

It turns out that the entire table of lyrics is a single line of HTML. That's a drag, but easily managed. Notice above that the href link to the individual song is of the form:

```
<a href="song-32461-yellow-submarine.html">Yellow Submarine</a>
```

That's the pattern I'm going to seek out in the raw HTML, noting that the links to the artist have a single quote, but the links to the lyrics are using double quotes:

```
curl -s "$url" | grep "Yellow Submarine" | sed 's/</\<br>
</g' | grep 'href="song-'
```

Notice the sed pattern above. I'm replacing every < with a carriage return followed by the < so that the net effect is that I unwrap the HTML source neatly and then can use grep to isolate the matching lines and exclude everything else.

That line alone gets the following:

```
<a href="song-32476-i-am-the-walrus.html">I Am The Walrus
<a href="song-32461-yellow-submarine.html">Yellow Submarine
<a href="song-32585-day-tripper.html">Day Tripper
<a href="song-32520-come-together.html">Come Together
. . . lots of lines removed for clarity . . .
<a href="song-32395-a-hard-day-s-night.html">A Hard Day's Night
<a href="song-32571-i-want-to-hold-your-hand.html">I Want To Hold
Your Hand
<a href="song-32527-here-comes-the-sun.html">Here Comes The Sun
<a href="song-32609-i-saw-her-standing-there.html">I Saw Her Standing
There
```

WORK THE SHELL

Nice. Now how about turning each into a `curl` page query? Well, hold on! Let's first figure out how to get the full list of every song—that is, how to go from page to page. To do that, the URL already shown has the clue: `from=XX` for each subsequent page.

Another quick test shows what happens if you specify a URL that is beyond the last song listed: no matches are returned. That's easy to deal with because `wc -l` will return a zero in that instance.

Put the pieces together, and here's a loop that will get as many matches as possible until there's a zero result:

```
url="http://www.mldb.org/search?mq=the+beatles&mm=2&si=1&ob=2"
output="lyrics-page." # you can put these in /tmp
start=0 # increment by 30, first page starts at zero
max=600 # more than 20 pages of matches = artificial stop

while [ $start -lt $max ]
do
    curl -s "$url&from=$start" | sed 's/</\
</g' | grep 'href="song-' > $output$start
    if [ $(wc -l < $output$start) -eq 0 ] ; then
        # zero results page. let's stop, but let's remove it first
        echo "hit a zero results page with start = $start"
        rm "$output$start"
        break
    fi
    start=$(( $start + 30 )) # increment by 30
done
```

I'll explain what's going on in the code momentarily, but let's just see what it does and then use an `ls` invocation to double-check it created non-zero output files:

```
$ sh getsongs.sh
hit a zero results page with start = 240
$ ls -s lyrics-page*
8 lyrics-page.0          8 lyrics-page.180      8 lyrics-page.60
```

WORK THE SHELL

```
8 lyrics-page.120    8 lyrics-page.210    8 lyrics-page.90
8 lyrics-page.150    8 lyrics-page.30
```

Perfect. I expected eight pages of songs, and that's what the script produced. Each has the same format as the output listed earlier, so it's now a matter of converting the href= format into an invocation to `curl` to get that particular page of lyrics. Since I'm already running out of space, however, I'm going to defer that part of the script until my next article.

Meanwhile, notice how `start` is incremented by 30 with the `$(())` notation for calculations (you could use `expr`, but it's faster to stay in the shell and not spawn a subshell for the math). Also, the test to identify an empty output file should be easy for you to understand:

```
if [ $(wc -l < $output$start) -eq 0 ]
```

There is a nuance to catch here, however: the `$()` notation gets you a sub-shell akin to using backticks, while the `$(())` notation allows you to do rudimentary calculations within the Bash shell itself.

I'll expand on all of this in my next article. See ya then! ■

Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

[RETURN TO CONTENTS](#)

InterDrone

The International Drone Conference and Exposition

Discover the Future – at the World’s Largest Commercial Drone Conference & Expo

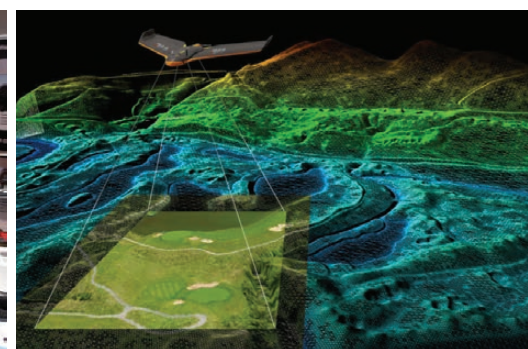


Image credit: Future Aerial Innovations



“If you want to see the state-of-the-art and expand your knowledge about the drone industry, InterDrone is the place to be.”

—George Gorrill, Structural Engineer, Thomas Engineering Group

September 6-8, 2017

Las Vegas

www.InterDrone.com

Register Early for the Biggest Discount!



Update Tickets from the Command Line

Why use that pesky mouse when you can pipe command-line output and have it appear as a comment in your ticketing system?



KYLE RANKIN

Kyle Rankin is VP of engineering operations at Final, Inc., the author of many books including *Linux Hardening in Hostile Networks*, *DevOps Troubleshooting* and *The Official Ubuntu Server Book*, and a columnist for *Linux Journal*.

Follow him @kylerankin.

PREVIOUS

◀ Dave Taylor's Work the Shell

NEXT

Shawn Powers' The Open-Source Classroom ▶

IN THE APRIL 2017 ISSUE, I wrote about how to use ticketing systems as a sysadmin to organize your tasks better. In that article, I made a brief reference to the fact that I've integrated some of my own scripts with my ticketing system, so here I'm going to talk about a basic bash script I whipped up in a few minutes that can add a comment to a Jira ticket. Although my examples specifically are for use with the Jira ticketing system, you can adapt the same kind of idea to any ticketing system that allows you to interact with it via a web-based API.

One reason many sysadmins dislike ticketing

systems is due to the fact that updating and maintaining tickets requires a shift in focus and a break from their regular workflow. To me, tickets are a great way to build an audit trail to show that you've completed a task, and one of the best ways to demonstrate that a task is complete is to paste in the proof of your work into comments in a ticket. Unfortunately, all of that copying and pasting can slow things down and discourages sysadmins from updating their tickets with command output.

My solution to the pain of keeping tickets updated with command output is to create a script I can pipe output to and have it appear as a comment in the ticket I specify. My tasks often generate output into log files, and it's nice just to pipe those log files into a script and have them show up in a ticket. I generally use my `create_jira_comment` script like this:

```
$ somecommand | create_jira_ticket -t TICKETNAME
```

My command may be as simple as an `echo` command or something much more sophisticated. In some cases, I've wanted to wrap the output inside a code block within the ticket comment and pass along a header to describe what the code block is, so I've added `-C` and `-H` options, respectively:

```
$ somecommand | create_jira_ticket -t TICKETNAME -C -H "Here  
↳is the output from somecommand"
```

This makes it really easy for an administrator to update a ticket with command output without messing with copying and pasting pages of output into a ticket. The output shows up formatted properly, and when it's in a code block, the ticket shows it in such a way that someone can scroll through it to read the whole thing, but it doesn't fill up a whole page.

Before I get into the Jira-specific bits, let me go over the more generic parts of the script. First, there's the opening section of the script where I define a few variables, set some defaults and source a settings file so I don't have to have the password be hard-coded into

this script:

```
#!/bin/bash

JIRA_HOST="somehost.example.com"
JIRA_USER="someuser"
JIRA_PASS="somepass"
# Set the user and password in a settings file
# instead of in the script
. /etc/default/jira_settings

OUTFILE="/tmp/create_jira_comment-$(date +%Y%m%d-%H%M%S)"
```

Next, I add a typical usage function (like all good script writers should) to output if someone doesn't use the right syntax with my script:

```
# Show usage information
usage() {
    cat >&2 <<EOF
Usage:
    $0 [-h | -t TICKET <-f FILENAME> <-H "Header text">
    ↪<-F "Footer text"> <-C>]
```

This script adds a comment to a Jira ticket based on command-line arguments.

OPTIONS:

```
-h          Show usage information (this message).
-t TICKET   The Jira ticket name (ie SA-101)
-f FILENAME A file containing content to past in the Jira
comment (or - to read from pipe)
-H HEADER_TEXT Text to put at the beginning of the comment
-F FOOTER_TEXT Text to put at the end of the comment
-C          Wrap comment in a {code} tags
EOF
}
```

As you can see in the usage output, the script can accept a number of command-line arguments. The one required option is `-t`, which specifies the name of the ticket to which you want to add the comment. All of the other options are optional.

As I started using this script, I realized that I often was piping command output into this ticket, and the default formatting inside a Jira comment just made it a huge wall of text. The `-C` option will wrap all of the text into a tag so that it shows up like code and is easier to read. Once I started wrapping output inside proper formatting, I realized sometimes I wanted to add text above or below the code block that explained what the code block was. I added the `-H` and `-F` arguments at that point, which let me specify text to use as a header or footer around the code block.

The next section of the script is where I gather up the command-line options using the standard `getopts` tool:

```
# Parse Options
while getopts ":t:f:H:F:Ch" flag; do
  case "$flag" in
    h)
      usage
      exit 3
      ;;
    t)
      TICKET="${OPTARG}"
      ;;
    f)
      FILENAME="${OPTARG}"
      ;;
    H)
      HEADER="${OPTARG}"
      ;;
    F)
      FOOTER="${OPTARG}"
      ;;
    C)

```

HACK AND /

```
        CODETAG='{code}'
        ;;
    \?)
        echo "Invalid option: -$OPTARG"
        exit 1
        ;;
    :)
        echo "Option -$OPTARG requires an argument"
        exit 1
        ;;
esac
done

# Shift past all parsed arguments
shift $((OPTIND-1))

test -z "$TICKET" && usage && echo "No ticket specified!" && exit 1
test -z "$FILENAME" && FILENAME='- '
```

There's really not all that much to elaborate on with the `getopts` tool. You can see how I handle the case where a ticket isn't defined and how I set the default file to be pipe input if the user doesn't set it.

Now that I have all of the options, I can do the actual work of creating the Jira ticket. First, I need to create a file that's formatted in JSON in a way that the JIRA API can accept:

```
echo -n -e '{\n  "body": "' > ${OUTFILE}.json
test -z "$HEADER" || echo -n -e "${HEADER}\n" >> ${OUTFILE}.json
test -z "$CODETAG" || echo -n -e "${CODETAG}\n" >> ${OUTFILE}.json
cat ${FILENAME} | perl -pe 's/\r//g; s/\n/\\r\\n/g; s/"/\\"/g' >>
  ➡${OUTFILE}.json
test -z "$CODETAG" || echo -n -e "\n${CODETAG}" >> ${OUTFILE}.json
test -z "$FOOTER" || echo -n -e "\n${FOOTER}" >> ${OUTFILE}.json
echo -e '"\n}' >> ${OUTFILE}.json
```

You can see in the previous code where I test whether a header,

That said, the meat of the formatting is right in the middle where I cat the main output into a series of Perl regular expressions to clean up carriage return and newlines in the output and also escape quotes properly.

code argument or footer was defined, and if so, I inject the text or the appropriate code tags at the right places in the JSON file. That said, the meat of the formatting is right in the middle where I cat the main output into a series of Perl regular expressions to clean up carriage returns and newlines in the output and also escape quotes properly. This would be the section where you'd apply any other cleanup to your output if you noticed it broke JSON formatting.

Once I have a valid JSON file, I can use `curl` to send it to Jira in a `POST` request with the following command:

```
curl -s -S -u $JIRA_USER:$JIRA_PASS -X POST --data @${OUTFILE}.json -H
  ➤ "Content-Type: application/json"
  ➤ https://$JIRA_HOST/rest/api/latest/issue/${TICKET}/comment
  ➤ 2>&1 >> $OUTFILE
```

```
if [ $? -ne 0 ]; then
  echo "Creating Jira Comment failed"
  exit 1
fi
```

If the command fails, I alert the user, and since I captured the `curl` output in the `$OUTFILE` log file, I can review it to see what went wrong.

Here is the full script all in one piece:

```
#!/bin/bash

JIRA_HOST="somehost.example.com"
JIRA_USER="someuser"
JIRA_PASS="somepass"
# Set the user and password in a settings file
# instead of in the script
. /etc/default/prod_release

OUTFILE="/tmp/create_jira_comment-$(date +%Y%m%d-%H%M%S)"

# Show usage information
usage() {
    cat >&2 <<EOF
Usage:
    $0 [-h | -t TICKET <-f FILENAME> <-H "Header text">
    ↪<-F "Footer text"> <-C>]

This script adds a comment to a Jira ticket based on
command-line arguments.

OPTIONS:
    -h                Show usage information (this message).
    -t TICKET         The Jira ticket name (ie SA-101)
    -f FILENAME       A file containing content to past in the Jira
    ↪comment (or - to read from pipe)
    -H HEADER_TEXT    Text to put at the beginning of the comment
    -F FOOTER_TEXT    Text to put at the end of the comment
    -C                Wrap comment in a {code} tags
EOF
}

# Parse Options
while getopts ":t:f:H:F:Ch" flag; do
```



```
case "$flag" in
  h)
    usage
    exit 3
    ;;
  t)
    TICKET="${OPTARG}"
    ;;
  f)
    FILENAME="${OPTARG}"
    ;;
  H)
    HEADER="${OPTARG}"
    ;;
  F)
    FOOTER="${OPTARG}"
    ;;
  C)
    CODETAG='{code}'
    ;;
  \?)
    echo "Invalid option: -$OPTARG"
    exit 1
    ;;
  :)
    echo "Option -$OPTARG requires an argument"
    exit 1
    ;;
esac
done

# Shift past all parsed arguments
shift $((OPTIND-1))

test -z "$TICKET" && usage && echo "No ticket specified!"
➡&& exit 1
```

```
test -z "$FILENAME" && FILENAME='- '

echo -n -e '{\n  "body": "' > ${OUTFILE}.json
test -z "$HEADER" || echo -n -e "${HEADER}\n" >> ${OUTFILE}.json
test -z "$CODETAG" || echo -n -e "${CODETAG}\n" >> ${OUTFILE}.json
cat ${FILENAME} | perl -pe 's/\r//g; s/\n/\\r\\n/g;
  ↪s/"/\\"/g' >> ${OUTFILE}.json
test -z "$CODETAG" || echo -n -e "\\n${CODETAG}" >> ${OUTFILE}.json
test -z "$FOOTER" || echo -n -e "\\n${FOOTER}" >> ${OUTFILE}.json
echo -e '"\n}' >> ${OUTFILE}.json

curl -s -S -u $JIRA_USER:$JIRA_PASS -X POST --data @${OUTFILE}.json -H
  ↪"Content-Type: application/json"
  ↪https://$JIRA_HOST/rest/api/latest/issue/${TICKET}/comment
  ↪2>&1 >> $OUTFILE

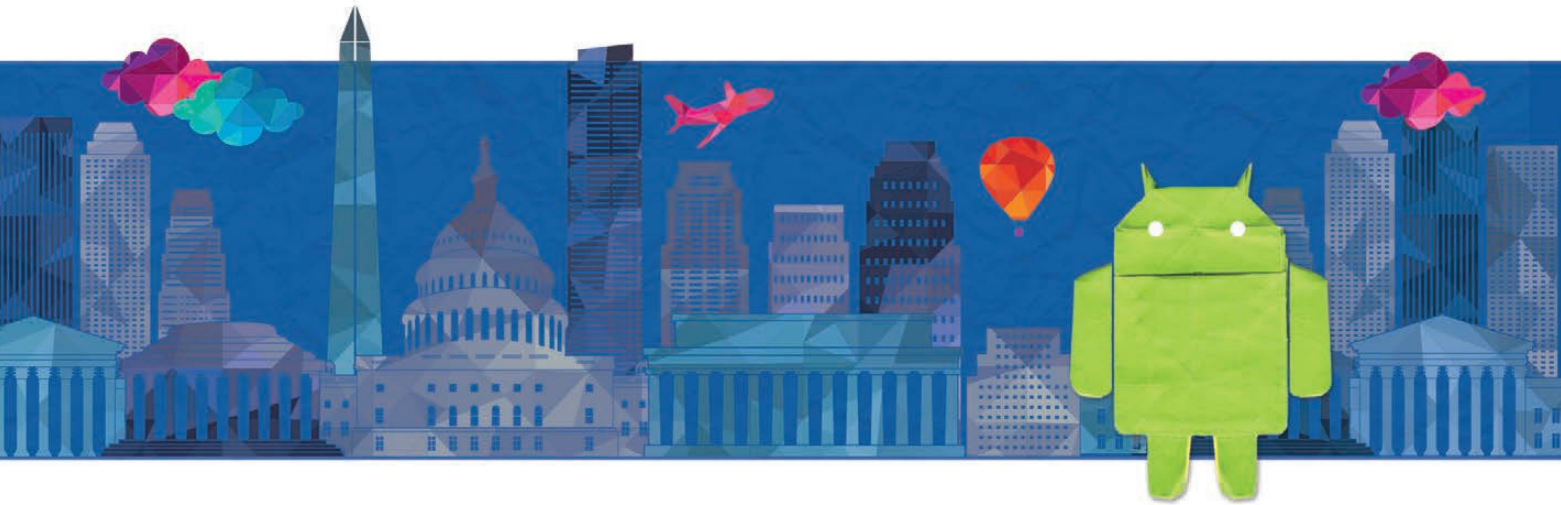
if [ $? -ne 0 ]; then
  echo "Creating Jira Comment failed"
  exit 1
fi
```

I've found I now use this script all the time to interact with my ticketing system. In the past, there were times when I could get a bit lazy with archiving proof of work into Jira tickets unless I knew it was truly necessary, but with this script, it's easy, so I find I do it more. In general, I've found if you can make the correct workflow the easiest workflow, your team is more likely to follow it. This script is just one example of how that can work in practice. ■

Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

[RETURN TO CONTENTS](#)

➔ Register Early and SAVE!



Create/Design/Develop/Connect **AnDevCon**

The Android Developer Conference

3 BIG TRACKS!

Android Fundamentals

A deep dive into the fundamentals all the way to the most advanced capabilities of Android and the Android ecosystem.



Cross-Platform Development

This track goes beyond native Android development to teach developers to master cross-platform development tricks, techniques, and tools.



Machine Learning/AI

An entire track at AnDevCon is devoted to machine learning and AI practices that are at the forefront of the next wave of Android and mobile technology!



Great classes, the reception is awesome and there's better technical content than Google I/O!

— Kevin Cousineau, Mobility Architect, Ideas Improved

AnDevCon is definitely worth attending. You get very useful information from very experienced speakers, and get to network with others.

