

MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

ALL ABOUT FREENAS

INSIDE

WHICH FREENAS? LESSONS FROM A YEAR IN THE TRENCHES WITH BSD'S KILLER APP

FREENAS PLUGINS: EVERYTHING YOU EVER WANTED TO KNOW

HANDMADE NAS – PLUG IN MORE COMPONENTS TO ENRICH ITS FEATURES

INTERVIEW WITH ALFRED PERLSTEIN, VP OF SOFTWARE ENGINEERING AT IXSYSTEMS

FROM READING TO REAL LIFE – TRY OUT THE DEMO HOST OF FREENAS

WHAT'S NEW IN FREENAS 8.X BY DRU LAVIGNE

CASE STUDIES ON FREENAS IN AN ENTERPRISE ENVIRONMENT

VOL.7 NO.4
ISSUE 04/2013(45)
1898-9144



855-GREP-4-IX
www.iXsystems.com
Enterprise Servers and Storage
for Open Source

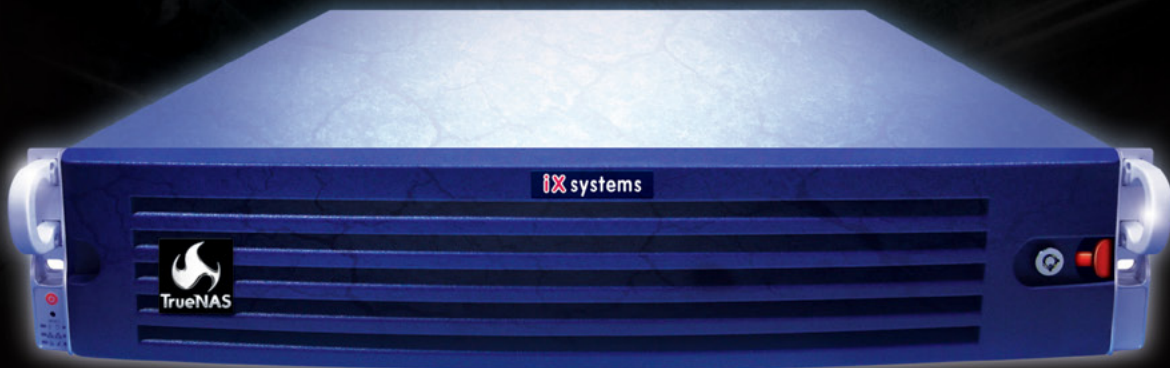


- ✓ Rock-Solid Performance
- ✓ Professional In-House Support



TrueNAS™

UNIFIED. SCALABLE. FLEXIBLE.



Across all industries the demands of data infrastructure have soared to new heights.

As capacity requirements continue to rise at an ever-increasing rate, performance must not be compromised. The hybrid architecture and advanced software capabilities of the TrueNAS appliance enable users to be more agile, effectively manage the explosion of unstructured data and deploy a centralized information storage infrastructure. Whether it's backing virtual machines, business applications, or web services, there's a TrueNAS appliance suited to the task.

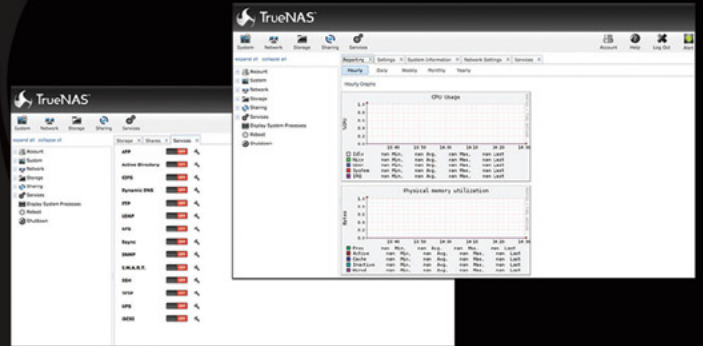
TrueNAS™ Storage Appliances: Harness The Cloud

iXsystems' TrueNAS Appliances offer scalable high-throughput, low latency storage

All TrueNAS Storage Appliances feature the Intel® Xeon® Processors 5600 series, powering the fastest data transfer speeds and lowest latency possible. TrueNAS appliances come in three lines: Performance, Archiver, & High Availability. High-performance, high-capacity ioMemory modules from Fusion-io are available in the TrueNAS Enterprise, Ultimate, and Archiver Pro models.

Key Features:

- One or Two Six-Core Intel® Xeon® Processors 5600 series
- Share Data over CIFS, NFS and iSCSI
- Hybrid storage pool increases performance and decreases energy footprint
- 128-bit ZFS file system with up to triple parity software RAID



*Optional component

	TrueNAS Pro	TrueNAS Enterprise	TrueNAS Ultimate	TrueNAS Fileshare	TrueNAS Archiver Pro	TrueNAS Pro-HA	TrueNAS Enterprise-HA	TrueNAS Ultimate-HA
	PERFORMANCE			ARCHIVER		HIGH AVAILABILITY		
Fusion-io Card		X	X		X			
Deduplication					X			
High Availability						X	X	X
Gigabit NICs	Quad	Dual	Dual	Dual	Dual	Six	Quad	Dual
10 Gigabit NICs		Dual*	Quad*		Dual*			Dual
Max Main Memory	48Gb	96Gb	192Gb	48Gb	192Gb	48Gb	96Gb	192Gb
Max Capacity	220TB	500TB	1.5PB	580TB	2.2PB	250TB	310TB	1.4PB
Rack Units	2U	2U/4U	4U	2U	4U	3U	3U	Dual 3U



Call iXsystems toll free or visit our website today!
1-855-GREP-4-IX | www.iXsystems.com



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

MAGAZINE BSD

Dear Readers,

The full April issue is dedicated to FreeNAS. Thanks to iXsystems, the sponsor of the FreeNAS project, we have a great collection of articles from the FreeNAS development team.

At the end of the issue you will find a nice surprise, prepared by one of our contributors – Imad Soltani. It's an article about a hosted FreeNAS demo with details where to find it, so you can try out everything you've read about in this issue.

The issue starts with the Let's Talk section in which Rob Somerville and Michael Dexter will share with you their thoughts on FreeNAS.

Then, Dru Lavigne will guide you through the new features in the latest release of FreeNAS.

In Dev Corner, John Hixson describes the FreeNAS plugins system and how its architecture works, so plugins authors like him can create new ones.

Next, Edward Tan will show you how to build a NAS by plugging different components together using command line and FreeBSD.

In the section dedicated to business solutions, Mark VonFange and Annie Zhang will explain the difference between FreeNAS and TrueNAS. You will learn more about FreeNAS' big brother. Just after this read, Mark will share with you a case study about Creative Integrations, where both "brothers" did a good job.

The Successful Story section contains Luca Ferrari's article, where he describes his own experience with a data migration.

An interview with Alfred Perlstein – Vice President of Software Engineering at iXsystems, who is responsible for the FreeNAS Project – closes the issue.

We hope you will enjoy reading.

Some of you will be happy to hear that since April issue of BSD Magazine, we decided to publish it in epub format as well. Now, you can enjoy reading BSDMag on your tablets and other mobile devices:

Please, spread the word about BSD Magazine among your friends and colleagues.

Patrycja Przybyłowicz
Editor of BSD Magazine
& BSD Team

Editor in Chief:

Ewa Dudzic
ewa.dudzic@software.com.pl

Supportive Editor

Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Contributing:

Rob Somerville, Michael Dexter, Dru Lavigne, Edward Tan, John Hixson, Mark VonFange, Annie Zhang, Luca Ferrari, Imad Soltani

Top Betatesters & Proofreaders:

Ahmed Aneeth, Radjis Mahangoe, Barry Grumbine, Bjørn Michelsen, Paul McMath, Eric Geissinger, Eric De La Cruz Lugo, Imad Soltani, Luca Ferrari, Ewa Duranc

Special Thanks:

Denise Ebery
Matt Olander

Art Director:

Ireneusz Pogroszewski

DTP:

Ireneusz Pogroszewski
ireneusz.pogroszewski@software.com.pl

Cover Art:

Jenny Rosenberg
Lisa Liang

Senior Consultant/Publisher:

Paweł Marciniak
pawel@software.com.pl

CEO:

Ewa Dudzic
ewa.dudzic@software.com.pl

Production Director:

Andrzej Kuca
andrzej.kuca@software.com.pl

Advertising Sales:

Patrycja Przybyłowicz
patrycja.przybylowicz@software.com.pl

Publisher

Software Media Sp. z o.o. SK
ul. Bokserska 1, 02-682 Warszawa
Poland
worldwide publishing
tel: 1 917 338 36 31
www.bsdmag.org

Software Press Sp z o.o. SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org.

All trade marks presented in the magazine were used only for informative purposes. All rights to trade marks presented in the magazine are reserved by the companies which own them.

Mathematical formulas created by Design Science
MathType™.

Let's Talk

08 FreeNAS – Detachable Storage has Come of Age

By Rob Somerville

As FreeNAS is very light, it is ideal for installing on older hardware, and provided the target box has > 6GB of RAM, the security and additional functionality of ZFS is available facilitating snapshots.

10 Which FreeNAS? Lessons from a Year in the Trenches with BSD's Killer App

By Michael Dexter

We have developed an insatiable need for digital storage and most people probably don't realize how fragile their storage infrastructure is until it's too late. FreeNAS stands out from other Open Source storage appliances by bringing enterprise-class features to within the reach of just about everyone with the key feature being the ZFS "Zettabyte File System" developed by Sun Microsystems.

What's New

12 What's New in FreeNAS 8.x

By Dru Lavigne

This article highlights some of the new features which have been added to FreeNAS 8.x since July, 2012. These include the Plugins Jail, ZFSv28, and GELI encryption.

Developer's Corner

16 FreeNAS Plugins: Everything You Ever Wanted to Know

By John Hixson

Beginning with FreeNAS 8.2.0, a plugin architecture was introduced. This architecture allows FreeNAS systems to be extended in any way that the user sees fit. In this article John describes the technical details of how the architecture works so that plugin authors have the knowledge to create new plugins.

How To

22 Handmade NAS

By Edward Tan

From this article you will learn how to build a NAS by hand, plugging different components together using the command line and FreeBSD. It's a jumpstart guide on the basic components that make up a NAS, which the reader can expand on and enrich with more features.

In Business

28 FreeNAS in an Enterprise Environment

By Mark VonFange and Annie Zhang

Despite the massive popularity of FreeNAS, most people aren't aware of its big brother, dutifully storing and protecting data in some of the most demanding enterprise business environments: the proven, enterprise-ready, and professionally-supported line of appliances known as TrueNAS.

32 Creative Integrations: Workflow Improvements with FreeNAS and TrueNAS

By Mark VonFange

FreeNAS provides an Open Source storage platform for its customers to use in their media environment. TrueNAS has additional features that provide extra data security with commercial grade support from the developers that bring the world FreeNAS.

Successful Story

34 FreeNAS: a Migration Story

By Luca Ferrari

This article briefly summarizes the migration process and what advantages the usage of FreeNAS provided. This is not a technical article, it is just a "tale" of how the author has managed the migration.

Interview

38 The Interview with Alfred Perlstein, VP of Software Engineering at iXsystems

By Luca Ferrari & BSD Team

Alfred's areas of interest have been file systems, multi-processor support, performance, and stability of FreeBSD. His current role is FreeNAS project manager and VP Software Engineering at iXsystems. Recently, he agreed to give an interview to BSD Magazine.

Extras

44 From Reading to Real Life: Try Out a Demo of FreeNAS!

By Imad Soltani

Thanks to one of the BSD Magazine contributors you can try out what you have read about in this issue. In this article you will find all the needed details about the demo host of FreeNAS.

Why I Use FreeNAS

I work as a Systems Analyst, and as such I often experiment with different operating systems and software. Through the chaos of switching systems around, I need to ensure that my personal files are protected.

A few years ago I was looking to add a network NAS solution, to replace the USB connected hard drives I had been using to store my files. I had a spare computer with a couple of hard drives in it, so I wanted a solution that could use that hardware. I didn't want to run a full commercial server OS like a Windows or Novell, as I didn't need all the features of a full server or the bother of the license. I started looking for open source options, and found three or four that looked promising. I tested all the packages over the next while and eventually decided on FreeNAS. FreeNAS detected all the hardware properly during the install and worked properly on the first try. I really liked the web interface as it was easy to set up the share for file storage, and to monitor the NAS. I used this for file storage and was very pleased with the stability of it.

A couple years later, I came across a couple of surplused LaCie NAS boxes. They have 1TB of storage using a RAID 0 array of four IDE 250GB drives. They are powered with a VIA C3 800Mhz processor and had 256MB of RAM which I upgraded to 512MB. They were running Windows XP Embedded as the operating system, and had worked well as a Windows file share at work for years before being decommissioned. I figured these would make a good file storage system for home, but I didn't want to use the Windows OS that came with them as it could not easily be patched. FreeNAS was the logical choice of operating system for me to try, although I had reservations as to whether or not it would work with the hardware. To maximize the hard disk space on the NAS, I decided to install FreeNAS to a 4 GB USB key. As the RAM was limited to the 512MB, I went with UFS for the file system, although I would have preferred to use ZFS. I was pleasantly surprised when FreeNAS detected the hardware, installed and ran without issue. I created a Windows (CIFS) share on the NAS and copied my documents there. As RAID 0 gives me no fault tolerance, I set up a second FreeNAS on the other LaCie NAS. I configured RSync to mirror the files from the first NAS to the second on a daily basis.

Since I've set these FreeNAS systems up, they've been solid and dependable. FreeNAS just works like it should, and that's exactly the security I expect for my files. Being able to repurpose lower end hardware into a useful system instead of throwing it out, is an added bonus. That's why I use FreeNAS.

Eric Geissinger

Eric Geissinger lives in London, Ontario, Canada and likes to tinker with BSD and Open Source Software in his spare time.

BSDCAN 2013

THE BEST EVENT OF 2013
<http://www.bsdcn.org/>

Ottawa, Canada



BSDCan 2013 – The event to be at this year

BSDCAN 2013

WHERE

15-16 May – tutorials
17-18 May – conference

WHO

All who are working on and with 4.4BSD based operating systems and related projects.

VENUE

University of Ottawa
<http://www.uottawa.ca/>

High value. Low cost. Something for everyone.

AT FEES YOU CAN AFFORD

We plan to keep to a minimum. As such, the conference will be held at University of Ottawa and accommodation is available within the University residences. Hotels are also within close walking distance of the conference venue.

WHAT DOES IT COST?

Type	CAD
Individual	\$195
Corporate	\$350
Additional Corporate	\$175
Student	\$60
Tutorial (per half day)	\$60
University of Ottawa Staff&Student	\$45

Take the BSDA Certification exam.

For details see

<http://bsdcertification.org/>

- A NetBSD based Tracking Radar
- FreeBSD Kernel Security
- Automating the deployment of FreeBSD & PC-BSD
- Backup and Restore with Bacula
- Benchmarking FreeBSD
- Switching from Linux to FreeBSD
- DNSSec: Troubleshooting and Deployment
- Embedding NetBSD: VOIP applications
- FreeBSD, Capsicum, GELI and ZFS
- FreeBSD Doc Sprint
- FreeBSD storage options
- Hands-on bhyve, the BSD Hypervisor
- Introduction to pkgsrc
- MCLinker BSD
- Making FreeBSD Ports
- Managing FreeBSD at scale
- Modern package management
- Mozilla on OpenBSD
- OpenIKED
- Runtime Process Infection
- The BSD ISP
- Buffer Cache in OpenBSD
- Wireless networking - mobile, gigabit and beyond
- Complexity of checksums in TCP/IP

FreeNAS Detachable Storage has Come of Age

While regular readers of this column will appreciate that I am long in tooth (the corollary being that I am short in memory) – I still remember 8 inch 1.2MB floppy disks that would only hold less than 1MB under CPM.

My first I.T. task before I jumped ship from electronics to computing was diagnosing an elderly CPM based word processing system that had 5.25 inch floppy drives and an equally mature Enhanced Small Disk Interface (ESDI) hard drive. Based on the old ST506/412 interface originally devised by Seagate Technology, in the early 80's ESDI was a common sight for the field engineer. While I cannot remember the exact capacity of the drive, it was only double digit Megabytes – if that. After a time examining the system, it was clear that the hard disk was either encountering a large number of consecutive bad sectors or had suffered a head crash, as the endless “Skip Retry Ignore” prompt testified.

