

UNPACKING JAVA CONTAINERS

A Crash Course
in

Virtualized Container Technology

Melissa McKay

@melissajmckay

<http://jfrog.com/shownotes>



THE SWAG

WIN
A JFROG SHIRT!



<https://bit.ly/GIDSJavaMj>



MELISSA MCKAY
Developer Advocate @JFrog

 @melissajmckay

THE AGENDA

- Brief History
- The Container Market
- What is Docker?
- What is a Container?
- A few Container Gotchas
- Managing Images with JFrog Artifactory



**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



THE CONTAINER MARKET (according to Sysdig)

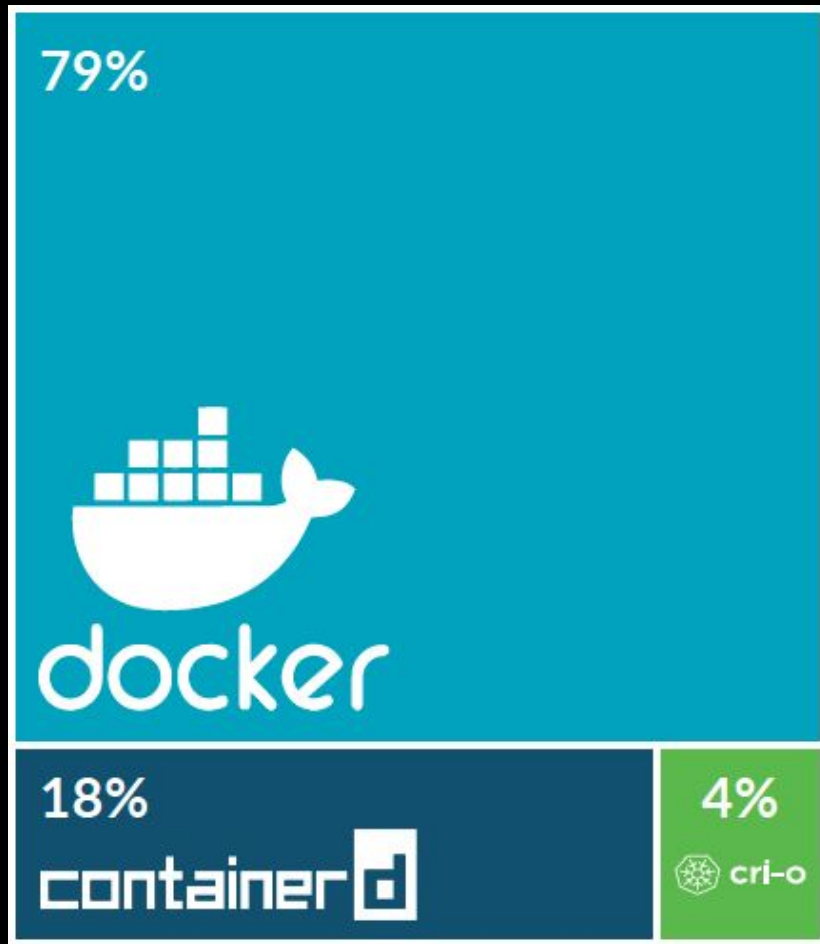
2017 - 45,000 Containers, 99% Docker

2018 - 90,000 Containers



Fig. 1. 2018 Container Runtimes from: "2018 Docker usage report," 29 May. 2018, sysdig.com/blog/2018-docker-usage-report/. Accessed 10 Jun. 2020.

THE CONTAINER MARKET



2019 - 2 million Containers
(includes both SaaS & on prem users)

Fig. 2. 2019 Container Runtimes from: "Sysdig 2019 Container Usage Report: New Kubernetes and security insights," 29 Oct. 2019, sysdig.com/blog/sysdig-2019-container-usage-report/. Accessed 10 Jun. 2020.



**WHAT
EXACTLY
IS
DOCKER?**

WHAT DO WE ACTUALLY NEED/WANT?