— Anil K. Dokula, Software Engineer, Vedicsoft Solutions

July 17-19, 2017

Washington, DC

www.AnDevCon.com



A **BZ Media** Event

AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.

Live Stream Your Pets with Linux and YouTube!

YouTube live streams are complicated to set up, until now.



**SHAWN
POWERS**

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 email at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

PREVIOUS

◀ Kyle Rankin's
Hack and /

NEXT

New Products ▶

ANYONE WHO READS *LINUX JOURNAL* KNOWS ABOUT MY FASCINATION WITH BIRDWATCHING.

I've created my own weatherproof video cameras with a Raspberry Pi. I've posted instructions on how to create your own automatically updating camera image page with JavaScript. Heck, I even learned CSS so I could make a mobile-friendly version of BirdCam that filled the screen in landscape mode.

Recently, however, I've finally been able to create an automated system that streams my BirdCam live

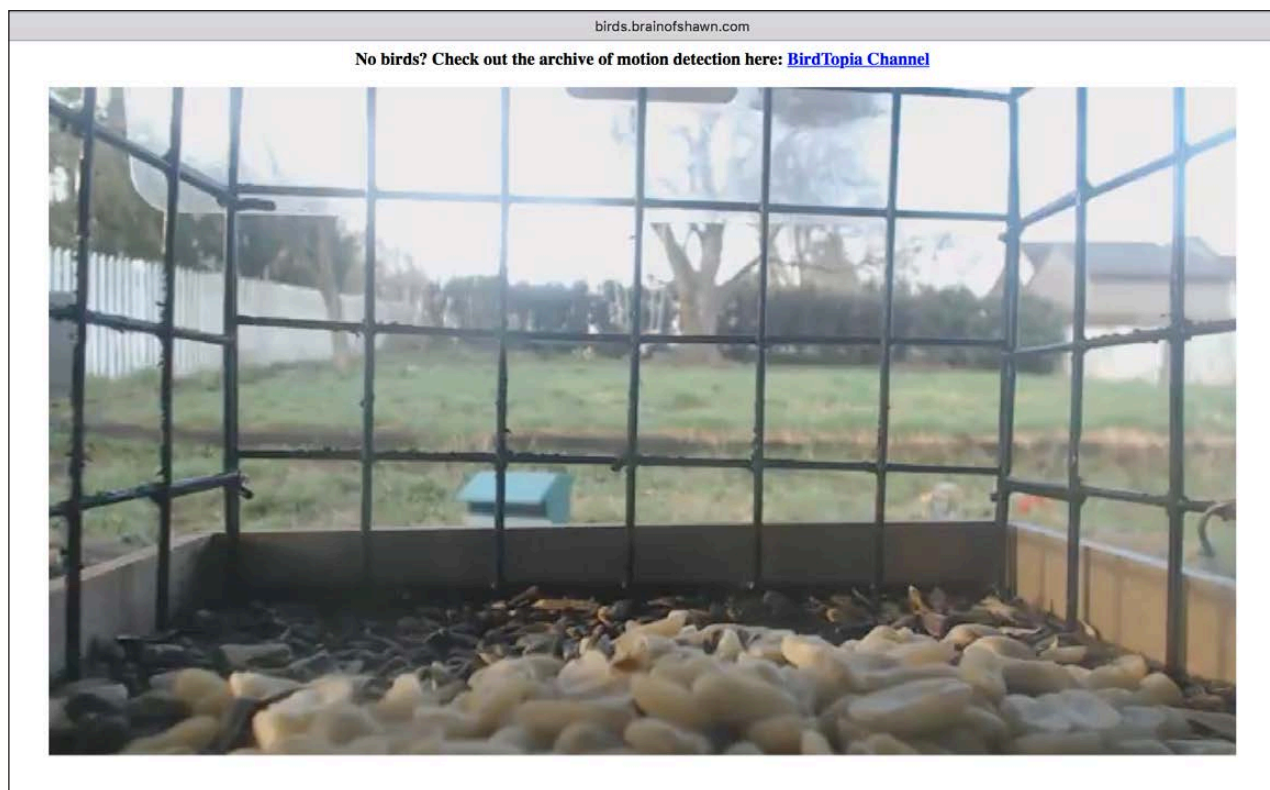


Figure 1. Birds are always camera-shy. If you watch long enough, however, they come and steal peanuts!

over YouTube. It starts when the sun comes up and stops when the sun goes down. And thanks to some powerful open-source software, I never have to touch the system!

Some of the tools I describe here have been covered in other articles, but this is the first time I've been able to create a stream that anyone can see utilizing bandwidth Google pays for!

My List of Ingredients

First off, I want to be clear about what sort of hardware and software is required in order to accomplish something similar to what I'm doing:

1. A Linux computer: if you plan to use USB cameras, this needs to be a physical computer. If your video source is network-based, this can be a virtual machine on your network. A Raspberry Pi isn't really powerful enough for the video work that has to be done, unless maybe it's low-resolution. I have an old i5 CPU running at 1.6GHz,

and it's more than enough.

2. A video source: this can be pretty much any video source you have at hand. If you plan to use a USB webcam, you'll need to be sure you are using a physical Linux computer as noted above. I've used USB, MJPEG over http (see my old BirdCam articles), cheap wireless security cameras that have an RTSP stream, and most recently, I started using UniFi video cameras. In fact, if you are considering purchasing outdoor video cameras for a project like this, I can't recommend UniFi cameras enough. They are PoE, HD and the free software handles recording and provides RTSP streams that have both HD video and top-notch audio.
3. A YouTube account with Live Streaming enabled: you'll need to verify your account (<https://www.youtube.com/verify>), and then enable live streaming here: https://www.youtube.com/live_dashboard. It's not a difficult process, but without following those steps, you won't be able to use the free service.
4. Open Broadcaster Software: I've tried multiple ways to use a CLI solution to stream directly to YouTube with FFmpeg or mencoder, but I've never been able to make it work consistently. I was hesitant to use OBS, because it's a GUI solution and doesn't have a CLI interface, but I worked around that problem, and I'm actually happy to have the GUI now.
5. A web server to host your embedded channel: you could just share the URL to your YouTube channel, but embedding is much cooler, because you can integrate it into your own site.
6. Enough upstream bandwidth to support 1.5–2mbps while streaming: since YouTube is going to redistribute, the local bandwidth requirements don't change regardless of how many people are watching your stream. For some folks (like me, unfortunately), sacrificing that much bandwidth is difficult and sometimes causes issues. Just know that it takes a small, but not insignificant amount of constant upstream bandwidth to stream live video. That should be obvious, but it's something to consider.

7. A few other utilities like crontab and sunwait: the latter is only if you want to time your streams with sunrise and sunset. And, crontab is needed only if you want to automate the starting and stopping. Those touches really make a difference for me though, so I encourage you to consider it.

Gather Your Info

YouTube: In order to live stream, you'll need a few bits of information. As I mentioned above, you'll need to verify your account to turn on streaming. Then you'll need to get your streaming key (Figure 2). It's important that you not share the streaming key, because it acts like your

The screenshot shows the YouTube Live Stream settings interface. At the top, there is a status bar with a grey circle and the text "OFFLINE ?" on the left, and "welcome back, Sna..." and "Still have questions about streami..." on the right. Below this is a stream title field containing "Birdcam Live!" and a chat icon with the number "8". A description field contains the text "Check out the birds outside my office window!". There is a checkbox for "Schedule next stream" which is unchecked. Below that is a "Category" dropdown menu set to "Pets & Animals". A "Privacy" dropdown menu is set to "Public". An "Advanced settings" link is visible at the bottom right of this section. The "ENCODER SETUP" section is highlighted with a red underline. It contains a "Server URL" field with the value "rtmp://a.rtmp.youtube.com/live2". Below that is a "Stream name/key" field with the value "5uew-4uq4-auq4-5ewg", a "Hide (2)" button, and a "Reset" button. A warning icon (yellow triangle) is followed by the text: "Anyone with this key can live stream on your YouTube channel. Keep it secret."

Figure 2. That's not my real streaming key, just FYI.

authentication. If others get your key, they can stream to your channel, even without your user name.

The other bit of information you'll need from YouTube is your channel ID. It's not easy to find the channel ID, but if you want to embed your video, you'll need it later. Head over to https://www.youtube.com/account_advanced, and look for the line that looks like, "YouTube Channel ID: UCbUTB3bVg3cmeyJUtUC9DPA" (your channel ID will be different from mine). The long string of text is your channel ID, copy that somewhere easy to find.

Video Camera Feeds: I can really give you only hints about what to look for here. You need to find the streaming video feed coming from your camera. Make sure you don't use the web page that has the stream embedded (most cameras have a rudimentary web server that embeds the stream). You need the raw feed itself. Google or the user's manual will be your best bet for figuring out the raw stream URL.

I have an Onvif-compatible video camera that has an MJPEG stream URL that looks like this: `http://192.168.1.170:9090/stream/video.mjpeg`.

One of my Foscam cameras requires a user name and password in the URL in order to get the stream. It looks like this: `http://192.168.1.180:88/cgi-bin/CGIStream.cgi?cmd=GetMJStream&usr=admin&pwd=xxx`.

And my new UniFi cameras actually use an RTSP URL that comes from the UniFi server instead of from the cameras directly. It looks like this: `rtsp://192.168.1.16:7447/58cf11bef14c359f4b3c7b2e_1`.

The point I'm trying to make is that finding your video camera's streaming URL often is challenging. If you do it before you start, it can save hours of frustration. An easy way to test if you've found the correct URL is to try opening it in VLC. I haven't found a video camera that VLC can't view, so if it complains about an invalid video source, you probably don't have the correct URL. Google, along with your camera's model number, is probably the best way to figure it out.

The Software

There are many scripts online claiming to stream from a camera source to YouTube using FFmpeg. I'm sure they work for someone, but I've never gotten them to work, no matter how many settings I tweak. In fact, I gave up for quite a while because I didn't want to rely on a GUI interface to stream. I

THE OPEN-SOURCE CLASSROOM

wanted my server to do the dirty work and do it without my interaction. One day recently, however, I discovered that Open Broadcaster Software (OBS) supports command-line flags for starting streaming. That means I could have the server start streaming without the need to “click” anything.

One problem I had to overcome was the lack of the X Window System on my BirdCam server. There’s no monitor connected to the server, but in order for OBS to work, it has to have a logged-in GUI desktop. I hooked up a monitor long enough to get a GUI installed and then set the system to log in automatically. I also disabled all power-saving features for the monitor, because I wouldn’t have one logged in anyway. Once it was set up, I installed TeamViewer so I could control the system remotely if I needed to. There have been some issues with TeamViewer’s security recently, so it might not be the software you choose for controlling the server, but it’s what I have installed, and it works. Figure 3 shows my

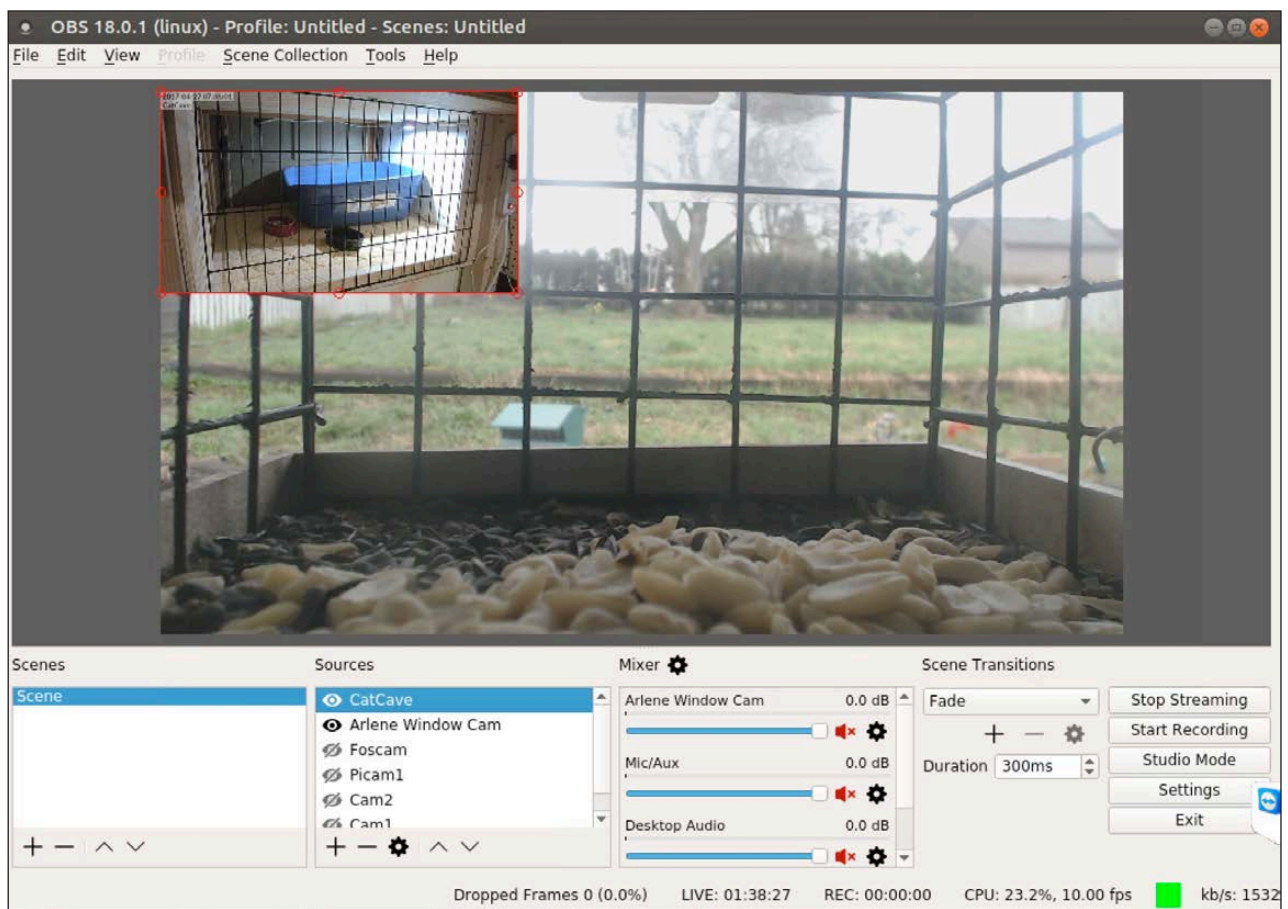


Figure 3. I don’t normally have the security camera to my cat’s litter cave on my live stream, but I wanted to show multiple cameras.

OBS also has the great feature of saving your last-used session. That means once you set up your cameras, you don't have to worry about readjusting them on the next launch.

“server” controlled remotely via TeamViewer.

Installing OBS is simple. Head over to <https://obsproject.com> and download the latest version, or simply install their PPA if you're using Ubuntu. The software has matured since I last mentioned it, and I didn't have any problems with dependencies, even when connecting over a remote session.

OBS also has the great feature of saving your last-used session. That means once you set up your cameras, you don't have to worry about readjusting them on the next launch. OBS just uses the same settings you had before. If you look back at Figure 3, you'll see there are multiple cameras added to the preview window. Without the need to save a layout, OBS just remembers from launch to launch how you had the cameras positioned.

In order to get the best results, you need to tweak a few OBS settings. Click the settings button, and then head over to the Video tab (Figure 4). This is a little confusing, but you have two different resolutions to set. The “canvas” is how big you want OBS to show on your preview window. The “output” resolution is what it scales your video to for streaming and recording. I just set them both to 720p, because I figure scaling takes CPU. You also set the frames per second (FPS) for the output video. I use 10FPS with the 720p size. You can adjust this if you want 1080p, or down if you don't have bandwidth.

Next, click on the “Output” tab (Figure 5). My settings are visible, and I recommend keeping them close to mine, except for the bitrate of the video and audio. If you want higher quality video (and you can

THE OPEN-SOURCE CLASSROOM

afford the bandwidth), this is where you set the average upload speed. You also can change the audio quality if you want higher quality. Keep in mind that the resolution you chose in the last step will work with the bandwidth you selected here to give you the video quality users will see. A video rate of 1500 (measured in kbps) works well with my 10FPS and 720p resolution. But if you try to stream 1080p, 30FPS video

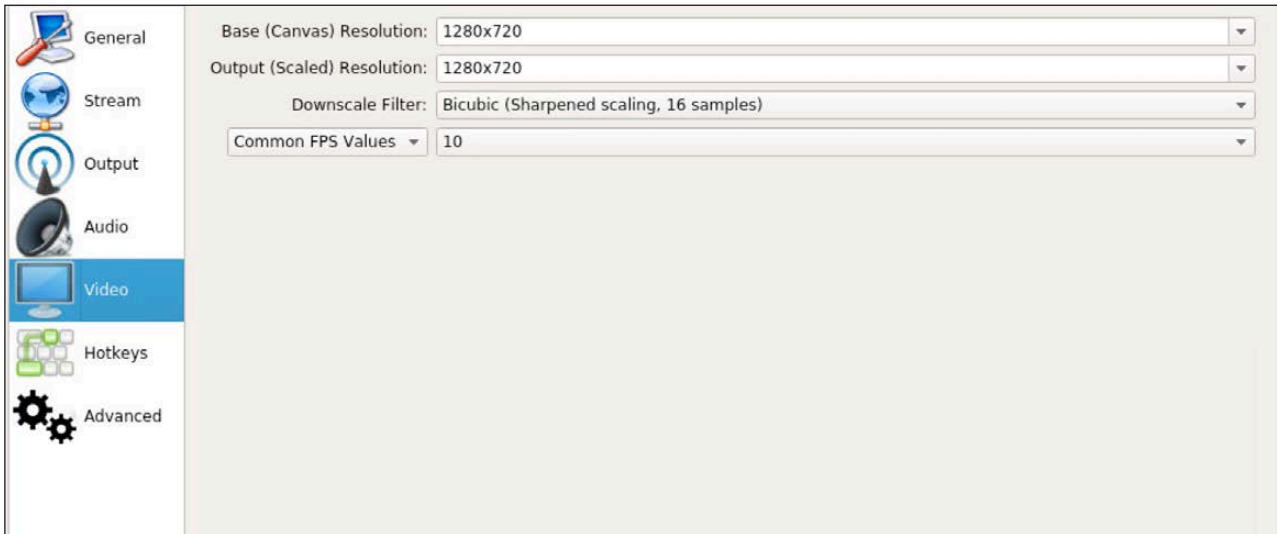


Figure 4. The resolutions are flexible, but I like to keep it simple.

