ORACLE

JVM in Containers - Best Practices

JFokus 2020

David Delabassée - @delabassee

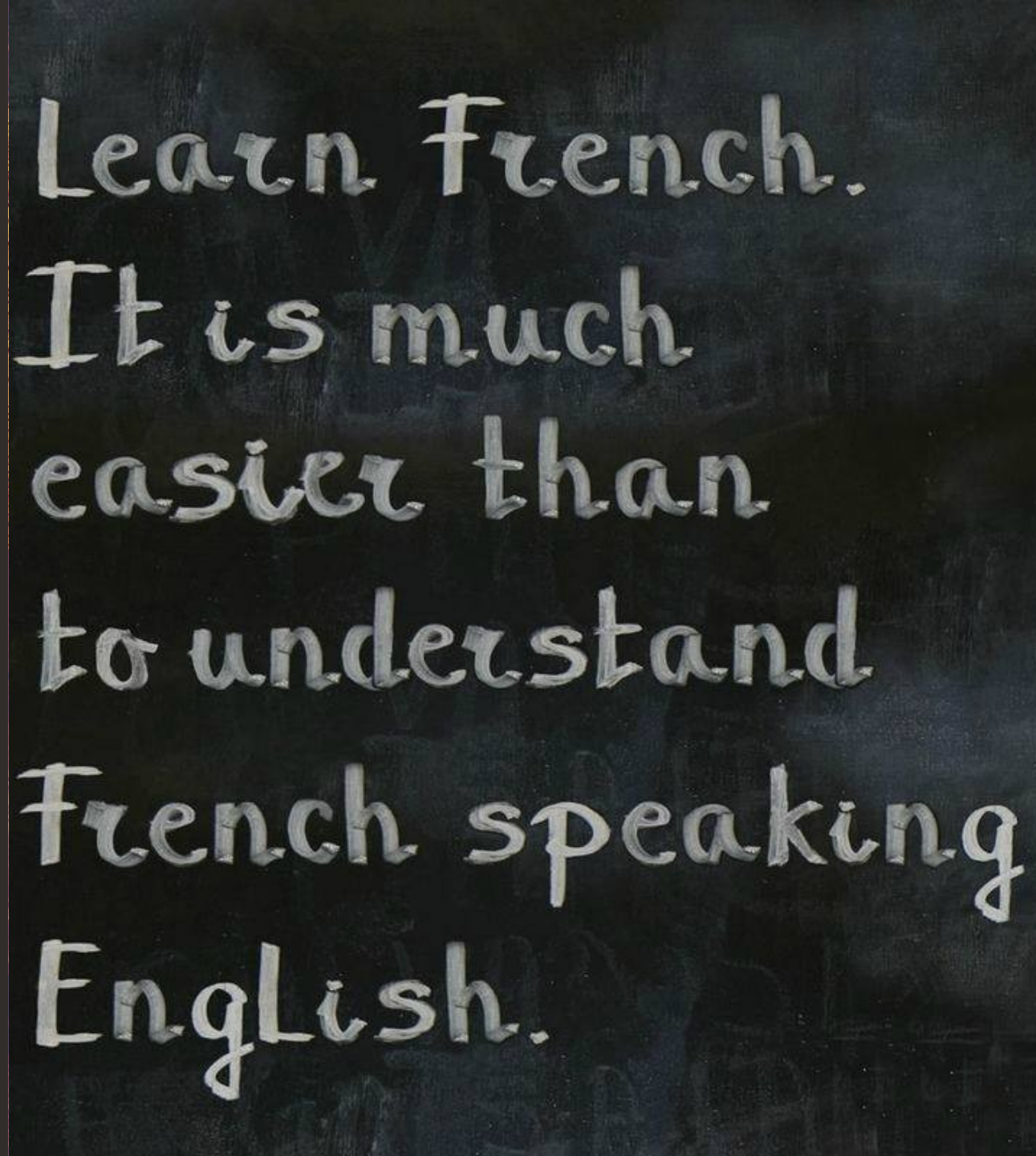
