Container für Pragmatiker

Tom Lazar <tomster@pyfidelity.com>

Berlin Operations Summit 2016

“Containers! Is there any word more thrilling to the human soul?”

John Arundel

in his upcoming edition of 'The Puppet Beginner's Guide'
https://twitter.com/bitfield/status/772832645243887620

Why not just use containers?

Containers! Is there any word more thrilling to the human soul? Many people feel as though containers are going to make configuration management problems just go away. This feeling rarely lasts beyond the first few hours of trying to containerize an app. Yes, containers make it easy to deploy and manage software, but where do containers come from? It turns out someone has to build and maintain them, and that means managing Dockerfiles, volumes, networks, clusters, image repositories, dependencies, and so on. In other words, configuration. There is an axiom of computer science called 'The Law of Conservation of Pain'. If you save yourself pain in one place, it pops up again in another. Whatever cool new technology comes along, it won't solve all our problems; at best, it will replace them with refreshingly different problems.

The fisherman and the businessman

http://paulocoelhoblog.com/2015/09/04/the-fisherman-and-the-businessman/

Disclaimer

- rein anekdotisch
- kleine Firma (zu viert)
- dutzende, nicht tausende von Servern und Services
- Kerngeschäft ist Softwareentwicklung, nicht Hosting oder Deployment

Enter FreeBSD

- Direkter Nachfahre von Berkely Unix
- 1. Release 1993
- stabile Entwicklung
- “opinionated”

Enter FreeBSD

- Kernel + Welt
- Jails
- ZFS
- strikt vorgegebene Verzeichnisstruktur
  (`/` vs `/usr/local`)
- Ports (pkg + poudriere)

FreeBSD World

- schlankes Basissystem (65Mb)
- streng behütet – kein bash, kein perl :-)
- alle Komponenten aufeinander abgestimmt
- “definierte Kante” für die Ports

FreeBSD Ports

- aktuell ca. 26.000 Pakete
- besteht prinzipiell aus Konfiguration und Patchfiles für FreeBSD
- compiled aus den Sourcen (wie bspw. gentoo)

pkg

- basiert auf den Ports
- bietet Binärinstallationen
- Support für Multi-Repo

Jails

- Kernel Feature zur Prozessisolierung
- Erste Implementierung im Jahr 2000
- im Prinzip “chroot für Prozesse” - kaum Overhead

ezjail

- Jail Management Tool, seit 2005, de facto Standard, tausende von Usern
- ca. 2.000 Zeilen `sh`, ein Autor
- Konzept eines read-only Basejails via nullfs
- Aus der Innenperspektive kaum von einer vollwertigen FreeBSD Installation unterscheidbar
- ZFS Support

ZFS

- Copy-On-Write Filesystem
- Fokus auf Datenintegrität
- Cheap Snapshots
- Im FreeBSD Kernel seit 2007

Jails + ZFS + nullfs = Magic

- `/usr/jails/basejail -> /usr/jails/foo/basejail`
- readonly
- “nacktes” Jail ca. 2Mb
- Ports etc. schreiben nach **`/usr/local/`**

BSDploy

- Remote Jail Management Tool, seit 2014
- erweitert ezjail um Provisioning und Configuration
- ca. 2.000 Zeilen Python + 600 Zeilen Ansible
- zwei Autoren, beide In-House
- bestenfalls dutzende von Usern

BSDploy

- Erweitert ezjail um persistente Volumes
  z.B. /data/postgres -> /usr/jails/db/usr/local/pgsql/data/
- modulares Provisioning für derzeit EC2, Digital Ocean, Hetzner, Virtualbox
- Verwendet Ansible für Konfiguration (deklarativ)
- Verwendet Fabric für imperatives Scripting via Python
- Support für poudriere und eigene pkg Repos