It wasn't until I moved from the electronics environment to I.T. that I encountered the 8 inch drives. Attached to some arcane piece of big iron, these drives were well past their sell by date as demonstrated by the protective layer of dust that gathered around the front bezel and indeed the door handle that was used to secure the floppy disk in place and lower the heads onto the media. Oh, how in these days the demons of bad sectors, corrupt media, and dirt used to haunt the I.T tech – a backup floppy with a year or two of accounting data could be destroyed by a well aimed fingerprint onto the media itself. Saying that, with a scalpel, a sacrificial carcass from a new floppy disk and some industrial alcohol the data could often be recovered. The old school trick of holding the media up against a bright light would allow the technician to see if the drive heads had come in contact with the media, and if no light showed through, a gentle clean with a non-abrasive lint-free tissue cured the problem. While magnetic drop-outs did occur, more often than not physical contamination was the cause of many issues (including dirty drive heads).

This progression and evolution of floppy media continued through 1.44MB and 2.88MB and onwards, until the time came for the hard drive and CDROM drive to become ubiquitous. I remember a fellow engineer proudly demonstrating his new CDROM drive, and commenting on how flawlessly the proprietary software had installed under Windows 3.1. Suddenly, attached media had grown up and the availability of an encyclopedia on the desktop via this shiny platter (which at the time seemed almost indestructible) was revolutionary. Yet the concept of quickly expandable mass online storage hadn't really been adopted in the commercial environment – yes we had dedicated servers, but these only



handled certain network protocols. If you wanted TCP/IP the stack itself was not included in the commercial Operating Systems of the time, so you were strictly limited in flexibility.

While the concept of Network Attached Storage (NAS) has been available for a long time (the original 3com 3server originally shipped in 1985), the idea of a separate network appliance didn't really take off until the late 90's. The market up until this point for storage expansion was dominated by Direct Attached Storage (DAS). Servers were often purchased with plenty of redundant internal and slot space for expansion, and adding an additional SCSI card and drive combination at a later date was fairly trivial. However, this was not always an ideal solution, as the server would need to be taken offline for the installation of the hardware, and even if the server had an external SCSI interface, hot-plug technology was still a long way off in the small / medium sized business sector.

The first truly innovative device that I encountered was the Cobalt Qube, which offered a wide array of function-

ality and user interface options. Initially designed to be fan-less, the Microprocessor without Interlocked Pipeline Stages (MIPS) based architecture only required 20 Watts of power to run, but a fan was added due to customer expectation that a quality device would require one. Running a modified version of Redhat Linux, the device hit the streets at \$1000. Everything was available via either the command line via SSH or Telnet, or a more sophisticated interface was available via the web interface. This is where a NAS based appliance offers great benefits to the network administrator – just plug and go, no formatting required – just copy your data across if required.

The big question in the free software community at the time was why? With a bit of tuning, any Linux / BSD distribution could offer this functionality but the unique selling point of the Qube was the support and the elegant design of both the user interface and the hardware itself. Appealing to a market sector that wanted a “plug and play” solution was critical, yet it was still just a pre-configured box. This is where FreeNAS truly offers the best of both worlds, a downloadable ISO for those that want to use custom hardware and an appliance for those that just want to plug and go.

The ability to plug in a dedicated appliance via a network cable and reach it via CIFS, SMB, NFS, FTP, SSH or RSYNC blows wide open traditional closed commercial environments. This is mass storage for the masses, whether it be the small business, the netizen with a large collection of downloads, or as a backup solution. As FreeNAS is very light, it is ideal for installing on older hardware, and provided the target box has > 6GB of RAM, the security and additional functionality of ZFS is available facilitating snapshots.

The bottom line is this – products like the award winning FreeNAS offer complex functionality driven by proven technology that is available free to anyone that is willing to download and install an ISO. As an appliance, the FreeNAS mini is also available off the shelf from iXsystems. That indeed, is technological evolution in action – power, choice and flexibility.

ROB SOMERVILLE

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid eighties, he has worked in many corporate sectors including finance, automotive, airlines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.



Which FreeNAS?

Lessons From a Year in the Trenches with BSD's Killer App

I will confess that the TCP/IP stack is truly BSD's killer app, giving us the Internet as we know it but that's pretty old news and it's no longer the de facto standard. Other contenders for this status are OpenBSD's OpenSSH and Packet Filter thanks to their reach and occasionally FreeBSD for setting Internet traffic records.

Today however I will argue that the single most valuable piece of BSD software to the greatest number of users is FreeNAS, the open source Network Attached Storage distribution maintained by FreeBSD-oriented hardware vendor iXsystems.

Why? For the simple reason that FreeNAS proves itself equally useful to users of Windows, Mac OS X, free and proprietary Unix, VMWare, XenServer, dlna appliances and just about anyone else who has something to store. We have developed an insatiable need for digital storage and most people probably don't realize how fragile their storage infrastructure is until it's too late. FreeNAS stands out from other Open Source storage appliances by bringing enterprise-class features to within the reach of just about everyone with the key feature being the ZFS "Zettabyte File System" developed by Sun Microsystems. ZFS delivers a level of data integrity protection that had previously only been available in high-end proprietary solutions. While ZFS is not perfect, it is the best all-around Open Source file system available today and of course is never a substitute for a rigorous backup strategy.

Sparing you the details, ZFS performs ongoing data integrity checks that other filesystems do not and it includes institutionalized rollback and replication. By virtue of having a "clean slate" design, ZFS does require some time to wrap your head around but FreeNAS users do not need to learn its syntax unless they choose to. This is a clue to why FreeNAS is a killer app: you can set it up in minutes but it gives you countless layers to

explore down to its underlying FreeBSD/NanoBSD operating system and Django-based web interface. FreeNAS does not yet lend itself to customization but an experienced system administrator can migrate a FreeNAS configuration to a dedicated FreeBSD server without too much trouble. The HUGE advantage of FreeNAS' FreeBSD underpinnings however is the fact that diagnosing a FreeNAS system is largely a matter of diagnosing FreeBSD. With both FreeNAS and FreeBSD getting better with every release and the introduction of the FreeNAS plugins architecture, we are dealing with not just an appliance but a platform with a growing ecosystem.

Hardware Considerations

Like just about any BSD, FreeNAS will boot on just about any hardware but you would be wise to do some careful planning in advance and be willing to invest a little money.

The first two golden rules of FreeNAS should be obvious: invest in the best hard drives and storage controller your budget will allow. The third rule is to invest in at least 1GB of RAM, preferably ECC, per TB of desired storage. While only time will tell which specific hard drives are truly the most reliable, you should avoid the lowest-



end consumer drives. Do your homework on the best drives currently available and note that some drive makers are now offering mid-range drives tuned for NAS use.

Not to make an endorsement, FreeNAS and FreeBSD have long had very good support for LSI controller cards and note that you want “Target” or “JBOD” mode when using ZFS, allowing ZFS to handle all redundancy features. FreeNAS is not a ZFS-only solution and will in fact allow you to create UFS arrays and even share a hardware RAID array. Each approach has certain advantages but ZFS’s portability is a very nice feature. It is by no means universally portable but ZFS is one of the most cross-platform portable file systems available. As for affordable motherboards that support ECC memory, they do exist and you need only do your homework.

The Weakest Link

Once you have FreeNAS-ready hardware, there is one thing you should know about ZFS to avoid shooting yourself in the foot: Your ZFS pool is only as strong as its weakest link. A “pool” is a ZFS storage array and is fundamentally a RAID 0-style “stripe” comprised of one or more “vdevs” or virtual devices. The most common type of vdev is ZFS’s “raidzN” which as its name implies, is like a traditional RAID array with “N” number of failable disks. That is, a raidz1 array can suffer

the loss of one member disk and two for raidz2. While raidz is considered the de facto strategy for ZFS pools, the cumulative nature of the vdev model also means that creating RAID 1+0 striped arrays of mirrors is pretty straight forward. To create a RAID 1+0 ZFS array in FreeNAS, simply configure a mirror of two or more drives and add additional mirrors to it. This will have the performance advantage of no relying on calculated parity for redundancy.

Herein however lies one of the greatest weaknesses of ZFS: it will allow you to add any device, including a USB key as a member vdev and removing it will compromise your entire pool. For this reason you should not think of ZFS as a safe way to cobble all of the hard drives you have lying around.

With these rules in mind, FreeNAS is remarkably forgiving and quick to set up. A Windows CIFS share can be set up in seconds by creating a Volume using the Storage: Volume Manager followed by Sharing: CIFS Shares: Add CIFS Share. Simply enable “Browsable to Network Clients” and “Allow Guest Access” and the resulting share

should be visible by Windows, Mac OS X and BSD/Linux clients. You would be wise to take steps to limit access to the resulting system but this starting point will meet the needs of most SOHO users. From there the documentation is pretty straight forward about setting up more sophisticated shares such as iSCSI targets and NFS shares for use by server operating systems like XenServer, VMware and Microsoft Windows Server. Add in replication and you have an open source storage solution that was unimaginable just a few years ago.

The FreeNAS Platform

It’s no secret that some users were upset when the recent rearchitecture of FreeNAS temporarily removed some home user-oriented features but the remedy is inadvertently taking FreeNAS to a new level. While the developers could have simply replaced the missing features, they opted to build a plugins architecture that is rapidly gaining PC-BSD features such as the PBI package management system and Warden jail management system. These two features will create countless opportunities for FreeNAS as a platform for print, monitoring or database serving in addition to the much-requested multimedia serving. I can’t think of a single Linux distribution or competing NAS solution that begins to embrace this platform-centric approach.

With all this going on, the question of “Which FreeNAS?” is turning from one of which demographic will use it to identifying one of many FreeNAS systems in a given environment. There is simply much more to FreeNAS than ZFS and I have even heard of people putting it in front of proprietary NAS systems to gain missing file sharing protocols. Add in a rich set of third party software daemons through the plugins architecture and the result is an unprecedented serving platform that is useful to every category of user.

MICHAEL DEXTER

Michael Dexter has used BSD Unix systems since 1991 and wrote his first FreeBSD jail management system in 2005. He has sponsored the BSD.lv sysjail and mult multiplicity research projects and took his BSD support public with the formation of BSD Fund in 2007. Michael is now the CTO of the BSD vendor Gainframe and Editor of the BSD technical journal Call For Testing.

What's New in FreeNAS 8.x

This article highlights some of the new features which have been added to FreeNAS 8.x since July, 2012. These include the Plugins Jail, ZFSv28, and GELI encryption.



FreeNAS®

Since its initial release in May, 2011, the newly designed FreeNAS 8.x series has added many features that make this open source storage operating system an attractive option for everyone from home users up to large enterprise users.

The initial releases concentrated on improving the graphical administrative interface and the “core” NAS features. These core features include the ability to perform the following within a graphical interface from a web browser:

- import existing UFS or ZFS RAID configurations
- import existing disks formatted with FAT, NTFS, or EXT2/3
- create volumes, datasets, and zvols
- import existing users, groups, and permissions from Active Directory or LDAP
- create Netatalk, NFS, and Samba shares and manage permissions to those shares
- share data over FTP/FTPS, SSH, and TFTP
- create iSCSI targets
- manage cron jobs, sysctls, and loader.conf values
- manage link aggregations, VLANs, and static routes
- schedule snapshots, replication, scrubs, and S.M.A.R.T. tests
- backup the configuration and perform upgrades

Subsequent releases added the following major features to augment the core NAS features:

- 8.2.0, released in July, 2012, added the Plugins Jail to allow for the installation of additional software.
- 8.3.0, released in October, 2012, added ZFSv28.
- 8.3.1, released in March, 2013, added the ability to create ZFS pools on GELI encrypted disks.

In addition to major features, each release incorporates bug fixes, new drivers, and minor features to improve the usability of FreeNAS. This article discusses some of these new features in more detail.

Plugins Jail

FreeNAS 8.2.0 introduced the Plugins Jail, which allows the FreeNAS administrator to extend core NAS functionality by installing additional applications in order to meet the needs of their specific environment. This functionality is provided through the following components:

FreeBSD Jail

Provides light-weight, operating system-level virtualization. Essentially, it installs a separate FreeBSD system onto the FreeNAS host. The jail has its own hostname, IP address, user accounts, processes, and configura-

tion. The FreeNAS implementation includes vimage, which gives the jail its own networking stack and IP broadcasting, as these are required by some file sharing applications.

PBI

The Push Button Installer format was created by the PC-BSD Project to provide a graphical front-end to the FreeBSD Ports Collection. Applications can be installed and uninstalled from a GUI interface which also provides information about which applications and versions are installed. PBIs are self-contained in that they include all the runtime and library dependencies required by the application.

Plugins

A FreeNAS plugin extends the PBI format by incorporating the installed software, as well as its configuration options, into the FreeNAS GUI. This allows the plugin to be installed, configured, started/stopped, and uninstalled, all from the FreeNAS GUI. Figure 1 shows how the FreeNAS Control Services screen indicates that three plugins have been installed. Figure 2 shows the configuration screen for the Firefly plugin.

In order to install plugins, the Plugins Jail must first be downloaded and installed. If a plugin is not available for the needed software, FreeBSD ports or packages can still be installed within the plugins jail. The only difference is that the installation, configuration, and starting/stopping of the application's service is performed from the command line of the jail, rather than from the FreeNAS GUI.

The Plugins chapter of the FreeNAS Users Guide describes in detail how to install and manage the plugins jail, install and manage plugins, install and manage FreeBSD packages and ports, and how to make custom plugins. This chapter is available at <http://doc.freenas.org/index.php/Plugins>.

ZFSv28

FreeNAS 8.3.0 added support for ZFSv28. This adds the following ZFS features:

RAIDZ3

This triple-parity version of ZFS RAID allows up to three disks to fail, with no restrictions on which drives fail, without losing data.

Replaceable ZIL

The ZFS Intent Log is effectively a filesystem journal that manages writes. You can increase performance by dedicating a device (typically an SSD or a dedicated disk) to hold the ZIL. If the ZIL is installed on a device and that

device fails, it can be replaced without losing the pool. The only data that is lost is the last few seconds of writes which had not yet been committed to the pool.

zpool split

This command allows you to split a disk from a mirrored pool. Essentially, the pool is cloned to the disk which can then be removed and used to recreate that pool on another system.

autoexpand

This ZFS property allows the administrator to replace smaller disks with larger disks in order to increase the size of the pool. While this is not the recommended way to increase pool size, it is the only option when the hardware does not support adding more disks or controllers.

ZLE

Zero Length Encoding is a fast and simple compression algorithm which only compresses blocks that are filled

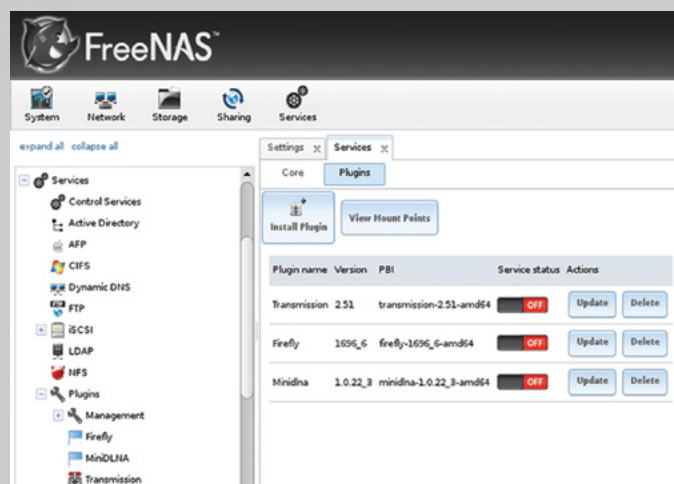


Figure 1. Managing Plugins from Control Services Screen

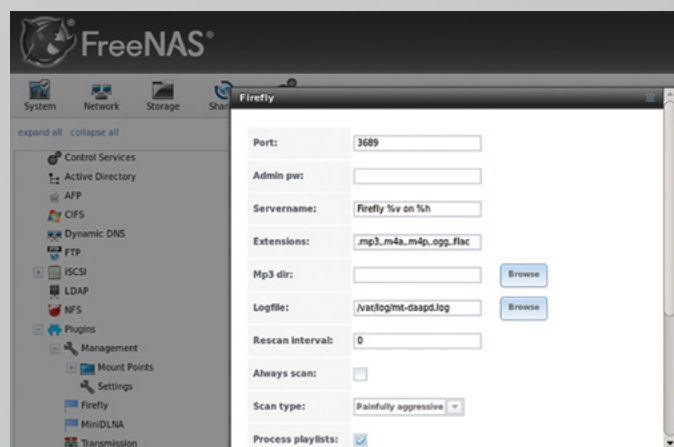


Figure 2. Configuration Screen for Firefly Plugin

with zeroes. This saves space when a thin-provisioned zvol has only used a portion of the size allocated to it.

