

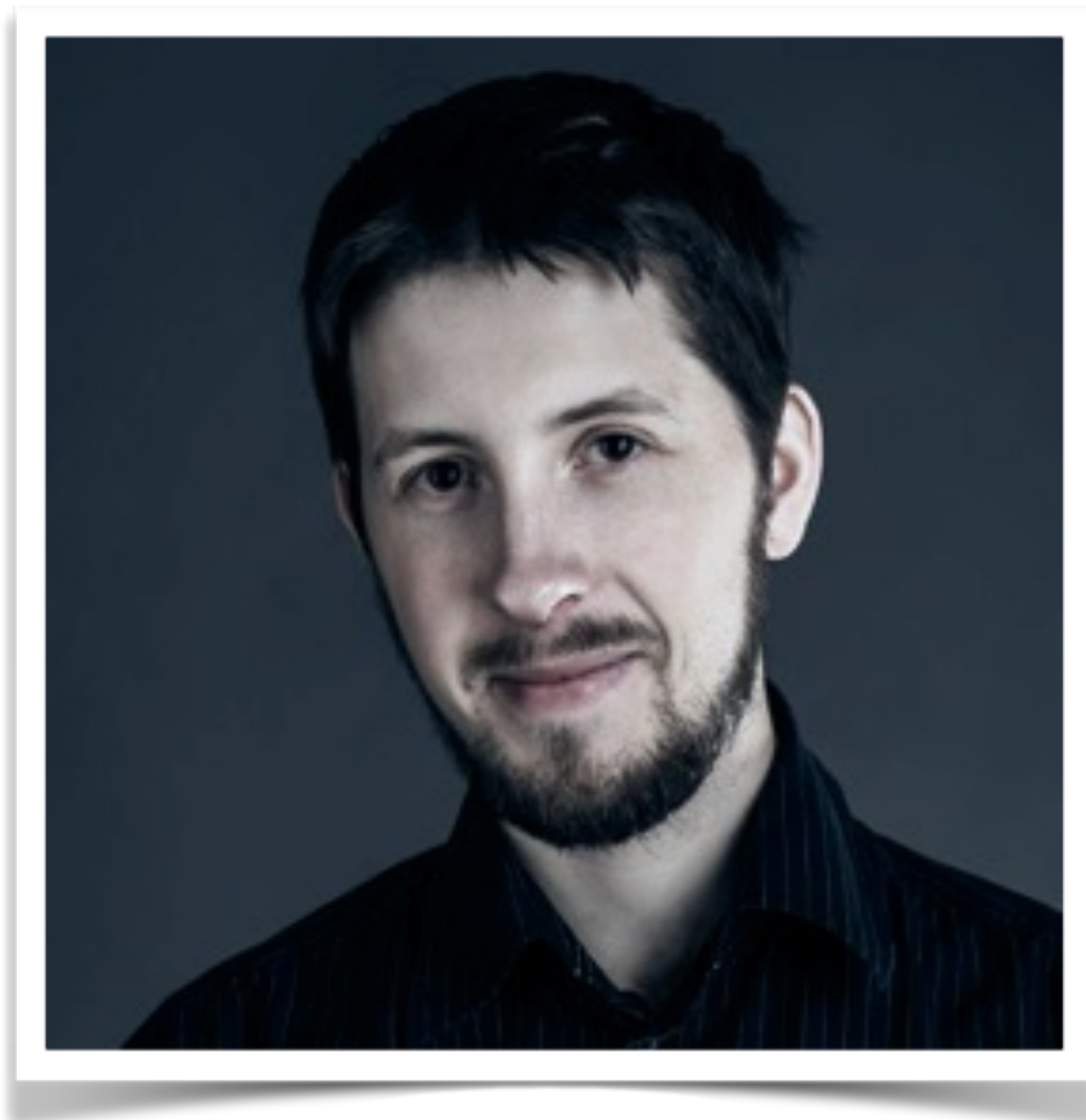
A year with JDK 11 in production

The logo for QCon London features the text "QCon" in a large, bold font, with the "Q" in green and "Con" in blue. Below it, the word "LONDON" is written in a smaller, black, sans-serif font. The background of the logo is a light gray silhouette of the London skyline, including the London Eye, Big Ben, and the Gherkin.

QCon
LONDON

ags @ QCON London 2020
2020-03-04T11:50Z

andrzej grzesik



Principal Engineer @ **Revolut**

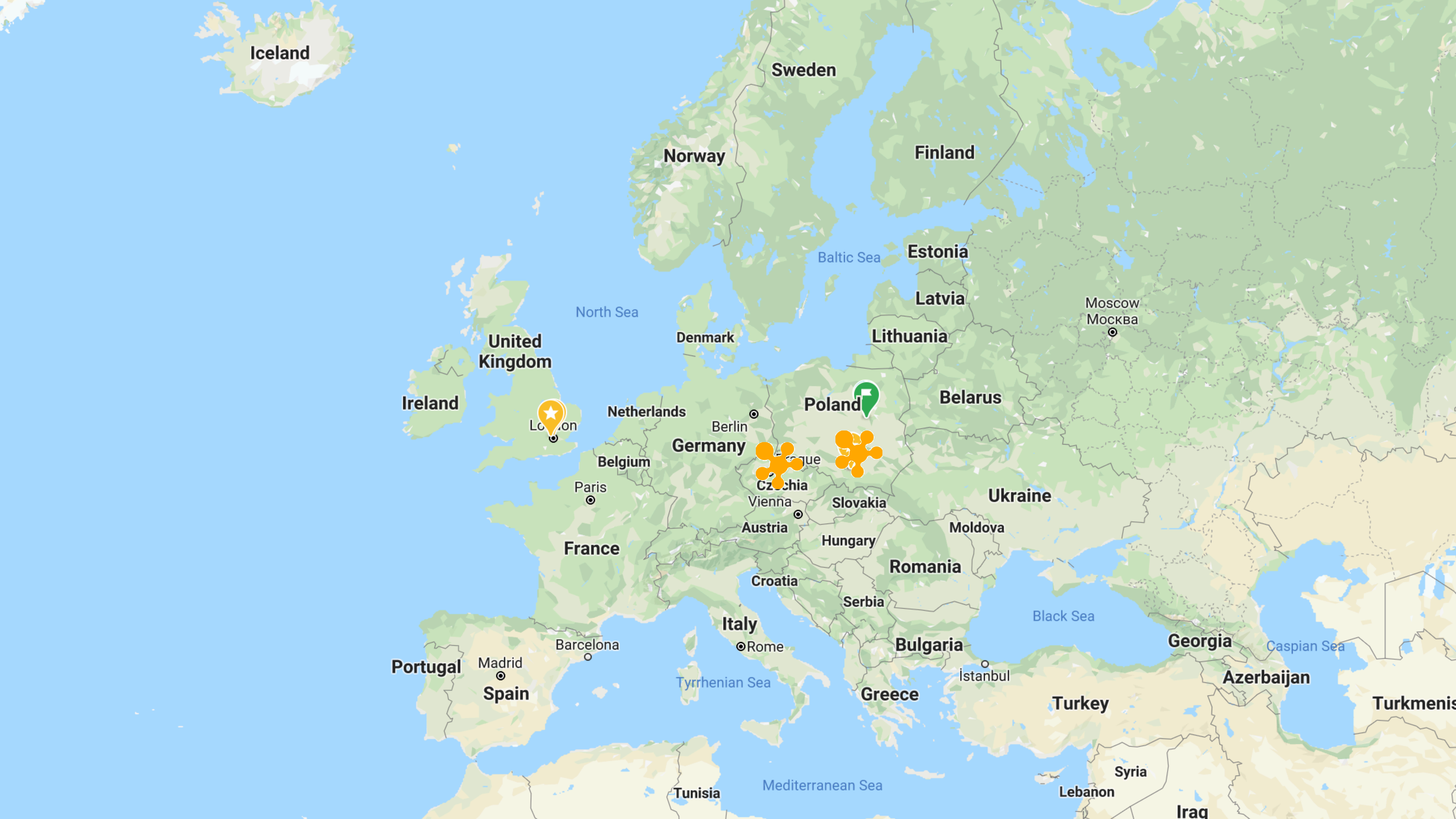
twitter://ags313

ags@revolut.com

andrzejgrzesik.info

about:me





Iceland

Sweden

Norway

Finland

Baltic Sea

Estonia

Latvia

Moscow
Москва

Lithuania

United
Kingdom

Denmark

Ireland

London

Netherlands

Germany

Berlin

Poland

Belgium

Paris

France

Czechia

Vienna

Austria

Slovakia

Hungary

Ukraine

Moldova

Romania

Croatia

Serbia

Italy

Rome

Barcelona

Portugal

Madrid

Spain

Tyrrhenian Sea

Tunisia

Mediterranean Sea

Greece

Bulgaria

Istanbul

Black Sea

Turkey

Syria

Lebanon

Georgia

Azerbaijan

Caspian Sea

Turkmenistan

Iraq

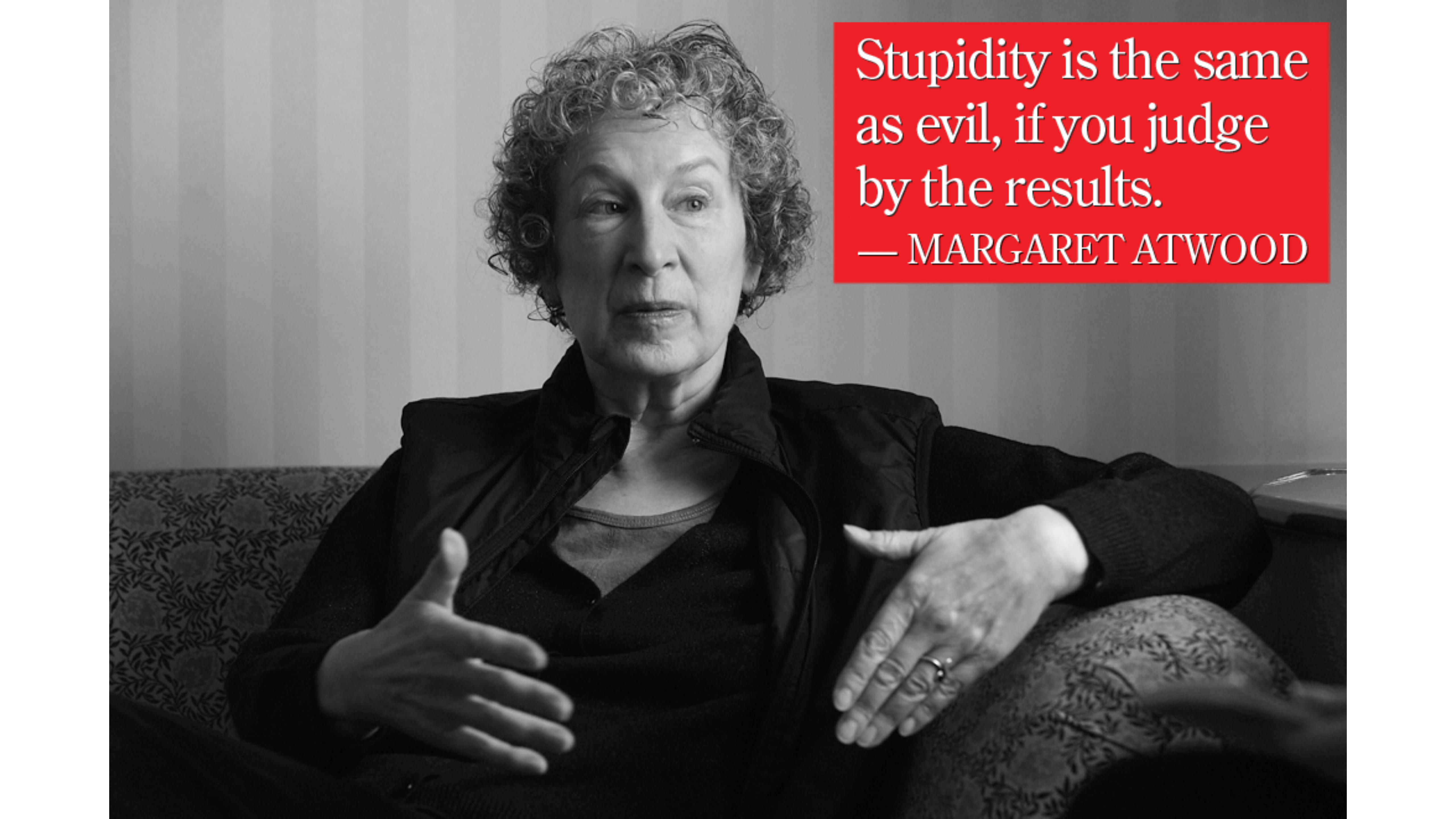
my opinions are my own

questions about the talk?

just ask!

questions about ***Revolut***?

talk to me after!