- An isolated environment where a user/application can operate, sharing the host system's OS/kernel without interfering with the operation of another isolated environment on the same system (a container)
- A way to define a container (an image format)
- A way to build an image of a container
- A way to manage container images
- A way to distribute/share container images
- A way to create a container environment
- A way to launch/run a container (a container runtime)
- A way to manage the lifecycle of container instances

DOCKER, THE WHOLE PACKAGE

DOCKER ENGINE

DOCKER IMAGE FORMAT

Dockerfile `docker build`

`docker images`

`docker rm`

`docker push`

`docker pull`

DOCKER HUB

`docker run`

`docker stop`

`docker ps`



BREAKING UP THE MONOLITH

OCI IMAGE FORMAT

- Docker V2 Image Spec

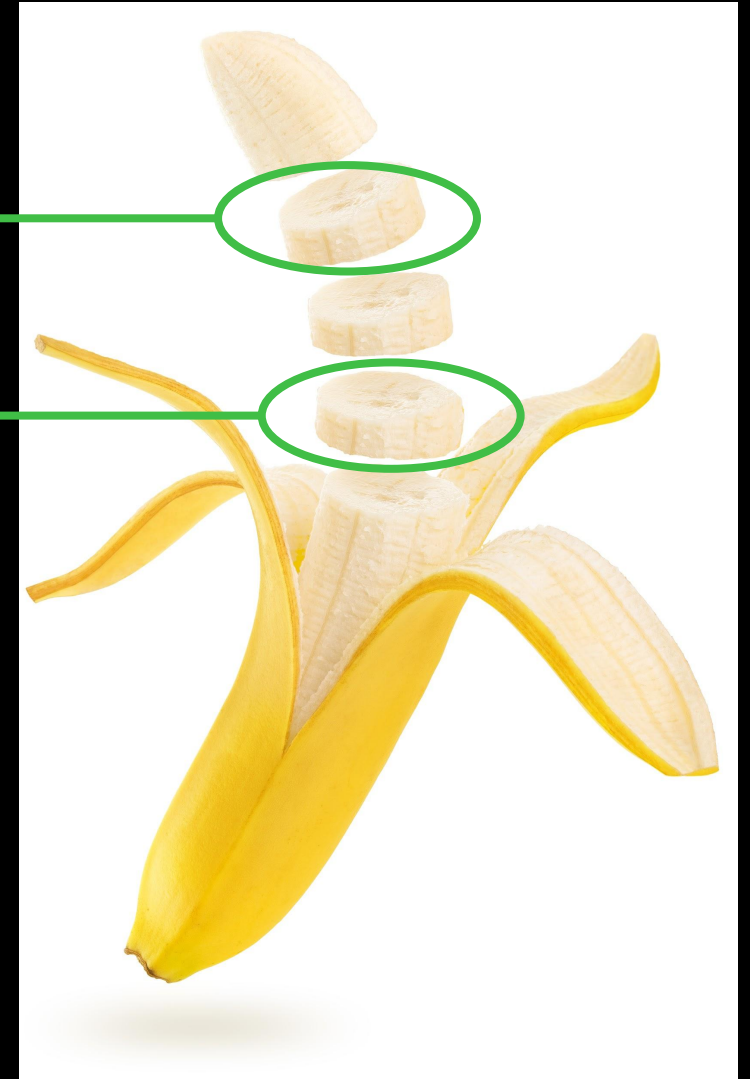
OCI CONTAINER RUNTIME

- runC (*which used to be libcontainer... which was written by Docker*)

OTHERS - containerd, rkt, cri-o, Kata, etc...

