

im·mu·ta·ble **FreeBSD** ĭ-myooō'tə-bəl

Experiments building & running immutable infra

*Not subject or susceptible to change.*

*Coimbra*

*EuroBSDcon 2023*

# ~ whoami

---

dch@skunkwerks.at

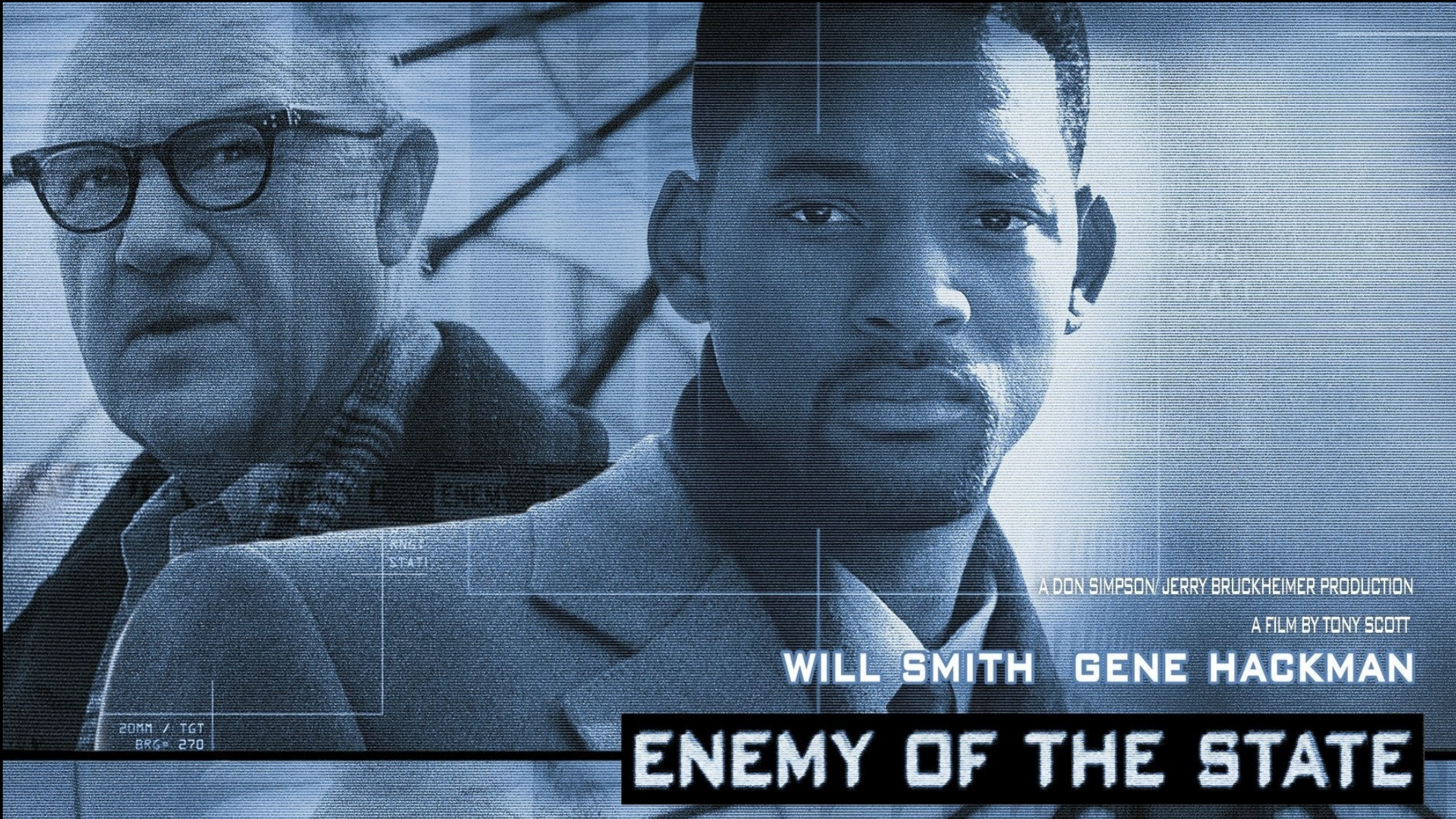


@dch@bsd.network

<https://people.FreeBSD.org/~dch>

Automater of Things

Lazy Ops Engineer



ENEMY  
OF THE  
STATE

A DON SIMPSON / JERRY BRUCKHEIMER PRODUCTION

A FILM BY TONY SCOTT

**WILL SMITH GENE HACKMAN**

**ENEMY OF THE STATE**

20MM / TGT  
BRG 270

# Principles

---

***“idempotent, repeatable, composable, loosely coupled”***

FreeBSD is ideally suited to immutable infrastructure, with powerful primitives

- jails, zfs, boot environments
- poudriere for building complete systems

**minimise runtime tooling and ops effort**

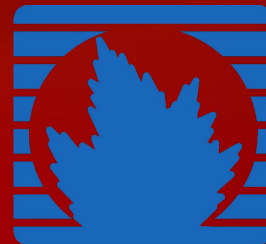
- preferring up-front dev effort
- let the network do the heavy lifting
- minimise the moving parts
- automate the deploys
- assume fungible hardware

# Plumbing

- Anycast, BGP
- Load balancers
- Mesh VPN and DNS



klara



Juniper<sup>®</sup>  
NETWORKS

# Networking – it just works™

---

- AnyCast or GeoDNS + healthcheck failover
- 3 global regions (EU, Americas, Asia)
- ISP router provides iBGP within region to servers
- Servers run haproxy to jails
- Jails are linked via ipv6 mesh network



EQUINIX

# BGP using bird

---

```
1 # http://bird.network.cz/
2
3 filter packet_bgp {
4     if net = {{ net.bgp.public_ipv4 }}/32 then accept;
5 }
6
7 protocol bgp {
8     export filter packet_bgp;
9     local as {{ net.bgp.local_as }};
10
11     source address {{ net.private.ipv4 }};
12     neighbor {{ net.bgp.neighbor }} as {{ net.bgp.upstream_as }};
13     password "{{ net.bgp.md5_password }}";
14 }
```

# The Load Balancer

- Present on each server
- Starts before bird BGP announcer
- sends traffic to nearest “up” jail even if not local
