SPRING IO

#springio24

Efficient Containers with Spring Boot 3, Java 21 and CDS

Sébastien Deleuze

https://seb.deleuze.fr

MAY · 30-31 · 2024

SPRINGIO.NET

Why do we care about runtime efficiency?

Cheaper hosting

Sustainability

Optimized containers

Different kinds of optimizations

Startup

Fast startup even on cheap servers

Warm up

Peak performance available asap

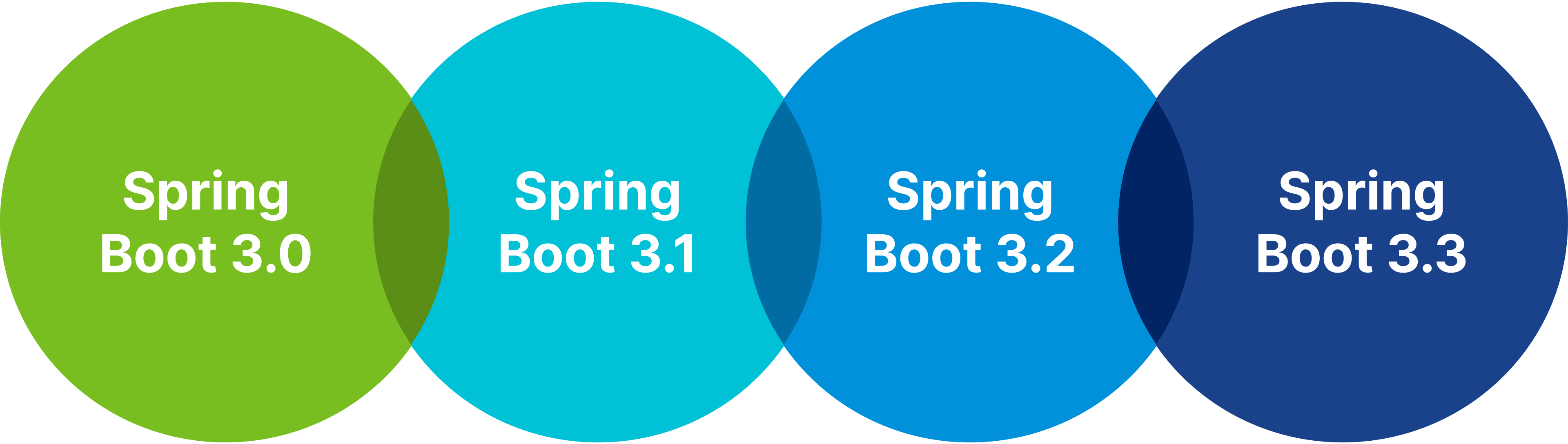
Scalability

Same throughput with less resources

Memory

Reduced memory consumption

Spring Boot 3 main features



Spring Boot 3.0

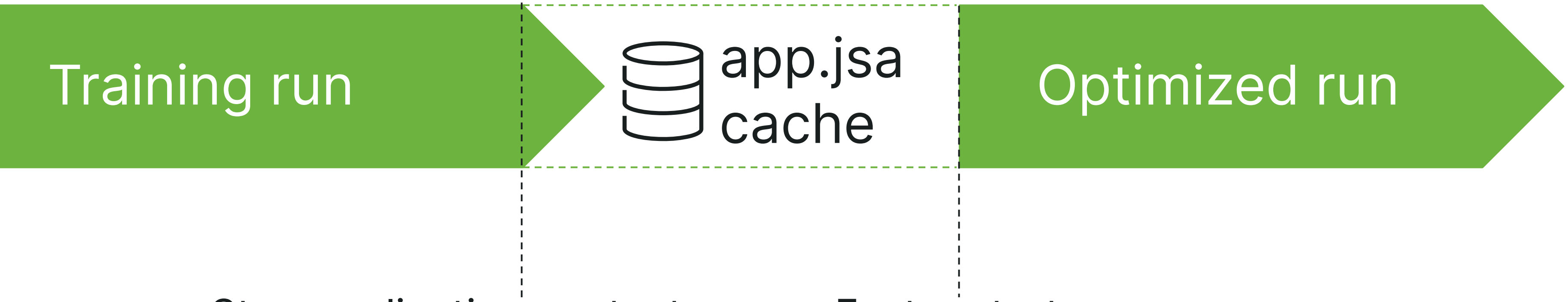
Spring Boot 3.1

Spring Boot 3.2

Spring Boot 3.3

GraalVM Spring AOT	Testcontainers Docker Compose	Virtual Threads Project CRaC	CDS New doc

Class Data Sharing (CDS)



Training run

app.jsa
cache

Optimized run

Stop application context
before starting
Spring lifecycle

Faster startup
Lower memory
consumption

Comparing GraalVM, CRaC and CDS

	Fast startup	Fast warm-up	Optimal peak throughput	Memory consumption	Compilation	Compatibility issues	Security	Regular Java distribution
Native with GraalVM	🟩 Instant	🟩 Instant	🟧 Training run with Oracle GraalVM	🟩 Reduced	🟥 Heavy and slow	🟧 Hints + AOT bean conditions	🟩	🟧 No but not needed at runtime
JVM with CRaC	🟩 Instant	🟧 Training run	🟩	🟧 No gain	🟩 Fast	🟥 Complex lifecycle issues	🟥 Secret leaking in snapshots	🟥 No Linux only
JVM with CDS	🟩 1.5x faster 2x with Spring AOT	🟧 Training run with Project Leyden	🟩	🟩 Slightly reduced	🟩 Fast	🟧 Training run with custom configuration	🟩	🟩

JVM with CDS

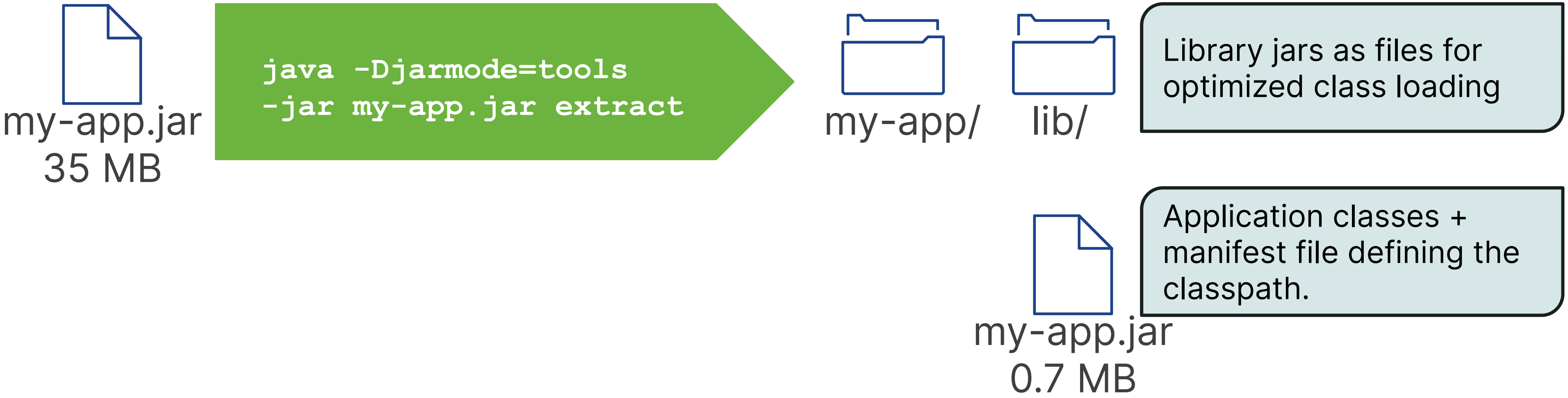
Recommended for better efficiency with few constraints.