DevRel

Java Platform Group - Oracle

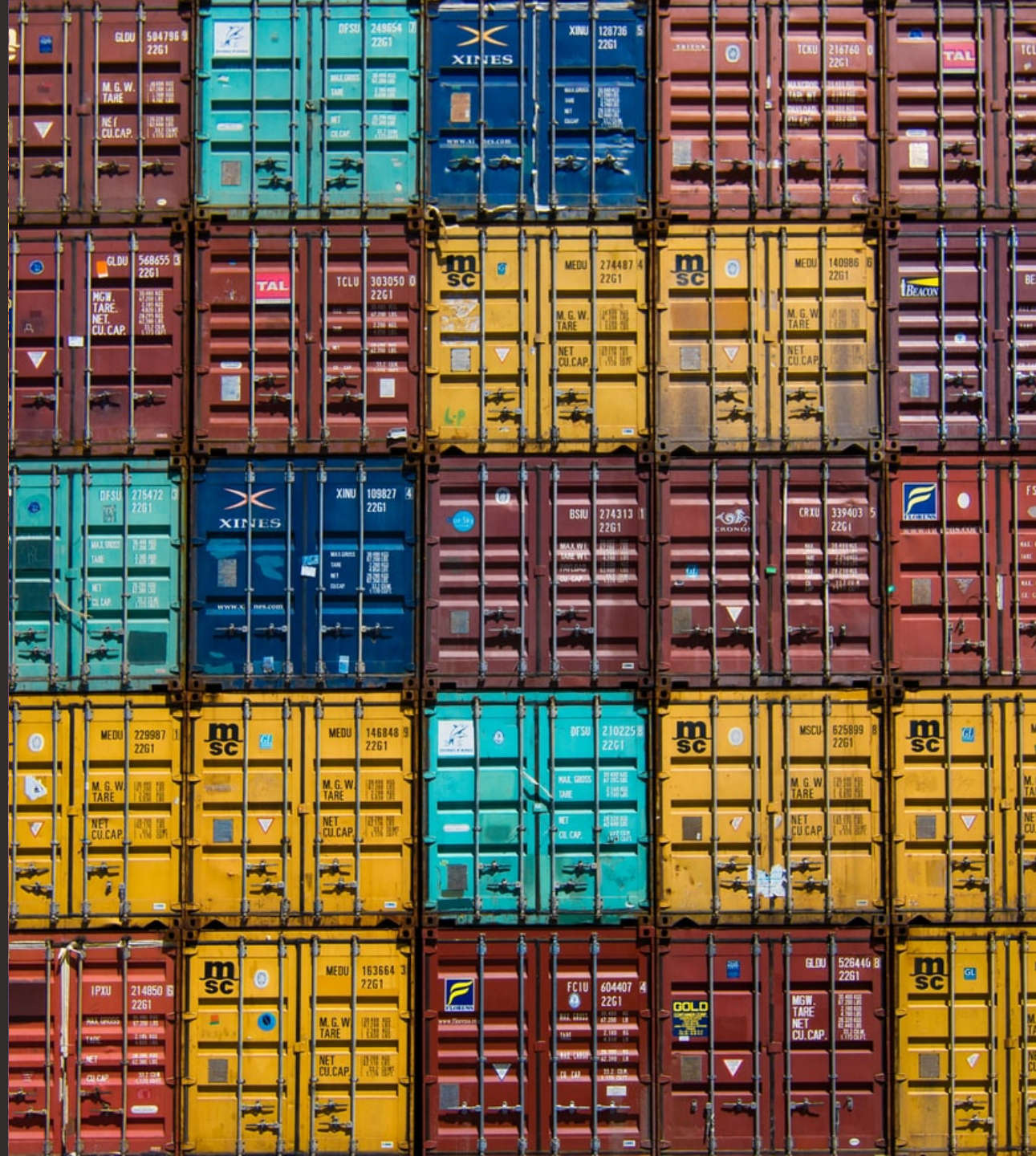
Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