- haproxy has awesome lua integration

```
397 backend couch_be
398   option      httpchk      GET /_up
399   http-check  expect      status 200
400   http-check  disable-on-404
401   # prefer front end nodes for consistent performance and less race
402   # condition risk from concurrent activities on front end nodes
403   # these vars are set in either group_vars/all.yml or overridden in
404   # host_vars/*.yml
405   server      c01_couch [{{ config.couchdb.nodes.c01 }}]:{{ config.couchdb.port }} check observe layer7
406   server      c02_couch [{{ config.couchdb.nodes.c02 }}]:{{ config.couchdb.port }} check observe layer7
407   server      c03_couch [{{ config.couchdb.nodes.c03 }}]:{{ config.couchdb.port }} check observe layer7 back
```



# Load Balancers have 2 sides

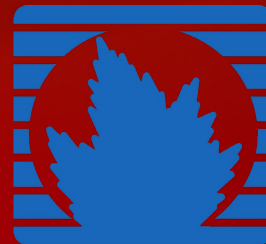
```
391 frontend couch_fe
392   bind                {{ net.private.ip.haproxy }}:5984
393   http-request        add-header    x-forwarded-port      %[dst_port]
394   http-response       add-header    x-couch                {{ inventory_hostname }}
395   default_backend     couch_be
396
397 backend couch_be
398   option              httpchk      GET /_up
399   http-check          expect       status 200
400   http-check          disable-on-404
401   # prefer front end nodes for consistent performance and less race
402   # condition risk from concurrent activities on front end nodes
403   # these vars are set in either group_vars/all.yml or overridden in
404   # host_vars/*.yml
405   server              c01_couch [{{ config.couchdb.nodes.c01 }}]:{{ config.couchdb.port }} check observe la
406   server              c02_couch [{{ config.couchdb.nodes.c02 }}]:{{ config.couchdb.port }} check observe la
407   server              c03_couch [{{ config.couchdb.nodes.c03 }}]:{{ config.couchdb.port }} check observe la
```

