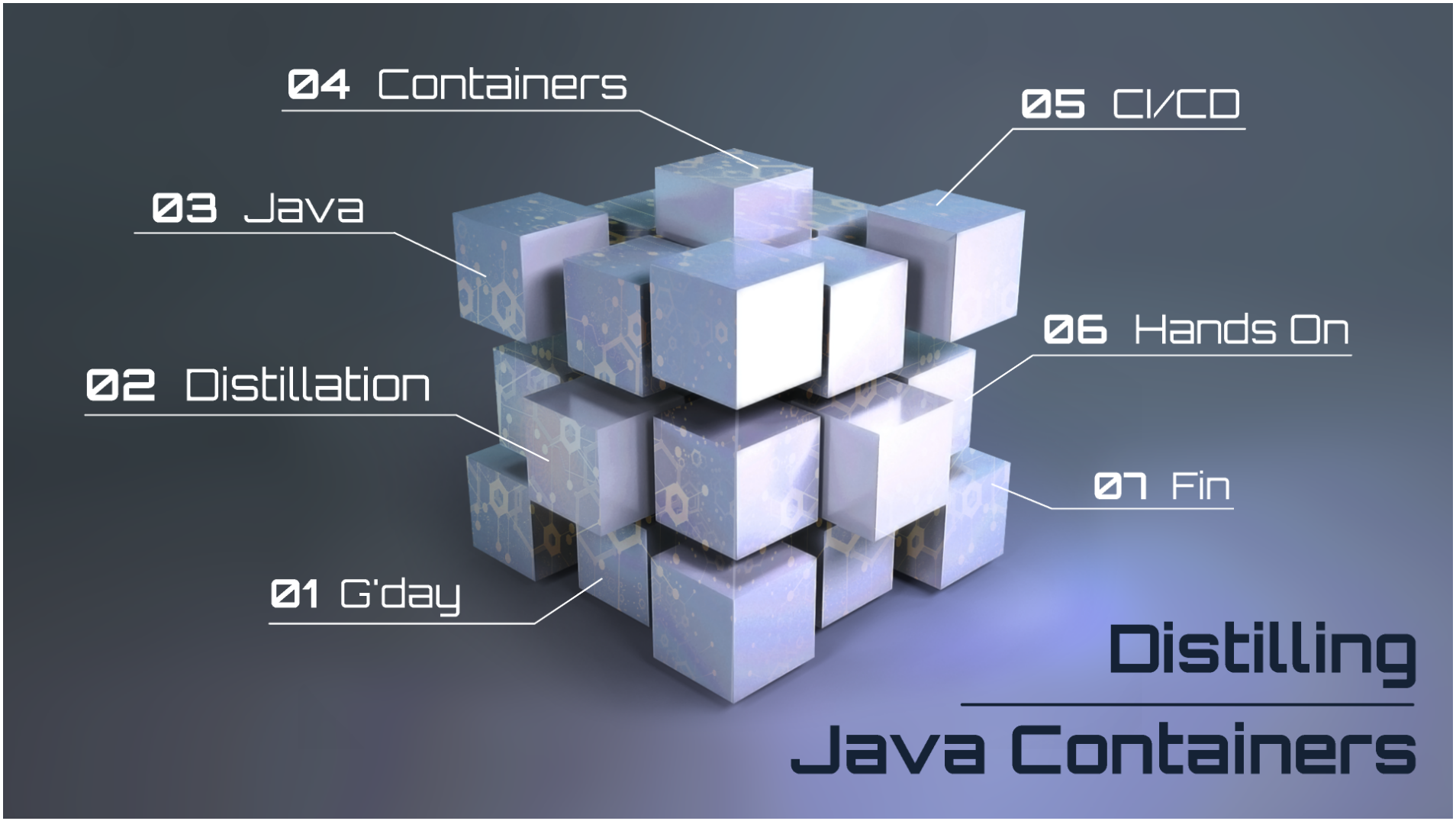

04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Distilling Java Containers

2020

Jonathan Johnson

Independent
Cloud Native
Workshops & Training
Software Architectures
Kubernetes & Java Specialist

ThermoFisher SCIENTIFIC

We are Curios...

Have you packaged
any app in a container?

We are Curios...

Have you packaged any app in a container?

What languages are you permitted to use?

We are Curios...

Have you packaged any app in a container?

What languages are you permitted to use?

What languages pique your interests?

“Nobody puts Java in a container” 2017/2018

Troubles with isolation of **memory** and **CPU**

Jörg Schad

- Head Of Engineering and Machine Learning at ArangoDB
- Mesosphere

“Nobody puts Java in a container” 2017/2018

Troubles with isolation of **memory** and **CPU**

Tl;dr: The Java Virtual Machine (not even with the Java 9 release) is not fully aware of the isolation mechanisms containers use internally, this can lead to unexpected behavior...

https://jaxenter.com/nobody-puts-java-container-139373.html

Jörg Schad

- Head Of Engineering and Machine Learning at ArangoDB
- Mesosphere

“Nobody puts Java in a container” 2017/2018

Troubles with isolation of **memory** and **CPU**

Tl;dr: The Java Virtual Machine (not even with the Java 9 release) is not fully aware of the isolation mechanisms containers use internally, this can lead to unexpected behavior...

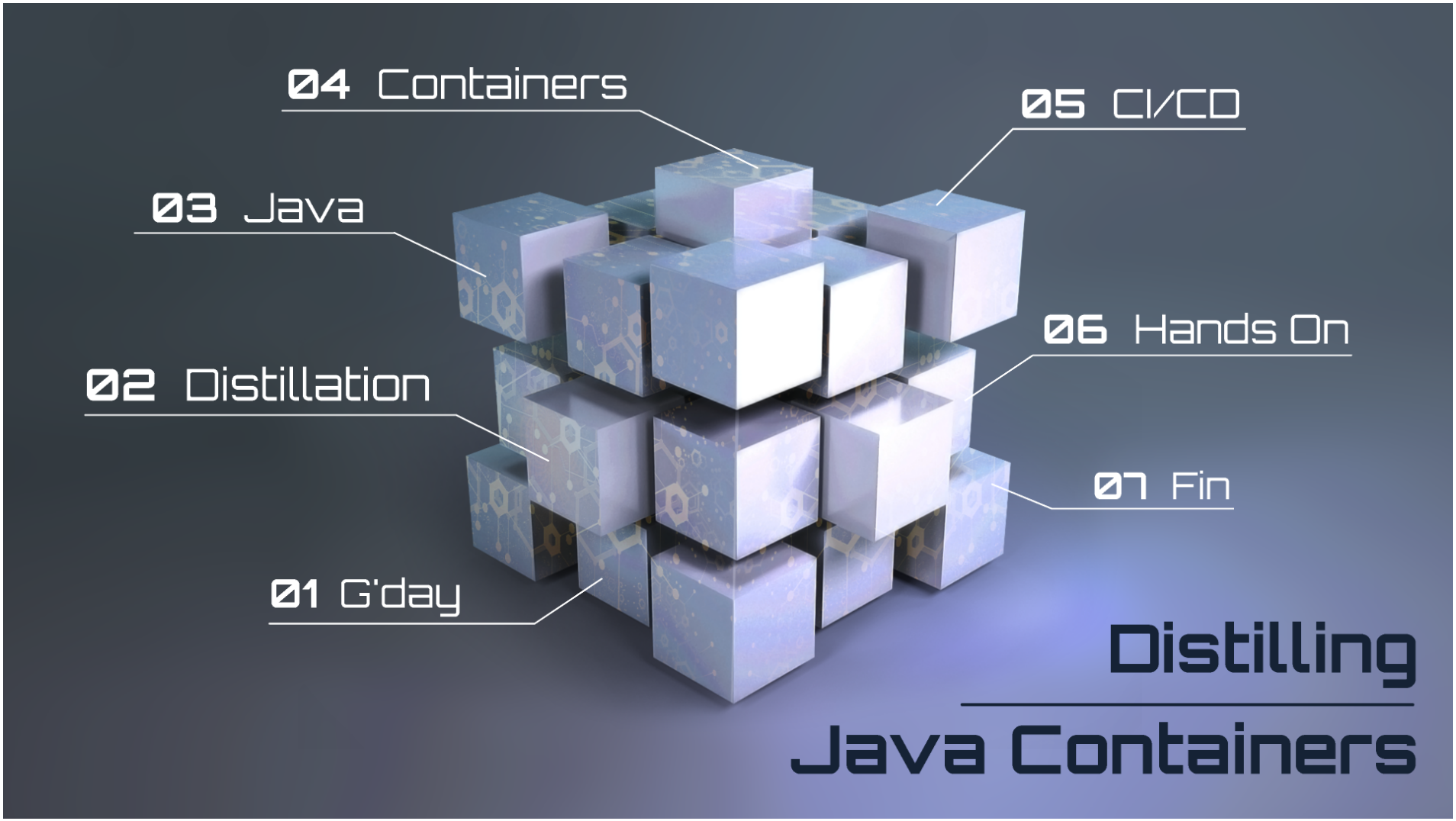
https://jaxenter.com/nobody-puts-java-container-139373.html

Jörg Schad

- Head Of Engineering and Machine Learning at ArangoDB
- Mesosphere

Ken Sipe

- Cloud Systems Architect at D2IQ (Mesosphere)
- No Fluff Just Stuff Speaker


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Distillation Pattern

Distilling to get just the right taste, in a container

Distillation Pattern

Distilling to get just the right taste, in a container

PeatOps 😉

Distillation Pattern

Java Apps can be distilled farther than you might expect

Distilling Java App and its Container

While these upcoming techniques can be applied to other languages, Java remains a significant language for our business features. It's important Java evolves with container ecosystems.

Distillation Pattern

8 Factors

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix

Low coupling

Exposed interfaces are minimal, simple, and stable.

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix

Low coupling

Exposed interfaces are minimal, simple, and stable.

Immutable

Container not capable of or susceptible to change.

Idempotent

Multiple instances all behave the same, over time.

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix

Immutable

Container not capable of or susceptible to change.

Attack Phobic

Security attack vectors are resisted as potential surfaces have been minimized.

Low coupling

Exposed interfaces are minimal, simple, and stable.

Idempotent

Multiple instances all behave the same, over time.

Small images

Less is more. Reduce I/O pressure. Images are revised, stored, transported, downloaded and scaled.

Distillation Pattern

8 Factors

High cohesion

All features in the container are well defined, and used.
"Do one thing and do it well." -- Unix

Low coupling

Exposed interfaces are minimal, simple, and stable.

Immutable

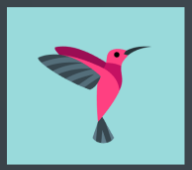
Container not capable of or susceptible to change.

Idempotent

Multiple instances all behave the same, over time.

Attack Phobic

Security attack vectors are resisted as potential surfaces have been minimized.