David Delabassée
@delabassee
Java Platform Group



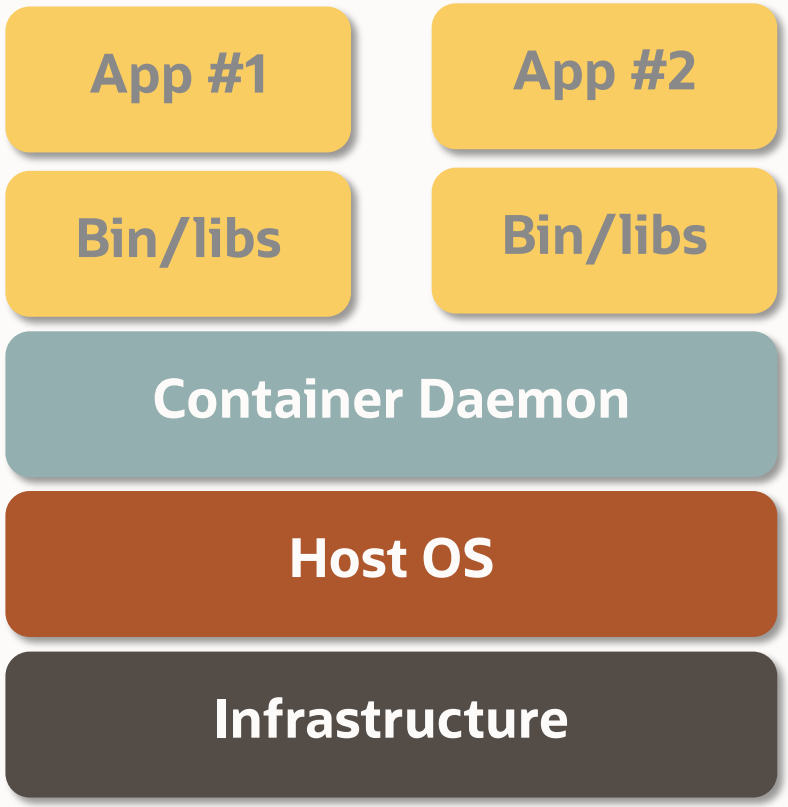
Containers

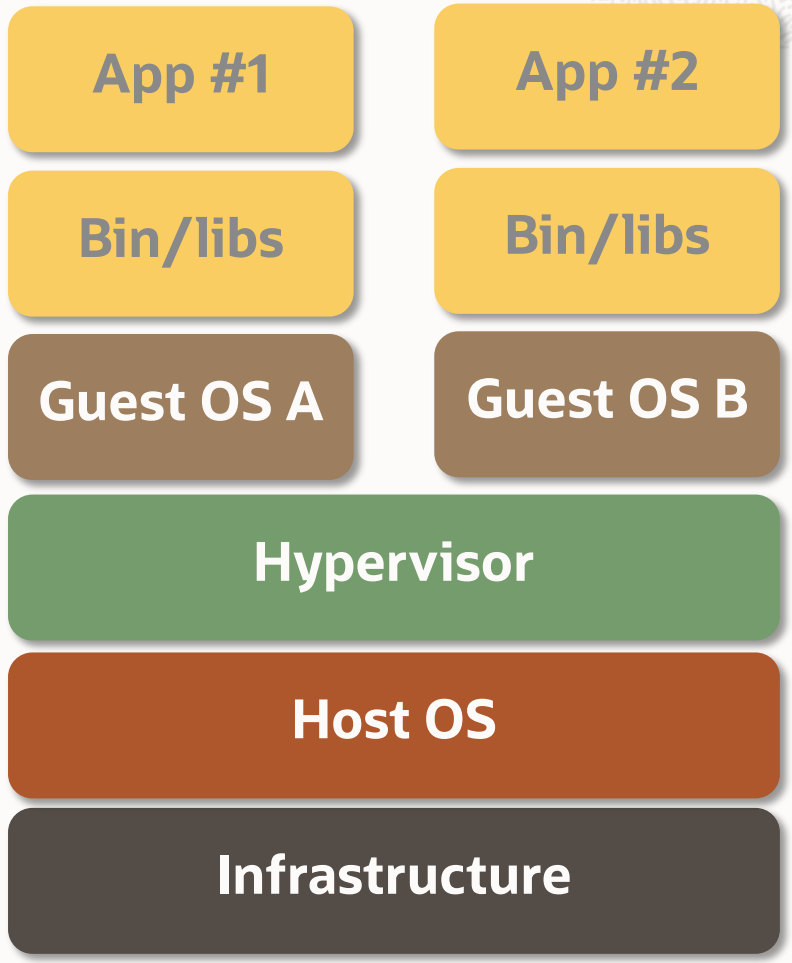


Container

- Package Software into Standardized Units
  - Development
  - Shipment
  - Deployment
- Runtimes
  - Docker, CRI-O, LXC, Rkt, runC, systemd-nspawn, OpenVZ, etc.

Container vs. VM





Java

JVM Container Landscape

Tools

- docker-maven-plugin
- jib + jib-maven-plugin
- Testcontainers
- IDE
- …

Frameworks

- Helidon
- Quarkus
- Micronaut
- JHipster
- …

FaaS

- Fn Project
- OpenFaaS
- OpenWhisk
- …

JVM in Containers - Towards a Symbiosis

- JVM should behave as a good (Container) citizen
- Improved “latency”

Demo

Hello Container

Latency

Latency

Container Startup

Stack of Layers

3 ‘core’ layers

- Java application and its dependencies
- Java Runtime
- Operating System

⇨ Reduce layers size

Java Application Layer

- Dependencies!
- Leverage Container cache layer mechanism
- Fat JAR?
- CDS Shared Archive
- Anything that is (relatively) static in its own layer

Java Runtime Layer

Serverless Java function (Fn) - openjdk:13

	Modules	jlink flags	MB		
JDK	Whole JDK!		**316**	**100%**	
Runtime image	All (explicit)	`--add-modules $(java --list-modules)`	**178**	56%	**100%**
Custom runtime image	Only required modules	`--add-modules $(jdeps --print-module-deps …)`	50	16%	28%
		`… --no-header-files --no-man-pages --strip-java-debug-attributes`	44	14%	25%
		`… --compress=1`	37	12%	21%
		`… --compress=2`	**34**	**11%**	**19%**

316 MB → 178 MB → 50 MB → 34 MB

Operating System Layer

- Slim distros
  - debian: bullseye (117 MB) vs. debian: bullseye-slim (71 MB)
- Distroless distros
  - gcr.io/distroless/java:11 (195 MB - Java included)
- Docker-slim
  - “Don't change anything in your Docker container image and minify it by up to 30x” (?)
- …

Operating System Layer

- Alpine - Security-oriented, lightweight Linux distro
- musl - Lightweight, fast, free, C standard library implementation

- alpine-pkg-glibc - glibc compatibility layer package for Alpine
  https://github.com/sgerrand/alpine-pkg-glibc
- Project Portola - Runs OpenJDK on musl (*)
  https://openjdk.java.net/projects/portola/

Demo

Java Runtime Layer

Minecraft server

java.base, java.compiler, java.desktop, java.management, java.naming, java.rmi, java.scripting, java.sql, jdk.sctp, jdk.unsupported, jdk.zipfs

openjdk:13 (*) (12 modules)	88 MB
`--strip-debug --strip-java-debug-attributes`	-14 MB
`--compress=1`	-18 MB
`--compress=2`	-31 MB
`--no-header-file --no-man-pages`	0 MB

(*) Oracle OpenJDK builds on OEL - YMMV!