# Jails

- How to find the jails
- Immutability
- Deployment



klara



Juniper<sup>®</sup>  
NETWORKS

# Exposing Jail State to LBs

```
curl -s localhost:8000 | jq '.jail-information[][] | select(.name=="www")'
```

```
1  #!/bin/sh
2
3  while ;; do
4      json=$(jls -vd --libxo json)
5      length=$(echo $json | wc -c)
6      printf "HTTP/1.1 200 OK\r\nContent-Length: %s\r\n\r\n%s" \
7          $length "$json" \
8          | nc -lN 8000
9      echo ok
10 done
```

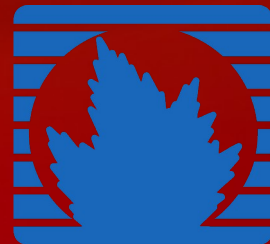
```
{
  "jid": 6,
  "hostname": "www",
  "path": "/jails/instances/13.2-RELEASE-arm64-aarch64/www",
  "name": "www",
  "state": "ACTIVE",
  "cpusetid": 5,
  "ipv4_addrs": [
    "100.64.186.216"
  ],
  "ipv6_addrs": [
    "fca2:927d:4d9b:bbdb:fdd2::bad8"
  ]
}
```

# Apps

- Immutability
- Packaging
- Deployment



klara



Juniper®  
NETWORKS

# Immutable Apps – a Study

---

- 2 web servers tested
- 8 databases tested
- many custom applications
  
- a general approach emerges

# Immutable Apps – a Study

---

- Databases are much trickier
- More mutable state
- Harder to load balance
- Lots of zfs tricks
  - FoundationDB, CouchDB, MariaDB, Postgresql, Graylog, MongoDB, ElasticSearch, OpenSearch

# Jail & split (im)mutable data

```
dch@continuity /> zfs list -o mounted,canmount,jailed,name,mountpoint |grep -i jail | column -t
```

MOUNTED	CANMOUNT	JAILED	NAME	MOUNTPOINT
no	off	off	zroot/jailed	none
no	on	on	zroot/jailed/forgejo	/var/db/forgejo
yes	on	on	zroot/jailed/hedgedoc	/var/db/hedgedoc
no	on	on	zroot/jailed/postgres	/var/db/postgres
no	on	on	zroot/jailed/softserve	/var/db/softserve
yes	on	on	zroot/jailed/sync	/var/db/sync
yes	on	off	zroot/jails	/jails
no	off	off	zroot/jails/downloads	/jails/downloads
yes	on	off	zroot/jails/downloads/13.2-RELEASE-arm64-aarch64	/jails/downloads/13.2-RELEASE-arm64-aarch64
no	off	off	zroot/jails/instances	/jails/instances
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64	/jails/instances/13.2-RELEASE-arm64-aarch64
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64/hedgedoc	/jails/instances/13.2-RELEASE-arm64-aarch64/hedgedoc
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64/invidious	/jails/instances/13.2-RELEASE-arm64-aarch64/invidious
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64/meringovia	/jails/instances/13.2-RELEASE-arm64-aarch64/meringovia
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64/sync	/jails/instances/13.2-RELEASE-arm64-aarch64/sync
yes	on	off	zroot/jails/instances/13.2-RELEASE-arm64-aarch64/www	/jails/instances/13.2-RELEASE-arm64-aarch64/www
no	off	off	zroot/jails/templates	/jails/templates
yes	on	off	zroot/jails/templates/13.2-RELEASE-arm64-aarch64	/jails/templates/13.2-RELEASE-arm64-aarch64

# Immutable Apps – zfs magic

- Use jailed zfs nested containers

JAILED	CANMOUNT	MOUNTED	MOUNTPOINT	NAME
on	off	no	/var/db	zroot/jailed/graylog_db
on	on	yes	/var/db/graylog	zroot/jailed/graylog_db/graylog
on	on	yes	/var/db/mongodb	zroot/jailed/graylog_db/mongodb
on	on	yes	/var/db/opensearch	zroot/jailed/graylog_db/opensearch

