BSD:
The OS which almost prevented that Linux was born!

FreeBSD and other variants

Typical use cases

A demo / try out after the session

Questions? After the presentation please. Thank you!

Content

01 Introduction

02 Some history

03 A different command line world

04 FreeBSD as Server

- Desktop
- SW develop machine
- Jails

05 Final Words / Demo

01 Introduction
Clues that your speaker might like (Free)BSD

Red text

A FreeBSD logo on every slide

But there is hope…

He uses **NixOS** on his laptop and more Linux variants…

Interested in a NixOS presentation / Workshop or demo?

01 Introduction

I developed software professionally,
now for Home use, IoT / OpenHAB / Python and I make beer

When do I use what OS?

FreeBSD:

Servers

Jails

VMs

NAS

Firewall

Linux (NixOS / Debian):

Desktop

Laptop

“Must have Linux” Server apps

SW development

Why? Let me explain!

02 Some History

- The history of the Berkeley Software Distribution

  began in the 1970s when University of California, Berkeley received a copy of Unix. 20 years of law suits and discussion later ...

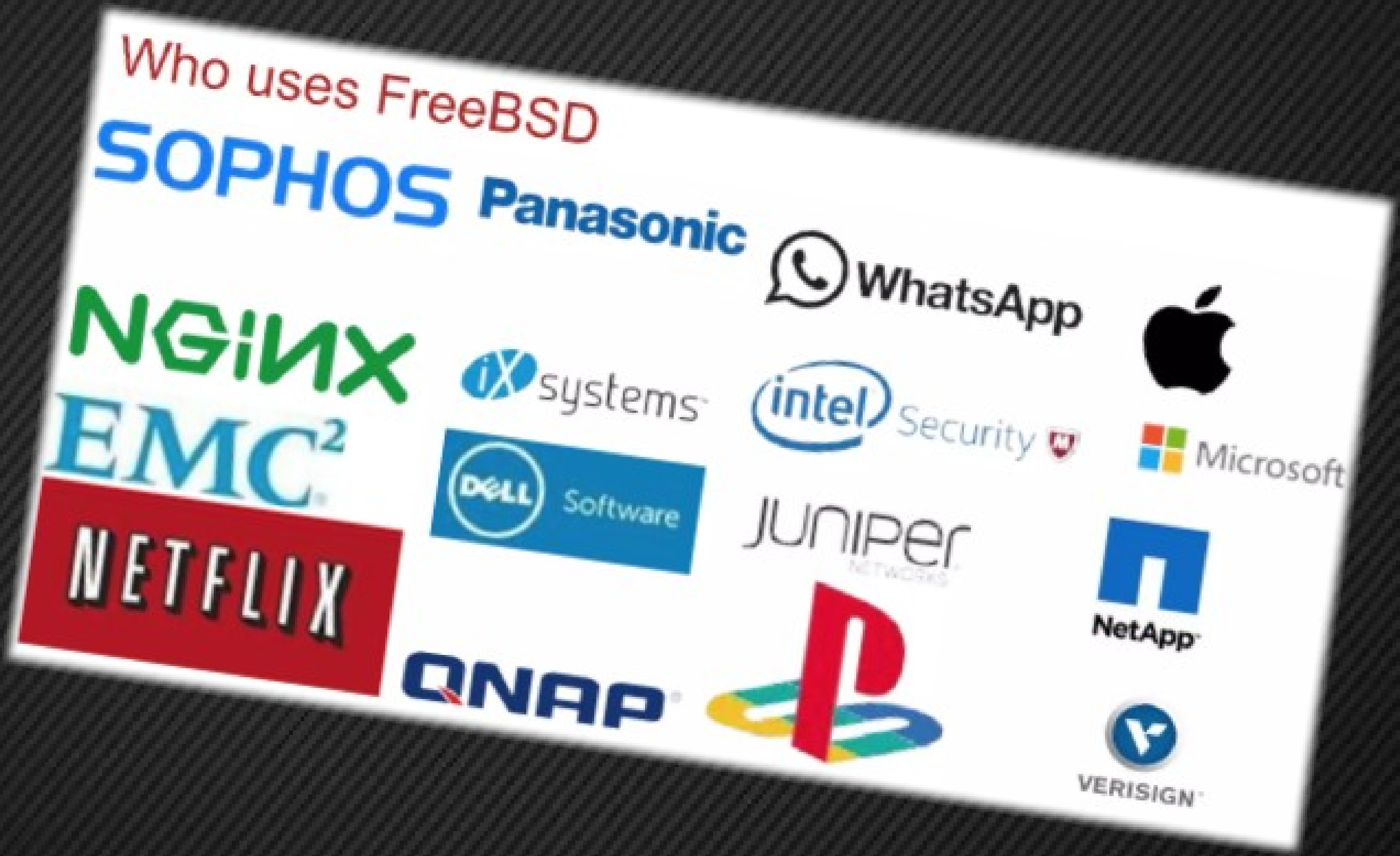
- In 1992, Bill and Lynne Jolitz, release 386BSD 0.0, the first version of BSD for the Intel 386. Later in 1992 386BSD 0.1 was released and set the stage for the formation of FreeBSD and NetBSD and more

And not to forget:

- “If 386BSD had been available when I started on Linux ? (1991) Linux would probably never had happened”

