

MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

SHADOWSOCKS PROXY SERVER ON FREEBSD

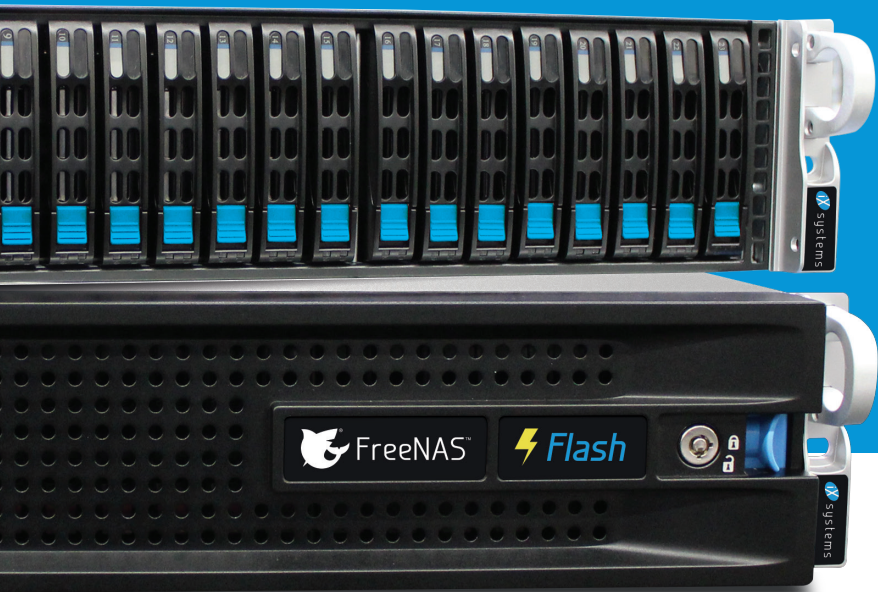
QUICKSTART WITH KUBERNETES AND GKE (PART 2/2)

INTRODUCTION TO MDB

OPENBSD 6.3

5 IMPERATIVES FOR CATALYSTS OF CHANGE

VOL 12 NO 04
ISSUE 04/2018 (104)
ISSN 1898-9144



IS AFFORDABLE FLASH STORAGE OUT OF REACH?

NOT ANYMORE!

IXSYSTEMS DELIVERS A FLASH ARRAY FOR UNDER \$10,000.

Introducing FreeNAS® Certified Flash: A high performance all-flash array at the cost of spinning disk.

- ⚡ Unifies NAS, SAN, and object storage to support multiple workloads
- ⚡ Runs FreeNAS, the world's #1 software-defined storage solution
- ⚡ Performance-oriented design provides maximum throughput/IOPs and lowest latency
- ⚡ OpenZFS ensures data integrity
- ⚡ Perfectly suited for Virtualization, Databases, Analytics, HPC, and M&E
- ⚡ 10TB of all-flash storage for less than \$10,000
- ⚡ Maximizes ROI via high-density SSD technology and inline data reduction
- ⚡ Scales to 100TB in a 2U form factor

The all-flash datacenter is now within reach. Deploy a FreeNAS Certified Flash array today from iXsystems and take advantage of all the benefits flash delivers.

Call or click today! **1-855-GREP-4-IX** (US) | **1-408-943-4100** (Non-US) | www.iXsystems.com/FreeNAS-certified-servers

DON'T DEPEND ON CONSUMER- GRADE STORAGE.



KEEP YOUR DATA SAFE!



USE AN ENTERPRISE-GRADE STORAGE SYSTEM FROM IXSYSTEMS INSTEAD.

The FreeNAS Mini: Plug it in and boot it up — it just works.

- Runs FreeNAS, the world's #1 software-defined storage solution
- Unifies NAS, SAN, and object storage to support multiple workloads
- Encrypt data at rest or in flight using an 8-Core 2.4GHz Intel® Atom® processor
- OpenZFS ensures data integrity
- A 4-bay or 8-bay desktop storage array that scales to 48TB and packs a wallop
- Backed by a 1 year parts and labor warranty, and supported by the Silicon Valley team that designed and built it
- Perfectly suited for SoHo/SMB workloads like backups, replication, and file sharing
- Lowers storage TCO through its use of enterprise-class hardware, ECC RAM, optional flash, white-glove support, and enterprise hard drives

And really — why would you trust storage from anyone else?



Call or click today! **1-855-GREP-4-IX** (US) | **1-408-943-4100** (Non-US) | www.iXsystems.com/Freenas-Mini or purchase on Amazon.

Intel, the Intel logo, Intel Inside, Intel Inside logo, Intel Atom, and Intel Atom Inside are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

EDITOR'S WORD

Dear Reader,

I hope that you are fine, and more importantly, you are optimistic that the future looks promising and bright. Here, it is Springtime, a warm and sunny period normally associated with many good thoughts and hope for fulfilling times ahead. As we enjoy this beautiful season, we need to keep tabs with the ever dynamic tech world. Hence, it's my pleasure to invite you to read and share this month's issue.

First, I would like to thank you for taking part in the survey we rolled out last week. I would also like to acknowledge Luca Ferrari with his help towards the preparation of some survey questions. Just to recap, the survey included 10 simple questions about BSD OS and its usage at work or at your home.

All your thoughts derived from the answers will not only help us create the editorial schedule, but also prepare more content that will continually appeal to you, our esteemed readers. As a matter of fact, we look forward to a more useful and practical BSD Magazine that will meet your real needs. I am grateful that you shared your thoughts. The survey is closed, but if you liked this type of engagement, your ideas such as how to streamline it are welcome. Additionally, if you would like to add your 2 cents, feel free to send me an email at ewa@bsdmag.org.

As I draft this Editor's Word, we are still working on the final look of the issue. Some articles are ready for publishing while others just need some minor modifications. So, let's see what we have prepared for you this time around. First of all, you can read the In Brief section to see and sum up what happened in April - last chance to reminisce. In this issue, you will find the second part of the article on Kubernetes and GKE. I believe that you will like the article about Shadowsocks Proxy Server On FreeBSD as many of our reviewers wanted to read it before its publication. You will also enjoy a highly-technical article by Carlos Neira, especially if you are an advanced C Programmer and SmartOS lover. To shed some light on the latest release of OpenBSD 6.3, Albert Hui's article presented the added features and identified what was changed. And if you are more into technical issues, I recommend that you read the interview with Sanel Zukan and the 5 Imperatives for Catalysts of Change as part of the Expert Speak column by E.G.Nadhan. Do not forget about Rob Somerville's column and check what Brinkmanship is.

So let's do it! Let's read!

See you next time and enjoy the issue!

Ewa & The BSD Team

P.S. Write to me any time if you need some details or would like to share your thoughts.

TABLE OF CONTENTS

In Brief

In Brief 08
Ewa & The BSD Team

This column presents the latest news coverage of breaking news, events, product releases, and trending topics from the BSD sector.

Kubernetes

Quickstart with Kubernetes and GKE (Part 2/2) 14
Leonardo Neves

This article discusses on how to deploy a simple Docker application on Google's Kubernetes Engine (GKE). At the end of the article, readers will be able to deploy any publicly available application on Docker Hub on GKE, taking advantage of many features on the platform, like high availability using several data-centers and unlimited scalability.

FreeBSD

Shadowsocks Proxy Server On FreeBSD 22
Abdorrahman Homaei

Shadowsocks is an open-source encrypted socks5 proxy server and client which is applicable to bypassing URL filtering or geographical limitations. It was created in 2012, and multiple implementations of the protocol have been made available since.

SmartOS

Introduction to MDB 26
Carlos Neira

Illumos comes out of the box with great observability and postmortem analysis tools. The modular debugger, commonly known as MDB, to some extent, has both capabilities since it can inspect a live kernel, a running process, a kernel crash image and a coredump.

OpenBSD

OpenBSD 6.3 36
Albert Hui

OpenBSD 6.3 was released on April 2, 2018. The 6.3 release comprised of numerous performance related enhancements and improvements pertaining to Meltdown/Spectre (variant 2) mitigations and VMM/VMD related updates.

Interview

Interview with Sanel Zukan 38
Founder & CEO of Hedron
The BSD Team

Expert Speak by E.G.Nadhan

5 Imperatives for Catalysts of Change 40 *E.G. Nadhan*

In his keynote address at the symposium, Gartner Executive Vice President and Analyst Peter Sondergaard had highlighted certain companies which scored high on the Gartner Digital IQ index – Great examples of enterprises that have treated change as a catalyst to play the game on digital terms.

Column

The doves and the hawks are gathering for a showdown, be it in geopolitics or the Internet. Facebook and Cambridge Analytica, the West, and Russia are all walking on a tightrope. Brinkmanship is the current name of the game. Who is going to come out on top? 44
Rob Somerville



Editor in Chief:

Ewa Dudzic
ewa@bsdmag.org
www.bsdmag.org

Contributing:

Sanel Zukan, Luca Ferrari, José B. Alós, Carlos Klop, Eduardo Lavaque, Jean-Baptiste Boric, Rafael Santiago, Andrey Ferriyan, Natalia Portillo, E.G Nadhan, Daniel Cialdella Converti, Vitaly Repin, Henrik Nyh, Renan Dias, Rob Somerville, Hubert Feyrer, Kalin Staykov, Manuel Daza, Abdorrahman Homaei, Amit Chugh, Mohamed Farag, Bob Cromwell, David Rodriguez, Carlos Antonio Neira Bustos, Antonio Francesco Gentile, Randy Ramirez, Vishal Lambe, Mikhail Zakharov, Pedro Giffuni, David Carlier, Albert Hui, Marcus Shmitt, Aryeh Friedman

Top Betatesters & Proofreaders:

Daniel Cialdella Converti, Eric De La Cruz Lugo, Daniel LaFlamme, Steven Wierckx, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Katherine Dizon, Natalie Fahey, and Mark VonFange.