Java Runtime Layer

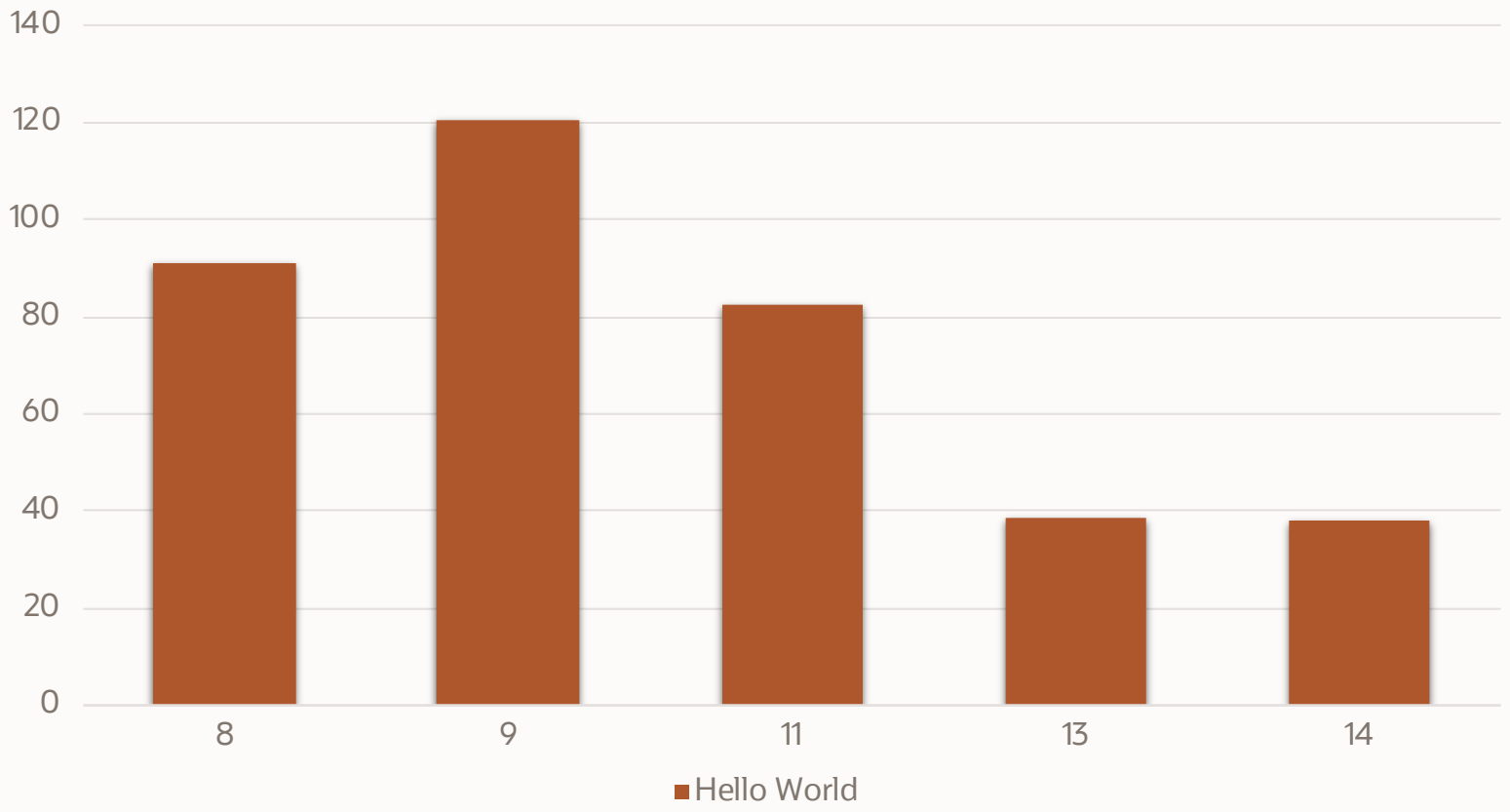
Base Image	Java	Module	Custom Runtime
openjdk:13	Inc. Oracle OpenJDK 13	java.base	48 MB
debian:buster	+ Debian openjdk-13-jdk	java.base	491 MB
`--strip-native-debug-symbols` (*)			
debian:buster	+ Debian openjdk-13-jdk	java.base	51 MB