Native with GraalVM

Great option if you can deal with long build times and compatibility challenges.

Spring Boot 3.3 introduces CDS support

Executable JAR self-extract in Spring Boot 3.3



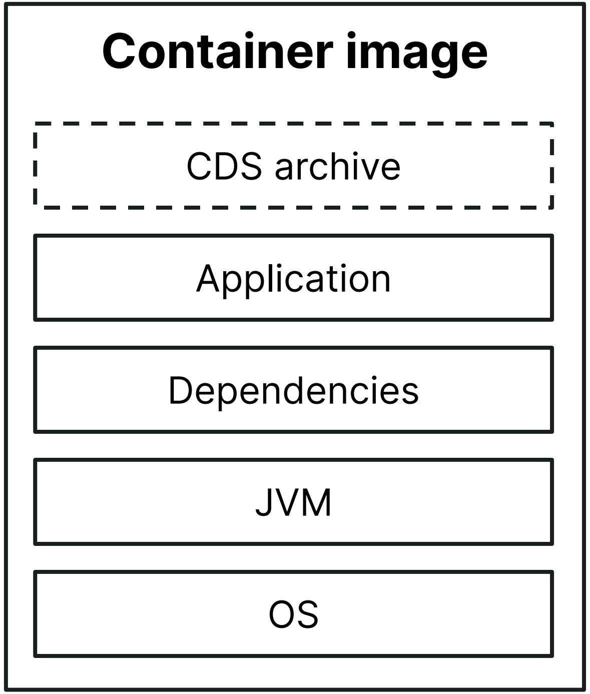
```
java -Djarmode=tools
-jar my-app.jar extract
```

Library jars as files for optimized class loading

Application classes + manifest file defining the classpath.

CDS support in Buildpacks

Spring Boot application + Paketo Buildpacks =



Buildpacks leverages Spring Boot 3.3 self-extracting feature which is CDS friendly.

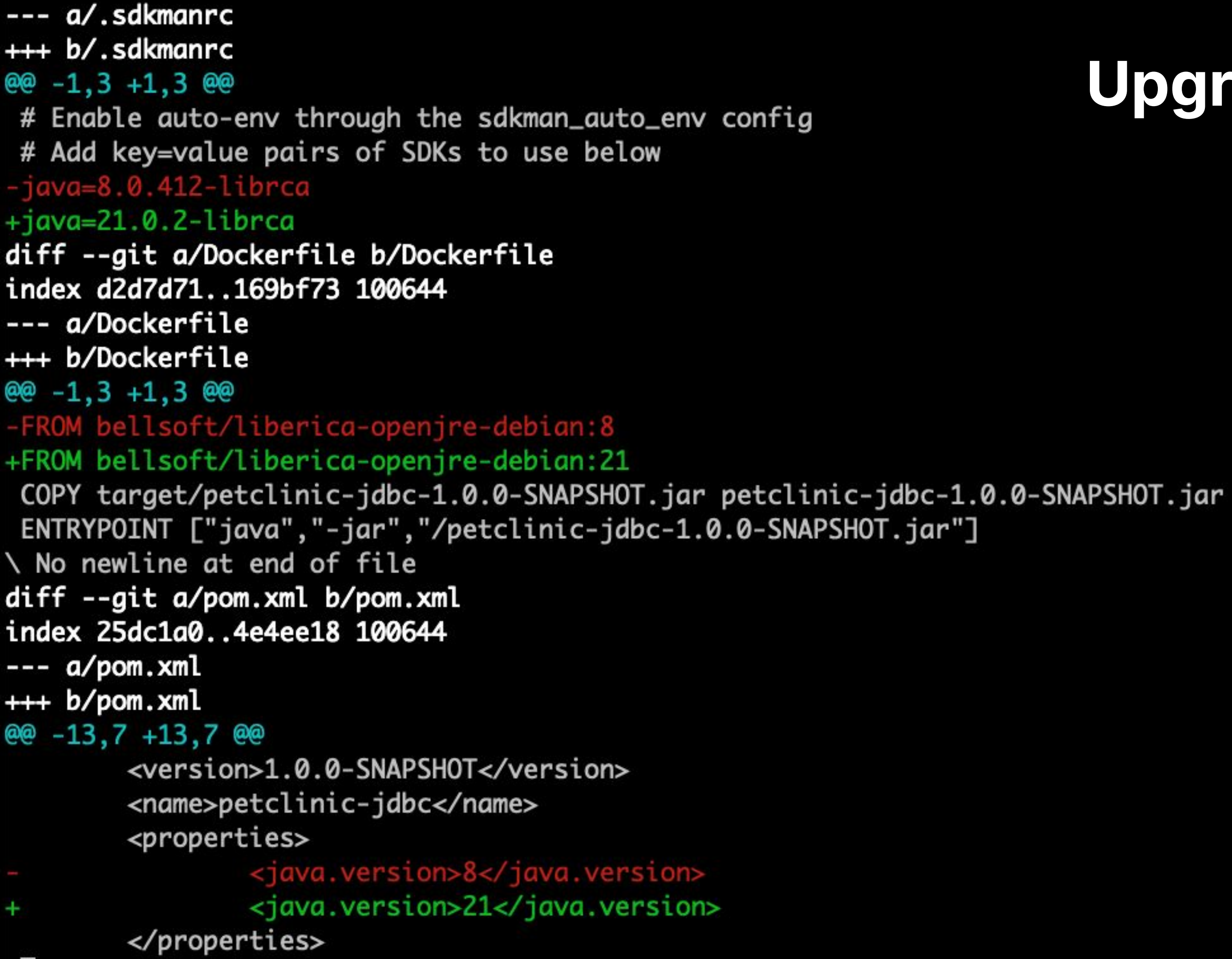
The CDS training run is performed transparently when building the container image with the same JVM used at runtime!