Special Thanks:

Denise Ebery
Katherine Dizon

Senior Consultant/Publisher:

Paweł Marciniak

**Publisher: Hakin9 Media SK,
02-676 Warsaw, Poland Postepu 17D, Poland
worldwide publishing
editors@bsdmag.org**

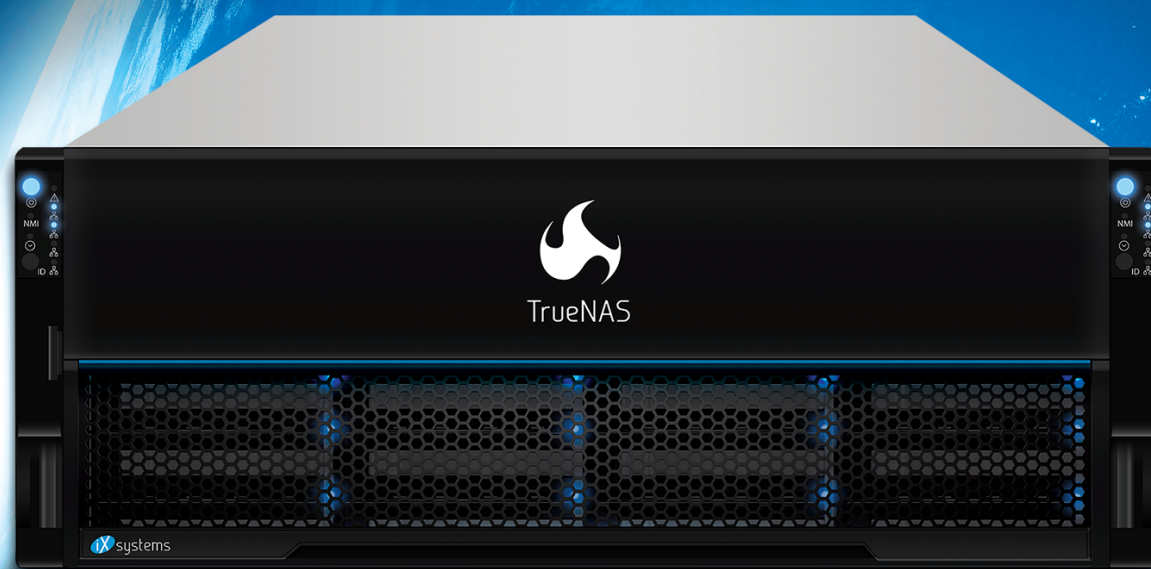
Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org

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

Groundbreaking TrueNAS® M Series



DISRUPTING STORAGE AT WARP SPEED



LOWEST TCO SHARED STORAGE

Intel® Xeon® Scalable Family Processors

RESILIENT

- Self-healing Data
- Continuous Operation
- Easy Replication

WARP FACTOR IX

- Faster than SSD-based Hybrid Storage Arrays
- Flash-Turbocharged Data Access

EXPANDABLE

- Scales to 10PB using HGST Drives
- 10Gb/s-100Gb/s per NAS
- Non-disruptive Upgrades

COMPATIBLE

- Citrix, Veeam, & VMware Certified
- Unifies File, Block, & S3 Data
- Supports Leading Cloud Providers

Visit ixsystems.com/TrueNAS or call (855) GREP-4-iX today!

In Brief

Nextcloud 13 on FreeBSD

It is worth visiting the [vermaden blog](#) and reading his full post. He shared a setup of Nextcloud 13 running on a FreeBSD system.

“To make things more interesting would be running inside a FreeBSD Jail. I will not describe the Nextcloud setup itself here as it’s large enough for several blog posts.



The official Nextcloud 13 documentation recommends the following setup:

MySQL/MariaDB

PHP 7.0 (or newer)

Apache 2.4 (with mod_php)

I prefer PostgreSQL database to MySQL/MariaDB, and also a fast and lean Nginx web server to Apache, so my setup is based on these components:

PostgreSQL 10.3

PHP 7.2.4

Nginx 1.12.2 (with php-fpm)

Memcached 1.5.7

The Memcached subsystem is least important, it can be easily changed into something more modern like Redis for example. I prefer not to use any third party tools for FreeBSD Jails management. Not because they are bad or something like that. There are just many choices for good FreeBSD Jails management and I want to provide a GENERIC example for Nextcloud 13 in a Jail, not for a specific management tool.”

Source: <https://vermaden.wordpress.com/2018/04/04/nextcloud-13-on-freebsd/>

TrueOS STABLE 18.03 Release by Ken Moore

The TrueOS team announced the availability of a new STABLE release of the TrueOS project (version 18.03). This is a special release due to the security issues impacting the computing world since the beginning of 2018.

“Important changes between version 17.12 and 18.03

“Meltdown” security fixes: This release contains all the fixes to FreeBSD which mitigate the security issues for systems that utilize Intel-based processors when running virtual machines such as FreeBSD jails. Please note that virtual machines or jails must also be updated to a version of FreeBSD or TrueOS which contains these security fixes.

“Spectre” security mitigations: This release contains all current mitigations from FreeBSD HEAD for the Spectre memory-isolation attacks (Variant 2). All 3rd-party packages for this release are also compiled with LLVM/Clang 6 (the “retpoline” mitigation strategy). This fixes many memory allocation issues and enforces stricter requirements for code completeness and memory usage within applications.

Unfortunately, some 3rd-party applications became unavailable as pre-compiled packages due to non-compliance with these updated standards. These applications are currently being fixed either by the upstream authors or the FreeBSD port maintainers. If there are any concerns about the availability of a critical application for a specific workflow, please search through the changelog of packages between TrueOS 17.12 and 18.03 to verify the status of the application.”



Source: <https://www.trueos.org/blog/trueos-stable-18-03-release/>

pfSense 2.4.3 Released

In this month, the release of pfSense® software version 2.4.3 was announced and it is now available for new installations and upgrades!



pfSense 2.4.3 is full of security patches, has several new features, includes support for new Netgate hardware models and stability fixes for issues from pfSense 2.4.x branch releases.

This release includes several important security patches:

Kernel PTI mitigations for Meltdown (optional tunable) [FreeBSD-SA-18:03.speculative_execution.asc](#)

*IBRS mitigation for Spectre V2 (requires updated CPU microcode)
[FreeBSD-SA-18:03.speculative_execution.asc](#)*

Fixes for FreeBSD-SA-18:01.ipsec

Fixed three potential XSS vectors, and two potential CSRF issues

CSRF protection for all dashboard widgets

Updated several base system packages to address CVEs

In addition to security fixes, pfSense software version 2.4.3 also includes important bug fixes.

Notable bug fixes in 2.4.3 include:

Fixed hangs due to Limiters and pfsync in High Availability configurations

Imported a netstat fix to improve performance and reduce CPU usage, especially on the Dashboard and ARM platforms

Fixed a memory leak in the pfSense PHP module

Fixed DHCPv6 lease display for entries that were not parsed properly from the lease database

Fixed issues on assign_interfaces.php with large numbers of interfaces

Fixed multiple issues that could result in an invalid ruleset being generated

Fixed multiple Captive Portal voucher synchronization issues with HA

Fixed issues with XMLRPC user account synchronization causing GUI inaccessibility on secondary HA nodes

... and many more!

Source: <https://www.netgate.com/blog/pfsense-2-4-3-release-now-available.html>

NomadBSD 1.0.1 Released

NomadBSD is a 64bit live system for USB flash drives, based on FreeBSD[®]. Together with automatic hardware detection and setup, it is configured to be used as a desktop system that works out of the box, but can also be used for data recovery.



This release includes several changes:

- Fix a problem with graphics driver detection.

- Fix a boot problem on *Lenovo® X220*.
- Disable the terminal bell.
- Add a rc script to automatically load the correct acpi module.
- Close/lock root shells on `ttyv{0,1,2}`.

Source: <http://nomadbsd.org/index.html>

iXsystems Unveils New TrueNAS M-Series Unified Storage Line

iXsystems, the leader in Enterprise Open Source servers and software-defined storage, announced the TrueNAS M40 and M50 as the newest high-performance models in its hybrid, unified storage product line.

The TrueNAS M-Series harnesses NVMe and NVDIMM to bring all-flash array performance to the award-winning TrueNAS hybrid arrays. It also includes the Intel® Xeon® Scalable Family of Processors and supports up to 100GbE and 32Gb Fibre Channel networking. Sitting between the all-flash TrueNAS Z50 and the hybrid TrueNAS X-Series in the product line, the TrueNAS M-Series delivers up to 10 Petabytes of highly-available and flash-powered network attached storage and rounds out a comprehensive product set that has a capacity and performance option for every storage budget.



Designed for On-Premises & Enterprise Cloud Environments

As a unified file, block, and object sharing solution, TrueNAS can meet the needs of file serving, backup, virtualization, media production, and private cloud users thanks to its support for the SMB, NFS, AFP, iSCSI, Fibre Channel, and S3 protocols.

At the heart of the TrueNAS M-Series is a custom 4U, dual-controller head unit that supports up to 24 3.5" drives and comes in two models, the M40 and M50, for maximum flexibility and scalability. The TrueNAS M40 uses NVDIMMs for write cache, SSDs for read cache, and up to two external 60-bay expansion shelves that unlock up to 2PB in capacity. The TrueNAS M50 uses NVDIMMs for write caching, NVMe drives for read caching, and up to twelve external 60-bay expansion shelves to scale upwards of 10PB. The dual-controller design provides high-availability failover and non-disruptive upgrades for mission-critical enterprise environments.

By design, the TrueNAS M-Series unleashes cutting-edge persistent memory technology for demanding performance and capacity workloads, enabling businesses to accelerate enterprise

applications and deploy enterprise private clouds that are twice the capacity of previous TrueNAS models. It also supports replication to the Amazon S3, BackBlaze B2, Google Cloud, and Microsoft Azure cloud platforms and can deliver an object store using the ubiquitous S3 object storage protocol at a fraction of the cost of the public cloud.

Fast

As a true enterprise storage platform, the TrueNAS M50 supports very demanding performance workloads with up to four active 100GbE ports, 3TB of RAM, 32GB of NVDIMM write cache and up to 15TB of NVMe flash read cache. The TrueNAS M40 and M50 include up to 24/7 and global next-business-day support, putting IT at ease. The modular and tool-less design of the M-Series allows for easy, non-disruptive servicing and upgrading by end-users and support technicians for guaranteed uptime. TrueNAS has US-Based support provided by the engineering team that developed it, offering the rapid response that every enterprise needs.

Award-Winning TrueNAS Features

Enterprise: Perfectly suited for private clouds and enterprise workloads such as file sharing, backups, M&E, surveillance, and hosting virtual machines.

Unified: Utilizes SMB, AFP, NFS for file storage, iSCSI, Fibre Channel and OpenStack Cinder for block storage, and S3-compatible APIs for object storage. Supports every common operating system, hypervisor, and application.

Economical: Deploys an enterprise private cloud and reduces storage TCO by 70% over AWS with built-in enterprise-class features such as in-line compression, deduplication, clones, and thin-provisioning.

Safe: The OpenZFS file system ensures data integrity with best-in-class replication and snapshotting. Customers can replicate data to the rest of the iXsystems storage lineup and to the public cloud.

Reliable: High availability option with dual hot-swappable controllers for continuous data availability and 99.999% uptime.

Familiar: Provisions and manages storage with the same simple and powerful WebUI and REST APIs used in all iXsystems storage products, as well as iXsystems' [FreeNAS](#) software.

Certified: TrueNAS has passed the Citrix Ready, VMware Ready, and Veeam Ready certifications, reducing the risk of deploying a virtualized infrastructure.

Open: By using industry-standard sharing protocols, the OpenZFS Open Source enterprise file system and FreeNAS, the world's #1 Open Source storage operating system (and also engineered by iXsystems), TrueNAS is the most open enterprise storage solution on the market.

Availability

The TrueNAS M40 and M50 will be generally available in April 2018 through the iXsystems global channel partner network. The TrueNAS M-Series starts at under \$20,000 USD and can be easily expanded using a linear “per terabyte” pricing model. With typical compression, a Petabyte can be

stored for under \$100,000 USD. TrueNAS comes with an all-inclusive software suite that provides NFS, Windows SMB, iSCSI, snapshots, clones and replication.