Small images

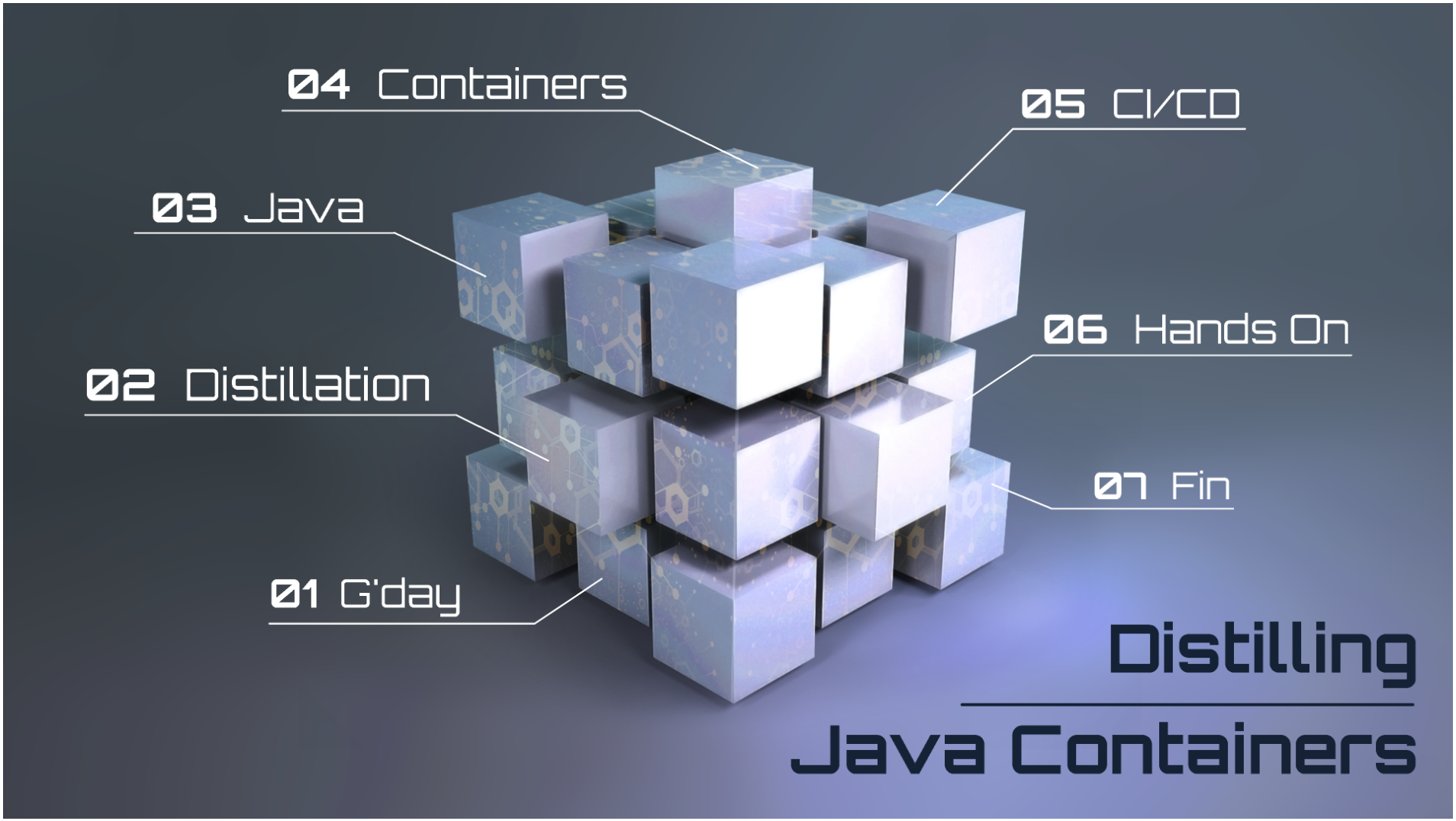
Less is more. Reduce I/O pressure. Images are revised, stored, transported, downloaded and scaled.

Rapid Startup

Containers are embraced as ephemeral, fragile, expendable, and scalable cattle.

Resource frugality

Every bit and CPU tick is a limited resource as containers are scaled throughout your datacenter.


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Java to Containers

Piqued Topics Related to Containers and Cloud Native

Java to Containers

Piqued Topics Related to Containers and Cloud Native

Cloud-native

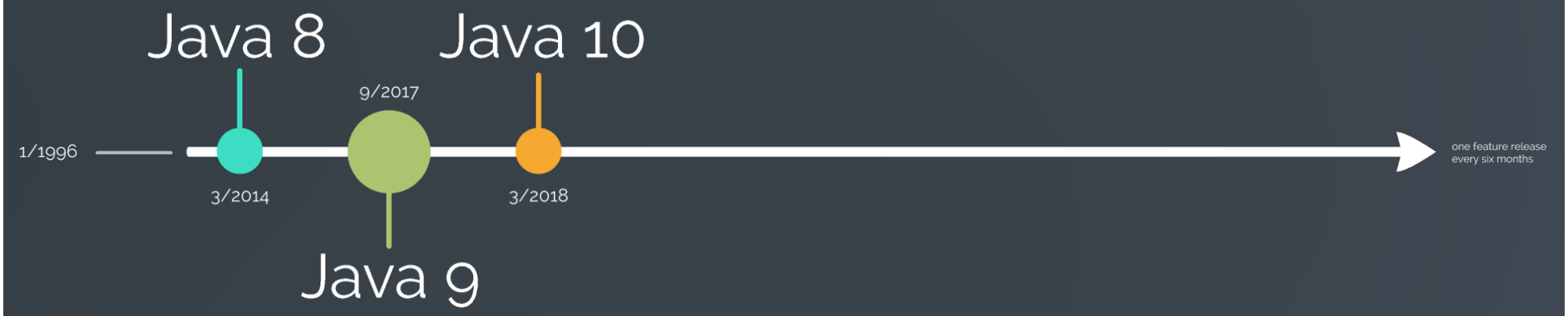
- Approach to building and running applications that exploits advantages of cloud computing delivery model.
- How applications are created and deployed, not where.

- Pivotal
https://pivotal.io/cloud-native

Milestones toward containers

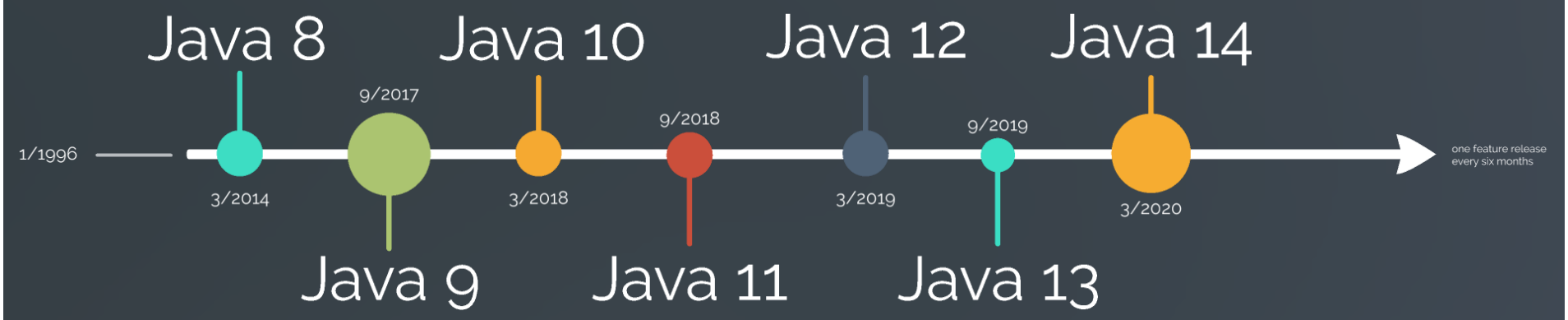
Milestones toward containers

Milestones toward containers

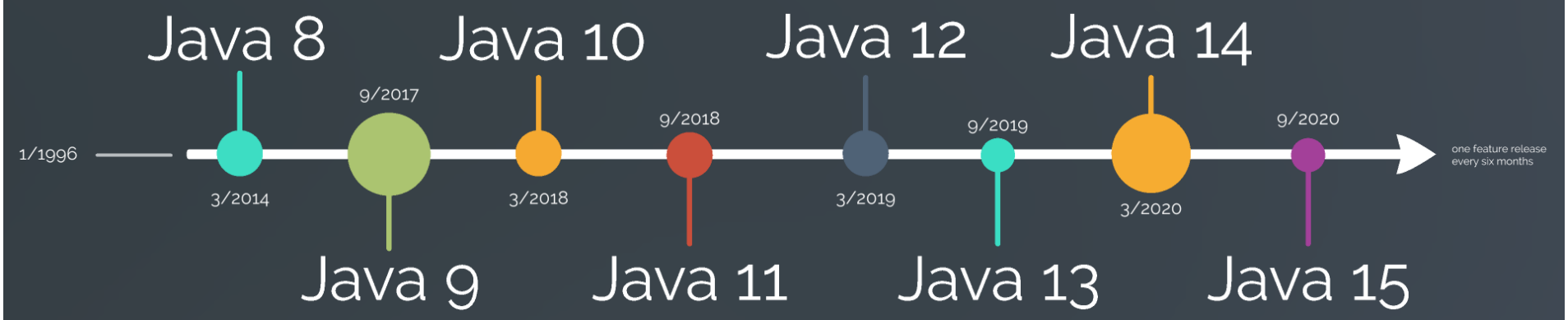


Milestones toward containers

Milestones toward containers

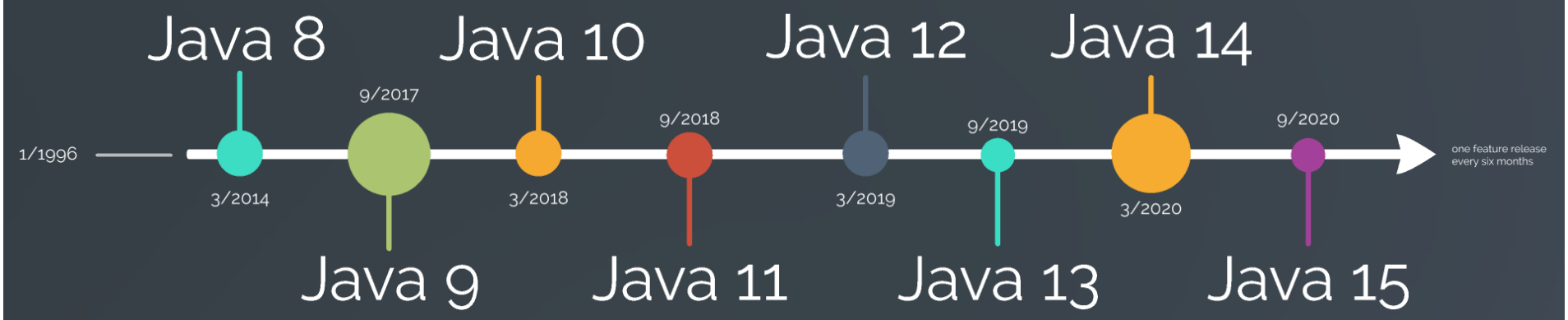


Milestones toward containers



Milestones toward containers

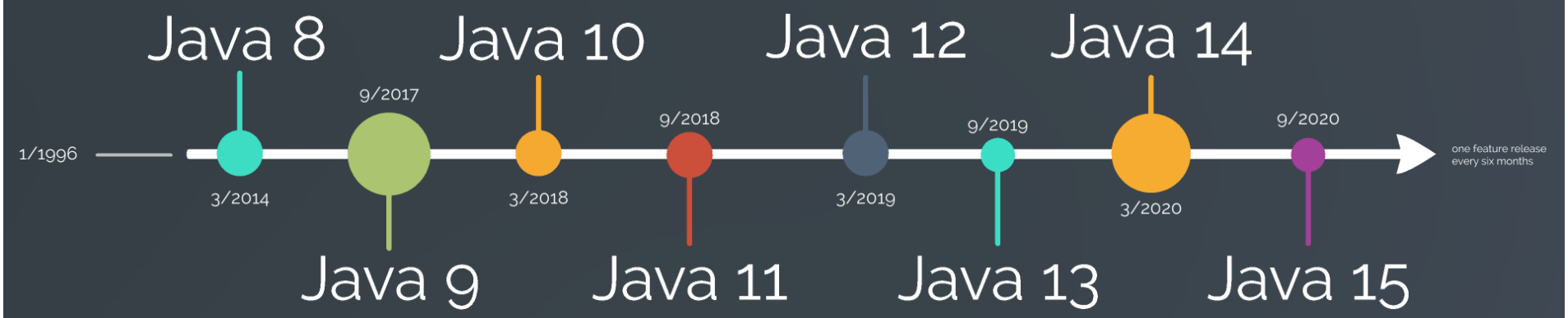
- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**