A black and white photograph of Margaret Atwood. She is seated on a patterned couch, wearing a dark jacket over a dark top. She has short, curly, light-colored hair and is looking directly at the camera with a serious expression. Her hands are resting on her lap, with fingers slightly spread. The background is a plain, light-colored wall.

Stupidity is the same
as evil, if you judge
by the results.

— MARGARET ATWOOD



Michel Schudel
@MichelSchudel



What java version are you using in production?



312 votes · Final results

6:09 PM · Feb 25, 2020 · [Twitter for Android](#)

Which Java are you on?

<Java 8



Which Java are you on?

<Java 8

Java 8



Which Java are you on?

<Java 8

Java 8

Java 9 or 10



Which Java are you on?

<Java 8

Java 8

Java 9 or 10

Java 11



Which Java are you on?

<Java 8

Java 8

Java 9 or 10

Java 11

>Java 11



Java upgrade






Java upgrade

in context

About Revolut

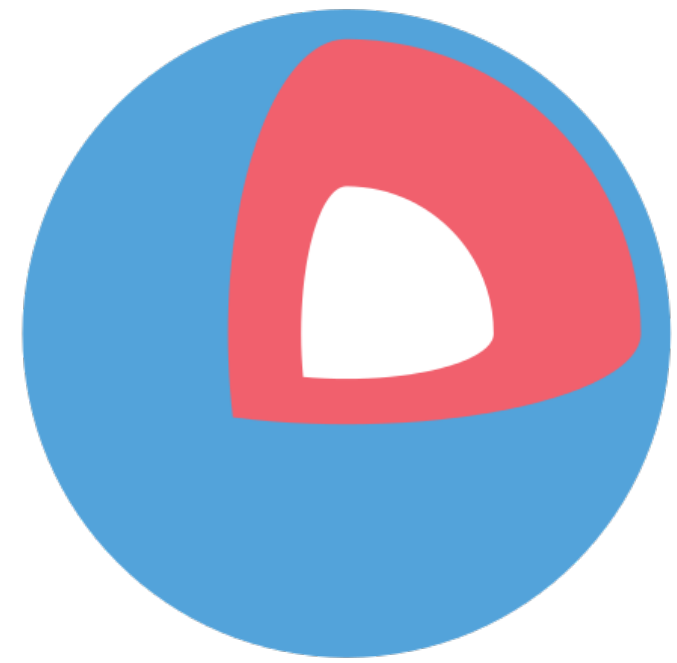
Check the actual number of retail customers here.
Check the actual number of business customers here.
Check the number of employees in the upper right corner.

R

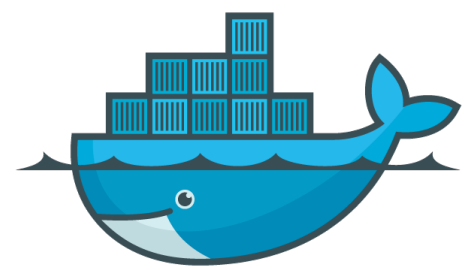
-  Launched in **July 2015**
-  **10M+** retail and **340K+** business users
-  **2200+** employees
-  **20+** offices around the world
-  Total investment to date: **\$836M**



Deployment



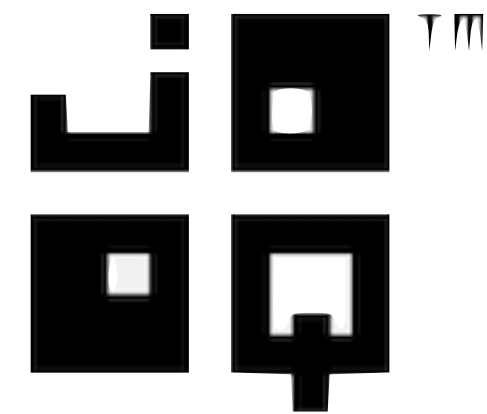
Core OS



docker



Spark



...



Why upgrade?



Why upgrade?

Get sh!T done



Why upgrade?

Get sh!T done

Move fast, keep things working



Why upgrade?

Get sh!T done

Move fast, keep things working

Would have to upgrade anyway



Why upgrade?

Get sh!T done

Move fast, keep things working

Would have to upgrade anyway

Want new features and fixes





java.lang.String

Up to Java 8

```
110
111 public final class String
112     implements java.io.Serializable, Comparable<String>, CharSequence {
113     /** The value is used for character storage. */
114     private final char value[];
115
116     /** Cache the hash code for the string */
117     private int hash; // Default to 0
```

From Java 9

```
125 public final class String
126     implements java.io.Serializable, Comparable<String>, CharSequence {
127
128     /** The value is used for character storage. ... */
139     @Stable
140     private final byte[] value;
141
142     /** The identifier of the encoding used to encode the bytes in ... */
153     private final byte coder;
```

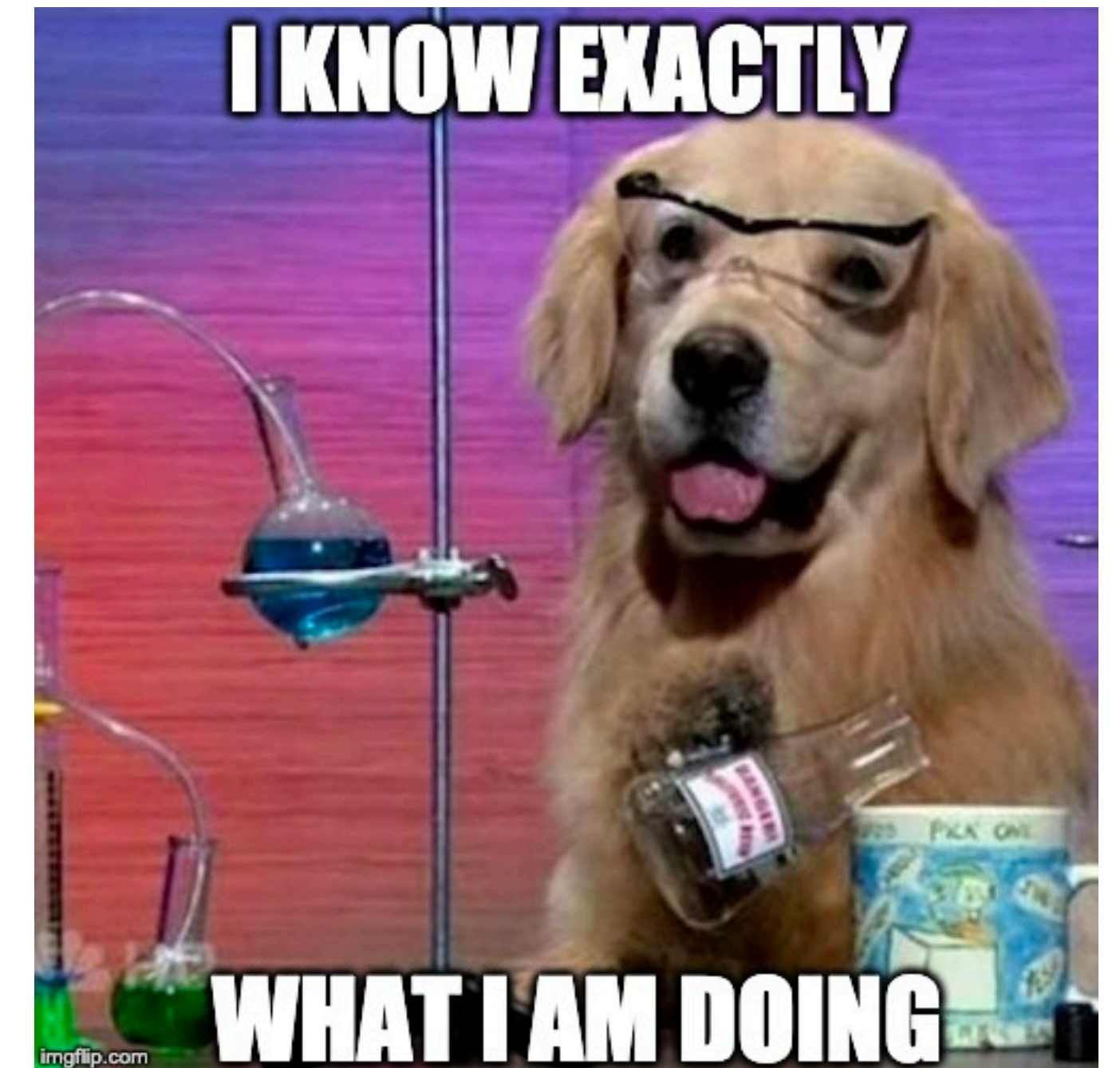



Upgrade and Ops

automationⁿ

Developers take care of many infrastructure tasks

DevOps work on specialised projects



Java 11 upgrade

from jdk8u212-b03 to 11

Java 11 upgrade

from jdk8u212-b03 to 11, all apps

Why not...