Deduplication

Is the process of eliminating duplicate copies of data in order to save space. Once deduplication occurs, it can improve ZFS performance as less data is written and stored.

These features, including how to enable them and any caveats to doing so, are described in more detail in the Volumes chapter of the FreeNAS Users Guide: <http://doc.freenas.org/index.php/Volumes>.

Any ZFS volume created in FreeNAS 8.3.0 or later will automatically be formatted with ZFSv28. Existing FreeNAS ZFS pools running ZFSv15 can be easily upgraded using the instructions at http://doc.freenas.org/index.php/Upgrading_FreeNAS#Upgrading_a_ZFS_Pool. Upgrading a pool only takes a few seconds and does not disrupt the use of the FreeNAS system.

Encryption

FreeNAS 8.3.1, released on March 20, 2013, adds FreeBSD GELI disk encryption, allowing a ZFS pool to be created on top of the AES-256 encrypted disks. This type of encryption is primarily targeted at users who store sensitive data and want to retain the ability to remove disks from the pool without having to first wipe the disk's contents.

The design is as follows:

- This is *not* the encryption method used by Oracle ZFSv30. That version of ZFS has not been open sourced and is the property of Oracle.
- This is full disk encryption and *not* per-filesystem encryption. The underlying drives are first encrypted, then the pool is created on top of the encrypted devices.
- This design is suitable for safe disposal of disks independent of the encryption key. As long as the key and the disks are intact, the system is vulnerable to being decrypted. The encryption key should be protect-

ed by a strong passphrase and any backups of the key should be securely stored.

- As a backup recovery method (should the passphrase be forgotten), a recovery key can be used with the encryption key to decrypt the disks.
- The encryption key is per ZFS volume (pool). If you create multiple pools, each pool has its own encryption key.
- If the system has a lot of disks, there will be a performance hit if the CPU does not support AES-NI. If the processor does support the AES-NI instruction set, there should be very little, if any, degradation in performance when using encryption.
- Data in the ZFS ARC cache and the contents of RAM are unencrypted.
- Swap is always encrypted, even on unencrypted volumes.
- There is no way to convert an existing, unencrypted volume. Instead, the data must be backed up, the existing pool must be destroyed, a new encrypted volume must be created, and the backup restored to the new volume.
- Hybrid pools are not supported. In other words, newly created vdevs must match the existing encryption scheme. When extending a volume, FreeNAS will automatically encrypt the new vdev being added to the existing encrypted pool.

When creating an encrypted ZFS volume, an option is available to initialize the disks with random data. This is recommended as it writes the disks with random data before enabling encryption, which can increase its cryptographic strength. However, it will take longer for the volume to be created.

Once an encrypted ZFS volume is created, the user should immediately set a passphrase on the encryption key, make a backup of the encryption key, and create a recovery key. Without these, it will be impossible to re-import or replace the disks at a later time. Figure 3 shows the options for managing the encryption and recovery keys which are added to the FreeNAS GUI for managing the volume. Details on how to use these options can be found at http://doc.freenas.org/index.php/Volumes#Key_Management_for_Encrypted_Volumes.

Miscellaneous Features

Some of the other features introduced since 8.2.0 include:

- a web shell built into the FreeNAS GUI. Clicking this opens a root shell to allow for command line management of the FreeNAS system from a web browser.

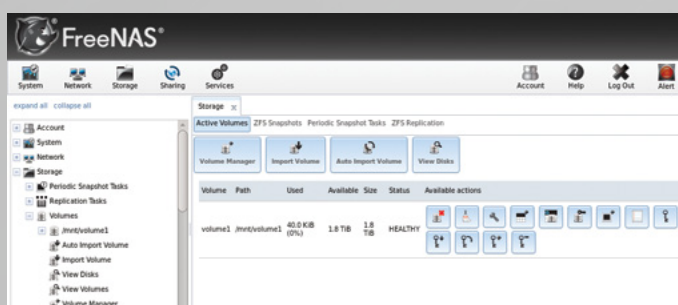


Figure 3. Key Management Options for Encrypted Volume

- support for multipath devices on systems containing dual expander SAS backplanes, SAS drives, or dual expander JBODs with SAS drives. Such hardware will be automatically configured for multipath.
- an autotuning script can be used to set various loader values and sysctls based on system resources and installed hardware components.
- a replication window can be set, allowing snapshots taken during the day to be replicated during the evening.
- improved reporting graphs make it easier to scroll through time intervals to monitor performance trends.
- ZFS ARC stats have been added to top(1).

Additional Resources

Many resources are available to FreeNAS 8.x users. They include:

- the Sourceforge download page: <http://sourceforge.net/projects/freenas/files/FreeNAS-8.3.1/>
- the per-release documentation, in various downloadable formats: http://doc.freenas.org/index.php/Main_Page
- the support page for viewing/creating support tickets and feature requests: <http://support.freenas.org>
- the user forums: <http://forums.freenas.org>
- the Freenode IRC channel: #freenas

DRU LAVIGNE

Dru Lavigne is author of BSD Hacks, The Best of FreeBSD Basics, and The Definitive Guide to PC-BSD. As Director of Community Development for the PC-BSD Project, she leads the documentation team, assists new users, helps to find and fix bugs, and reaches out to the community to discover their needs. She is the former Managing Editor of the Open Source Business Resource, a free monthly publication covering open source and the commercialization of open source assets. She is founder and current Chair of the BSD Certification Group Inc., a non-profit organization with a mission to create the standard for certifying BSD system administrators, and serves on the Board of the FreeBSD Foundation.

The BSD Certification Group Inc. (BSDCG) is a non-profit organization committed to creating and maintaining a global certification standard for system administration on BSD based operating systems.

? WHAT CERTIFICATIONS ARE AVAILABLE?

BSDA: Entry-level certification suited for candidates with a general Unix background and at least six months of experience with BSD systems.

BDSP: Advanced certification for senior system administrators with at least three years of experience on BSD systems. Successful BDSP candidates are able to demonstrate strong to expert skills in BSD Unix system administration.

✓ WHERE CAN I GET CERTIFIED?

We're pleased to announce that after 7 months of negotiations and the work required to make the exam available in a computer based format, that the BSDA exam is now available at several hundred testing centers around the world. Paper based BSDA exams cost \$75 USD. Computer based BSDA exams cost \$150 USD. The price of the BDSP exams are yet to be determined.

Payments are made through our registration website:
<https://register.bsdcertification.org/register/payment>

i WHERE CAN I GET MORE INFORMATION?

More information and links to our mailing lists, LinkedIn groups, and Facebook group are available at our website:
<http://www.bsdcertification.org>

Registration for upcoming exam events is available at our registration website:
<https://register.bsdcertification.org/register/get-a-bsdcg-id>

FreeNAS Plugins

Everything You Ever Wanted to Know

In this article, we will look at the FreeNAS plugin architecture from the perspective of a programmer, how to develop a plugin, and review the transmission plugin.



FreeNAS[®]

What you will learn...

- How the FreeNAS plugin architecture works
- How to make FreeNAS plugins

What you should know...

- Basic understanding of FreeBSD
- Some programming knowledge

When FreeNAS entered the 8.x series, many people were not happy that functionality that previously existed was no longer included. Such functionality was mainly multimedia focused and targeted at the home user. Services such as bittorrent, DLNA and iTunes media services are the primary examples. Beginning with FreeNAS 8.2.0, a plugin architecture was introduced. This architecture allows FreeNAS systems to be extended in any way that the user sees fit. The purpose of this paper is to describe the technical details of how the architecture works so that plugin authors have the knowledge to create new plugins. As a working example, the *transmission* bittorrent client plugin will be reviewed.

FreeNAS is a very powerful open source operating system based on FreeBSD. However, once you get beyond all the great capabilities it offers, your options for extending it become limited. Your choices are using FreeBSD's built in package management system, or modifying the

source code and building your own image. Packages can be installed using FreeBSD's package management system, but care must be taken. You must be aware of what paths and files the package management system uses as well as the package itself. You have to very carefully select what is used and where all the files go because once the system is rebooted, several key files can be overwritten or disappear.

FreeNAS creates memory disks for `/var` and `/etc` at boot time and copies the contents of `/conf/base/var` and `/conf/base/etc` to these file systems. FreeBSD's package tools and ports work with files from `/var/db/pkg` and `/var/db/ports`. Also, the root file system is mounted read only. What this means is that when attempting to install a package, most files won't be allowed to be written to the system and the records in `/var/db` will be erased on boot. This can of course all be circumvented, but the point is that it's an involved process to get working right.

The major problem with using package management is that once you do an upgrade, everything you installed will get wiped out. Upgrades to FreeNAS only save the configuration and the volumes that are created, everything else is wiped clean.

The other option is to hack the build system to include the packages you want. This is certainly an option. The caveat with this is that you must have a FreeBSD system with development tools, an understanding of the build system and how it works, knowledge of what files to edit, and so on. This simply isn't feasible for most people. Most FreeNAS users simply aren't technical enough for this.

To address these problems and more, FreeNAS has introduced the plugin system. The plugin system is modular, self contained and allows everyday users to install programs that fit their needs onto FreeNAS from an easy to use interface. This also allows users to use their FreeNAS system as more than just a file server.

The Plugins Jail

In order to install plugins on FreeNAS, a plugins jail must first be installed, configured and running. A FreeNAS plugins jail is a standard FreeBSD jail packaged as a PBI and pre-configured with several necessary packages that allow the stock plugins to work. The plugins jail can be found in the plugins directory under the FreeNAS release directory that is being used.

To install a plugins jail, you must first upload it. This can be done from the web interface under services->plugins. You must specify where the jail will be stored temporarily when it's uploaded. The next step requires you to configure a jail path, jail name, IP address, netmask, and a plugins archive path. The plugin jail configuration is stored in the database in the table `services_pluginsjail`. The following describes each column and what it is used for:

- `jail_path` – The file system path where the jail resides
- `plugins_path` – The file system path where the plugins reside
- `jail_mac` – MAC address for the jail interface (if configured)
- `jail_ipv4address` – The IPv4 address for the jail
- `jail_ipv4netmask` – The IPv4 netmask for the jail
- `jail_name` – The name of this plugins jail

Currently, Only a single IPv4 address is supported. In the future, multiple Ipv4 and Ipv6 addresses will be supported, as well as multiple plugin jails.

When the plugin jail is uploaded and configured, `pbi_add` is run and the jail is extracted to the `jail_path + jail_name`. Once this is done, the plugins jail is ready to be run. When you turn the service on, `/etc/rc.d/ix-jail` is invoked.

This script generates the proper `/etc/rc.conf` lines to configure the jail with `vnet` and allows `/etc/rc.d/jail` to start the jail. Once the jail is up and running, plugins are ready to be installed.

Installing a Plugin

FreeNAS plugins use the PC-BSD PBI9 format. Plugins are installed using the web interface. Installing a plugin is very easy, navigate to Services->plugins->Install Plugin. When a plugin is installed, the PBI information is stored in the database in the table `plugins_plugins` which has the following columns:

- `plugin_version` – plugin version number
- `plugin_enabled` – enabled/disabled status
- `plugin_ip` – fastcgi server IP address
- `plugin_port` – fastcgi server port
- `plugin_arch` – i386 or amd64
- `plugin_api_version` – RPC API version
- `plugin_name` – name of the plugin
- `plugin_pbi_name` – PBI file name as uploaded
- `plugin_path` – where in the file system the plugin is installed

Once the PBI information is saved, an OAuth secret and key are generated record in the `services_rpctoken` table. This table contains the columns:

- `secret` – the OAuth secret
- `key` – the OAuth key

Once the PBI and OAuth information is recorded in the database, the following steps occur:

- The PBI is installed into the plugins jail in `/usr/pbi/${plugin}-${arch}/`
- The OAuth key and secret are written into `/usr/pbi/${plugin}-${arch}/oauth`
- The plugin information is written into `plugins.conf` which is included by `nginx.conf`. This tells `nginx` that all URL's that specify the plugin path are to be passed to the plugins fastcgi server.
- The plugins control script is started in the jail (`/usr/pbi/${plugin}-${arch}/control start`). This starts the plugin fastcgi server on the IP/port combination recorded in the database.

- The web interface will refresh. The navtree makes a request to the plugin's `_s/treemenu` and `treemenu-icon` methods. The `treemenu` method returns a description of how to display the plugin information in the navtree. The `treemenu-icon` method passes the icon for the plugin to the navtree. Once these methods are called, the plugin appears in the navtree menu under `Services->plugins->${plugin}` with the plugin icon. The plugin will also appear under the `Services->plugins` menu in the main interface.

How they Work

When the plugin icon is clicked, `django` matches the plugin URL and sends the request to the plugin fastcgi server. Requesting a plugin method is of the form:

```
base_url + "/plugins/" + ${plugin} + "/" + ${method}
```

The methods that are available are:

- `edit` – edit the plugin configuration
- `treemenu-icon` – icon to be displayed in the navtree
- `_s/treemenu` – what/how to display in the navtree
- `_s/start` – start the plugin
- `_s/stop` – stop the plugin
- `_s/status` – status of the plugin

Plugins have access to the base system via RPC calls. All RPC requests are signed with the `oauth` credentials given to the plugin at install time. The following RPC methods are available:

- `api.version()` – get the plugin API version
- `plugins.plugins.get()` – get a listing of installed plugins
- `plugins.jail_info()` – get information about the plugins jail
- `plugins.is_authenticated()` – test if the plugin is currently authenticated
- `fs.mountpoints.get()` – get a listing of available file systems
- `fs.mounted.get()` – get a list of mounted file systems
- `fs.mount()` – mount a file system into the jail
- `fs.umount()` – unmount a jailed file system
- `fs.directory.get()` – get a directory listing
- `fs.file.get()` – get a file
- `os.arch()` – get OS architecture
- `api.test()` – verify RPC calls are working

When an RPC request to the base system takes place, the following things happen:

- An RPC request is built of the form: `base_url + "/plugins/json-rpc/v1"`
- The RPC request is signed with the `oauth` credentials
- The RPC request is sent with the requested method
- The method is invoked if the `oauth` credentials are correct and the method exists. The results are then returned back to the plugin

The fastcgi server accepts the plugin request, then dispatches accordingly. This allows anything that can communicate the fastcgi protocol to be a plugin, or even to manage plugins. Because of this flexibility, plugins can be developed using any language or framework one wishes to use. All that is required for a FreeNAS plugin to work is that it implement the described methods and be packaged using the PBI9 format.

Making a Plugin

Currently, making a plugin for FreeNAS is somewhat cumbersome. This process is expected to be streamlined in coming releases. While there are several methods to create a plugin, the one described was used to develop the 3 reference plugins included on Sourceforge.

Documentation for creating PBI files using the PBI9 format already exists, so only the FreeNAS specific portions will be covered. Creating a PBI for FreeNAS requires FreeBSD 8.x, PC-BSD 8.x, or FreeNAS 8.2.0 or higher. In all cases, `pbi-manager` and the ports collection must be installed. The basic procedure for creating a plugin is this:

- Create plugin directory: `myplugin`
- Create resource and scripts directories under this directory: `myplugin/resources` and `myplugin/scripts`
- Create a PBI configuration file: `myplugin/pbi.conf`
- Edit the `pbi.conf` file for your particular plugin. Documentation for how to do this can be found at wiki.pcbbsd.org "PBI Module Builder Guide". It's relatively straight forward.
- If there are any pre/post script needs, create the necessary scripts in the scripts directory as specified in the PBI module builder guide.
- Invoke `pbi_makeport` to create the PBI

At this point, a PBI will have been created. Upload the PBI as previously described and it will be installed into the plugins jail. It will not be functional from within the web interface, but it is ready to be worked on from within the jail. This process can be repeated as the plugin is refined and developed.