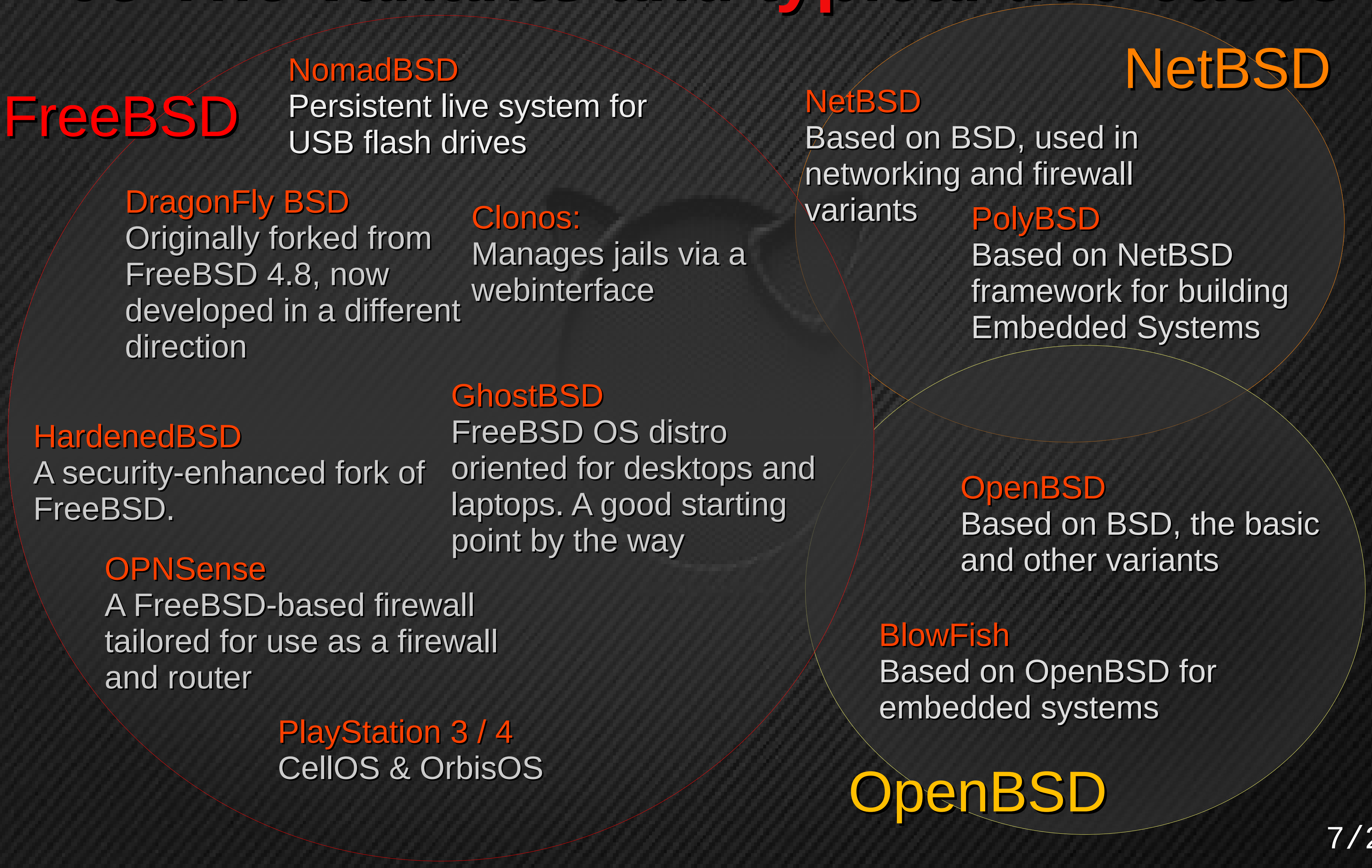
  Linus Torvalds in 1993

02 Some history: Used widely



And many more ...

03 The variants and typical use cases



FreeBSD

NomadBSD
Persistent live system for USB flash drives

DragonFly BSD
Originally forked from FreeBSD 4.8, now developed in a different direction

Clonos:
Manages jails via a webinterface

HardenedBSD
A security-enhanced fork of FreeBSD.

GhostBSD
FreeBSD OS distro oriented for desktops and laptops. A good starting point by the way

OPNSense
A FreeBSD-based firewall tailored for use as a firewall and router

PlayStation 3 / 4
CellOS & OrbisOS

NetBSD

NetBSD
Based on BSD, used in networking and firewall variants

PolyBSD
Based on NetBSD framework for building Embedded Systems

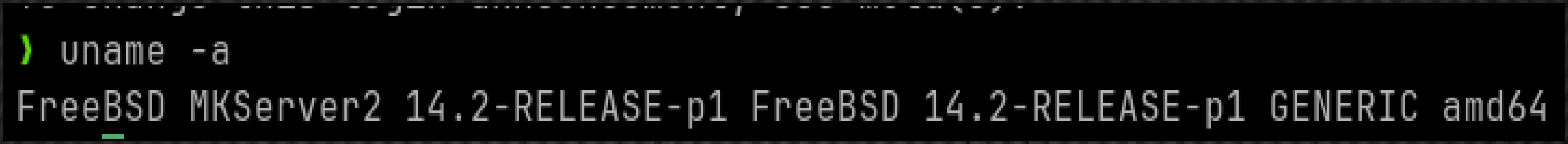
OpenBSD
Based on BSD, the basic and other variants

