An Operator Centric Way to Update Application Containers

tycho@tycho.ws, tycander@cisco.com
github.com/tych0

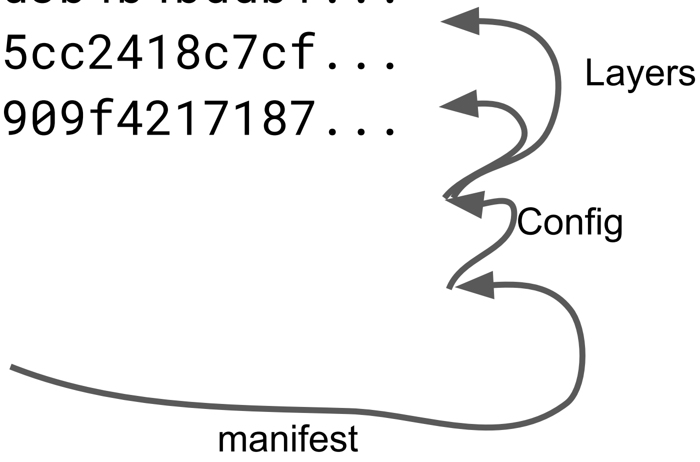
CISCO™

Stay awhile and listen...

- System containers
    - LXC: tarballs
    - OpenVZ: Ploop https://wiki.openvz.org/Ploop
    - Building rootfses generally painful
- Application Containers
    - Docker: layers of tarballs
    - OCI: layers of tarballs
    - Building rootfses generally easy

OCI format basics

```
oci
├── blobs
│   └── sha256
│       ├── 020351670a035e...
│       ├── 17cd3b4b4bddb1...
│       ├── f205cc2418c7cf...
│       └── fab909f4217187...
├── index.json
└── oci-layout
```



OCI format basics

- Each layer is a tar(.gz) file

OCI format drawbacks

- Each layer is a tar(.gz) file
  - No dedup
  - Whiteouts are painful (.wh.foo)
  - Large layers are painful
  - https://www.cyphar.com/blog/post/20190121-ociv2-images-i-tar

What do we actually want?

- Image Provenance
  - Signatures at build time
- Auditability
  - Same signatures can be verified at run time
- Updatability
  - Don’t force a rebuild to swap out dependencies
- Use less space
  - Dedup within the image
  - The image itself should take up less space

Image Provenance

```
oci
├── blobs
│   └── sha256
│       ├── 020351670a035e...
│       ├── 17cd3b4b4bddb1...
│       ├── f205cc2418c7cf...
│       └── fab909f4217187...
├── index.json
└── oci-layout
```

Auditability

```
oci
├── blobs
│   └── sha256
│       ├── 020351670a035e...
│       ├── 17cd3b4b4bddb1...
│       ├── f205cc2418c7cf...
│       └── fab909f4217187...
├── index.json
└── oci-layout
```

Auditability

- IMA
    - Checksums/signatures on individual files stored in an xattr
    - Checksums/signatures are verified at open()
- Why not IMA?
    - Then you have to use IMA
    - Not necessary with previous design: content addressability gives us this for free

Auditability

```
oci
├── blobs
│   └── sha256
│       ├── 020351670a035e...
│       ├── 17cd3b4b4bddb1...
│       ├── f205cc2418c7cf...
│       └── fab909f4217187...
├── index.json
└── oci-layout
```

Use squashfs instead!

What is squashfs?

- Mountable readonly filesystem
- “Squashfs is intended for general read-only filesystem use, for archival use (i.e. in cases where a .tar.gz file may be used)...”
    - Documentation/filesystems/squashfs.txt
- Metadata stored separately
    - Seekable
- Parallel compression

How do we implement this?

- Use squashfs instead of tar for blobs
- Mount each layer blob as -t squashfs
- Mount the rootfs with each layer as a lower_dir for overlay

Overlay issues

- Mount options limited to 4096 characters
  - = ~55 layers with reasonable path names
- Non-customizable whiteout format
  - .wh.foo vs mknod foo c 0 0
- Doesn’t support exactly one layer
  - Many base images have this format

