Jetpack

A container runtime for FreeBSD

Maciej Pasternacki <maciej@3ofcoins.net>

FOSDEM 2016

@mpasternacki

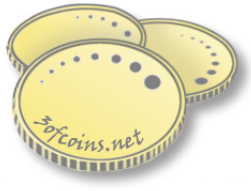
3ofcoins

Outline

Why?

What?

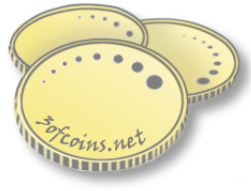
How?



Why?

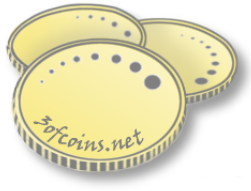
Containers are cool!

- Old technology, new paradigm
- Manage services, not whole systems
- Separate build from execution:
  RO, verifiable, distributable images;
  fast copy–on–write provisioning
- Separate valuable data from state
- Decent isolation, low overhead,
  resource sharing possible



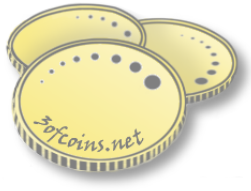
FreeBSD is cool!

- Jails, native ZFS, *pf* firewall, *DTrace*
- No *systemd*
- Mature & reliable
- Good engineering culture
- I just like it



FreeBSD is cool!

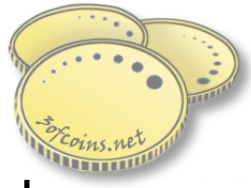
- Jails, native ZFS, *pf* firewall, *DTrace*
- No *systemd*
- Mature & reliable
- Good engineering culture
- I just like it
- No containers... (as of late 2014)



Maybe port Docker?

(as of late 2014)

- Linux–only[1]
- Monolythic architecture
- Incomplete & unclear documentation
- Fast, feature–oriented development
- Feels like overgrown prototype



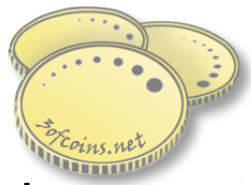
[1] Literal `iptables` invocations inlined in middle of code

Maybe port Docker?

(as of late 2014)

- Linux–only[1]
- Monolythic architecture
- Incomplete & unclear documentation
- Fast, feature–oriented development
- Feels like overgrown prototype

Nope.

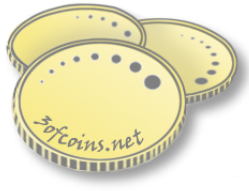


[1] Literal `iptables` invocations inlined in middle of code

Existing jail management tools?

(as of late 2014)

- All focused on managing whole system
- None properly utilized ZFS
- Most were multi–KLOC blobs of shell script

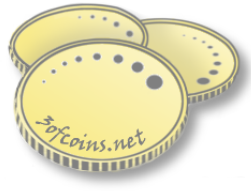


Existing jail management tools?

(as of late 2014)

- All focused on managing whole system
- None properly utilized ZFS
- Most were multi–KLOC blobs of shell script

Nope.

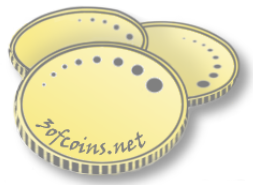


Conclusion: Prototype a jail management tool that I could use kind of like containers, and see what happens.