Milestones toward containers

- **JEP 310** Application class-data sharing
- **JEP 317** Experimental Java-based JIT compiler (Graal)
- **JDK-8146115** Improve docker container detection and resource configuration usage
- **JDK-8186248, JDK-8186309** Allow more flexibility in selecting Heap % of available RAM
- **JDK-8179498** attach in linux should be relative to /proc/pid/root and namespace aware

- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**



https://medium.com/@jnsrikanth/docker-support-in-java-10-fbff28a31827

https://blogs.oracle.com/java-platform-group/java-se-support-for-docker-cpu-and-memory-limits

Milestones toward containers

- **JEP 310** Application class-data sharing
- **JEP 317** Experimental Java-based JIT compiler (Graal)
- **JDK-8146115** Improve docker container detection and resource configuration usage
- **JDK-8186248, JDK-8186309** Allow more flexibility in selecting Heap % of available RAM
- **JDK-8179498** attach in linux should be relative to /proc/pid/root and namespace aware

- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**

1/1996

Java 8 — 3/2014

Java 9 — 9/2017

- Module System
- Linker, jlink
- Ahead-of-time-compilation (AOT)
- GraalVM

Java 10 — 3/2018

Java 11 — 9/2018

Java 12 — 3/2019

Java 13 — 9/2019

Java 14 — 3/2020

Java 15 — 9/2020

one feature release every six months

https://medium.com/@jnsrikanth/docker-support-in-java-10-fbff28a31827

https://blogs.oracle.com/java-platform-group/java-se-support-for-docker-cpu-and-memory-limits

Milestones toward containers

- **JEP 310** Application class-data sharing
- **JEP 317** Experimental Java-based JIT compiler (Graal)
- **JDK-8146115** Improve docker container detection and resource configuration usage
- **JDK-8186248, JDK-8186309** Allow more flexibility in selecting Heap % of available RAM
- **JDK-8179498** attach in linux should be relative to /proc/pid/root and namespace aware

- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**

1/1996

Java 8 — 3/2014

Java 9 — 9/2017

- Module System
- Linker, jlink
- Ahead-of-time-compilation (AOT)
- GraalVM

Java 10 — 3/2018

Java 11 — 9/2018

- OpenJDK / GNU
- **JEP 320** Remove the Java EE and CORBA Modules
- **JEP 332** Transport Layer Security (TLS) 1.3

Java 12 — 3/2019

Java 13 — 9/2019

Java 14 — 3/2020

Java 15 — 9/2020

one feature release every six months

https://medium.com/@jnsrikanth/docker-support-in-java-10-fbff28a31827

https://blogs.oracle.com/java-platform-group/java-se-support-for-docker-cpu-and-memory-limits

Milestones toward containers

- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**

- **JEP 310** Application class-data sharing
- **JEP 317** Experimental Java-based JIT compiler (Graal)
- **JDK-8146115** Improve docker container detection and resource configuration usage
- **JDK-8186248, JDK-8186309** Allow more flexibility in selecting Heap % of available RAM
- **JDK-8179498** attach in linux should be relative to /proc/pid/root and namespace aware

- **JEP 189** Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)
- **JEP 341** Default Class Data Sharing (CDS) Archives
- **JEP 346** Promptly Return Unused Committed Memory from G1

1/1996

Java 8
3/2014

Java 9
9/2017

- Module System
- Linker, jlink
- Ahead-of-time-compilation (AOT)
- GraalVM

Java 10
3/2018

Java 11
9/2018

- OpenJDK / GNU
- **JEP 320** Remove the Java EE and CORBA Modules
- **JEP 332** Transport Layer Security (TLS) 1.3

Java 12
3/2019

Java 13
9/2019

- **JEP 351** ZGC: Uncommit Unused Memory
- **JEP 353** Reimplement the Legacy Socket API

Java 14
3/2020

Java 15
9/2020

one feature release every six months

https://medium.com/@jnsrikanth/docker-support-in-java-10-fbff28a31827

https://blogs.oracle.com/java-platform-group/java-se-support-for-docker-cpu-and-memory-limits

Milestones toward containers

- Large Enterprise Adaption
- Container CPU and heap fix backported to Java 8u131 and again to **8u191**

- **JEP 310** Application class-data sharing
- **JEP 317** Experimental Java-based JIT compiler (Graal)
- **JDK-8146115** Improve docker container detection and resource configuration usage
- **JDK-8186248, JDK-8186309** Allow more flexibility in selecting Heap % of available RAM
- **JDK-8179498** attach in linux should be relative to /proc/pid/root and namespace aware

- **JEP 189** Shenandoah: A Low-Pause-Time Garbage Collector (Experimental)
- **JEP 341** Default Class Data Sharing (CDS) Archives
- **JEP 346** Promptly Return Unused Committed Memory from G1

- Garbage collectors refinements
- Non Volatile Memory

1/1996

Java 8 — 3/2014

Java 9 — 9/2017

- Module System
- Linker, jlink
- Ahead-of-time-compilation (AOT)
- GraalVM

Java 10 — 3/2018

Java 11 — 9/2018

- OpenJDK / GNU
- **JEP 320** Remove the Java EE and CORBA Modules
- **JEP 332** Transport Layer Security (TLS) 1.3

Java 12 — 3/2019

Java 13 — 9/2019

- **JEP 351** ZGC: Uncommit Unused Memory
- **JEP 353** Reimplement the Legacy Socket API

Java 14 — 3/2020

Java 15 — 9/2020

one feature release every six months

https://medium.com/@jnsrikanth/docker-support-in-java-10-fbff28a31827

https://blogs.oracle.com/java-platform-group/java-se-support-for-docker-cpu-and-memory-limits

Jlink

Wait, it's not linking my app?

Jlink inspects app dependencies and generates new, custom JRE with bare minimum.

Your classes, jar, or war are loaded as normal.

It's still Java JRE, just all unused modules omitted.

This new JRE loaded into your container.

For many microservices and high cohesion containers, an ideal distillation technique.

```
jlink \
--module-path $JAVA_HOME/jmods:mlib \
--add-modules com.greetings \
--output greetingsapp
```

http://www.progdoc.de/papers/intSwing/I18nNotepad.jar

https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdeps.html

Jlink

Wait, it's not linking my app?

Jlink inspects app dependencies and generates new, custom JRE with bare minimum.

Your classes, jar, or war are loaded as normal.

It's still Java JRE, just all unused modules omitted.

This new JRE loaded into your container.

For many microservices and high cohesion containers, an ideal distillation technique.

```
jlink \
--module-path $JAVA_HOME/jmods:mlib \
--add-modules com.greetings \
--output greetingsapp
```

- New, customer JRE generated in greetingsapp dir

- jdeps utility added to Java 8
- jlink utility added to Java 9

- Part of modular Java scheme

http://www.progdoc.de/papers/intSwing/I18nNotepad.jar

https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jdeps.html

Java Microframeworks

Minimalistic web application frameworks

Features often provided:

- Tooling
- Libraries
- Observability *
- Debugging *
- Connectivity *
- APIs: REST, gRPC, QueryML

* Although a mesh can also help

App Class-Data Sharing

Redundant work and information is cached and shared

What if you gave each runner full moment *before* the starting gun?

App Class-Data Sharing

Redundant work and information is cached and shared

What if you gave each runner full moment *before* the starting gun?

Given a class name, JVM must:

- look class up on disk,
- load class,
- verify bytecode,
- pull class into internal data structure.

Multiply by hundreds or thousands of classes.

App Class-Data Sharing

Redundant work and information is cached and shared

What if you gave each runner full moment *before* the starting gun?

Given a class name, JVM must:

- look class up on disk,
- load class,
- verify bytecode,
- pull class into internal data structure.

Multiply by hundreds or thousands of classes.

AppCDS creates a class-data archive once and then shares it, so JVM need not recreate it.

Decreases startup time 20-50%

```
$ time java -Xshare:off -jar app.jar
> Hello, application class-data sharing!
>
> real    0m0.078s
> user    0m0.094s
> sys     0m0.012s

$ time java -jar app.jar
> Hello, application class-data sharing!
>
> real    0m0.043s
> user    0m0.053s
> sys     0m0.014s
```

Java 10, JEP 310: Application Class-Data Sharing

https://blog.codefx.org/java/application-class-data-sharing/#Application-Class-Data-Sharing-In-A-Nutshell

Ahead of Time Compilation (AOT)

Improved start up and runtime performance.

Where opportunities exist, code compiled to native code.

Compiled code is called by runtime engine.

Similar to Just in Time (JIT) yet JIT does little to help startup time.

Graal has a AOT compiler.

Java 9 experimental feature, JEP-295

GraalVM™

- Developed by Oracle
- Replacement for Hotspot
- Non JVM languages: Node.js, Ruby, R, Python
- Implementations of:
    - Class Data Sharing (CDS)
    - Ahead of Time Compilation (AOT)

GraalVM™

- Developed by Oracle
- Replacement for Hotspot
- Non JVM languages: Node.js, Ruby, R, Python
- Implementations of:
  - Class Data Sharing (CDS)
  - Ahead of Time Compilation (AOT)

Still write once run anywhere, because it's in a container

Linker creates native image binary!

GraalVM™

Daniel Hinojosa
GRAAL The Magnificent!

- Developed by Oracle
- Replacement for Hotspot
- Non JVM languages: Node.js, Ruby, R, Python
- Implementations of:
  - Class Data Sharing (CDS)
  - Ahead of Time Compilation (AOT)

Still write once run anywhere, because it's in a container

Linker creates native image binary!

GraalVM™ Microframeworks

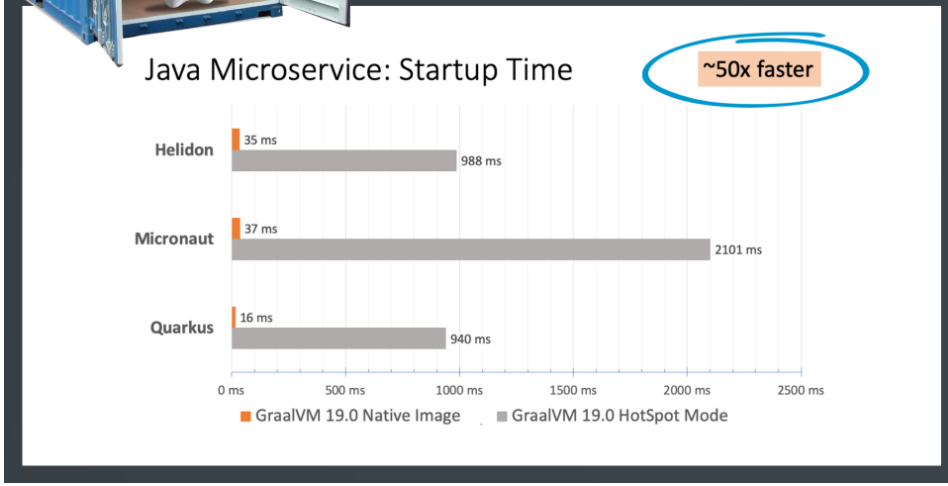
A subculture of frameworks around GraalVM wateringhole.