Let's upgrade and optimize Petclinic!

https://github.com/sdeleuze/petclinic-efficient-container

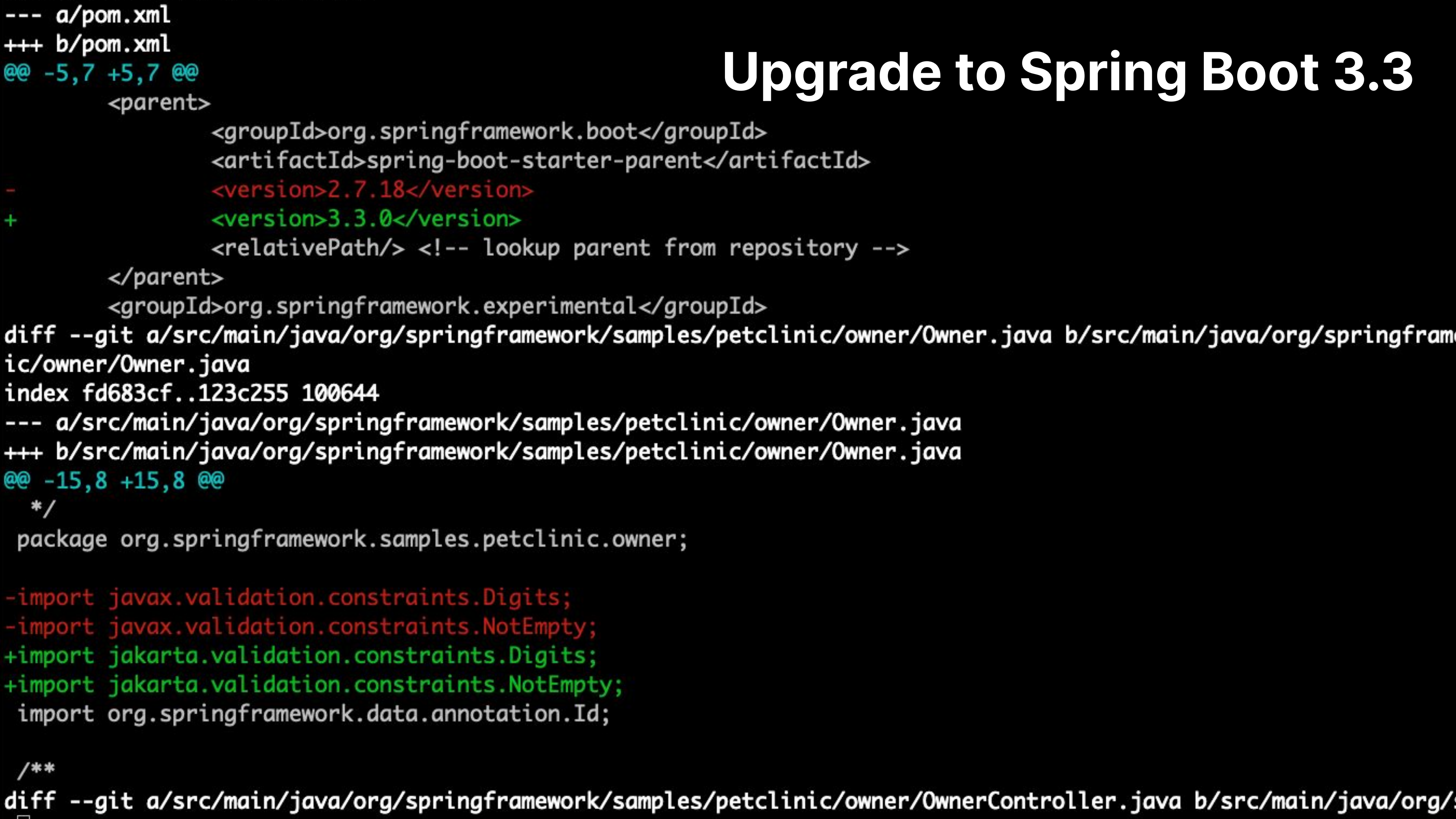
Spring Boot 2.7 Java 8 baseline

Upgrade to Java 21

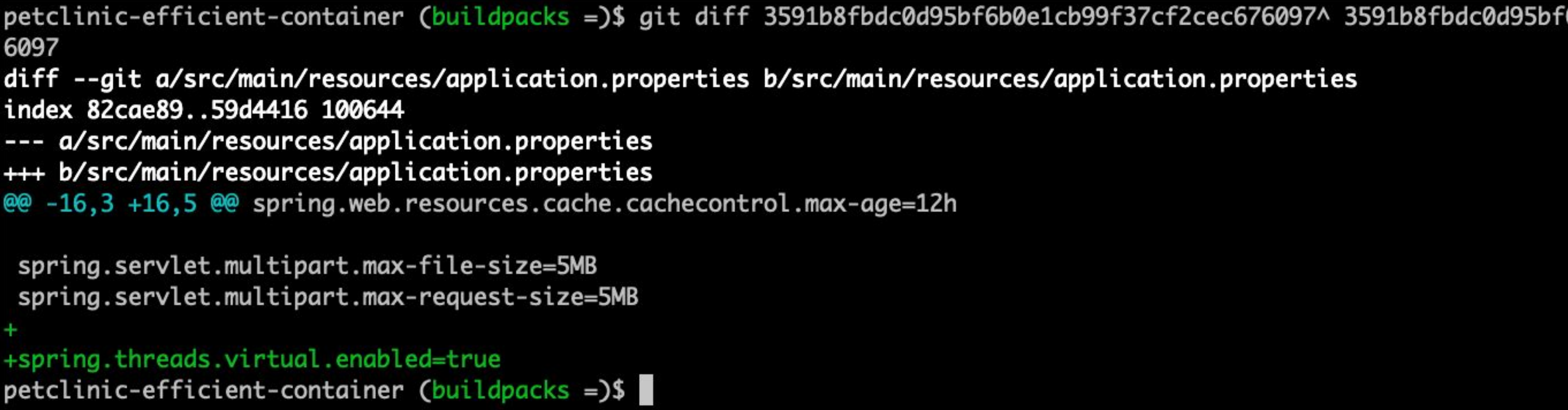


```
--- a/.sdkmanrc
+++ b/.sdkmanrc
@@ -1,3 +1,3 @@
 # Enable auto-env through the sdkman_auto_env config
 # Add key=value pairs of SDKs to use below
-java=8.0.412-librca
+java=21.0.2-librca
diff --git a/Dockerfile b/Dockerfile
index d2d7d71..169bf73 100644
--- a/Dockerfile
+++ b/Dockerfile
@@ -1,3 +1,3 @@
-FROM bellsoft/liberica-openjre-debian:8
+FROM bellsoft/liberica-openjre-debian:21
 COPY target/petclinic-jdbc-1.0.0-SNAPSHOT.jar petclinic-jdbc-1.0.0-SNAPSHOT.jar
 ENTRYPOINT ["java","-jar","/petclinic-jdbc-1.0.0-SNAPSHOT.jar"]
\ No newline at end of file
diff --git a/pom.xml b/pom.xml
index 25dc1a0..4e4ee18 100644
--- a/pom.xml
+++ b/pom.xml
@@ -13,7 +13,7 @@
        <version>1.0.0-SNAPSHOT</version>
        <name>petclinic-jdbc</name>
        <properties>
-               <java.version>8</java.version>
+               <java.version>21</java.version>
        </properties>
```