- Use **.zfs/snapshot/\$NAME** for backups

```
root@i09 /u/h/dch# cd /graylog/var/db/graylog/.zfs/snapshot/
root@i09 /g/v/d/g/.z/snapshot# l
total 43
drwxr-x--- 4 848 848    5B Jul  7 08:49 20230831-1425:13.2-RELEASE-p1/
drwxr-x--- 4 848 848    5B Jul  7 08:49 20230906-1010:13.2-RELEASE-p2/
drwxr-x--- 4 848 848    5B Jul  7 08:49 20230907-1134:13.2-RELEASE-p2/
drwxr-x--- 4 848 848    5B Jul  7 08:49 daily-2023-09-10/
drwxr-x--- 4 848 848    5B Jul  7 08:49 daily-2023-09-11/
drwxr-x--- 4 848 848    5B Jul  7 08:49 daily-2023-09-12/
drwxr-x--- 4 848 848    5B Jul  7 08:49 daily-2023-09-13/
```



# Immutable Tricks – Summary

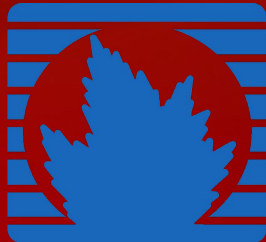
---

- finagle all the config files
- unix sockets & softlinks for /tmp, /var/run etc
- move syslog to network service
- nested zfs datasets for custom perf & tuning
- zfs diff to find mutable locations
- zfs read-only once complete

# Container Deploys



klara



Juniper<sup>®</sup>  
NETWORKS

# Deploying Containers

## What is webhook? build passing



[webhook](#) is a lightweight configurable tool written in Go, that allows you to easily create HTTP endpoints (hooks) on your server, which you can use to execute configured commands. You can also pass data from the HTTP request (such as headers, payload or query variables) to your commands. [webhook](#) also allows you to specify rules which have to be satisfied in order for the hook to be triggered.

- App source code in git repo
- Push code → generate HMAC signed webhook
- haproxy further restricts webhook origin via mTLS & route protection
- Webhook daemon checks HMAC
  - runs CI script and builds new package
  - requests pkg-based deploy

# HMAC signed Webhooks for CI

```
34 # github auto-deploys for internal projects
35 - id: github
36   execute-command: '{{ config.ci.dir }}/src/ansible/github.sh'
37   command-working-directory: '{{ config.ci.dir }}/src/ansible'
38   include-command-output-in-response: false
39   pass-environment-to-command:
40     - envname: CI_REPO           52
41       name:    repository.full_name 53
42       source:  payload             54
43     - envname: CI_REF           55
44       name:    head_commit.id      56
45       source:  payload             57
46     - envname: CI_USERNAME      58
47       name:    pusher.name         59
48       source:  payload             60
49     - envname: CI_USERMAIL      61
50       name:    pusher.email        62
51       source:  payload             63
52                                     64
53                                     65
54   trigger-rule:
55     and:
56       - match:
57         type: payload-hmac-sha1
58         secret: '{{ config.ci.github_hmac_secret }}'
59         parameter:
60           source: header
61           name: X-Hub-Signature
62       - match:
63         type: value
64         value: refs/heads/main
65         parameter:
66           source: payload
67           name: ref
```

# Bonus: Arbitrary plays via Webhook

```
1 ---
2 # vim: filetype=yaml
3 # docs at https://github.com/adnanh/webhook/wiki
4 - id: ansible
5   execute-command: '{{ config.ci.dir }}/src/ansible/deploy.sh'
6   command-working-directory: '{{ config.ci.dir }}/src/ansible'
7   include-command-output-in-response: false
8   pass-arguments-to-command:
9     - source: 'payload'
10       name: 'play'
11   trigger-rule:
12     and:
13       # ensures payload is secure -- headers are not trusted
14       - match:
15           type: payload-hmac-{{ config.ci.cabal_hmac_algorithm }}
16           secret: {{ config.ci.cabal_hmac_secret }}
17           parameter:
18             source: header
19             name: x-cabal-signature
20       # allows routing via haproxy
21       - match:
22           type: value
23           value: ansible
24           parameter:
25             source: header
26             name: x-cabal-daemon
```