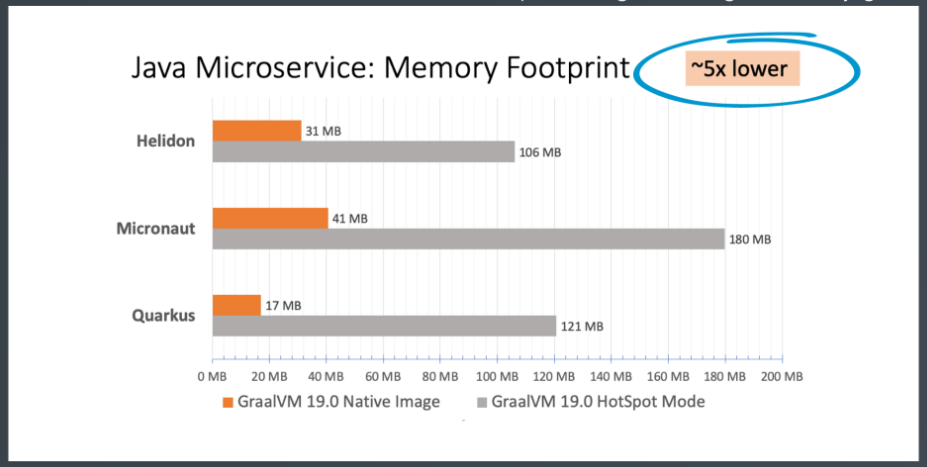
GraalVM™

Distillation nailed !

Native images significantly reduce runtime memory requirements compared to HotSpot.

http://www.graalvm.org/docs/why-graal





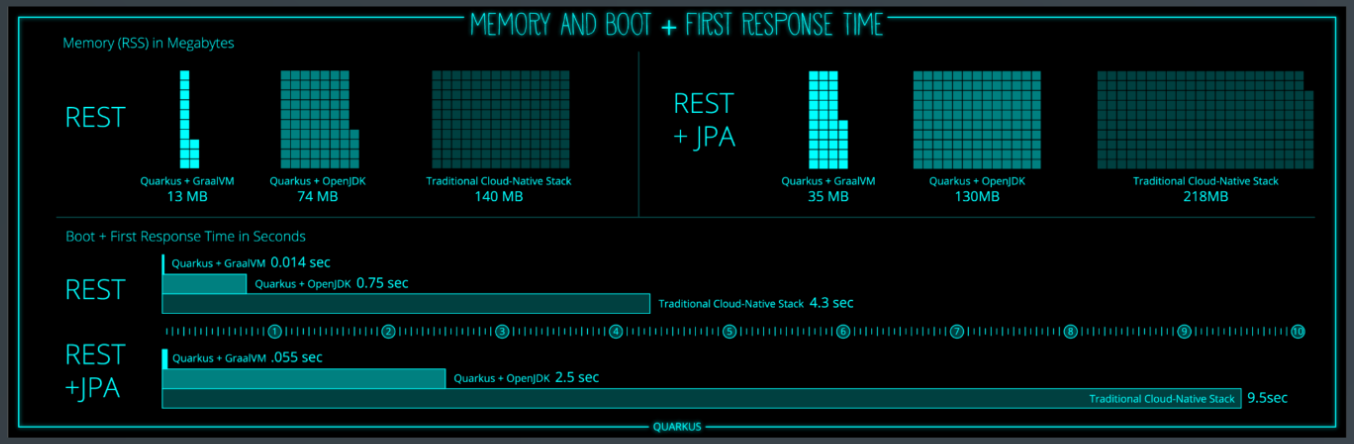
Quarkus

Container-first approach for cloud-native Java applications

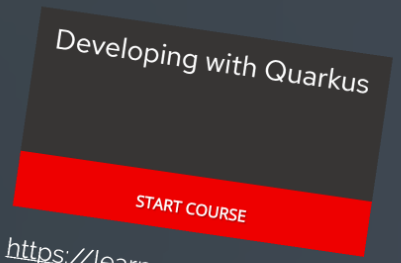
- Make Java leading platform in Kubernetes and serverless environments.
- Programming model to address wider range of distributed application architectures.

Red Hat

Introduced by Redhat in March 2019



https://quarkus.io/



https://learn.openshift.com/middleware/

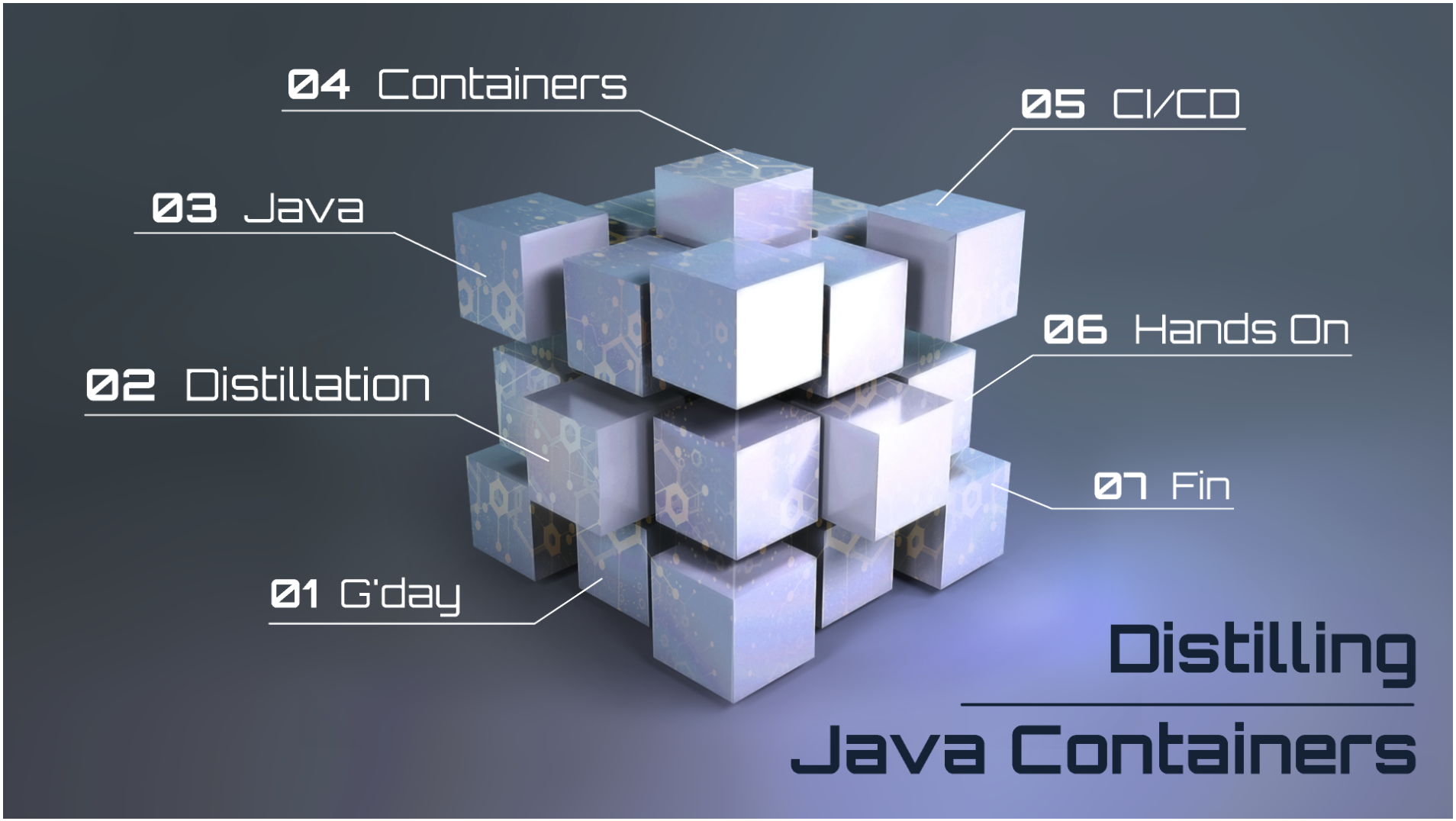
"Every main method deserves picocli!"

- Aims to be easiest way to create rich command line applications.
- Run **on** and **off** the JVM.

```
$ java Greet
Missing required option '-u=<userName>'

Hello world demo
Usage: Greet -u=<userName>
  -u, --user=<userName>       The user name.
```

Parameter parsing
Help generation
Type conversions
Subcommands
Error handling
Fast startup with Graal native compile


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Java to Container

How hard could it be?

Dockerfile

Linux

openjdk-8_linux-x64_bin.tar.gz

HelloWorld.class

Java to Container

How hard could it be?

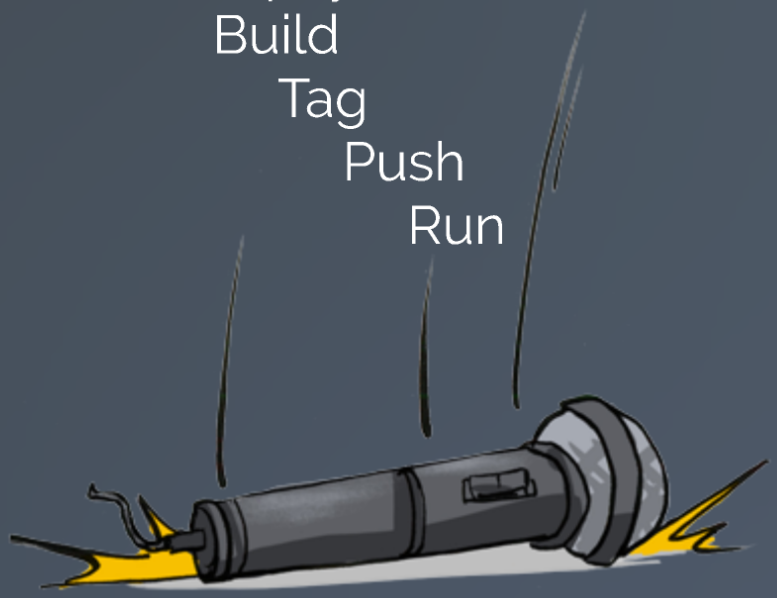
Dockerfile

Linux

openjdk-8_linux-x64_bin.tar.gz

HelloWorld.class

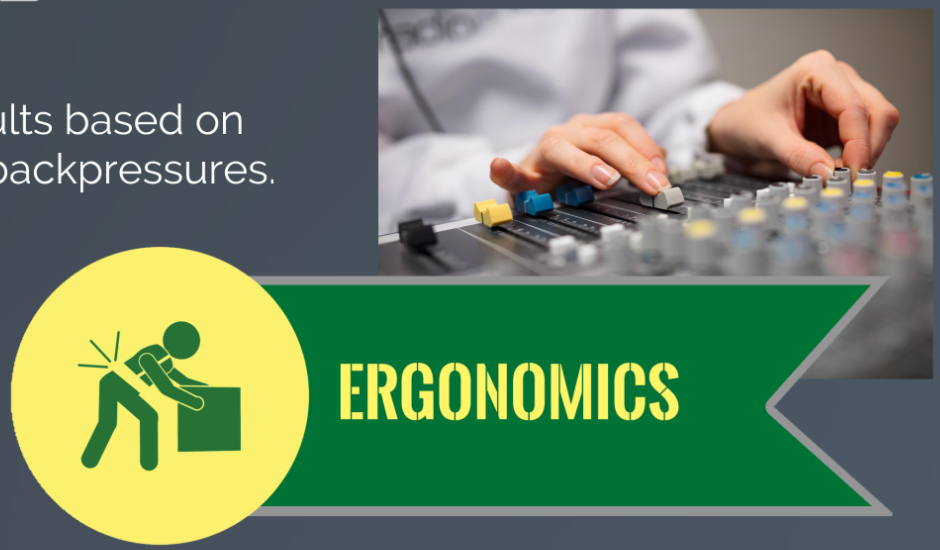
Declare
Add payload
Build
Tag
Push
Run



JVM Ergonomics

Self tuning of JVM switches to optimal defaults based on context of underlying target resources and backpressures.