- 9 and 10 no longer maintained, no support (backports)
- 12 wasn't available

Why not Oracle JDK

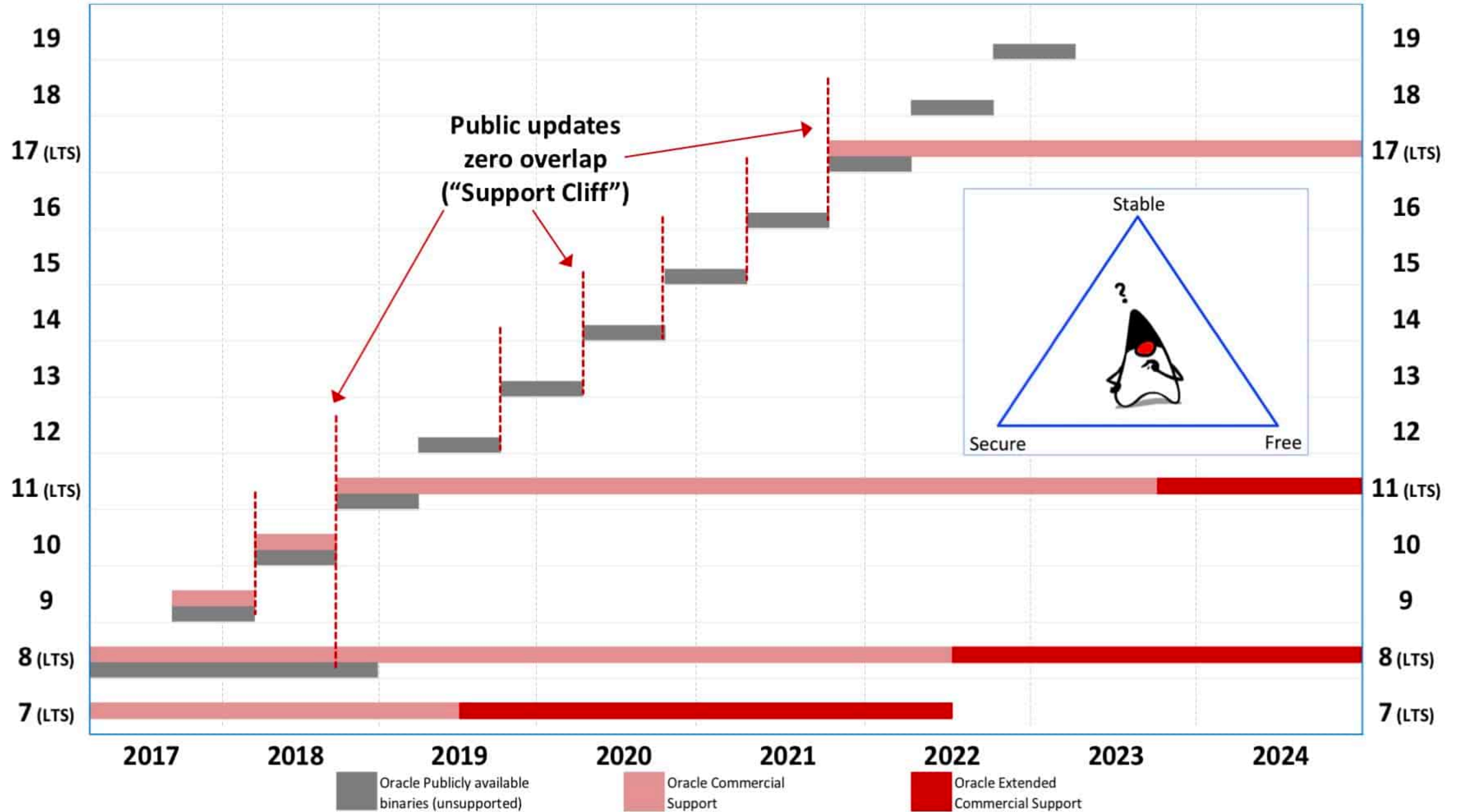
- Can't build our own images
- Can't redistribute (this definition is broad)
- Requires a license

Java - release cadence reminder

- New release every 6 months
- 3 builds for each major version
- LTS (11 + $n \times 6$) edition get fixes for a longer period

Java SE Lifecycle – 5+ Year Timeline

Java SE Version



Java 11

Compiles on 11

Runs on 11



Compiles on 8

Runs on 11

Java 11 everywhere

runtime AND toolchain

Using Java 11

And not full java 11 toolchain

~~I've got~~ I'm holding you back!

Cassandra <4.0

Grails <4.0.0

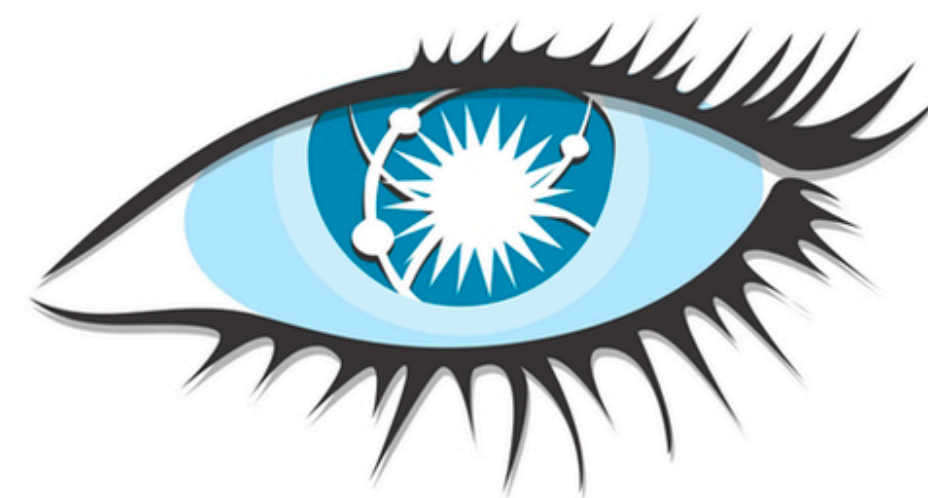
Hadoop <3.3.0

Kafka <2.1.0

Spring-boot <2.0

Thrift <0.13.0

...



<http://cassandra.apache.org/doc/latest/new/java11.html>

<https://github.com/apache/thrift/blob/master/LANGUAGES.md>

<https://issues.apache.org/jira/browse/KAFKA-7264>

<https://issues.apache.org/jira/browse/HADOOP-15338>

Build tools

jdeps

Shows module dependencies (including internal API usage)

```
jdeps --jdk-internals --classpath theWorld.jar app.jar
```

- Gradle:
<https://github.com/kordamp/jdeps-gradle-plugin>
- Maven: maven-jdeps-plugin
<https://maven.apache.org/plugins/maven-jdeps-plugin/index.html>

jdeprscan

Scans for usage of deprecated APIs

```
jdeprscan --release 11 --list --for-removal --classpath theWorld.jar app.jar
```

- Gradle: jdeprscan-gradle-plugin
<https://github.com/kordamp/jdeprscan-gradle-plugin>
- Maven: maven-jdeprscan-plugin
(<https://maven.apache.org/plugins/maven-jdeprscan-plugin/index.html>)

Maven

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
  <configuration>
    <source>${jdk.version}</source>
    <target>${jdk.version}</target>
    <compilerArgs>
      <arg>-Xlint:all</arg>
      <arg>-g:source,lines</arg>
    </compilerArgs>
  </configuration>
</plugin>
```

Maven

```
🍏 ~/vcs/oss/adoptopenjdk/jitwatch on 🐱 master  
> clear; mvn versions:display-plugin-updates  
[INFO] Scanning for projects...  
[INFO] -----  
[INFO] Reactor Build Order:  
[INFO]  
[INFO] JITWatch Parent [pom]  
[INFO] JITWatch Core [jar]  
[INFO] JITWatch UI [jar]
```

...

```
[INFO] Require Maven 2.0.1 to use the following plugin updates:  
[INFO]   maven-antrun-plugin ..... 1.8 -> 1.2  
[INFO]   maven-release-plugin ..... 2.5.3 -> 2.0-beta-7  
[INFO]  
[INFO] Require Maven 2.0.2 to use the following plugin updates:  
[INFO]   maven-javadoc-plugin ..... 2.10.4 -> 2.2  
[INFO]   maven-site-plugin ..... 3.6 -> 2.0-beta-7  
[INFO]  
[INFO] Require Maven 2.0.3 to use the following plugin updates:  
[INFO]   maven-dependency-plugin ..... 3.0.0 -> 2.2  
[INFO]  
[INFO] Require Maven 2.0.4 to use the following plugin updates:  
[INFO]   maven-antrun-plugin ..... 1.8 -> 1.3  
[INFO]   maven-assembly-plugin ..... 3.0.0 -> 2.2-beta-1  
[INFO]   org.codehaus.mojo:exec-maven-plugin ..... 1.6.0 -> 1.1  
[INFO]  
[INFO] Require Maven 2.0.5 to use the following plugin updates:
```


Maven

```
Apple ~/vcs/oss/adoptopenjdk/jitwatch on master
> mvn versions:display-dependency-updates
[INFO] Scanning for projects...
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] JITWatch Parent [pom]
[INFO] JITWatch Core [jar]
[INFO] JITWatch UI [jar]
```

...

```
[INFO] --- versions-maven-plugin:2.7:display-dependency-updates (default-cli) @ jit
[INFO] artifact junit:junit: checking for updates from central
[INFO] artifact org.slf4j:slf4j-api: checking for updates from central
[INFO] artifact ch.qos.logback:logback-core: checking for updates from central
[INFO] artifact ch.qos.logback:logback-classic: checking for updates from central
[INFO] The following dependencies in Dependency Management have newer versions:
[INFO]   ch.qos.logback:logback-classic ..... 1.2.3 -> 1.3.0-alpha5
[INFO]   ch.qos.logback:logback-core ..... 1.2.3 -> 1.3.0-alpha5
[INFO]   junit:junit ..... 4.12 -> 4.13
[INFO]   org.slf4j:slf4j-api ..... 1.7.25 -> 2.0.0-alpha1
[INFO]
[INFO]
```

Gradle

Gradle

v5 and v6 are good

Mockito

- Formally, supports JDK 11 since 2.20.1
- Releases with JDK 11 support: 2.19.1, 2.23.16, 2.27.5

jooq

- Supports JDK 11 since 3.12.0
- We're on 3.11.x



An **awesome** framework to manage migrations of your database

Combined with testcontainers, great for isolated and repeatable tests

<https://flywaydb.org/>

Flyway

- Supports JDK 11 since 5.2.0
- Encountered issues, moved back to 4.2.0

JEP 320