Source: <https://www.ixsystems.com/blog/truenas-m-series/>

TrueNAS 11.1 – What’s New

TrueNAS Software Update Delivers Compelling ZFS Improvements, Better Resilver Tools, and Cloud Sync Additions

TrueNAS software version 11.1 provides ZFS improvements and expanded integration with cloud services. In addition to Amazon S3, TrueNAS Cloud Service Integration supports Microsoft Azure, Backblaze B2 Cloud, and Google Cloud Platform, making it easier than ever to use TrueNAS for all of your cloud storage needs.

TrueNAS 11.1 includes improvements for handling multiple snapshots and large files. The new Resilver Priority tab allows the administrator to schedule specific dates and times for resilvering drives, and mitigates the challenges and risks associated with storage array rebuilds on high capacity drives. TrueNAS 11.1 introduces built-in optimizations that greatly reduce the time required to perform a scrub or resilver on pools with a large percentage of their space in use. Scrubs can also now be paused and resumed from the command line. Once resumed, the scrub continues from where it left off.



“The integration of TrueNAS with Backblaze B2 Cloud Services is ideal for our needs. The use of Cloud Sync gives us an easy to use and cost effective off-site disaster recovery solution.” – Aaron Echols, Systems Administrator at Benjamin Franklin Charter School

Benjamin Franklin Charter School (BFCS) deployed TrueNAS and TrueRack to replace an aging and poorly performing IT infrastructure. With the new updates to TrueNAS cloud service integration included in TrueNAS 11.1, BFCS can now quickly and easily recover data, as well as supplement the data storage capacity of their TrueNAS Storage Appliances. Read more about why BFCS chose TrueNAS and TrueRack in this [case study](#).

TrueNAS software updates are available through the updater included in the TrueNAS web GUI. The update will show as TrueNAS 11.1-U4. The update also includes the fixes for [CVE-2018-1050](#) and [CVE-2018-1057](#). For more information on the update, please check out our [TrueNAS 11.1-U4 release notes](#).

Source: <https://www.ixsystems.com/blog/truenas-11-1-whats-new/>

Quickstart with Kubernetes and GKE (Part 2/2)

This article will discuss how to deploy a simple Docker application on Google's Kubernetes Engine (GKE). Readers will be able to deploy any publicly available application on Docker Hub on GKE, taking advantage of many features of the platform, like high availability using several data-centers and unlimited scalability.

What you will learn...

How to get started with Kubernetes quickly

How to get started with GKE quickly

How to deploy a simple Docker application on Google Kubernetes Engine

What you should know...

Basic understanding of Linux and Linux commands

Basic understanding of Docker

Introduction

We covered many concepts about Docker, Kubernetes, and GKE in the first part of this article and also created a simple high-availability

environment on GKE using two hosts. In the second part, we will explain the deployment of a simple container in the environment created previously. This article will also explain more

about *kubectl* and introduce some of its basic and useful sub-commands.

With both parts of this article you will be able to run any simple application available on Docker Hub using Docker, Kubernetes and GKE. This small application in our environment will have almost the same level of high-availability as other mature applications from big companies running on Kubernetes/GKE. For someone starting out with Kubernetes this environment can be very useful for testing until they get used to the commands and become qualified to manage a critical environment.

Current Environment

After following the first part of the article you already have a Kubernetes cluster running two nodes. Each node is running in a different zone (data-center), but both are in the same region (metropolitan area). There is nothing running on top of it, so our cluster is still useless.

Containers and pods

Kubernetes groups Docker containers into pods. Even when you intend to run a single container, Kubernetes will run a pod with the container inside it. The advantage of using pods is that the containers inside it can communicate using the *localhost* interface, which is quite convenient and fast. A pod is indivisible, therefore all containers in the same pod will always run on the same node. The relation between pods, containers and nodes is shown in Figure 1:

Kubernetes Pods

collections of containers that are co-scheduled

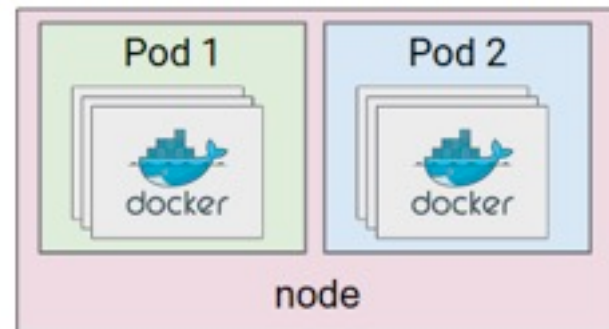


Figure 1: Relation between pods, containers and nodes
(Source:

<https://1ambda.github.io/infrastructure/container/kubernetes-intro/>)

Why is Google Cloud Shell the preferred management tool?

Besides being able to manage the cluster using just the web user interface, it's recommended that you learn and use the Google Cloud Shell and/or install the *kubectl* tool on your desktop. Kubectl is the fastest way to manage the cluster and has comprehensive functionality. To be able to manage a production cluster you must learn the major subcommands of *kubectl*. This article will cover the Google Cloud Shell and the kubectl tool because they are the most important ways to manage a Kubernetes cluster.

Opening the Google Cloud Shell

Go to <https://console.cloud.google.com/kubernetes> and click on the cluster created in the first part of the article. Next, click on 'Run in Cloud Shell' and a gcloud command will be shown. This command will properly configure the kubectl command to manage your cluster. Just hit enter and you will get access to the shell. Now you are able to type any valid gcloud or kubectl command and fully manage both GKE and Kubernetes.

Running your first application

For those familiar with Docker, *kubectl* usually has an equivalent command to most of the Docker commands. For instance, we can run the following command to start a nginx container:

```
# docker run -d -p 80:80 nginx
```

To create a pod with a nginx container, you can run the following command:

```
$ kubectl run --image=nginx nginx-app --port=80
```

Despite looking pretty similar, the *kubectl* command does a lot more. ‘docker run’ just starts a container while *kubectl run* is creating deployments, replica sets and pods with the nginx Docker container inside it. In other words, *kubectl* is creating our Docker container ‘cluster aware’.

Checking and deleting the pod

After creating our first nginx container/pod using the command ‘*kubectl run --image=nginx nginx-app --port=80*’, we can check if it is really running using the following command:

```
$ kubectl get pods
```

Figure 2 shows the expected result of these commands.

Let’s try to delete our recently created pod by running the command ‘*kubectl delete pod “name of the pod”*’ and see what happens:

In my example I got the results showed on Figure 3.

As you can see in Figure 3, after we ran *kubectl delete pod*, Kubernetes started a new pod to replace the deleted one. When we ran *kubectl run* the first time, we instructed Kubernetes to create and keep the state with

```
leonardo_neves@myfirstproject-197621:~$ kubectl run --image=nginx nginx-app --port=80
deployment "nginx-app" created
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-app-7b679f697-fbd6t          1/1     Running   0           3s
leonardo_neves@myfirstproject-197621:~$
```

Figure 2: *kubectl run* and *kubectl get pods*

```
leonardo_neves@myfirstproject-197621:~$ kubectl delete pod nginx-app-7b679f697-fbd6t
pod "nginx-app-7b679f697-fbd6t" deleted
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-app-7b679f697-fbd6t          0/1     Terminating   0           17m
nginx-app-7b679f697-v27xr          1/1     Running        0           2s
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-app-7b679f697-v27xr          1/1     Running        0           10s
leonardo_neves@myfirstproject-197621:~$
```

Figure 3: *Deleting and checking pods*

one nginx pod/container running. When we deleted the pod, Kubernetes re-creates it in order to keep the current state consistent with the desired state.

To effectively delete our pod, we need to delete the deployment using the command 'kubectl delete deployment "name of the deployment"', as you can see on Figure 4.

Running multiple pods

Rather than running a single pod, let's now run 2 pods adding '--replicas' in our `kubectl run` command, as shown in Figure 5.

Adding '-o wide' to the 'kubectl get pods' command you can see where each pod is running (Figure 6).

```
leonardo_neves@myfirstproject-197621:~$ kubectl get deployment
NAME          DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-app     1         1         1             1           2h
leonardo_neves@myfirstproject-197621:~$ kubectl delete deployment nginx-app
deployment "nginx-app" deleted
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-app-7b679f697-v27xr          0/1     Terminating   0          1h
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
No resources found.
leonardo_neves@myfirstproject-197621:~$ kubectl get deployment
No resources found.
leonardo_neves@myfirstproject-197621:~$
```

Figure 4: Deleting deployment and pod

```
leonardo_neves@myfirstproject-197621:~$ kubectl run --image=nginx nginx-app --port=80 --replicas=2
deployment "nginx-app" created
leonardo_neves@myfirstproject-197621:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-app-7b679f697-cntkr          1/1     Running   0           13s
nginx-app-7b679f697-tjlds          1/1     Running   0           13s
leonardo_neves@myfirstproject-197621:~$
```

Figure 5: Running multiple pods

```
leonardo_neves@myfirstproject-197621:~$ kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP           NODE
nginx-app-7b679f697-cntkr          1/1     Running   0           4m    10.8.0.8     gke-myfirstcluster-default-pool-8300f27d-tfs9
nginx-app-7b679f697-tjlds          1/1     Running   0           4m    10.8.1.15    gke-myfirstcluster-default-pool-22biafaa-mcb7
leonardo_neves@myfirstproject-197621:~$
```

Figure 6: Kubectl get pods wide

Please note that the pods are running in different nodes (NODE column). By default, Kubernetes will try to spread the pods across the maximum number of nodes. It does that in order to increase the availability of the cluster. More nodes running pods mean less impact to the services when a node goes down.

Considerations about high-availability and unlimited scalability

Now we finally have a truly highly-available cluster. If the application/pod/container breaks, Kubernetes can terminate it and recreate a new one to replace it. To do that, Kubernetes needs to be configured to do health checks on the pod. To monitor a pod running a nginx container we can configure Kubernetes to monitor a URL so Kubernetes will delete/recreate the pod in case of 5xx return codes or timeouts.

In case of a node down or even an entire Google data-center down, Kubernetes will start new

pods in the remaining nodes to reach the desired state.

We can also freely increase/decrease pods/containers/nodes without any outage. Google Compute provides virtually infinite scalability so your very small application can grow as much as needed.

Kubernetes Dashboard

Kubernetes has a dashboard that’s not deployed by default. You can fully manage the environment using just the dashboard, but as a good Unix/BSD/Linux fan I guess that you will enjoy using a shell console more. An interesting feature of the Kubernetes dashboard is that you can see the commands that you run in yaml (.yaml) format. This is a fast way to generate the yaml file without fully understanding the details of each command. The Kubernetes dashboard is shown in Figure 7.