- CPU and Heap usage defaults
- Garbage collector type and defaults
- Added in Java 5



- Container Aware Java
- Goldilocks settings based on container resources
- Past misbehavior with early Java 8: CPU and Heap were not limited to container
- Java switches added, helps for containers: -XX:{ Initial | Min | Max } RAMPercentage

Base Containers

Some of many choices

Open Projects

- Alpine
- Portola
- Debian versions based on Toystory characters
  Stretch, Jessie, Buster, Bullseye, Bookwork, Sid
- Scratch
- Distroless: minimal Linux, glibc-based system

DockerSlim

Notable Freemiums, Frenemies

Minimal Ubuntu, 29MB

RED HAT OPENSHIFT Universal Base Image (UBI)

AZUL SYSTEMS zulu-openjdk

Bake your own base

Don't you have enough to do? Security, performance, patches.... oh my

https://hub.docker.com/_/ubuntu/
https://ubuntu.com/blog/minimal-ubuntu-released

C standard library, libc

libc:

- Library of types, functions, macros and operating systems services with resource management.
- CPU, Memory, I/O
- Language ports link to it.

C standard library, libc

libc:

- Library of types, functions, macros and operating systems services with resource management.
- CPU, Memory, I/O
- Language ports link to it.

~ dozen implementation,
but two you will see often:

GNU C Library (glibc)

- Used in Debian and other common distros
- Considered more complete, yet perhaps "bloated"

musl libc

- Several Linux distributions including Alpine Linux
- Partially binary compatible with glibc
- Lightweight, optimized for embedded

C standard library, libc

libc:

- Library of types, functions, macros and operating systems services with resource management.
- CPU, Memory, I/O
- Language ports link to it.

Langs, like **Java,** port and validate on different OSs with libc variants.

Your app
JRE
C++
libc
OS

~ dozen implementation, but two you will see often:

GNU C Library (glibc)

- Used in Debian and other common distros
- Considered more complete, yet perhaps "bloated"

musl libc

- Several Linux distributions including Alpine Linux
- Partially binary compatible with glibc
- Lightweight, optimized for embedded

Small Linux for Distillation Pattern

5MB, without Kernel
Security-oriented
Lightweight
musl libc and busybox
Popular because of size

Small Linux for Distillation Pattern

5MB, without Kernel
Security-oriented
Lightweight
musl libc and busybox
Popular because of size

OpenJDK not supported Alpine after 8u212-jre-alpine3.9, but has reappeared in 13-alpine3.9.

AdoptOpenJDK

Might be a good choice to consider

- Free, open
- Built from OpenJDK
- Community maintained
- Backporting patches are community-led
- Supporting Java in containers
- For Alpine Linux they add a full glibc package made for Alpine Linux, and it replaces musl. Somewhat odd, and increases size of Alpine.

Containers images:

- latest (Full JDK)
- slim (Stripped down JDK)
- alpine (Full JDK)
- alpine-slim (Stripped down JDK)

Portola

OpenJDK Port to Alpine

https://openjdk.java.net/projects/portola/

Try it out... https://www.katacoda.com/courses/docker/playground

```
docker run -it --rm linead/portola jshell
System.out.println("Hello")
```

Portola

OpenJDK Port to Alpine

https://openjdk.java.net/projects/portola/

- Sub project of OpenJDK
- Port to Alpine and support musl C library.
- Alpine Linux is ~4.5MB, very small
- glibc vs musl issue

Try it out... https://www.katacoda.com/courses/docker/playground

```
docker run -it --rm linead/portola jshell
System.out.println("Hello")
```

Freemium

- Zing ($) product is a JVM tuned for performance
- "A Better JVM"
- High performance, mostly for financial systems

Zulu Base Containers

- azul/zulu-openjdk-alpine
- azul/zulu-openjdk-centos
- azul/zulu-openjdk-debian
- azul/zulu-openjdk

Zulu

- OpenJDK 15EA, 14EA, 13, 11, 8, and 7.
- Community Edition, free version
- Enterprise Edition, commercial support
- Implementation for Class Data Sharing (CDS)

Dockerfile

Easy language, powerful containers

Domain specific language for Dockerfiles

- Dockerfile is text source code
- Infrastructure-as-code defines container contents
- Comment lines start with #
- Other lines are: INSTRUCTION arguments
- The instruction set is small
- Instructions are ALL CAPS
- Code must start with FROM instruction

Two line Dockerfile and you have a web server

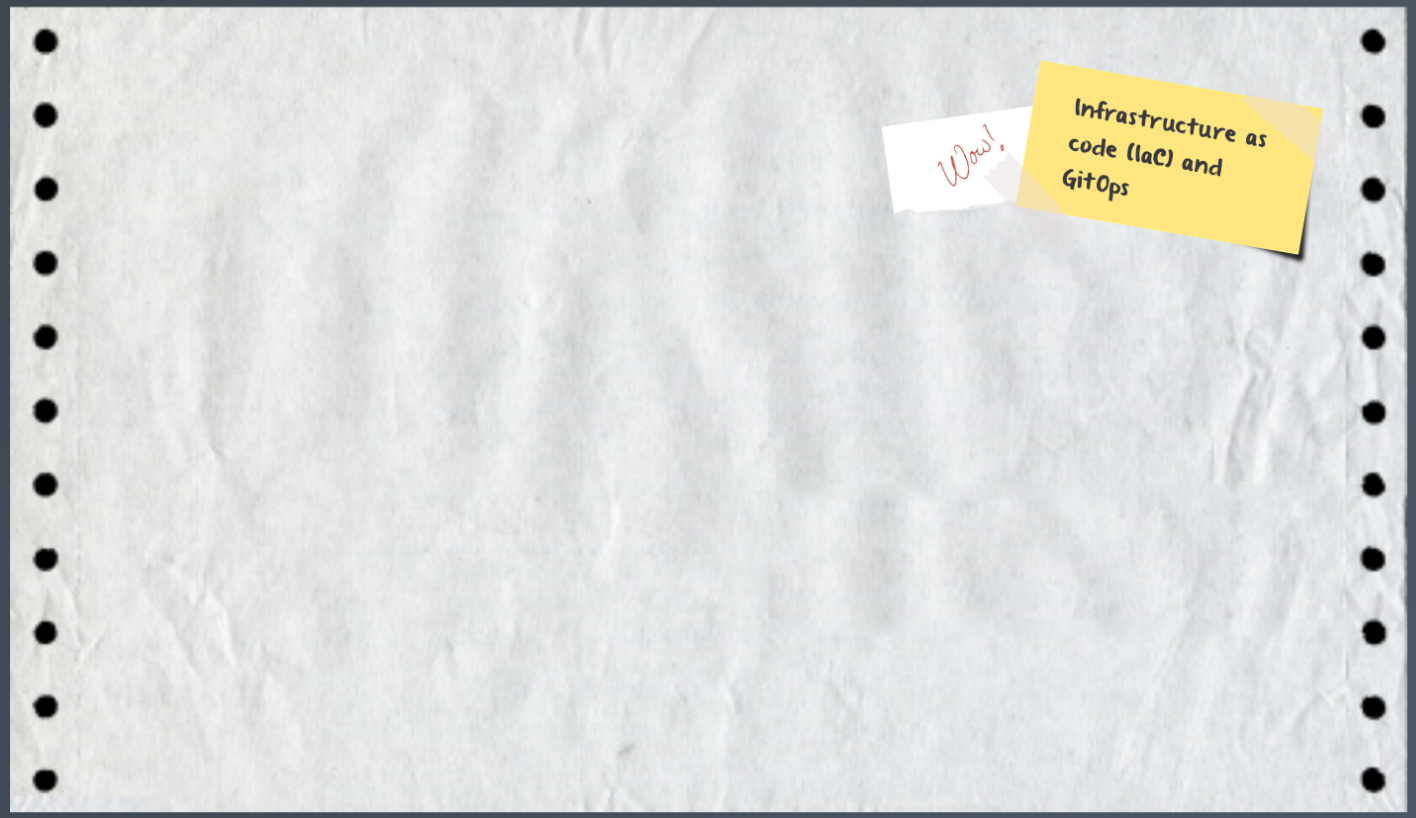
```
FROM nginx
COPY mysite /usr/share/nginx/html
```

Dockerfile Instruction Set (17)

Common:
FROM
RUN
COPY
CMD
ENTRYPOINT

Secondary:
LABEL
EXPOSE
ENV
ADD
VOLUME
USER
WORKDIR
ARG
ONBUILD
STOPSIGNAL
HEALTHCHECK
SHELL

Full Payloads with Few Lines



Full Payloads with Few Lines

NGINX

```
FROM nginx
ADD html /usr/share/nginx/html/
```

Wow!

Infrastructure as code (IaC) and GitOps

Full Payloads with Few Lines

NGINX

```
FROM nginx
ADD html /usr/share/nginx/html/
```

PostgreSQL

```
FROM postgres:9.5.3
ADD data /data/
RUN chmod -R 777 /data
ADD sql /docker-entrypoint-initdb.d/
```

Wow!

Infrastructure as code (IaC) and GitOps

Full Payloads with Few Lines

NGINX

```
FROM nginx
ADD html /usr/share/nginx/html/
```

Wow!

Infrastructure as code (IaC) and GitOps

PostgreSQL

```
FROM postgres:9.5.3
ADD data /data/
RUN chmod -R 777 /data
ADD sql /docker-entrypoint-initdb.d/
```

JAR

```
FROM java:openjdk-8-jre-alpine
ADD processor.jar processor.jar
ENTRYPOINT ["java", "-Dserver.port=9001", "-jar", "processor.jar"]
EXPOSE 9001
```

Multi-stage Dockerfile

New to Docker 17.05,
May 2017

Provide capabilities for [developers] to produce sleek images without the cruft of the intermediate build artifacts.

- Moby #31067

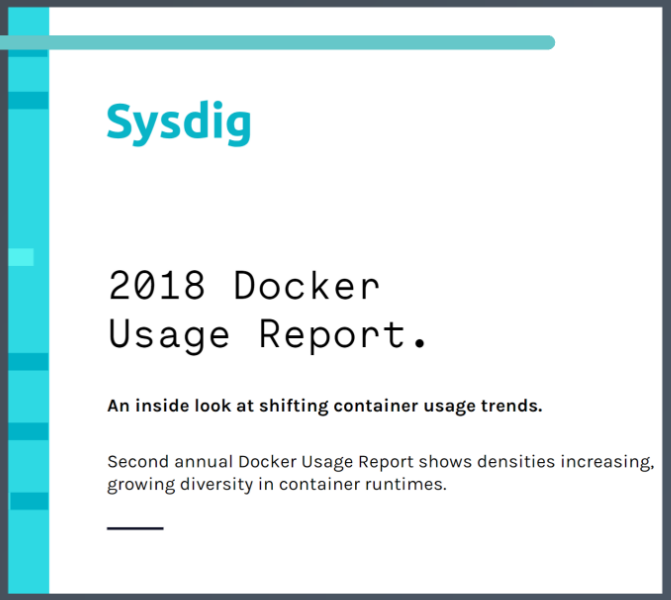
```
FROM golang:1.7.3 AS builder
WORKDIR /go/src/github.com/alexellis/href-counter/
RUN go get -d -v golang.org/x/net/html
COPY app.go    .
RUN CGO_ENABLED=0 GOOS=linux go build -a -installsuffix cgo -o app .

FROM alpine:latest
RUN apk --no-cache add ca-certificates
WORKDIR /root/
COPY --from=builder /go/src/github.com/alexellis/href-counter/app .
CMD ["./app"]
```

https://www.katacoda.com/courses/docker/multi-stage-builds

Top 12 app components in containers

- Java
- Redis
- Etcd
- Nginx
- PostgresSQL
- Fluentd
- Apache
- Elastic
- MongoDB
- Php
- RabbitMQ
- Tomcat


Sysdig

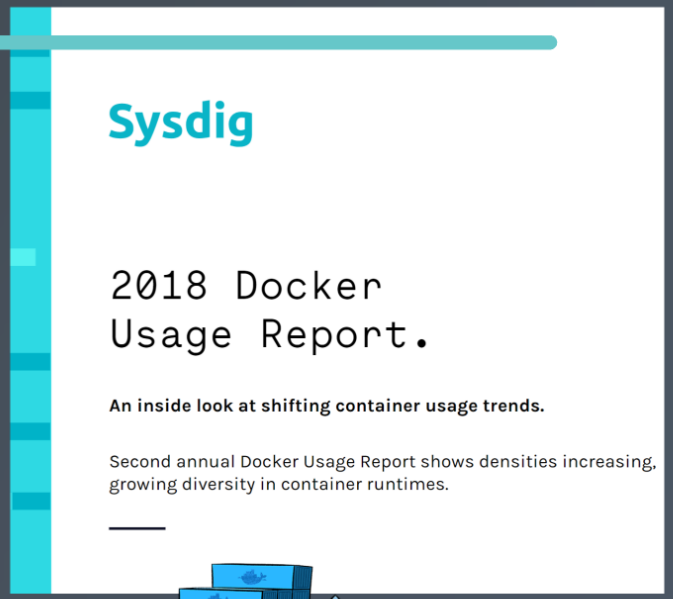
2018 Docker Usage Report.