Upgrade to Spring Boot 3.3



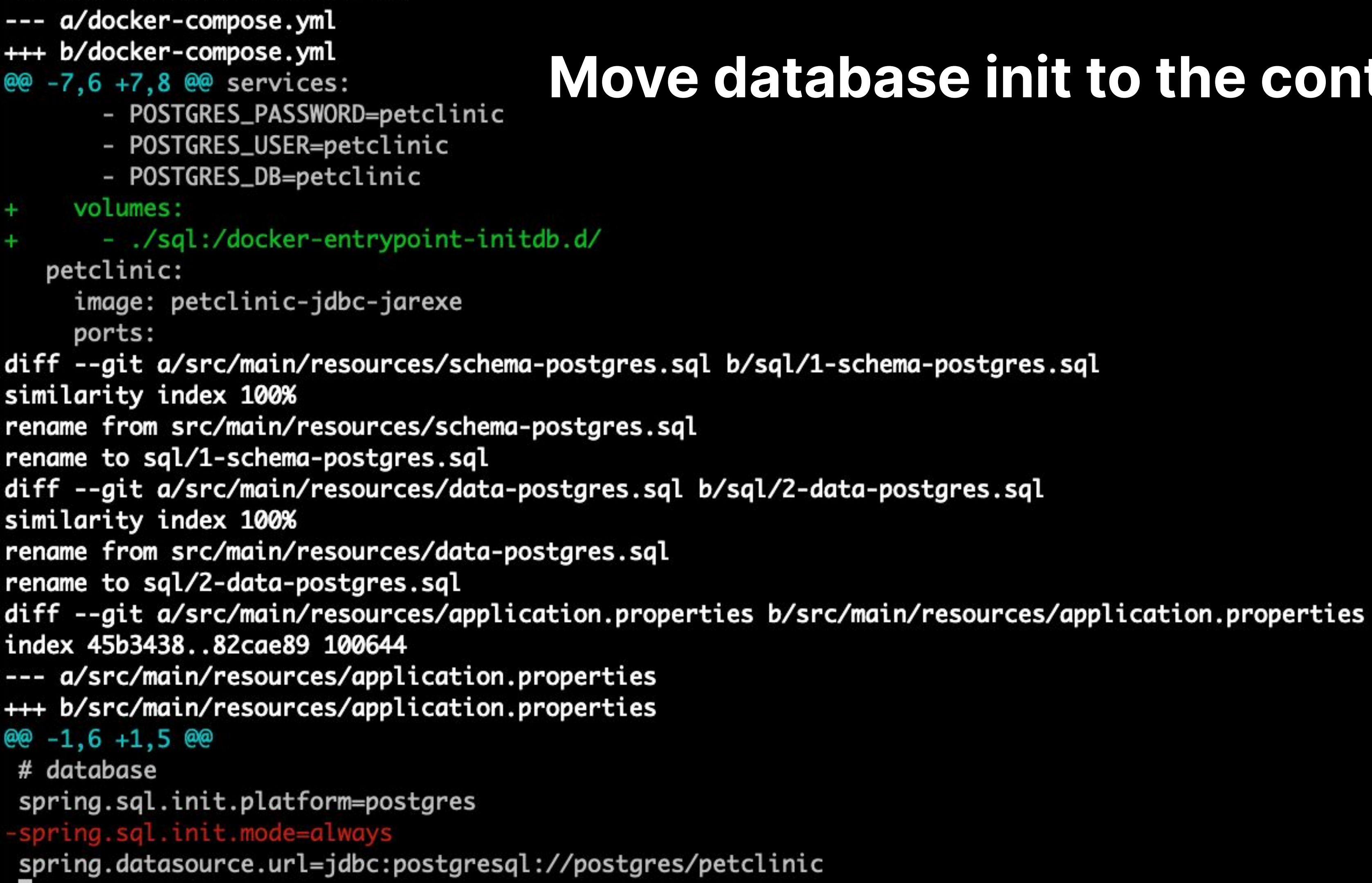
```
--- a/pom.xml
+++ b/pom.xml
@@ -5,7 +5,7 @@
        <parent>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-parent</artifactId>
-               <version>2.7.18</version>
+               <version>3.3.0</version>
                <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>org.springframework.experimental</groupId>
diff --git a/src/main/java/org/springframework/samples/petclinic/owner/Owner.java b/src/main/java/org/springfram
ic/owner/Owner.java
index fd683cf..123c255 100644
--- a/src/main/java/org/springframework/samples/petclinic/owner/Owner.java
+++ b/src/main/java/org/springframework/samples/petclinic/owner/Owner.java
@@ -15,8 +15,8 @@
  */
 package org.springframework.samples.petclinic.owner;

-import javax.validation.constraints.Digits;
-import javax.validation.constraints.NotEmpty;
+import jakarta.validation.constraints.Digits;
+import jakarta.validation.constraints.NotEmpty;
 import org.springframework.data.annotation.Id;

 /**
diff --git a/src/main/java/org/springframework/samples/petclinic/owner/OwnerController.java b/src/main/java/org/
```



```
petclinic-efficient-container (buildpacks =)$ git diff 3591b8fbdc0d95bf6b0e1cb99f37cf2cec676097^ 3591b8fbdc0d95bf
6097
diff --git a/src/main/resources/application.properties b/src/main/resources/application.properties
index 82cae89..59d4416 100644
--- a/src/main/resources/application.properties
+++ b/src/main/resources/application.properties
@@ -16,3 +16,5 @@ spring.web.resources.cache.cachecontrol.max-age=12h

 spring.servlet.multipart.max-file-size=5MB
 spring.servlet.multipart.max-request-size=5MB
+
+spring.threads.virtual.enabled=true
petclinic-efficient-container (buildpacks =)$ 
```