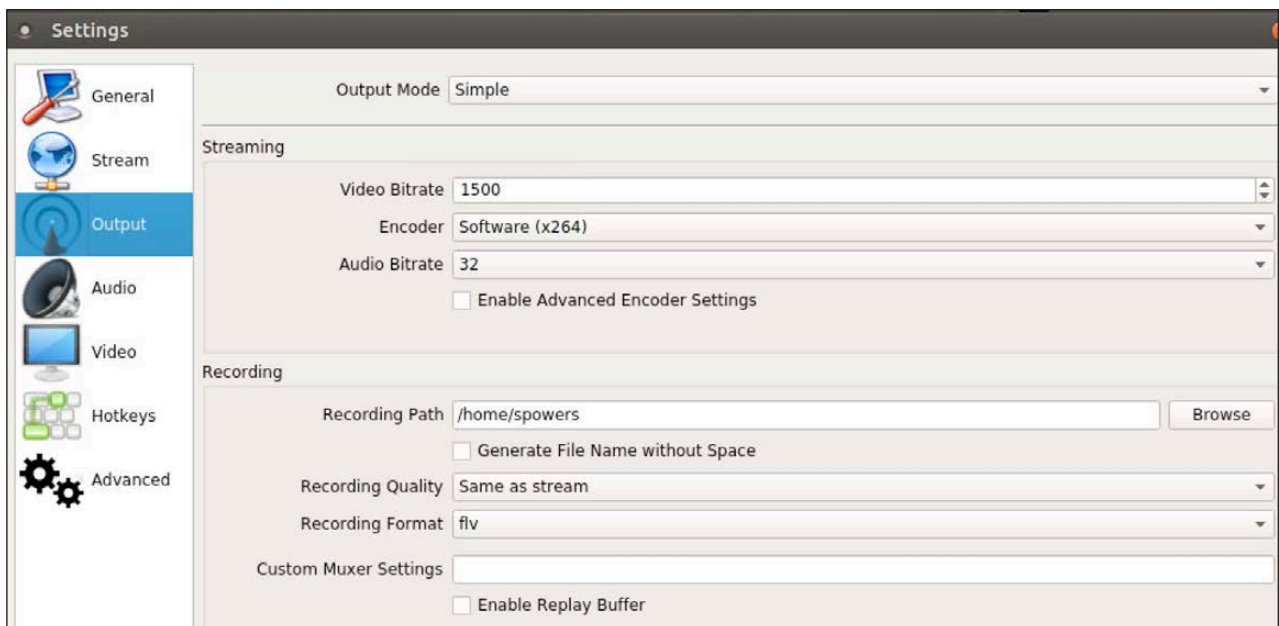


Figure 5. 1500 is the maximum my current internet connection can handle.

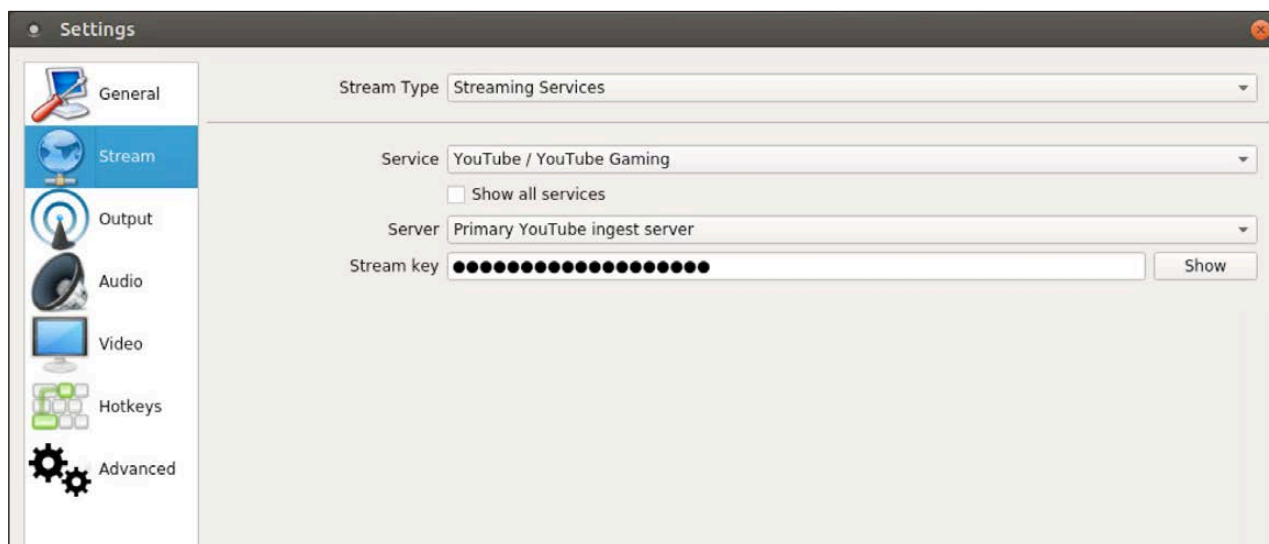


Figure 6. It's truly amazing how well OBS does with YouTube streaming.

with 1500kbps, it's going to be really poor quality video. You'll have to experiment to find the sweet spot.

The "Stream" tab is where you configure the streaming service you want OBS to use (Figure 6). You should be able to select YouTube and then paste that stream key you got from YouTube earlier. (This is *not* the channel ID; it's that hidden key from back in Figure 2.) Once entered, you shouldn't need to make any changes in settings. OBS will keep all the settings, including streaming information.

All that's left is to add the camera(s) to your preview screen. This is the nicest feature of OBS, well apart from actually being able to stream to YouTube. The setup is drag and drop, and you can resize cameras, overlap cameras and arrange them however you want. Since OBS supports so many types of inputs, you can get crazy with text overlays and so on. To add a network camera, click the + at the bottom middle of the main window, and select "media source" (Figure 7). Then uncheck "local file" and enter the camera URL in the "input" field (Figure 8). Once you click OK, your camera should appear on the preview window, and you can resize and move it. The interface also allows you to crop the section of the video you want to use. It's very powerful and incredibly user-friendly. Plus, as I mentioned earlier, OBS stores all your tweaks automatically, so the next time you start it, you'll get the same arrangement.

THE OPEN-SOURCE CLASSROOM

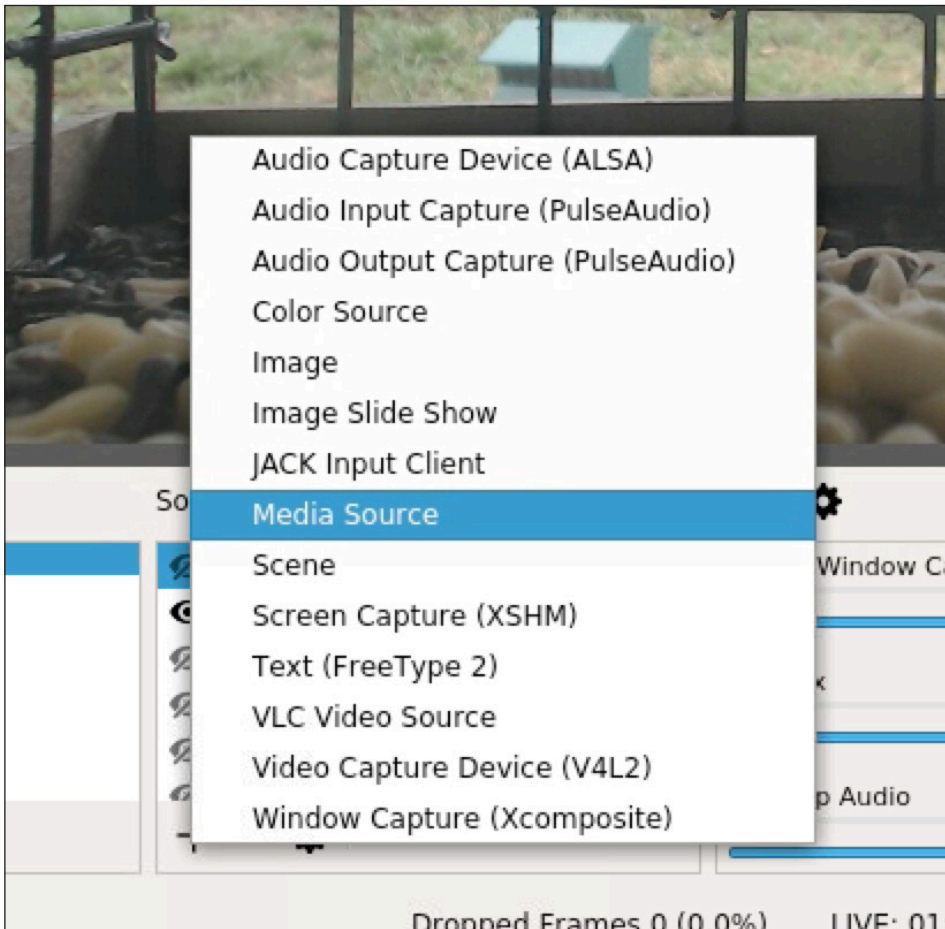


Figure 7.
Media source
isn't obvious as
the choice for
network cameras.

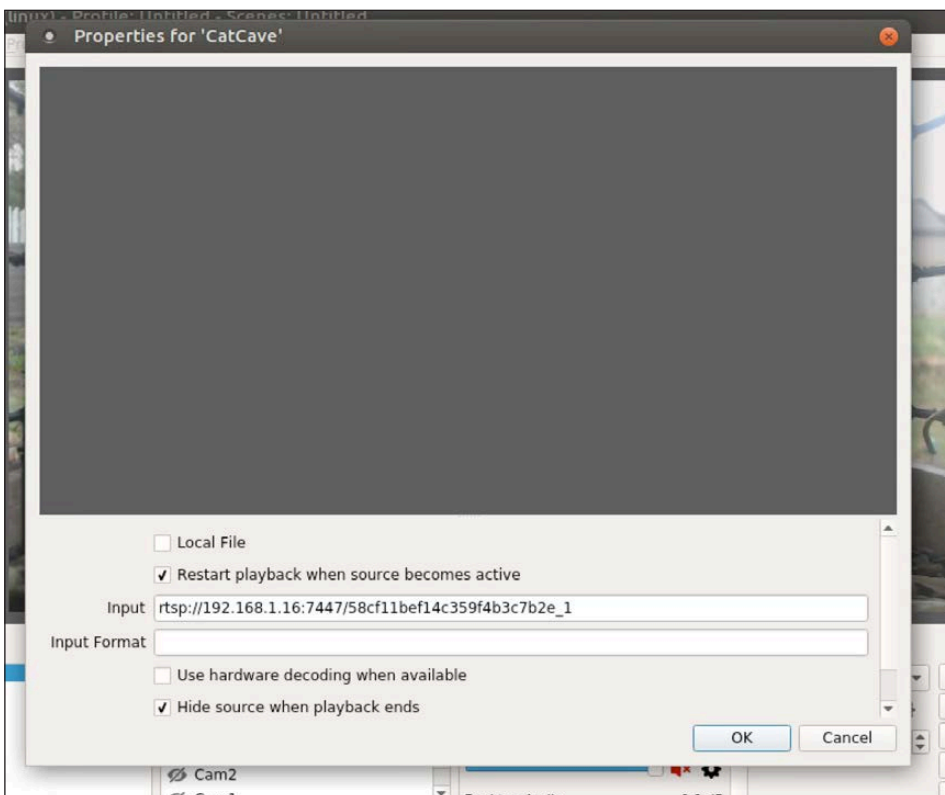


Figure 8.
Be sure to uncheck
the "local file", or
you won't have an
input field.

I not only want to automate the starting and stopping of OBS, but I also want to make sure that if something crashes, it starts back up the next day without me needing to fix it.

Once you have your camera(s) set up, you can decide whether you want to include audio if your camera supports it. The audio levels should appear in the column next to the list of cameras. Then just click “Start Streaming” to send your stream live to YouTube. It takes 30 seconds or so to show up in the YouTube dashboard, but now is the time to make sure streaming works.

Automation

I could just leave OBS running 24/7 and have it stream my bird feeders all night. Honestly, I’m not sure how YouTube would handle a 24/7 stream, but I don’t want to do that anyway. I not only want to automate the starting and stopping of OBS, but I also want to make sure that if something crashes, it starts back up the next day without me needing to fix it. Cron was the obvious way to manage that, but since OBS is a GUI program, cron proved to be challenging. In the end, I was able to include environment variables in my crontab, and things worked smoothly. Here’s what my OBS part of crontab looks like. Check it out, and I’ll explain it afterward:

```
DISPLAY=:0
@reboot sleep 10; obs --startstreaming
0 5 * * * /usr/local/bin/sunwait civ up 45.3733N 84.9553W;
  ➔obs --startstreaming
0 16 * * * /usr/local/bin/sunwait civ down 45.3733N
  ➔84.9553W; pkill obs
```

THE OPEN-SOURCE CLASSROOM

First off, setting the `DISPLAY` environment variable to `:0` means that `crontab` can launch a GUI application on the current desktop. I was embarrassed when I realized how easy it was to get `cron` to launch GUI apps. It is important to note that the user must be logged in, however.

The `@reboot` line starts OBS when the system boots. The simple `--startstreaming` flag tells OBS to launch and immediately start streaming. It's awesome. Really, if I had to figure out a way to automate actually clicking a button, we probably wouldn't be doing this project together.

The next two lines are a little confusing. First off, I have the program "sunwait" installed. It's an old program, but it's so incredible, I can't believe it's not in every distribution by default. I've mentioned it before in BirdCam articles, but basically, it's a C program that determines sunrise and sunset based on your longitude and latitude. The last version was released in 2004 (seriously), but it still compiles. You can get the source here: <http://www.risacher.org/sunwait>.

Anyway, those two cron lines tell the server to start and stop OBS at sunrise and sunset. At 5AM, I tell `sunwait` to "wait" until the sun rises. It literally just waits until sunrise and then ends. Once it ends, OBS is started up. Then at 4PM, I tell `sunwait` to wait until sunset, and after the `sunwait` program ends, `kill` stops OBS. Why 5AM and 4PM? Well, in my part of the world, the sun never rises before 5AM and never sets before 4PM. There is the potential problem that if I reboot my server after 4PM, it will stream all night. But that potential problem doesn't concern me enough to make the logic more complicated.

Since my server doesn't have a monitor or keyboard connected, a random GUI application starting and stopping in the middle of the screen doesn't affect anything. Since I connect to my server's desktop only when I want to make a change to OBS, it's actually convenient that it's always running front and center on my desktop! I couldn't be happier with the current live stream setup.

Embedding the Stream

Not long ago, YouTube made a change so that every time a live stream starts, it gets its own embed code. That means if you simply use the "share" button on the live stream to get the embed code, it will

work only for that current streaming session. For me, that means the next day it would show a recording of the previous day, but not the live stream. I'll be honest, that quiet change was very frustrating! Thankfully, there is a way to embed the actual live stream, so that any time you start live streaming, it becomes active—that's where the Channel ID you got earlier comes in.

Here is the embed code for my live stream at <http://birds.brainofshawn.com>:

```
<iframe width="1280" height="720"  
  ↪src="https://www.youtube.com/embed/live_stream?channel=  
↪UCbUTB3bVg3cmeyJUtUC9DPA&autoplay=1" frameborder="0"  
  ↪allowfullscreen></iframe>
```

Obviously, you'll need to make the changes for your own channel, but it should be clear what the various things mean. I stuck with the 720p size even on my embedded page. Since this is embed code, you don't have to put it on its own page like I did; you could embed a tiny resolution version on your blog, for instance.

Setting up the live stream through YouTube is nice for several reasons. One, your bandwidth requirements don't change even if you have 10,000 viewers. Also, since it's YouTube, you can "cast" the video to a television or Chromecast device and show off your channel to your friends. I still hope to get more cameras and maybe set up camera rotation on multiple bird feeders, but for right now, I couldn't be happier. Enjoy! ■

Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

[RETURN TO CONTENTS](#)

ASCEND

► Conference & Expo powered by Drone360

THE ESSENTIAL EVENT FOR THE COMMERCIAL DRONE INDUSTRY

📅 JULY 19-21, 2017 📍 OREGON CONVENTION CENTER, PORTLAND, OREGON

REGISTER NOW!

Linux Journal readers save \$50 on a full conference pass

Discover cutting-edge commercial drone software and technology.

Session topics include:

- LiDAR mapping software
- Advanced image processing
- Thermal and multi-spectral imaging
- Powering the commercial drone super-highway

FEATURED SPEAKERS



GRETCHEN WEST
Hogan Lovells



JONATHAN EVANS
Skyward



COLIN SNOW
Skylogic Research



SHARON ROSSMARK
AeroVista Innovations



Use coupon code linuxjournal to save \$50 off a full conference pass.

Flying is just the beginning. ASCEND-EVENT.COM

NEW PRODUCTS

PREVIOUS



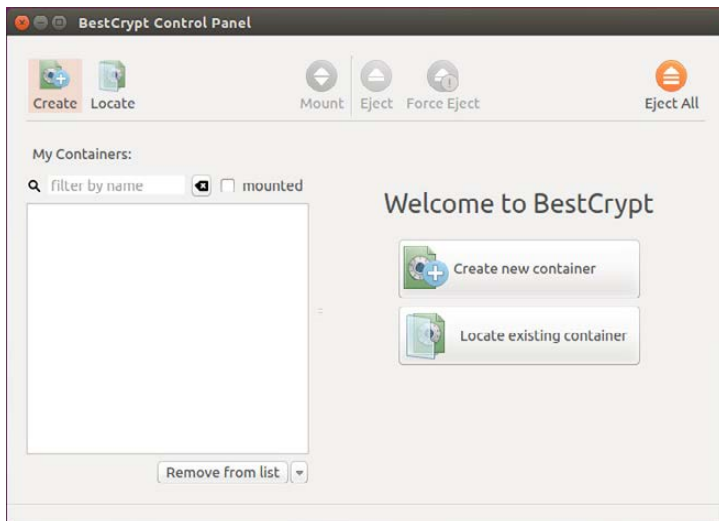
Shawn Powers'
The Open-Source
Classroom

NEXT

Feature: BYOC: Build
Your Own Cluster,
Part II—Installation



Jetico's BestCrypt Container Encryption



Cyber-attacks are now constant, threats to privacy are increasing, and more rigid regulations are looming worldwide. To help IT folks relax in the face of these challenges, Jetico updated its BestCrypt Container Encryption solution to include Container Guard.