Suddenly: CoreOS Rocket

December 1st, 2014

- New container runtime by CoreOS
- Follows the neutral App Container Specification (*appc*)
- Designed for “composability, security, and speed”
- Linux-only (systemd–dependent)

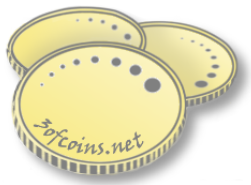


https://github.com/coreos/rkt

Suddenly: CoreOS Rocket

December 1st, 2014

- New container runtime by CoreOS
- Follows the neutral App Container Specification (*appc*)
- Designed for “composability, security, and speed”
- Linux-only (systemd–dependent)

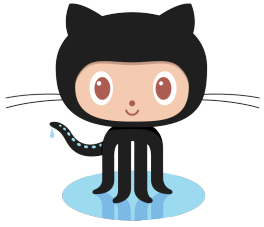


https://github.com/coreos/rkt

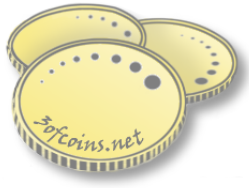
Jetpack

January 2015

App Container Specification implementation for FreeBSD



3ofcoins/jetpack

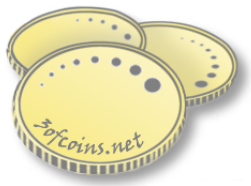


In the meantime...

FreeBSD Docker port

- Proof of concept released June 2015
- Last commit in July 2015
- Stuck at random Docker revision between 1.7 & 1.8.

Good luck keeping up with upstream!



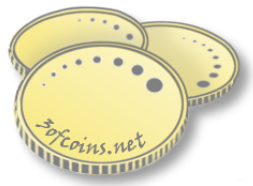
https://github.com/kvasdopil/docker

In the meantime...

Open Container Initiative

Founded in June 2015 to solve incompatible container runtimes by introducing a new standard.

This always works!



https://www.opencontainers.org/

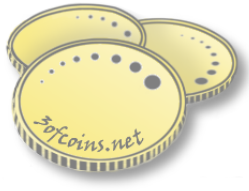
What?

App Container Specification

AKA *appc*

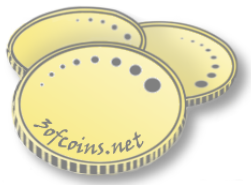
appc/spec

- Composable
- Secure
- Decentralized
- Open



App Container Image (*ACI*)

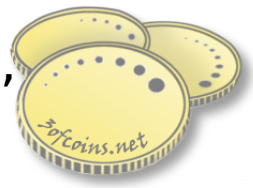
- A compressed *tar* file containing:
  - `manifest` JSON file
  - `rootfs/` directory
- Identified by SHA–512 checksum
- Addressed by *name* and a set of *labels*



https://github.com/appc/spec/blob/master/spec/aci.md

ACI Manifest

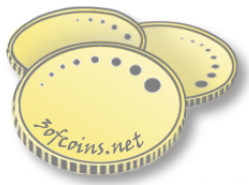
```
{ "acKind": "ImageManifest", "acVersion": "0.7.4",
  "name": "demo/fosdem2016/redis",
  "labels": [ { "name": "version", "value": "3.0.5_2" },
              { "name": "os",      "value": "freebsd" },
              { "name": "arch",    "value": "amd64" } ],
  "app": {
    "exec": [ "/usr/local/bin/redis-server",
              "/usr/local/etc/redis.conf"],
    "user": "redis",
    "group": "redis",
    "mountPoints": [ { "name": "data", "path": "/var/db/redis" } ],
    "ports": [ { "name": "redis", "protocol": "tcp", "port": 6379 } ]
  },
  "annotations": [
    { "name": "timestamp", "value": "2016-01-29T18:19:42Z" } ],
  "dependencies": [ {
      "imageName": "3ofcoins.net/freebsd-base",
      "imageID": "sha512-330a…f0a7",
      "labels": [ { "name": "version", "value": "10.2.8" },
                  { "name": "os",      "value": "freebsd" },
                  { "name": "arch",    "value": "amd64" } ]
  } ] }
```



App Container Image Discovery

From ACI name & labels to:

- ACI URL
- ACI Signature URL
- Public Key URL



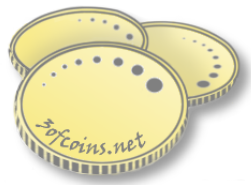
https://github.com/appc/spec/blob/master/spec/discovery.md

App Container Image Discovery

From ACI name & labels to:

- ACI URL
- ACI Signature URL
- Public Key URL

name	3ofcoins.net/freebsd-base
labels	version=10.1.12
	os=freebsd
	arch=amd64



https://github.com/appc/spec/blob/master/spec/discovery.md

App Container Image Discovery

name 3ofcoins.net/freebsd-base

labels version=10.1.12

os=freebsd

arch=amd64

https://3ofcoins-aci.s3.eu-central-1.amazonaws.com/...

