

Migrate early, migrate often! JDK release cadence strategies

Dan Heidings
Eclipse OpenJ9 Project Lead
Interpreter Lead, IBM Runtimes
 @danheidings
 DanHeidings

Theresa Mammarella
Eclipse OpenJ9 Software Developer
 @t_mammarella
 theresa-m

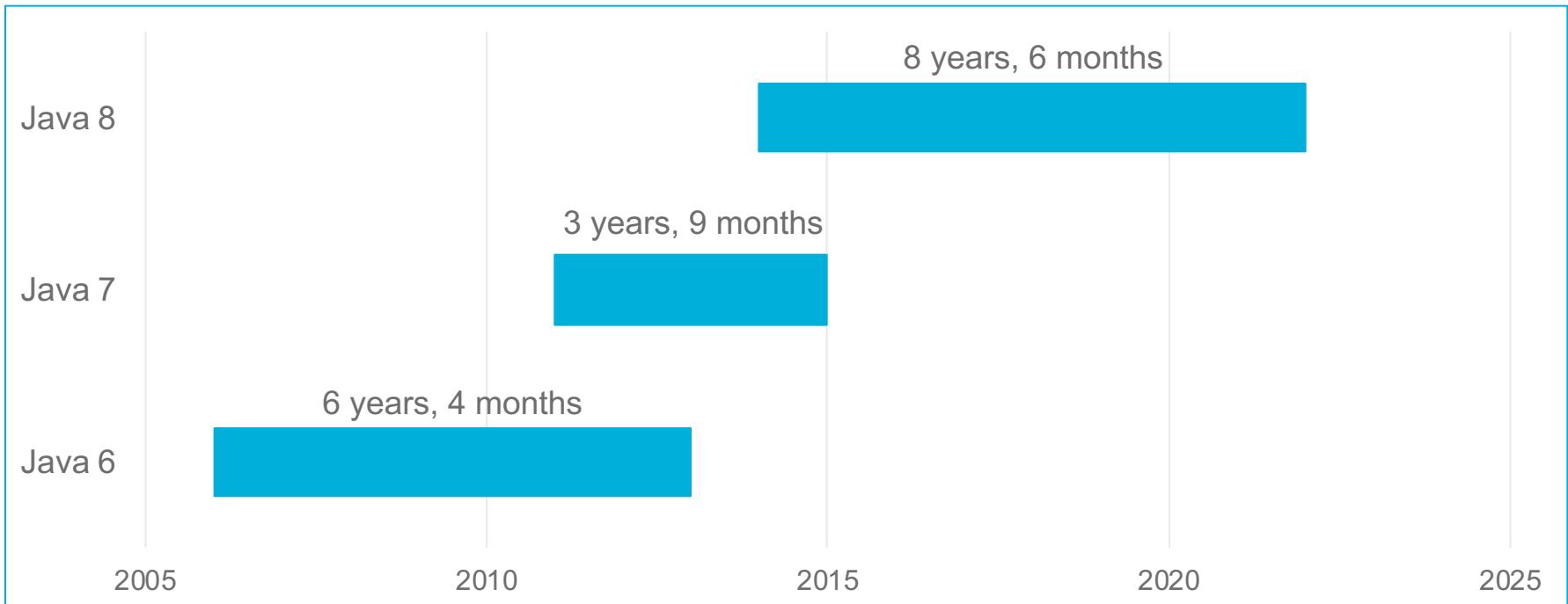
Important disclaimers

- THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
- WHILST EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.
- ALL PERFORMANCE DATA INCLUDED IN THIS PRESENTATION HAVE BEEN GATHERED IN A CONTROLLED ENVIRONMENT. YOUR OWN TEST RESULTS MAY VARY BASED ON HARDWARE, SOFTWARE OR INFRASTRUCTURE DIFFERENCES.
- ALL DATA INCLUDED IN THIS PRESENTATION ARE MEANT TO BE USED ONLY AS A GUIDE.
- IN ADDITION, THE INFORMATION CONTAINED IN THIS PRESENTATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM, WITHOUT NOTICE.
- IBM AND ITS AFFILIATED COMPANIES SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.
- NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:
 - CREATING ANY WARRANT OR REPRESENTATION FROM IBM, ITS AFFILIATED COMPANIES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS

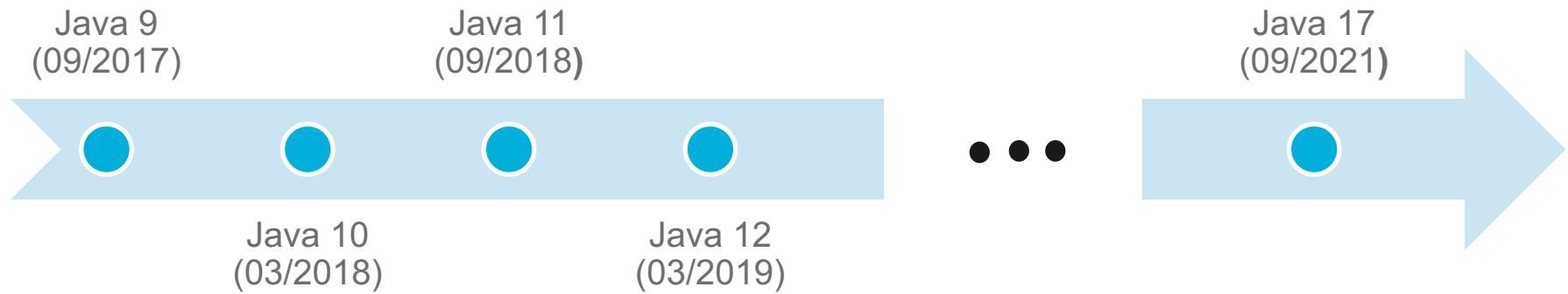
Outline

- The brave new world of Java
- Always be migrating
- Continuous migration strategies

Historical Java release model



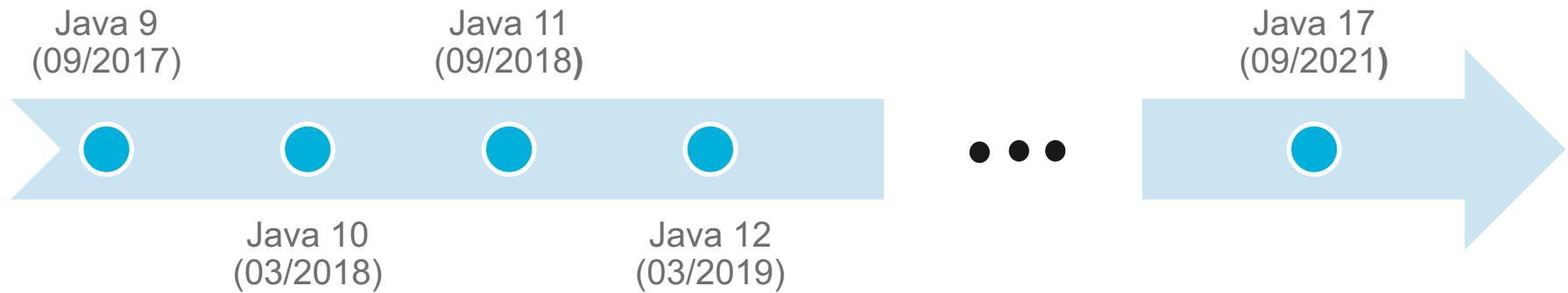
Brave new world



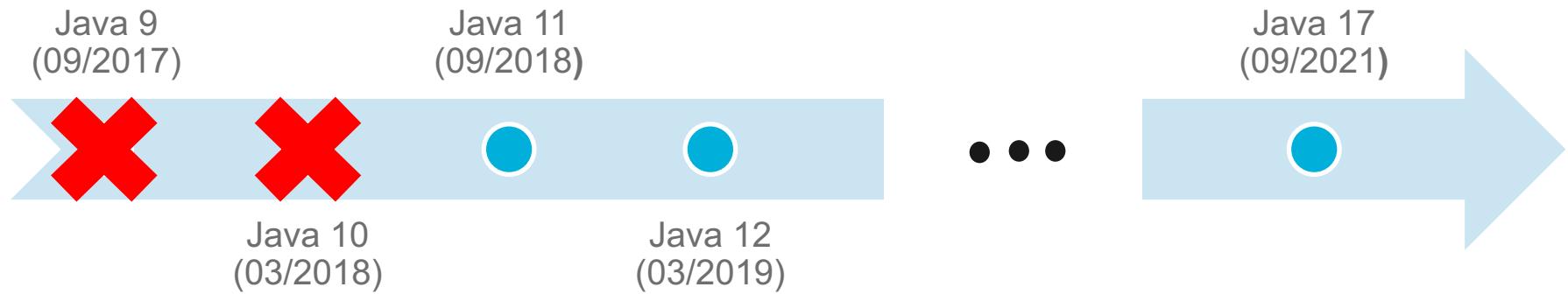


- Author(s): Videoplasty.com
- [*Man on a Treadmill Cartoon.svg*](#) from Wikimedia Commons
- License: [Creative Commons Attribution-ShareAlike 4.0](#)