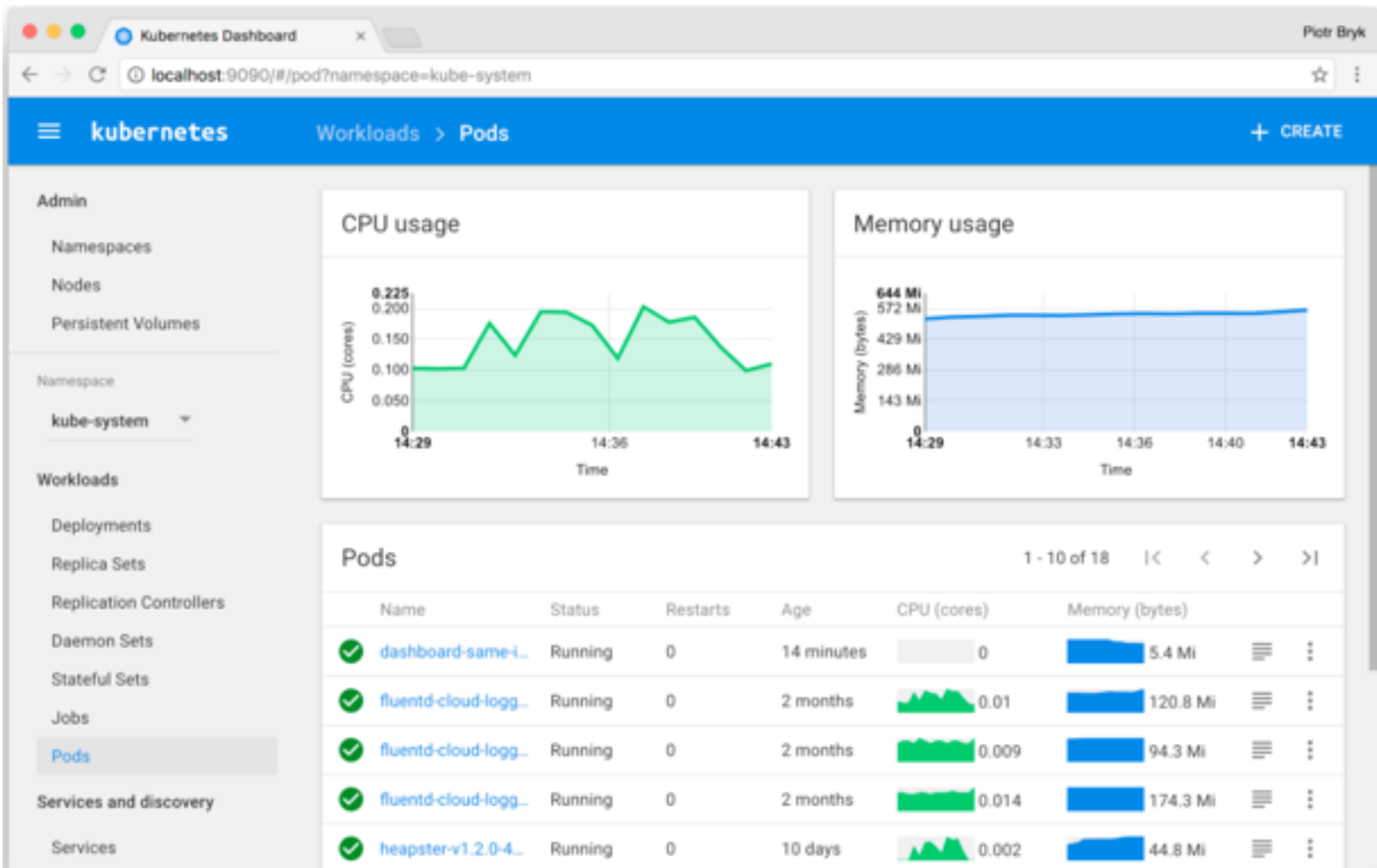


Figure 7: Kubernetes Dashboard

Yaml files

The most appropriate way to manage a Kubernetes cluster is by using yaml files (.yaml). Using yaml files is very convenient because you can store the files in a version control system like git and have all the history of changes there.

After creating the file, you just have to run 'kubectl apply -f "file"' (or kubectl create -f "file"). Figure 8 shows an example of an yaml file used to create an nginx pod:

```
---
apiVersion: v1
kind: Pod
metadata:
  name: site
spec:
  containers:
  - name: webserver
    image: nginx
    ports:
    - containerPort: 80
```

Figure 8: Nginx pod yaml file

Configuring a new administration console

The embedded Bash console in GKE is very useful for running simple commands like `kubectl get pods`, but it's not the most appropriate way to manage a big environment. You can install both `gcloud` and `kubectl` commands in your PC. Another good option is to create a VM on Google Cloud Compute and install the tools there so you can manage the environment from anywhere by just SSH'ing to this box.

Other resources

Kubernetes has many other resources like replica sets, deployments, and replication controllers. To properly manage critical or big environment it's required to understand the basics of these resources. Another important point required in order to access the pods from

outside is exposing ports. Exposing ports can create a service that will work like an internal load balancer. Figure 9 shows an environment with 3 pods running on two different nodes. The service was created to expose the port and an IP address was assigned to it. All requests will use the service IP and no direct connections to the pods will be allowed.

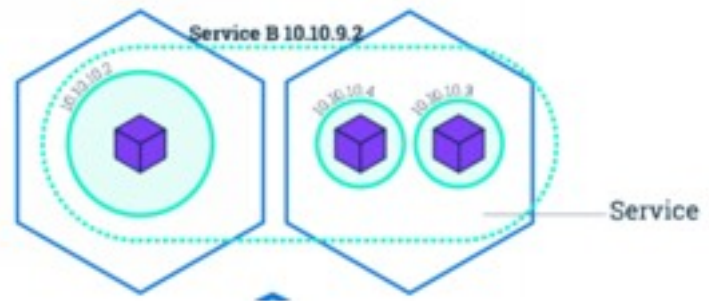


Figure 9: Service with exposed IP address/port

Namespaces

As explained in the first part of this article, we can create namespaces to isolate all resources from other namespaces. By default, Kubernetes uses the namespace 'default', but it's a good practice to create new namespaces like Dev, QA, Prod, and so on. It's important to note that one namespace can affect the performance of others namespaces. Therefore, if the environment is critical, please consider creating totally isolated Kubernetes clusters rather than just namespaces.

Volumes

Another important resource in Kubernetes are volumes. A volume is similar to a disk which can be shared between containers. Volumes can also be ephemeral or persistent.

Conclusion

In this article you have learned many concepts of Docker, Kubernetes and GKE and have created a simple and fully operational environment to play around with them. You have also learned how to deploy a single application from Docker Hub (nginx). After that the article discussed a bit about the high-availability of the cluster. In addition, some information about additional features of Kubernetes has been shown. Knowing a little about these features can help you focus on good paths to further learning.

In conclusion, the author hopes that this article was useful to someone who is starting to learn Kubernetes and GKE. There is nothing better than hands-on experience to really understand technology and this article tried to help you with creating your environment to get started. The path to supporting a critical environment with Kubernetes is long and here we attempted to guide you on your first steps.

Links

<https://kubernetes.io/>

<https://cloud.google.com/kubernetes-engine/docs/>

<https://courses.edx.org/courses/course-v1:LinuxFoundationX+LFS158x+2T2017/course/>

<https://docs.docker.com/get-started/>

<https://www.youtube.com/watch?v=H-FKBoWTVws>

Meet the Author



Leonardo Neves Bernardo got started with Unix in 1996 and since then he is always working with some related technology, especially using Linux systems. He holds many certifications including LPIC-3, LPIC-300, LPIC-302 and LPIC-303, RHCSA and the ITILv3 Foundation. He is from Florianópolis, Brazil, but currently lives in Toronto, Canada, where he is the Security Admin of VerticalScope Inc. His LinkedIn profile is <https://www.linkedin.com/in/leonardoneves>

WORKSHOP

DEVICE DRIVER DEVELOPMENT FOR BSD



Join Today

www.bsdmag.org

Shadowsocks Proxy Server On FreeBSD

[What Is The Shadowsocks?](#)

[What Is The Shadowsocks-libdev?](#)

[Shadowsocks VS SSH-Tunnel VS VPN](#)

[How to Install and Run Shadowsocks On FreeBSD?](#)

[Connect To Shadowsocks Server From FreeBSD Terminal](#)

[Shadowsocks-libdev Configurations](#)

What is the Shadowsocks?

Shadowsocks is an open-source encrypted socks5 proxy server and client, which is applicable to bypassing URL filtering or geographical limitations. It was created in 2012 and multiple implementations of the protocol have been made available since.

What is the Shadowsocks-libdev?

Shadowsocks-libev is a lightweight and secure socks5 proxy. It is a port of the original shadowsocks. Shadowsocks-libev is written in pure C and takes advantage of libev to achieve both high performance and low resource consumption. Shadowsocks-libev consists of five components. One is ss-server that runs on a

remote server to provide secured tunnel service. ss-local and ss-redir are clients on your local machines to proxy traffic(TCP/UDP or both). ss-tunnel is a tool for local port forwarding. While ss-local works as a standard socks5 proxy, ss-redir works as a transparent proxy and requires Netfilter's NAT module.

ss-manager is a controller for multi-user management and traffic statistics, uses UNIX domain socket to talk to with the ss-server. Also, it provides a UNIX domain socket or IP based API for other software.

Tip: ss-redir is not available on FreeBSD.

Shadowsocks Vs. SSH-Tunnel Vs. VPN

Unlike an early SSH tunnel, shadowsocks can also proxy UDP traffic. The latest SSH can handle UDP as well by creating layer 2 or layer 3 tunnels. This creates tun (layer 3) or tap (layer 2) virtual interfaces on both ends of the connection which allows you to route all the traffic inside the tunnel and brings you more security. Layer 2 SSH tunnel acts as a VPN. VPN or virtual private network is relatively old technology and needs more configuration on both sides.

Here is a comparison between the two:

- Shadowsocks connection is faster than VPN and SSH-Tunnel(Layer 2 and 3)
- SSH-Tunnel (layer 2 and 3) is more secure than shadowsocks and VPN.
- SSH-Tunnel setup is easier than VPN and shadowsocks.

See Table 1.

How to Install and Run Shadowsocks on FreeBSD?

Shadowsocks client and server are cross-platform. Since it's easier to run them on Windows with just a few clicks, let's cover how we can run them on FreeBSD.

Install shadowsocks with PKG

To install Shadowsocks-libdev issue this command:

```
# pkg install shadowsocks-libev
```

To run your FreeBSD server, issue the following command:

	Security	Connection Speed	Setup Easiness
Shadowsocks	Medium	High	Medium
SSH-Tunnel	Medium	Medium	High
SSH-Tunnel(L2, L3)	High	low	High
VPN	High	low	Low

Table 1. The comparison

```
# ss-server -s "your server valid ip" -p
1080 -k "password" -m aes-256-cfb -a
nobody -u &

-s: host name or IP address of your remote
server

-p: port number of your remote server

-k: password of your remote server

-m: encryption method
```

There are other ciphers you can use with -m :

```
aes-128-gcm, aes-192-gcm, aes-256-gcm,
aes-128-cfb, aes-192-cfb, aes-256-cfb,
aes-128-ctr, aes-192-ctr, aes-256-ctr,
camellia-128-cfb, camellia-192-cfb,
camellia-256-cfb, bf-cfb,
chacha20-ietf-poly1305,
xchacha20-ietf-poly1305, salsa20, chacha20
and chacha20-ietf. The default cipher is
rc4-md5.
```

Tip: Encryption on Both sides must be same.