```
1  ▶ dependencies {
2      implementation "org.glassfish.jaxb:jaxb-runtime:${jaxbVersion}"
3      implementation "org.glassfish.jaxb:jaxb-xjc:${jaxbVersion}"
4      implementation "com.sun.xml.ws:jaxws-rt:${jaxwsVersion}"
5      ⚡ implementation "javax.activation:activation:1.1.1"
6  }
```

Languages



- Groovy? Yes, for Spock (<http://spockframework.org/>)
- Upgraded Groovy to 2.4.16



Kotlin

Supports JDK 9-12 since 1.3.30

Scala

We're using 2.12.7, just switched versions



Scala



JDK 11 compatibility notes

Although the table above jumps from 8 to 11, JDK 9 and 10 will probably also work wherever 11 does. But unlike 9 and 10, 11 is an LTS release, so 11 is what we actually test on and recommend.

As of Scala 2.13.0, 2.12.8 and 2.11.12, **JDK 11 support is incomplete**. Notably, `scalac` will not enforce the restrictions of the Java Platform Module System, which means that code that typechecks may incur linkage errors at runtime. Scala 2.13.x will eventually provide **rudimentary support** for this, but likely only in nightlies built on Java 11.

JDK 11 support requires minimum sbt version 1.1.0, or 0.13.17 in the 0.13.x series.

Testing with Postgres

yandex-qatools / postgresql-embedded

Used by 536

Watch 31

Star 462

Fork 88

Code

Issues 45

Pull requests 7

Actions

Projects 0

Wiki

Security

Insights

Branch: master

Commits on Sep 27, 2019

Use up to date process embed version to be compatible with Java 9+. (#...



marshallpierce authored and smecksia committed on 27 Sep 2019



7ed616e



Commits on Apr 5, 2019

Not actively maintained anymore note



smecksia committed on 5 Apr 2019

1

Verified



0895ad0



Fix #160, Upgrade PgSQL version (#161)



srbala authored and smecksia committed on 5 Apr 2019



c9f2f9b



Code

Issues 45

Pull requests 7

Actions

Projects 0

Wiki

Security

Insights

Branch: master

Commits on Sep 27, 2019

Use up to date process embed version to be compatible with Java 9+. (#...



marshallpierce authored and smecksia committed on 27 Sep 2019



7ed616e



Commits on Apr 5, 2019

Not actively maintained anymore note



smecksia committed on 5 Apr 2019

1

Verified



0895ad0



Fix #160, Upgrade PgSQL version (#161)



srbala authored and smecksia committed on 5 Apr 2019



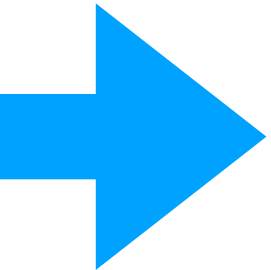
c9f2f9b



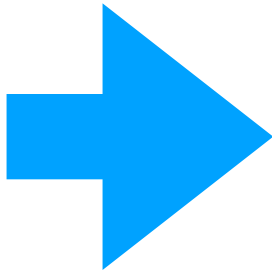
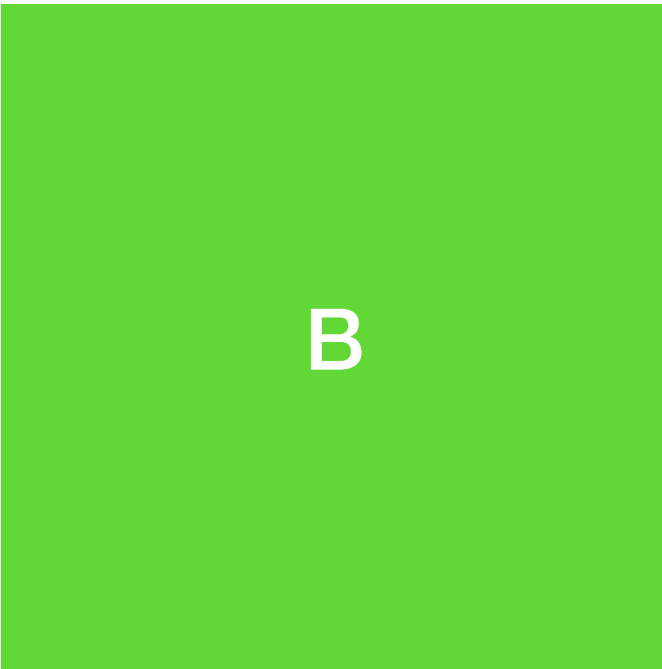
Solution



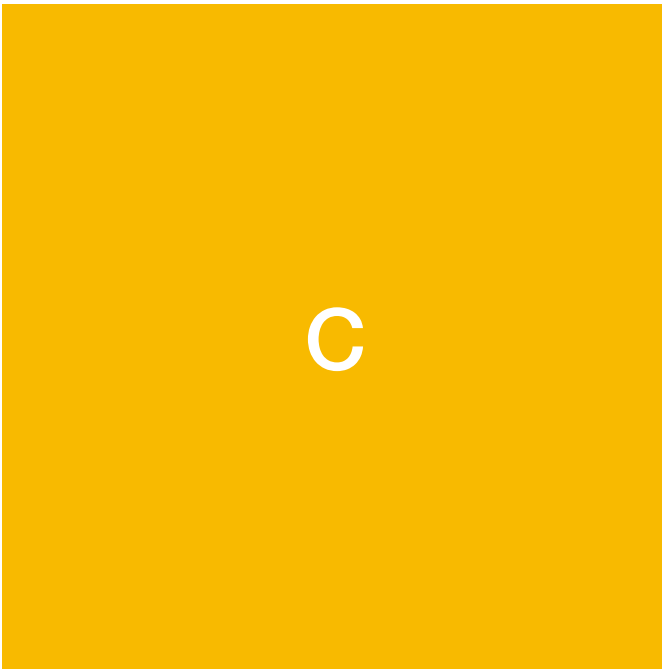
Dependencies



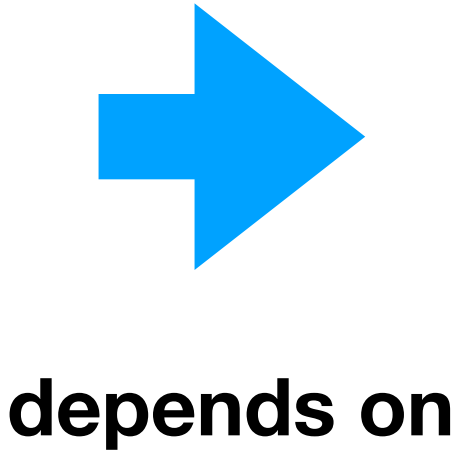
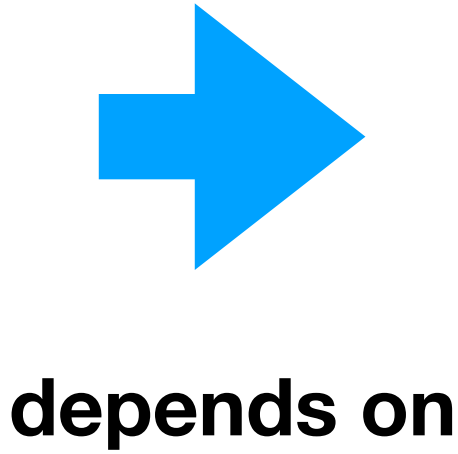
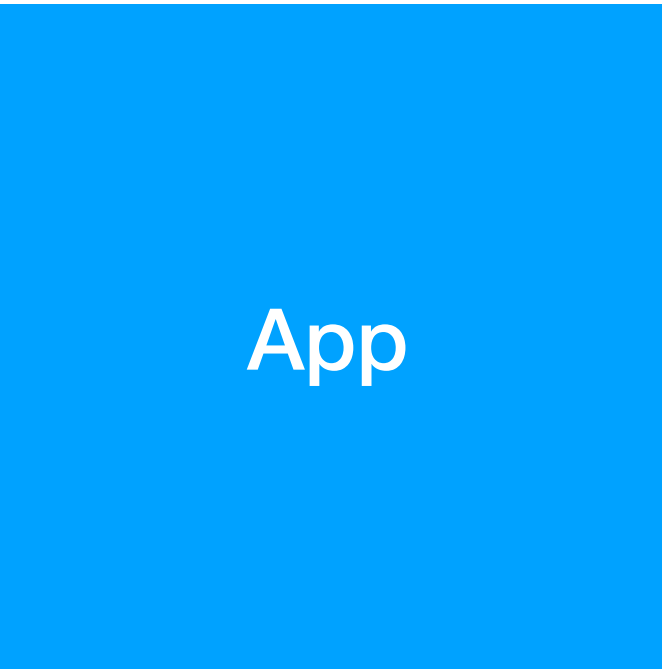
depends on



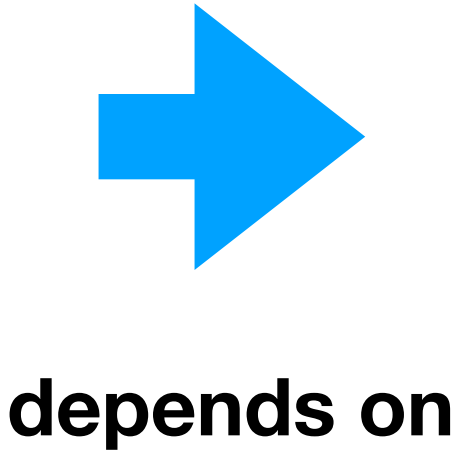
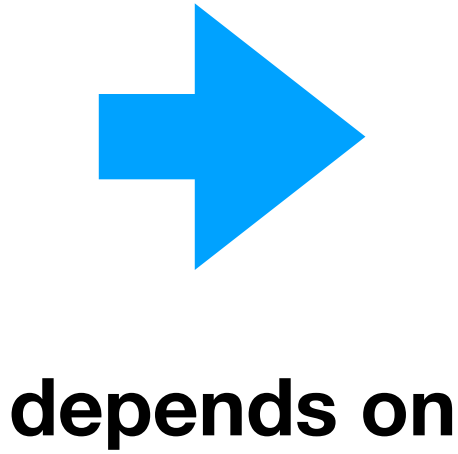
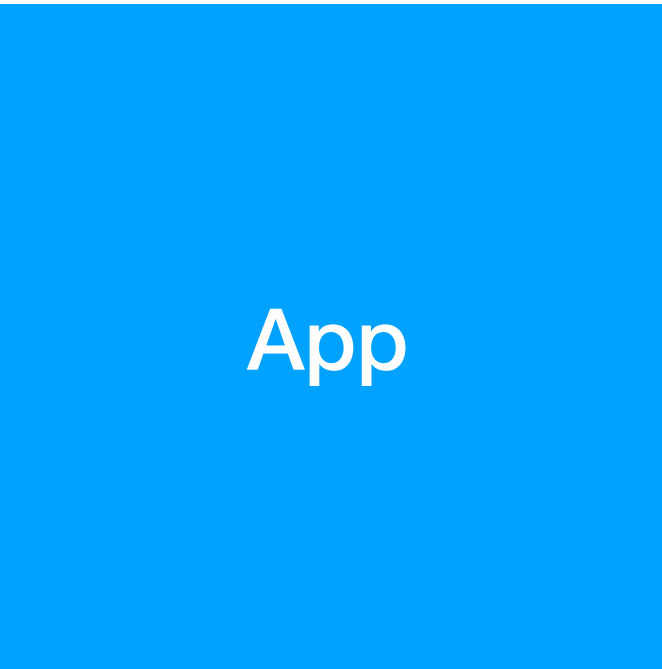
depends on



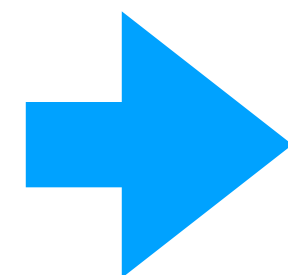
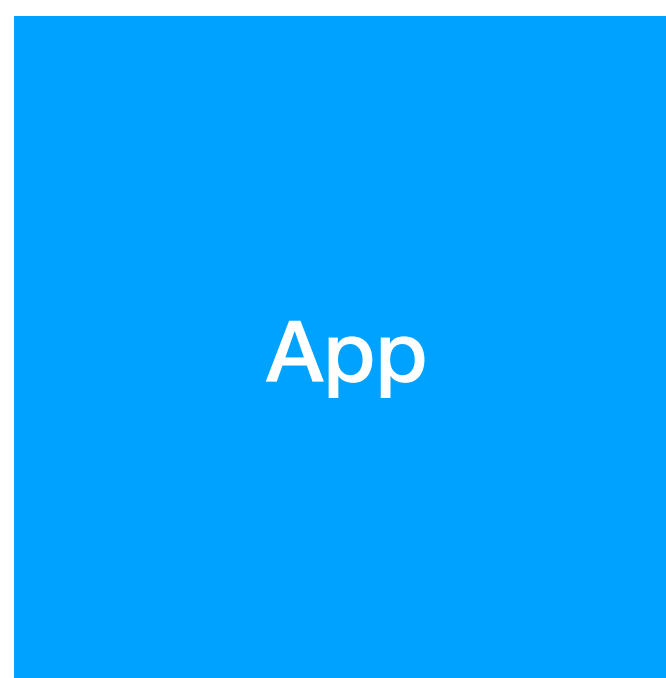




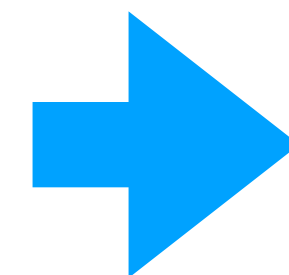
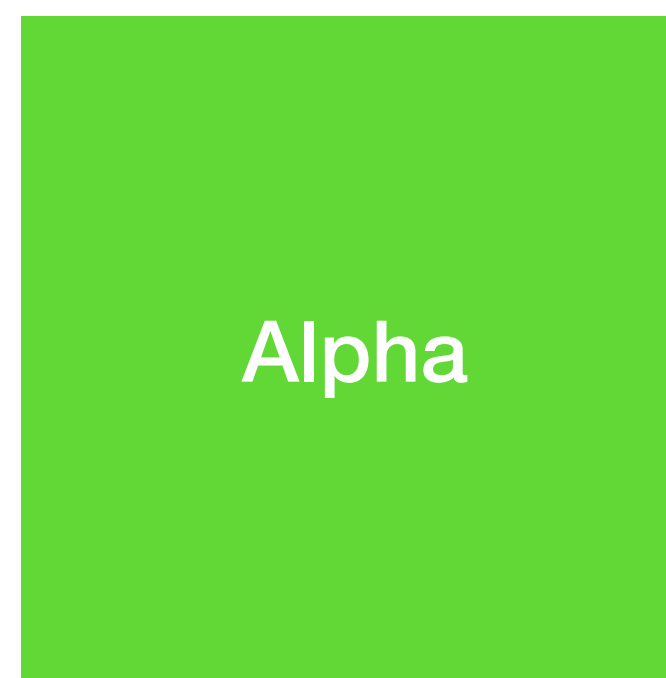
Compile using 11



Compile using 11
Test using 11



depends on



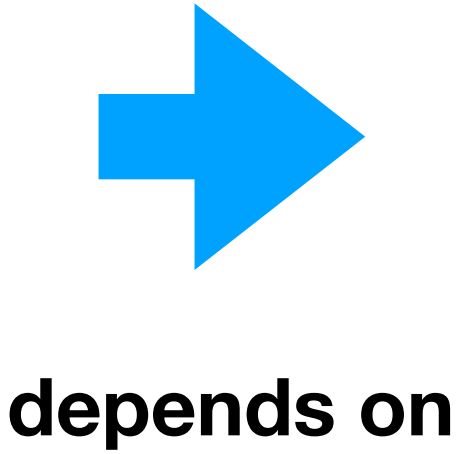
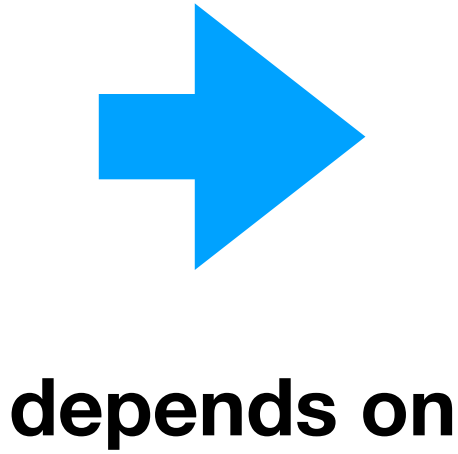
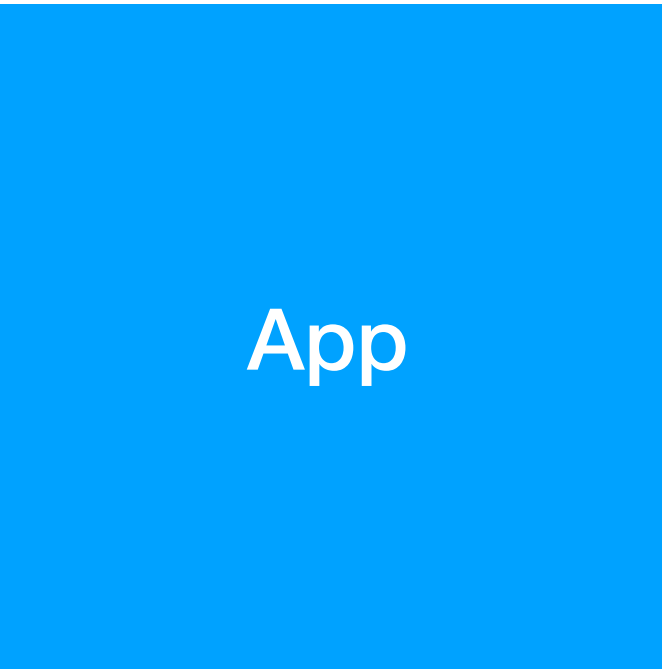
depends on



Compile using 11

Test using 11

Run using 11



Compile using 11
Test using 11
Run using 11

Compile using 11

BouncyCastle

- In JDK 8, it was common to install bouncycastle.jar into jre/lib/ext
- This does not exist in JDK 9+