Brave new world

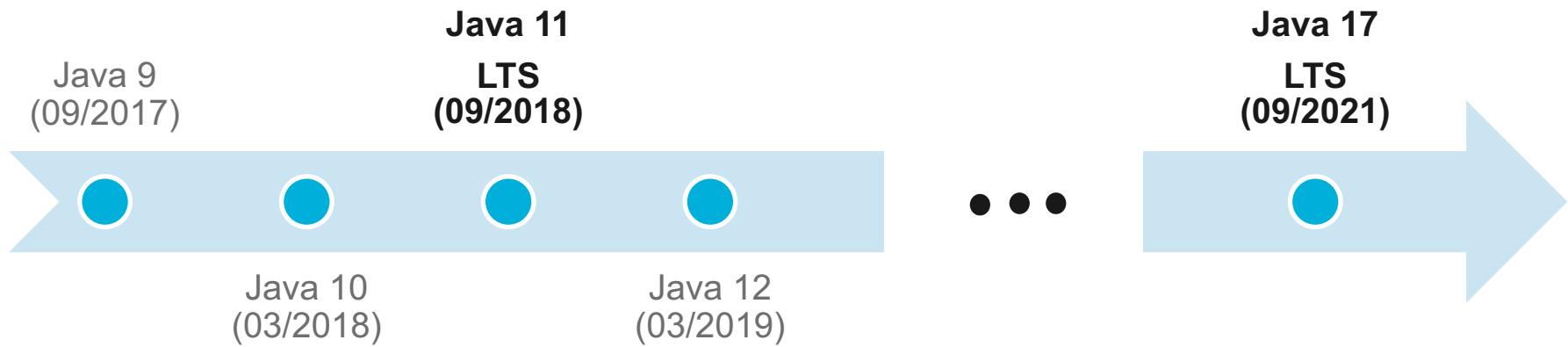


6 month release cadence



LTS = Long Term Support?

LTS: up to the community





Prebuilt OpenJDK binaries

adoptopenjdk.net

Docker Hub

hub.docker.com/u/adoptopenjdk



Support Levels

... bugs will be fixed on a “best effort” basis ...

... for higher levels of assurance you should contact commercial companies offering support ...

Who to contact?



Who to contact?

Everyone! Lots of vendors are providing support for OpenJDK builds:*

IBM

Red Hat

Amazon

Oracle

BellSoft

Azul

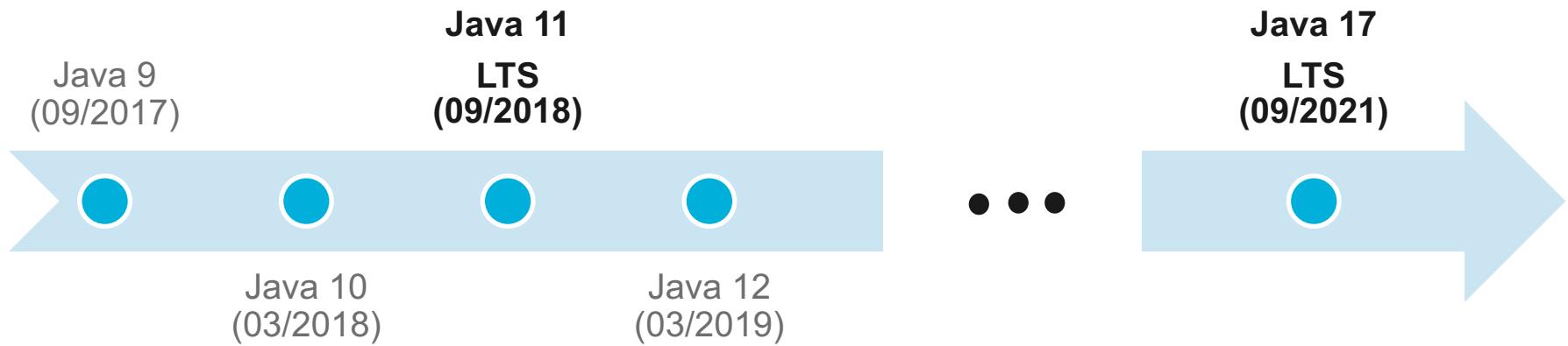
JClarity

Others?