Next, a control script must be created. The name of the script must be *control* and it must be located in the plugin directory `/usr/pbi/${plugin}-${arch}/`. The control script takes 3 arguments, an action verb, an IP address, and a port. The purpose of the script is to start a fastcgi server on the specified IP address and port.

The verbs that must be implemented are start, stop and status. The start verb starts the fastcgi server on the IP/port combination. The stop verb stops the server. The status verb exits with 0 if the server is running otherwise it exits with 1.

This script is called from the main system when the plugin is enabled or disabled.

Once the control script is completed, the interface portion of the plugin can be worked on. The job of the interface is to export the methods needed by FreeNAS to integrate with the web interface as described in section 4.

The start and stop methods must provide a means by which to start and stop the binary the plugin is in control of. This also includes any modifications to `/etc/rc.conf` if necessary. The `treemenu` method simply dumps out JSON. The `treemenu-icon` outputs the plugin icon. The workhorse of a FreeNAS plugin is the `edit` method. This is the method that presents the interface for configuring the plugin. This generally entails saving and restoring state and generating and modifying configuration files.

An Example – Transmission

When FreeNAS released 8.2.0, three reference plugins were also released. They were provided for two reasons: to provide the missing functionality that previously existed in FreeNAS 7.x, and to document and demonstrate how future plugins could be made.

One of the available plugins is *Transmission*. *Transmission* is a very popular bittorrent client. It's implementation is pretty simple and straight forward so it will be used for the example. Since *Transmission* is built into the build system, the build system configuration will be covered as well. Reviewing the build system process for making a plugin is recommended anyhow for plugin authors so they have a better understanding of how everything works. Here is an overview of the directory layout and key files for the *transmission* plugin:

```
`${freenas}/nanobsd/plugins/
```

This is top level directory for all FreeNAS plugins. All plugin files are located in this directory. The build system

will be aware of a plugin once it is placed in this directory. For *Transmission*, the following file is created:

```
`${pluginroot}/transmission
```

This is the NanoBSD configuration file for *Transmission*. It sets up the NanoBSD environment for the *Transmission* build and provides function(s) for doing so. Since NanoBSD is being used for the plugin build, a bit of trickery is done here. All of the NanoBSD functions are overridden with stub calls except the `last_orders()` function. This is the function that makes the actual call to *pbi_makeport* and does the plugin build.

```
`${pluginroot}/transmission_pbi/
```

This is the *Transmission* PBI directory. All plugins must have a PBI directory. Within this directory, two subdirectories must exist: `scripts/` and `resources/`. A *pbi.conf* file must also exist.

```
`${transmission_pbi}/pbi.conf
```

This file tells *pbi-manager* how to build *Transmission*. It contains information about the plugin such as the port(s) to be built, the icon(s) to be used, the make options for the binaries, etc.

```
`${transmission_pbi}/resources/
```

This directory contains the bulk of the plugin interface. It can be structured however the plugin author chooses. Since *Transmission* uses *django*, the *django* application resides in the directory along with an assortment of other scripts and programs.

```
`${transmission_pbi}/resources/control.py
```

This is the *transmission* fastcgi server control program. As discussed previously, this program has three responsibilities: to start the fastcgi server, stop the fastcgi server and report the status of the fastcgi server. The start and stop methods also start and stop the *django* web server application. The *django* application exports all the required hooks that FreeNAS requires to interface with the plugin. A wrapper script that calls *control.py* is also in this directory. This “control” wrapper is the only mandatory file that needs to be known by the base system.

```
`${transmission_pbi}/transmission
```


This is the RC script for *transmission* that controls the daemon. It's just like any other RC script that FreeNAS uses.

```
`${transmission_pbi}/resources/tweak-rcconf
```

The job of this script is to modify */etc/rc.conf* to enable or disable *transmission*.

```
`${transmission_pbi}/scripts/
```

This directory contains hooks for different stages of the PBI build process and installation process. The possible scripts are `pre-install.sh`, `post-install.sh`, `pre-portmake.sh`, `post-portmake.sh` and `pre-remove.sh`. `Pre-install.sh` allows you to do customizations to the system prior to the plugin being installed, such as adding users and groups. `Post-install.sh` is run immediately after the plugin is installed. Some typical post install operations are database initialization and migrations. `Pre-portmake.sh` and `post-portmake.sh` allow you to do operations before and after port compile. `Pre-remove.sh` is run prior to plugin removal. Operations typically done by `pre-remove.sh` are user and group removal.

The other files in ``${transmission_pbi}/resources` are *default.png*, *freenas* and *transmissionUI*. *Default.png* is the default icon for the PBI. *Freenas* is a file that contains the plugin api version. *TransmissionUI* is the django application.

```
`${transmission_ui}/freenas/
```

This is the *django* application. Under this directory are the typical *django* model, form, view and url files. In the `urls.py` file, the exported methods that FreeNAS requires are very visible and demonstrate what needs to be made available for a FreeNAS plugin to be functional.

To build the *transmission* plugin, run the command:

```
`${freenas}/build/do_build.sh -t plugins/transmission.
```

This will first create a ``${freenas}/sbin` directory and install `pbi-manager` into it. When `pbi_makeport` is invoked, it will compile a FreeBSD 8.x world and install it into a temporary directory which will later be tarred up and saved for future compilations. Once a world directory is ready, the ports that are needed to compile the plugin get compiled and installed. Any provided scripts get ran and then the PBI is made and placed in ``${freenas}/${plugin}/${arch}/${plugin}.pbi`. The plugin build is complete at this point.

On the Web

- <http://www.freenas.org/> – Download FreeNAS
- http://wiki.freenas.org/index.php/Main_Page – FreeNAS Documentation

To test and verify *transmission* works, upload the *Transmission* plugin through the FreeNAS web interface as previously described. Navigate to Services->plugins->Transmission from the navtree. Click on it and there should be an edit screen. There are default values filled in already but these can be modified and saved. Save the configuration and go to Services->plugins->Transmission from the main interface and turn the slider to on. At this point you can grab any torrent file, place it in the directory specified in the watch directory and watch it get downloaded in the download directory. Success!

Conclusion

FreeNAS plugins allow FreeNAS to be extended in ways anyone sees fit. They are very powerful in their flexibility and allow plugin authors to make a FreeNAS system into everything from a multimedia server to a print server. The purpose of this paper is to explain the plugin architecture so that more developers and people knowledgeable enough can make more plugins. Happy hacking!

JOHN HIXSON

John Hixson is a BSD geek that resides in California. He has been using Open Source Software and operating systems since the mid 90's. He has worked as both an engineer and systems administrator during the source of his technical career. His employment with iXsystems keeps him busy with FreeBSD, PC-BSD and FreeNAS. John enjoys working on anything that is challenging and requires learning new things. In his spare time he likes to read, cook, spend time with his family, hack on code and play with new geek toys.

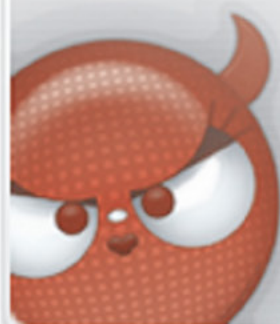
Starting from

8.95\$
~~8.22\$~~

FreeBSD
OpenBSD
NetBSD

VPS HOSTING

- ✓ **UNMETERED** bandwidth
- ✓ **VNC** console
- ✓ **instant** reimaging
- ✓ **native IPv6** network
- ✓ **always latest BSDs**
- ✓ **competent support**



<http://bsdvm.com>

Handmade NAS

In the past, the term Network Attached Storage (NAS) used to be associated with large, expensive networks and luxurious, enterprise-grade file servers. At the time, who would have thought that hard disk and RAM would become a common commodity that regular folks like us would possess?

What you will learn...

- Basic file sharing with SMB, AFP, and FTP server
- Use these as building blocks for a full-fledged NAS

What you should know...

- FreeBSD installation
- Network configuration
- Port installation and configuration

With the well-known FreeNAS distribution, building a multi-terabyte NAS seems to be a trivial affair. But today, we're going to talk about how to build a NAS by hand, plugging different components together using the command line and FreeBSD. This article does not aim to be a comprehensive guide to building a NAS. Rather, it serves as a jumpstart guide on the basic components that make up a NAS, which the reader can expand on and enrich with more features.

The components of the NAS we're going to build include:

- FreeBSD server – setup with ZFS as root filesystem using mirroring configuration
- Samba – file server targeting Windows client
- AFP through Netatalk – file server (with Time Machine) targeting Macs
- FTP server – generic file sharing server
- pf – packet filtering as firewall

Scenario requirements for this guide:

- modern Intel / AMD processor
- at least 2GB of RAM
- at least 2 hard disks
- megabit NIC or better

Disclaimer

This guide is a walkthrough on setting up a NAS and it WILL destroy your existing data on the hard disk. You have been warned.

FreeBSD Server Setup

As of this writing, the FreeBSD default installer does not support ZFS filesystem layout. As such, we'll have to hand craft the ZFS layout during the installation of FreeBSD 9.1 server. This will allow us to learn more about the underlying workings of FreeBSD root on ZFS filesystem.

First of all, get the installation CD from <http://freebsd.org> and boot it. Perform the installation as usual. Under the menu "Partitioning", choose "Shell".

After dropping to the shell, execute the following commands: Listing 1.

Type `exit` at the command prompt once the above is done. The installation should continue, extracting the necessary files to the hard disk. What the above does is create a GPT partition table in the very first step. Then follow this up by creating a boot partition of 128 kilobytes and a swap partition of 4 gigabytes (based on 2GB RAM). The rest of the hard disk space is allocated to the data area, root partition.

After the partitioning of the hard disk is done, initialize the boot area with boot code found in `/boot/gptzfsboot`.

The same disk layout is duplicated to the second hard disk in order to create a ZFS mirroring disk setup.



Next, create a ZFS pool with the name of “zeetank”, consisting of the partitions “primary-root” and “secondary-root” with ZFS mirroring. Then create a mount point to hold the root partitions and set the boot partitions to find “zeetank/root” for necessary booting files. Finally, list the disk layout we just did and double check it.

If a typo or a misconfiguration of the disk layout occurs, use the following commands for disk partition to delete and/or destroy:

```
# gpart delete -i 1 da0
# gpart destroy da0
```

The completed disk layout should look like this: Listing 2 and Listing 3.

For more disk redundancy setup on ZFS, you should refer to the FreeBSD Handbook. The ZFS Administration Guide from the Solaris Project can also serve as a good reference (see Reference for links).

The installation should resume after “exit”. Proceed as you would install FreeBSD normally. For example, set your root password, configure the network interface, set up the time zone, and etc. When the menu “Manual Configuration” appears, choose “No”. On the next menu, “Complete”, choose “Live CD” instead, because there are still a couple of files we need to set up. Do take note that the root partitions that we created earlier, are mounted on `/mnt`.

Log in as root (it should not prompt for password in “Live CD” mode) and execute the code from Listing 4.

The lines with “echo” are the necessary ZFS startup parameters FreeBSD needs to know. Next, copy the ZFS cache file onto the mounted file system in order for ZFS to boot properly: see Listing 5. List the ZFS properties to make a visual check. Reboot when finished:

```
# zpool get all zeetank
# reboot
```

After reboot, log into the box and set yourself as root. Fetch and extract the ports tree as necessary.

Then create a user named “bob”, so that we can log into the file sharing services that we’re going to set up. Since this user is just a common user accessing files through various services, we should prevent it from logging into the system through any shell. This will be covered in the section setting up FTP server later. For now, create the user:

```
# pw useradd -mn bob
# passwd bob
```

**BSD development
and consultancy**

Zabbix Monitoring

**Bacula enterprise
backup**

BSD Thin Client

**Corporate BSD
Desktop**

**Solution
management
with Puppet**

and more ...

www.mtier.org
contact@mtier.org

Listing 1. FreeBSD root on ZFS

```
# gpart create -s gpt da0
# gpart add -t freebsd-boot -s 128 -l primary-boot da0
# gpart add -t freebsd-swap -l primary-swap -s 4g da0
# gpart add -t freebsd-zfs -l primary-root da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i
    1 da0

# gpart create -s gpt da1
# gpart add -t freebsd-boot -s 128 -l secondary-boot da1
# gpart add -t freebsd-swap -l secondary-swap -s 4g da1
# gpart add -t freebsd-zfs -l secondary-root da1
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i
    1 da1

# zpool create -f -m none -o altroot=/mnt -o
    cachefile=/tmp/zpool.cache zeetank
    mirror gpt/primary-root gpt/
    secondary-root
# zfs create -o mountpoint=/ zeetank/root
# zpool set bootfs=zeetank/root zeetank

# zfs list
# zpool get all zeetank
```

Listing 2. Sample of partition layout

```
root@handmade-nas:/root # gpart show
=>          34  20971453  da0  GPT          (10G)
           34      128      1  freebsd-boot (64k)
           162  1048576      2  freebsd-swap (512M)
          1048738 19922749      3  freebsd-zfs  (9.5G)

=>          34  20971453  da1  GPT          (10G)
           34      128      1  freebsd-boot (64k)
           162  1048576      2  freebsd-swap (512M)
          1048738 19922749      3  freebsd-zfs  (9.5G)
```

Listing 3. Sample of ZFS setup

```
root@handmade-nas:/root # zpool get all zeetank
NAME      PROPERTY          VALUE          SOURCE
zeetank   size              9.44G         -
zeetank   capacity          0%            -
zeetank   altroot           /mnt          local
zeetank   health            ONLINE        -
zeetank   guid              7047189317064599296 default
zeetank   version           28            default
zeetank   bootfs            zeetank/root  local
zeetank   delegation        on            default
zeetank   autoreplace       off           default
zeetank   cachefile         /tmp/zpool.cache local
zeetank   failmode          wait          default
zeetank   listsnapshots     off           default
zeetank   autoexpand        off           default
zeetank   dedupditto        0             default
zeetank   dedupratio        1.00x         -
zeetank   free              9.44G         -
zeetank   allocated         152K          -
zeetank   readonly          off           -
zeetank   comment           -             default
zeetank   expandsize        0             -
```

Samba File Server

The Samba server that we're going to configure is meant for Windows client file sharing. First, install the port "net/samba36". For example:

```
# make -C /usr/ports/net/samba36 install clean
```

The Samba authentication we are going to use stores information in a trivial database. The password files holding user credentials are *passwd.tdb* and *secrets.tdb*, which are located in `/usr/local/etc/samba/`. Edit the file `/usr/local/etc/smb.conf` with the contents showed in Listing 6.

Add a user to the Samba trivial database:

```
# pdbedit -au bob
```

Next, start Samba services:

```
# echo `samba_enable="YES"` >> /etc/rc.conf
# service samba start
```

If all is well, the Windows client will be able to access the share named "bob" on this box.

AFP Through Netatalk

Macs can share files and back up using Time Machine with *Apple Filing Protocol (AFP)* through the Netatalk file sharing service.

Before starting, install the port "net/netatalk" and uncheck the option "ZEROCONF" unless you want to use the Bonjour function through zeroconf. There will be a few files we will need to create/edit:

- `/usr/local/etc/AppleVolumes.default`
- `/usr/local/etc/afpd.conf`
- `/usr/local/etc/netatalk.conf`
- `/etc/rc.conf`

Create the file `/usr/local/etc/AppleVolumes.default` with the following contents:

```
:DEFAULT: options:upriv,usedots
/home/bob "bob's home directory" allow:bob options:tm
```

The file `AppleVolumes.default` tells Netatalk that the share `/home/bob` can only be shared with user "bob" and that the "Time Machine" function is available with the share.