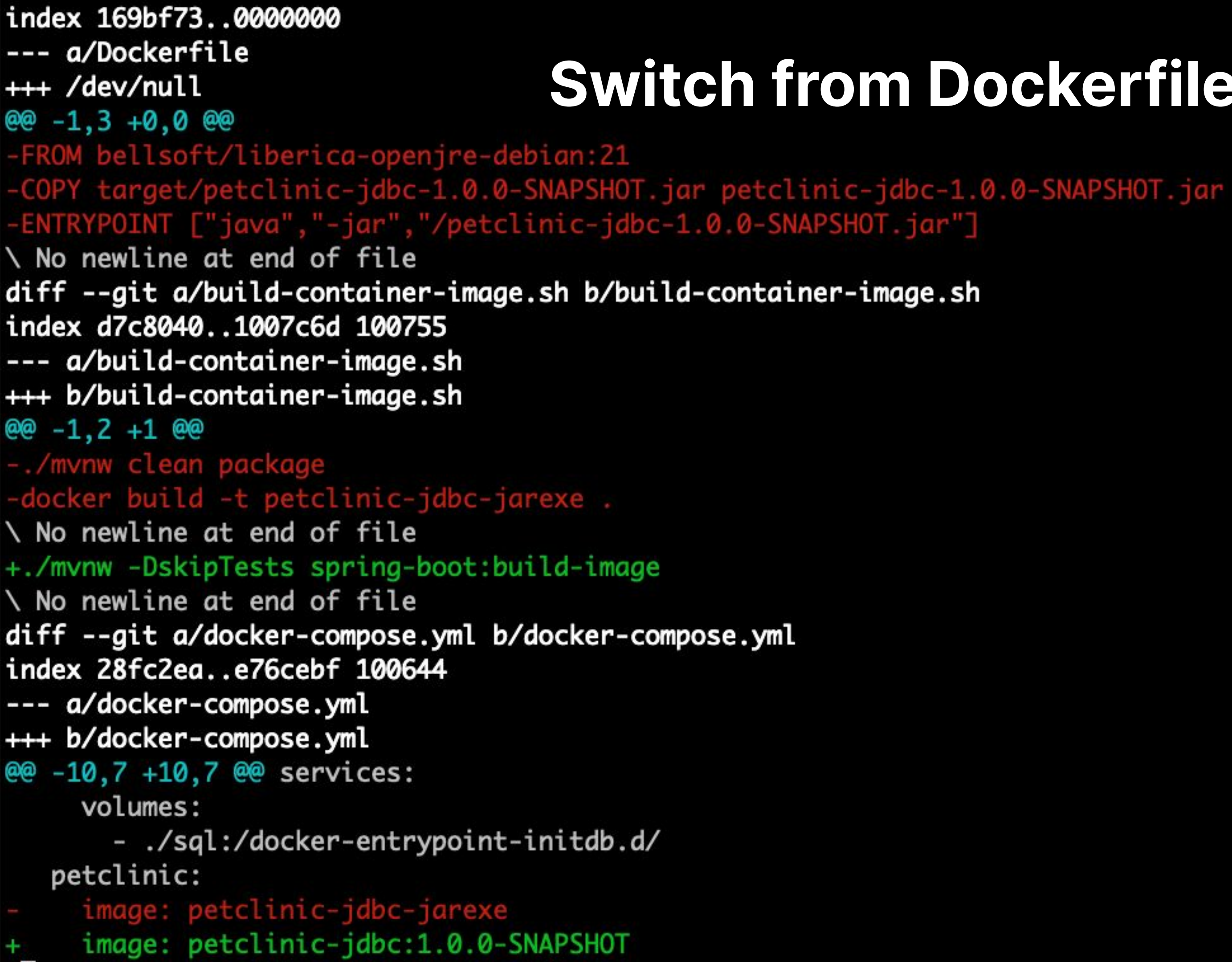
Enable Virtual Threads

Move database init to the container



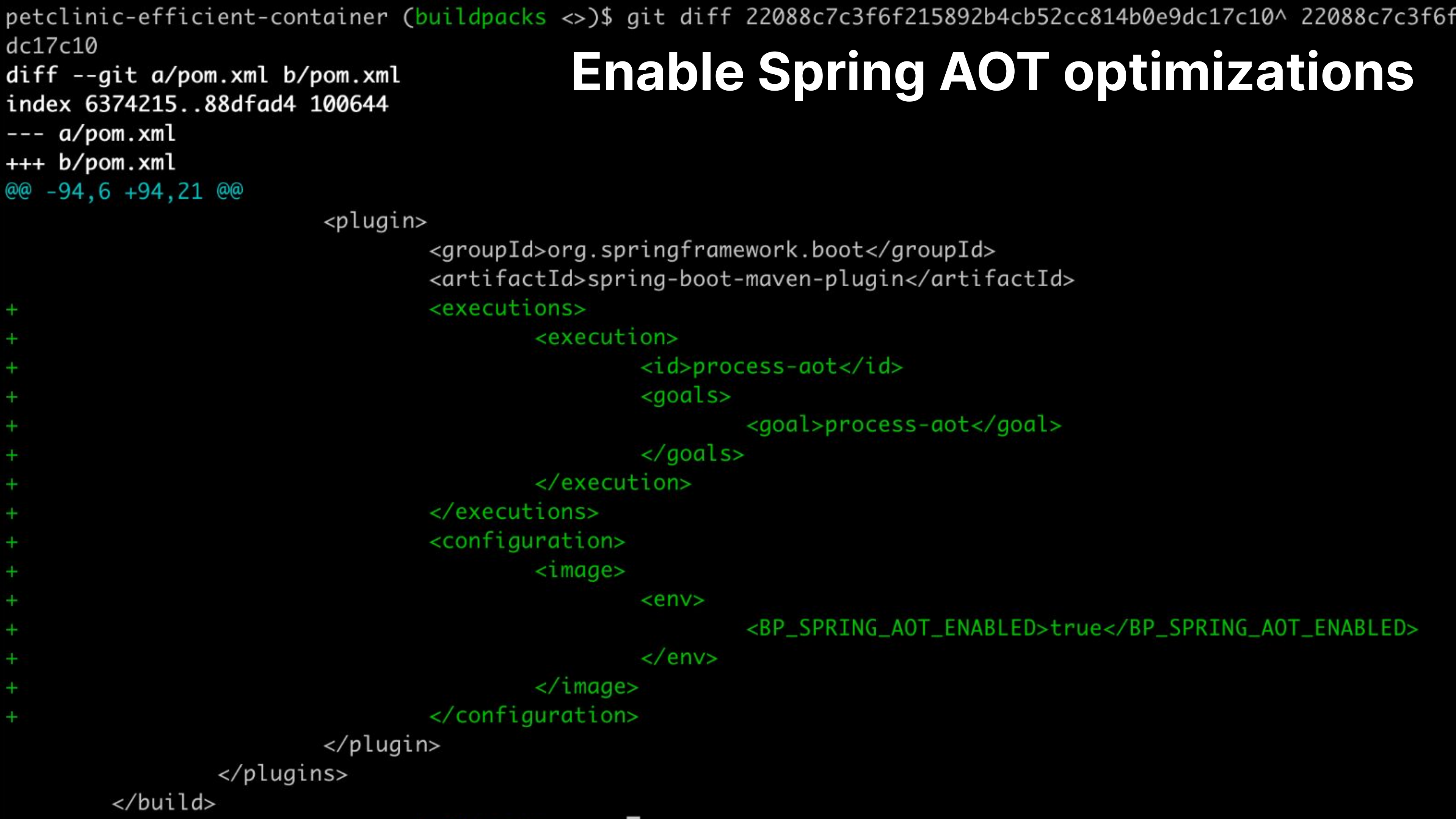
```
--- a/docker-compose.yml
+++ b/docker-compose.yml
@@ -7,6 +7,8 @@ services:
       - POSTGRES_PASSWORD=petclinic
       - POSTGRES_USER=petclinic
       - POSTGRES_DB=petclinic
+    volumes:
+      - ./sql:/docker-entrypoint-initdb.d/
   petclinic:
     image: petclinic-jdbc-jarexe
     ports:
diff --git a/src/main/resources/schema-postgres.sql b/sql/1-schema-postgres.sql
similarity index 100%
rename from src/main/resources/schema-postgres.sql
rename to sql/1-schema-postgres.sql
diff --git a/src/main/resources/data-postgres.sql b/sql/2-data-postgres.sql
similarity index 100%
rename from src/main/resources/data-postgres.sql
rename to sql/2-data-postgres.sql
diff --git a/src/main/resources/application.properties b/src/main/resources/application.properties
index 45b3438..82cae89 100644
--- a/src/main/resources/application.properties
+++ b/src/main/resources/application.properties
@@ -1,6 +1,5 @@
 # database
 spring.sql.init.platform=postgres
-spring.sql.init.mode=always
 spring.datasource.url=jdbc:postgresql://postgres/petclinic
```