(List may be incomplete / inaccurate)*



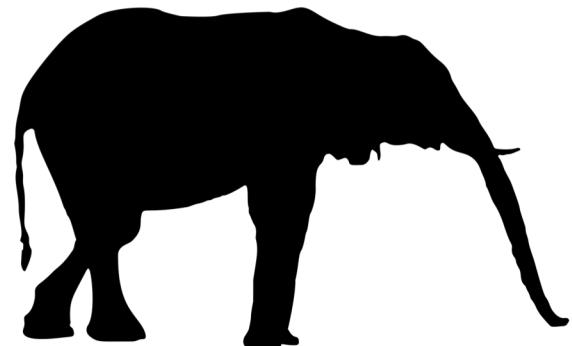
LTS: up to the community



Always be migrating

“eat the elephant a bite at a time”

- The gap between LTS releases is HUGE
- How can an application keep up?
 - Move to a stable base
 - Track changes in 6 month pieces



Migrate to a stable base: Java 8

“the best time to plant a tree is either 20 years ago or today”

- Migration 6-7-8 is mostly smooth
- Java 8 is the oldest stable base you should use, support through 2025*
- By not upgrading you are leaving performance improvements on the table!

Track PAST changes: Java 9

Major ecosystem changes!

- Java Platform Module System makes internal APIs inaccessible
- API elements are deprecated for removal (Java EE, Corba and more)
... **And gone in Java 11!**
- Version system has changed, URLClassLoader is replaced, Etc.

Java

JAN/FEB 2019

Hello, Java enthusiasts!

<https://developer.ibm.com/newsletters/java/>

JEP 277: Enhanced Deprecation

@Deprecated

- Element is dangerous or a better alternative exists, use is discouraged

@Deprecated(since="9")

- Since(): Release in which API element was deprecated

@Deprecated(since="1.4", forRemoval=true)

