

DISTROLESS IMAGES

Securing Your Docker Images

Melissa McKay
Developer Advocate @JFrog





MELISSA MCKAY

Developer Advocate @JFrog



@melissajmckay



[linkedin.com/in/melissajmckay](https://www.linkedin.com/in/melissajmckay)

THE AGENDA

- Container History
- Containers in Real Life
- Container Gotchas
- Distroless Images



**ALL
ABOUT...**

CONTAINERS



SHARING LIMITED RESOURCES



1979 / 1982- chroot

PROGRESS TOWARD VIRTUALIZATION

- 2000 - FreeBSD jail
- 2004 - Solaris Zones / snapshots
- 2006 - Google Process Containers / cgroups
- 2008 - IBM LinuX Containers (LXC)
- 2013 - Docker (open source!)
 - Google LMCTFY (open source!)
- 2014 - Docker trades LXC for libcontainer
- ... *more stuff happened*
- June 2015 - Open Container Project/Initiative (OCI)
 - Runtime Specification (runtime-spec)
 - Image Specification (image-spec)
- ... *even more stuff happened and is **still happening!***

2011
Java 7

2014
Java 8



**WHAT
EXACTLY
IS A
CONTAINER?**



CONTAINER COMPONENTS

TARBALL OF A FILESYSTEM

LINUX FEATURES

- namespaces
- cgroups
- Union File systems

```
docker-desktop:~# lsns
```

	NS	TYPE	NPROCS	PID	USER	COMMAND
4026532297	mnt		13	13436	999	postgres
4026532298	uts		13	13436	999	postgres
4026532299	ipc		13	13436	999	postgres
4026532300	pid		13	13436	999	postgres
4026532302	net		13	13436	999	postgres

```
~ ▷ docker stats
```

CONTAINER	CPU %	MEM USAGE / LIMIT
d99745e33562	0.01%	480KiB / 1GiB
9094a1844f8e	4.16%	1.338GiB / 1.944GiB
fcf20c230c2c	0.19%	19.41MiB / 1.944GiB

Mix these together to create and run a container! Voila!

<https://docs.docker.com/get-started/overview/>

FILESYSTEM DETAILS

```
~ ▶ docker info
```

```
...
```

```
Operating System: Docker Desktop
```

```
OSType: linux
```

```
Architecture: x86_64
```

```
CPUs: 8
```

```
Total Memory: 1.944GiB
```

```
Name: docker-desktop
```

```
ID: 2POK:GJEZ:EHHW:WDRH:PYOW:PQ6C:LYAB:XLOH:DYSW:4SSN:A3JR:NXUF
```

```
Docker Root Dir: /var/lib/docker
```

```
Debug Mode: true
```

```
File Descriptors: 67
```

```
Goroutines: 76
```

```
...
```



NOTE: On OSX, containers will actually be running in a tiny Linux VM (use screen)

```
screen ~/Library/Containers/com.docker.docker/Data/vms/0/tty
```

FILESYSTEM DETAILS

~ ▷ docker images

REPOSITORY

mjmckay-app-docker.jfrog.io/my-image

docker.bintray.io/jfrog/artifactory-jcr

TAG

latest

7.5.7

IMAGE ID

62165ddeceb6

8b3066e25260

~ ▷ docker inspect 8b3066e25260

[

{

"Id": "sha256:8b3066e252609e484b032c583dada4ebd6f59b6b5de0a2f597f91b5ed4bcf117",

...

"GraphDriver": {

"Data": {

"LowerDir": "/var/lib/docker/overlay2/b01599ceea2761004b4f6a0a0d3d5c368dc40c8f20808434de4ed312029b1ff/diff:/var/lib/docker/overlay2/47dbb7eff56c58762e84b943a98bd1b558b5800b000b98bc6a07bfae53c1d79e/diff:/var/lib/docker/overlay2/2f5763dd07792eb22869fd9118a80d2170eafe6936d78bc73dbc3dc600e

...

"MergedDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e719cffc4383722af9406/merged",

"UpperDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e719cffc4383722af9406/diff",

"WorkDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e719cffc4383722af9406/work"

FILESYSTEM DETAILS

~ ▶ docker ps -a

CONTAINER ID	IMAGE	COMMAND
d99745e33562	mjmckay-app-docker.jfrog.io/my-image:latest	"/bin/sh -c 'tail -f..."
9094a1844f8e	docker.bintray.io/jfrog/artifactory-jcr:7.5.7	"/entrypoint-artifac..."
fcf20c230c2c	docker.bintray.io/postgres:9.6.11	"docker-entrypoint.s..."

docker-desktop:~# ls /var/lib/docker/

builder	containers	overlay2	swarm	volumes
buildkit	image	plugins	tmp	
containerd	network	runtimes	trust	

docker-desktop:~# ls /var/lib/docker/containers/

9094a1844f8e398845a6ae8f44c1cd9b8ffa21101133a6042ec741faf1ff9b0d
d99745e335621a1ed138fa1812d7fc83d9c5e337a159f92efd70ed7ed46df4b0
fcf20c230c2cc706a82bc16a6b9e39ee8a8d82b6508bd03cfd80d1ea2715106c

~ ▶ docker rm my_image_name

~ ▶ docker system prune

~ ▶ docker run -d --memory=1g mjmckay-app-docker.jfrog.io/my-image:latest --rm



CONTAINER
GOTCHAS

CONTAINER GOTCHAS

- RUNNING AS ROOT



CONTAINER GOTCHAS

- NO CONSTRAINTS



CONTAINER GOTCHAS

- NEVER UPDATING



CONTAINER GOTCHAS

- JAVA/JVM GOTCHAS



CONTAINER GOTCHAS

- IMAGE BLOAT



DISTROLESS IMAGES

- AND MULTISTAGE BUILDS

- Waste Not Want Not (smaller images)
- No Shell
- No Exec

```
1  FROM openjdk:11-jdk-slim AS build-env
2  ADD . /app/examples
3  WORKDIR /app
4  RUN javac examples/*.java
5  RUN jar cfe main.jar examples.HelloJava examples/*.class
6
7  FROM gcr.io/distroless/java:11
8  COPY --from=build-env /app /app
9  WORKDIR /app
10 CMD ["main.jar"]
```

<https://github.com/GoogleContainerTools/distroless> (examples)

MANAGING YOUR IMAGES - REMOTE BY DEFAULT

<https://dzone.com/refcardz/getting-started-with-container-registries>

START FREE: <https://bit.ly/MelissaWKSHP>