BlowFish
Based on OpenBSD for embedded systems

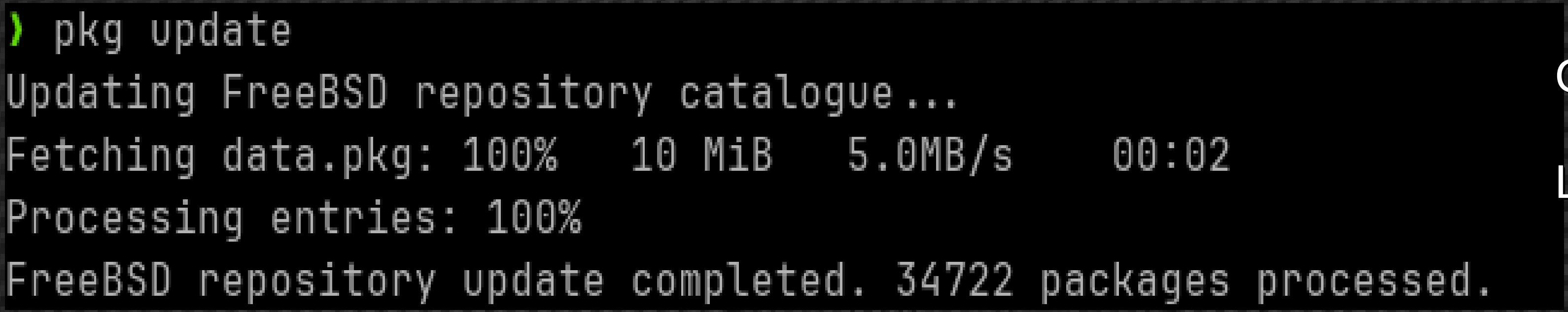
OpenBSD

An many more → let’s focus on FreeBSD

03 A different command line world



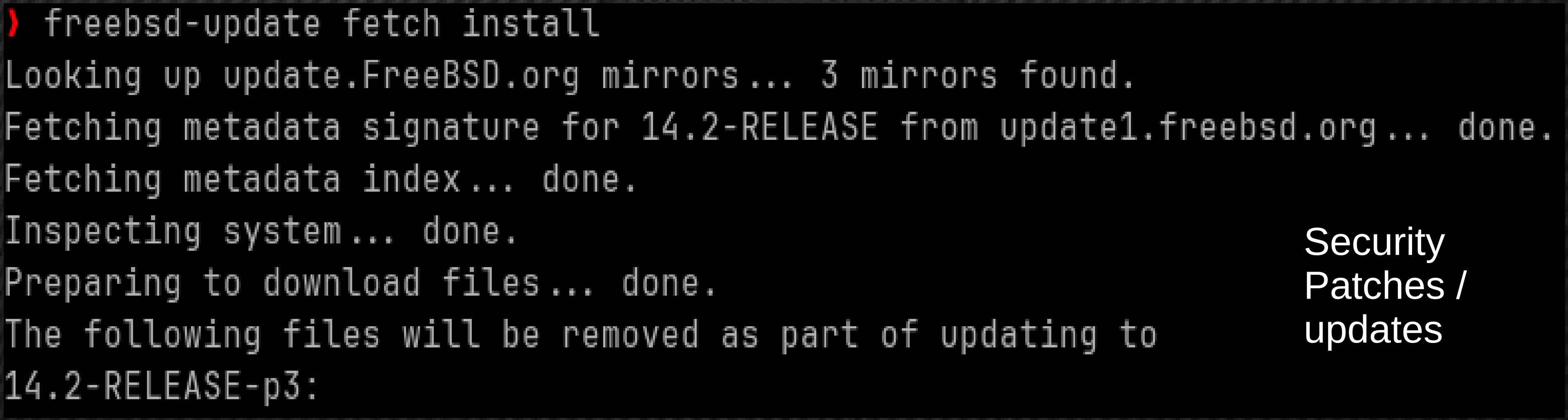
```
To change this login announcement, see motd(5).
❯ uname -a
FreeBSD MKServer2 14.2-RELEASE-p1 FreeBSD 14.2-RELEASE-p1 GENERIC amd64
```



```
❯ pkg update
Updating FreeBSD repository catalogue...
Fetching data.pkg: 100%    10 MiB   5.0MB/s    00:02
Processing entries: 100%
FreeBSD repository update completed. 34722 packages processed.
```

Current repo

Latest repo

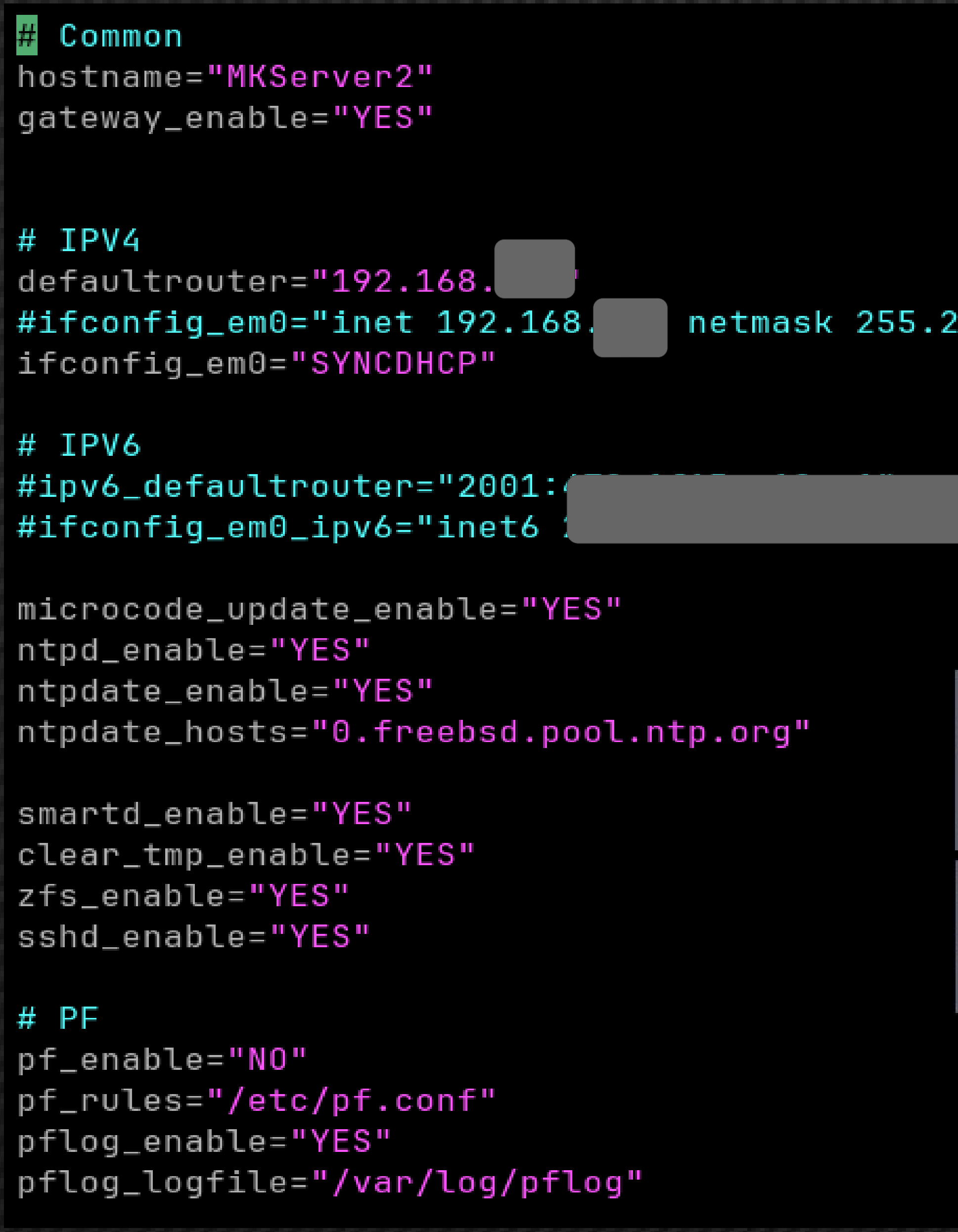


```
❯ freebsd-update fetch install
Looking up update.FreeBSD.org mirrors... 3 mirrors found.
Fetching metadata signature for 14.2-RELEASE from update1.freebsd.org... done.
Fetching metadata index... done.
Inspecting system... done.
Preparing to download files... done.
The following files will be removed as part of updating to
14.2-RELEASE-p3:
```