- forRemoval(): Marked true if API element will be removed in a future release

Continuous integration pipeline

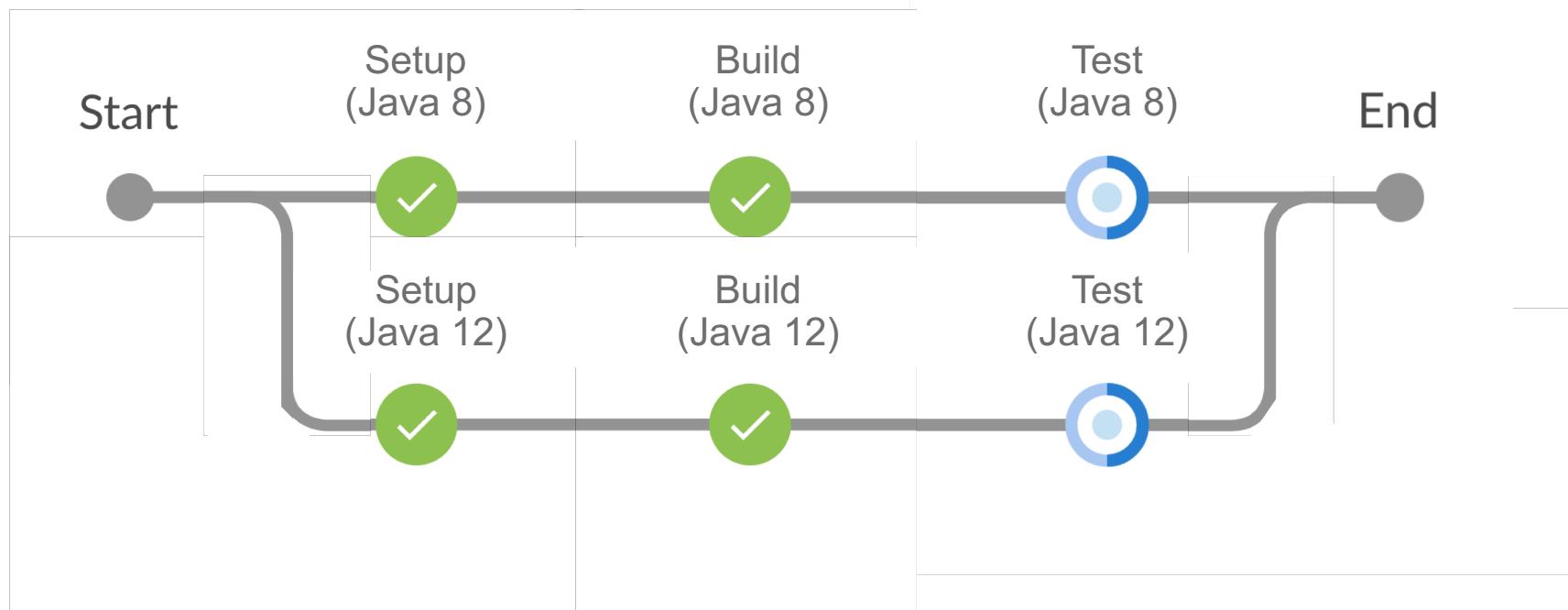


javac warnings

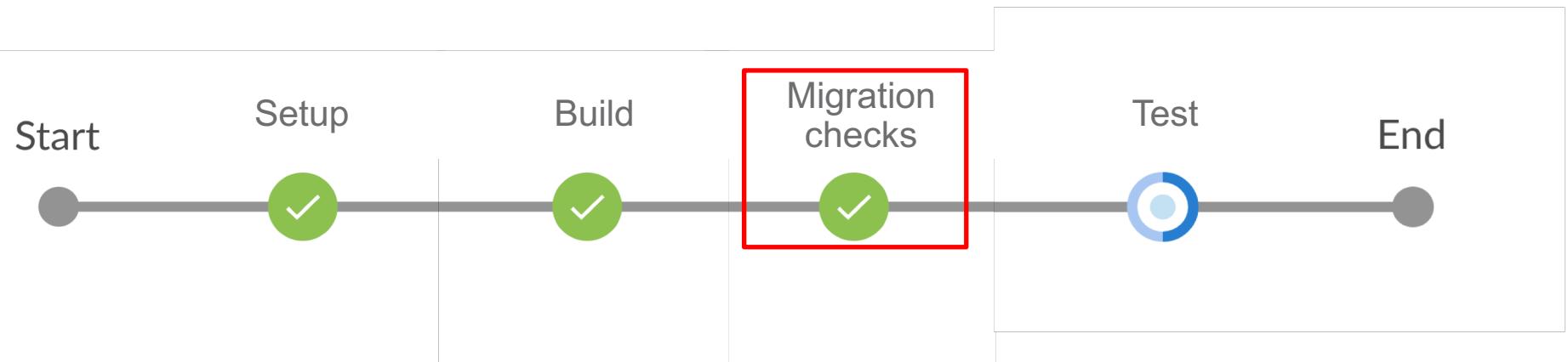
- Pay attention to warnings!
- Fixing them gets you clean on the current release
- Treat warnings as errors so build will fail if warnings are present

```
javac -Werror App.jar
```

Continuous migration strategies



Continuous migration strategies



jdeps Overview

JDEPS = Java Class Dependency Analyzer

- Java 8+ JDKs
- Analyzes the dependencies by class or package (default) level
- Not just for migrating to modularity!

jdeps docker commandline

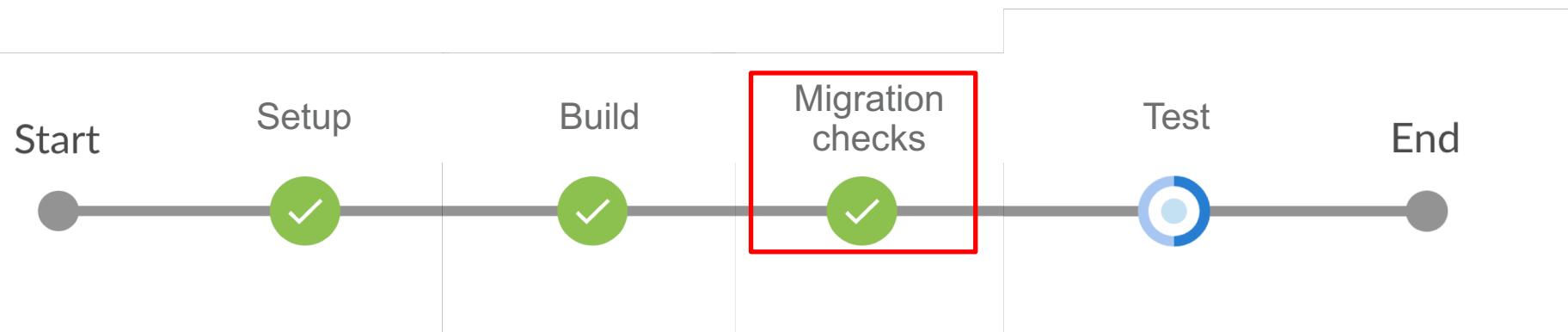
```
docker run --rm -it -v `pwd`/DemoApp:/DemoApp adoptopenjdk/openjdk12-openj9  
        jdeps /DemoApp/target/DemoApp.jar
```