Next, create the file `/usr/local/etc/afpd.conf` with these contents, which is only a SINGLE line:

```
- -ipaddr 0.0.0.0
```

This basically tells `afpd` to listen for incoming requests in all network interfaces. Next, create `/usr/local/etc/netatalk.conf` with the below contents:

```
ATALK_NAME=NAS_box_afp
```

The settings in `netatalk.conf` should be pretty self-explanatory to set up the hostname for Netatalk service.

Finally, set rc parameters in `/etc/rc.conf`:

```
# echo `afpd_enable="YES"` >> /etc/rc.conf
# echo `atalkd_enable="NO"` >> /etc/rc.conf
# echo `cnid_metad_enable="YES"` >> /etc/rc.conf
# echo `netatalk_enable="YES"` >> /etc/rc.conf
```

That's all for the Netatalk service. Start the service by:

```
# service netatalk start
```

Remember to check `/var/log/messages` for error messages. The Mac clients should be able to browse this server for AppleTalk shares and use this share for

Listing 4. ZFS rc script values & file system parameters

```
# echo `zfs_enable="YES"` >> /mnt/etc/rc.conf
# echo `zfs_load="YES"` >> /mnt/boot/loader.conf
# echo `vfs.root.mountfrom="zfs:zeetank/root"` >> /mnt/boot/loader.conf
# echo `/dev/da0p2 none swap sw 0 0' >> /mnt/etc/fstab
# echo `/dev/da1p2 none swap sw 0 0' >> /mnt/etc/fstab
```

Listing 5. ZFS cache

```
# zpool export zeetank
# zpool import -o altroot=/mnt -o cachefile=/tmp/zpool.cache zeetank
# cp /tmp/zpool.cache /mnt/boot/zfs/
```

Listing 6. smb.conf

```
[global]
    workgroup = Private
    netbios name = NAS_box_smb
    security = user
    encrypt passwords = yes
    client lanman auth = no
    log file = /var/log/samba/log.%m
    passdb backend = tdbsam
    load printers = no
    printcap name = /dev/null

[bob]
    path = /home/bob
    browseable = no
    writeable = yes
    valid users = bob
    admin users = bob
```

Listing 7. ftpd chroot

```
# cp /sbin/nologin /usr/local/bin/ftp-login-only
# echo "/usr/local/bin/ftp-login-only" >> /etc/shells
# mkdir /home/bob/ftpdire
# pw usermod -n bob -d "/home/bob/ftpdire/." -s "/usr/local/bin/ftp-login-only"
# echo "bob" >> /etc/ftpchroot
```


Time Machine backups if there are no error messages logged.

FTP (File Transfer Protocol)

The FTP server we are going to configure will be used for general file uploading and sharing. We will be using the FTP daemon that comes with the FreeBSD base installation, `ftpd`. We'll also restrict `ftp` user login to its home directory, effectively `chroot`-ing the user.

During the creation of user “bob” earlier, the default shell was assigned as `/bin/sh`, which allows the user to login using a shell. In order to restrict the user login to services we have configured (Samba, Netatalk, and FTP), we should disable its login capabilities. For this purpose, do the following: Listing 7.

The above will duplicate a copy of the `nologin` shell from base and list it in `/etc/shells` so that `ftpd` will recognize it. Take note of the separator `“/./”` as this will tell `ftpd` where to set the “root” directory structure of the user and effectively prevent the user from going beyond `/home/bob/ftplib`. Next, start the FTP daemon with:

```
# echo `ftpd_enable="YES" >> /etc/rc.conf
# service ftpd start
```

The FTP daemon will log its messages to `/var/log/xferlog` by default.

Packet Filter – pf

`pf` is an OpenBSD firewall that has been ported to the base system of FreeBSD since version 5.x. It is well known for its features, performance, and simple syntax. For these reasons, we will use it to add another layer of security to the ser-

Listing 8. `/etc/pf.conf`

```
services_tcp="{ 21, 22, 139, 445, 548 }"
services_udp="{ 137, 138 }"
clients_ip="{ 192.168.0.10, 192.168.0.11 }"

block in all
block out all

pass quick on lo0 all
pass in inet proto icmp all icmp-type echoreq

pass in proto tcp from $clients_ip to port $services_tcp
pass in proto udp from $clients_ip to port $services_udp
# for ftpd to work properly
pass in proto tcp from $clients_ip to port > 49151
```

References

- <http://www.freebsd.org/doc/handbook/index.html>
- <http://docs.oracle.com/cd/E19082-01/817-2271/817-2271.pdf>
- <https://wiki.freebsd.org/RootOnZFS>
- http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide
- <http://www.samba.org/>
- <http://netatalk.sourceforge.net/>
- <http://www.openbsd.org/faq/pf/>
- <http://errata.psybermonkey.net/bsdmag/handmade-nas>

vices configured above. Put the following into `/etc/pf.conf` (Listing 8). The `pf` rules above allow the IP in `$clients_ip` to access the services listed in `$services_tcp` and `$services_udp`. It also allows pings from all IPs for troubleshooting purposes. Loopback interface (`lo0`) checks will be skipped and on default will “block” both inbound and outbound traffic if no rules are matched. Now, you can start up packet firewall:

```
# echo `pf_enable="YES" >> /etc/rc.conf
# service pf start
```

Summary

This wraps up the guide on how to set up a NAS by hand and configure the services. The configurations shown above are the bare minimum and the purpose of this guide is to help someone interested in setting up a NAS by poking around the system. More reading should be done and care should be taken in securing the services. Of course, the configurations in this article can certainly be altered to provide more features. For example, Samba and Netatalk users can be authenticated with LDAP in the backend. The feature-rich ZFS filesystem has barely been touched – just to mention a few: various disk redundancy (mirror, striping with parity) configurations, file compression, and filesystem snapshot. The FTP daemon is capable of virtual-host-like hosting and `pf` is a full-fledged, feature-rich firewall. There is so much to talk about. I hope this guide serves a good purpose and provides a picture of how a NAS can be built. Below are some links for reference. Have fun.

EDWARD TAN

Edward Tan's day-to-day job is administrating a bunch of servers running on FreeBSD. In his free time, he blogs about techie stuff at <http://psybermonkey.net>, learns about Perl and thinks about how to contribute back to the FreeBSD community.

Great Specials

On FreeBSD® & PC-BSD® Merchandise

Give us a call & ask about our
SOFTWARE BUNDLES

1.925.240.6652

\$39.95

FreeBSD 9.1 Jewel Case CD Set
or FreeBSD 9.1 DVD

\$29.95

PC-BSD 9.1 DVD

\$49.95

The PC-BSD 9.0 Users Handbook
PC-BSD 9.1 DVD



\$99.95

The FreeBSD CD or DVD Bundle

Inside each CD/DVD Bundle, you'll find:
FreeBSD Handbook, 3rd Edition
Users Guide FreeBSD Handbook, 3rd Edition, Admin Guide
FreeBSD 9.1 CD or DVD set
FreeBSD Toolkit DVD

Stylish Dress Attire
Look Your Professional Best



Comfy Apparel
Stay Warm in Zip Ups & Pullovers

T-Shirts
Lots of Styles to Choose From

FreeBSD 9.1 Jewel Case CD/DVD.....\$39.95

CD Set Contains:

- Disc 1** Installation Boot LiveCD (i386)
- Disc 2** Essential Packages Xorg (i386)
- Disc 3** Essential Packages, GNOME2 (i386)
- Disc 4** Essential Packages (i386)

FreeBSD 9.0 CD.....\$39.95

FreeBSD 9.0 DVD.....\$39.95

FreeBSD Subscriptions

Save time and \$\$\$ by subscribing to regular updates of FreeBSD

FreeBSD Subscription, start with CD 9.1.....\$29.95

FreeBSD Subscription, start with DVD 9.1.....\$29.95

FreeBSD Subscription, start with CD 9.0.....\$29.95

FreeBSD Subscription, start with DVD 9.0.....\$29.95

PC-BSD 9.1 DVD (Isotope Edition)

PC-BSD 9.1 DVD.....\$29.95

PC-BSD Subscription.....\$19.95

The FreeBSD Handbook

The FreeBSD Handbook, Volume 1 (User Guide).....\$39.95

The FreeBSD Handbook, Volume 2 (Admin Guide).....\$39.95

The FreeBSD Handbook Specials

The FreeBSD Handbook, Volume 2 (Both Volumes).....\$59.95

The FreeBSD Handbook, Both Volumes & FreeBSD 9.1.....\$79.95

PC-BSD 9.0 Users Handbook.....\$24.95

BSD Magazine.....\$11.99

The FreeBSD Toolkit DVD.....\$39.95

FreeBSD Mousepad.....\$10.00

FreeBSD & PCBSD Caps.....\$20.00

BSD Daemon Horns.....\$2.00



Bundle Specials!
Save \$\$\$

Just Plain Fun
Mousepads & Novelty Horns



BSD Magazine
Available Monthly



For even MORE items
visit our website today!

www.FreeBSDMall.com

FreeNAS

in an Enterprise Environment

By the time you're reading this, FreeNAS has been downloaded more than 5.5 million times. For home users, it's become an indispensable part of their daily lives, akin to the DVR. Meanwhile, all over the world, thousands of businesses, universities, and government departments use FreeNAS to build effective storage solutions in myriad applications.

What you will learn...

- How TrueNAS builds off the strong points of the FreeBSD and FreeNAS operating systems
- How TrueNAS meets modern storage challenges for enterprises

What you should know...

- Basic understanding of server technology
- Some networking knowledge
- Some knowledge of BSD Operating Systems

The FreeNAS operating system is freely available to the public and offers thorough documentation, an active community, and a feature-rich platform for the storage environment. Based on FreeBSD, FreeNAS can share over a host of protocols (SMB, CIFS, AFP, NFS, FTP, iSCSI, etc) and features an intuitive Web Interface, the ZFS file system, a plug-in system for applications, and much more.

Despite the massive popularity of FreeNAS, most people aren't aware of its big brother dutifully storing and protecting data in some of the most demanding enterprise business environments: the proven, enterprise-ready, and professionally-supported line of appliances known as TrueNAS.

But what makes TrueNAS different from FreeNAS? Well, I'm glad you asked...

Commercial Grade Support

When a mission critical storage system goes down, an organization's whole operation can come to a screeching halt. While community-based support is wonderful (and free), it can't always get an ailing storage server back up and running in a timely manner. Businesses need the responsiveness and expertise that only a professional, dedicated support team can provide. TrueNAS storage appliances provide that safety net.

Created by the same technical team at iXsystems that developed FreeNAS v8 for the home user, TrueNAS is

intended to be a "FreeNAS" for the enterprise business, complete with around-the-clock support from that very same team. For customers, a support team with direct insight into the base operating system is crucial, saving wasted hours spent investigating possible known issues or configuration missteps. With decades of combined experience in both FreeBSD and enterprise storage software, the TrueNAS Support Team at iXsystems know the

The screenshot displays the FreeNAS WebGUI interface. At the top, there's a navigation bar with icons for System, Network, Storage, Sharing, and Services. Below this is a sidebar menu with expand/collapse options. The main content area shows the 'System Information' page, which includes a table of system details:

Hostname	freenas.local
Build	FreeNAS-8.3.1-RELEASE-x64 (r13421)
Platform	Intel(R) Core(TM) i3-2100T CPU @ 2.50GHz
Memory	8086MB
System Time	Mon Mar 11 09:36:34 PDT 2013
Uptime	9:36AM up 9 mins, 0 users
Load Average	0.10, 0.15, 0.12
Connected through	192.168.1.70

Figure 1. System Information page of the FreeNAS WebGUI

ins and outs of configuring storage in virtually any environment, while also possessing intimate knowledge of the underlying operating system.

The production environment is the heartbeat of every organization, designed to serve customers and employees around the clock. TrueNAS appliances help organizations achieve this goal. With software support available 24x7, service models with guaranteed response times, and a web-based support system, TrueNAS covers all the bases. But the advantages don't stop there.

Enterprise Features

FreeNAS is designed as a storage solution for any environment and is used in a wide variety of institutions and organizations. For the enterprise user though, not having experienced UNIX or FreeBSD administrators on hand can provide some challenges. TrueNAS does the work to provide additional enterprise features in a qualified storage appliance:

- High availability (active/passive failover)
- Automatic read cache management and protected write log
- Robust, purpose-built hardware platform
- Automated hard drive removal/replacement & hot spare capability
- Integrated remote management and monitoring
- Tuned for 10Gb Ethernet performance (up to 40GbE)
- ZFS tuning for enhanced storage performance
- Mirrored boot device support
- /swap on flash-based boot device instead of on pool components
- Integrated support module
- Enhanced graphical reporting module
- Qualified for VMWare, Citrix Xen, and Hyper-V

With its purpose-built hardware platform, TrueNAS leverages enclosure management through the integration of *SCSI Enclosure Services* (SES), which allows a technician to quickly and easily identify which drive in the hardware enclosure is failing. Having a hard drive fail without this can be the source of maintenance confusion and frustration. While integrating this feature into FreeNAS isn't easy due to the need for multi-platform compatibility, TrueNAS makes swapping hard drives a straightforward operation.

One serious point of failure in standard storage configurations is if a head unit goes down, all of the storage becomes inaccessible. This can result in serious disruptions of day-to-day operations, costing time and money. TrueNAS offers protection through its high

availability feature with active/passive failover, giving organizations a way of avoiding downtime with dual-node head units.

Along with the greater fault-tolerance and robust hardware platform, TrueNAS appliances bring several other features and performance enhancements to the enterprise data center: built-in remote management, storage tiering through integrated high-performance flash-based read and write cache, mirrored boot devices, NFS support, as well as 10GbE and ZFS performance enhancements. TrueNAS also comes with all the features FreeNAS already provides and is based on the FreeBSD operating system, which is renowned for its stability and security.

Secure, Robust Storage with the ZFS

When it comes to modern file systems, there is none more flexible or powerful than ZFS, the file system at the heart of both TrueNAS and FreeNAS. Originally developed by Sun Microsystems, ZFS was forged around the major tenets of storage: data integrity, security, flexibility, and scalability. With multiple data integrity checks, ZFS is structured so that each block of data has independent checksums. The filesystem utilizes this *Copy on Write* (CoW) transactional model to organize these blocks in a tree system where each block contains its "children's" checksums, creating a self-validating storage pool from the top down.

ZFS also comes with powerful performance-enhancing features. First, it supports read and write caching, which allows for tiered hybrid storage. This means that the system can utilize high performance SAS and SSD drives (or other flash-based storage) in conjunction with high capacity storage drives to optimize storage pool performance. Most workloads can leverage the performance benefits of storage tiering, though tests have shown performance increases of 8x to 23x under certain workloads

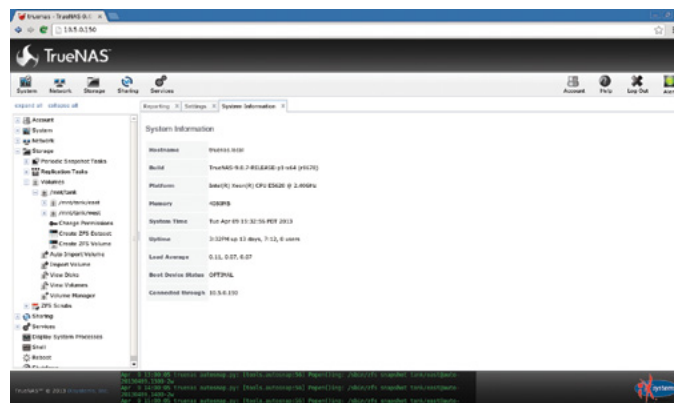


Figure 2. System Information page of the TrueNAS WebGUI

than without the cache (<https://blogs.oracle.com/brendan/entry/test>).

As part of the hybrid and tiered storage feature, ZFS increases performance through its *Most Frequently Used* (MFU) and *Most Recently Used* (MRU) caching algorithms which categorize data by date and usage, allowing prioritization of the system's resources. Active data will be first in the queue for RAM and caching devices, minimizing the need to access slower spinning disks.

Qualified Bill of Materials, Tested Features

TrueNAS is specifically designed around its own hardware platform, which allows for tightly-integrated and platform-specific functionality that cannot perform on disparate hardware. Also, since the software development team uses TrueNAS qualified hardware during their coding, that further ensures that the hardware and software function in harmony. Users get the convenience of a full-fledged appliance that saves them the trial and error process of assembling one from scratch.