Security
Patches /
updates

I never (until now) experienced a broken system due to upgrades or installs

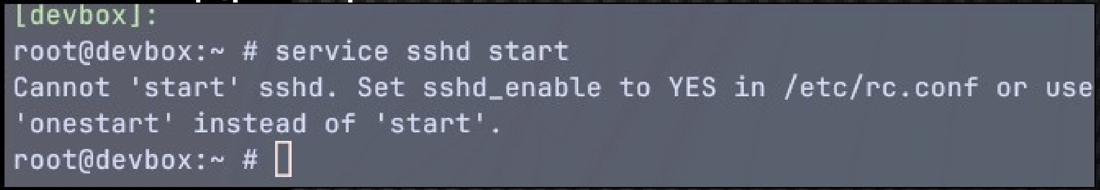
03 A different command line world



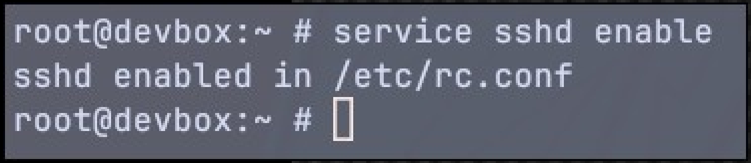
```
# Common
hostname="MKServer2"
gateway_enable="YES"


# IPV4
defaultrouter="192.168.[redacted]"
#ifconfig_em0="inet 192.168.[redacted] netmask 255.255.255.255"
ifconfig_em0="SYNCDHCP"

# IPV6
#ipv6_defaultrouter="2001:[redacted]
#ifconfig_em0_ipv6="inet6 "

microcode_update_enable="YES"
ntpd_enable="YES"
ntpdate_enable="YES"
ntpdate_hosts="0.freebsd.pool.ntp.org"

smartd_enable="YES"
clear_tmp_enable="YES"
zfs_enable="YES"
sshd_enable="YES"

# PF
pf_enable="NO"
pf_rules="/etc/pf.conf"
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
```

Manage services / settings in rc.conf

One config file

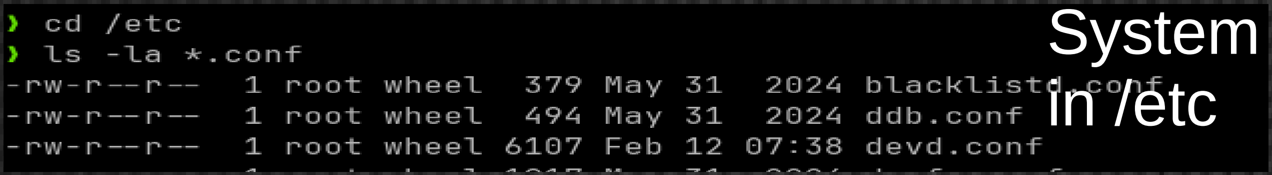


```
[devbox]:
root@devbox:~ # service sshd start
Cannot 'start' sshd. Set sshd_enable to YES in /etc/rc.conf or use
'onestart' instead of 'start'.
root@devbox:~ # 
```



```
root@devbox:~ # service sshd enable
sshd enabled in /etc/rc.conf
root@devbox:~ # 
```

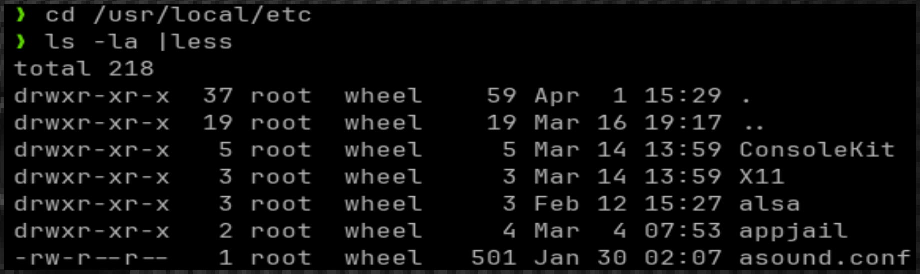
03 A different command line world



```
❯ cd /etc
❯ ls -la *.conf
-rw-r--r--  1 root wheel  379 May 31  2024 blacklistd.conf
-rw-r--r--  1 root wheel  494 May 31  2024 ddb.conf
-rw-r--r--  1 root wheel 6107 Feb 12 07:38 devd.conf
```