```
echo \  
security.provider.10=org.bouncycastle.jce.provider.BouncyCastleProvider \  
>> ${JAVA_HOME}/lib/security/java.security
```

VM options

JEP 271: Unified GC Logging

<i>Author</i>	Jon Masamitsu
<i>Owner</i>	Bengt Rutisson
<i>Type</i>	Feature
<i>Scope</i>	JDK
<i>Status</i>	Closed / Delivered
<i>Release</i>	9
<i>Component</i>	hotspot / gc
<i>Discussion</i>	hotspot dash gc dash dev at openjdk dot java dot net
<i>Effort</i>	L
<i>Duration</i>	L
<i>Relates to</i>	JEP 158: Unified JVM Logging
<i>Reviewed by</i>	Mikael Vidstedt
<i>Endorsed by</i>	Mikael Vidstedt
<i>Created</i>	2014/10/06 21:19
<i>Updated</i>	2017/06/02 19:18
<i>Issue</i>	8059805

java

- verbose:gc
- Xloggc:my_very_own_gc-%t.log
- XX:+PrintGC
- XX:+PrintGCApplicationStoppedTime
- XX:+PrintGCCause
- XX:+PrintGCDateStamps
- XX:+PrintGCDetails
- XX:+PrintReferenceGC
- XX:+PrintTenuringDistribution

becomes

java

- Xlog:gc*,gc+ref=debug,gc+phases=debug,gc+age=debug,safepoint:file=gc.log
- jar jar.binks



[Sponsor me on GitHub!](#)

jaCoLine.dev

JaCoLine - Java Command Line Inspector

InspectStatisticsAPIAboutPrivacy

Describe your system

JDK	Operating System	CPU Architecture	Debug JVM?
JDK8	Linux	x86	<input type="checkbox"/>

Enter your Java command line

Clear inputShow example

```
java -Xloggc -Xms1234M -Xmx2345M
-Djavax.net.ssl.trustStore=/opt/alamakota/certs
-Dslowquery.threshold.millis=100
-Dfile.encoding=UTF-8
-Dlogback.configurationFile=/opt/alamakota/config/logback.xml
-jar /opt/alamakota/lib/jar.jar
app/config.yml
```

Inspect Command Line

Results

Switches Identified

-Xloggc: -Xms1234M -Xmx2345M

Switch Analysis

Name	Type	Description	Default Value	Your Value
loggc	<file>	log GC status to a file with time stamps		
Analysis	OK			

Name	Type	Description	Default Value	Your Value
ms	<size>	set initial Java heap size		1234M
Analysis	OK			



[Sponsor me on GitHub!](#)

jaCoLine.dev

JaCoLine - Java Command Line Inspector

InspectStatisticsAPIAboutPrivacy

Describe your system

JDK	Operating System	CPU Architecture	Debug JVM?
JDK11	Linux	x86	<input type="checkbox"/>

Enter your Java command line

Clear inputShow example