Demo

github.com/theresa-m/migrateearly-demo

hub.docker.com/u/adoptopenjdk

Continuous migration strategies



```
docker run --rm -it -v `pwd`/DemoApp:/DemoApp adoptopenjdk/openjdk12-openj9  
jdeps --jdkinternals /DemoApp/target/DemoApp.jar
```

jdeprscan Overview

- Java 9+ JDKs
- Scans jar/class files to identify deprecated API elements
- Deprecated elements may cause behavioral issues or be removed

Demo

github.com/theresa-m/migrateearly-demo

hub.docker.com/u/adoptopenjdk

jdeprscan docker commandline

```
docker run --rm -it -v `pwd`/DemoApp:/DemoApp adoptopenjdk/openjdk12-openj9  
    jdeprscan --for-removal /DemoApp/target/DemoApp.jar
```

Continuous migration strategies



Multi-release JAR files

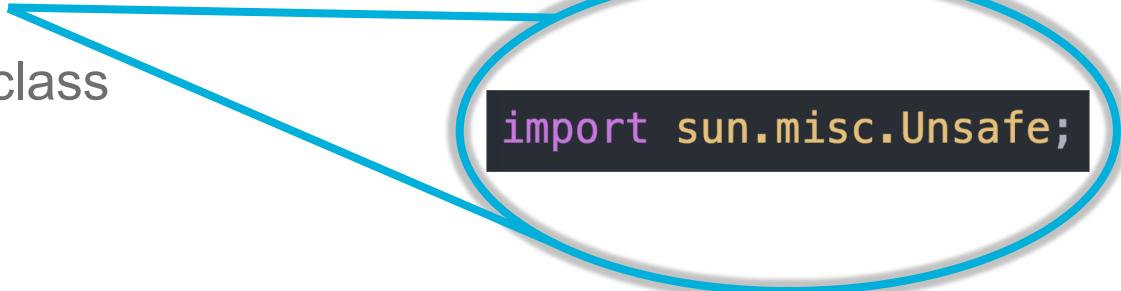
- Java 9+ SDKs
- Enables JAR files to support multiple Java versions
- Combination of shared and version specific classes

`sun.misc.Unsafe` to `java.lang.invoke.VarHandle`

Multi-release JAR files

DemoApp.jar

- migrate
 - early
 - Demo.class
 - DemoRunnable.class
- META-INF
 - MANIFEST.MF



```
import sun.misc.Unsafe;
```

Multi-release JAR files

DemoApp.jar

- migrate
 - early
 - Demo.class
 - DemoRunnable.class
- META-INF
 - MANIFEST.MF



MultiReleaseDemoApp.jar

- migrate
 - early
 - Demo.class
 - VersionedCode.class
 - DemoRunnable.class
- META-INF
 - MANIFEST.MF

Multi-release JAR files

DemoApp.jar

- migrate
 - early
 - Demo.class
 - DemoRunnable.class
- META-INF
 - MANIFEST.MF



MultiReleaseDemoApp.jar

- migrate
 - early
 - Demo.class
 - VersionedCode.class
 - DemoRunnable.class
- META-INF
 - MANIFEST.MF
 - versions
 - 9
 - VersionedCode.class