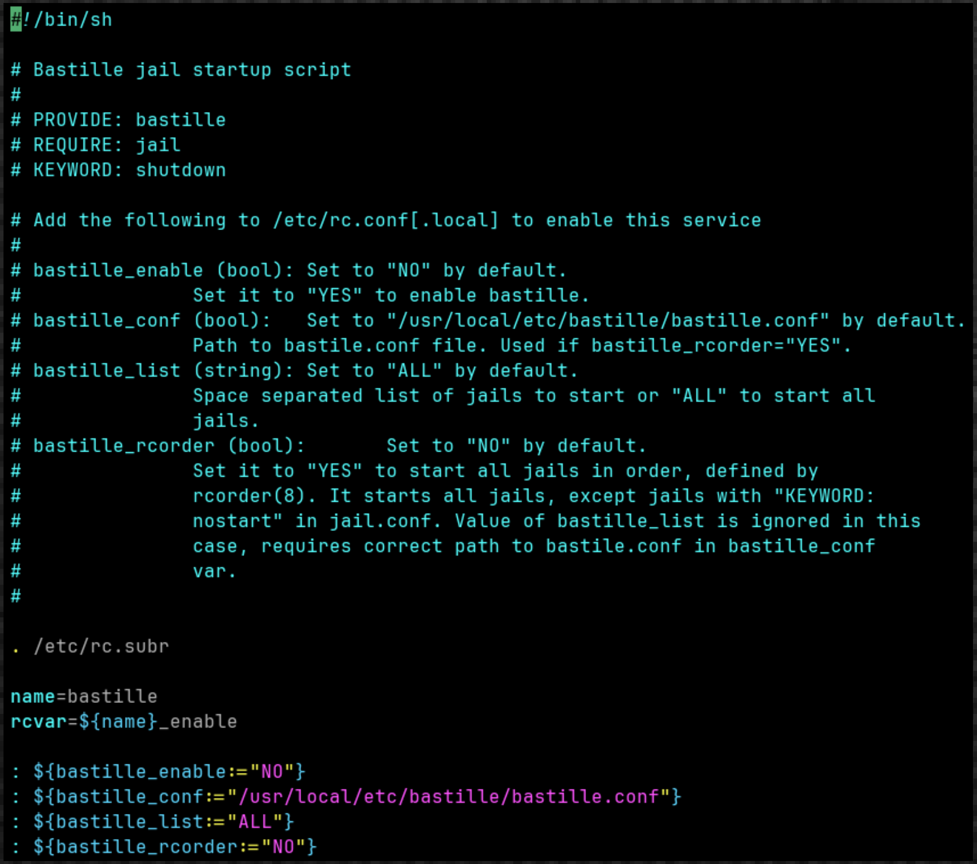
System config in /etc

Userland in /usr/local/etc



```
❯ cd /usr/local/etc
❯ ls -la |less
total 218
drwxr-xr-x  37 root  wheel    59 Apr  1 15:29 .
drwxr-xr-x  19 root  wheel    19 Mar 16 19:17 ..
drwxr-xr-x   5 root  wheel     5 Mar 14 13:59 ConsoleKit
drwxr-xr-x   3 root  wheel     3 Mar 14 13:59 X11
drwxr-xr-x   3 root  wheel     3 Feb 12 15:27 alsa
drwxr-xr-x   2 root  wheel     4 Mar  4 07:53 appjail
-rw-r--r--   1 root  wheel   501 Jan 30 02:07 asound.conf
```

Services in /etc/rc.d or /usr/local/rc.d
Start / stop etc: service bastille start

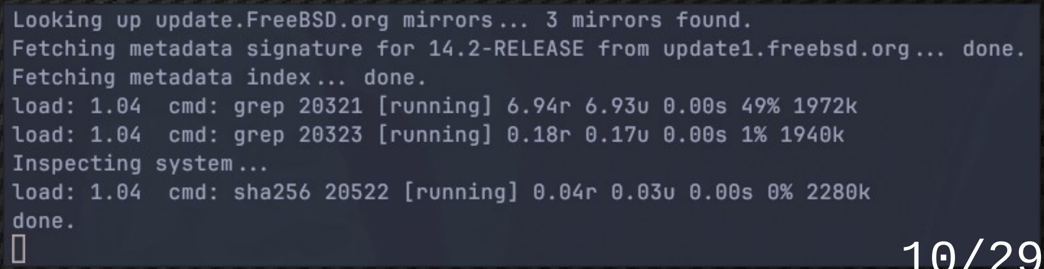


```
#!/bin/sh

# Bastille jail startup script
#
# PROVIDE: bastille
# REQUIRE: jail
# KEYWORD: shutdown

# Add the following to /etc/rc.conf[.local] to enable this service
#
# bastille_enable (bool): Set to "NO" by default.
#               Set it to "YES" to enable bastille.
# bastille_conf (bool):   Set to "/usr/local/etc/bastille/bastille.conf" by default.
#               Path to bastile.conf file. Used if bastille_rcorder="YES".
# bastille_list (string): Set to "ALL" by default.
#               Space separated list of jails to start or "ALL" to start all
#               jails.
# bastille_rcorder (bool):       Set to "NO" by default.
#               Set it to "YES" to start all jails in order, defined by
#               rcorder(8). It starts all jails, except jails with "KEYWORD:
#               nostart" in jail.conf. Value of bastille_list is ignored in this
#               case, requires correct path to bastile.conf in bastille_conf
#               var.
#

. /etc/rc.subr

name=bastille
rcvar=${name}_enable

: ${bastille_enable:="NO"}
: ${bastille_conf:="/usr/local/etc/bastille/bastille.conf"}
: ${bastille_list:="ALL"}
: ${bastille_rcorder:="NO"}
```

Startup: no run levels

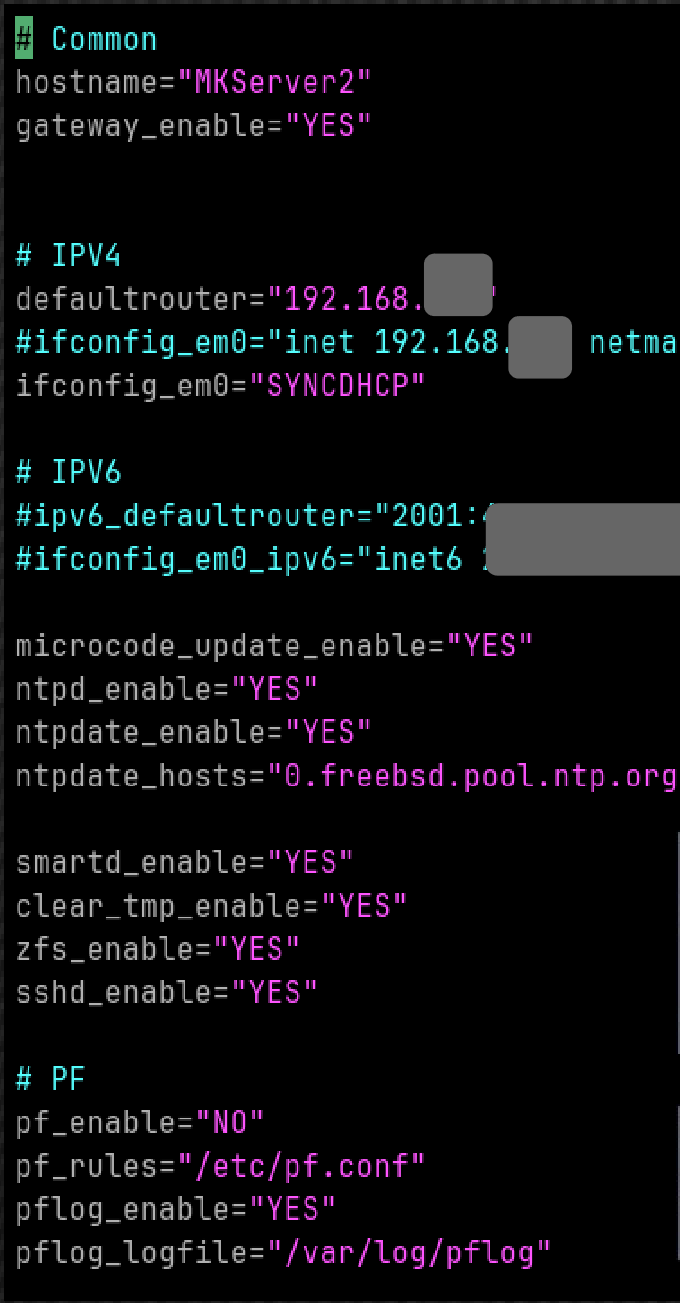
No systemd

Dmesg as normal user



```
Looking up update.FreeBSD.org mirrors... 3 mirrors found.
Fetching metadata signature for 14.2-RELEASE from update1.freebsd.org... done.
Fetching metadata index... done.
load: 1.04  cmd: grep 20321 [running] 6.94r 6.93u 0.00s 49% 1972k
load: 1.04  cmd: grep 20323 [running] 0.18r 0.17u 0.00s 1% 1940k
Inspecting system...
load: 1.04  cmd: sha256 20522 [running] 0.04r 0.03u 0.00s 0% 2280k
done.
▯
```