-a: run as another user

-u: enable UDP relay

Installing shadowsocks with PIP

If you encountered some errors, you can also use PIP application. PIP is designed for installing and managing Python packages.

```
# pkg install py27-pip

# pip install shadowsocks

# ssserver -p 1080 -k "password" -m
aes-256-cfb --user nobody -d start
```

You can stop this service by:

```
# ssserver -d stop
```

Connecting to Shadowsocks Server From the FreeBSD Terminal

As we mentioned earlier shadowsocks client also supported on Windows.

First, you need to install Shadowsocks-libdev on your client:

```
# pkg install shadowsocks-libev
```

On your FreeBSD client issue this command:

```
# ss-local -s "your server valid IP" -p
1080 -l 9090 -m aes-256-cfb -k "password"
```

Shadowsocks will listen on port 9090, then you set this port on your browser or any other application that supports socks5.

Shadowsocks-libdev Configurations

If you want to run shadowsocks _libev easily at boot time, it's better to set arguments in a config file.

The config file is placed at:

```
/usr/local/etc/shadowsocks-libev/config.json
```

Open it with ee and:

```
#
/usr/local/etc/shadowsocks-libev/config.js
on

{

    "server": "127.0.0.1",

    "server_port": 8388,

    "local_port": 1080,

    "password": "barfoo!",

    "timeout": 60,

    "method": "chacha20-ietf-poly1305"

}
```

you can change it as per your needs then save the file.

You can also find details about this option by issuing this command:


```
# man shadowsocks-libev
```

Then add shadowsocks-libev to boot services:

```
# sysrc shadowsocks_libev_enable="YES"
```

and start the shadowsocks service:

```
# service shadowsocks_libev start
```

Conclusion

Running shadowsocks proxy server on FreeBSD is such a brilliant idea. The point is, FreeBSD and shadowsocks-libev, are lightweight and secure, and as a result, we will have a reliable and cost-effective socks5 proxy server.

Useful Links

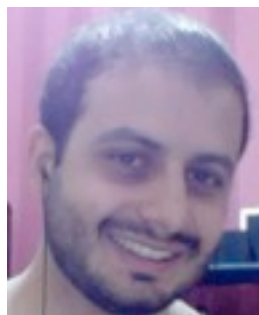
<https://shadowsocks.org/en/download/clients.html>

https://en.wikipedia.org/wiki/Tunneling_protocol

<https://github.com/shadowsocks>

Meet the Author

Abdorrahman Homaei has been working as a software developer since 2000. He has used FreeBSD for more than ten years. He became involved with the meetBSD dot ir and performed serious training on FreeBSD. He also started his own company (etesal amne sara Tehran) in Feb, 2017 that is based in Iran Silicon Valley. Full CV: <http://in4bsd.com>



His company: <http://corebox.ir>

BSD Certification

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>

Introduction to MDB

Illumos comes out of the box with great observability and postmortem analysis tools. The modular debugger, commonly known as MDB, to some extent, has both capabilities since it can inspect a live kernel, a running process, a kernel crash image, and a coredump.

What you will learn...

- The basic usage of MDB to debug programs and coredumps.
- How to use MDB to debug a live process

What you should know...

- Familiarity with the C-programming language.
- SmartOS familiarity.

What you will need...

- The latest version of SmartOS.

Invoking MDB

One can invoke MDB on a core file, a live process or in a live kernel.

```
$ mdb core
```



```

.rw----- 1 root    root    11568256 Apr 12 17:54 core.ld-2.17.so.28414
.rw----- 1 root    root    11568256 Apr 12 17:56 core.ld-2.17.so.28467
.rw----- 1 root    root    3923692 Apr 13 12:41 core.systemd-cgroups-.87470
.rw----- 1 root    root    3923692 Apr 13 12:41 core.systemd-cgroups-.88204
.rw----- 1 root    root    3923692 Apr 13 12:41 core.systemd-cgroups-.88667
.rw----- 1 root    root    3923692 Apr 13 12:42 core.systemd-cgroups-.91211
.rw----- 1 root    root    3923692 Apr 13 12:44 core.systemd-cgroups-.98481
.rw----- 1 root    root    102889324 Apr 15 19:54 core.nwserver.49123
[root@krondor /zones/26c4e9b2-3885-432d-a8de-c5be7215e348/cores]# mdb core.nwserver.49123
mdb: core file data for mapping at fead8000 not saved: Bad address
.loading modules: [ libc.so.1 ld.so.1 libc.so.6 ]
* █

```

\$ mdb -p <pid>

```

3082 root    7400K 3160K sleep 59 0 0:00:49 0.0% ntpd/2
1488 root    2960K 1276K sleep 59 0 0:00:00 0.0% dhcpagent/1
3034 root     59M  27M sleep 59 0 0:00:56 0.0% metadata/7
2931 root    4008K 592K sleep 59 0 0:00:23 0.0% ipmon/1
11070 root      0K   0K sleep 60 - 0:00:00 0.0% zsched/1
2998 daemon  4444K 972K sleep 29 0 0:00:00 0.0% rpcbind/3
 247 root     14M 2256K sleep 29 0 0:00:05 0.0% syseventd/22
 277 root    5832K 2056K sleep 29 0 0:00:12 0.0% devfsadm/8
3063 root    2080K 996K sleep 1 0 0:00:00 0.0% cron/1
3138 smmsp    7060K 1352K sleep 59 0 0:00:00 0.0% sendmail/1
3233 root    2004K 1112K sleep 59 0 0:00:00 0.0% ttymon/1
3095 root    1720K 876K sleep 29 0 0:00:00 0.0% utmpd/1
2897 root    3908K 1440K sleep 29 0 0:00:00 0.0% picld/4
  21 root    2876K 768K sleep 29 0 0:00:00 0.0% dlmgmt/5
2929 root    2436K 996K sleep 1 0 0:00:00 0.0% svc.ipfd/1
  18 netadm   3644K 476K sleep 29 0 0:00:00 0.0% ipmgmt/3
 217 root    2284K 660K sleep 29 0 0:00:00 0.0% powerd/4
  58 root    2704K 860K sleep 29 0 0:00:00 0.0% pfexecd/3
3044 root    9536K 2956K sleep 29 0 0:00:39 0.0% nscd/35
  10 root     11M 2464K sleep 29 0 0:00:18 0.0% svc.configd/18
total: 143 processes, 670 lwps, load averages: 0.13, 0.12, 0.14
[root@krondor /zones/26c4e9b2-3885-432d-a8de-c5be7215e348/cores]# mdb -p 3082
.loading modules: [ libc.so.1 ]
* $c
libc.so.1`__sigsuspend+0x15(80478c0, e, 80478c0, 0)
wait_for_signal+0x4c(1, 4, 19, 8047908, 8040200, fe9f2000)
itpdmmain+0xd1a(0, 8047e00, 8047d98, 80e94d0)
main+0x1b(8047d8c, fe9ff348, 8047dc8, 806c5b8, 4, 8047df0)
start crt+0x97(4. 8047df0. fefd1f10. 0. 0. 0)

```

\$ mdb -k

```

[root@krondor /zones/26c4e9b2-3885-432d-a8de-c5be7215e348/cores]# mdb -k
Loading modules: [ unix genunix specfs dtrace mac cpu.generic uppc apix scsi_vhci ufs ip hook neti s
ockfs arp usba xhci fctl stmf_sbd stmf zfs mm sd lofs idm sata crypto fcp random cpc logindmux ptm k
vm spps nsmb smbsrv nfs vmm ipc ]
> ::status
debugging live kernel (64-bit) on krondor
operating system: 5.11 joyent_20180315T080815Z (i86pc)
image uuid:

```

MDB command Syntax

The Language syntax in MDB is designed around the concept of operating on the resulting value of an expression, which is typically a memory address. The basic form is expressed as a value followed by a command.

[value] [,count] command


```
> ::help
```

Each debugger command in mdb is structured as follows:

```

      [ address [, count]] verb [ arguments ... ]
      ^           ^           ^           ^
the start --+ |           |           |           |
address can be an |           |           |           |
expression         |           |           |           |
                   |           |           |           |
the repeat count --+ |           |           |           |
is also an expression |           |           |           |
                     +-----+ the verb is a name which begins
                               with either $, :, or ::. it can also
                               be a format specifier (/ \ ? or =)
```

For example:

```
> 0x08046a48,100/nap
```

Which means repeat 100 times from start address **0x08046a48** the format specified nap (n = newline, a = dot as symbol + offset, p = symbol 4 bytes). More format specifiers are available if you type ‘::formats’.

```
> ::formats
```

```
+ - increment dot by the count (variable size)
- - decrement dot by the count (variable size)
B - hexadecimal int (1 byte)
C - character using C character notation (1 byte)
D - decimal signed int (4 bytes)
E - decimal unsigned long long (8 bytes)
F - double (8 bytes)
G - octal unsigned long long (8 bytes)
H - swap bytes and shorts (4 bytes)
I - address and disassembled instruction (variable size)
J - hexadecimal long long (8 bytes)
K - hexadecimal uintptr_t (4 bytes)
L - match int (4 bytes)
M - match long long (8 bytes)
N - newline
O - octal unsigned int (4 bytes)
P - symbol (4 bytes)
Q - octal signed int (4 bytes)
R - binary unsigned long long (8 bytes)
S - string using C string notation (variable size)
T - horizontal tab
U - decimal unsigned int (4 bytes)
V - decimal unsigned int (1 byte)
W - write default radix unsigned int (4 bytes)
X - hexadecimal int (4 bytes)
Y - decoded time32_t (4 bytes)
Z - write hexadecimal long long (8 bytes)
^ - decrement dot by increment * count (variable size)
>> More [<space>, <cr>, q, n, c, a] ?
```

```

0x8046a48:
0x8046a48:      0x208
0x8046a4c:      1
0x8046a50:      0
0x8046a54:      0x8046b20
0x8046a58:      0x39
0x8046a5c:      0xfec406c0
0x8046a60:      0xffffffff
0x8046a64:      0x51
0x8046a68:      0xfef79fbc
0x8046a6c:      0xfecd0ac8
0x8046a70:      ld-linux.so.2`_rtld_global+0x51c
0x8046a74:      ld-linux.so.2`dl_main+0x2668
0x8046a78:      0
0x8046a7c:      0
0x8046a80:      0xf
0x8046a84:      0
0x8046a88:      0x6e2f0031
0x8046a8c:      0x636f0031
0x8046a90:      0
0x8046a94:      0
0x8046a98:      0
0x8046a9c:      0xfefd10a8
0x8046aa0:      0xfef400a8
0x8046aa4:      0xfef400a8
0x8046aa8:      0x8046ac8
0x8046aac:      LMfd`ld.so.1`pnavl_compare+0x2f
0x8046ab0:      0x8046b40
>> More [<space>, <cr>, q, n, c, a] ? █

```

Debugging a coredump using MDB

To use MDB in a real example, we will debug a coredump using MDB debugger commands (dcmds).