(*) JDK 13 https://bugs.openjdk.java.net/browse/JDK-8219257

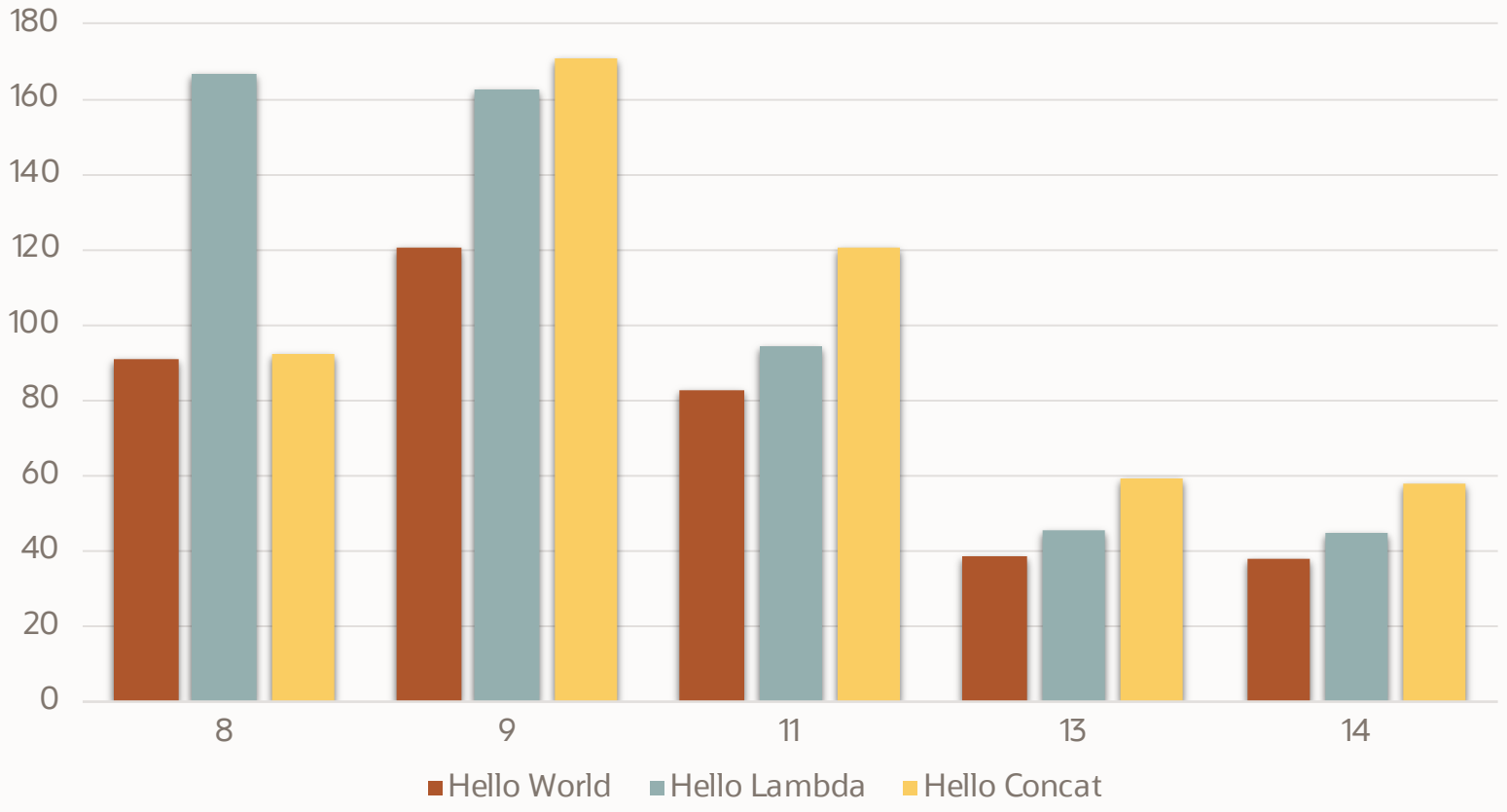
Latency

Application Startup

Java - Startup Time



Java - Startup Time



https://cl4es.github.io

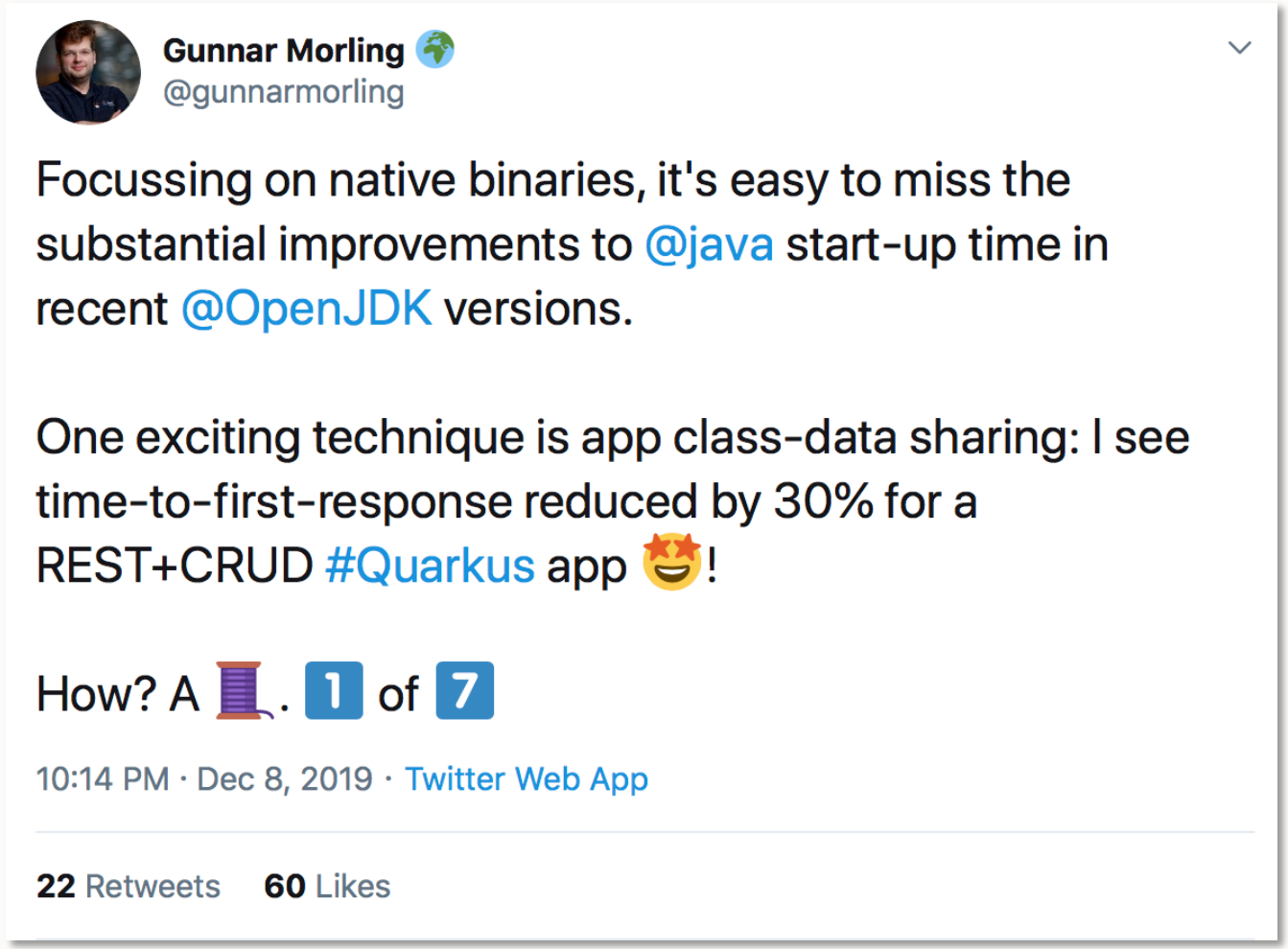
Class Data Sharing

- Reduce memory footprint between multiple JVMs by sharing common class metadata
- Improve startup time
- How?
  - Loads classes from JAR file into a private internal representation
  - Dumps it to a shared archive
  - When JVMs (re)starts, the archive is memory-mapped to allow sharing of R/O JVM metadata for these classes among multiple JVMs

Demo

CDS

Application CDS



Application CDS

Application CDS

jdk-08-u202-b08-hotspot

```
jruby -e 1
    real      0m1.601s
    …

jruby --dev -e 1
    real      0m1.216s
    …

jruby --disable-gems --dev -e 1
    real      0m0.853s
    …
```

jdk-13.jdk

```
… -J-XX:SharedArchiveFile=jruby.jsa
    real      0m1.491s
    …

… -J-XX:SharedArchiveFile=jruby.jsa
    real      0m1.089s
    …

… -J-XX:SharedArchiveFile=jruby.jsa
    real      0m0.717s
    …
```