Switch from Dockerfile to Buildpacks



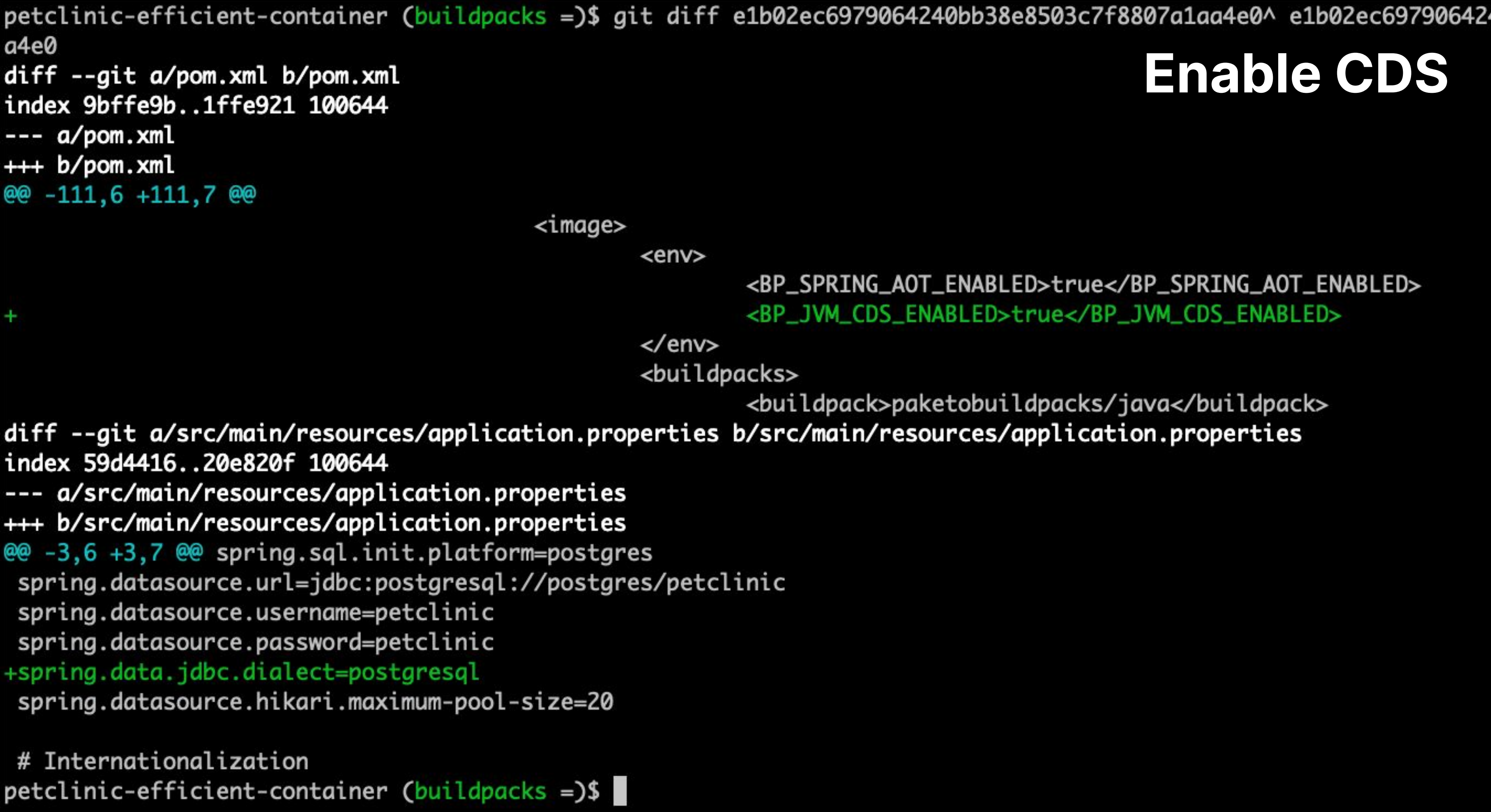
```
index 169bf73..0000000
--- a/Dockerfile
+++ /dev/null
@@ -1,3 +0,0 @@
-FROM bellsoft/liberica-openjre-debian:21
-COPY target/petclinic-jdbc-1.0.0-SNAPSHOT.jar petclinic-jdbc-1.0.0-SNAPSHOT.jar
-ENTRYPOINT ["java","-jar","/petclinic-jdbc-1.0.0-SNAPSHOT.jar"]
\ No newline at end of file
diff --git a/build-container-image.sh b/build-container-image.sh
index d7c8040..1007c6d 100755
--- a/build-container-image.sh
+++ b/build-container-image.sh
@@ -1,2 +1 @@
-./mvnw clean package
-docker build -t petclinic-jdbc-jarexe .
\ No newline at end of file
+./mvnw -DskipTests spring-boot:build-image
\ No newline at end of file
diff --git a/docker-compose.yml b/docker-compose.yml
index 28fc2ea..e76cebf 100644
--- a/docker-compose.yml
+++ b/docker-compose.yml
@@ -10,7 +10,7 @@ services:
     volumes:
       - ./sql:/docker-entrypoint-initdb.d/
   petclinic:
-    image: petclinic-jdbc-jarexe
+    image: petclinic-jdbc:1.0.0-SNAPSHOT
```

Enable Spring AOT optimizations



```
petclinic-efficient-container (buildpacks <>)$ git diff 22088c7c3f6f215892b4cb52cc814b0e9dc17c10^ 22088c7c3f6f
dc17c10
diff --git a/pom.xml b/pom.xml
index 6374215..88dfad4 100644
--- a/pom.xml
+++ b/pom.xml
@@ -94,6 +94,21 @@
                        <plugin>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-maven-plugin</artifactId>
+                               <executions>
+                                       <execution>
+                                               <id>process-aot</id>
+                                               <goals>
+                                                       <goal>process-aot</goal>
+                                               </goals>
+                                       </execution>
+                               </executions>
+                               <configuration>
+                                       <image>
+                                               <env>
+                                                       <BP_SPRING_AOT_ENABLED>true</BP_SPRING_AOT_ENABLED>
+                                               </env>
+                                       </image>
+                               </configuration>
                        </plugin>
                </plugins>
        </build>
```