First, create a SmartOS vm using this json file, save it as b01.json, modify it if you need to, but the most important attribute in this is the image being used.

```

{
  "brand": "joyent",
  "fs_allowed": "ufs,pcfs,tmpfs",
  "image_uuid": "e69a0918-055d-11e5-8912-e3ceb6df4cf8",
  "alias": "build01",
  "hostname": "b01",
  "max_physical_memory": 8024,
  "quota": 70,
  "resolvers": ["8.8.8.8", "8.8.8.4"],
  "nics": [
    {

```

```

"nic_tag": "admin",

"ips": ["dhcp"],

"primary": true

} ]

}

```

Then, create the vm as usual with `vmadm`. Save the generated UUID for the zone, we'll need it later.

```
$ vmadm create -f b01.json
```

Now with our zone ready for development, login using `ZLOGIN(1)` and your zones's UUID.

```
$ zlogin b340284d-2051-e694-b81f-9c36168c1d84
```

```

> ::status
debugging core file of err (64-bit) from b01
initial argv: ./err
threading model: native threads
status: process terminated by SIGSEGV (Segmentation Fault), addr=0
~ █

```

We will use this sample C-program, name it `err.c`

```

> $C
fffffc7fffdffce0 libc.so.1`memset+0x32c()
fffffc7fffdffd10 _start_crt+0x83()
fffffc7fffdffd20 _start+0x18()
> █

```

Compile it with:

```
$ cc -m64 -O0 err.c -o err
```

If you execute `err`, you will see the following message:

```
Memory fault(coredump)
```

At this point, we can finally inspect the coredump using `mdb`. Logout of your zone and go to the following directory where all the coredumps for that region are stored:

```
$ cd /zones/b340284d-2051-e694-b81f-9c36168c1d84/cores
```

```
$ mdb core.err.48834
```

Let's check what happened.

As expected, `SIGSEGV` on address 0 (we tried to write on a NULL pointer)

Next, let's check the stack to see which was the last function executed.

We could also check the last executed instruction by inspecting the rip register. \$r will give us the values of the cpu registers.

```
> $r
%rax = 0x0000000000000000      %r8  = 0x0000000000000008
%rbx = 0xffffffff7fffdffd38    %r9  = 0x0101010101010101
%rcx = 0x000000000000002dc     %r10 = 0x00000000000001f80
%rdx = 0x0000000000000000      %r11 = 0xffffffff7fef20e06c
%rsi = 0x0000000000000000      %r12 = 0x0000000000000001
%rdi = 0x0000000000000008      %r13 = 0x00000000000411820
                                   %r14 = 0x0000000000000000
                                   %r15 = 0x0000000000000000

%cs = 0x0053      %fs = 0x0000      %gs = 0x0000
%ds = 0x0000      %es = 0x0000      %ss = 0x004b

%rip = 0xffffffff7fef20e06c libc.so.1`memset+0x32c
%rbp = 0xffffffff7fffdffce0
%rsp = 0xffffffff7fffdffcc8

%rflags = 0x00010202
  id=0 vip=0 vif=0 ac=0 vm=0 rf=1 nt=0 iopl=0x0
  status=<of,df,IF,tf,sf,zf,af,pf,cf>

%gsbase = 0x0000000000000000
%fsbase = 0xffffffff7fef162a40
%trapno = 0xe
  %err = 0x6
> █
```

Now that we know the address of the last instruction executed, we can see the assembler code for that.

```
> 0xffffffff7fef20e06c::dis
libc.so.1`memset+0x32c:      movq    %rdx, -0x8(%rdi)
libc.so.1`memset+0x330:      ret
libc.so.1`memset+0x331:      nopl    0x0(%rax,%rax)
libc.so.1`memset+0x336:      nopw    %cs:0x0(%rax,%rax)
libc.so.1`memset+0x340:      movq    %rdx, -0x8a(%rdi)
libc.so.1`memset+0x347:      movq    %rdx, -0x82(%rdi)
libc.so.1`memset+0x34e:      nop
libc.so.1`memset+0x350:      movq    %rdx, -0x7a(%rdi)
libc.so.1`memset+0x354:      movq    %rdx, -0x72(%rdi)
libc.so.1`memset+0x358:      movq    %rdx, -0x6a(%rdi)
libc.so.1`memset+0x35c:      movq    %rdx, -0x62(%rdi)
```

Here, we see the instruction that caused the SIGSEGV highlighted. Additionally, we see that the user tried to copy to the memory address in register rdx which was 0, case closed.

Other type of information that we could gather is with the help of walkers. Walkers, as the name implies, lets you “walk” structures. To check the available walkers, ::walkers dcmd is used.

```

> ::walk ulwp
0xffffffff7fef162a40
> ::walk ulwp | ::print ulwp_t ul_uberdata
ul_uberdata = libc.so.1`_uberdata
> ::walk ulwp | ::print ulwp_t
{
    ul_self = 0xffffffff7fef162a40
    ul_dinstr = [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ... ]
    ul_uberdata = libc.so.1`_uberdata
    ul_tls = {
        tls_data = 0
        tls_size = 0
    }
    ul_forw = 0xffffffff7fef162a40
    ul_back = 0xffffffff7fef162a40
    ul_next = 0
    ul_hash = 0
    ul_rval = 0
    ul_stk = 0xffffffff7fff400000
    ul_mapsiz = 0
    ul_guardsize = 0
    ul_stktop = 0xffffffff7fffe00000

> ::walkers
Rt_maps           - Walk a List of Rt_maps
oldcontext        - walk per-lwp oldcontext pointers
thread            - walk list of valid thread identifiers
ucontext          - walk ucontext_t uc_link list
ulwp              - walk list of ulwp_t pointers
ulwps             - walk list of ulwp_t pointers
> █

```

For example:

```
::walk ulwp
```

will return the address for the ulwp.

And,

```
::walk ulwp | ::print ulwp_t ul_uberdata
```

Will take the result of ::walk ulwp and take it as a ulwp_t data type and print the member ul_uberdata from that struct. It's the same concept of unix pipes.

Debugging a running process with MDB

To attach the debugger to a running process, we just need the pid of the process which we are interested in.

```

cnb@b01:~$ mdb -p $(pgrep vim)
> $c
ufs.36.64.4193`__pollsys+0x15(80436c0, 1, 0, 0, 0, c0)
ufs.36.64.4193`pselect+0x232(1, 80457c8, feebc060, 80437c8, 0, 0)
ufs.36.64.4193`select+0x8e(1, 80457c8, 0, 80437c8, 0, 0)
RealWaitForChar+0xfd(0, ffffffff, 0, a)
WaitForChar+0x29(ffffffff, 81f1caf, 8047838, 81489f2)
mch_inchar+0x90(81f1caf, 33, ffffffff, 70, 0, 81f1caf)
ui_inchar+0x51(81f1caf, 33, ffffffff, 70, 0, 0)
inchar+0x1bc(81f1caf, 99, ffffffff, 70)
vgetorpeek+0xaf(1, 0, 0, 0, 0, fef3b000)
vgetc+0x72(81fdd90, e, 8047b18, 8130292, fef3e380, 0)
safe_vgetc+0xb(fef3e380, 0, 8047b08, 8047acc, 81fdd90, 174)
normal_cmd+0xae(8047b40, 1, 8047bb8, 80f83fb)
main_loop+0x320(0, 0, 8047ca8, 80fa2e6)
main+0xd32(8047cac, feebf348, 8047ce8, 8078fd8, 2, 8047d18)
_start_crt+0x97(2, 8047d18, fefd1cf0, 0, 0, 0)
_start+0x1a(2, 8047e0c, 8047e10, 0, 8047e39, 8047e44)
> █

```

The debugger will attach and stop the process. To set a break point, we need the function name or address and use the `:b` cmd

To resume execution, we use `:c` , `:next` , `:s` , `:e` or `:u`

```

> WaitForChar:b
> $C
080436a8 ufs.36.64.4193`__pollsys+0x15(80436c0, 1, 0, 0, 0, c0)
08043738 ufs.36.64.4193`pselect+0x232(1, 80457c8, feebc060, 80437c8, 0, 0)
08043778 ufs.36.64.4193`select+0x8e(1, 80457c8, 0, 80437c8, 0, 0)
080477e8 RealWaitForChar+0xfd(0, ffffffff, 0, a)
08047808 WaitForChar+0x29(ffffffff, 81f1caf, 8047838, 81489f2)
08047838 mch_inchar+0x90(81f1caf, 33, ffffffff, 70, 0, 81f1caf)
08047868 ui_inchar+0x51(81f1caf, 33, ffffffff, 70, 0, 0)
08047948 inchar+0x1bc(81f1caf, 99, ffffffff, 70)
08047a18 vgetorpeek+0xaf(1, 0, 0, 0, 0, fef3b000)
08047a78 vgetc+0x72(81fdd90, e, 8047b18, 8130292, fef3e380, 0)
08047a88 safe_vgetc+0xb(fef3e380, 0, 8047b08, 8047acc, 81fdd90, 174)
08047b18 normal_cmd+0xae(8047b40, 1, 8047bb8, 80f83fb)
08047bb8 main_loop+0x320(0, 0, 8047ca8, 80fa2e6)
08047ca8 main+0xd32(8047cac, feebf348, 8047ce8, 8078fd8, 2, 8047d18)
08047ce8 _start_crt+0x97(2, 8047d18, fefd1cf0, 0, 0, 0)
08047d0c _start+0x1a(2, 8047e0c, 8047e10, 0, 8047e39, 8047e44)
> 08047808:b
> █

```


Conclusion

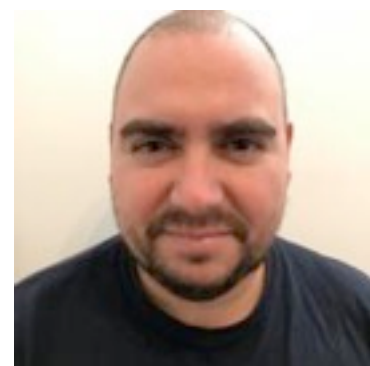
In this introduction, we have only scratched the surface of what MDB could do for us. We have not even talked about dmods which extend the utility of MDB. One example is the mdb_v8 dmod, which allows us to get more information and eases debugging of nodejs based programs. If you are using Illumos for development or even if you are running Linux on a lx branded zone, mdb will be of great help in debugging your problem.

References

https://github.com/joyent/mdb_v8

<https://wiki.smartos.org/display/DOC/Download+SmartOS>

<https://illumos.org/books/mdb/preface.html>



Meet the Author

Carlos Neira is a software engineer interested in performance, debuggability and observability of systems. He has spent most of his career as a C and kernel programmer, debugging issues on Linux, FreeBSD, Solaris and Z/OS environments.