Class Data Sharing

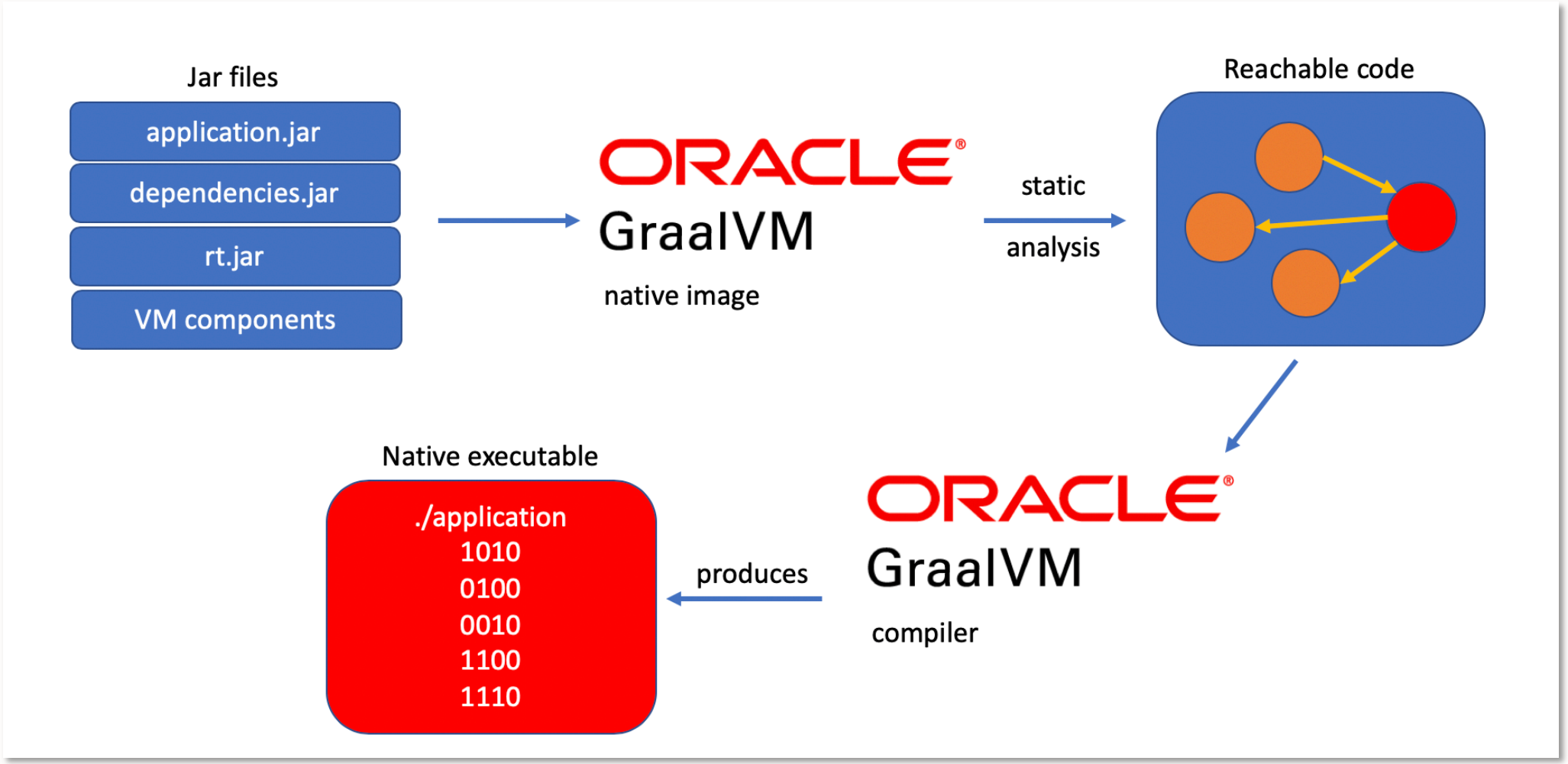
- Java 5 - Limited to system classes and serial GC
- Java 9 - Application CDS and other GCs (commercial feature + JEP 250)
- Java 10 - Application CDS (JEP 310)
- Java 12 - Default CDS Archives (JEP 341)
- Java 13 - Dynamic CDS Archives (JEP 350)

GraalVM

- High-performance Polyglot Virtual Machine
- …
- JIT Compiler
- AOT Compiler - native-image
  - Reduced startup time
  - Improved foot-print
  - Reduced image size

https://www.graalvm.org

GraalVM - native-image



GraalVM - native-image limitations

- Java 8 & 11
- Mostly supported
  - Reflections, Dynamic Proxy, JNI, Unsafe Memory Access, Static Initializers, References
- Not supported
  - InvokeDynamic and Method Handles, Dynamic Class Un/Loading, Finalizers, Security Manager, Serialisation
  - Native VM interfaces (JVMTI, JMX, etc.)

https://github.com/oracle/graal/blob/master/substratevm/LIMITATIONS.md

Container

JVM should behave correctly

JVM Container Awareness

JDK-8186248	More flexibility in selecting Heap % of available RAM (8u144)
JDK-8179498	attach should be relative to `/proc/pid/root` and namespace aware as `jcmd`, `jstack`, ... fail to attach (10)
JDK-8146115	Improve Docker container detection & resource config usage (10)
JDK-8193710	`jcmd -l` & `jps` do not list Java processes running in containers (11)
JDK-8203357	Container Metrics (11)
JDK-8203359	JFR jdk.CPUInformation event reports incorrect information when running in Docker container (in progress)
...	...
JDK-8230305	Cgroups v2: Container awareness (15)

https://bugs.openjdk.java.net

Ergonomics

- The JVM tunes itself based on the system it runs on
- Behavior-Based Tuning dynamically optimizes the sizes of the heap to meet an expected behavior
  - Maximum Pause-time (`-XX:MaxGCPauseMillis`)
  - Or Application Throughput (`-XX:GCTimeRatio`)
- Sets defaults for the GC, heap size, and runtime compiler

https://docs.oracle.com/en/java/javase/13/gctuning/ergonomics.html

Demo

Ergonomics

Wrap-up

JVMs in Containers

- Reduce “latency”
  - Container Startup
  - Application Startup
- JVM behaves as a good (Container) citizen