```
java -Xloggc -Xms1234M -Xmx2345M
      -Djavax.net.ssl.trustStore=/opt/alamakota/certs
      -Dslowquery.threshold.millis=100
      -Dfile.encoding=UTF-8
      -Dlogback.configurationFile=/opt/alamakota/config/logback.xml
      -jar /opt/alamakota/lib/jar.jar
      app/config.yml
```

Inspect Command Line

Results

Switches Identified

-Xloggc: -Xms1234M -Xmx2345M

Switch Analysis

Name	Type	Description	Default Value	Your Value
loggc				
Analysis	OK			

Name	Type	Description	Default Value	Your Value
ms	<size>	set initial Java heap size		1234M
Analysis	OK			

-XX changes

VM Options Explorer - JDK11 HotSpot

Sponsor me on GitHub!

HotSpot										Options added/removed	GraalVM 20.0.0				GraalVM native-image 20.0.0				OpenJ9		Azul Zing	
VM Options	JDK6	JDK7	JDK8	JDK9	JDK10	JDK11	JDK12	JDK13	JDK14	JDK15	JDK8		JDK11		JDK8		JDK11		OpenJ9	Zing JDK8	Zing JDK11	
Intrinsics	JDK6	JDK7	JDK8	JDK9	JDK10	JDK11	JDK12	JDK13	JDK14	JDK15	Community	Enterprise	Community	Enterprise	Community	Enterprise	Migrating from HotSpot					
												EE-only options		EE-only options		EE-only options		EE-only options				

Hey Reader! If these notes float your boat you'll find more great JVM info in this book I co-wrote! [Optimizing Java: Practical techniques for improving JVM application performance](#)

Search JDK11 HotSpot Options:

Name	Since	Deprecated	Type	OS	CPU	Component	Default	Availability	Description
<div><div></div><div>Show All</div><div>Deprecated</div><div>Show All</div><div>Show All</div><div>Show All</div><div>Show All</div><div>Show All</div></div>									
AbortVMOnException	JDK6		ccstr			runtime	NULL	diagnostic	Call fatal if this exception is thrown. Example: java -XX:AbortVMOnException=java.lang.NullPointerException Foo
AbortVMOnExceptionMessage	JDK6		ccstr			runtime	NULL	diagnostic	Call fatal if the exception pointed by AbortVMOnException has this message
ActiveProcessorCount	JDK10		int			gc	-1	product	Specify the CPU count the VM should use and report as active
AdaptiveSizeDecrementScaleFactor	JDK6		uintx			gc	4 range(1, max_uintx)	product	Adaptive size scale down factor for shrinking
AdaptiveSizeMajorGCDecayTimeScale	JDK6		uintx			gc	10 range(0, max_uintx)	product	Time scale over which major costs decay
AdaptiveSizePolicyCollectionCostMargin	JDK6		uintx			gc	50 range(0, 100)	product	If collection costs are within margin, reduce both by full delta
AdaptiveSizePolicyGCTimeLimitThreshold	JDK6		uintx			gc	5 range(1, max_uintx)	develop	Number of consecutive collections before gc time limit fires
AdaptiveSizePolicyInitializingSteps	JDK6		uintx			gc	20 range(0, max_uintx)	product	Number of steps where heuristics is used before data is used
AdaptiveSizePolicyOutputInterval	JDK6		uintx			gc	0 range(0, max_uintx)	product	Collection interval for printing information; zero means never
AdaptiveSizePolicyReadyThreshold	JDK6		uintx			gc	5	develop	Number of collections before the adaptive sizing is started
AdaptiveSizePolicyWeight	JDK6		uintx			gc	10	product	Weight given to exponential resizing, between 0 and 100

JDK 9+ `java.util.zip.ZipFile`

The new `java.util.zip.ZipFile` implementation does **NOT** use `mmap` to map ZIP file central directory into memory anymore.

–`Dsun.zip.disableMemoryMapping` property is gone.

Time precision



JDK / JDK-8068730

Increase the precision of the implementation of java.time.Clock.systemUTC()

▼ Details

Type:	⚡ Enhancement	Status:	CLOSED
Priority:	3 P3	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	9
Component/s:	core-libs		
Labels:	autoverify	jsr379-annex1-na	release-note=yes
Subcomponent:	java.time		
Resolved In Build:	b50		
Verification:	Verified		

▼ Backports

Issue	Fix Version	Assignee	Priority	Status	Resolution	Resolved In Build
JDK-8082849	emb-9	Daniel Fuchs	P3	Resolved	Fixed	team

▼ Description

The java.time.Clock.system() method (and variants thereof) are specified to "obtain a clock that returns the current instant using best available system clock". However the current implementation of the clock returned is based on System.currentTimeMillis() whereas the underlying native clock used by System.currentTimeMillis() has often a greater precision than milliseconds (for instance, on Linux, System.currentTimeMillis() is based on gettimeofday, which offers microseconds precision).

This RFE propose to enhance the implementation of the system clocks returned by java.time.Clock, so that they offer at least the same precision than the underlying clock available on the system.

Stephen Colebourne's blog

Thoughts and Musings on the world of Java and beyond

Tuesday, 7 February 2017

Java Time (JSR-310) enhancements in Java SE 9

The `java.time.*` API (JSR-310) was added to Java SE 8, but what has been going on since then?

Java Time in Java SE 9

There are currently [117 java time issues](#) targetted into Java SE 9. Most of these are not especially interesting, with a lot of mistakes in the Javadoc that needed fixing. What follows are some of the interesting ones:

Main enhancements:

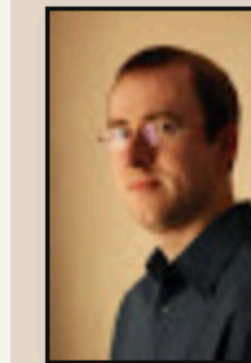
[JDK-8146218](#) - Add `LocalDate.datesUntil` method producing Stream.

Adds two new methods - `LocalDate.datesUntil(LocalDate)` and `LocalDate.datesUntil(LocalDate,Period)` - returning a stream of dates.

[JDK-8068730](#) - Increase precision of `Clock.systemUTC()`.

The clock in Java - `System.currentTimeMillis()` - has ticked in milliseconds since Java was first released. With Java SE 9, users of `Clock` will see higher precision, depending on the available clock of the operating system.

About Me



 **Stephen Colebourne**

London, United Kingdom

Java developer, blogger and conference speaker

[View my complete profile](#)

Blog Archive

► [2019](#) (3)

► [2018](#) (9)

▼ [2017](#) (5)

► [May](#) (1)

► [April](#) (3)

▼ [February](#) (1)

[Java Time \(JSR-310\) enhancements in Java SE 9](#)

► [2016](#) (3)

Stephen Colebourne's blog

Thoughts and Musings on the world of Java and beyond

Tuesday, 7 February 2017

Java Time (JSR-310) enhancements in Java SE 9

The `java.time.*` API (JSR-310) was added to Java SE 8, but what has been going on since then?

Java Time in Java SE 9

There are currently **117 java time issues** targetted into Java SE 9. Most of these are not especially interesting, with a lot of mistakes in the Javadoc that needed fixing. What follows are some of the interesting ones:

Main enhancements:

JDK-8146218 - Add `LocalDate.datesUntil` method producing Stream.

Adds two new methods - `LocalDate.datesUntil(LocalDate)` and `LocalDate.datesUntil(LocalDate, Period)` - returning a stream of dates.

JDK-8068730 - Increase precision of `Clock.systemUTC()`.

The clock in Java - `System.currentTimeMillis()` - has ticked in milliseconds since Java was first released. With Java SE 9, users of `Clock` will see higher precision, **depending on the available clock of the operating system.**

About Me



Stephen Colebourne

London, United Kingdom

Java developer, blogger and conference speaker

[View my complete profile](#)

Blog Archive

► 2019 (3)

► 2018 (9)

▼ 2017 (5)

► May (1)

► April (3)

▼ February (1)

[Java Time \(JSR-310\) enhancements in Java SE 9](#)

► 2016 (3)

Nanotrusting the Nanotime

<https://shipilev.net/blog/2014/nanotrusting-nanotime/>

Run: TimeEight x

▶

⬆

■

📷

⚙️

🔍

📄

⬆

⬇

↺

↻

🖨

/Library/Java/JavaVirtualMachines/jdk1.8.0_241.jdk/Contents/Home/bin/java ...

2020-02-26T13:38:42.426Z

Process finished with exit code 0

Run: TimeEleven x

▶

⬆

■

📷

⚙️

🔍

📄

⬆

⬇

↺

↻

🖨

/Library/Java/JavaVirtualMachines/jdk-11.0.6.jdk/Contents/Home/bin/java

Support/JetBrains/Toolbox/apps/IDEA-U/ch-1/193.6494.35/IntelliJ IDEA.ap

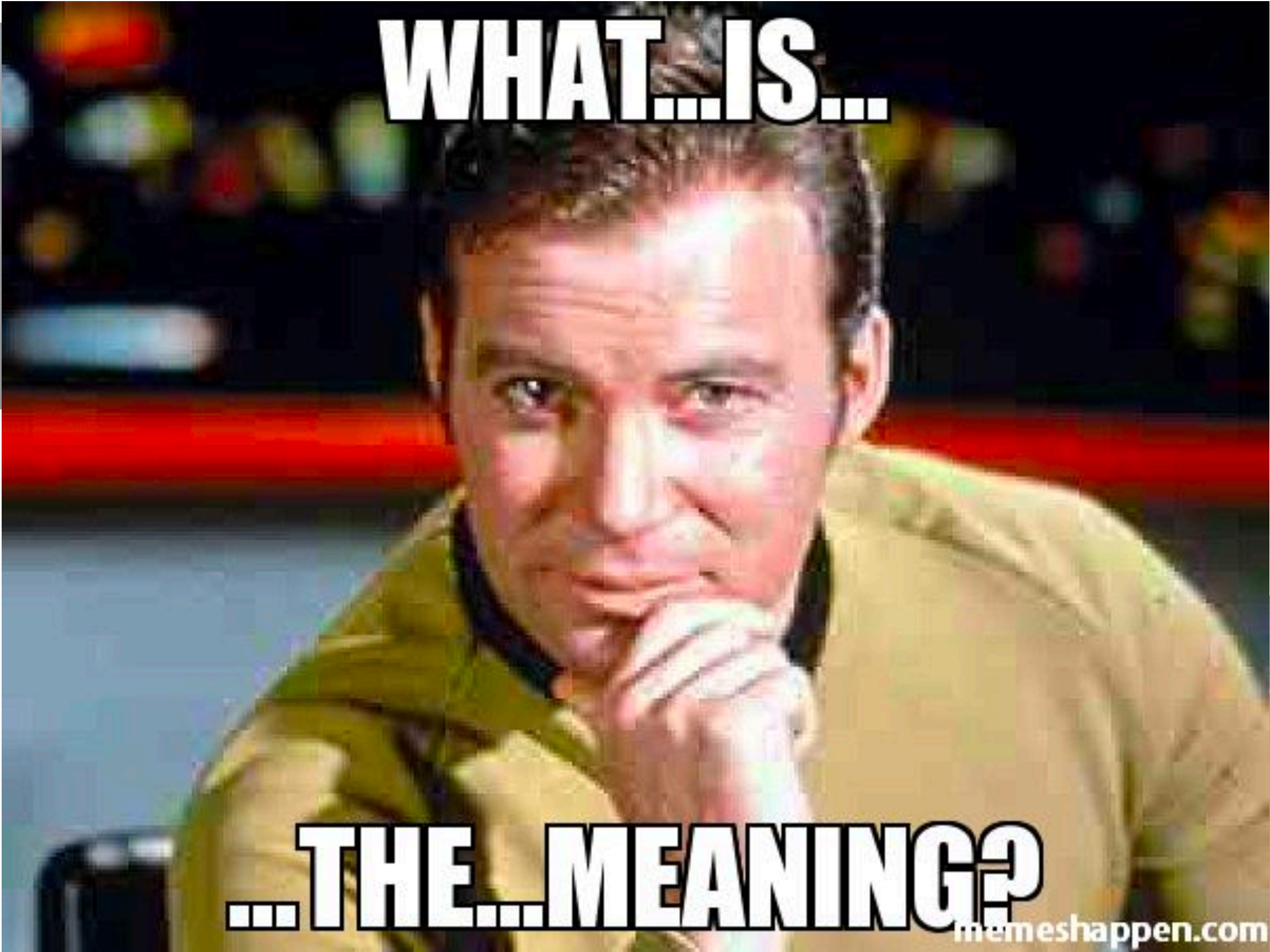
2020-02-26T13:37:35.616510Z

Process finished with exit code 0

|

Run: TimeEight x

```
/Library/Java/JavaVirtualMach  
2020-02-26T13:38:42.426Z  
Process finished with exit cod
```



```
Contents/Home/bin/java  
4.35/IntelliJ IDEA.ap
```



Recommendation

```
public final class TimeProvider11 {  
  
    private final Clock clock;  
  
    public static Instant instant() {  
        return Instant.now(TimeProviderHolder.INSTANCE.clock);  
    }  
  
    public static LocalDate date() {  
        return LocalDate.now(TimeProviderHolder.INSTANCE.clock);  
    }  
  
    private static Clock defaultClock() {  
        return Clock.systemUTC();  
    }  
  
    private static Clock millisClock() {  
        return Clock.tick(Clock.systemUTC(), Duration.ofMillis(1));  
    }  
}
```

Builder:

```
57 57    public static abstract class Builder<E extends AuditedEntity<E, ID>, ID extends Comparable<ID>, B extends Builder<E, ID, B>> {  
58 58  
59 59        protected ID id;  
60 60        protected Instant createdAt = now();  
60 60        protected Instant createdAt = instant();  
61 61        protected Instant updatedAt;  
62 62  
63 63        public B id(ID id) {
```

```
    private static class TimeProviderHolder {  
        private static final TimeProvider11 INSTANCE = new TimeProvider11();  
    }  
}
```

DateTimeFormatter bug



DateTimeFormatter with Locale.UK throw a NullPointerException when parsing zone

▼ Details


Type:	Bug	Status:	CLOSED
Priority:	P3	Resolution:	Fixed
Affects Version/s:	8u60	Fix Version/s:	9
Component/s:	core-libs		
Labels:	<div>7-na8u66-defer-requestPSU15_04-defer-approvedautoverifydcs-psoregressionregression_8078284webbug</div>		
Subcomponent:	java.time		
Introduced In Version:	8u60		
Resolved In Build:	b88		
CPU:	x86_64		
OS:	windows_7		
Verification:	Verified		



JDK / JDK-8139107

DateTimeFormatter with Locale.UK throw a NullPointerException when parsing zone

▼ Details

Type:  Bug

Status: **CLOSED**

Priority:  P3

Resolution: Fixed

Affects Version/s: 8u60

Fix Version/s: 9

Component/s: [core-libs](#)

Labels: [7-na](#) [8u66-defer-request](#) [PSU15_04-defer-approved](#) [autoverify](#) [dcs-pso](#) [regression](#) [regression_8078284](#) [webbug](#)

Subcomponent: [java.time](#)

Introduced In Version: [8u60](#)

Resolved In Build: b88

CPU: x86_64

OS: windows_7

Verification: Verified