An inside look at shifting container usage trends.

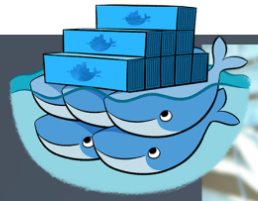
Second annual Docker Usage Report shows densities increasing, growing diversity in container runtimes.

Top 12 app components in containers

- Java
- Redis
- Etcd
- Nginx
- PostgresSQL
- Fluentd
- Apache
- Elastic
- MongoDB
- Php
- RabbitMQ
- Tomcat



Or look at most popular



https://hub.docker.com/search?q=&type=image

Shrinky Dinks

Shrinky Dinks

Alex Goodman

Tool: **Dive**

- Explores container images
- Explores layer contents
- Shrink image size
- Docker and OCI images
- CLI for pipelines or UI

```
1 / 1                                   Tilix: Default
1: Terminal
[Layers]
Cmp Image ID                    Size  Command
    sha256:cd7100a72410606589  4.1 MB  FROM sha256:cd7100a72410606589
    sha256:f03b1ccbacace8c82e  2.1 kB  #(nop) ADD file:63d4894bd0857354b
    sha256:d2a26525cdac6c4358     0 B  mkdir /root/example
    sha256:9f50561518484bb62a  2.1 kB  cp /somefile.txt /root/example/so
    sha256:edf209d09263ab03c7  2.1 kB  cp /somefile.txt /root/example/so
    sha256:bb70c7aab83602e8a6  2.1 kB  cp /somefile.txt /root/example/so
    sha256:b0a712d3d00bc83c48  2.1 kB  mv /root/example/somefile3.txt /r
    sha256:05e8660b8f90e77b93     0 B  rm -rf /root/example/

[Image & Layer Details]

Layer Command
/bin/sh -c cp /somefile.txt /root/example/somefile3.txt

Image efficiency score: 99 %
Potential wasted space: 6.2 kB

Count   Total Space  Path
    2        4.2 kB  /root/example
    2        2.1 kB  /root/example/somefile3.txt

[● Aggregated Layer Contents]
Permission     UID:GID       Size  Filetree
drwxr-xr-x         0:0     805 kB  ├─⊕ bin
drwxr-xr-x         0:0        0 B  ├─ dev
drwxr-xr-x         0:0     251 kB  ├─⊕ etc
drwxr-xr-x         0:0        0 B  ├─ home
drwxr-xr-x         0:0     2.7 MB  ├─⊕ lib
drwxr-xr-x         0:0        0 B  ├─ media
drwxr-xr-x         0:0        0 B  │   ├─ cdrom
drwxr-xr-x         0:0        0 B  │   ├─ floppy
drwxr-xr-x         0:0        0 B  │   └─ usb
drwxr-xr-x         0:0        0 B  ├─ mnt
dr-xr-xr-x         0:0        0 B  ├─ proc
drwx------         0:0     6.2 kB  ├─ root
drwxr-xr-x         0:0     6.2 kB  │   └─ example
-rw-r--r--         0:0     2.1 kB  │       ├─ somefile1.txt
-rw-r--r--         0:0     2.1 kB  │       ├─ somefile2.txt
-rw-r--r--         0:0     2.1 kB  │       └─ somefile3.txt
drwxr-xr-x         0:0        0 B  ├─ run
drwxr-xr-x         0:0     213 kB  ├─⊕ sbin
-rw-rw-r--         0:0     2.1 kB  ├─ somefile.txt
drwxr-xr-x         0:0        0 B  ├─ srv
drwxr-xr-x         0:0        0 B  ├─ sys
drwxrwxrwx         0:0        0 B  ├─ tmp
drwxr-xr-x         0:0     176 kB  ├─⊕ usr
drwxr-xr-x         0:0        0 B  └─⊕ var

^C Quit | ^Space Switch view | ^/ Filter files | Space Collapse dir | ^A Added files | ^R Removed files | ^M Modified files | ^U Unmodified fi
```

https://github.com/wagoodman/dive

Minify and Secure Containers

Java application images:

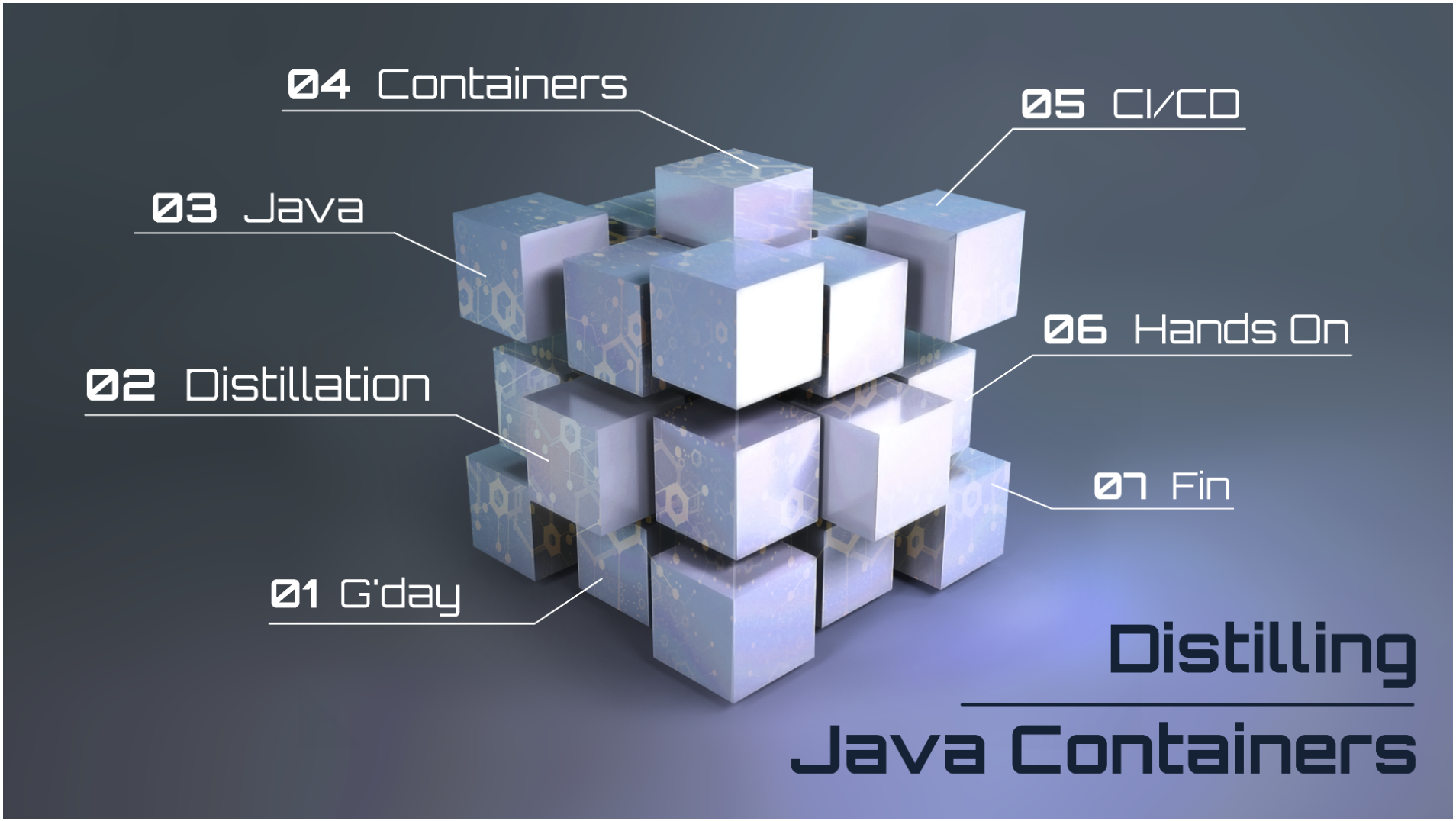
from ubuntu:14.04 - **743.6 MB => 100.3 MB**

Node.js application images:

from ubuntu:14.04 - 432MB => 14MB (by 30.85X)
from debian:jessie - 406MB => 25.1MB (16.21X)
from node:alpine - 66.7MB => 34.7MB (1.92X)
from node:distroless - 72.7MB => 39.7MB (1.83X)

Python application images:

from ubuntu:14.04 - 438MB => 16.8MB (by 25.99X)
from python:2.7-alpine - 84.3MB => 23.1MB (by 3.65X)
from python:2.7.15 - 916MB => 27.5MB (by 33.29X)
from centos:7 - 647MB => 23MB (by 28.57X)
from centos/python-27-centos7 - 700MB => 24MB (29.01X)
from python2.7:distroless - 60.7MB => 18.3MB (by 3.32X)


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

It's Just a Tar of Tars

Home of the Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana

It's Just a Tar of Tars

```
$ docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8d691f585fa8: Pull complete
8ccd02d17190: Pull complete
4719eb1815f2: Pull complete
200531706a7d: Pull complete
eed7c26916cf: Pull complete
e1285fcc6a46: Pull complete
Digest:
sha256:fe80393a67c7058590ca6b6903f64e35b50fa411b0496f604a85c526fb5bd2d2
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest

$ docker save --output redis.tar redis
```

Home of the Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana

It's Just a Tar of Tars

```
$ docker pull redis
Using default tag: latest
latest: Pulling from library/redis
8d691f585fa8: Pull complete
8ccd02d17190: Pull complete
4719eb1815f2: Pull complete
200531706a7d: Pull complete
eed7c26916cf: Pull complete
e1285fcc6a46: Pull complete
Digest:
sha256:fe80393a67c7058590ca6b6903f64e35b50fa411b0496f604a85c526fb5bd2d2
Status: Downloaded newer image for redis:latest
docker.io/library/redis:latest

$ docker save --output redis.tar redis
```

```
$ tar -tf redis.tar
1167eb15dc99e9c437519ce5dfcfc46fa776b5be1b1acb86636b0c434d266eec/
   VERSION
   json
   layer.tar
3084577582f46c3659e22d20aa16607803299745d87d4f4d36bc9a632bbcdaec/
799674d5f31766be4c5b95433657c01959c242160a38cfc6ccdbd5af46cf31f2/
bad490e5568e6fad2311dc0217b79408eb4cd61b56f9d6b22ad6b96179241525/
f9c316ad0bbad00d22843076db519a1fc385b305dc908fd7c7027125e1e458e3/
fd359d877a550df140b3480605e4e56d0aa245ad2d6a6ba4f1af70b62fd7549e/
manifest.json
repositories
```