JVMs in Containers

- Use the latest Java version, never java:latest !!!
  - `-XX:+UseContainerSupport`
- Only rely on actively-supported versions!
- Use a JRE/Java runtime image instead of a JDK
- It's "containers as usual"
  - Move costs to the build phase
  - Small(er) is better!
    - Reduce the potential attack surface
    - Use as few layers as possible
    - Multi-stage build
  - Docker-bench-security, Snyk, Clair, Anchore, etc.
  - ...

Mystery meat OpenJD

Gil Tene gil at azul.com
Wed May 15 18:49:55 UTC 2019

- Previous message: RFR(S) Backport: 821
- Next message: Mystery meat OpenJDK b
- **Messages sorted by:** [date] [thread] [s

Umm...

```
Lumpy.local-43% docker run -it --rm ope
openjdk version "1.8.0_212"
OpenJDK Runtime Environment (build 1.8.
OpenJDK 64-Bit Server VM (build 25.212-
Lumpy.local-44% date
Wed May 15 11:41:12 PDT 2019

Look at the build
than March 27, 201
on April 16, 2019.

Similarly:

Lumpy.local-46% dc
openjdk version "1
OpenJDK Runtime Environment (build 11.0
OpenJDK 64-Bit Server VM (build 11.0.3+
Lumpy.local-47% date
Wed May 15 11:43:12 PDT 2019

This one was populate dno later than Ap
the actual 11.0.3 was released on April

If anyone was wondering about the impor
"EA" (or some other "THIS IS NOT a RELE
and all OpenJDK builds that are not an
```

This one was
the actual 1

If anyone wa
"EA" (or som

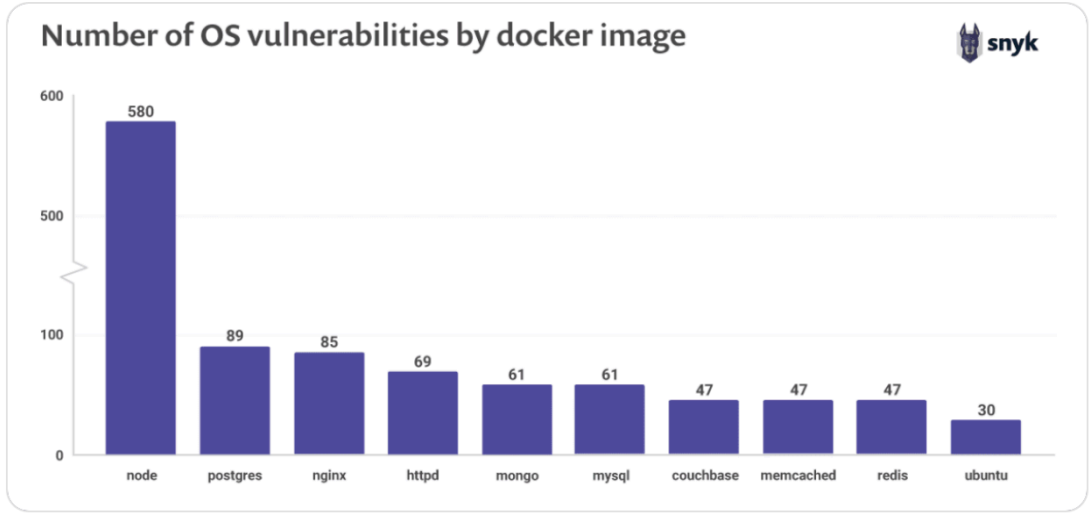
...on strings say
...) on any

Simon Maple
@sjmaple

Follow

The top 10 most popular @Docker containers each contain at least 30 vulnerabilities. The official @nodejs image ships with 580 system library vulnerabilities.

snyk.io/blog/top-ten-m …



11:08 pm - 26 Feb 2019

107 Retweets **90** Likes

4 107 90

“so you can consider it as the (OpenJDK 11 Alpine) General-Availability Release”

```
RUN echo "Downloading jdk build"
RUN wget http://drive.jku.at/ssf/s/readFile/share/8207/4867522971216226929/publicLink/openjdk-11-GA_linux-x64-musl_b

RUN echo "Downloading sha256 checksum"
RUN wget http://drive.jku.at/ssf/s/readFile/share/8208/-1932052387783488162/publicLink/openjdk-11-GA_linux-x64-musl_

ENV JDK_ARCHIVE="openjdk-11-GA_linux-x64-musl_bin.tar.gz"
RUN echo "Verify checksum"
RUN sha256sum -c ${JDK_ARCHIVE}.sha256
```

Choose your base image wisely !
And secure it!

Tack!

David Delabassée
@delabassee

ORACLE