.../3ofcoins.net/freebsd-base-10.1.12-freebsd-amd64.aci

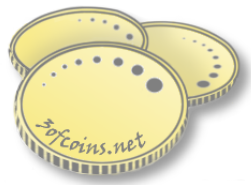
.../3ofcoins.net/freebsd-base-10.1.12-freebsd-amd64.aci.asc

.../aci-pubkeys.asc

appc Pods

A list of apps that will be launched together inside a shared execution context

- Shared PID space, network, IPC, hostname
- Separate filesystem root for each app
- Shared, persistent volumes
- Isolators

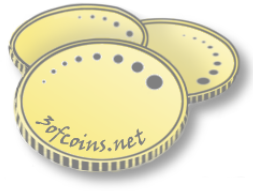


https://github.com/appc/spec/blob/master/spec/pods.md

Pod Manifest

template

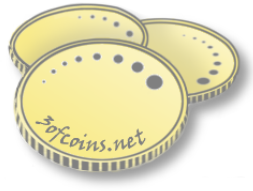
```
{ "acKind": "PodManifest", "acVersion": "0.7.4",
  "apps": [
    { "name": "redis",
      "image": { "name": "demo/fosdem2016/redis" },
      "mounts": [{ "volume": "redis", "path": "data" }] },
    { "name": "tipboard",
      "image": { "name": "demo/fosdem2016/tipboard" },
      "mounts": [{ "volume": "tipboard", "path": "data" }] }],
  "volumes": [
    { "name": "tipboard", "kind": "host", "readOnly": true,
      "source": "/home/japhy/FOSDEM2016-jetpack/demo/data" },
    { "name": "redis", "kind": "empty"  }
  ] }
```



Pod Manifest

reified

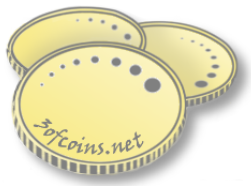
```
{ "acKind": "PodManifest", "acVersion": "0.7.4",
  "apps": [
    { "name": "redis",
      "image": { "name": "demo/fosdem2016/redis",
                 "id":   "sha512-7af6…a5fe" },
      "mounts": [{ "volume": "redis",    "path": "data" }] },
    { "name": "tipboard",
      "image": { "name": "demo/fosdem2016/tipboard",
                 "id":   "sha512-8578…c480" },
      "mounts": [{ "volume": "tipboard", "path": "data" }] }
  ],
  "volumes": [
    { "name": "tipboard", "kind": "host", "readOnly": true,
      "source": "/home/japhy/FOSDEM2016-jetpack/demo/data" },
    { "name": "redis",    "kind": "empty",
      "mode": "0755", "uid": 0, "gid": 0 }
  ],
  "annotations": [
    { "name": "ip-address", "value": "172.23.0.2" } ]
}
```



appc Executor

Executor Perspective

- Assigns pod UUIDs
- Renders apps' filesystems
- Sets up volumes
- Configures network
- Collects logs from stdout & stderr

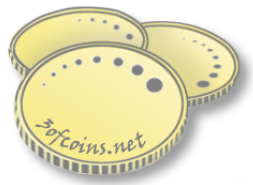


https://github.com/appc/spec/blob/master/spec/ace.md

appc Executor

App Perspective

- Environment variables, UID, GID, working directory as per image/pod manifest
- Resource isolation
- Access limits
- Metadata service

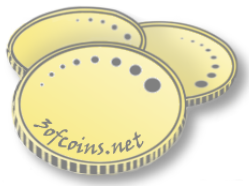


https://github.com/appc/spec/blob/master/spec/ace.md

appc Metadata Service

$AC_METADATA_URL/acMetadata/v1/...