Since the system operates on professionally qualified hardware, it's easier to pinpoint the source of any issues that may arise. Any bug fixes and issue escalations are brought directly to the TrueNAS team. This open line of communication also gives the user the power to influence the development roadmap. Today's bugfix is tomorrow's patch!

Based on Open Standards

TrueNAS is an enhanced version of FreeNAS and is ultimately based on FreeBSD. This means that FreeNAS runs on open standards and provides the user with benefits over closed, proprietary alternatives. Both the FreeNAS and FreeBSD projects have active communities, large user bases, and thorough documentation. By extension, TrueNAS users enjoy these benefits as well. There are thousands of people downloading the newest versions of FreeNAS every day and testing out its features on a daily basis. This ensures that only mature, tested features are rolled into TrueNAS's codebase. As FreeBSD and FreeNAS improve and evolve, so does TrueNAS.

A Complete Unified Storage Offering for the Enterprise

All of these factors converge to make TrueNAS an integrated hardware and software solution that is all at once reliable, powerful, and highly scalable. By utilizing hybrid storage with tiered caching, TrueNAS outstrips standard spinning disk storage configuration performance by or-

ders of magnitude. The intuitive web interface puts powerful tools at the user's fingertips for a wide range of storage functionality. TrueNAS takes the guesswork out of storage and comes supported by the team of professionals that brings the world FreeNAS, providing a complete unified storage solution for the enterprise environment that you can rely on.

If you want to utilize "FreeNAS" in an enterprise environment, TrueNAS delivers a time-tested product backed by a team of top-notch BSD developers. Whether it's for primary or secondary storage, FreeNAS's robust feature set, combined with proven hardware configurations, allow TrueNAS to meet the needs of businesses and organizations. And now you know.

MARK VONFANGE

Mark VonFange is the Professional Services Manager at iXsystems, providing oversight and coordination of its FreeBSD, PC-BSD, and FreeNAS support and development services. The Professional Services Team provides services ranging from mission critical support to software and firmware development to private consultation. Mark also develops internal and external documentation for division sales and marketing.

ANNIE ZHANG

Annie Zhang works in the marketing department at iXsystems. She started using open source software during her sophomore year at Wellesley College. Since then, she has increasingly relied on Open Source products; she appreciates the security, flexibility, interoperability, and of course, price.



NET OPEN SERVICES IS AN APPLICATION HOSTING COMPANY FOCUSED ON OPEN SOURCE APPLICATIONS MANAGEMENT IN HIGH AVAILABILITY ENVIRONMENT.

NET OPEN SERVICES IS PROUD TO PROVIDE A HIGH QUALITY SERVICE TO OUR CUSTOMERS SINCE 10 YEARS.

OUR EXPERTISE INCLUDES:

- CLOUD COMPUTING, PUBLIC, PRIVATE AND HYBRID CLOUD MANAGEMENT (OPENSTACK, CLOUDSTACK, RED HAT ENTERPRISE VIRTUALIZATION)
- REMOTE MONITORING AND MANAGEMENT 24/7
- NETWORKING AND SECURITY (OPEN BSD, IP TABLE, CHECKPOINT, CISCO,...)
- OS AND APPLICATION MANAGEMENT (FREE BSD, OPEN BSD, SOLARIS, UNIX, LINUX, AIX, MS WINDOWS)
- DATABASE MANAGEMENT (ORACLE, MYSQL, CASSANDRA, NOSQL, MS SQL, SYBASE...)
- MANAGED HOSTING IN CARRIER CLASS DATA CENTERS
- DISASTER RECOVERY



WE PROVIDE SERVICES IN EVERY STEP OF THE PROJECT LIFE, DESIGN, DEPLOYMENT, MANAGEMENT AND EVOLUTIONS. **NETOPENSERVICES** TEAM INCLUDES EXPERIENCED LEADERS AND ENGINEERS IN THE INTERNET SERVER INDUSTRY.

OUR TEAM HAS 15 YEARS OF EXPERIENCE IN DEVELOPING INTERNET INFRASTRUCTURE-GRADE SOLUTIONS AND PROVISIONING INTERNET DATACENTERS AND GLOBAL SERVICE NETWORKS TOGETHER.

WE OFFER EXCEPTIONAL HARDWARE SUPPORT AS SOFTWARE SUPPORT ON UNIX/LINUX AND OPEN SOURCE APPLICATION. **NETOPENSERVICES** DELIVERS THESE CUSTOM-BUILT LINUX AND UNIX SERVERS, AS WELL AS PRECONFIGURED SERVERS AND SCALABLE STORAGE SOLUTIONS, TO OUR CUSTOMERS. WE ALSO OFFER CUSTOM DEVELOPMENT AND ADVANCED-LEVEL UNIX/LINUX CONSULTING SOLUTIONS.

Creative Integrations

Workflow Improvements with FreeNAS and TrueNAS

“We’ve deployed FreeNAS and TrueNAS at several client locations. The visual effects people are loving the TrueNAS system and how easy it is to manage.”

- Tim Nagle, Owner of Creative Integrations

Creative Integrations is a full service engineering/integration firm that specializes in post-production, recording, animation, and broadcast facilities. With over a decade of experience, Creative Integrations helps their clients design, improve, and streamline workflows.

The company is owned and operated by Tim Nagle, with experience on many sides of the industry from engineering, production, post-production, music and broadcast all over North America and Europe. Creative Integrations offers a proactive, technical view of the landscape customers are faced with and works with clients to find the most effective, economical solutions available in the current marketplace. With offices in Dallas, TX and New York City, NY, Creative Integrations is positioned to fulfill the needs of any project on schedule and on budget.

The Storage Challenge

Tim Nagle, owner of Creative Integrations, is always on the lookout for new solutions that can help their customers improve operational efficiency. While at the *National Association of Broadcasters* (NAB) conference in April of 2012, Tim discovered FreeNAS as an open source unified storage operating system. He had a client with an existing NAS, but the client didn’t have anyone to administer it. Although Tim could do it himself, the time investment needed to learn his customer’s storage platform would cause

delays and keep him from other critical duties. It would also mean constant additional oversight.

Improved Workflow on Open Standards

The solution was clear for Creative Integrations. They completely rebuilt their client’s backup server with FreeNAS for a fresh start. As a result, workflow improved greatly, throughput increased significantly, and workers found the intuitiveness of the FreeNAS interface easy to use. After the success of the initial deployment, Creative Integrations converted the other NAS server to FreeNAS and set it to run nightly replication over Rsync, preserving critical data automatically and providing the client with extra peace of mind.

“I heard about FreeNAS at a conference and realized it was an immediate necessity for one of my clients. They had an existing NAS system that was not being used to its full potential. After I rebuilt their backup system using FreeNAS, their workflow improved by probably 5 times what they were able to do previously. This was largely due to the increases in throughput and the ease of management of the FreeNAS platform.”

-Tim Nagle

TrueNAS in the Media Production Environment

One of Creative Integrations’ clients, a children’s hospital in Ft. Worth, TX, started a hospital TV station for the

patients to watch and participate in. They wanted to build a FreeNAS system for their multimedia storage, so the company helped consult them on which hardware would be the most compatible with their operating system and use case. During this process, they contacted iXsystems and found that the TrueNAS appliances were already using the exact kind of hardware Creative Integrations was considering.

TrueNAS also came with commercial-level support, something no enterprise can go without. Having expert assistance during deployment and in the production environment can mean the difference between minutes or hours of downtime. It's also an extra line of security for a business's critical data. The professionals at iXsystems will provide Creative Integrations with timely support throughout the lifetime of their TrueNAS appliances.

"The support was really it. I can handle a lot of this, but as soon as I get in over my head, I need to have someone I can call. I love forums, but the problem is I usually get more than one answer. When a system is in production, it becomes critical to get a prompt, unambiguous response."

-Tim Nagle

Scalable Storage in a Post-Production Environment

Another of Creative Integrations' customers, a full-service post production studio in Dallas, needed centralized storage to house all their media, so they turned to TrueNAS. They decided on a 36-bay solution with dual, 6-core Xeon processors, 96GB of RAM, and two 10GbE, 4-port cards. These ports were configured with LACP and connected to a 24-port Netgear SXM7724 10GbE switch which serves all of the main client editing stations and utility computers. Non-production employees and the audio system are connected through a 24-port GSM7252 1GbE switch due to lower throughput needs.

About iXsystems

iXsystems builds rock solid enterprise-class server and storage solutions. All of our products are assembled, tested, and shipped from our company headquarters in Silicon Valley. Technical support is provided in-house by the same engineers that build the systems. Thousands of companies, universities, and U.S. Government departments have come to rely on iXsystems' customer-first commitment to excellence. iXsystems champions the cause of Open Source technology by dedicating extensive resources to several FreeBSD community projects: FreeNAS, PC-BSD, FreeBSD, and TrueOS.



This system serves eighteen users, twelve of them requiring high bandwidth. It shares primarily over AFP, with CIFS used as a separate login system for inter-office access. The client uses their TrueNAS system to send projects, store media, and even render media out to the storage servers. The TrueNAS appliance eliminated the bottlenecks associated with using spinning disks to locally store data. With separate login systems, access to their critical media storage is limited to authorized production users only, adding an additional level of data security.

Over time, the client started running out of memory on their TrueNAS system and needed to expand. iXsystems provided Creative Integrations with a 45-bay expansion shelf, adding another 135TB of raw capacity to the initial 70TB. In total, they have 158TB of usable storage running RAID-Z, single disk parity with ZFS, with disks grouped into 5 drive vdevs (LUNs). They were able to introduce the additional storage into their operation quickly and without any hang-ups.

"There was one point where we were getting really low on storage space and it was getting a bit uncomfortable. Adding on an expansion shelf was a very quick transition and I know I can just get another the next time we are running low on data. It's nice to know that I won't have a huge headache on my hands every time we need to scale up."

-Tim Nagle

Flexible Solutions for any Media Environment

Both FreeNAS and TrueNAS have been critical in improving the operations of a growing number of clients for Creative Integrations. FreeNAS provides an open source storage platform for its customers to use in their media environment. TrueNAS has additional features that provide extra data security with commercial grade support from the developers that bring the world FreeNAS. Both platforms provide the functionality, performance, and stability any organization needs to improve operational efficiency, and Creative Integrations looks forward to utilizing both in the future.

MARK VONFANGE

Mark VonFange is the Professional Services Manager at iXsystems, providing oversight and coordination of its FreeBSD, PC-BSD, and FreeNAS support and development services. The Professional Services Team provides services ranging from mission critical support to software and firmware development to private consultation. Mark also develops internal and external documentation for division sales and marketing.

FreeNAS a Migration Story

In 2010, I was employed as a system administrator, and one of my aims was to administer the file sharing service. After trying a few different solutions, all based on Linux and well-known protocols (CIFS and Netatalk), I decided to switch to FreeNAS. While it was an old version, and therefore without any of today's gadgets, the choice was the right one. Even years after I left the company, the other admins are able to run the machine without any problems.

This article briefly summarizes the migration process and what advantages FreeNAS provided. This is not a technical article, it is just a "tale" of how I've managed the migration. Please note the procedure described here is not necessarily the best method or the one that optimally applies to other scenarios.

The Context

The company was running several Linux physical servers – specifically, one for ERP, one for file sharing, one for e-mail, one for external services (FTP, Web Server), and one as a gateway/proxy/firewall. There were around 200 workstations, with mostly Microsoft Windows PCs and forty-five Apple Macs (as graphical workstations). The entire network was wired as Gigabit Ethernet and there were three access points used by the sales team who were using Microsoft Windows laptops.

While the majority of the clients shared small to medium files, the graphical workstations shared files ranging from 50MB-600MB each. Moreover, the Microsoft Windows PCs frequently accessed many documents, meaning that the users were usually opening a file several times a day to add, modify, and save the content. Meanwhile graphical workstations accessed a small subset of the shared files infrequently. While it was true that the PCs seldom accessed graphical files, the graphical workstations often had to access files shared by PCs. Finally, all file access

had to be authorized and granted on a per-user and per-office policy. Luckily, all the users and their roles were already enumerated via an OpenLDAP server (the machine running the ERP).

Due to the above requirements, I decided to implement a NAS solution that had to:

- connect to the OpenLDAP server in order to authenticate the users;
- provide a per-office share as user(s) workspace;
- provide a way to do automatic backups at the fastest speed possible;
- provide a way to specify exactly which user(s) can access which document(s);
- be as reliable as possible; and
- have an easy way to add extra disk space on demand.

The motivation that pushed me to choose a NAS dedicated system was the following: as time passed by, the sharing service got more and more complex with the need to deal with different operating systems and versions, as well as different platforms, file formats, and so on. While Open Source software like Samba and AppleTalk covered the scope well, I was also looking for a more integrated solution, and most notably, something easier to use, without having to give up flexibility and stability. The "ease of use" point was

motivated by the fact that I needed to share the NAS configuration with other administration profiles, and while I felt (and still feel) comfortable with the command line, others do not.

Migrating to FreeNAS

The choice was, as readers can imagine, to adopt FreeNAS. The reason was that the company had already migrated some of the servers from Linux to FreeBSD-based solutions. The first was, in fact, the gateway/firewall appliance that was running pfSense, then the external server was migrated to FreeBSD to provide both Web and FTP and then a few internal workstations (mainly by the IT employees). Therefore, choosing a FreeBSD system for the NAS seemed like a natural step, and since I wanted to have something dedicated and integrated, the FreeNAS project seemed like the right choice to me.

Of course, the decision was based on several factors related to FreeBSD and FreeNAS, and in particular:

- a low startup cost;
- the stability of the system;
- the availability of ZFS as a file system;
- compatibility with the main sharing technologies (CIFS, AppleTalk);
- an easy to use interface for administration.

In the following, all the above will be addressed.

A Low Startup Cost

Luckily, the company I was employed by was already used to Open Source solutions and did not want to get locked into vendors' products if possible. And luckily, FreeNAS is a solution that does not impose a vendor lock-in; it is free and does not require a lot of resources to run. This allowed me to implement a NAS solution exploiting good, but cheap, hardware with no cost at all for the software.

The Stability of the System

One thing I noticed while working with both Linux and FreeBSD machines was that the latter ran smoothly for a longer time. Usually, keeping a FreeBSD machine up-to-date was a simpler task than keeping a Linux one up-to-date. This was particularly true also because the company was running different Linux distributions, each with its own update policies, release schedules, and internal mechanisms. In other words, the FreeBSD machines all behaved coherently. Finally, FreeBSD proved it can run longer than Linux without hanging or needing reboots on

the same (old) hardware. Of course, while this strictly depends on the Linux distributions, the versions, the hardware, and the administrator(s) skills, the above was true in this context.

The Availability of ZFS as a File System

Two features of ZFS were particularly attractive to me for this job: the availability of snapshots and the integrated volume manager.

On the other Linux machines, we had to play with partitions and mount point each time a new disk was added, and that was due to the error of not setting up a volume manager from the beginning. With ZFS the problem disappeared.

Moreover, the idea of snapshots was appealing to me because it allowed me to implement an automated backup system transparent to the user. In fact, often a user started modifying a large graphical file just to discover he or she made a mistake and had to start over. Forcing the users to learn a revision control system was too complicated; therefore I decided to make snapshots on a regular schedule to help the users to recover "damaged" files. On the other hand, the users were informed that after a certain amount of time snapshots would be made persistent, meaning history would no longer be available.

Just for the sake of clarity, besides the snapshots, there was a regular backup technique.

The Compatibility with the Main Sharing Technologies

Having a heterogeneous environment and having to grant simultaneous access to Unix/Linux machines, Apple OS X computers, and Microsoft Windows PCs requires deep support for several sharing protocols and technologies. One of my aims was to reduce overlap in choosing protocols, possibly down to a single one for ease of administration. The fact that FreeNAS supports CIFS, AppleTalk, and NFS allowed me to switch to the best protocol available depending on the client machine.

An Easy to Use Administration UI

Usually Unix administrators are tied to the *Command Line Interface* (CLI), and to some extent, they also believe that using a GUI is child proof. However, not all administrators are Unix ones, and this was the case: the company had one Microsoft Windows and an Apple (Junior) administrator, both with little or no Unix skills. I decided that apart from the need for them to acquire some Unix skills, they had to be enabled to participate in the NAS management

to both reduce my overhead and also make better use of the resources (in this case, people) available.