This unique feature of

Jetico's Linux file encryption protects container files from unauthorized or accidental commands—like copying, modification, moving, deletion and re-encryption—resulting in bolstered security and more peace of mind. Only users with the admin password can disable Container Guard, increasing the security of sensitive files. The BestCrypt update also adds the Resident feature, an automatic password prompt for mounting containers at startup. That same feature will dismount containers after a time period of inactivity as set by the user. While user-friendly and time-saving, these added features also provide an extra layer of protection when working on shared computers. On endpoints or in the cloud, data encrypted with BestCrypt can be accessed via Linux, Android, Windows and Mac devices.

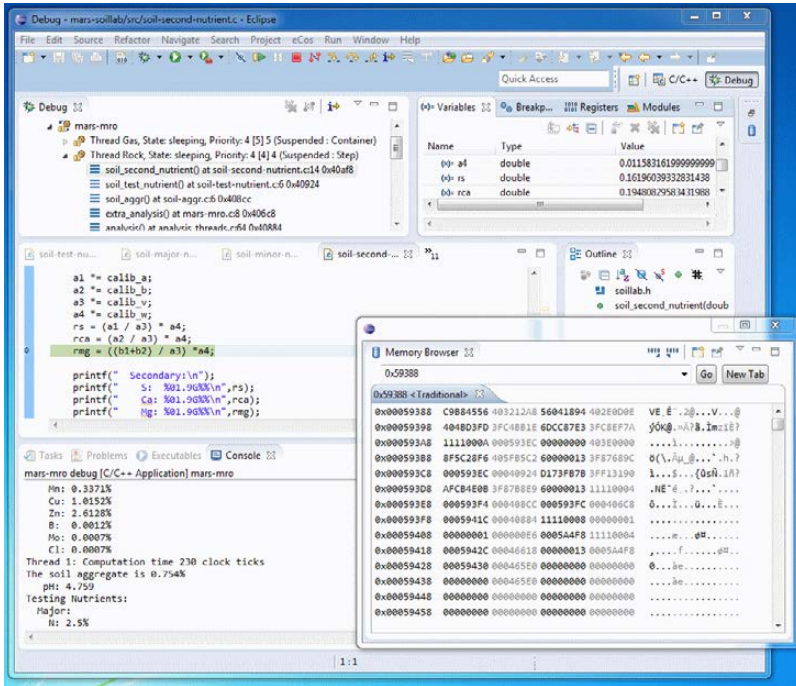
<http://jetico.com>



BlueCat DNS Edge

Migration to the cloud, the flexibility of network virtualization and the promise of IoT involve IT transformations that have placed incredible strain on enterprise security. To identify, assess and block threats proactively in this milieu, BlueCat has developed BlueCat DNS Edge, a first-of-its-kind solution that uniquely leverages DNS data to enhance enterprise security. By driving policy down to DNS control points inside the network, BlueCat DNS Edge leverages the ubiquity of DNS to gain enterprise-wide visibility into the actions of every device on a network, including non-traditional devices, such as wireless security cameras, point-of-sale systems, ATMs and IoT devices. Deployed inside the network, DNS Edge control points identify suspicious internal behavior, patterns and threats, regardless of whether the device is communicating inside or outside the network. BlueCat DNS Edge is promoted as the first DNS security solution with the flexibility to deploy anywhere businesses need it—on premises, in the cloud and as a core part of IoT architectures. BlueCat DNS Edge's simple, cloud-managed deployment model leverages existing DNS infrastructure, enabling organizations to deploy with no impact to existing infrastructure while driving resilience and reducing latency. Delivered as a service, BlueCat DNS Edge dynamically scales to meet anticipated and unanticipated spikes in network activity, such as seasonal activity or a sudden attack against enterprise infrastructure.

<http://bluecatnetworks.com>



eCosCentric Limited's eCosPro

In contrast to general-purpose operating systems for the Raspberry Pi, the new eCosPro from eCosCentric Limited is a lightweight, multithreaded, industrial-strength RTOS delivering reduced latency with bounded response times. eCosPro's resource requirements are a fraction of those demanded by a general-purpose OS and maximize the RAM resources available to applications. The RTOS environment is ideal for time-critical control systems, and by leveraging the ultra-low-cost Raspberry Pi range of single-board computers, eCosPro provides cost-effective, full-featured performance ideal for IoT and M2M applications. Direct boot from an SD card provides an "instant-on" capability, enabling embedded applications to be responsive within milliseconds. eCos is portable across a wide range of embedded architectures and microcontrollers, such that applications prototyped using eCosPro on Raspberry Pi can be readily ported to other targets. eCosPro delivers deterministic, real-time performance on the Raspberry Pi 3, Pi 2, Pi 1, Pi Zero and Pi Zero Wireless boards, as well as the Pi Compute Modules 1 and 3.

<http://ecoscentric.com>

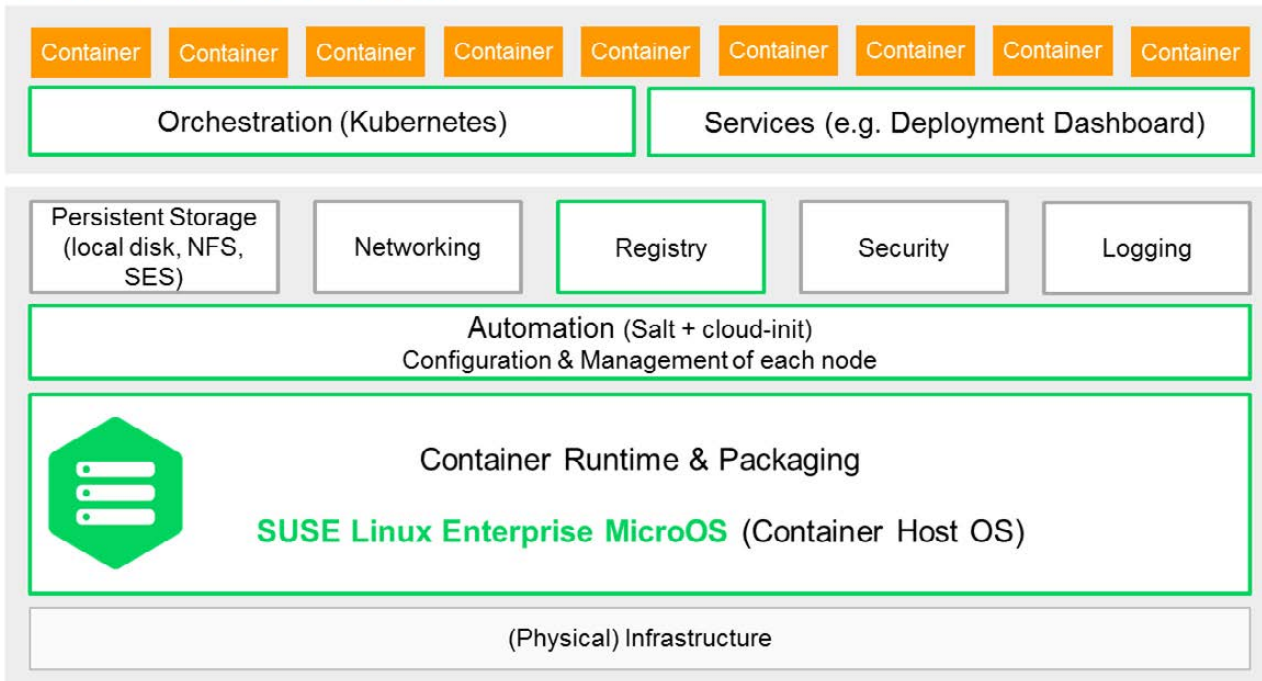


Linux Lite

Linux Lite is a beginner-friendly Linux distribution that is based on the well known Ubuntu LTS and targeted at Windows users. Its mission is to provide a complete set of applications to support users' everyday computing needs, including a complete office suite, media players and other essential applications. The new version, Linux Lite 3.4, simplifies scheduling of software updates, installing third-party drivers and creating a restore point for the OS. Meanwhile, the new Lite Updates Notify application informs the user of all available updates. Users can set update reminders anywhere from once every hour to every three weeks. The updated Lite Welcome has a fresh new look and reminds users to install updates and drivers and sets a restore point after a fresh install of Linux Lite. Other new features in Lite Tweaks include Hibernate & Suspend, Login & Logout options, Manage Saved Sessions and zRam. zRam is a compressed RAM block device for faster I/O and is perfect for older computers.

<http://linuxliteos.com>

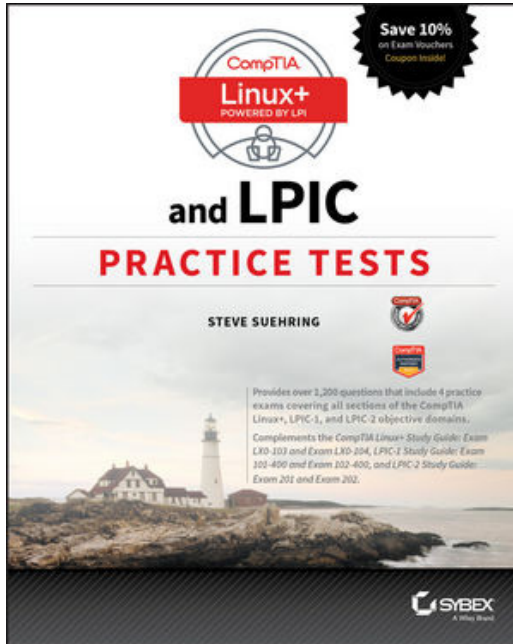
SUSE CaaS Platform



SUSE CaaS Platform

There are a lot of decisions to be made before enterprises are ready for production and deployment of container apps, asserts SUSE. To help enterprises derive full value from containerized apps and not “re-create the wheel”, the SUSE engineering team is busy creating the next-generation application development and hosting platform for container applications and services. The novel SUSE Container as a Service (CaaS) Platform is an application development and hosting platform for container applications and services that lets users provision, manage and scale container-based applications and services, letting them focus on development of container applications to meet business goals faster while reducing costs in developing and maintaining container infrastructure. SUSE CaaS Platform comes with the following ingredients: a tasty new flavor of SUSE Linux Enterprise—container host OS called SLE MicroOS, a good dose of Kubernetes, a pinch of Salt and more special ingredients.

<http://suse.com>

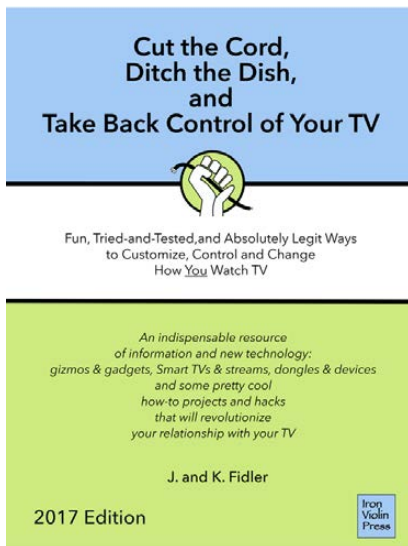


Steve Suehring's *CompTIA Linux+ and LPIC Practice Tests* (Sybex)

Possessing Linux skills is valuable in today's IT job market where demand for talent outstrips supply. Getting certified proves you have the chops to do the job, and two well worn paths to Linux certification are the Computing Technology Industry

Association's CompTIA Linux+ and the Linux Professional Institute Certification (LPIC). To boost your Linux+/LPIC readiness and obtain 100% coverage of all exam objectives on both certifications, you'd be wise to check out Steve Suehring's new *CompTIA Linux+ and LPIC Practice Tests*. Covered in the Sybex-published title are CompTIA Linux+ exams LX0-103 and LX0-104 and the LPIC exams 101-400, 102-400 and 201 and 202, replete with 1,200+ expertly crafted practice questions. Two 60-question practice exams per section help readers gauge their readiness and hone test-taking strategies well in advance of exam day. Buyers of the book also gain access to the Sybex interactive learning environment containing all questions and the ability to create one's own practice tests based on areas where further review is needed. This book can be used alone or with the Sybex study guides.

<http://wiley.com>



J. and K. Fidler's *Cut the Cord, Ditch the Dish, and Take Back Control of Your TV* (Iron Violin Press)

Prospective TV cable-cutters, even those with technical abilities, often are flummoxed in the face of choosing between all of the content options and new technologies available. Reliable sources of complete and neutral information in this space are hard to find, and the fun evaporates rapidly when you're faced with hours of stumbling through forums and strings of searches. A quicker route to TV nirvana is to read J. and K. Fidler's book *Cut the Cord, Ditch the Dish, and Take Back Control of Your TV* from Iron Violin Press. Subtitled *Fun, Tried-and-Tested, and Absolutely Legit Ways to Customize, Control and Change How You Watch TV*, the Fidler team's book is a "time-saving, easily understood, roadmap for readers of all technical levels to be able to have fun, save money, and get the content and TV experience they want". *Cut the Cord* provides "been-there-done-that" tips and explores the new technologies now available to cord-cutters. The nearly 400-page ebook gives basic information needed to get started, then builds on this knowledge to document a variety of simple to advanced DIY projects. One of those projects includes creating a DIY Linux OS-based DVR using MythTV and an Intel NUC.

<http://controltv.ironviolin.com>



ONF/ON.Lab's ONOS Project

Networks have become indispensable infrastructure in modern society. The danger is that these networks tend to be closed, proprietary, complex, operationally expensive and inflexible, all of which impede innovation and progress rather than enabling them. Presenting an alternative vision—that networking can serve the public interest—is the Open Network Operating System, or ONOS Project. ONOS is an open-source, software-defined networking (SDN) OS for service providers that has scalability, high availability, high performance and abstractions to simplify creation of apps and services. The platform is based on a solid architecture and quickly has matured to be feature-rich and production-ready. Recently ONF/ON.Lab announced availability of a new ONOS release that broadens the ability to bring SDN and NFV agility to mission-critical networks. By adding support for “incremental SDN” alongside the “disruptive SDN” capabilities for which it long has been known, ONOS now can address an ever-wider array of deployment scenarios. Additional enhancements include an improved whitebox leaf-spine fabric solution to support IPv6 routing, vLAN tagged external interfaces and AAA endpoint authentication, as well as an enhanced GUI v2.0 that improves usability on large-scale networks by providing regional topology views with drill-down, context-sensitive help and global search, among others.

<http://onosproject.org>

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.

RETURN TO CONTENTS

BYOC

Build Your Own Cluster, Part II—**Installation**

Installing Linux can be fun. Installing it hundreds of times isn't. Learn how Linux installations can be automated, making the installation of a cluster scalable to any number of nodes.

NATHAN R. VANCE, MICHAEL L. POUBLON and WILLIAM F. POLIK



PREVIOUS
New Products

NEXT

Feature: Testing the
Waters: How to
Perform Internal
Phishing Campaigns



In Part I of this three-part series, we left off with bare-metal hardware assembled, a disk partitioning scheme for the head node and compute nodes, and a design for the network.

In this article, we bootstrap ourselves up to performing fully automated operating system installations on both the head node and compute nodes, a vitally important step for the cluster to be scalable to large numbers of compute nodes. To create the kickstart scripts used in automated installations, we'll perform the installation many times, each time with a larger amount of the process being automated. By the end of this article, we'll have an operating system on all nodes and network connectivity.

Head Node Manual Installation

We like to start with doing things manually for several reasons. First, it's a great opportunity to make sure that the hardware is set up correctly. Second, a manual installation generates the kickstart file template that will be used in subsequent installations. Although you can find examples of kickstart files online, it's more useful to generate it yourself, because then you can be sure it contains the specifics for your hardware setup.

To perform the installation, you'll need to download the distribution as an ISO file, burn it to a DVD and boot from it. This guide uses CentOS 7, the redistributable version of Red Hat Enterprise Linux. Alternatively, you could burn it to a USB drive; however, Linux currently names hard disks and flash drives in an arbitrary order, changing the sda and sdb labels on different boots. This will become a problem when you start kickstarting installations. There are workarounds, like using the more verbose UUID naming scheme, but to avoid confusion, we're going to assume you use the DVD.

When installing, we assume the disk partitions as described in Part I of this series:

- / — 200GB
- /admin — 200GB
- /home —rest of space

For networking, configure the interface on the external network as your network administrator dictates. As you've probably discovered by now, it's important always to be on your network administrator's good side, since he or she has nearly unlimited power over your internet connectivity. But, you have full control over the internal network of the cluster.

When configuring the network interface on the internal network, set it to use a statically assigned IP address. The address should be selected from one of the private IP address ranges, which are the following:

- 192.168.0.0 – 192.168.255.255
- 172.16.0.0 – 172.31.255.255
- 10.0.0.0 – 10.255.255.255

In this article, we use a subset of the 192.168 range, specifically 192.168.1.100 – 192.168.1.199, which provides 100 IP addresses. The head node should use the first IP address in the range. Therefore, the configuration for the head node on the internal network is:

- Address: 192.168.1.100
- Netmask: 255.255.255.0

When you select packages to install, choose Server With GUI, and choose E-mail Server and Development Tools as add-ons. We'll fine-tune this selection later, but this is a good starting point.

Head Node Automated Installation

To achieve reliability, one needs to have a reproducible installation method that provides consistent results. Luckily, Red Hat's installer, anaconda, has a reliable and scalable method called kickstart. Kickstarting means that the installer uses a configuration file to install Linux automatically. Anaconda generates a kickstart file after every installation, which you can find at `/root/anaconda-ks.cfg`.

After manually installing the head node, locate this file and copy it

as `ks.cfg` to a separate computer as a backup. You'll be editing `ks.cfg` over the course of this article if you're following along, and if the only copy resides on the same machine being re-installed, a typo could result in its destruction.

Editing the `ks.cfg` File

You can edit the kickstart file to include all desired installation options and post-installation configurations. The kickstart file must use UNIX end-of-line characters, so if you're going to edit the file on a Windows machine, use an editor like notepad++ that respects this difference. Otherwise, it's easiest just to edit it on the head node and back it up to some other machine.

Kickstart files have a specific formatting so that they can be parsed by the system installer. A few important features include:

- **Comments** — the installer ignores any line with a leading `#`. Such lines are used to comment on code or to disable small sections of code.
- **Partitioning** — there are several lines with partition info that you selected during the manual installation. Add the option `--noformat` to the partition entries that you don't want to be formatted by the installer, such as `/admin` and `/home`. Do not add this option to the `/` partition, as it contains the previously installed OS, which should be erased and replaced during a re-installation.
- **Repository** — this tells the installer where to find the repository from which to install. Currently, it is set to install from a CD:

```
# Use CDROM installation media
cdrom
```

The repository line will be modified several times during this guide.

- **Miscellaneous Settings** — settings may be applied, such as disabling SELinux, changing bootloader options and much more. Modify them to disable SELinux: `selinux --disabled`. Visit Red Hat's Kickstart

Options guide for an exhaustive list of kickstart options at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/sect-kickstart-syntax.html.