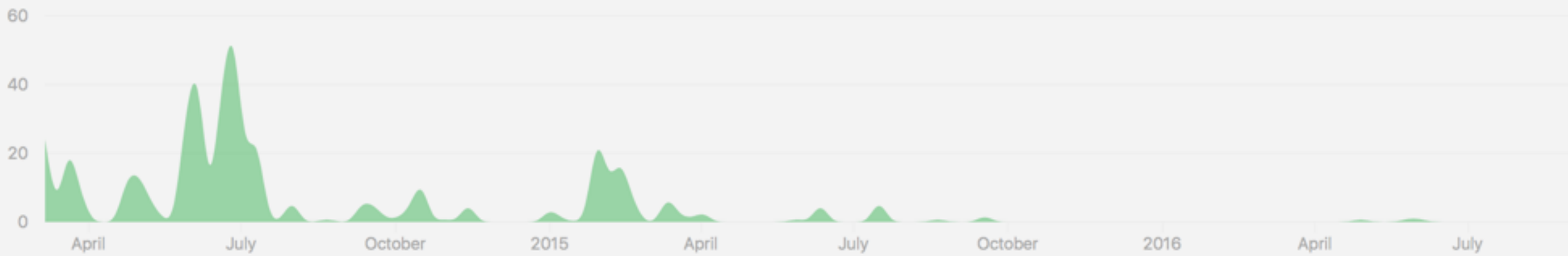
BSDploy

- “aktiv entwickelt”
- das ist ein Feature!

Mar 16, 2014 – Sep 9, 2016

Contributions to master, excluding merge commits

Contributions: **Commits**



poudriere

Build Jobs Results Logs

Queued	Built	Failed	Skipped	Ignored	Remaining
4	4	0	0	0	0

Load Averages	Swapinfo	Elapsed	Pkg/Hour	Impulse
(32%) 1.29 1.12 1.12	0.00%	04:34:41	1	--

Build

Master	102amd64-briefkasten-briefkasten
Build	2015-11-12_16h15m55s
Status	stopped:done:
Jail	102amd64
Set	briefkasten
Ports Tree	briefkasten
SVN	

Built ports

Show 10 entries Search:

#	Package	Origin	Log
1	cairo-1.14.2,2	graphics/cairo	success
2	harfbuzz-1.0.6	print/harfbuzz	success

poudriere

- Tool zur Portkompilierung
- Teil der offiziellen FreeBSD Infrastruktur
- Erlaubt mit wenig Aufwand eigene Repository
- z.B. pro Kunde, pro Projekt

Zusammenfassung

FreeBSD (hier) + ZFS + nullfs + ezjail + BSDploy + Ansible + poudriere + pkg = “containerish”

Vorteile

- extrem stabil, angenehm langweilig
- ausschliesslich “frische Zutaten” aus kuratierten Quellen
- immutable Systems + persistent Volumes
- “best of both” (Ersetzen vs. Updaten)

Nachteile

- keine Erfahrung mit grossen Systemen (mal Netflix oder WhatsApp fragen…)
- Provisioning von Hosts dauert Minuten, nicht Sekunden
- Provisioning von Containern dauert Sekunden, nicht Millisekunden
- gelegentlich Reibungsverluste “weil nicht Linux”

Anwendungsbeispiel “Briefkasten”

ZEIT ONLINE

ZEIT ONLINE-Briefkasten

Sie können uns hier über eine verschlüsselte Verbindung Dokumente und eine Nachricht hochladen. Ihre Dateien (z.B. Word, PDF, Excel) werden automatisch von allen versteckt gespeicherten Informationen

Briefkasten

- vollständige Applikation

- prinzipiell keine Distribution als fertiges Image, Dockerfile oder ähnliches

Fazit

- Container Paradigmen sind ne feine Sache!
- Mut zu eigenen Lösungen!
- aber möglichst wenig Code schreiben!

Noch Fragen?

- tom@tomster.org (Email)
- @tomlazar (Twitter)
- in Persona

Danke für’s Zuhören!

Linksammlung

- https://www.freebsd.org
- https://erdgeist.org/arts/software/ezjail/
- https://www.freebsd.org/doc/handbook/zfs.html
- https://github.com/ployground/bsdploy
- https://www.freebsd.org/doc/handbook/ports-poudriere.html
- https://github.com/ZeitOnline/briefkasten