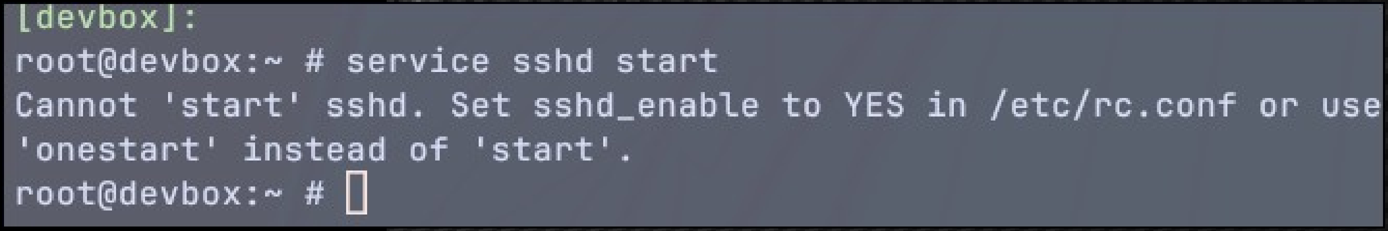
03 A different command line world



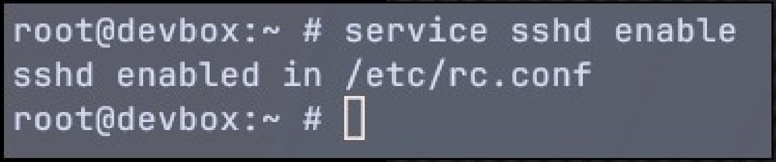
```
# Common
hostname="MKServer2"
gateway_enable="YES"


# IPV4
defaultrouter="192.168.[redacted]"
#ifconfig_em0="inet 192.168.[redacted] netmask 255.255.255.255"
ifconfig_em0="SYNCDHCP"

# IPV6
#ipv6_defaultrouter="2001:[redacted]
#ifconfig_em0_ipv6="inet6 [redacted]"

microcode_update_enable="YES"
ntpd_enable="YES"
ntpdate_enable="YES"
ntpdate_hosts="0.freebsd.pool.ntp.org"

smartd_enable="YES"
clear_tmp_enable="YES"
zfs_enable="YES"
sshd_enable="YES"

# PF
pf_enable="NO"
pf_rules="/etc/pf.conf"
pflog_enable="YES"
pflog_logfile="/var/log/pflog"
```

Manage services / settings in rc.conf

“One” config file philosophy

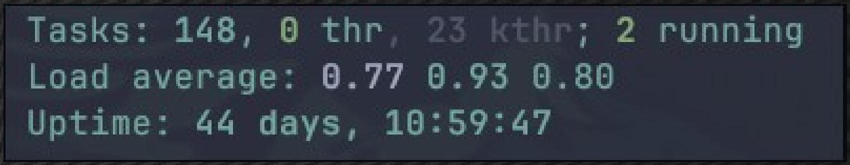


```
[devbox]:
root@devbox:~ # service sshd start
Cannot 'start' sshd. Set sshd_enable to YES in /etc/rc.conf or use
'onestart' instead of 'start'.
root@devbox:~ #
```



```
root@devbox:~ # service sshd enable
sshd enabled in /etc/rc.conf
root@devbox:~ #
```

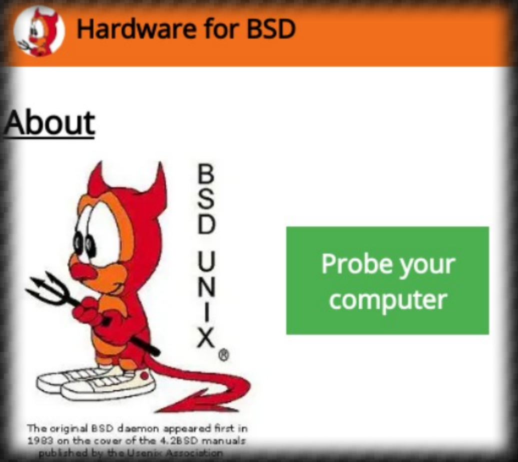
04 FreeBSD as Server



```
Tasks: 148, 0 thr, 23 kthr; 2 running
Load average: 0.77 0.93 0.80
Uptime: 44 days, 10:59:47
```

- I use older hardware and also some newer small client (N100 based for example)

  Until now I could install without issues: https://bsd-hardware.info
- ZFS is default: snapshots are a dream
- Process and jail isolation is a good security solution
- I replaced Proxmox by one FreeBSD server:
  - > 6 jails using Bastille as management tool
  - Bhyve for Linux Vms (Debian etc)