<https://lwn.net/Articles/741897/>

<https://www.ianlewis.org/en/container-runtimes-part-1-introduction-container-r>



**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:EHW: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	TAG	IMAGE ID
mjmckay-app-docker.jfrog.io/my-image	latest	62165ddeceb6
docker.bintray.io/jfrog/artifactory-jcr	7.5.7	8b3066e25260

```
~ ▷ docker inspect 8b3066e25260
```

```
[
  {
    "Id": "sha256:8b3066e252609e484b032c583dada4ebd6f59b6b5de0a2f597f91b5ed4bcf117",
    ...
    "GraphDriver": {
      "Data": {
        "LowerDir": "/var/lib/docker/overlay2/b01599ceea2761004b4f6a0a0d3d5c368dc40c8f208084
34de4ed312029b1ff/diff:/var/lib/docker/overlay2/47dbb7eff56c58762e84b943a98bd1b558b5800b000b98bc6a07b
fae53c1d79e/diff:/var/lib/docker/overlay2/2f5763dd07792eb22869fd9118a80d2170eafe6936d78bc73dbc3dc600e
...
        "MergedDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e71
9cffc4383722af9406/merged",
        "UpperDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e719
cffc4383722af9406/diff",
        "WorkDir": "/var/lib/docker/overlay2/a0bbe2014fe5a7befe1eaaca401a3d2ac54340e7513e719c
ffc4383722af9406/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 prune
```

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



**A FEW
CONTAINER
GOTCHAS**

CONTAINER GOTCHAS - RUNNING AS ROOT



CONTAINER GOTCHAS - NO CONSTRAINTS



CONTAINER GOTCHAS - NEVER UPDATING



CONTAINER GOTCHAS - JAVA/JVM GOTCHAS



MANAGING YOUR IMAGES - ARTIFACTORY

The screenshot displays the JFrog Platform interface. The top navigation bar includes the JFrog logo, the text 'JFrog Platform', and a breadcrumb trail 'Packages > docker'. A search bar and user profile 'Welcome, admin' are also visible. The left sidebar contains a menu with 'Application' (highlighted), 'Dashboard', 'Artifactory', 'Packages', 'Builds', 'Artifacts', 'Distribution', 'Pipelines', and 'Security & Compliance'. The main content area shows the details for a Docker image: 'docker-app / 249'. It includes a Docker logo, a 'High' Xray Severity rating with a red '1' notification, a size of '446.5 MB', and '1 Downloads'. Below this, there are tabs for 'Builds', 'Xray Data', 'Distribution', and 'Docker Layers' (which is active). The 'Docker Layers' section shows a 'CMD' layer with a '249' badge, an 'ADD' layer with a SHA256 hash and a file path, and a 'MAINTAINER' layer. The bottom left corner of the interface contains a trial license notice for JFrog.

JFrog Platform

Packages > docker

Orbitera: My Orbitera instance

Packages > docker-app > 249

docker-app / 249

03-12-19 15:09:50 -0700

High Xray Severity

446.5 MB Size

1 Downloads

Builds Xray Data Distribution Docker Layers

> CMD 249

ADD

sha256:a457d3d48831efb1fb9820b0bef47b08e76c602c55438626ae3ed86f179ae865

file:fd20a1d64236fc2939f42e1179676f0db94637f1108147418825e9bf6b24e502 in /home/exec/tomcat/webapps/swampup.war

2019-12-03 22:09:45 20.5 MB

MAINTAINER

Enterprise Plus trial license
7.2.1 rev 70201900 Licensed to JFrog TEST
© Copyright 2020 JFrog Ltd

MANAGING YOUR IMAGES - XRAY

The screenshot displays the JFrog Platform Xray interface. A modal window titled "ISSUE DETAILS" is open, showing the following information:

Component Id	docker-app:249
Package Type	<i>maven</i>
Type	Security
Provider	Jfrog
Summary	Apache struts security bulletin incorrect version upgrade issue
Description	Apache struts contains a flaw that is triggered

The "Impact" section features a target diagram with a red bullseye. A legend on the right lists the components in the dependency tree:

- docker-app:249
- sha256_a457d3d48831...
- swampup.war
- m org.apache.struts:strut...** (highlighted with a red line to the bullseye)

The background interface shows the "Packages" view for "docker-app" with version "249". The bottom status bar indicates the issue is for "Apache Struts Security ..." with a severity of "Me" (Medium) and a version range of "2.5 ≤ Ver...".



THE SWAG

WIN
A JFROG SHIRT!



<https://bit.ly/GIDSJavaMJ>



THANK YOU!

Melissa McKay

@melissajmckay

<http://jfrog.com/shownotes>