- /pod/annotations/NAME
- /pod/manifest (fully reified)
- /pod/UUID
- /apps/*$AC_APP_NAME*/...
  - /annotations/NAME
  - /image/manifest
  - /image/id

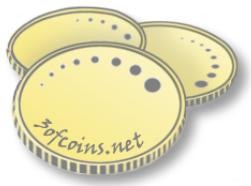


https://github.com/appc/spec/blob/master/spec/ace.md

appc Metadata Service

$AC_METADATA_URL/acMetadata/v1/...

- /pod/hmac/sign — POST to have ACE sign any data as this pod
- /pod/hmac/verify — verify another pod's (or own) signature on data



https://github.com/appc/spec/blob/master/spec/ace.md

How?

Jetpack

- Written mostly in Go
- Jails for process isolation & lockdown
- ZFS for layered storage
- Linux images supported via ABI emulation[1]
- Alpha, not suitable for production

[1] appc/docker2aci converts Docker images to ACI format

https://github.com/3ofcoins/jetpack/

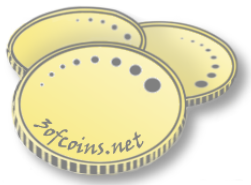
Jetpack: Use Cases So Far

- Pass *appc* validation suite
- Get console on a clean system
- Run a Minecraft server for myself and a friend[1] since summer
- Build some Omnibus packages[2]

[1] Yes, a real non-technical end user!

[2] Built the Chef Development Kit for FreeBSD

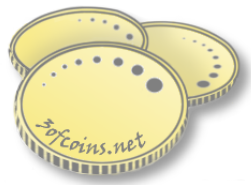
https://github.com/3ofcoins/jetpack/



Jetpack: ZFS Storage

`man zfs`? TL;DR

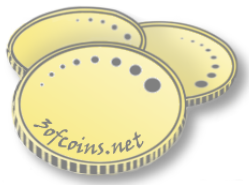
- ZFS pool consists of nested *datasets*
- You can take *snapshots* of a dataset
- You can *clone* a snapshot to create new datasets
- A cloned dataset *shares data* with the parent snapshot
  - Cloning a dataset is fast
  - Only new data written to a cloned dataset uses disk space



https://github.com/3ofcoins/jetpack/

Jetpack: ZFS Storage

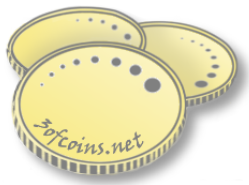
- Image's *rootfs* is a ZFS snapshot
- Child image's *rootfs* starts as parent's clone
- Pod app's *rootfs* is a dataset cloned from its image
- Each empty volume is a separate dataset



https://github.com/3ofcoins/jetpack/

Jetpack: Runtime

- Each pod is a `jail(2)`
- Each app is additionally `chroot(2)`–ed inside pod's jail
- Volumes are `nullfs(5)`–mounted into app's rootfs



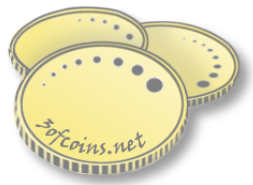
https://github.com/3ofcoins/jetpack/

Jetpack: Runtime

- No daemon with remote control API, `jetpack` binary does real work[1]
- Entering app context implemented as a shim in C
- Metadata service is separate binary, read–only, no root