Demo

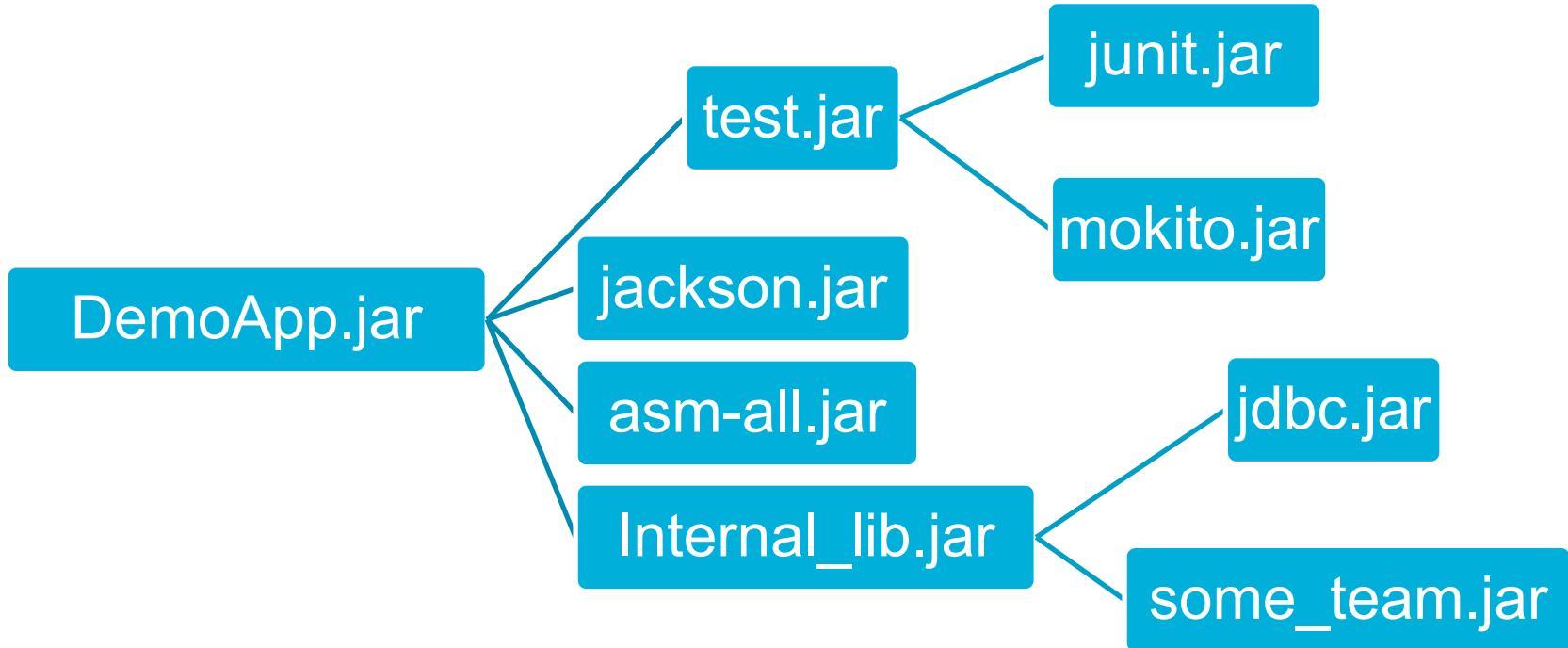
github.com/theresa-m/migrateearly-demo

hub.docker.com/u/adoptopenjdk

Continuous migration strategies



Dependencies



Strategies

- Expect to run parallel CI pipelines at least once every 6 months
- Run jdeps & jdeprscan as part of every build to prevent new regressions
- Use multi-release jars if no workaround for current and latest versions
- Beware the dependency graph

Key points to take away

- The Java ecosystem is delivering innovation faster than ever before!
- The free lunch is over. Migrate early, migrate often
- Even on an LTS, use the JDK tools to prepare for the next migration



<https://adoptopenjdk.net/releases.html?variant=openjdk11&jvmVariant=openj9>

<https://adoptopenjdk.net/releases.html?variant=openjdk8&jvmVariant=openj9>



<https://github.com/eclipse/openj9>



<https://openj9.slack.com/>



Thank you!

Are you suffering from a large Java support bill from Oracle?

IBM Runtimes for Business

Enterprise support for open source Java

Security and bug fixes
1-800-CALL-IBM support

Trusted and Secure
25+ years of Java expertise

Performance advantages
66% smaller footprint
100% throughput performance

Migrate your app with no app changes

Save on Cost
Only pay for support for what you need