# Using pkg-create(8)

```
1 name:      indie
2 origin:    indie/indie
3 comment:   "Zen practice is to open up our small mind – Shunryu Suzuki"
4 arch:      freebsd:13:x86:64
5 www:       https://github.com/indiesites/indie
6 maintainer: root@indiesites.org
7 prefix:    /usr/local
8 licenselogic: single
9 licenses:  [MIT]
10 categories: [indie]
11 conflict:  indie-★
12 deps:      {}
13 flatsize:  0
14 options:   {git: "GITSHA"}
15 desc:      <<EOD
16 Zen is a liberation from time. For if we open our eyes and see clearly, it
17 becomes obvious that there is no other time than this instant, and that
18 the past and the future are abstractions without any concrete reality.
19
20 – Alan Watts
21 EOD
22
23 message:   <<EOM
24 Zen is an effort to become alert and awake – Osho.
```

# pkg-create(8) and pkg-sign(8)

```
$ pkg create --verbose \  
--root-dir ${STAGING} \  
--manifest ${MANIFEST} \  
--out-dir ${BUILD}  
...  
# cp ${ARTEFACT} /ci/var/db/ci/pkg/  
# pkg repo -o \  
  /ci/var/db/ci/pkg \  
  /ci/var/db/ci/pkg \  
  /usr/local/etc/ssl/keys/pkg.key  
...  
# jexec $JAIL pkg upgrade -y $foo
```

# Apps Summary

---

- Immutable containerised apps via
  - zfs readonly clone of template “/” for jail
  - nullfs RO mounts for config and www data
  - zfs nested datasets for mutable databases
  - syslog-ng outside jail
  - no internal daemons (syslog, cron, ntp ..)
- Immutable deploys via webhooks and pkg-\* tools
- Load balancers and networks make this invisible

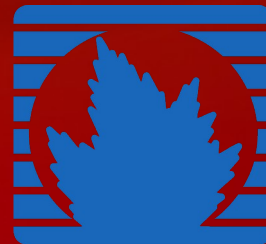


# Immutable Servers

- ZFS Boot Envs
- Poudriere
- SyncBE deploy
- tarfs(8)



klara



Juniper®  
NETWORKS

# ZFS Boot Envs

- clone a snapshot of your “/”
- mount it, and edit or update it
- test it in a jail
- activate it and reboot
- woops, roll back, phew!
- ***app data*** is separate and intact
  
- uses zfs properties:
  - zroot/ROOT/...
  - canmount=noauto
  - mountpoint=/
- uses zpool property:
  - bootfs=zroot/ROOT/yolo

```
dch@akai ~> doas bectl list
BE
13.2-RELEASE-p1_2023-08-18_010858 - - 1012K 2023-08-18
13.2-RELEASE-p2_2023-08-18_011113 - - 2.08M 2023-08-18
13.2-RELEASE-p2_2023-09-10_213046 - - 468M 2023-09-10
current - - 16.8G 2023-09-10
thirteen NR / 18.7G 2023-09-10
dch@akai ~> doas bectl activate -t current
Successfully activated boot environment current
for next boot
dch@akai ~> doas bectl list
BE
13.2-RELEASE-p1_2023-08-18_010858 - - 1012K 2023-08-18
13.2-RELEASE-p2_2023-08-18_011113 - - 2.08M 2023-08-18
13.2-RELEASE-p2_2023-09-10_213046 - - 468M 2023-09-10
current T - 16.8G 2023-09-10
thirteen NR / 18.7G 2023-09-10
```

# Loader Prompt – IPMI supported

FreeBSD

Welcome to FreeBSD

1. Active: zfs:zroot/ROOT/test
2. bootfs: zfs:zroot/ROOT/default
3. Page: 1 of 1

Boot Environments:

4. test
5. default



# poudriere-(devel)

---

- Build FreeBSD from src
- Build packages from ports tree
- Great doc coverage on wiki
- Build deployable images in many formats
  - memstick, iso
  - zfs dataset
  - tarball

# Inputs

---

- git source & ports tree
- overlay directory for images
  - boot/loader.conf
  - etc/fstab
  - etc/rc.conf.d/sshd
  - etc/resolv.conf
  - usr/local/bin/sync-be \*\*
  - usr/local/etc/pkg/repos/FreeBSD.conf
- a list of packages we want to build
  - sysutils/spiped
  - sysutils/tmux
  - ...