- **Packages** — after the `%packages` tag is a list of packages to be installed. Those that start with `@` correspond to groups of packages, such as `@ Base`. The others are individual packages.
- **Postscript** — at the end of the installation, some additional commands may be executed. Add the following to the end of the file:

```
%post  
%end
```

Between these tags, you'll add post installation scripts that will be executed once the installation completes. In the DHCP configuration section later in this article, we give a sample script to configure DHCP automatically. We highly recommend that you include all similar system modifications here as well for documentation, backup and re-installation purposes.

Creating an automated installation is an iterative process in which one modifies the kickstart file, re-installs the system and verifies that the changes took hold as intended. For example, to check that `/admin` and `/home` (and any others that you specified) don't get formatted during the install, create a file on the partition and see if it exists after anaconda finishes re-installing. Also check that SELinux is indeed disabled. You will perform a multitude of installations during the creation of the cluster, each time automating and then testing a new capability or feature added to the cluster. Hence, it is vital to automate the process.

You can access the kickstart file for an automated installation in a variety of ways. The kickstart file can be located on an ext2- or fat32-formatted USB drive, or on a partition on the hard drive from where it can be read by the installer.

Table 1 summarizes the different installation methods we use to bootstrap ourselves up to a fully automated installation without removable

Table 1. Installation Plan for the Head Node

INSTALLATION METHOD	BOOT/ INSTALLER LOCATION	KICKSTART LOCATION	DISTRO LOCATION
Manual	DVD	None	DVD
Automated DVD	DVD	USB	DVD
Automated Hard Drive	DVD	Hard Drive	Hard Drive
Sans Removable Media	Hard Drive	Hard Drive	Hard Drive

media. In this table, the boot/installer is the collection of files from which the CentOS installer boots, which currently resides on the DVD. Likewise, the distro is the repository of software that the installer uses to set up the CentOS operating system, also currently located on the DVD.

The end goal is to eliminate the need for removable media and instead to use the hard disk for the entire installation process.

DVD-Based Booting with Kickstart on USB

Before starting, this method has one caveat: depending on the hardware configuration of your system, when the installer boots, the USB drive *may* show up as a different drive from normal. For example, in a configuration with two hard drives, `sda` and `sdb`, when you insert a USB drive, it comes up as `sdc`. However, upon booting into the installer, the USB drive might be `sda`, while the hard drives are `sdb` and `sdc`. This scenario would require the kickstart file to be edited so that *all* references to any `sdX` hard drives are shifted one letter.

To install CentOS using a kickstart file on a USB drive, perform the following steps.

- 1) Copy the kickstart file to a blank ext2- or fat32-formatted USB drive. Make sure the repository line is set to:

```
cdrom
```

2) With the USB inserted into the head node, you now can use the kickstart file while booting from the DVD by pressing Tab at the initial welcome screen and appending the following to the boot options:

```
inst.ks=hd:sdX1:/ks.cfg
```

Note that sdX corresponds to the USB device when booting with it in place, and that the drive letter X might not be the same as when you inserted it into a running operating system. The installer will allow you to edit this line if it fails to find the kickstart file on the specified device.

By performing an installation this way, you have verified that the kickstart file is formatted properly. The next step is to eliminate the USB drive and reduce your reliance on the DVD.

DVD-Based Booting with Kickstart and Distro on Hard Drive

The first step in migrating away from external media is to move the kickstart and distro to the hard drive, thus eliminating the USB drive and reducing the responsibility of the DVD down to just booting.

To use the kickstart and distro from the hard drive, complete the following steps.

1) Make sure you know the device name of the partition mounted as /admin. You can discover this information with the `lsblk` command.

2) Create the following folder hierarchy in /admin:

```
# mkdir -p /admin/iso/centos7/  
# mkdir -p /admin/ks/headnode/
```

3) Copy the installation media to iso/centos7. If you have the original ISO file floating around, you simply can copy it across the network. If not, you can use the `dd` command to create it from the DVD:

```
# dd if=/dev/cdrom of=/admin/iso/centos7/CentOS-7-x86_64-DVD-1511.iso
```

4) Copy the kickstart file to the newly created ks/headnode folder. In the ks.cfg file, comment out the repository line and replace it with

the following:

```
harddrive --partition=sdXY --dir=/iso/centos7/
```

Note that `sdXY` is the partition in which `/admin` is located. You can discover this using the `lsblk` command. Also, while you have `ks.cfg` open, verify that it isn't set to format `/admin`. You've just made some fairly significant additions to that partition, and it would be a bummer to wipe them all out.

5) Insert the DVD and reboot. Press Tab at the initial welcome screen and append the following options:

```
inst.ks=hd:sdXY:/ks/headnode/ks.cfg
```

The installation now should proceed using both `ks.cfg` and the ISO from the hard drive. This makes the installation go much faster, since it doesn't have to read everything from the DVD.

The next step in eliminating external media is to set up the `/admin` partition to be the installer.

Installation Sans Removable Media

Currently, the only role the DVD serves is to boot the installer. In this final step, you'll transfer that role to the hard drive, eliminating the need for all external media. To do so, you'll configure `grub` to boot directly into the installer, pass it the boot options for locating the kickstart file, and install CentOS completely automatically, without any external media or typing `inst.ks=<location>` options into the terminal.

To run the installer from the hard drive, do the following steps.

1) Make a directory to house boot files:

```
# mkdir /admin/boot/
```

2) Mount the iso and copy the internal files to the boot directory:

```
# mount -o loop /admin/iso/centos7/CentOS-7-x86_64-DVD-1511.iso /mnt
# cp -a /mnt/* /admin/boot/
```

3) At the bottom of `/etc/grub.d/40_custom`, insert the following:

```
menuentry "Install" {
    set root=(hdW,msdosZ)
    linux /boot/images/pxeboot/vmlinuz ks=hd:sdXY:/ks/headnode/ks.cfg
    initrd /boot/images/pxeboot/initrd.img
}
```

Note the `(hdW,msdosZ)` and `ks=hd:sdXY` lines. These correspond to the drive and partition for `/admin`. Use the `lsblk` command to find the partition in `sdXY` format. `W` then corresponds to the drive number; `sda` is 0, `sdb` is 1 and so forth. `Z` is the partition number `/admin` is on. So, if `/admin` is located on `sda2`, use `(hd0,msdos2)`.

4) Regenerate `grub.cfg`:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

5) At the end of the `ks.cfg` file, between `%post` and `%end`, insert the following:

```
#grub configuration
cp -p /boot/grub/grub.conf /boot/grub/grub.conf.000
cat >> /etc/grub.d/40_custom << EOF
menuentry "Install" {
    set root=(hdW,msdosZ)
    linux /boot/images/pxeboot/vmlinuz ks=hd:sdXY:/ks/headnode/ks.cfg
    initrd /boot/images/pxeboot/initrd.img
}
EOF
grub2-mkconfig -o /boot/grub2/grub.cfg
```

This script now will modify your `grub.cfg` file automatically, as done in steps 3 and 4. Make sure to modify the `(hdW,msdosZ)` and `sdXY` lines as shown in step 3. The method used here for writing this text to `/etc/grub.d/40_custom` is called a here file. As opposed to using the `echo` command, here files have far fewer characters that must

be escaped, although some characters still need to be—for example, every \$ or ' symbol must be escaped with \\$ or \'. Otherwise, bash will attempt to resolve it as a variable or executable script.

6) Reboot. At grub's splash screen, arrow down to the entry titled "Install" and press Enter. The system should install. Now the system installs automatically without external media. This is a very useful ability to have when rebuilding the head node, and it will be essential for building compute nodes!

Head Node DHCP

Before installing the compute nodes, Dynamic Host Configuration Protocol (DHCP) must be configured on the head node. DHCP allows the compute nodes to receive their network configuration from a central server. This step is vital for a scalable system, because it allows the nodes to be configured identically but have unique IP numbers.

To configure DHCP on the head node, complete these steps:

1) Install DHCP:

```
# yum install dhcp
```

2) Add the following to /etc/dhcp/dhcpd.conf:

```
#dhcpd config options
authoritative;
default-lease-time -1;
option broadcast-address 192.168.1.255;
ddns-update-style none;
next-server 192.168.1.100;
filename "pxelinux.0";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.101 192.168.1.199;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 8.8.8.8;
    option routers 192.168.1.100;
}
```

Change the range option to include enough addresses for all of your nodes. Note that some options may need to be adjusted if the head node's internal IP is not 192.168.1.100. The `domain-name-servers` option may point to whatever DNS you desire; in this example we use Google's at 8.8.8.8, although some network administrators may require you to use their own.

3) To start the DHCP service on the head node at boot time, execute the command:

```
# systemctl enable dhcpd
```

And to start the service immediately, execute the command:

```
# systemctl start dhcpd
```

4) To automate this process, in `ks.cfg`, append `dhcp` to the `%packages` list, and between the `%post` and `%end` tags, add the following:

```
#dhcp
cp -p /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.000
cat > /etc/dhcp/dhcpd.conf << EOF
    [Contents of dhcpd.conf as determined in step 2 go here]
EOF
systemctl enable dhcpd
```

5) To use DHCP, the firewall must allow it. Since you want all communications on the internal network to go unobstructed, you simply can add the interface on that network to `firewalld`'s trusted zone:

```
# firewall-cmd --permanent --zone=trusted
  ↳--change-interface=[INTERNAL INTERFACE]
# echo "ZONE=trusted" >>
  ↳/etc/sysconfig/network-scripts/ifcfg-[INTERNAL INTERFACE]
# nmcli con reload
# firewall-cmd --reload
```

COMPUTE NODES HAVE A DIFFERENT PARTITION TABLE, NETWORK SETUP AND PACKAGE SELECTION FROM THE HEAD NODE.

Use the `ip addr` command to discover [INTERNAL INTERFACE]. Add these changes to the kickstart using the following lines:

```
firewall-offline-cmd --zone=trusted
  ➔ --change-interface=[INTERNAL INTERFACE]
echo "ZONE=trusted" >>
  ➔ /etc/sysconfig/network-scripts/ifcfg-[INTERNAL INTERFACE]
```

Note: we'll include a more thorough explanation on `firewalld` in Part III of this series.

Now that the head node is installed fully automatically, and support for a basic network is in place, you can proceed to the compute nodes.

Compute Node Manual Installation

On a single compute node, boot from the installation DVD and perform a manual installation.

Compute nodes have a different partition table, network setup and package selection from the head node. The example partitioning scheme we used in Part 1 of this series is:

- / — 200GB
- /scratch — rest of space

When configuring networking for the compute node, connect using

a dynamic IP address. If DHCP is set up on the head node correctly, you should receive an address.

For now, make the package selection a Minimal installation.

The generated kickstart file on the newly installed compute node is located at `/root/anaconda-ks.cfg` and should be transferred to the head node. In `/admin`, create a directory to house the compute node kickstart file:

```
# mkdir /admin/ks/computenode/
```

Copy it over the network from the compute node to the head node:

```
# scp 192.168.1.101:/root/anaconda-ks.cfg /admin/ks/computenode/ks.cfg
```

substituting the node's actual IP address as determined by running the `ip addr` command on it.

Compute Node Automated Installation

Automated installations are incredibly important for compute nodes in a cluster. While manual installations may be practical (though still undesirable because of reliability) for small test clusters, they scale poorly and are impractical for medium to large clusters.

For the compute nodes, the process of automating the installation will be similar to that of the head node. In the kickstart file, be sure to disable SELinux as you did for the head node, and also disable the firewall because the compute nodes won't be connected to the outside world anyway:

```
selinux --disabled  
firewall --disabled
```

Like with the head node, you'll transfer responsibilities away from the DVD. This time, rather than the final destination for installation files being the compute node itself, it will be the head node accessed over the network. In the final configuration shown in Table 2, the compute nodes will boot over Pre-boot eXecution Environment (PXE),

Table 2. Installation Plan for the Compute Nodes

METHOD	BOOT/ INSTALLER LOCATION	KICKSTART LOCATION	DISTRO LOCATION
Manual	DVD	None	DVD
Automated DVD	DVD	USB	DVD
Automated NFS	DVD	NFS	NFS
Automated PXE	PXE	NFS	NFS

and they'll retrieve their kickstart files and distributions via a Network File System (NFS).

DVD-Based Booting with Kickstart on USB

The procedure here is the same as under the head node. If you are still learning about kickstart files, feel free to follow the DVD-Based Booting with Kickstart on USB instructions under the head node section. But otherwise, save yourself time and use NFS as described in the next section.

DVD-Based Booting with Kickstart and Distro on NFS

NFS allows compute nodes to access files stored on the head node over the internal network. This is a necessary step for the scalability of the cluster, since it would be impractical to have a copy of the kickstart and distro locally on each node at installation time.

In this section, we assume that you've configured the directory structure on the head node as follows:

- /admin/boot contains the contents of the DVD as in the Head Node Installation Sans Removable Media section.
- /admin/ks/computenode/ks.cfg is the kickstart file for the compute nodes.
- The head node's IP address on the internal network is 192.168.1.100.

- The head node is assigning addresses using DHCP (this was verified during the compute node manual installation).

If your setup differs, modify the following commands accordingly. To install over NFS, do the following steps.

1) On the head node, open the `/etc/exports` file in a text editor and add the following:

```
/admin 192.168.1.100/255.255.255.0(ro,sync,no_root_squash)
```

This gives all computers on the 192.168.1.0/24 subnet read-only (ro) access to the `/admin` directory.

2) Restart the NFS service so that this change takes effect. If NFS wasn't already running, this will start it:

```
# systemctl restart nfs
```

3) In the compute node `ks.cfg` file (located on the head node), comment out the repository line and replace it with:

```
nfs --server=192.168.1.100 --dir=/admin/boot
```

This tells the installer to look for the installation files on the network rather than on a DVD.

4) On a compute node, determine the name of the network interface that connects to the internal network using the `ip addr` command. The interface name could be formatted in one of several different ways depending on your motherboard, such as `ethX`, `enoX`, `enpXsY`, or if you're unlucky, `en<mac address>`.

5) Now you can use the kickstart file while booting from the DVD by pressing Tab at the initial welcome screen and appending the following to the boot options:

```
ks=nfs:192.168.1.100:/admin/ks/computenode/ks.cfg ksdevice=ethX
```

substituting `ethX` for the interface name found in step 4.

THINK ABOUT IT: THROW A SWITCH, TAKE A LUNCH BREAK, AND WHEN YOU GET BACK, THE ENTIRE CLUSTER IS INSTALLED. THAT'S SCALABILITY!

6) To make the changes on the head node persistent between installs, add the following to the end of the `ks.cfg` file for the head node between the `%post` and `%end` tags:

```
#nfs
echo "/admin 192.168.1.100/255.255.255.0(rw, sync, no_root_squash)" >>
  /etc/exports
systemctl enable nfs
```

The compute nodes now are capable of retrieving kickstarts and installation files over the network, but that is only half the story. To make the installation proceed without any media, the nodes must boot over the network as well.

PXE-Based NFS

Pre-boot eXecution Environment (PXE), called MBA on some BIOSes, allows nodes to retrieve their boot media via the network. Here are some basic prerequisites before using PXE:

- Your motherboard supports PXE.
- PXE is enabled in your BIOS.
- PXE is set before local boot methods on the BIOS boot order.

PXE allows you to perform kickstart installations on the nodes without having to load any disks physically. It's then possible to start an automated installation merely by powering on the cluster. Think about it: throw a switch, take a lunch break, and when you get back, the entire cluster is installed. That's scalability!

To make this claim a reality and install the compute nodes from the head node, do the following.

1) On the head node, install `syslinux`, `tftp-server` and `tftp` using `yum`. Add these to the Packages section of the kickstart file for documentation and re-installation purposes.

2) On the head node, make and populate the directory `/admin/tftpboot`:

```
# mkdir /admin/tftpboot
# cp /usr/share/syslinux/pxelinux.0 /admin/tftpboot/
# cp /usr/share/syslinux/menu.c32 /admin/tftpboot/
# mkdir -p /admin/tftpboot/images/centos7/
```

3) Copy in a compressed kernel and initial ramdisk from which the compute nodes can boot:

```
# cp /admin/boot/images/pxeboot/vmlinuz
  ↳/admin/tftpboot/images/centos7/
# cp /admin/boot/images/pxeboot/initrd.img
  ↳/admin/tftpboot/images/centos7/
```

Those two files are necessary for booting a bare-bones Linux system. `vmlinuz` is a compressed Linux kernel, and `initrd.img` is a temporary root filesystem. When you boot a compute node over PXE, those two files will be transferred to the compute node, giving it the software required to access the kickstart file and the rest of the installation files over NFS.

4) Create a directory to hold the PXE configuration files:

```
# mkdir /admin/tftpboot/pxelinux.cfg
```

5) Create the new `/admin/tftpboot/pxelinux.cfg/default` file

containing the following:

```

DEFAULT menu.c32
PROMPT 0
TIMEOUT 100
ONTIMEOUT kickstart
MENU TITLE PXE Menu
MENU separator
LABEL local
LOCALBOOT 0
MENU separator
LABEL    kickstart
        kernel images/centos7/vmlinuz
        append initrd=images/centos7/initrd.img
            ↪ks=nfs:192.168.1.100:/admin/ks/computenode/ks.cfg
            ↪ksdevice=ethX

```

Modify the `ksdevice` as needed. Notice that when the menu times out (`ONTIMEOUT`), it defaults to the `kickstart` option. This is useful for installing the cluster without manual intervention. But this must be changed later to `local` for the cluster to reboot without re-installing the OS.

6) Edit the `/usr/lib/systemd/system/tftp.service` file, changing the line:

```
ExecStart/usr/sbin/in.tftpd -s /var/lib/tftpboot
```

to:

```
ExecStart/usr/sbin/in.tftpd -s /admin/tftpboot
```

and restart the service so that this change takes effect. This change could be added to the head node's kickstart file using a `here` file, but it's more concise to use `sed`:

```

# sed -i.000 's|/var/lib/tftpboot|/admin/tftpboot|'
# /usr/lib/systemd/system/tftp.service

```

In this command, the `-i.000` flag specifies that `sed` is to do the modification in place—that is, it will perform the change and write it back to the original file. The `.000` part makes it so that `sed` will save the original file with a `.000` extension as a backup. The next component, in single quotes, specifies the operation that `sed` is to perform. The `s` tells it to do a substitution (as opposed to a deletion or insertion), the pipe (`|`) serves as a delimiter between parts of the command, and the two strings are the parts to exchange. Finally, the file path at the end of the command specifies the file that `sed` operates on.

7) Reboot a compute node—or all of them. If everything is set up correctly, they will install automatically.

8) Change `ONTIMEOUT kickstart` to `ONTIMEOUT local`. Otherwise, every reboot will result in a re-install.