Hardware for BSD

About

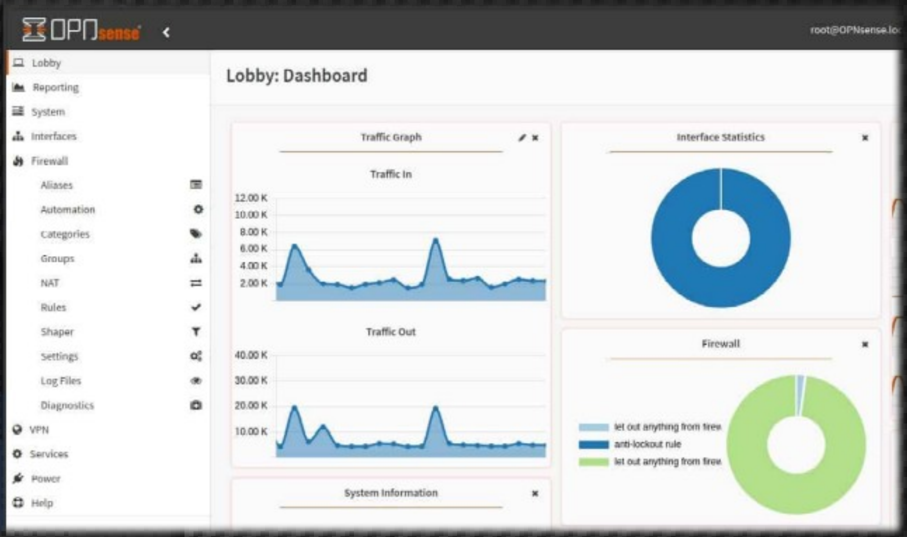
The original BSD daemon appeared first in 1983 on the cover of the 4.2BSD manuals published by the Usenix Association

04 FreeBSD as Server

- I replaced my OS on my NAS

  Using ZFS and 4 disks

  Restic as backup tool



- I use OPNsense as firewall (FreeBSD based)

- FreeBSD on Raspberry Pi and Zero?

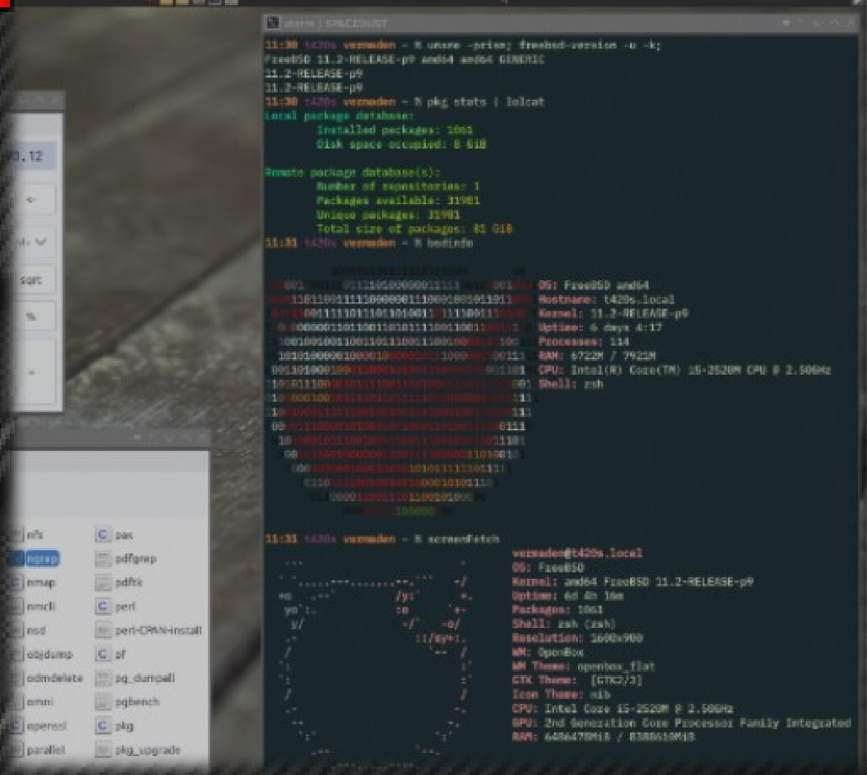
  Great! Running UFS on a R/O SD card

04 FreeBSD as Desktop

- I have a Thinkpad X1 Carbon
- FreeBSD works fine as desktop
  - Gnome / KDE / Sway / Hyprland
  - Not a straightforward install

    Takes some time to configure

    See https://vermaden.wordpress.com/freebsd-desktop/
- but…



```
11:30 t420s vermaden ~ % uname -prism; freebsd-version -u -k;
FreeBSD 11.2-RELEASE-p9 amd64 amd64 GENERIC
11.2-RELEASE-p9
11.2-RELEASE-p9
11:30 t420s vermaden ~ % pkg stats | lolcat
Local package database:
        Installed packages: 1061
        Disk space occupied: 8 GiB

Remote package database(s):
        Number of repositories: 1
        Packages available: 31981
        Unique packages: 31981
        Total size of packages: 81 GiB
11:31 t420s vermaden ~ % bsdinfo
                                  OS: FreeBSD amd64
                                  Hostname: t420s.local
                                  Kernel: 11.2-RELEASE-p9
                                  Uptime: 6 days 4:17
                                  Processes: 114
                                  RAM: 6722M / 7921M
                                  CPU: Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz
                                  Shell: zsh

11:31 t420s vermaden ~ % screenfetch
                               vermaden@t420s.local
                               OS: FreeBSD
                               Kernel: amd64 FreeBSD 11.2-RELEASE-p9
                               Uptime: 6d 4h 16m
                               Packages: 1061
                               Shell: zsh (zsh)
                               Resolution: 1600x900
                               WM: OpenBox
                               WM Theme: openbox_flat
                               GTK Theme:  [GTK2/3]
                               Icon Theme: nib
                               CPU: Intel Core i5-2520M @ 2.50GHz
                               GPU: 2nd Generation Core Processor Family Integrated
                               RAM: 6486478MiB / 8388610MiB
```

04 FreeBSD as Desktop

but…

- I need a good Wifi adapter to get the right speed

  There is WifiBox to containerize Linux and use those drivers via routing

- Bluetooth support is a no go, very limited,
  only a Headset implementation

- The knobs for light and sound and more need to be setup

- It is ideal for 90% of the time

--> Debian / Arch or any other Linux works out of the Box
NixOS: great OS for development and experimenting

So… who knows the laptop project brings light: “Deliver a package of improved or new FreeBSD functionality that, together, will ensure that it runs well “out of the box” on a broad range of personal computing devices.”

04 FreeBSD for SW Development

- A lot of base repositories use a Docker setup: Docker ~ Podman but with a learning curve
- For PlatformIO and C++ the libraries are named differently and sometimes even not available for FreeBSD or are not maintained
- Developing from scratch is easier: you can select your own libraries or develop / adopt them
- OK there is
  - Linuxulator (Linux Emulation) Linux binary compatibility is a mechanism to run unmodified Linux binaries under FreeBSD: It works but has a lot of limits.
    Anything that requires systemd dependency will probably not work
  - NDISulator (for Windows (uch) drivers)

but aaargh…. It is doable but…. For my use NixOS is super

04 FreeBSD jails and Linux Docker

Same goal: lightweight persistent virtualization

Docker: running prepared binary images from various public or private repositories.