Home of the Original **Muffuletta**
Central Grocery and Deli
New Orleans, Louisiana

Container Building

Many way to build a container

Gradle / Maven plugins

- Jib, Google maven and Gradle
- Ben Muschko's Gradle plugin
- Fabric 8/docker-maven
- Palantir/gradle-docker

Container Build Tools

- Kaniko
- Knative
- Buildah with Podman
- Tekton
- Orca build
- Img

Running Containers without Docker using Interactive Browser-Based Scenarios

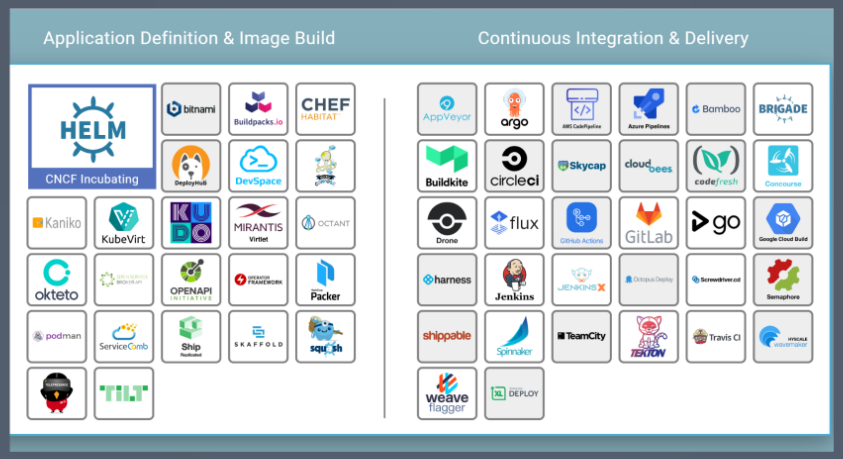
https://www.katacoda.com/courses/containers-without-docker

CLOUD NATIVE COMPUTING FOUNDATION

https://landscape.cncf.io/

CD.FOUNDATION

https://cd.foundation/



PROJECTS

Jenkins

JENKINS X

Spinnaker

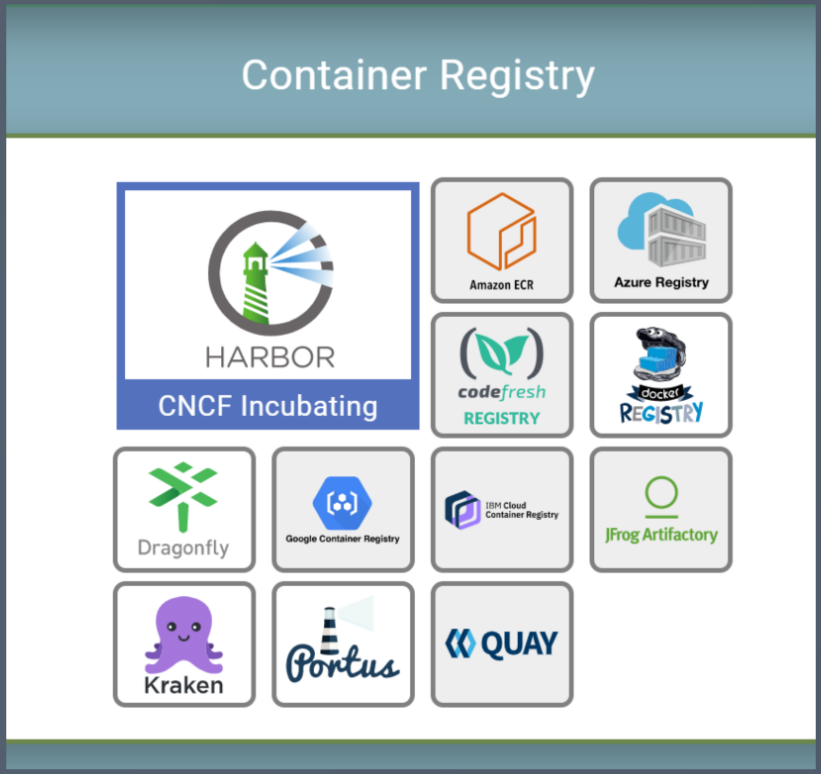
TEKTON

Explore

- Build tools
- CI/CD tools
- Registries

Container Registries

What hosts your library of private images?



Container Security Scanning

Static analysis
in your pipelines

Runtime analysis
on your cluster

Container Security Scanning

Static analysis in your pipelines

Runtime analysis on your cluster

Clair	**CVE scans**
Banyan Collector	Linting and scanning
Docker Bench for Security	Inspection reports
Cilium	Runtime network connectivity
Anchore	CVE scans
OpenSCAP Workbench	SCAP and CVE
Dagda	CVE scans
Notary	Image signing
Grafaes	Custom rules, IBM Vulnerability Adviser
Falco	**Sysdig auditing tool**

OpenSCAP
Audit, Fix and be Merry

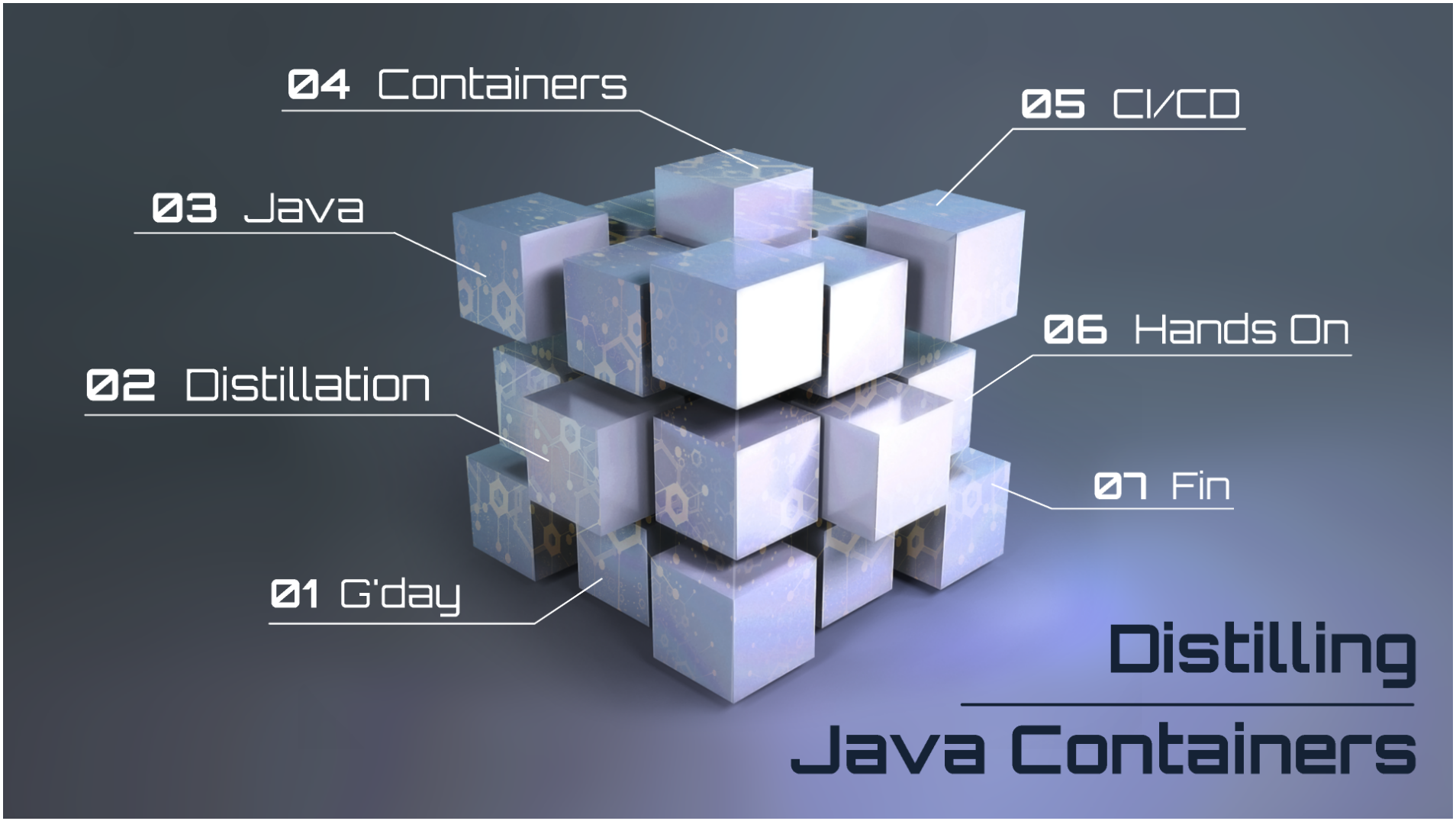
clair

NIST
NVD
National Vulnerability Database

Falco

notary

cilium


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Hands On Scenario

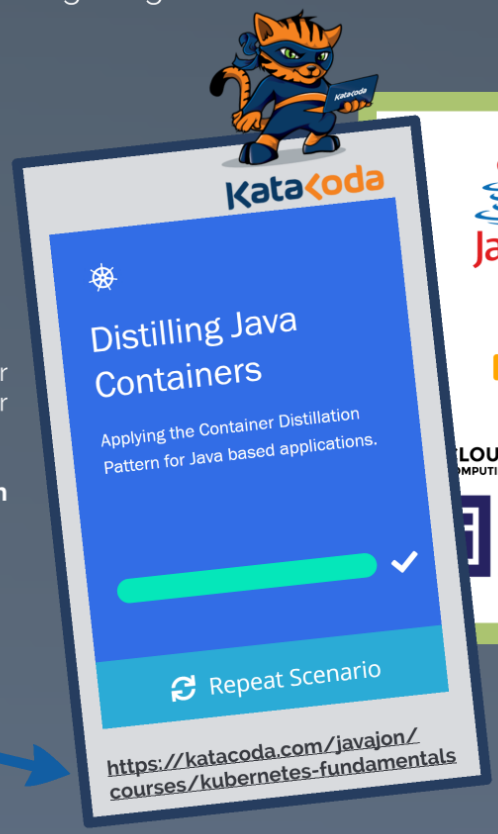
Techniques for getting Java into containers