```
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy HH:mm:ss");  
|  
DateTimeFormatter broken  = formatter.withLocale(Locale.UK);  
DateTimeFormatter working = formatter.withLocale(Locale.ENGLISH);
```

Compiler bugs



JDK / JDK-8222754

Javac compiler message file broken: key=compiler.misc.msg.bug arguments

▼ Details

Type:  Bug

Status:

CLOSED

Priority:  P3

Resolution:

Duplicate

Affects Version/s: 11.0.3

Fix Version/s:

None

Component/s: [tools](#)

Labels:

[dcsfai](#)

[regression](#)

[reproducer-no](#)

[webbug](#)

Subcomponent: [javac](#)

CPU: x86_64

OS: linux

▼ Description

ADDITIONAL SYSTEM INFORMATION :

uname -a: Linux bud.local 5.0.7-100.fc28.x86_64 #1 SMP Mon Apr 8 16:46:50 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
opt/jdk/bin/java -version: openjdk version "11.0.3" 2019-04-16 LTS

A DESCRIPTION OF THE PROBLEM :

On one of our code modules we have the javac tool fail with the following stack trace:

information:java: compiler message file broken: key=compiler.misc.msg.bug arguments=11.0.3, {1}, {2}, {3}, {4}, {5}, {6}, {7}

Information:java: at jdk.compiler/com.sun.tools.javac.util.Assert.check(Assert.java:46)

Information:java: at jdk.compiler/com.sun.tools.javac.comp.DeferredAttr\$2\$1.setOverloadKind(DeferredAttr.java:172)

Information:java: at jdk.compiler/com.sun.tools.javac.comp.ArgumentAttr.visitReference(ArgumentAttr.java:283)

Information:java: at jdk.compiler/com.sun.tools.javac.tree.JCTree\$JCMemberReference.accept(JCTree.java:2190)



JDK / JDK-8222754

Javac compiler message file broken: key=compiler.misc.msg.bug arguments

Turns out to be a duplicate of:



JDK / JDK-8210483

AssertionError in DeferredAttr at setOverloadKind caused by JDK-8203679

Details

Type:	Bug	Status:	CLOSED
Priority:	P2	Resolution:	Fixed
Affects Version/s:	11, 12	Fix Version/s:	12
Component/s:	tools		
Labels:	<div>10-na11.0.2.0.1-bpr11bpr-critical-approved11ea8-nadcsfaijdk11u-fix-requestjdk11u-fix-yeswebbug</div>		
Subcomponent:	javac		
Introduced In Build:	b18		
Introduced In Version:	11		
Resolved In Build:	b12		
CPU:	x86_64		
OS:	generic		
Verification:	Verified		

Backports

Issue	Fix Version	Assignee	Priority	Status	Resolution	Resolved In Build
JDK-8218946	11.0.4-oracle	Ivan Gerasimov	P2	Resolved	Fixed	b02
JDK-8218943	11.0.4	Liam Miller-Cushon	P2	Resolved	Fixed	b01



JDK / [JDK-8203195](#)

Anonymous class type inference results in NPE

▼ **Details**

Type:	 Bug
Priority:	 P2
Affects Version/s:	9, 10, 10.0.1, 11
Component/s:	tools
Labels:	dcsfai reproducer=yes webbug
Subcomponent:	javac
CPU:	generic
OS:	generic

Status:	OPEN
Resolution:	Unresolved
Fix Version/s:	tbd



JDK / JDK-8203195

Anonymous class type inference results in NPE

▼ Details

Type:  Bug

Priority:  P2

Affects Version/s: 9, 10, 10.0.1, 11

Component/s: [tools](#)

Labels: [dcsfai](#) [reproducer-yes](#) [webbug](#)

Subcomponent: [javac](#)

CPU: generic

OS: generic

Status: [OPEN](#)

Resolution: Unresolved

Fix Version/s: [tbd](#)

```
A() {  
    Object baz = foo.foo(new B<>() {});  
}
```



JDK / [JDK-8206142](#)

type inference: javac is incorrectly applying capture conversion during incorporation

▼ Details

Type:	Bug	Status:	IN PROGRESS
Priority:	P3	Resolution:	Unresolved
Affects Version/s:	9, 10.0.1, 11	Fix Version/s:	tbd_major
Component/s:	tools		
Labels:	dcsfai regression reproducer-yes webbug		
Subcomponent:	javac		
Introduced In Build:	b13		
Introduced In Version:	9		
CPU:	generic		
OS:	generic		



JDK / [JDK-8206142](#)

type inference: javac is incorrectly applying capture conversion during incorporation

▼ Details

Type:	Bug	Status:	IN PROGRESS
Priority:	P3	Resolution:	Unresolved
Affects Version/s:	9, 10.0.1, 11	Fix Version/s:	tbd_major
Component/s:	tools		
Labels:	dcsfai regression reproducer-yes webbug		
Subcomponent:	javac		
Introduced In Build:	b13		
Introduced In Version:	9		
CPU:	generic		
OS:	generic		

133	-	final Profile<? extends Profile, ?> user = retrieveUserProfile(userId);
133	+	final Profile<?, ?> user = retrieveUserProfile(userId);

Life with G1 as default

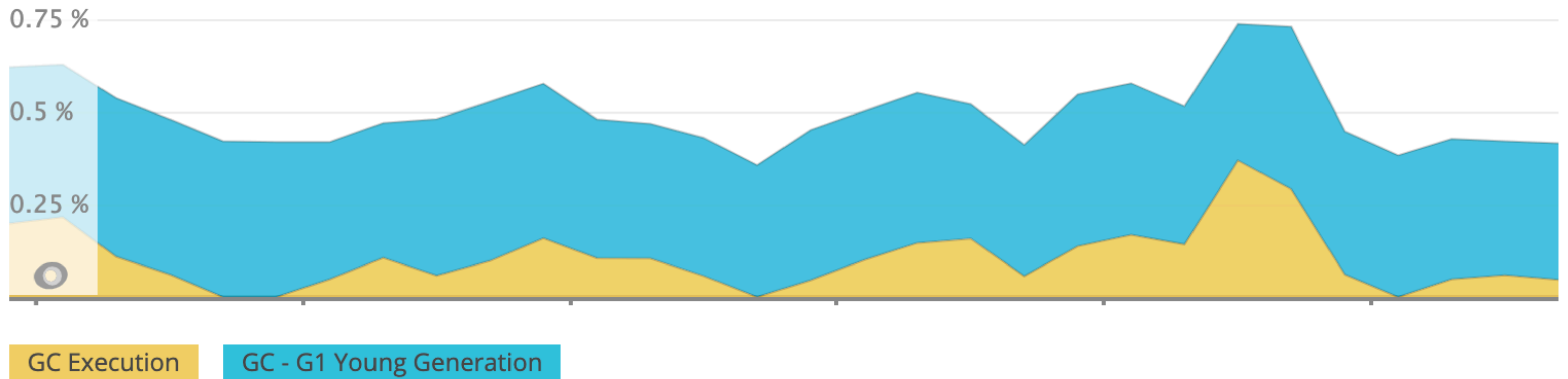
G1@JDK11 is different than G1@JDK8!

Life with G1 as default

G1: JDK 11 is different than JDK8!

Life with G1 by default

Garbage collection CPU time



Life with G1 by default

- In JDK 11, full GCs are multi-threaded (since JDK 10, JEP-307)
- No *magic* required to run human-perception pauseless
- Kept increasing heap sizes, and observed no issues

Life with G1 by default

- Less problems requiring gc logs
- Faster OOM failures

```
2019-09-28 21:15:44,547 [ERROR] [Executor task launch worker for task 48]
org.apache.spark.util.SparkUncaughtExceptionHandler – Uncaught exception in thread Thread[Executor task launch worker
for task 48,5,main] java.lang.OutOfMemoryError: GC overhead limit exceeded
    at sun.reflect.GeneratedConstructorAccessor72.newInstance(Unknown Source)
    at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45)
    at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
    at java.lang.Class.newInstance(Class.java:442)
```

```
scala git:(develop) x java -XX:+PrintFlagsFinal | grep GCTimeLimit
uintx GCTimeLimit = 98 {product}
```


Memory appetite

After moving to JDK 11, some containers died on us.

Changes to containers

Let's “live deploy” to Oracle Cloud and check!