# Usage – OS + Package Build

```
# poudriere jail -c -j 13_2_builder_amd64 \  
  -v releng/13.2 \  
  -m git+https \  
  -b -K GENERIC  
  
# poudriere bulk -j 13_2_builder_amd64 \  
  -f ./packages.lst
```

# Usage – Image Build

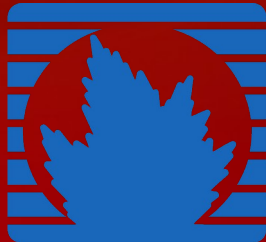
---

```
# poudriere image -t zfs+send+be \  
-j 13_2_builder_amd64 \  
-f ./packages.lst \  
-s 4G \  
-h '' \  
-o /usr/local/poudriere/images/ \  
-c overlay \  
-n ${IMAGE}
```

# Server Deploys



klara



Juniper®  
NETWORKS



# Deploy – curl → BE

---

```
# bectl list
```

```
BE Active Mountpoint Space Created
```

```
13.1-RELEASE_2023-03-21_152313 - - 836K2023-03-21 15:23
```

```
default NR / 2.24G 2023-03-21 13:49
```

```
# curl -#L https://pkg/images/be202303262144.be.zfs \  
| /usr/local/bin/sync-be 13.2-RELEASE /etc/syncbe.conf
```

```
using config file: /etc/syncbe.conf
```

```
receiving full stream of zroot.356600197/ROOT/default@202303262144 \  
into zroot/ROOT/13.2-RELEASE@202303211523
```

```
##### 70.1%
```

```
...
```

```
received 1.77G stream in 35 seconds (51.7M/sec)
```

```
...
```

# Config Hacking

---

- Same tricks as usual
  - softlinking mutable dirs out into a separate location
  - read-only zfs datasets
  - unix sockets everywhere, or network services
  - nullfs mounts to clean things up
- Works for Appliances, less for Generic Servers
- Dammit.

# Enter sync-be

---

- Klara Systems tool
  - creates a new boot env
  - from your stdin-supplied zfs
  - mounts it temporarily
  - transfers in your local /etc/ /usr/local/\* changes
  - unmounts the BE
  - temporarily activates it

# Deploy – pristine BE → existing state

```
...
copying boot/loader.conf to /tmp/QilKale4/boot/loader.conf
copying boot/loader.conf.d to /tmp/QilKale4/boot/loader.conf.d
copying etc/login.conf.db to /tmp/QilKale4/etc/login.conf.db
copying etc/pwd.db to /tmp/QilKale4/etc/pwd.db
copying etc/spwd.db to /tmp/QilKale4/etc/spwd.db
...
copying root to /tmp/QilKale4/root
zfs bootenv is successfully written
ready for reboot!

# reboot
```

# tarfs(8)

---

- Mount a tarball as a (readonly) filesystem
- Can be jailed & nullfs-mounted
- Built by Klara Systems and Juniper Networks
- Coming in 14.0-RELEASE
- May not be as fast as other filesystems yet
- Only supports plain tarball, or tar+zstd only

# tarfs(8) in action

---

```
# unxz < /dl/13.2-RELEASE-arm64-aarch64/base.txz \  
  > 13.2-RELEASE.tar  
  
# mkdir jail  
  
# mount -t tarfs 13.2-RELEASE.tar jail  
  
# mount -t devfs devfs jail/dev  
  
# mount -t tmpfs tmpfs jail/tmp  
  
  
# jail -cv name=tar path=jail command=/bin/sh  
jail_set(JAIL_CREATE) persist name=tar path=jail  
created  
run command in jail: /bin/sh
```

# tarfs(8) in action

---

```
... run command in jail: /bin/sh
```

```
# mkdir /coimbra
```

```
mkdir: /: No such file or directory
```

```
# df -h / /tmp /dev
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
13.2-RELEASE.tar	929M	929M	0B	100%	/
tmpfs	20G	4.0K	20G	0%	[restricted]
devfs	1.0K	0B	1.0K	0%	[restricted]

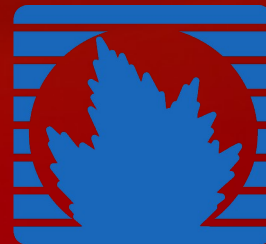
```
#
```

# Credits & Thanks

- malloc(questions[])
- free(&dave)
- madvise(\*social)



klara



Juniper<sup>®</sup>  
NETWORKS