FreeNAS came with an excellent Web UI. I was able to let my colleagues take part in the configuration phase and to grow up to the point that they were able to create and manage shares, crontab and backup scripts, and users' privileges without any further help. In this way, I was mainly needed for the harder tasks and low level configuration while they were able to do day-to-day maintenance.

The Migration

There was around 1TB of data to be migrated from one Linux server, and another 400GB to be consolidated to the FreeNAS installation. In fact, while the main project was to move only the Linux file shared part (the former), reviewing the sharing status revealed that other stuff needed to be placed on the FreeNAS system (such as files and folders users shared among their workstations without informing us).

Since the total size of the data was not that huge, I was able to migrate everything within hours, and in particular, during the night. However, I did not run a single migration, but proceeded with different steps.

In the beginning, I set up the FreeNAS machine using a USB stick to boot, so as not to waste even a single byte on the hard disks. After the initial required setup (network interfaces, admin password, etc.) I connected the FreeNAS to the OpenLDAP server to sync all the accounts. This was really simple thanks to the great tools FreeNAS provides.

After that, I decided to remotely mount the already available shares into the FreeNAS and to "republish" them on the network, so that the FreeNAS was effectively a proxy for the shares. Thanks to the fact that the FreeNAS and the original server were accessing the same account database, I did not have any problems with the permissions. My choice at this point was to republish shares using only the CIFS protocol in order to have a simpler situation than using a per-client protocol.

I ran a few tests for a week to ensure that everything was working fine, even though performance was not good due to the fact that the FreeNAS machine was not serving local data.

After that, I implemented a local backup, that was a set of scripts, to do regular backups of the FreeNAS and its content over a part of the FreeNAS itself, as well as to the central backup machine. This was the only phase in which I needed to use the FreeNAS CLI, since we had to deal with shell script development. More in detail, I had to do some "tricks" to place scripts on the USB stick and to be

able to install a few other tools out of the ports. However, as soon as the scripts were in place, I took back the Web UI to place them in the crontab table.

Having the FreeNAS machine acting as a proxy and the backup ready to work, I started the migration of the clients so that they were redirected to the FreeNAS server instead of the original Linux NAS one.

I deployed a ZFS file system with roughly one per office/user file system, setting up quotas and assigning ownership and permissions.

Finally, I migrated all the data, using rsync on a regular basis for a couple of passes, to reduce the total copy overload. The migration was performed during a weekend to reduce the problems with live data and to avoid having to disconnect the users.

So Far, So Good!

While the migration could seem long for such a moderate amount of data, I preferred to do it in that way because it was my first experience with FreeNAS in an enterprise environment. It is worth saying I did not have any particular problem with the migration, and the system is very stable and rock-solid.

During this time, I added a few extra features to our FreeNAS machine, like the aggregation of the two on-board network cards, as well as regular updates of the whole system, without incurring any downtime.

The last time I had the opportunity to check the FreeNAS machine, it was running for a year without any particular problem, and remember I'm talking about a machine used for day-to-day work but that we, the administrators, kept forgetting we had!



Physical or Virtual?

In the past years, virtualization has been strongly advocated, and today it is quite common to find whole servers virtualized onto a single hardware machine. It may sound strange that the described FreeNAS server was implemented physically instead of virtually, but the reason for that was to keep the total cost as low as possible. As described in the "Low Startup Cost" section, the company was already in the Open Source and commodity hardware mindset. Therefore, instead of investing money in a huge supercluster, the company preferred to buy and assemble "small" and cheap servers. This also allowed the company to implement replication solutions quite easily.

The only problem I ran into was tied to the USB stick I used as root filesystem: one day I was physically moving the machine from one place to another only to discover that the USB stick was unable to boot the machine anymore. It was, however, a "no pain situation" since I had another "clone" USB stick from which I started the machine.

Conclusions

My experience with FreeNAS is nothing but good on every aspect. It is worth noting that the effort done by the FreeNAS team to provide a clean, usable, and easy to understand Web UI is really important in my opinion. While "real" Unix system administrators will always be able to fire up a console and see what is happening under the hood, having a good UI allows other people to jump into the admin side. Of course this does not mean that everyone is automatically skilled as an administrator, but it means that the admin burden can be scattered amongst different people with different skills. This was indeed my experience. Thanks to the UI, non-Unix administrators were able to learn quickly and perform many routine tasks on the FreeNAS configuration.

Today, even years after I left the company, the other admins are still able to keep the FreeNAS box up-to-date and it runs without any particular problem. This is proof of the stability of the system itself.

LUCA FERRARI

Luca Ferrari lives in Italy with his wife and son. He is an Adjunct Professor at Nipissing University, Canada, a co-founder and the vice-president of the Italian PostgreSQL Users' Group (ITPUG). He simply loves the Open Source culture and refuses to log-in to non-Unix systems. He can be reached on line at <http://fluca1978.blogspot.com>.

Keep
FreeBSD
Free!



The
FreeBSD
FOUNDATION

Support FreeBSD
by donating to
The FreeBSD
Foundation



To find out more,
please visit
our Web site:

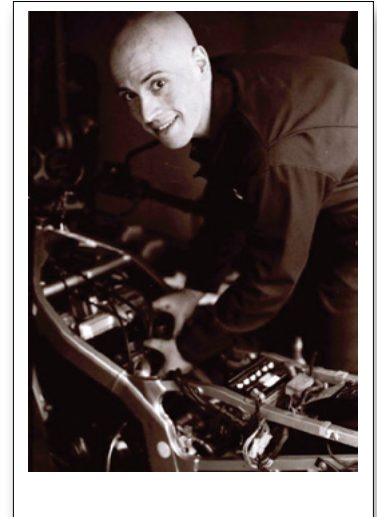
www.freebsdoundation.org

The Interview with

Alfred Perlstein

VP of Software Engineering at iXsystems

Alfred has been working on the FreeBSD kernel for the past 13 years. His areas of interest have been file systems, multi-processor support, performance, and stability of FreeBSD. He has alternated between CTO/VP roles at companies like OkCupid, to kernel developer positions at Apple and Juniper Networks. His current role is FreeNAS Project Manager and VP Software Engineering at iXsystems. He continues to contribute to the FreeBSD project when time allows and encourages his team to do so as well. Recently, he agreed to give an interview to BSD Magazine.



BSD Magazine: Hello Alfred, could you tell us how you got into FreeNAS?

Alfred Perlstein: I had known iXsystems for many years through their support of the FreeBSD project. When it came time to purchase some storage at my previous job I naturally looked to iXsystems for a solution. We picked a FreeNAS – based server and were really impressed with the software. I was so impressed with the FreeNAS project, the developers behind it, and iXsystems that I reached out to work at iXsystems so that I could work on FreeNAS as well as TrueNAS.

BSD: Tell us the story behind FreeNAS project. How it started and was it Open Source in nature from the beginning? What was and is the role of iXsystems in it?

AP: FreeNAS started with a developer named Olivier Cochard-Labbé in 2005. He wanted to convert an old PC into a NAS box for his home, but there wasn't an easy way to do that. He worked on that for several years with a couple of other developers, using a M0n0wall-based web UI.

Eventually his interest and that of the other developers waned, and one of them proposed moving FreeNAS to a Debian Linux – based system instead of FreeBSD. At iXsystems, we'd been using FreeNAS for years and selling people servers that were specifically for FreeNAS, so Matt Olander decided we should step up and offer to continue BSD-based development of the project. Olivier agreed, and iXsystems began the FreeNAS 8 project, re-implementing FreeNAS using Django and FreeBSD 8. Since then we've brought FreeNAS up to the most recent version of ZFS and added new features like encryption and jails support.

BSD: Can you give us a couple of examples of successful stories with FreeNAS?

AP: One time we got a random call from the United Nations, looking for FreeNAS support for a 100 TB deployment! Actually, that sort of thing happens all the time. Customers call in from huge organizations around the globe telling us about these giant FreeNAS installs they've got running critical business functions. Every story brings a smile to our faces.

BSD: Can you also describe what is the average deployment of FreeNAS and the biggest one you know?

AP: Most FreeNAS deployments are small home users, probably not more than 12 TB. This is due to most FreeNAS devices being small boxes with just a handful of consumer hard drives. Businesses will often deploy systems with over 20TB in a rackmount chassis, often for backup purposes. We've heard of universities, however, who have created FreeNAS deployments upwards of a petabyte.

BSD: Have there been any mishaps with FreeNAS? Any technical problems? How were they solved?

AP: Like any software, FreeNAS has gone through some challenges. Early in FreeNAS 8, it became necessary to double the size of the system image. This meant that from 8.0 to 8.0.1, everyone had to use the CD image to upgrade, or perform a reinstall and import if their install media was too small. Most of the time, we're able to identify problems early, document and fix them, and communicate with the community very quickly what the issue is.

BSD: How about the relationship with the community? Is it possible for a community member to contribute, and how are those contributions handled?

AP: There are lots of ways for community members to help out. One great way is to produce plugins – in that case, the contributor is free to distribute the plugin however they like. We always appreciate people who help others on the forums, as well as reporting bugs or making feature suggestions. It's also possible to submit patches, but that doesn't seem to happen very often.

BSD: Is there official training or tutorials, for example at conferences, related to the adoption or migration to FreeNAS?

AP: Our Community Manager, Dru Lavigne, goes to many conferences and runs small classes on how to get started with FreeNAS. There is also a series of video walkthroughs available on FreeNAS.org to help people get started, along with a fairly exhaustive online manual: <http://doc.freenas.org/>.

BSD: What is the target audience for FreeNAS? Is it suitable for enterprise or home/small office deployments? Or, both?

AP: FreeNAS is about home users first, but really anyone except businesses with the most stringent requirements will

If you wish to contribute to BSD magazine, share your knowledge and skills with other BSD users – do not hesitate – read the guidelines on our website and email us your idea for an article.

Join our team!

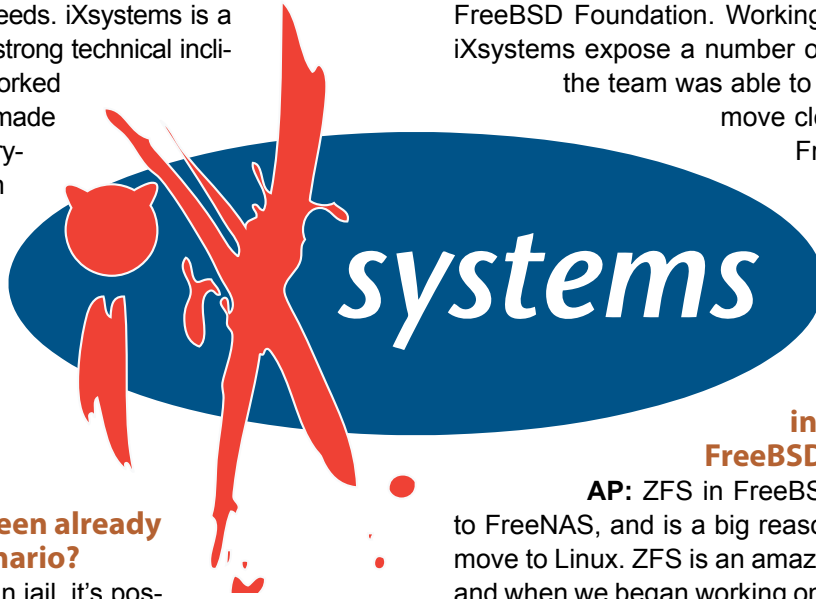


Become BSD magazine Author or Betatester

As a betatester you can decide on the contents and the form of our quarterly. It can be you who read the articles before everybody else and suggest the changes to the author.

Contact us:
editors@bsdmag.org
www.bsdmag.org

find that it suits their needs. iXsystems is a small business with a strong technical inclination, and while we worked on FreeNAS, we also made sure to include everything we would want in a storage solution. As I mentioned, we've used FreeNAS in one way or another to host our internal file share for years.



BSD: Is FreeNAS suitable for cloud storage or has it been already used in such a scenario?

AP: Through the plugin jail, it's possible to install Owncloud or any similar FreeBSD-compatible tool. Owncloud lets a user upload and access their files on a self-hosted server from anywhere in the world.

BSD: Is FreeNAS based on pure FreeBSD or does iXsystems modify the kernel tree?

AP: FreeNAS uses NanoBSD to create a stripped-down image, and unnecessary drivers are excluded, such as Wi-Fi and video card drivers. NanoBSD also offers us the flexibility of a backup system image in case an upgrade fails or the user needs to revert to a previous version.

BSD: Does FreeNAS support all the platforms that FreeBSD does? What are the better architectures on which FreeNAS can run?

AP: FreeNAS runs exclusively in x86 and x86-64 architectures. 64-bit is really the target architecture for FreeNAS, due to the RAM requirements of ZFS. Users with older hardware are encouraged to run UFS – based systems, although they will not experience the full utility of ZFS. There is progress being made to fix ZFS into smaller memory machines. This may be available in later versions of FreeNAS.

BSD: What is the relationship between the FreeNAS and FreeBSD projects? Does iXsystems contribute back to the FreeBSD project?

AP: iXsystems is a major sponsor of FreeBSD, and most of the FreeNAS development team are also developers of the FreeBSD project as well. Our CTO, Matt Olander, is part of the FreeBSD Marketing Team, and the Community Manager, Dru Lavigne, is a member of the board of the

FreeBSD Foundation. Working on FreeNAS has helped iXsystems expose a number of bugs in FreeBSD, which the team was able to fix. With FreeNAS soon to move closer to the cutting edge of FreeBSD, we expect even more mutual benefit in the future.

BSD: It seems that FreeNAS storage is strongly based on ZFS. Did the inclusion of ZFS into FreeBSD drive that choice?

AP: ZFS in FreeBSD has been a great boon to FreeNAS, and is a big reason why we didn't want it to move to Linux. ZFS is an amazing technology for storage, and when we began working on FreeNAS 8 we decided to center it around ZFS. Since then, we've made ZFS more accessible than ever, and even brought in improvements like the encryption option in FreeNAS 8.3.1.

BSD: Does FreeNAS cover all the features that ZFS provides, like deduplication?

AP: As of FreeNAS 8.3, FreeNAS is at ZFS v28 just like FreeBSD. This means that RAID-Z3, deduplication, and detachable ZILs are all available in FreeNAS. The web UI reflects the ZFS pool, so even if someone performs an unconventional pool configuration from the shell, that will be fully reflected in the UI.

BSD: Does FreeNAS support other less memory consuming file systems like UFS/FFS as well?

AP: Most of FreeNAS' functions work perfectly well on UFS, which is supported in the Web UI. A handful of other filesystems are also available in the importer for read-only.

BSD: How does FreeNAS compare to other Unix/Linux storage solutions?

AP: FreeNAS is very good at having a core selection of stable services for NAS available and easy to configure. FreeNAS is also probably the best at offering the power of ZFS. Others are very adaptable to third-party software, and this is something we're working on improving in FreeNAS as well.

BSD: Are there any guidelines or automated tools for migrating from other (not strictly BSD) solutions to FreeNAS?

AP: None that we currently know of however, many people have successfully used rsync to migrate data. If a user decides to create a tool, we would love to integrate it into FreeNAS.

BSD: What is the added value of using FreeNAS instead of using pure FreeBSD (or another Unix system)?

AP: Certainly ease of use. The power of FreeNAS is that you can have network storage running with advanced filesystem features as well as integration with Active Directory in very little time, all through a user friendly GUI. We have many video tutorials online as well as written documentation that can help even your most basic user get up and running in a matter of minutes. The same configuration would take an expert days, if not weeks, to set up on FreeBSD.

Another bonus feature of using FreeNAS is that we have a huge userbase on a specific version of FreeBSD. This allows us to maintain a level of stability specifically tailored to our use-case. We inbox only vetted fixes from FreeBSD that we know address issues our users are seeing.

BSD: FreeBSD provides mainly two technologies for disk data encryption, GELI and GBDE. Does FreeNAS have any way to encrypt some or all of its storage?