FreeBSD jails are more like LXC in Linux: it's a method of creating your container from scratch. You create it, install the software into it.

04 Jails (1)

- Jails were developed as a tool for system administrators to enhance the security of a FreeBSD system
- Jails were created to expand upon the chroot concept, which is used to change the root directory of a set of processes
- A jail is characterized by four elements:
  - A directory subtree: the starting point from which a jail is entered. Once inside the jail, a process is not permitted to escape outside of this subtree.
  - A hostname: which will be used by the jail.
  - An IP address: which is assigned to the jail. The IP address of a jail is often an alias address for an existing network interface.
  - A command: the path name of an executable to run inside the jail.

04 Jails (2)

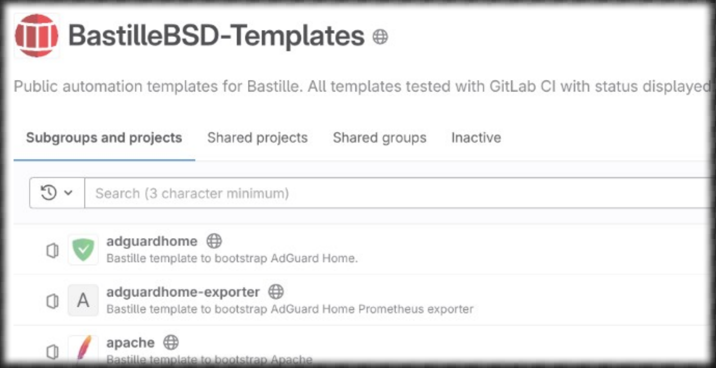
- Typical use cases: DNS server, Mail server, a secure browser, testing and experimenting and blocky DNS Proxy

- Jail command can be used without installation
- Thick jails: good for isolation yet heavy on maintenance and manual work
- Thin jails: seize the benefits of ZFS datasets, and set default configuration on the dataset. Every other jail clones the dataset and then adds the jails packages and configuration on its own dataset.
- BastilleBSD as tool makes life easier

```
bastille create -V devbox2 14.2-RELEASE 192.168.1.21 em0

-V = VNET  and em0 = interface
```

04 Jails / BastilleBSD (3)

- One config file for the definition of jails:
  - Templating:
    Gitea install & add nginx as HTTP server on top
  - Privileges / devfs rules / port sharing
  - Security levels / mounts / paths
  - Networking: Shared / VNET / Bridged VNET
    https://alfaexploit.com/en/posts/managing_jails_in_freebsd_with_bastille
- Linux Jails are also possible athough limited and experimental



BastilleBSD-Templates

Public automation templates for Bastille. All templates tested with GitLab CI with status displayed

adguardhome
Bastille template to bootstrap AdGuard Home.

adguardhome-exporter
Bastille template to bootstrap AdGuard Home Prometheus exporter

apache
Bastille template to bootstrap Apache

For Linux VMs

- Bhyve-VM is great!

04 VM Bhyve

- Management system for FreeBSD bhyve virtual machines
- Some of the main features include:
  - Windows/UEFI support
  - Simple commands to create/start/stop bhyve instances
  - Simple configuration file format
  - Virtual switches supporting vlans & automatic device creation
  - ZFS support
  - FreeBSD/MidnightBSD/NetBSD/OpenBSD/Linux guest support

```
vm create -c 2 -m 2G -S 15G -t Linux-zvol debian
```

Note:
A bit resource hungry
Serial port sharing is tricky

05 Final Words / Demo

- The world of Linux and BSD have both their own specifics, usecases and pittfalls
- I am not a fanatic (BSD) geek but I would like to use one platform ofcourse….
  ”apt update” or “pkg update” mixups ;)
- A short Demo after the talk on my heavy 2012 Macbook Pro
  8gb RAM / SSD / Single boot

  (boot logo is very slow, the rest is ok and yes a Wifi Adapter)

```
$ neofetch
```
                                  `
  ` `.....---......--.```   -/   jeremy@digitalstorm
 +o   .--`         /y:`   +.   -------------------
  yo`:.            :o      `+-   OS: FreeBSD 12.1-RELEASE-p2 amd64
   y/   -/`   -o/   Uptime: 15 hours, 15 mins
  .-   ::/sy+:.   Packages: 823 (pkg)
  /   `-- /   Shell: sh
 `:                          :`   Resolution: 1920x1080, 1920x1080
 `:                          :`   DE: LXDE
  /   /   WM: Openbox
  .-   -.   WM Theme: Onyx
   --   -.   Theme: Adwaita [GTK3]
    `:`   `:`   Icons: Adwaita [GTK3]
      .--   `--.   Terminal: lxterminal
         .---.....----.   Terminal Font: Monospace 15
                                   CPU: Intel i9-9920X (24) @ 3.504GHz
                                   GPU: TU117 [GeForce GTX 1650]
                                   Memory: 3910MiB / 65150MiB

$
```

# Thank You!
```

Backup slides

Linux vs FreeBSD Kernel

Linux is more like a Kernel

Userland and Kernel development is split

Uses often SystemD

The kernel used in Linux is Monolithic

FreeBSD is a complete Operating System – Kernel and Userland developed by the same teams

Its kernel type is Monolithic with modules

Linux vs FreeBSD Packages

Its package management depends on the distribution

Debian/Ubuntu: often prebuilt

Third party repos

Arch: Huge: compile time

NixOS: Huge # 120.000 compile time

Stable / Unstable system / Pre-built

Its source package management is Port collections/packages (compiled sources)

Linux vs FreeBSD
Performance / Maintenance

Good Performance

Good to maintain

Sometime a conflicting update or when cleaning up

(when using multiple repos from often various sources)

Consistent and good performance

Easy to maintain

No conflicting upgrades

Stable

Linux vs FreeBSD

Security / Hardening

Linux has good security

Firewalls

Docker for isolation

Many many users and bug hunters

FreeBSD has better security than Linux

“famous” PF firewall

Jails: process isolation

Podman as Docker equivalent

Less vulnerable to common attacks

Linux vs FreeBSD

Hardware support

Works often out of the box without extra drivers

Server hardware support is excellent

Nvidia: nah…

Intel and others: ok

Wifi, some adapters: slow speeds and less supported HW

(broadcom)

(note: Laptop project is launched)