An explanation on the inner workings of PXE booting is in order. When `pxelinux.0` boots on a machine, it tftp's back to the boot server and tries to find a configuration file in the `pxelinux.cfg` directory. The filename is determined by converting the IP address given to it by the DHCP server to hexadecimal. For example:

```
192 168 1 101
C0 A8 01 65 -> C0A80165
```

`pxelinux.0` attempts to find files in `pxelinux.cfg` in the following order:

- C0A80165
- C0A8016
- C0A801
- C0A80
- C0A8

THIS PXE FEATURE IS USEFUL FOR SUPPLYING SPECIFIC KICKSTART FILES FOR DIFFERENT SETS OF COMPUTE NODES BECAUSE OF HARDWARE DIFFERENCES; FOR EXAMPLE, IT CAN SUPPLY DIFFERENT PARTITIONING SCHEMES FOR DIFFERENT HARD DISK SIZES.

- C0A
- C0
- C
- default

This PXE feature is useful for supplying specific kickstart files for different sets of compute nodes because of hardware differences; for example, it can supply different partitioning schemes for different hard disk sizes. Right now DHCP on the head node is configured to dole out IP addresses in an arbitrary order, making it hard to take advantage of this feature. (In the next article, we'll fix that.)

When booting a compute node over the network, `mlinuz` (the compressed kernel) and `initrd.img` (the compressed initial filesystem) are transferred back to the compute node, along with the boot options.

In the case with the `kickstart` option, the boot options tell the installer (included in `initrd.img`) the location from which to retrieve

ks.cfg, which in turn includes the location of the distro.

PXE also is useful for booting other images, such as Memtest and other diagnostic tools.

Conclusion

In this article, we covered the basics of kickstart files on CentOS, and we set up a scalable method for installing the entire cluster. The resulting system is capable of intercommunication over ssh as root, but it doesn't have any useful cluster-wide application software or users on it yet.

In the final article, we'll address the Linux services that are vital for cluster operation, culminating on a resource manager called SLURM. With this software in place, the cluster will be fully fledged and ready for its end users. ■

Nathan Vance is a computer science major at Hope College in Holland, Michigan. He discovered Linux as a high-school junior and currently uses Arch Linux. In his free time, he enjoys running, skiing and writing software.

Mike Poublon is a senior data-center network engineer and technical lead at Secant Technologies in Kalamazoo, Michigan. He has extensive professional experience in networking and high-performance computing systems. As a student, he built Hope College's first production computer cluster.

William Polik is a computational chemistry professor at Hope College in Holland, Michigan. His research involves high-accuracy quantum chemistry using computer clusters. He co-founded WebMO LLC, a software company that provides web and portable device interfaces to computational chemistry programs.

Send comments or feedback via
<http://www.linuxjournal.com/contact>
or to ljeditor@linuxjournal.com.

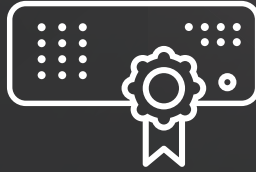
[RETURN TO CONTENTS](#)

SUPERMICRO[®] MARKETPLACE

Powered by Silicon Mechanics



**Broad
Selection**



**Zero
Defects**



**3-Year
Warranty**

Your Source for Supermicro Platform Technology

Twin, TwinPro & BigTwin

High-density, high-value servers



Configure
Now

FatTwin

Highest performance per watt



Configure
Now

SuperStorage

Flexible and efficient storage



Configure
Now

1U Servers

Entry & enterprise 1U form factor servers



Configure
Now

2U Servers

Flexible 2U form factor servers



Configure
Now

3U+ Servers

All 3U and larger form factor servers



Configure
Now

Ultra Servers

Unrivaled performance, flexibility & scalability



Configure
Now

MP Servers

Servers based on the Intel® Xeon® E7 Product Family



Configure
Now

SuperWorkstations

Server-grade performance at your desk



Configure
Now

Talk to a Supermicro Expert! [866.352.1173](tel:866.352.1173)

TESTING THE WATERS

How to Perform Internal Phishing Campaigns

Use Gophish to evaluate phishing risks
in your organization.

JERAMIAH BOWLING

PREVIOUS



Feature: BYOC: Build
Your Own Cluster,
Part II—Installation

NEXT
Doc Searls' EOF



Phishing is one of the most dangerous threats to modern computing. Phishing attacks have evolved from sloppily written mass email blasts to targeted attacks designed to fool even the most cautious users. No defense is bulletproof, and most experts agree education and common sense are the best tools to combat the problem. The question is how can you safely test your users to determine their response? The answer in most cases is a phishing campaign—an ongoing attempt to test your own users on these types of risks.

In this article, I examine an open-source tool called Gophish (<http://www.getgophish.com>) that fits the bill for most businesses. I describe how to perform multiple phishing campaigns with Gophish and create a foundation for ongoing testing. For the example campaigns, I have selected three popular types of phishing threats: malicious links within the body of an email that redirect to unwanted sites, links to phony sites that can capture credentials and, finally, attachment-borne malware.

Before proceeding, I feel the need to insert a few disclaimers. One, do not perform this work at your business, or any business for that matter, without the express written approval from that company's management. Two, make sure to define the scope of your campaign. What types of attacks will you use? Who do you want to target? What is the time frame for your campaigns? Answer as many of these questions as thoroughly as you can. Three, don't diverge from your scope. Limit your testing only to defined areas. Follow these disclaimers, and if you do encounter any issues arising from your campaigns, always use caution and consult with the same management that signed off on them.

Installing Gophish is a snap. You can install it on Linux or Windows. I chose to use a CentOS 7 distribution for my Gophish server. To install the program, simply download and extract the install file provided on the project's site. In my case, I extracted it to the /etc folder. Use the `chmod` command to allow the Gophish executable to run.

To start the program, run `gophish` from a terminal window. This launches a script that starts the various components of the Gophish program. Once the script has completed, you are notified that an admin page is running on `http://127.0.0.1:3333` (Figure 1). Open a browser on the local machine and log in with the default credentials of "admin/gophish". Upon logging in, you are presented with a minimalist

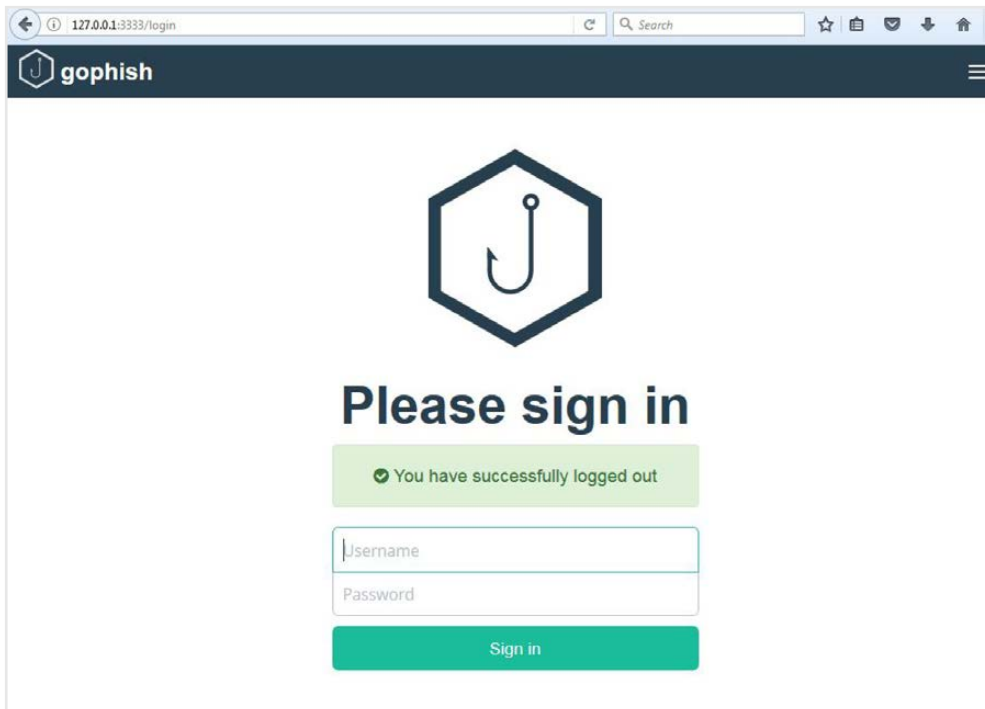


Figure 1.
Gophish Login
Page

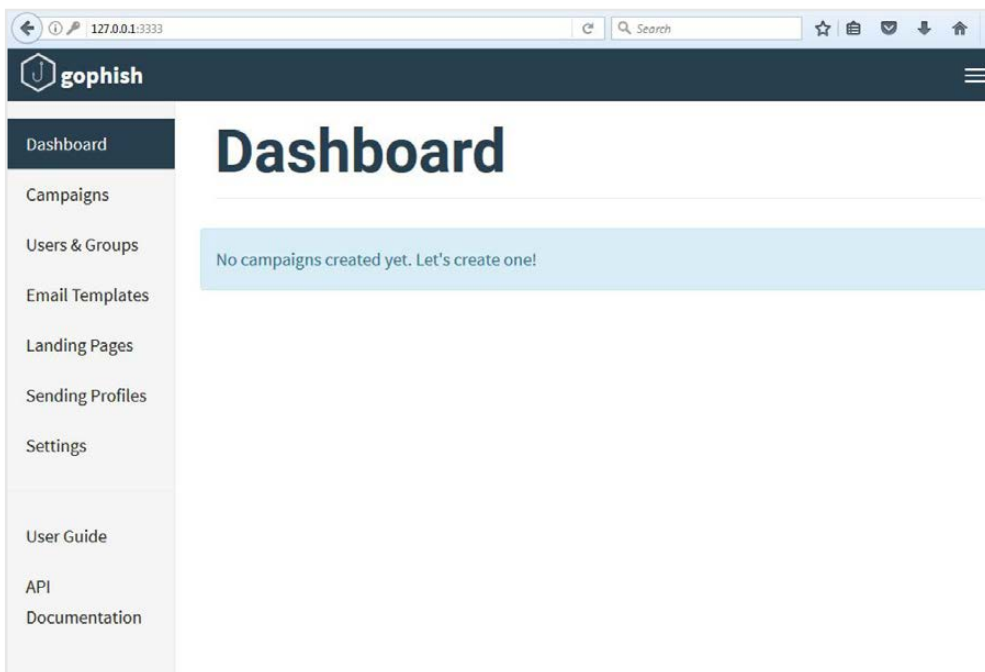


Figure 2.
Gophish
Interface

interface from which you can start working (Figure 2).

Before proceeding to the first campaign, you need to complete some preliminary work that will be re-used throughout your testing. The first item is to create a test domain and email address to use with your campaigns. It's generally a good idea to use a different

LEAVE BREADCRUMBS THAT CAN ASSIST USERS IN THE THREAT IDENTIFICATION PROCESS—THINGS LIKE MISPELLED WORDS, POOR GRAMMAR, STRANGE PHRASING AND SO ON.

email/domain combination for each campaign, but you're going to re-use this information between the campaigns to conserve space here.

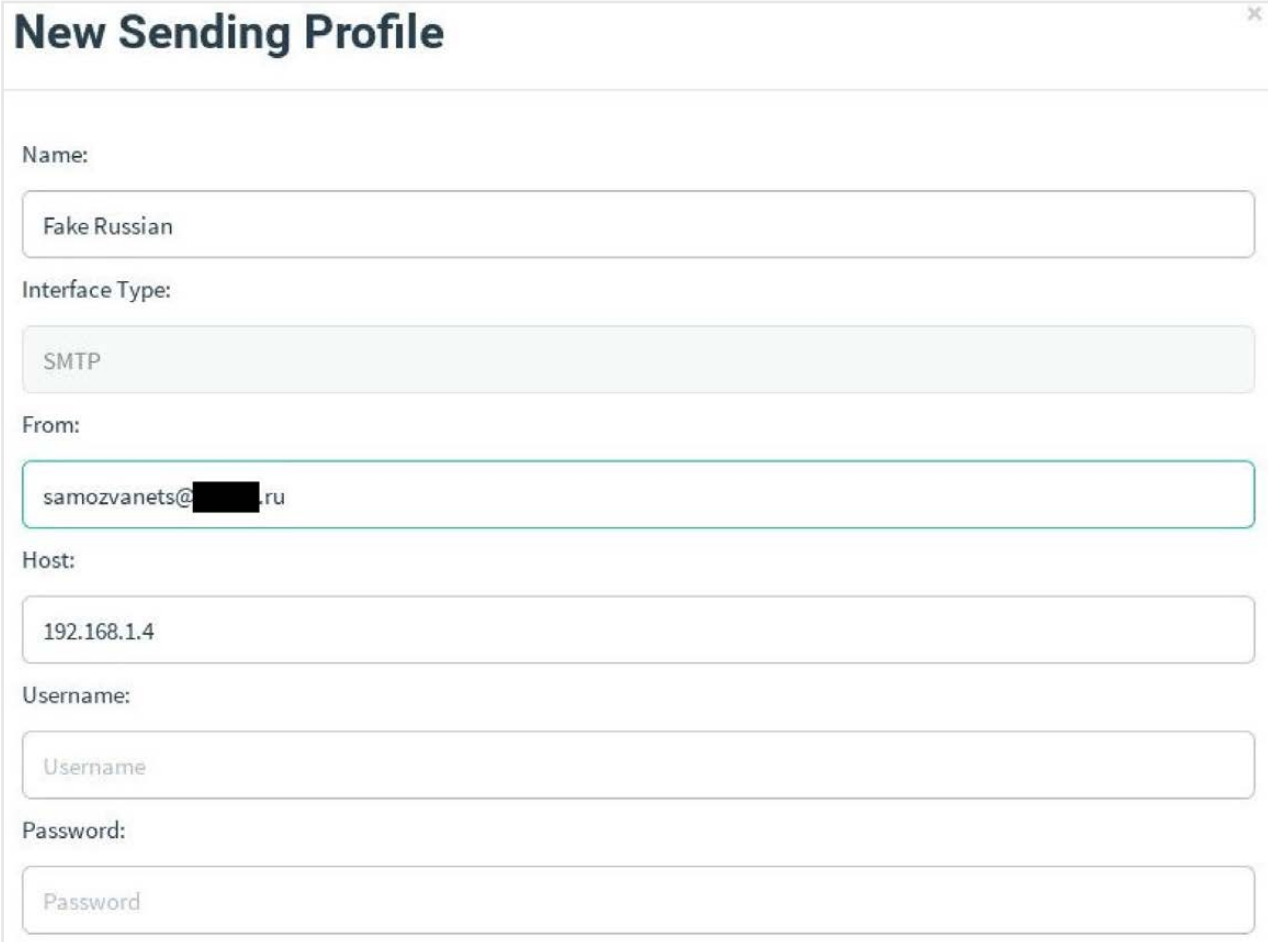
As it is a known haven of phishing attacks, I have chosen to use an unused Russian domain for these purposes. This obviously will not work well if you send or receive a lot of email from Russia.

Once you have chosen your domain or domains, create a DNS zone in your environment and create a host record for "tlbank". This host will come into play during your second campaign.

I landed on the e-mail address of somzvanets@fakerussiandomain (use an actual domain) as the testing address. Make sure to mark your domain as safe and/or whitelist it on any spam-filtering software or agents you have deployed in your environment. This also includes anti-virus, as many products combine protection into one agent.

Let me add one more thing. You have the tools and the ability to get really creative and successfully deceive your users. However, I believe the goal is not to dupe users completely, but to give them clues to trigger the critical thinking centers of their brains. It is specifically those skills that you want to test and measure, as they are the most valuable in combatting phishing attacks. Leave breadcrumbs that can assist users in the threat identification process—things like misspelled words, poor grammar, strange phrasing and so on. You have to give your users a hand through the process. Otherwise, you aren't really testing your users, you're simply testing your ability to deceive them. Now, on to the first campaign.

Campaigns in Gophish are made up of several components. The first



New Sending Profile

Name:
Fake Russian

Interface Type:
SMTP

From:
samozvanets@[REDACTED].ru

Host:
192.168.1.4

Username:
Username

Password:
Password

Figure 3. Sending Profile

is a Sending Profile. This is the phony address from which you will send mail. You can have multiple sending profiles on your Gophish server, but you can use only one at a time per campaign. Click on the Sending Profile link and fill in the fields displayed (Figure 3). Enter your fake address in the From field and enter an internal SMTP host.

Note, I strongly recommend using only internal resources available to you in your testing. Some paid phishing services are web- or cloud-based and may require additional network configuration. I like keeping everything inside so I absolutely know what is taking place when and on what hardware. It also will help keep your company's mail servers off internet blacklists. If your internal SMTP host requires a login, enter that as well. When your Profile information is complete, use the Send Test Mail button to confirm that your settings work. When you are

comfortable with your settings, click Save Profile.

The next component to configure is a Landing Page. A landing page is where the link in your phishing message will send users if they click the link you provide. Click the Landing Page link on the left. On the new window, name the page "Blank Page". For the first campaign, let's use a simple redirect page. Click the Source button and enter the following code in the space (enter a site your users commonly use in the url= section):

```
<html>
<head><meta http-equiv="refresh" content="0;
  ↪url=http://somewebsite/" />
  <title></title>
</head>
<body></body>
</html>
```

Click Save.