[1] Yes, this means it needs root

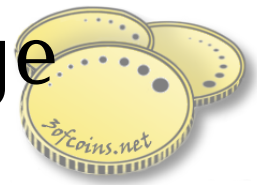
https://github.com/3ofcoins/jetpack/



Jetpack: Image Building

`jetpack build` *`IMAGE COMMAND ARGS…`*

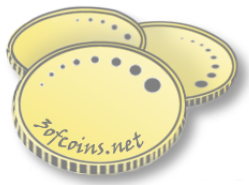
1. Clone build pod from parent `IMAGE`
2. Copy build dir (`./`) into pod
3. Run `COMMAND…` in the build pod, inside its copy of build dir
4. Get new manifest from pod's build dir
5. Remove build dir from pod
6. Snapshot pod's rootfs as new image



https://github.com/3ofcoins/jetpack/blob/master/IMAGES.md

Jetpack: Image Building

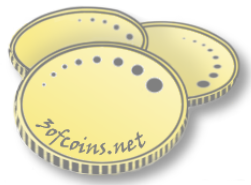
```
.MAKEFLAGS: -I${HOME}/Src/github.com/3ofcoins/jetpack/share

PARENT_IMAGE = 3ofcoins.net/freebsd-base
PKG_INSTALL = python27 py27-virtualenv libyaml

basedir=/opt/tipboard
projdir=${basedir}/home/.tipboard
build:
        virtualenv ${basedir}
        ${basedir}/bin/pip install tipboard
        install -m 0755 pre-start.sh ${basedir}/bin/pre-start.sh
        install -d ${basedir}/data ${projdir}
        install settings-local.py ${projdir}/settings-local.py
        ln -s /dev/null ${basedir}/home/tipboard.log
        install -m 0755 tipboard.sh /usr/local/bin/tipboard

manifest.json:
        ./manifest.json.sh > $@

.include "jetpack.image.mk"
```



https://github.com/3ofcoins/jetpack/blob/master/IMAGES.md

Jetpack: Image Building

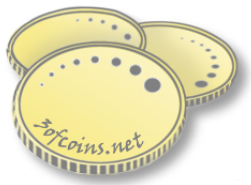
```
#!/bin/sh
set -e

version="$(tipboard --version)"
version="${version#Tipboard }"

cat <<EOF
{
  "name": "demo/fosdem2016/tipboard",
  "labels": [{ "name": "version", "value": "${version}" }],
  "app": {
    "exec": ["/usr/local/bin/tipboard", "runserver", "0.0.0.0", "7272"
    "eventHandlers": [
      { "name": "pre-start", "exec": [ "/opt/tipboard/bin/pre-start.sh
    "user": "www",
    "group": "www",
    "ports": [{ "name": "http", "protocol": "tcp", "port": 7272 }],
    "mountPoints": [{ "name": "data", "path": "/opt/tipboard/data" }]
  }
}
EOF
```



https://github.com/3ofcoins/jetpack/blob/master/IMAGES.md

Jetpack: Image Building

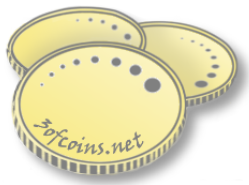
```
import os, os.path, urllib

execfile(os.path.expanduser("~/.tipboard/settings.py"))

AC_MDS_BASE = os.getenv('AC_METADATA_URL') + '/acMetadata/v1'
REDIS_HOST = urllib.urlopen(
    AC_MDS_BASE+'/pod/annotations/ip-address').read()
REDIS_PORT = 6379
```



https://github.com/3ofcoins/jetpack/blob/master/IMAGES.md

Jetpack: TODO

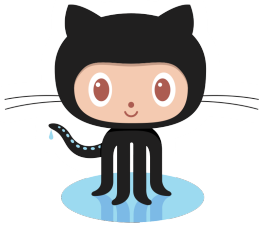
- Isolators
- *pf* anchor management
- Better UI: commands, output
- Boring stuff: docs, more acceptance tests
- Logging
- Sandbox desktop applications



https://github.com/3ofcoins/jetpack/

Demo time!

Questions?



3ofcoins/jetpack