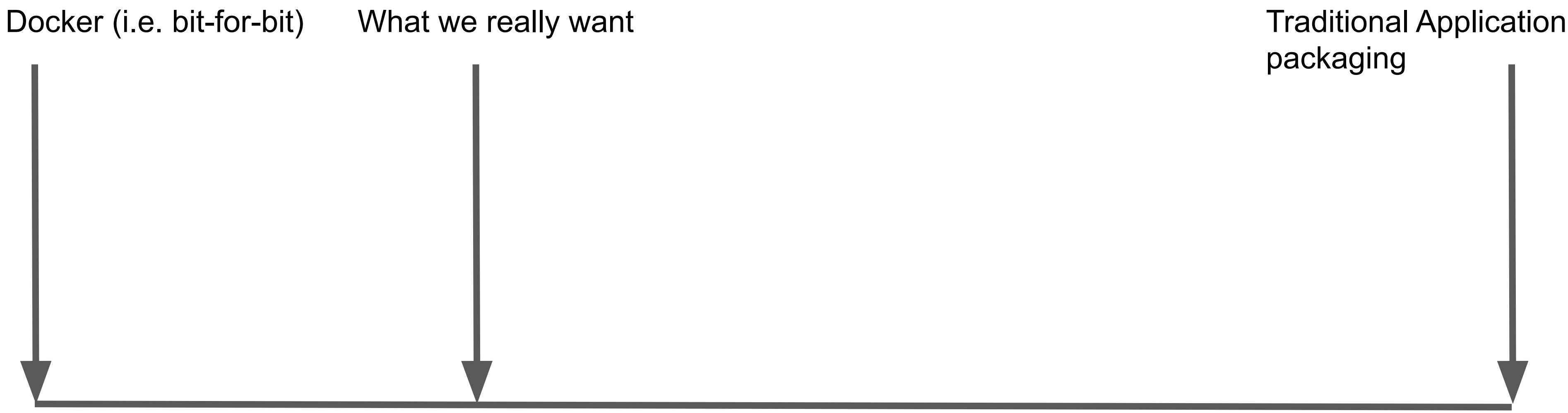
Squashfs issues

- Not really active
  - last commit a3f94cb99a85 ("Squashfs: Compute expected length from inode size rather than block length") from Aug 2018
- No userspace libraries for generating blobs
  - Current implementation has a fairly brutal hack w/ mksquashfs
- Doesn’t support some FS primitives containers use
  - ACLs
  - Others?

But we’re doing it anyway!

- http://github.com/anuvu/stacker
    - Generate “OCI”
        - Squashfs blobs
        - Overlay whiteout style vs. strict OCI style
- http://github.com/anuvu/atomfs
    - Ingest “OCI” images

updating.



Docker (i.e. bit-for-bit)

What we really want

Traditional Application packaging

A strategy for container updating

```
A:
  from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install openssl
    - yum install python3
    - git clone https://example.com/A
    - ./A/install
```

```
B:
  from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install openssl
    - yum install python3
    - git clone https://example.com/B
    - ./B/install
```

A strategy for container updating

```
ssl:
 from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install openssl
python3:
  from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install python3
```

```
A:
  from:
    type: docker
    url: docker://centos:latest
  apply:
    - docker://ssl:latest
    - docker://python3:latest
  run:
    - git clone https://example.com/A
    - ./A/install
```

A strategy for container updating

```
ssl:
  from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install openssl
```

```
64fabd853e4de75a7e... -> ssl:latest
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

A strategy for container updating

```
python3:
  from:
    type: docker
    url: docker://centos:latest
  run:
    - yum install python3
```

```
8ab6c5e1cb34a35a35... -> python:latest
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

End result

```
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

End result

```
64fabd853e4de75a7e... -> ssl:latest, included verbatim
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

End result

```
8ab6c5e1cb34a35a35... -> python:latest, included verbatim
64fabd853e4de75a7e... -> ssl:latest, included verbatim
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

End result

```
c34553482dda4a28dd... -> diff from app install
8ab6c5e1cb34a35a35... -> python:latest, included verbatim
64fabd853e4de75a7e... -> ssl:latest, included verbatim
e05fab2a890d758805... -> centos:latest
39ad9e63562e5d7087...
```

End result

```
c34553482dda4a28dd...   A:
8ab6c5e1cb34a35a35...     from:
64fabd853e4de75a7e...       type: docker
e05fab2a890d758805...       url: docker://centos:latest
39ad9e63562e5d7087...     apply:
                            - docker://ssl:latest
                            - docker://python3:latest
                          run:
                            - git clone https://example.com/A
                            - ./A/install
```

Updating

```
c34553482dda4a28dd...   A:
4aa9fc2a435abe95a1...     from:
64fabd853e4de75a7e...       type: docker
e05fab2a890d758805...       url: docker://centos:latest
39ad9e63562e5d7087...     apply:
                            - docker://ssl:latest
                            - docker://python3:latest+1
                          run:
                            - git clone https://example.com/A
                            - ./A/install
```

size.

EC5274
SS/SD
СДЕЛАНО В БОЛГАРИИ
boedem
FLOPPY DISK
5,25" HD,DS
HD
WRITE ENABLE
WRITE PROTECT
INDEX

Can we do better?

- https://github.com/openSUSE/umoci/issues/256
  - “[rfc] OCIv2 implementation”
- What would a new container image format look like?
  - No duplication across layers
  - Reasonable performance when mounted in-place

Thanks!

We are hiring! Linux, containers, go, packaging, etc.
tycho@tycho.ws, tycander@cisco.com
http://github.com/tych0