You can reach him at cneirabustos@gmail.com

Among clouds Performance and Reliability is **critical**

A fighter jet, possibly an F-35, is shown from a low angle, flying upwards and to the right through a sky filled with white, fluffy clouds. The jet is grey with some orange markings on the nose and tail. The overall image has a blue and white color scheme.

Download syslog-ng Premium Edition
product evaluation [here](#)

Attend to a free logging tech webinar [here](#)



BalaBit
IT Security

www.balabit.com

syslog-ng log server

The world's first High-Speed Reliable Logging™ technology

HIGH-SPEED RELIABLE LOGGING

- above 500 000 messages per second
- zero message loss due to the
Reliable Log Transfer Protocol™
- trusted log transfer and storage

OpenBSD 6.3



OpenBSD 6.3 was released on April 2, 2018. The 6.3 release comprised of numerous performance related enhancements and improvements pertaining to Meltdown/Spectre (variant 2) mitigations and VMM/VMD related updates. Kernel page isolation is now implemented on OpenBSD arm64 to remediate Spectre meltdown (variant 3) vulnerabilities. The new OpenBSD 6.3 release can be downloaded from the OpenBSD mirrors and continues the tradition of media-less installations and upgrades. Please consult the OpenBSD install and documentation for more details under the heading “How to install” from <https://www.openbsd.org/63.html>. This article will highlight and go into details of the major changes for this new release.

Syspatch is now supported for the amd64 and i386 releases, and on boot, it automatically checks for available syspatch updates. Processor microcode updates can now be

installed and configured using the `fw_update(1)` for Intel and amd64 architectures.

Virtualization Features

This release features several new enhancements for OpenBSD’s `vmm(4)` and `vmd(8)`, with support for DVD/CD-ROM ISO media and support up to four network interfaces per virtual machine. It also includes native base uni-kernel interface support for `ukm` and `Solo5` kernels in `vmm(4)`, various bug fixes and related improvements.

ARM64 Features and Enhancements

For the arm64 platform on 6.3, OpenBSD release has full support for symmetric multi-processing (SMP). The Broadcom system on a chip (SoC) for the Raspberry Pi now has full support for the temperature and random number generator. For

quick reference, the mappings between the Broadcom chipset and the Raspberry Pi models are shown in the following table:

Broadcom Chipset	Raspberry Pi Models
BCM2835	Raspberry Pi 1 Model A Raspberry Pi 1+ Model A Raspberry Pi 1 Model B Raspberry Pi 1+ Model B Raspberry Pi 1 compute module Raspberry Pi Zero Raspberry Pi Zero W
BCM2836	Raspberry Pi 2 Model B
BCM2837	Raspberry Pi 3 compute module Raspberry Pi 3 lite compute module Raspberry Pi 2 v1.2 Model B Raspberry Pi 3 Raspberry Pi 3+ Model B

For a list of hardware driver support, please refer to <https://www.openbsd.org/arm64.html>.

Other related embedded platforms which are significantly supported include Allwinner SoCs, Pine64, and Rockchip RK 3328/RK3288 SoCs platforms. A notable enhancement is the full support for general purpose input and output (GPIO) ports for the various Allwinner SoCs using the `gpioctl(8)` management interface.

Security, OpenSSH, Networking and LibreSSL Features

The new OpenSSH 7.7 `sshd` daemon enhancements consist of key expiry via the `expiry-time` option for `authorized_keys`. The new OpenSSH server `BindInterface` option binds outbound connection to an interface address and supports automatic tun/tap interface forwarding configuration which is controlled by the new `SSH_TUNNEL` environment setting.

Similarly, the new OpenSSH client now features the tun/tap interface forwarding support using

the `LocalCommand` and the `%T` expansion options to be executed when post-connecting to the SSH server. It is important to note that legacy support for OpenSSH server and clients released in or before 2001 has been deprecated.

In particular, an interesting new feature is PF firewall support for controlling TCP syncookie behaviour using the `set syncookies` options to `never`, `always` or `adaptive` which allows for setting the state table percentage thresholds for commencing and terminating syncookie mode. This feature reinforces OpenBSD PF ability to mitigate synflood denial of service attacks. (For additional details, please refer to: <http://man.openbsd.org/OpenBSD-6.3/pf.conf.5>) . Address resolution protocol (ARP) behaviour can now be controlled using the `ifconfig(8)` `staticarp/-staticarp` options to only reply to ARP requests for its respective interface addresses and the latter to enable normal ARP functionality.

Finally, the new LibreSSL 2.7.2 release contains compatibility enhancements for legacy OpenSSL API and support for OpenSSL 1.0.2 and OpenSSL 1.1 for backwards compatibility. Additionally, performance enhancements were implemented for the ARMv7 architecture.

Conclusion

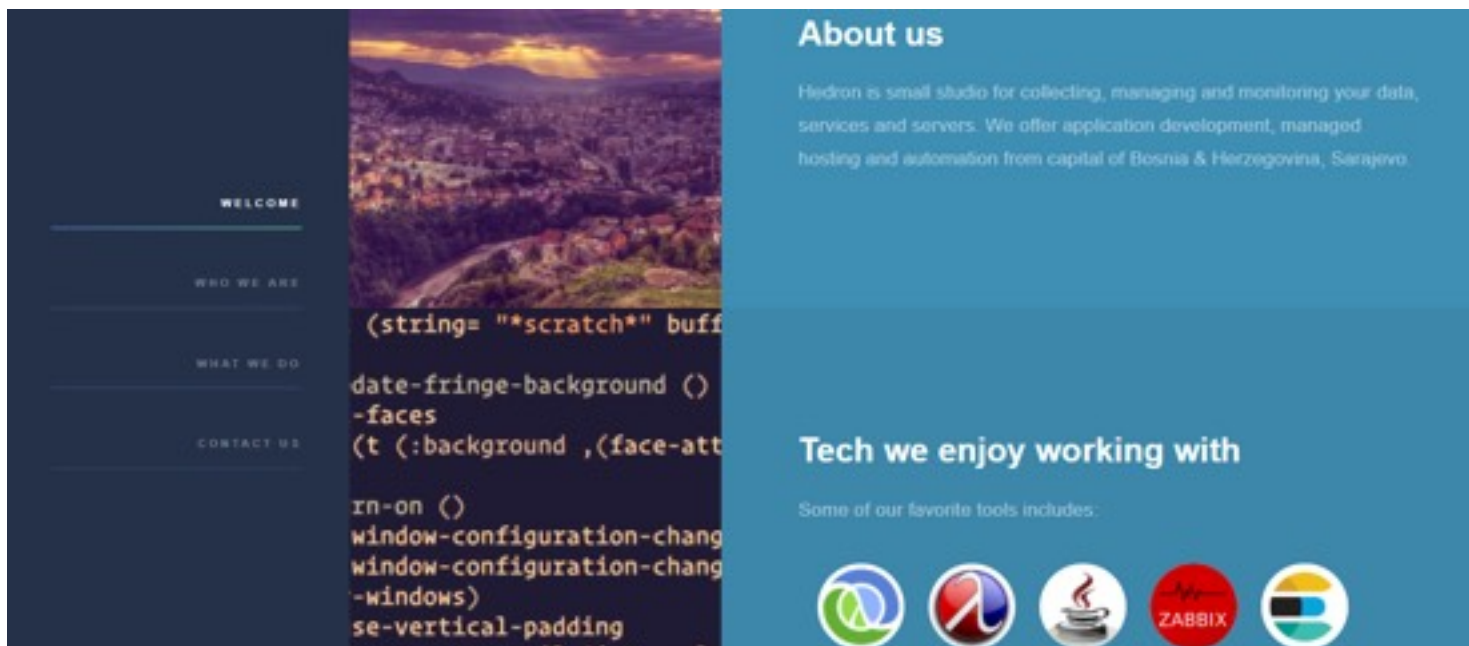
The OpenBSD 6.3 release contains many significant performance and enhancement features in all areas of the operating system.

Meet the Author

Albert Hui

To contact the author: alberthui3@yahoo.com

Interview with Sanel Zukan



Can you tell our readers about yourself?

I'm an open source enthusiast, LISP hacker, and free software devotee. I also run my company, Hedron d.o.o., and I'm doing all of that from Emacs.

When was your first contact with a computer, and what attracted you at first?

My mother is accountant and I had a chance to play with some old 386 she used for her work. What attracted me? Games, like many of us.

Please tell us more about your company and what you do?

Hedron (<https://hedron.cc>) is a small, one-man show firm which is mainly focused on collecting data, analytics and resource monitoring. Most of the work is done in Clojure, but there are parts in C++, Racket, and newLISP.

Since I'm the only one in the company (for now), I'm in charge of everything: from development, company management to accounting. Luckily, I enjoy automating things and thus most of the daunting tasks are done by scripts and web services.

Do you remember your very first development ? How do you consider it now?

Oh, sure. It was a small Pascal application (high school assignment) I used to sell to those who didn't like or know how to code. It was an embarrassingly simple application for modern standards, to be honest.

What was your best work? Can you tell us the idea behind it? What was its purpose?

Rewriting and modernizing EDE Desktop (<https://edeproject.org>). The main idea was to make it more standard on distributions by using standard FLTK toolkit instead of the custom one. Sadly, I had to pause the work on EDE due to daily activities.

What tools do you use most often and why?

EDE, Emacs and standard terminal. They accomplish 99% of the tasks I need to get done.

What was the most difficult and challenging implementation you've done so far? Could you give us some details?

Probably writing my own window manager. This kind of programs wasn't that common 10 years ago and X.org (or XFree86) API isn't the most friendly thing around. Therefore, it was really difficult to find a straightforward and simple tutorial. However, I never finalized it, instead, we added pekwm (really nice window manager) in EDE.

Do you have your own development works?

Yes. I use org-mode (and Emacs) for almost everything - from organizing things, planning, charting, to writing specifications and technical documentation. Other than that, I try to keep things simple and manageable.

What future do you see for FreeBSD and other OSes? Can you tell us about your favorite features in the new releases?

The one aspect I like the most about FreeBSD (and other *BSD implementations) is its ability to keep stuff unix-way, plain simple. Sadly, Linux got infested with systemd (which I don't like at all) and the only sound distro not using it is Slackware - it is quite similar to *BSD philosophy. For FreeBSD future releases, I look forward to more hardware support and less crap like systemd.

Do you have any specific goals for the rest of this year?

Many. The most important one: get myself organized better.

What's the best advice you can give to the BSD magazine readers?

Never stop hacking, exploring, breaking and learning about things.

Expert Speak by E.G.Nadhan

5 Imperatives for Catalysts of Change

In his keynote at this symposium, [Gartner Executive VP and Analyst Peter Sondergaard](#) had highlighted certain companies which scored high on the Gartner Digital IQ index – Great examples of enterprises that have treated change as a catalyst to play the game on digital terms. **Change is not just about what you do but where you do it** – the channels you choose to play in, who you work with as well as the time and frequency of these interactions. More importantly, change can be a catalyst rather than an adversary. However, change does not always come with notice. Change can happen through continuous injection of incremental, minute triggers that have a cumulative effect suddenly manifesting itself and taking us completely unawares :: Hello Disruption! So, what can enterprises do to deal with such changes? What are the imperatives for partnering with change?