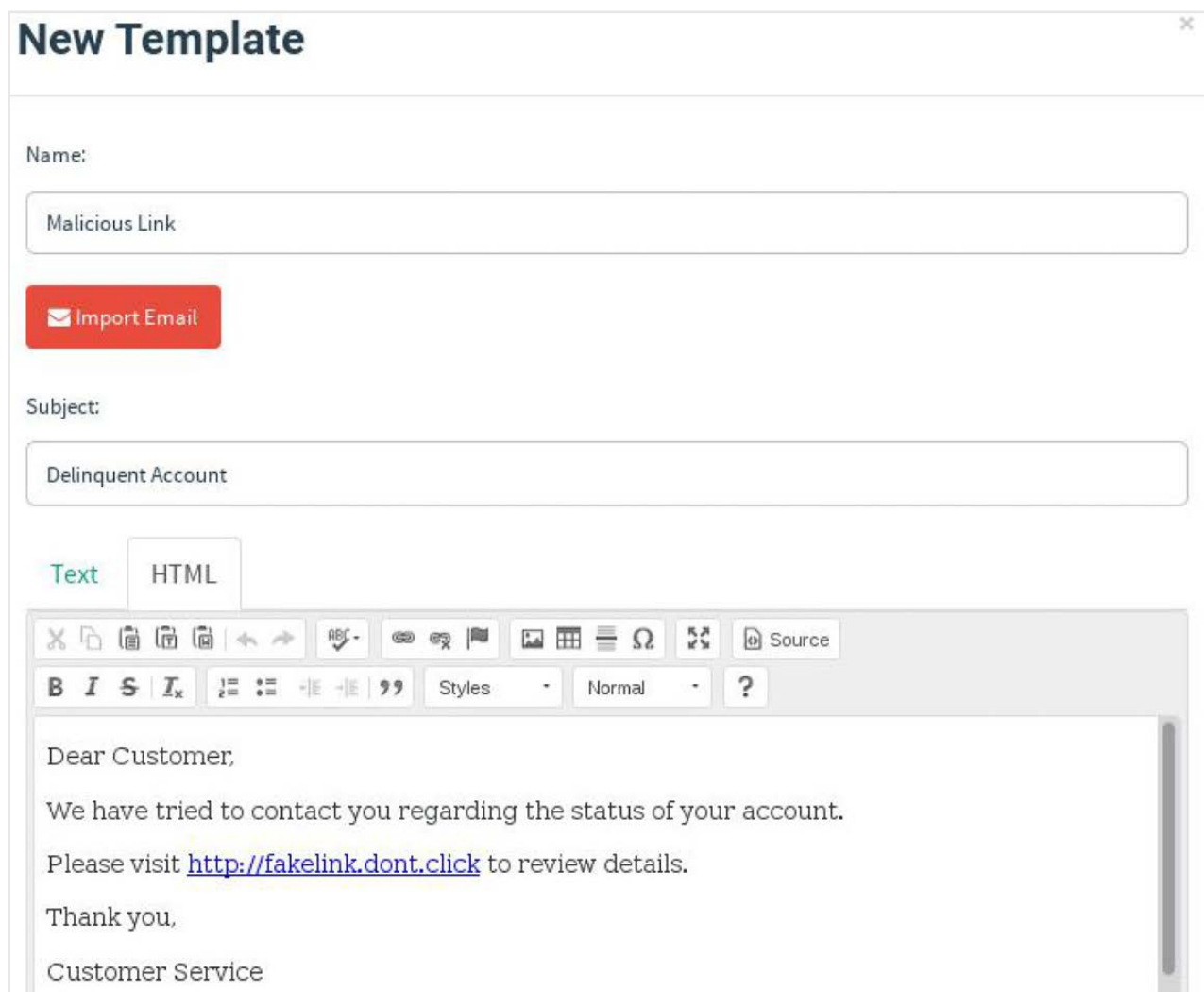
Next, you'll create your first Email Template. Templates are the email messages used in the campaign. Click the Email Templates link on the left, then click New Template at the top of the page. Give your template the name of "Malicious Link" as an identifier for this campaign. On the New Template screen, you have the option of creating your own template or importing a custom email. Here you'll use a simple message with a link to your Blank Page with the redirect code.

If you choose to use a custom email tailored after a real-world phishing message, do not directly use anything from the web. You can scrub those messages with a fine-tooth comb, but the last thing you want is to miss something that inadvertently brings malware onto your network. My advice is to transcribe any examples you want to use. Never copy and paste. Transcribing is the only sure-fire way to avoid accidentally using any malicious code in your testing. Thoroughly scan any images you want to download and use. Be cautious in using images that are not your own.

It is not necessarily a bad idea to create a template that resembles

a well-known company or, let's say, financial institution, but be aware there is a chance your users may actually use services from that company/financial institution. This becomes a double-edged sword. On the one hand, users actually may have a connection/account with the company you are impersonating, which could lead them to click on something they are not sure of. On the other hand, you want your users to view every message with a critical eye, even the ones that may affect them.


You can see the text of the message I've created in Figure 4. I have set the Subject to "Delinquent Account" as it is both generic and something that may still catch users' eyes. When creating the link in the message, use Link Type = URL and set the URL to {{.URL}} (Figure 5).



New Template

Name:

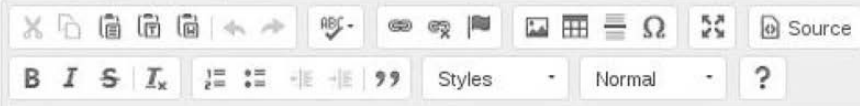
Malicious Link

 Import Email

Subject:

Delinquent Account

Text HTML



Dear Customer,

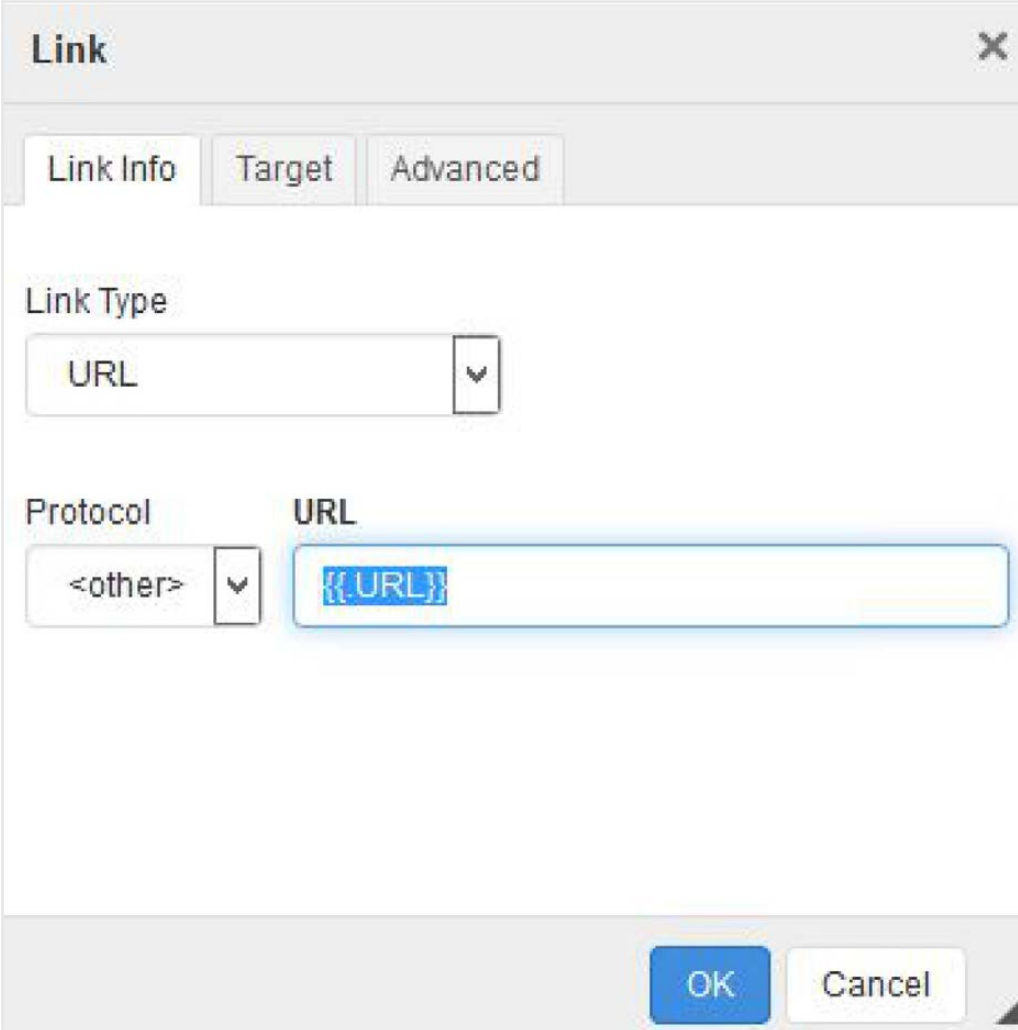
We have tried to contact you regarding the status of your account.

Please visit <http://fakelink.dont.click> to review details.

Thank you.

Customer Service

Figure 4. Creating a Template



The screenshot shows a 'Link' dialog box with three tabs: 'Link Info', 'Target', and 'Advanced'. The 'Link Info' tab is selected. Under 'Link Type', a dropdown menu shows 'URL'. Below that, the 'Protocol' dropdown menu is set to '<other>'. To the right of the protocol dropdown is a text input field labeled 'URL' containing the placeholder text '{{URL}}'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Figure 5.
Adding the
Landing
Page URL

This sends users who click the link to a unique URL on the Landing Page you just set up, which the Gophish server uses to track data for the campaign. Click on Save Template to save and close the template.

Click the Users & Groups link. Give the group a descriptive name, and add users either one at a time using the provided fields or bulk import a .csv file. Divide your users into groups as you feel necessary. You could divide them by geography or site, by job title or by internal groups. Use whatever strategy best matches your testing goals. I have simply named mine "Test Group".

With all the necessary components in place, click the Campaigns link, and fill out the fields using the items you created earlier to match Figure 6. In the URL field, enter the IP address or host name of your Gophish server. If you don't want the campaign to kick off right away,

click the Schedule button. You can see the test message as delivered to the test group in Figure 7.

New Campaign ✕

Name:

Email Template:

Landing Page:

URL: ?

Schedule:

Sending Profile:

Figure 6. Creating a New Campaign

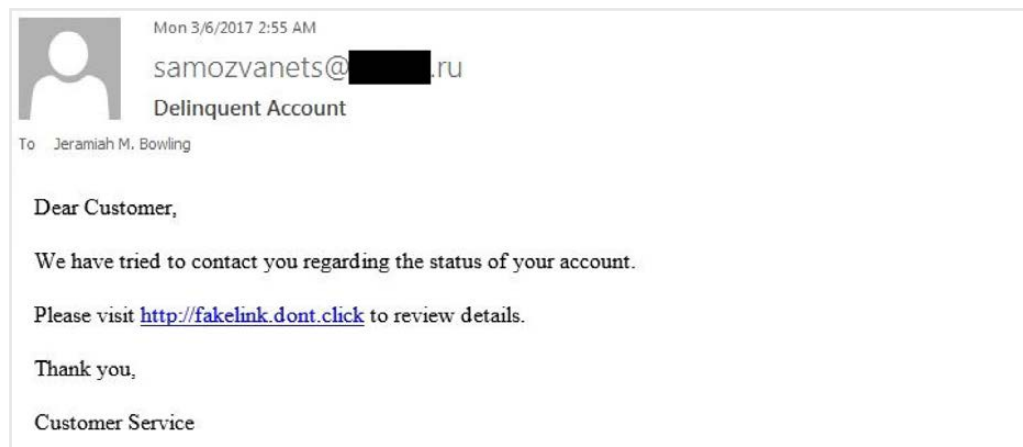


Figure 7.
Test Message

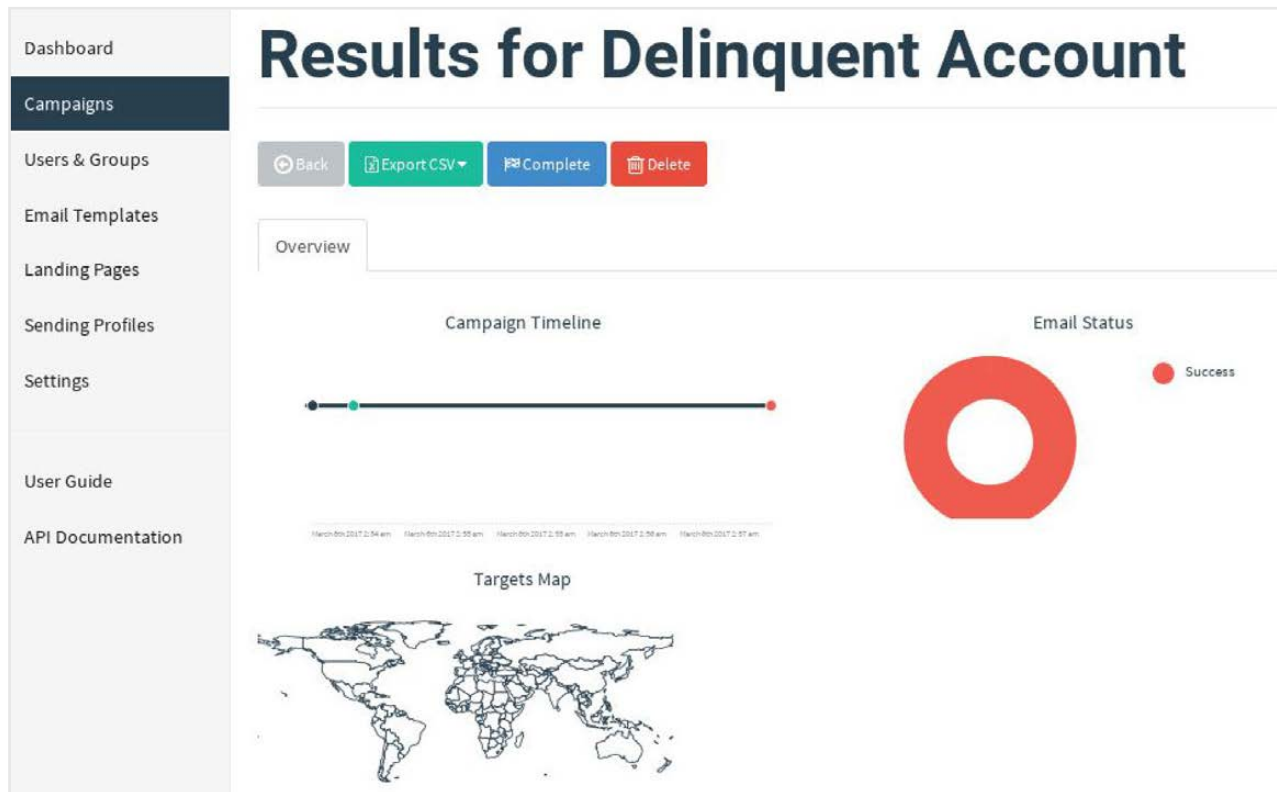


Figure 8. Viewing the Results

With the campaign underway, you can view the results on either the Dashboard or Campaigns link (Figure 8). Leave your campaigns in place for as long as you feel necessary. A few days normally should suffice, as users have short memories. When you are ready, you can complete your campaign by using the Complete button. You should see a Timeline displaying when the emails were sent and (if successful) when the link was clicked. If you scroll down, you'll see the results listed by the users in your group. Any success represents legitimate opportunities for a user to be lured to a malicious site using a phishing message.

The next campaign is centered around a phony web site that captures credentials. Because you're re-using the same Sending Profile for each campaign, you can move on to the Landing Page. This will be a simple page with a form input for a user name/password combo. On the new Landing Page window, enter the "Capture Credentials". You are free to use my basic HTML code below and customize it to your needs, but my suggestion is to use the Import Site feature to clone a real-world site that would require a login. I personally have received phishing email messages

of this sort—claiming to be from a well-known bank with which I have an account, a data provider I use or popular streaming service to which I subscribe. You may get better results by mimicking a real site than an obvious fake site like this one. To use my page, copy the code below in the source view of the Landing Page window. When you click the submit button, it'll redirect the user to whatever page you like. I have removed some of the style tags to keep the code short:

```
<html>
<head>
  <title>The Totally Legitimate Bank</title>
</head>
<body>
<h2><strong>$$$$$ The Totally Legitimate Bank
  ↳$$$$$</strong><br />
    <em>Your Trusted (wink, wink) Hometown Bank</em><br />
</h2>
<p>User ID:<input type="text" /></p>
<p>Password:<input type="password" /></p>
<p><input onclick="document.location.href
  ↳='http://tlbank.tresk.ru'" type="button"
  ↳value="Click to Login" /></p>
<p>&nbsp;</p>
<p><strong>Member FDIP</strong>&trade;2017</p>
</body>
</html>
```

You may notice the option to Capture Submitted Data under the code box. I wouldn't use this option unless management or the decision-makers explicitly agree to it in your scope. A phishing campaign can be paired with other kinds of security testing where it may be relevant to capture this data, but that's not what the goal is here.

For this new Email Template, as you may have guessed, you'll be impersonating the Totally Legitimate Bank. I have crafted the email shown in Figure 9 to entice users to visit the site. As before, use `{{.URL}}` for your link code in the body. When crafting phishing email

here to confirm.' followed by 'Customer Service 1-888-555-1234' and 'The Totally Legitimate Bank Member FDIP™2017'. The editor toolbar includes icons for undo, redo, bold, italic, strikethrough, bulleted list, numbered list, link, unlink, text color, background color, indent, outdent, quote, styles, normal, and source."/>

Name:

Fake Bank

Import Email

Subject:

Unusual Activity Detected

Text HTML

We have detected unusual activity on your bank account.
Please login to your account [here](#) to confirm.

Customer Service
1-888-555-1234
The Totally Legitimate Bank
Member FDIP™2017

body p font

Figure 9. Impersonating the Totally Legitimate Bank

that also uses a complementary site, it's important to match the branding (either real or fake). Users rarely fall for a site that doesn't match up visually or otherwise in these scenarios.

Click the Campaigns link and open a new campaign. Set the options to match Figure 10. The URL I've entered is the FQDN of the host (tlbank) created earlier. If your DNS server has records for the zone and host, a valid URL will show in the user's browser. This is important, as you don't want any savvy users familiar with your IP scheme to catch on just by looking at the URL. Click Launch Campaign when ready. When you monitor this campaign, you will see the "Event: Clicked Link", and if the user entered data into the fake site, you will see a second red dot with the "Event: Submitted Data" indicating a user submitted information in the form.

It's possible that users could have left the fields blank and clicked on

New Campaign ✕

Name:

Email Template:

Landing Page:

URL: ?

Schedule:

Sending Profile:
 ✉ Send Test Email

Figure 10. Creating Another New Campaign

the button, and there are two ways to deal with that if you want to be sure. One, code your form to check and make sure the fields are filled in before the submit occurs or capture the credentials, which I don't recommend. When you are satisfied with the results, complete the campaign. If you have a number of the second "Event: Submitted Data" messages in your results, you should be particularly concerned about your users' unknowingly submitting their credentials to an unknown party.

The third and last campaign involves sending users a malicious attachment. This is a very popular way to install ransomware. The two most currently used applications that infect users this way are Adobe Acrobat and Microsoft Word. Unfortunately, Gophish does not currently possess all of the tools needed to test this, so you'll need to set up additional resources for this campaign.

Like the previous “Totally Legitimate” web page, you’ll use the quick-and-dirty method to get what you need. There is so much more you can do with this type of test, especially with tools like Metasploit, but that is beyond the scope of this article.