AP: The most recent release of FreeNAS, 8.3.1, added a GELI-based encryption option for ZFS. This allows the on-disk data of a ZFS pool to be encrypted with a key and optional passphrase, with a recovery key option. Obviously this means that only FreeNAS or FreeBSD can import an encrypted pool, but all the other functions of ZFS including snapshot replication are unaffected. This is designed only to protect offline disks, or disks in which failed firmware prevents the deletion of sensitive data.

BSD: iXsystems provides a complete solution, both hardware and software, based on FreeNAS. What are the advantages for using such devices instead of using commodity hardware or other supported hardware?

AP: iXsystems is a hardware company with a long history of excellence. When you purchase a FreeNAS solution from iX, you can be assured that months of hardware qualification have been performed to make sure every single component is within specifications set by the CPU, motherboard, disk and controller manufactur-

ers. We first make sure the recommended hardware is fully compatible, and then we install FreeNAS to your exact specifications. A FreeNAS solution purchased from iXsystems will be the most stable FreeNAS system that can be assembled and ready to deploy out of the box.

There is also a full enterprise-grade appliance we offer, known as TrueNAS, which provides the user full enterprise support and additional features, like high-availability active/passive failover, for instance.

Either way, by doing business with iXsystems, you're supporting the FreeNAS, FreeBSD, and PC-BSD projects.

BSD: Consider a scenario where several FreeNAS machines are involved, does FreeNAS support (or will support) replication/mirroring and centralized management (let's say clustering)? Moreover, are there any plans to implement a distributed file system between FreeNAS nodes?

AP: FreeNAS already supports replication snapshots to backup servers. Since FreeNAS is based on FreeBSD, any file system technology that appears in FreeBSD (and we decide whether it will provide utility to FreeNAS users) will be included in the upcoming FreeNAS releases. At this time there are no clustering solutions. However we do have plans for a distributed backup system to help people save their data.

BSD: Apart from the "raw" storage, what are the main features that FreeNAS provides? For example link aggregation, backup of the configuration, scalability, support for different protocols, and so on...

AP: I'd say you hit all the big points of FreeNAS right here. We support link aggregation of multiple 10gigE interfaces, configuration backup is just a single click in the UI, we already have a great track record for stability, and we support multiple protocols: SMB/CIFS, NFS and AFP.

BSD: Does FreeNAS include (or will it include) the PCBSD PBI format for additional applications or are administrators required to use FreeBSD packages and ports?

AP: Yes! PBIs are supported under the plug-ins architecture (<http://doc.freenas.org/index.php/Plugins>). With our next release, the entire PBI catalog of PC-BSD (<http://pcbsd.org/>) will become available to FreeNAS users as well.

BSD: NAS based systems for home and domestic users often offer multimedia streaming and Peer-to-Peer features. Does FreeNAS embed any of them natively or via a third party package?

AP: One of our most popular p2p apps is Transmission. There are also a number of usenet related plugins. By leveraging the plug-in jail, just about any FreeBSD port that supports streaming can be installed and leveraged to build a media center.

BSD: Usually a home user will buy some sort of NAS hardware to plug in the network and forget about. FreeNAS requires a little extra effort, since the user has to install and configure the machine. Therefore, why should a home user decide to run FreeNAS?

AP: This is exactly why a home user would pick FreeNAS. It is an “install and forget” appliance if you want it to be. However, because it is based on the modern and powerful FreeBSD operating system, is open source, and supports plug-ins, the possibilities for expansion are endless.

We really have a sweet spot here, the user who just wants to “set and forget” can do that, and the tinkerer can do as much tinkering as they like.

BSD: What are the differences between the community based FreeNAS solution and the TrueNAS solution proposed by iXsystems?

AP: TrueNAS takes the powerful interface of FreeNAS and puts it on a purpose-built hardware platform, with every piece tested and verified by iXsystems. Since we provide both the hardware and the software, iXsystems can guarantee a much higher level of support with TrueNAS. In addition, the stable hardware platform also allows for features like High Availability that would be very difficult to implement in FreeNAS, due to tremendous difficulty of supporting any and all hardware the user chooses.

BSD: What is the future of FreeNAS? Could you tell us a bit about ongoing development?

AP: Well right now two big things are happening.

First off, we are moving to FreeBSD 9.1/stable. This will give us more hardware support and big performance gains. There is some talk in the FreeBSD community of virtualization coming to FreeBSD 9, so we may be able to pick that up as well.

Second, the entire project is being converted to git by our two senior engineers William Grzybowski and Josh Paetzel as I write this. Once this is complete, we will be able to leverage the abilities of a distributed SCM in order

to attract more users to our project as well as help us with the ability to qualify changes in a unified build system before they are even submitted to the main project.

BSD: Why should a user migrate from another storage solution to the one based on FreeNAS?

AP: In an enterprise setting we find that FreeNAS performs as well as a number of solutions that are many times more expensive than a FreeNAS machine. For people looking for professional support and enterprise features we also have our TrueNAS product which competes on the same level of performance, stability, and features as the major SAN and NAS offerings currently on the market, at a fraction of the cost.

In the SOHO market we find that most of the solutions out there have a niche that they are good at. However, along with that you wind up with a SOHO based solution and all the limitations involved. FreeNAS is very fast compared to the rest of the options out there, constantly evolves and improves, and is an open platform.

If you find yourself wishing for better performance, FreeNAS can help you on that front with the ZFS filesystem and our carefully tuned FreeBSD operating system. Or maybe you wish to move some of the applications off your desktop PC and onto the appliance but you do not want to interrupt services for the rest of your office or family when the family PC is “installing updates” or a new video card is being installed.

BSD: Is there anything you would like to add or tell BSD Magazine readers?

AP: I’m really excited about what is going on. We have a great 8.3 release under our belts now. Our team is the best team I’ve ever worked with. We have the users (over 5 million downloads to date). What we’re hoping for is more developers. If you have done something cool on FreeNAS, either with your configuration, or by hacking the code yourself, we really want to hear from you.

I want to thank the community, our forum members, and our developers very much for all the time they put into making FreeNAS a serious product that has had an impact on a large number of users. You guys rock.

BSD: Thank you for your time and this excellent interview.

by Luca Ferrari & BSD Team

OWNED!



Quick & Easy Security and Compliance Through RedSphere's Secure Hosting Solutions

- Instantly Become PCI Compliant
- We Address Other Compliance and Industry Security Requirements
 - Penetration Testing Services
 - Source Code Security Review and Design
 - Custom Security Services and Solutions

powered by



REDSPHERE™
Our Promise is Your Peace of Mind

RedSphere Global Security
Call now to speak to a representative
719.924.5266 sales@redsphereglobal.com

www.redsphereglobal.com

From Reading to Real Life: Try Out a Demo of FreeNAS!

When I heard that the next issue of BSD Magazine would be dedicated to FreeNAS, an idea came to my mind. I thought: "We need a demo!"



NetOpenServices (<http://www.netopenservices.com>) gave me the opportunity to do it not only in terms of hosting but also in terms of security. I'm aware that giving out the admin account has its consequences. The demo is hosted like a professional one, so there should be no problems with keeping up the host, even if it becomes popular among BSD Magazine readers.

The FreeNOS is a name for a FreeNAS 8.3.1 demo host, that you can find at <http://freenos.netopenservices.org>. There is no catch – you can play with all the stuff freely and even destroy all of the data! Everything can be done with no consequences for the rest of the hosting, since an automatic kind of refresh of all the components is done every two hours. So, with every two hours all the data is erased. That's what I call a real demo.

You have a full VM with 8 GB of RAM and 12 virtual 40GB HDDs to play with.

System Information	
Hostname	freenos.netopenservices.org
Build	FreeNAS-8.3.1-RELEASE-x64 (r13452)
Platform	AMD Opteron(tm) Processor 4180
Memory	8176MB
System Time	Wed Apr 03 16:19:31 CEST 2013
Uptime	4:19PM up 1 day, 16:38, 0 users
Load Average	0.01, 0.02, 0.00
Connected through	195.154.100.15:6969

Figure 1. System

The plugins system is up, with two plugins: Firefly and Transmission.

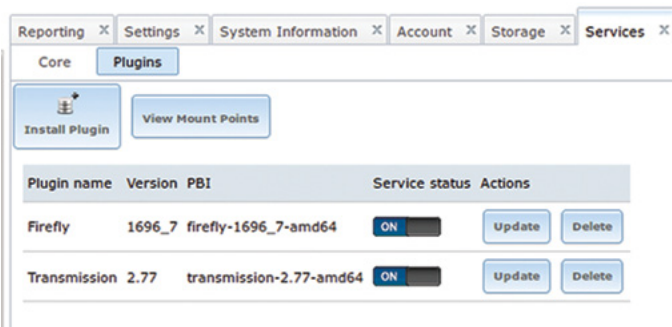


Figure 2. Plugins

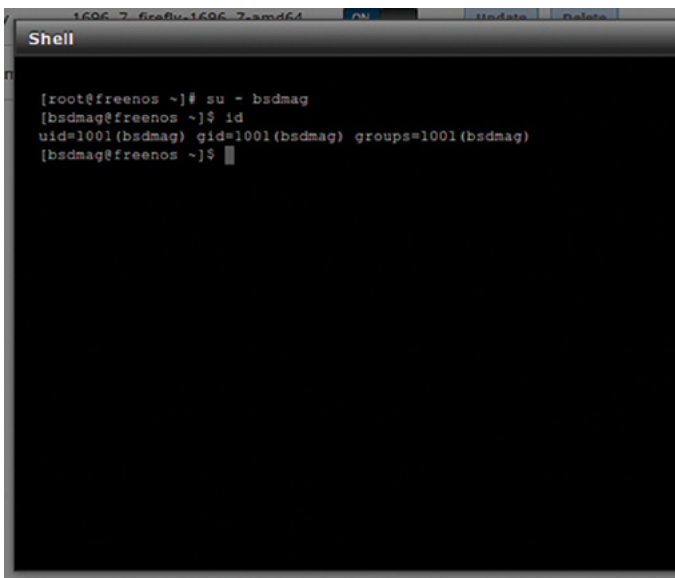


Figure 3. Shell



Figure 4. Services

For the demo, you have a user called 'bsdmag' to test the 'shell' for example.

I've put the maximum number of services up.

I'm really impressed by the work done by FreeNAS on ZFS, so I have divided the disks into encrypted ZFS. During your tests you can change it, erase, or do whatever else you want.

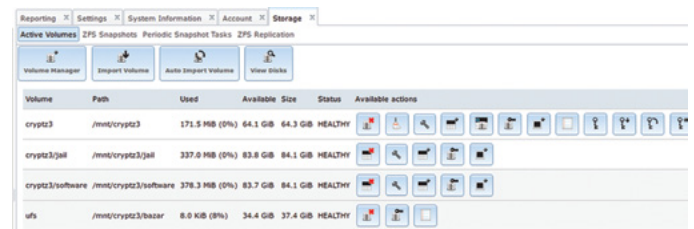


Figure 5. ZFS

All who want to test it, need to send an e-mail to bsdmag@netopenservices.org only, with 'FreeNAS – BSD-MAG' in the subject of a message to receive all the login/password details.

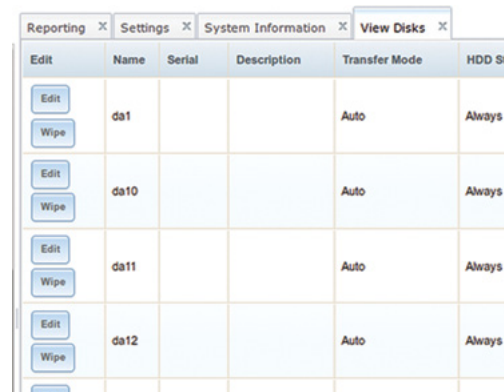


Figure 6. Disk

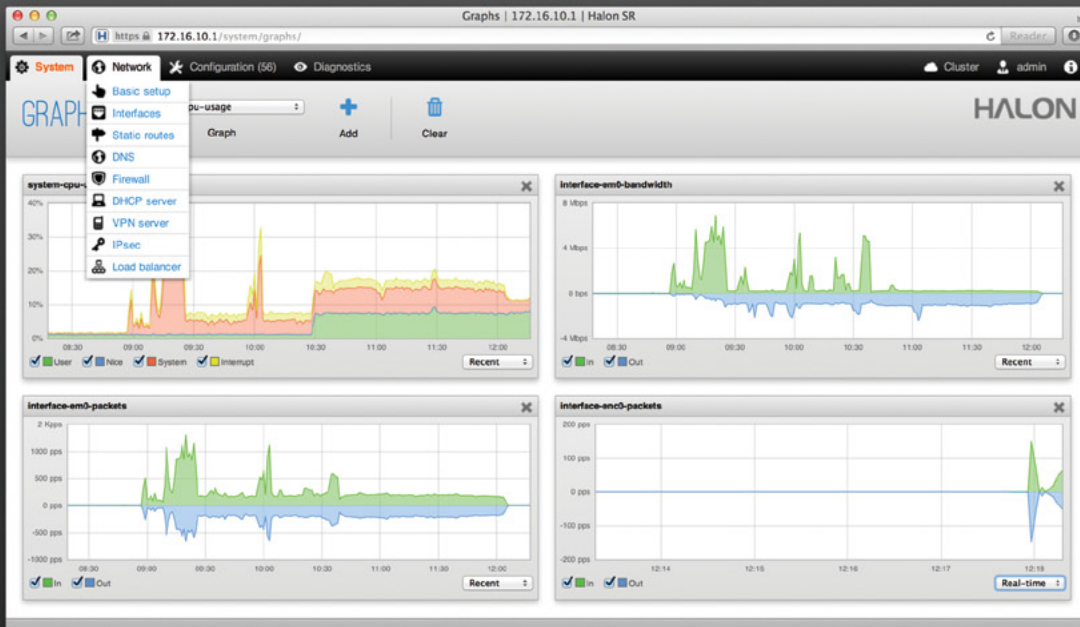
The demo is available till the end of June. However if you need more time to test it, don't hesitate to e-mail info@netopenservices.fr with a subject "FreeNAS demo" and ask for extension of this period.

I hope you enjoy the reading and let the show begin! Come and play with FreeNAS!

IMAD SOLTANI

Imad Soltani is an "old school" unix/linux/infosec/bsd/atari lover. You can reach him at <https://twitter.com/ximad>.

Networking and Security with OpenBSD Dead Simple and FREE



Key Features:

- ✓ Easy installation
- ✓ Use on inexpensive x86 hardware
- ✓ Wide diversity of platforms (VM, dedicated hardware)
- ✓ Full IPv6 support
- ✓ Dual-stack
- ✓ Address family translation, PD, etc
- ✓ Enterprise routing: BGP, MPLS, etc
- ✓ Fully featured load balancer: L7, SSL, etc
- ✓ IPsec VPN, Layer2 IP tunneling, L2TP and PPTP
- ✓ Revision based and fully readable configuration
- ✓ Open API with SOAP integration
- ✓ Full OpenBSD network capabilities without limitations
- ✓ FREE Edition for personal use

Download FREE Firewall and Router
<http://www.halonsecurity.com>

HALON



solutions

Open Source Systems Design - Administration - Consulting

- design and administration of server farms
- load balancing and high availability solutions
- ZFS file servers and storage appliances
- FreeBSD, OpenIndiana and Linux

Our open source projects:

VX ConnectBot: SSH and telnet client for Android

mfsBSD: memory-resident FreeBSD installations

zfs-stats: ZFS statistics tools

Contact information:

VX Solutions s. r. o.
Mag. Martin Matuška
E-Mail: office@vx.sk
Web: <http://www.vx.sk>





Headquarters:
San Jose, CA



855.GREP.4.IX | Contact Us

99% Compatibility

online now...

IXSYSTEMS AND YOU ARE
THE PERFECT MATCH



SHARED INTERESTS

- Enterprise Storage Solutions
- Personalized Customer Service
- Bold New Information Technology

I'm a

In

Looking for

A Technology Partner
More Technical Experience
New Business Opportunities

Visit Today!



iXsystems

Technology Partner Seeking
Resellers/Integrators for
TrueNAS™ Storage Appliance



WWW.IXSYSTEMS.COM/PERFECTMATCH