Join me as I elaborate on these imperatives that have stood the test of time and hold the promise for dealing with any Change in the future. These are the triggers that emerged from [my session at the Gartner ITXPO 2017 Conference](#).



Click [here](#) and watch the full presentation.

Customer Matters. The simple Rule #1 about the customer being right that went into place since the first lemonade stand and is as true today as it ever has been. The customer is the perfect barometer to drive relevant change. In the chaotic world of myriad paradigms, platforms, technologies and tools, the imperative of doing whatever it takes to keep the customer happy can never go wrong. Note that the customers themselves may be driving change by shifting their expectations. *Imperative 1: Just do what the customer wants.*

History Matters. Change can go through a cyclic pattern over a period of time just like the economy or fashion trends. In my session, I suggest that Amazon can actually look decades back into the history of how Sears dealt with change. The steady transition from a mail-order catalog company to a brick and mortar store is akin to what Amazon is going through through the acquisition of Whole Foods to augment their online presence. History is replete with patterns of socio-economic behavior that give more character to future trends. *Imperative 2: Look back into the future of history.*

Collaboration Matters. A closer study of world leaders who accomplished a lot with very little to start with reveals the art of collaboration as a key mantra that empowered these maestros to achieve the impossible. [Mother Teresa](#). Mahatma Gandhi. [Malala Yousafzai](#). Nelson Mandela. Martin Luther King. They used collaboration to instrument long-lasting change by partnering with the underlying sentiments of the masses. Collaboration is the name of the game in the digital world too. The [Open Earth Community](#) is an open community of scientists, engineers and software developers in oil and gas companies, all working together to speed up and lower the cost of digital innovation for the entire industry. Cross-functional teams across Red Hat came together for a single mission, to accelerate various IT initiatives. [Click to see the experience of the innovator](#). *Imperative 3: Collaboration is a key driver for systemic innovation.*

Leadership Matters. Leaders must sustain an environment that fuels continuous change by removing perceived “roadblocks” and opening up non-traditional channels of creative interactions. Leaders must drive careers of achievements and not accomplishments. Leaders like [Jeff Harmening, CEO of General Mills](#), actually suggests that even large, global companies like General Mills can drive systemic change. *Imperative 4: You don't have to be a startup to embrace change.*

Culture Matters. When Jim Whitehurst joined Red Hat as the CEO, he went through a transition from a very structured environment to the open organization that symbolizes the Red Hat culture. The Open Organization book, authored by Jim, is for leaders who want to create business environments that can respond quickly in today's fast-paced world. It's for those who want to encourage the best ideas, hear honest advice, and attract (and retain) the brightest talent. Whitehurst embraced this [culture to drive change](#) as he elaborates in this executive roundtable. *Imperative 5: Partnership with change starts with the individual.*

There you have it.

Partnering with change is about a culture of continuous collaboration with the support of forward-thinking leadership while looking back and learning from the history to always ensure a happy customer!

Change is happening and is bound to impact every one of us one way or the other.

Are there other imperatives that you would suggest to partner with change?

Welcome to our brave new digital world.

See you there!

Meet the Author



E.G.Nadhan is the Chief Technology Strategist for the Central Region at Red Hat. He provides thought leadership on various concepts including Cloud, Big Data, Analytics and the Internet of Things (IoT) through multiple channels including industry conferences, Executive Roundtables as well as customer specific Executive Briefing sessions. With 25+ years of experience in the IT industry selling, delivering and managing enterprise solutions for global corporations, he works with the executive leadership of enterprises to innovatively drive Digital Transformation with a healthy blend of emerging solutions and a DevOps mindset. Follow Nadhan on [Twitter](#) and [LinkedIn](#).

WE'RE Add-Ons FOR OTRS

EMPOWER
your **TEAM**
with a tool
to lift their
performance

ADD-ONS FOR OTRS

Marketplace for OTRS

WE HAVE the expertise in:

- Premium Add-Ons for OTRS
- OTRS Update on demand
- OTRS Implementation
- Add-Ons development for OTRS
- OTRS support 24/7

We adjust to ad hoc requests and solve demanding tasks remotely. We cooperate with partners worldwide such as France, Germany, Greece, Macedonia, Poland, the Netherlands, the USA and Uruguay.

CONTACT US



+48 694 509 649



info@otrsaddons.com



www.otrsaddons.com



OPGK Rzeszow S.A.
ul. Geodetow 1,
35-328 Rzeszow,
Poland



ADD-ONS FOR OTRS

The doves and the hawks are gathering for a showdown, be it in geopolitics or the Internet. Facebook and Cambridge Analytica, the West and Russia, are all walking on a tightrope. Brinkmanship is the current name of the game. Who is going to come out on top?

by Rob Somerville

OK, I'll admit it. As an individual over the past few years, I've moved from the technology camp more into the arena of politics, writing, and philosophy. If you find my column boring, uninspiring or something that is not to your taste, please let your feelings be known to the editor. I'll happily hand over my keyboard to someone else. If my observations and arguments don't carry any weight in the grinding mill of time or speculative commentary, it is then time for me to hang up my spurs.

I cannot, however, go away quietly and just ignore a number of coincidences, that if were placed at the foot of general public (never mind technologists), a few years ago, would have raised the flag of "conspiracy theorist". Yes, we are talking about the previous US election, Brexit, Facebook, Cambridge Analytica, and the current stand-off via the totally immoral proxy war that continues in Syria between the West and Russia. Anyone who cannot join the dots between these entities is sorely in need of some education. All are joined at the hip in one regard, be it using whatever word you wish to choose. Conspiracy. Transformation. Change agent. Disruptor. We are currently hanging on to the tail of the tiger, jumping the shark, or riding the wave. Irrespective of the metaphor used, it is not a pleasant ride. Be you a Republican or Democrat, Socialist or Conservative, facing a scenario where Russia, British and USA relations are at an all-time low since the Cuban missile crisis is no laughing matter. Some of you reading this column were not even born then, but I was a two-year-old child at the time. So, I hope, that at least you will allow me to borrow your ears.

Having spent far too many hours reading and researching, I think I have worked out roughly where we are at on the roadmap to Armageddon. Provided the channels stay open and dialogue is maintained, I don't think we are in any immediate danger of either nation being turned to glass. Even in the depths of the bitterest of cold wars, the respective militaries were a significant buffer zone. Unless you are under the jackboot of a vicious dictator, common sense, courage, and integrity are the watchwords of a

mature and professional army. It may be a very British phrase, but “old soldiers” is a very peculiar phrase, in that it is paradoxical. Most, who have experienced the battlefield, will be the first to defend their “opponent”, in knowing the bloodshed, sacrifice and pure senselessness of it all creates a bond that is not easily broken. You have got to have lived through a war to get it.

This is probably one of the most important columns I have ever written. I am acutely aware that while BSD advocates are worldwide, to some, the Open-Source movement is considered surreptitious, recidivist, and beyond the pale in some circles. Or to put it another way, you can be patriotic (e.g. Western computing methodology, Microsoft and IBM *et al*) or be an awkward cuss and follow the path of the East (mass production, innovation and academic research). The two cultures are very different, a ying and yang of outlook, experience, and approach. I have worked closely with many individuals across the nations who have a passion for computing, including a Russian, and despite our passion for beer and technology, I have found no flaw in anyone. The only IT professional I can sincerely complain about on a nationality basis was 5 foot 4, and they had a serious attitude problem. Their country was seriously messed up though, and still is to this day. Thinks cars with built-in flame-throwers. Come to think of it, the other only serious argument I had with another IT “professional” had the same issues. However, they were just an idiot, having, if I remembered correctly, eyes and hands on my girlfriend at that time. Neither were Russian nor Eastern Bloc.

Globally, we can sort this out if cool heads and diplomacy avails. The bigger issue is the war on the Internet. I really want to vomit on my cornflakes when this whole issue of “fake news” and “propaganda” comes up, especially when it comes down to the censorship of the common man. Both Facebook and Cambridge Analytica now clearly fall into that category. While the corporate lawyers and politicians will argue until hell freezes regarding the rights and wrongs of the situation, data will be mined and people will be left hanging out to dry. As the old adage goes, if it is free, generally you are not the consumer, but the product.

I have been a staunch Open-Source advocate since 2000. I look at the development lists, the contributors, and those that contribute behind the scenes. I see a plethora of contribution across national and global boundaries. Although I see differences, problems, difficulties, misunderstandings, one thing I don’t see is war. The important thing to realise is that we come to a point where we can agree to disagree. I think Microsoft is terrible, but getting better. I think Open-Source has lost its way to a certain degree, and missed too many opportunities. I know what side I’m on, but I’m far too old to argue about it.

There is a bigger fight going on 24/7 around us. The first casualty in war is always the truth, and the Internet is now the new battlefield. It is clear that Russia is being demonised, be it in the mainstream media or the darkened rooms of intelligence services where the whisper of “cyber-attack” is being mentioned. I won’t mention the name of my local paper, but sadly, even they have got in on the game. Problem is, I actually know what is going on as far as a local political level is concerned, and if a herd of bulls ate their way through a container ship of silage, hay and grass (the field kind, not the aromatic one), there would be less mess to sort out on their front page.

We know who the bad guys are. I will go to bed tonight, safe in the knowledge that I will wake tomorrow. I have no doubt that IT admins, security professionals, and those who know what they are doing will not act without three essential witnesses – Evidence, experience and intuition.



Dr.WEB®
CureIt!™

EMERGENCY CURING

for Windows workstations and servers
including those running other anti-virus software



FUNCTIONS:

- Cures Windows workstations and servers.
- Verifies the quality of the anti-virus software currently in use.

FEATURES:

- Dr.Web CureIt! doesn't require installation and doesn't conflict with any known anti-virus; consequently there is no need to disable the anti-virus currently in use to check a system with Dr.Web CureIt!.
- Improved self-protection and an enhanced mode for more efficient countermeasures against Windows blockers.
- Dr.Web CureIt! is updated at least once an hour.
- The utility can be launched from removable media including USB storage devices.

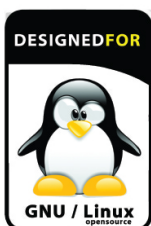
LICENSING FEATURES:

The utility is available for free when used for non-business purposes.



Rack-mount networking server

Designed for BSD and Linux Systems



Designed. Certified. Supported

Up to **5.5Gbit/s**
routing power!



KEY FEATURES

- ▶ 6 NICs w/ Intel igb(4) driver w/ bypass
- ▶ Hand-picked server chipsets
- ▶ Netmap Ready (FreeBSD & pfSense)
- ▶ Up to 14 Gigabit expansion ports
- ▶ Up to 4x10GbE SFP+ expansion



PERFECT FOR

- ▶ BGP & OSPF routing
- ▶ Firewall & UTM Security Appliances
- ▶ Intrusion Detection & WAF
- ▶ CDN & Web Cache / Proxy
- ▶ E-mail Server & SMTP Filtering