Enable CDS



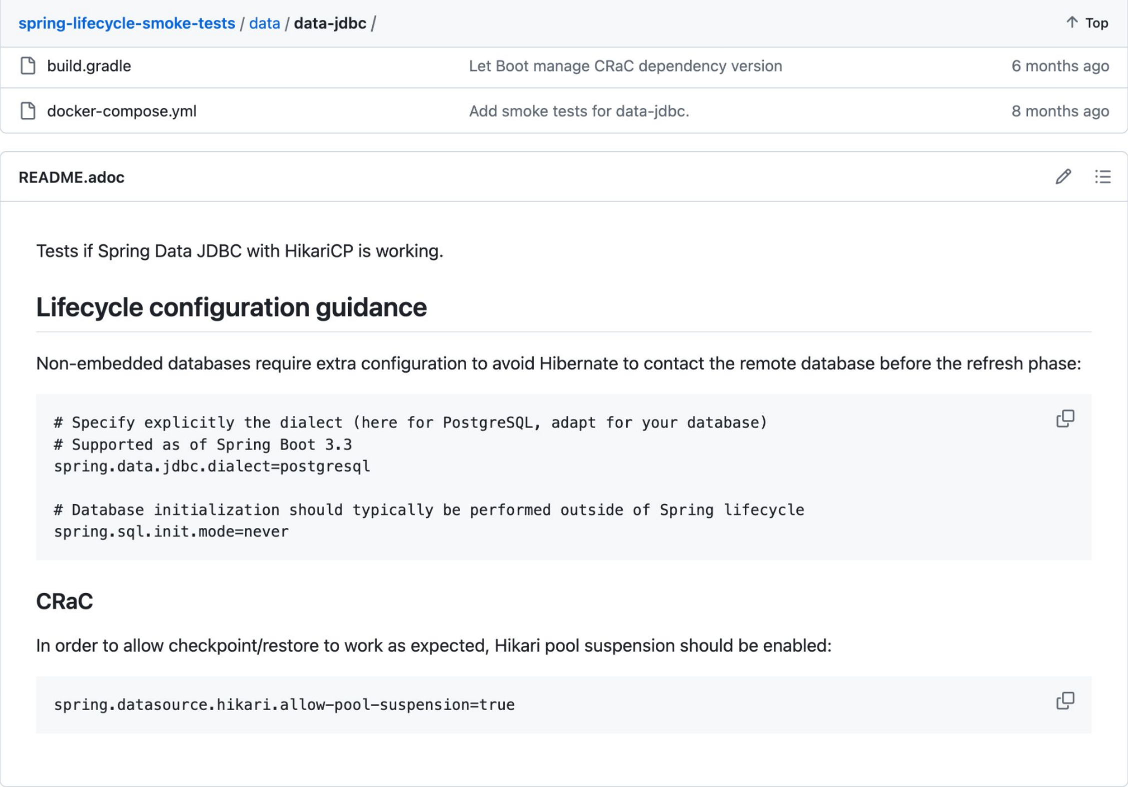
```
petclinic-efficient-container (buildpacks =)$ git diff e1b02ec6979064240bb38e8503c7f8807a1aa4e0^ e1b02ec69790642
a4e0
diff --git a/pom.xml b/pom.xml
index 9bffe9b..1ffe921 100644
--- a/pom.xml
+++ b/pom.xml
@@ -111,6 +111,7 @@
                                        <image>
                                                <env>
                                                        <BP_SPRING_AOT_ENABLED>true</BP_SPRING_AOT_ENABLED>
+                                                       <BP_JVM_CDS_ENABLED>true</BP_JVM_CDS_ENABLED>
                                                </env>
                                                <buildpacks>
                                                        <buildpack>paketobuildpacks/java</buildpack>
diff --git a/src/main/resources/application.properties b/src/main/resources/application.properties
index 59d4416..20e820f 100644
--- a/src/main/resources/application.properties
+++ b/src/main/resources/application.properties
@@ -3,6 +3,7 @@ spring.sql.init.platform=postgres
 spring.datasource.url=jdbc:postgresql://postgres/petclinic
 spring.datasource.username=petclinic
 spring.datasource.password=petclinic
+spring.data.jdbc.dialect=postgresql
 spring.datasource.hikari.maximum-pool-size=20

 # Internationalization
petclinic-efficient-container (buildpacks =)$
```

Configuration to prevent early database interactions

https://github.com/spring-projects/spring-lifecycle-smoke-tests/

Use regular configuration (apply to both training and optimized runs)
or use Buildpacks
`CDS_TRAINING_JAVA_TOOL_OPTIONS`
environment variable for:

- Spring Data JPA
- Spring Data JDBC
- Spring Data R2DBC
- Spring Data MongDB
- Spring Data Redis
- Spring Batch
- Flyway
- Liquibase
- …



spring-lifecycle-smoke-tests / data / data-jdbc /

build.gradle	Let Boot manage CRaC dependency version	6 months ago
docker-compose.yml	Add smoke tests for data-jdbc.	8 months ago

README.adoc

Tests if Spring Data JDBC with HikariCP is working.

Lifecycle configuration guidance

Non-embedded databases require extra configuration to avoid Hibernate to contact the remote database before the refresh phase:

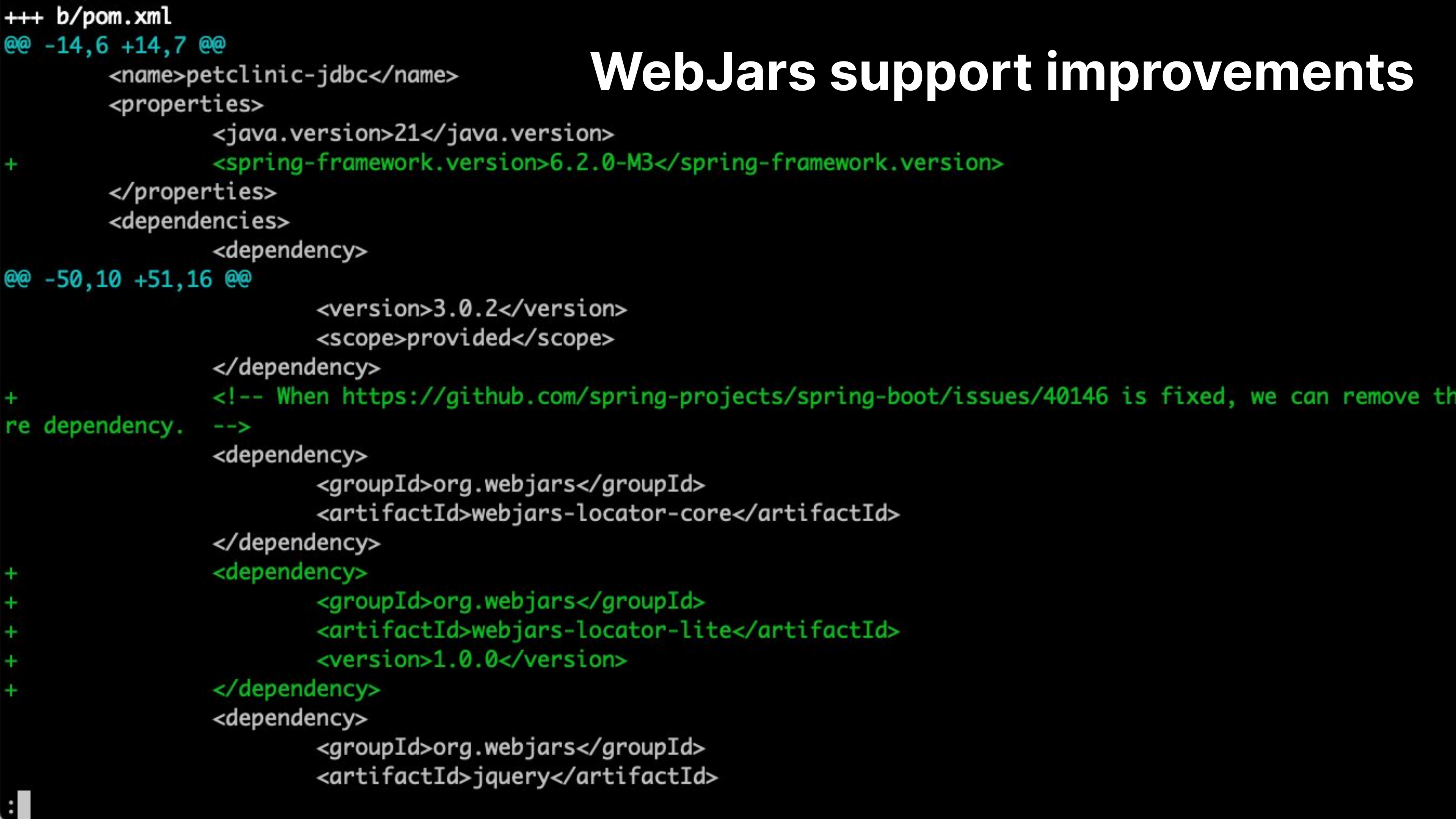
```
# Specify explicitly the dialect (here for PostgreSQL, adapt for your database)
# Supported as of Spring Boot 3.3
spring.data.jdbc.dialect=postgresql

# Database initialization should typically be performed outside of Spring lifecycle
spring.sql.init.mode=never
```