Start by downloading a LAMP appliance from <https://www.turnkeylinux.org/lampstack>. I had mine up and running in less than five minutes. Create a web page called `verify.php` right off the root site using the code below:

```
<?php
session_start();
$_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
$counter_name = "/var/www/counter/counter.txt";
$iplog_name = "/var/www/counter/ip.txt";

// Check if a text file exists. If not create
// one and initialize it to zero.
if (!file_exists($counter_name)) {
    $f = fopen($counter_name, "w");
    fwrite($f, "0");
    fclose($f);
}

// Read the current value of our counter file
$f = fopen($counter_name, "r");
$counterVal = fread($f, filesize($counter_name));
fclose($f);

// Has visitor been counted in this session?
// If not, increase counter value by one and append ip.txt file
if(!isset($_SESSION['hasVisited'])){
    $_SESSION['hasVisited']="yes";
    $counterVal++;
    $f = fopen($counter_name, "w");
    fwrite($f, $counterVal);
    fclose($f);
    $file = fopen($iplog_name, "a");
```

```
$ip=$_SERVER['REMOTE_ADDR'];  
echo fwrite($file,$ip);  
echo fwrite($file, "\n");  
fclose ($file);  
header('Location: http://somewebsite');  
}  
header('Location: http://somewebsite');
```

This simple page will count the users as visiting, note their IP address in a text file and then redirect them to some external site.

Now, let's create the malicious attachment. Assuming you have Microsoft Word, open the program and a blank document, and press Alt-F11 to open the VB editor. Create a new module, and use the following code where `http://somewebsite` is the name of your LAMP web server:

```
Sub AutoOpen()  
myURL = http://mylampserver/verify.php  
ShellExecute 0, "OPEN", myURL, "", "", 0  
End Sub
```

Save the document as type `.docm`, and close out of Word.

Back on the Gophish server, create a new Email Template named "Malicious Attachment", and use the document file you created as an attachment by clicking on the Add Files button. See Figure 11 for the wording of the template.

In this example, you are claiming that the user has an unpaid invoice. You don't need a landing page, so set it as "Blank Page" like in the first campaign. Match the rest of the settings to Figure 12 and Launch the campaign. You can use a hostname in the URL field, but since you're not using Gophish to track the campaign, you can just use the the Gophish server's IP.

Unlike the previous campaigns, you'll have to track your results using the text files created with the `verify.php` page. One note to this campaign—most current word processors possess some form of macro protection, usually a warning prompt. Users will have to bypass those or enable macros to open the attachment, which means they

New Campaign ✕

Name:

Past Due Invoice

Email Template:

Malicious Attachment

Landing Page:

Blank Page

URL: ?

http://192.168.1.5

Schedule:

03/01/2017 9:22 AM

Sending Profile:

Fake Russian
 Send Test Email

Figure 12. Campaign Settings

is an old adage in computer security “The bad guys only have to be right once.” Make sure your users are prepared. With turnover, promotions and responsibility changes, the last thing on many users’ minds is email security. Consistent reinforcement of good security practices and regular testing to validate your training approach is crucial to avoiding catastrophe. ■

Jeremiah Bowling has been a systems administrator and network engineer for more than 17 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.

Send comments or feedback via <http://www.linuxjournal.com/contact> or to ljeditor@linuxjournal.com.

RETURN TO CONTENTS

drupalize.me

Instant Access to Premium Online Drupal Training

- ✓ *Instant access to hundreds of hours of Drupal training with new videos added every week!*
- ✓ *Learn from industry experts with real world experience building high profile sites*
- ✓ *Learn on the go wherever you are with apps for iOS, Android & Roku*
- ✓ *We also offer group accounts. Give your whole team access at a discounted rate!*

Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!

Go to <http://drupalize.me> and get Drupalized today!





A Field Guide to the World of Modern Data Stores

There are many types of databases and data analysis tools to choose from when building your application. Should you use a relational database? How about a key-value store? Maybe a document database? Is a graph database the right fit? What about polyglot persistence and the need for advanced analytics?

If you feel a bit overwhelmed, don't worry. This guide lays out the various database options and analytic solutions available to meet your app's unique needs.

You'll see how data can move across databases and development languages, so you can work in your favorite environment without the friction and productivity loss of the past.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/field-guide-world-modern-data-stores>



Why NoSQL? Your database options in the new non-relational world

The continual increase in web, mobile and IoT applications, alongside emerging trends shifting online consumer behavior and new classes of data, is causing developers to reevaluate how their data is stored and managed. Today's applications require a database that is capable of providing a scalable, flexible solution to efficiently and safely manage the massive flow of data to and from a global user base.

Developers and IT alike are finding it difficult, and sometimes even impossible, to quickly incorporate all of this data into the relational model while dynamically scaling to maintain the performance levels users demand. This is causing many to look at NoSQL databases for the flexibility they offer, and is a big reason why the global NoSQL market is forecasted to nearly double and reach USD3.4 billion in 2020.

Sponsor: IBM

> <https://geekguide.linuxjournal.com/content/why-nosql-your-database-options-new-non-relational-world>

Estimating CPU Per Query With Weighted Linear Regression



Your database server is suddenly using a lot of CPU resources. Quick, what caused it? This is a familiar question for engineers of all persuasions. And it's often impossible to answer.

There are good reasons why it's hard to figure out what consumes resources like CPU, IO, and memory in a complex piece of software such as a database. The first problem is that most database server software doesn't offer any way to measure or inspect that type of performance data. The database server isn't observable. This problem arises in turn from the complexity of the database server software and the way it does its work, which actually precludes measuring resource consumption accurately!

Author: Baron Schwartz

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/estimating-cpu-query-weighted-linear-regression>



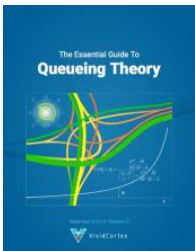
Database Performance Monitoring Buyer's Guide

More and more companies have begun to recognize database performance management as a vital need. Despite its widespread importance, good database performance management requires specialized expertise with custom approaches--yet all too often, organizations rely on one-size-fits-all solutions that theoretically "check the box" but in practice do little or nothing to help them find or prevent database-related outages and performance problems.

This buyer's guide is designed to help you understand what database management really requires, so your investments in a solution provide the greatest possible ultimate value.

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/database-performance-monitoring-buyer%E2%80%99s-guide>



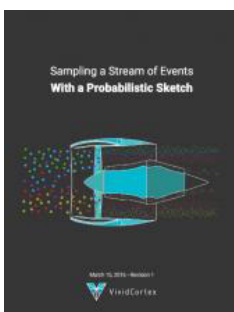
The Essential Guide To Queueing Theory

Whether you're an entrepreneur, engineer, or manager, learning about queueing theory is a great way to be more effective. Queueing theory is fundamental to getting good return on your efforts. That's because the results your systems and teams produce are heavily influenced by how much waiting takes place, and waiting is waste. Minimizing this waste is extremely important. It's one of the biggest levers you will find for improving the cost and performance of your teams and systems.

Author: Baron Schwartz

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/essential-guide-queueing-theory>



Sampling a Stream of Events With a Probabilistic Sketch

Stream processing is a hot topic today. As modern Big Data processing systems have evolved, stream processing has become recognized as a first-class citizen in the toolbox. That's because when you take away the how of Big Data and look at the underlying goals and end results, deriving real-time insights from huge, high-velocity, high-variety streams of data is a fundamental, core use case. This explains the explosive popularity of systems such as Apache Kafka, Apache Spark, Apache Samza, Apache Storm, and Apache Apex—to name just a few!

Author: Baron Schwartz

Sponsor: VividCortex

> <https://geekguide.linuxjournal.com/content/sampling-stream-events-probabilistic-sketch>

Open Source Comes of Age

How do you organize and manage something so hugely successful and widely varied? Not easy, but Harvard's Cyberlaw Clinic has some good advice.



DOC SEARLS

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.

PREVIOUS

◀ Feature: Testing the Waters: How to Perform Internal Phishing Campaigns

As of today (June 1, 2017), we've been talking about open source for exactly 19 years, 3 months and 23 days. The start date was February 8, 1998, when Eric S. Raymond distributed an open letter by email with the subject line *Goodbye, "free software"; hello, "open source"*. What followed was a deliberate (though barely coordinated) effort by many geeks (including yours truly and this magazine) to make open source a thing.

It worked. In books alone, the result looked like what's shown in Figure 1.

I am sure that the line would have continued rising toward the sky if Google hadn't tired of scanning books in 2008 (<https://backchannel.com/how-google-book-search-got-lost-c2d2cf77121d>).

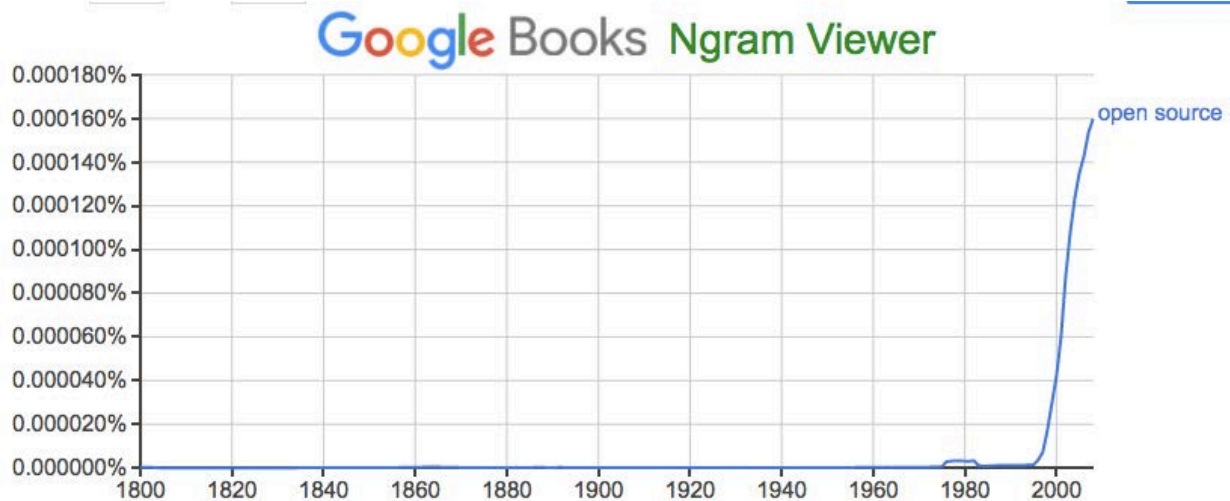


Figure 1. Google Books Ngram Viewer: Open Source

Anyway, we succeeded. As both an concept and a practice, open source is embedded in technology, business, culture, government—you name it. In fact, it is so widely uttered, you might even call it mature.

But it's not, because making full sense of open-source development is still an uphill struggle, especially if you're an organization trying to manage it—especially in a world that still doesn't fully understand it, even though it gets talked about constantly.

This is why it's good to have help such as just came from Organization and Structure of Open Source Development Initiatives (<https://dash.harvard.edu/handle/1/30805146>), a new report by Dalia Topelson Ritvo (<https://tara.law.harvard.edu/cyberlawclinic/bio/dalia-topelson-ritvo>), Kira Hessekiel (<https://cyber.harvard.edu/people/khessekiel>) and Christopher T. Bavitz (<https://cyber.harvard.edu/people/cbavitz>) of the Cyberlaw Clinic (<https://cyber.harvard.edu/teaching/cyberlawclinic>) at the Berkman Klein Center for Internet & Society (<https://cyber.harvard.edu>) and Harvard Law School. (Disclosure: these are all colleagues through ProjectVRM and the Berkman Klein Center, where I am an alumnus fellow.)

The angle of the report is organizational, and the organizations it addresses are less those of the development efforts themselves than of companies that need to get some kind of handle—or several handles—on the simple fact that they already support open-source work, either by employing developers of open-source code or because they have a code base they would like to open

up and release to the world. There are as many answers to *What should we do?* as there are companies and developers, which also doesn't make things easy. But there are controlling factors in the real world that can help guide decisions, even as the same factors can be deeply frustrating.

For example, taxes.

Early on it was easy for an open-source project in the US to obtain 501(c)(3) status, which relieves an organization of the need to pay taxes. Apache and Mozilla both got their 501(c)(3) status right out of the gate, without much sweat. Coming along later, the Open Source Elections Technology Foundation (OSET) had to wait six years for it, after an initial denial by the Internal Revenue Service. Then there were the IRS's BOLO—"Be On the Look-Out"—lists. According to the report, "Every released BOLO dated between August 2010 and July 2012 included open source organizations." Why? Because, said the IRS, "[t]he members of these organizations are usually the for-profit business or for-profit support technicians of the software. The software is provided for free, however; fees are charged for technical support by the for-profit." This, says the report, "misrepresented the wide variety of open source business models, ignoring that many open source organizations do not have affiliations with for-profit businesses."

BOLOs are gone now, but the IRS's sphincter remains no less tight. Concludes the report:

While some open source software organizations may continue to obtain 501(c)(3) status, this option may be closed off—in particular, for organizations that choose not to or cannot offer activities to educate the public in software development or a demonstrable connection to a charitable purpose. Given these changes, open source software organizations can no longer count on obtaining 501(c)(3) status and may want to evaluate the pros and cons of adopting alternative business structures.

Those include 501(c)(4) and 501(c)(6), which are alternative tax-exempt status recognitions. The former is for "civic leagues or organizations... devoted exclusively to charitable, educational, or recreational purposes". The latter is for "business leagues". Specifically:

In the open source software community, 501(c)(6) organizations generally

act as umbrella organizations for many projects rather than working directly on a single project. For example, the Dojo Foundation and the Eclipse Foundation are 501(c)(6) organizations that support open source software projects.²⁵ But, a shift in IRS policy toward open source software organizations applying for 501(c)(6) status—which parallels the shift regarding 501(c)(3) applications—may be underway.

That happened, for example, when the IRS turned down the OpenStack Foundation’s application for 501(c)(6) recognition.

Advice:

Given the complexities associated with achieving exemptions from federal income tax, open source software organizations may have to begin reassessing their business models. Organizations may decide to continue to operate as nonprofits while paying federal income taxes, generate and rely on profits, or adopt a middle path between those two extremes.

There are many choices for middle paths: Limited Liability, S, C and Benefit Corporations, for example. The report unpacks all of them.

Then there’s the matter of governance. Back in the early days, governance wasn’t much of an issue. It still isn’t in cases where leadership is strong and clear, such as with Linux. But in countless other cases, governance can get complicated and contentious—because humans, basically.

Recently, for example, the Drupal Association went through a major crisis when one of its main contributors was exiled for an unspecified infraction involving sexual kinks—or...something. Not clear—if you want the particulars, links abound on the web. The two most relevant to the governance issue, however (at least at the time of this writing), are “Working through the concerns of our community” (March 31, 2017, <https://www.drupal.org/association/blog/working-through-the-concerns-of-our-community>) and “Next steps for evolving Drupal’s governance” (April 10, 2017, <http://buytaert.net/next-steps-for-evolving-drupal-governance>).

Both Linus Torvalds and Dries Buytaert (of Drupal) are what the report calls benevolent dictators. But while Linux remains at the bazaar end of the bazaar-cathedral spectrum of projects, Drupal has some cathedral

elements. For example, Dries also runs Acquia, a for-profit company that deploys Drupal in the world.

The report also describes other models: meritocracies (Apache for example), delegated governance (Ubuntu, possibly, with qualifications), dynamic governance (or sociocracy). There are also models from outside the open-source world—for example, federated nonprofits, aka “Model E”. Examples of that are the Girl Scouts and the American Red Cross. These are important to recognize and learn from, because they’ve been around a long time and have survived troubles that many open-source projects still haven’t encountered. Hence this advice:

One major challenge for all organizations—but particularly those focused on specific skill sets (like institutions engaged in open-source software development)—is recognizing the value of outside perspectives and non-technical skills. Because the heart of every open source project is the development and maintenance of the project’s code, it’s not always apparent how non-technical contributions should be recognized, or how they might improve development processes or community relations. In the same vein, those who are the best developers and code contributors may not always possess the soft skills required for open-source leadership roles.

As we know.

Bottom lines:

When open source creators launch new projects, their primary concerns may be technical. But expanding their focus to the organizational can have enormous benefits. Projects that make thoughtful decisions around corporate formation may streamline dealings with the IRS and also create stable entities that will help them be sustainable and dedicated to their ultimate missions. Projects that spend time considering questions about corporate formation set themselves up to create a welcoming project community with engaged and invested contributors; and then govern that community effectively....

Though it is sometimes overlooked, the history of the open source movement shows us that the projects that defined their corporate structure and governance practices early and concretely set themselves up for

success. While some elements of the landscape have changed—most notably the IRS’s attitude toward open source projects—the benefits of intentionality remain.

To me, the most important issue is the tax one, because we still haven’t made clear the leverage on both technology and the economy that comes from nonprofit work on open-source code. If we had made this clear, the IRS’s job would be easier today, and fewer projects seeking nonprofit status would be stuck in limbo.

But maybe that job is an impossible one.

A few years back, a former FCC chairman told me there were two things that federal lawmakers, almost across the board, didn’t understand: “One is technology and the other is economics.”

Still, it’s worth trying.

Maybe the sell is to ask them if they think technology is a Good Thing. If they say yes, we can add that open source causes commercial successes, and that it’s not the other way around. In other words, if you want to maximize commercial successes and technology benefits in the world, go ahead and give nonprofit status to open-source projects that want it. The good work those projects do will throw off a lot more tax money as well. ■

Send comments or feedback via <http://www.linuxjournal.com/contact> or to ljeditor@linuxjournal.com.

RETURN TO CONTENTS

ADVERTISER INDEX

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

ADVERTISER	URL	PAGE #
AnDevCon	http://www.AnDevCon.com	51
ASCEND	http://ascend-event.com	65
Drupalize.me	http://drupalize.me	113
InterDrone	http://www.InterDrone.com	41
Peer 1 Hosting	http://go.peer1.com/linux	122
Silicon Mechanics	http://www.siliconmechanics.com	95
SUSE	http://suse.com/storage	7
WITI	http://www.witi.com/	27

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>



Where every interaction matters.

break down your innovation barriers

power your business to its full potential

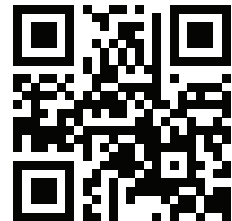
When you're presented with new opportunities, you want to focus on turning them into successes, not whether your IT solution can support them.

Peer 1 Hosting powers your business with our wholly owned FastFiber Network™, global footprint, and offers professionally managed public and private cloud solutions that are secure, scalable, and customized for your business.

Unsurpassed performance and reliability help build your business foundation to be rock-solid, ready for high growth, and deliver the fast user experience your customers expect.

Want more on cloud?

Call: 844.855.6655 | go.peer1.com/linux | [View Cloud Webinar:](#)



Public and Private Cloud | Managed Hosting | Dedicated Hosting | Colocation