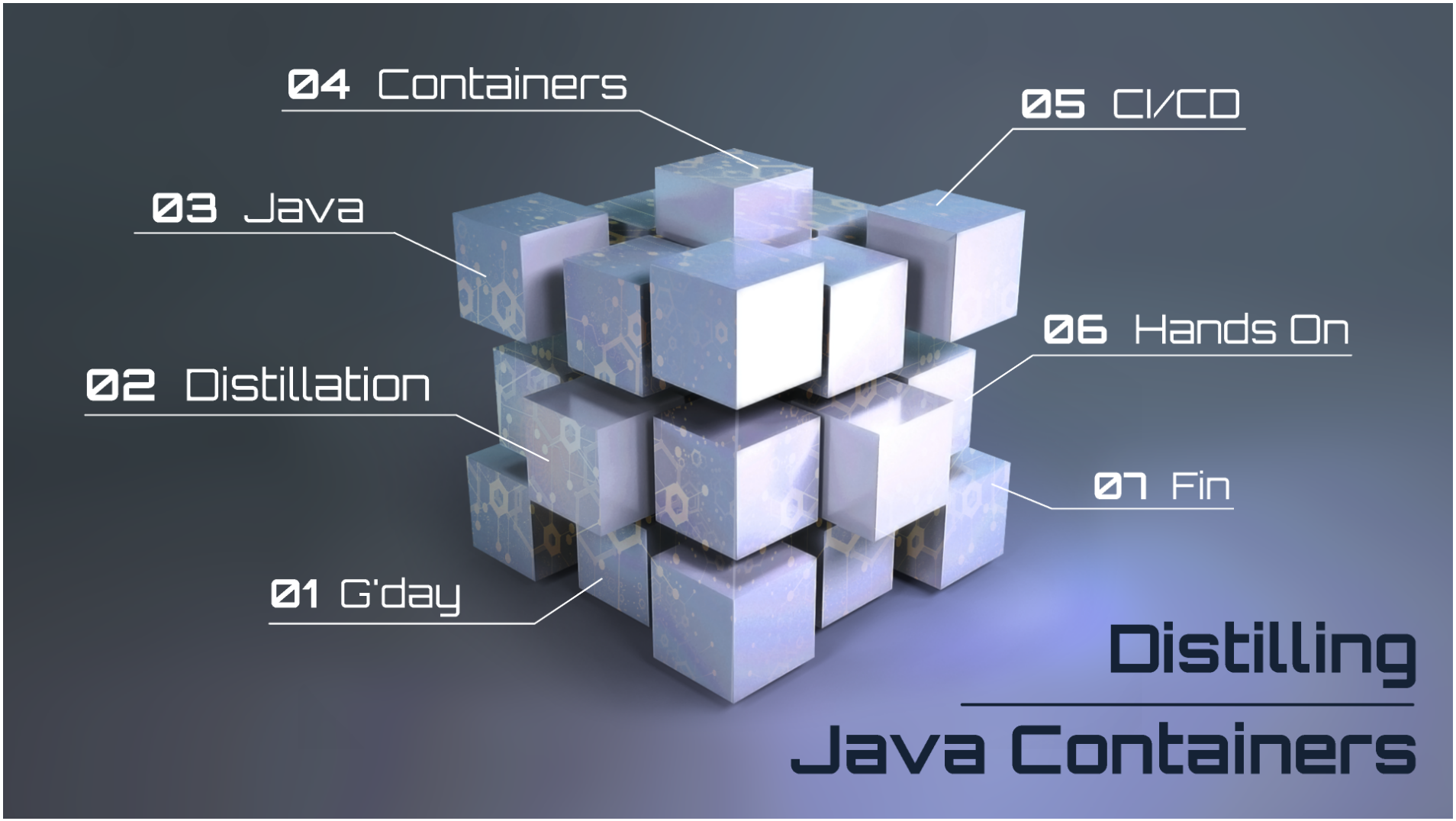
Distilling Java Containers

The scenario walks through a series of steps for distilling a container to its minimum to support your Java application.

What's the smallest image size we can achieve, with Java as the source, running on Linux?

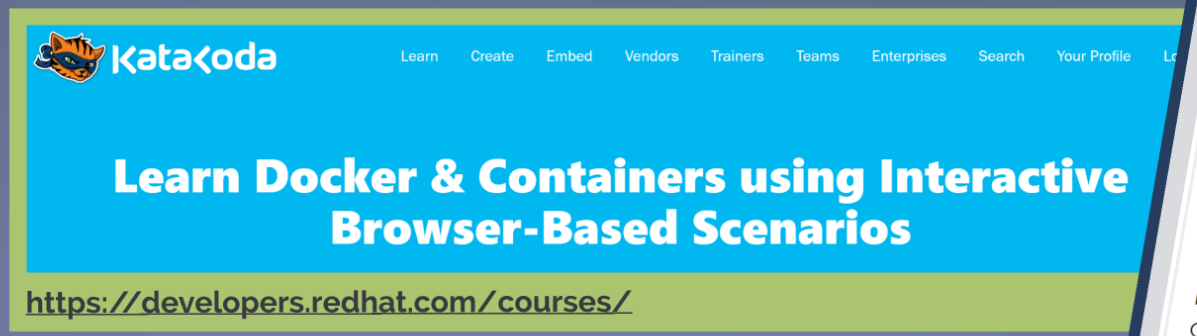
Let's find out...



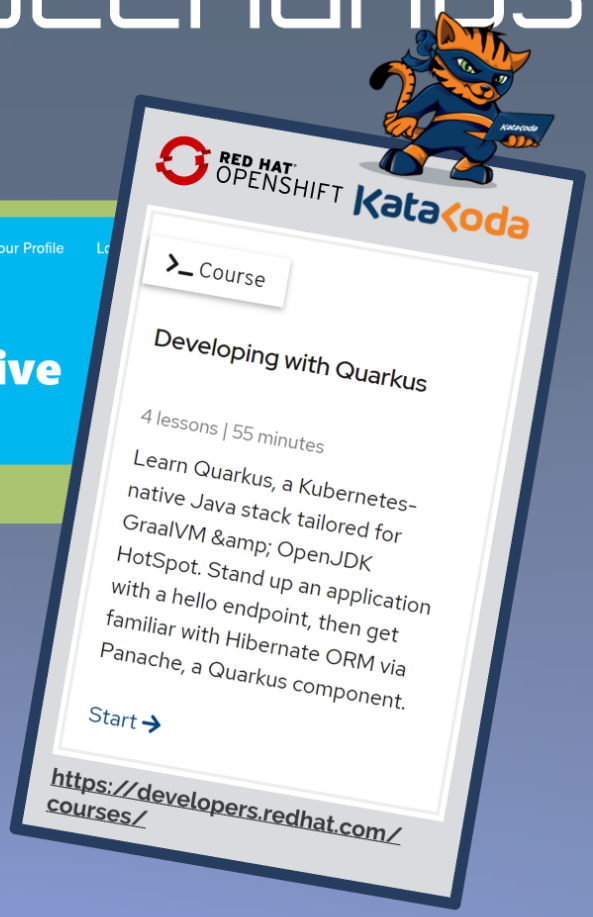

04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers

Other Katacoda Scenarios

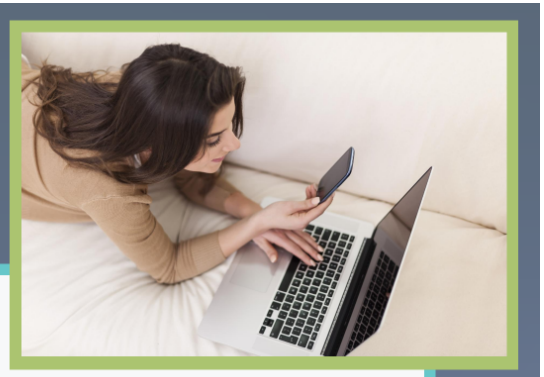
Curious minds want to know



Learning Docker and Quarkus



Additional Reads



Jlink
https://docs.oracle.com/javase/9/tools/jlink.htm

Multi-stage
https://docs.docker.com/develop/develop-images/multistage-build/

Portola
https://openjdk.java.net/projects/portola/

GraalVM
https://www.graalvm.org/

Alpine
https://en.wikipedia.org/wiki/Alpine_Linux

Docker JVM article
https://www.javacodegeeks.com/2018/12/docker-jvm.html

Quarkus
https://quarkus.io/

Native-image
https://www.graalvm.org/docs/reference-manual/native-image/

Matthew Gilliard's blog
http://blog.gilliard.lol/

Thank you

Veni, Vidi, Didici

I came, I saw, I learned

Containers have become the preferred packaging for cloud native.
Java continues to match these new targets.

Distillation Pattern · Jlink · Alpine · GraalVM · Native

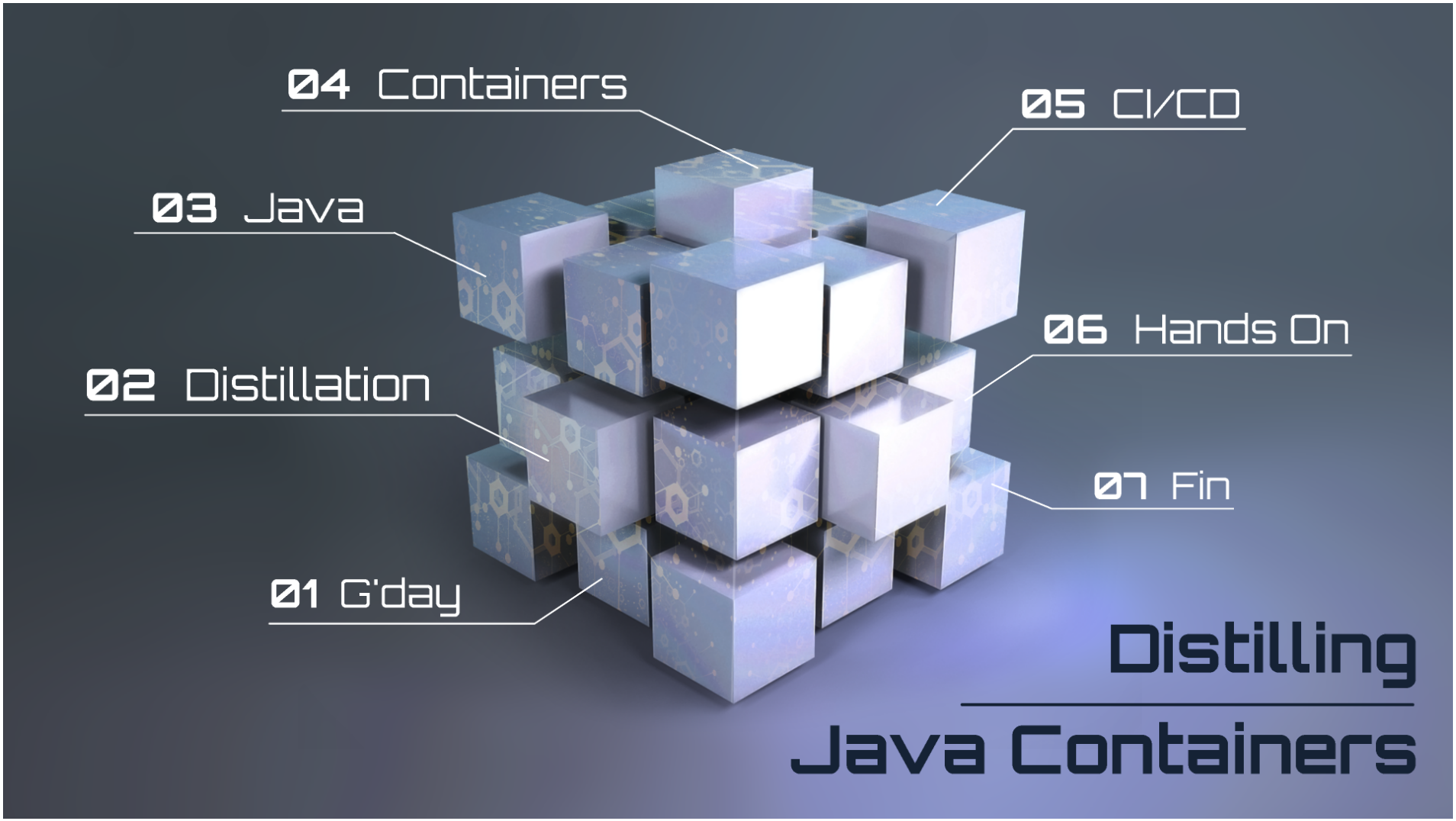
jonathan.johnson
@dijure.com

twitter.com/
javajonjohn

github.com/
javajon

linkedin.com/in/
javajon

nofluffjuststuff.com/
conference/speaker/
jonathan_johnson


04 Containers
05 CI/CD
03 Java
06 Hands On
02 Distillation
07 Fin
01 G'day
Distilling
Java Containers