```
[opc@instance-20200228-1824 ~]$ ps uxwwf 23416
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
opc       23130  0.0  0.0 163092  4492 ?        S    05:44   0:00 sshd: opc@pts/1
opc       23131  0.0  0.0 116112  4108 pts/1    Ss   05:44   0:00  \_ -bash
opc       23477  0.0  0.0 155380  3980 pts/1    R+   05:46   0:00      \_ ps uxwwf 23416
opc       22890  0.0  0.0 163092  4328 ?        S    05:41   0:00 sshd: opc@pts/0
opc       22891  0.0  0.0 116232  4356 pts/0    Ss   05:41   0:00  \_ -bash
opc       23416 88.5 54.5 13775576 8811456 pts/0    Sl+  05:45   0:14      \_ /usr/java/jdk-11.0.5/bin/java -Xms10G -Xmx10G -XX:-UseAdaptiveSizePolicy MemoryDemo11 4
```

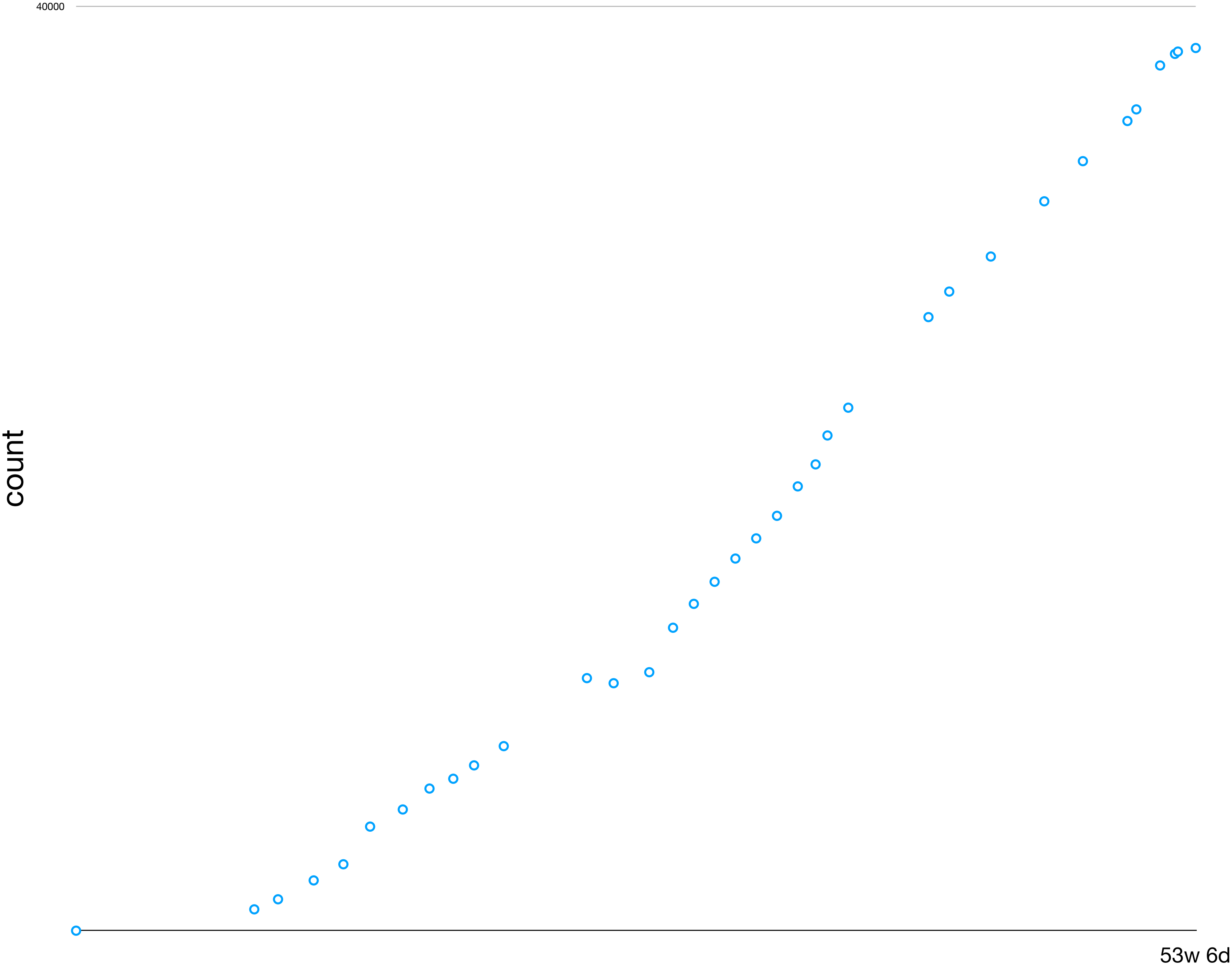
```
[opc@instance-20200228-1824 ~]$ ps uxwwf 25020
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
opc       25049  0.0  0.0 163092  4444 ?        S    06:07   0:00 sshd: opc@pts/1
opc       25050  0.2  0.0 116112  4032 pts/1    Ss   06:07   0:00  \_ -bash
opc       25117  0.0  0.0 155380  4008 pts/1    R+   06:07   0:00      \_ ps uxwwf 25020
opc       22890  0.0  0.0 163092  4328 ?        S    05:41   0:00 sshd: opc@pts/0
opc       22891  0.0  0.0 116232  4356 pts/0    Ss   05:41   0:00  \_ -bash
opc       25020 105 32.5 13270860 5248868 pts/0    Sl+  06:07   0:12      \_ /usr/java/jdk1.8.0_231-amd64/bin/java -Xms10G -Xmx10G -XX:-UseAdaptiveSizePolicy MemoryDemo8 4
```

Our finding

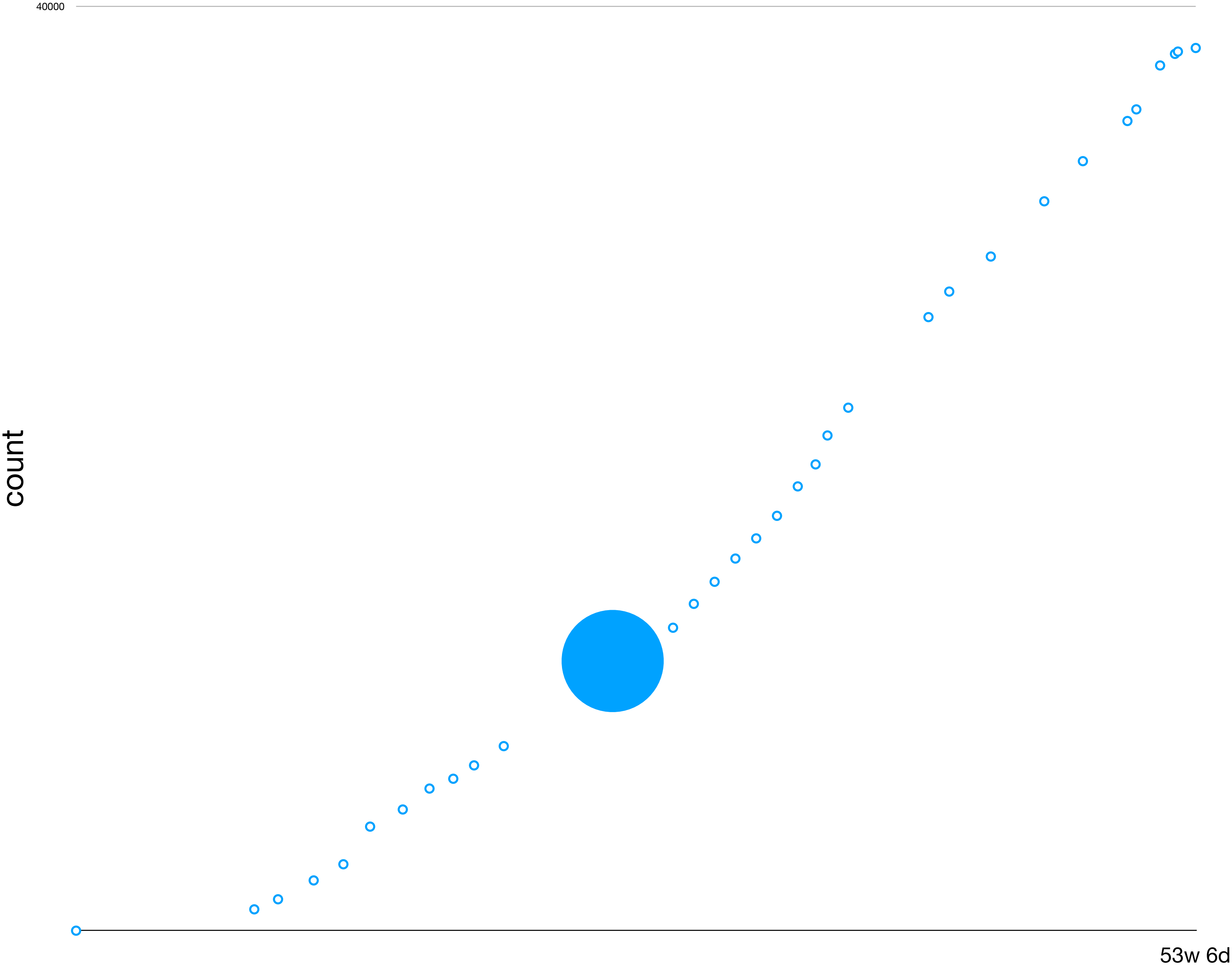
- RSS of a Java 11 process would be higher than of Java 8.
- For a fixed container size, we needed to decrease -Xmx and -Xms
- Faster OOMs

Language features adoption

adoption of var, project S



adoption of var, project S




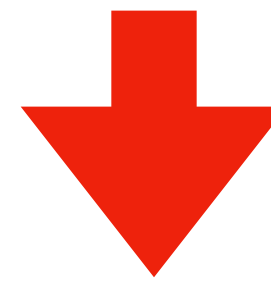
AS OF RELEASE 9, ' _ ' IS A KEYWORD,




AND MAY NOT BE USED AS AN IDENTIFIER

—

▼  /Users/andrzej.grzesik/vcs/talks/java11upgrade/java8/src/main/java/java8/Keyword.java
⚠ Warning:(6, 9) java: '_' used as an identifier
(use of '_' as an identifier might not be supported in releases after Java SE 8)



▼  /Users/andrzej.grzesik/vcs/talks/java11upgrade/java11/src/main/java/java11/Keyword.java
❗ Error:(6, 9) java: as of release 9, '_' is a keyword, and may not be used as an identifier


```
return some(business)
  .map(b → b.updateNatureOfBusinessInfoIfDifferent(nob))
  .filter(not(business::equals))
  .map(batch::update)
  .map(__ → ok(batch))
  .orElseGet(Result::empty);
```


Java Flight Record && Mission Control

- Free && open source
<https://github.com/openjdk/jmc>



What is next?

q: what is next?

Java 13 has **a lot** of improvements in G1 space...

... and no support anymore

q: what is next?

Java 14 has features amazing for observability
(JEP-349, JFR event streaming, <https://openjdk.java.net/jeps/349>)

However...

Gradle does not support Java 14 yet (<https://docs.gradle.org/current/userguide/compatibility.html>, <https://github.com/gradle/gradle/issues/10248>)

q: what is next?

We plan to try **GraalVM**TM, especially with Native Image.

How should you go about it?

Thanks!



Slides?
ags@revolut.com

Question time

Slides? Email ags@revolut.com

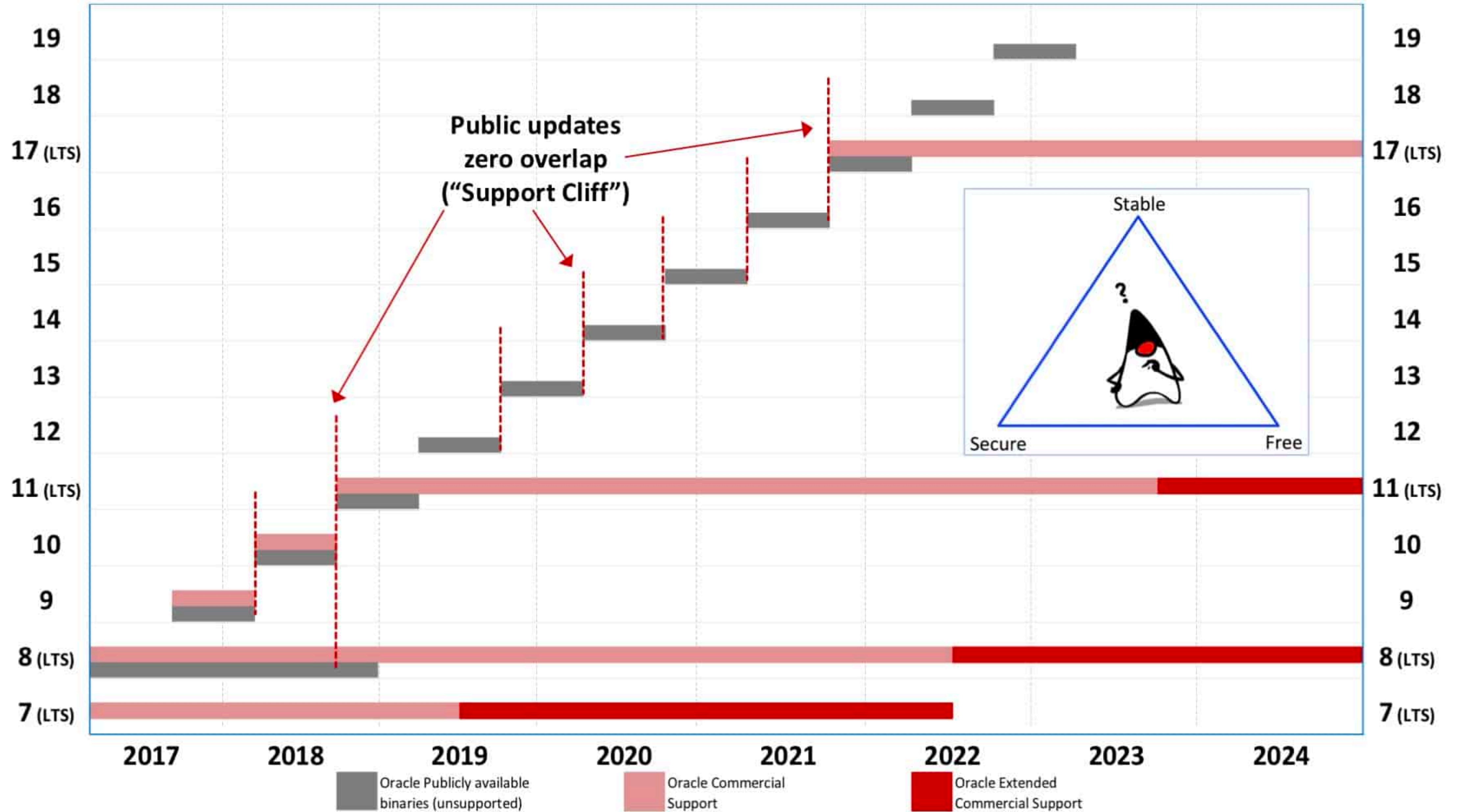
q: THE_PERFORMANCE

Java - release cadence reminder

- New release every 6 months
- 3 builds for each major version
- LTS (11 + $n \times 6$) edition get fixes for a longer period

Java SE Lifecycle – 5+ Year Timeline

Java SE Version



q: application CDS?

not using

q: what about modules?

In short, not yet.

We are on a monolith -> modules journey

q: benefits from byte[] Strings?

Yes, immediately taken over by increased traffic

q: which JDK

We have adopted OpenJDK

We' hiring!

(and help with relocation - check out revolut.com/careers)

Resources

Hand icon: <http://icon-library.com/icon/raised-hand-icon-21.html>

Margaret Atwood quote: <https://www.facebook.com/theweekuk/photos/a.282293375144981/3574042412636711/?type=3&theater>

Get IT done: <https://community.fintechtalk.co.uk/t/whod-work-in-a-place-like-this/226>

Java release graph: <https://dzone.com/articles/thinking-of-moving-to-jdk-9-next-year-think-again>

Java 9 graphic: https://jcp.org/en/press/news/JCP_JavaOne2017

I want you: <https://imgflip.com/memegenerator/23648483/I-WANT-YOU>

Hold my beverage: <https://imgflip.com/memegenerator/96314734/Hold-my-beer>

Turn picture: <https://www.dailymail.co.uk/femail/article-7227619/Drivers-left-baffled-road-marking-says-turn-LEFT-RIGHT.html>

Dog meme: <https://imgflip.com/memegenerator/I-Have-No-Idea-What-I-Am-Doing-Dog>

Star Trek meme: <https://memeshappen.com/meme/capt-kirk/what-is-the-meaning-25449/3>