CRaC

In order to allow checkpoint/restore to work as expected, Hikari pool suspension should be enabled:

```
spring.datasource.hikari.allow-pool-suspension=true
```

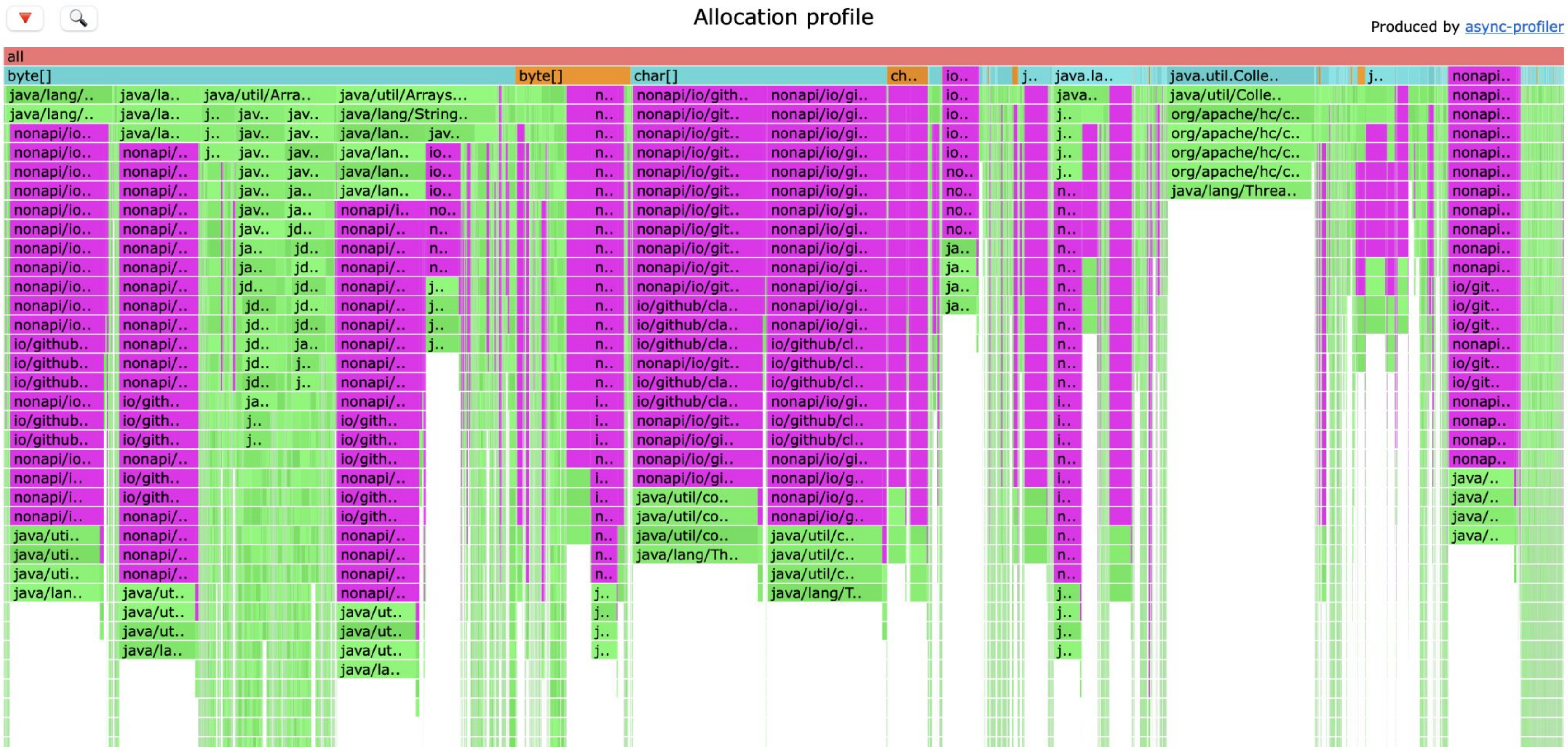
WebJars support improvements



```
+++ b/pom.xml
@@ -14,6 +14,7 @@
        <name>petclinic-jdbc</name>
        <properties>
                <java.version>21</java.version>
+               <spring-framework.version>6.2.0-M3</spring-framework.version>
        </properties>
        <dependencies>
                <dependency>
@@ -50,10 +51,16 @@
                        <version>3.0.2</version>
                        <scope>provided</scope>
                </dependency>
+               <!-- When https://github.com/spring-projects/spring-boot/issues/40146 is fixed, we can remove th
re dependency.  -->
                <dependency>
                        <groupId>org.webjars</groupId>
                        <artifactId>webjars-locator-core</artifactId>
                </dependency>
+               <dependency>
+                       <groupId>org.webjars</groupId>
+                       <artifactId>webjars-locator-lite</artifactId>
+                       <version>1.0.0</version>
+               </dependency>
                <dependency>
                        <groupId>org.webjars</groupId>
                        <artifactId>jquery</artifactId>
:
```

WebJars support improvements

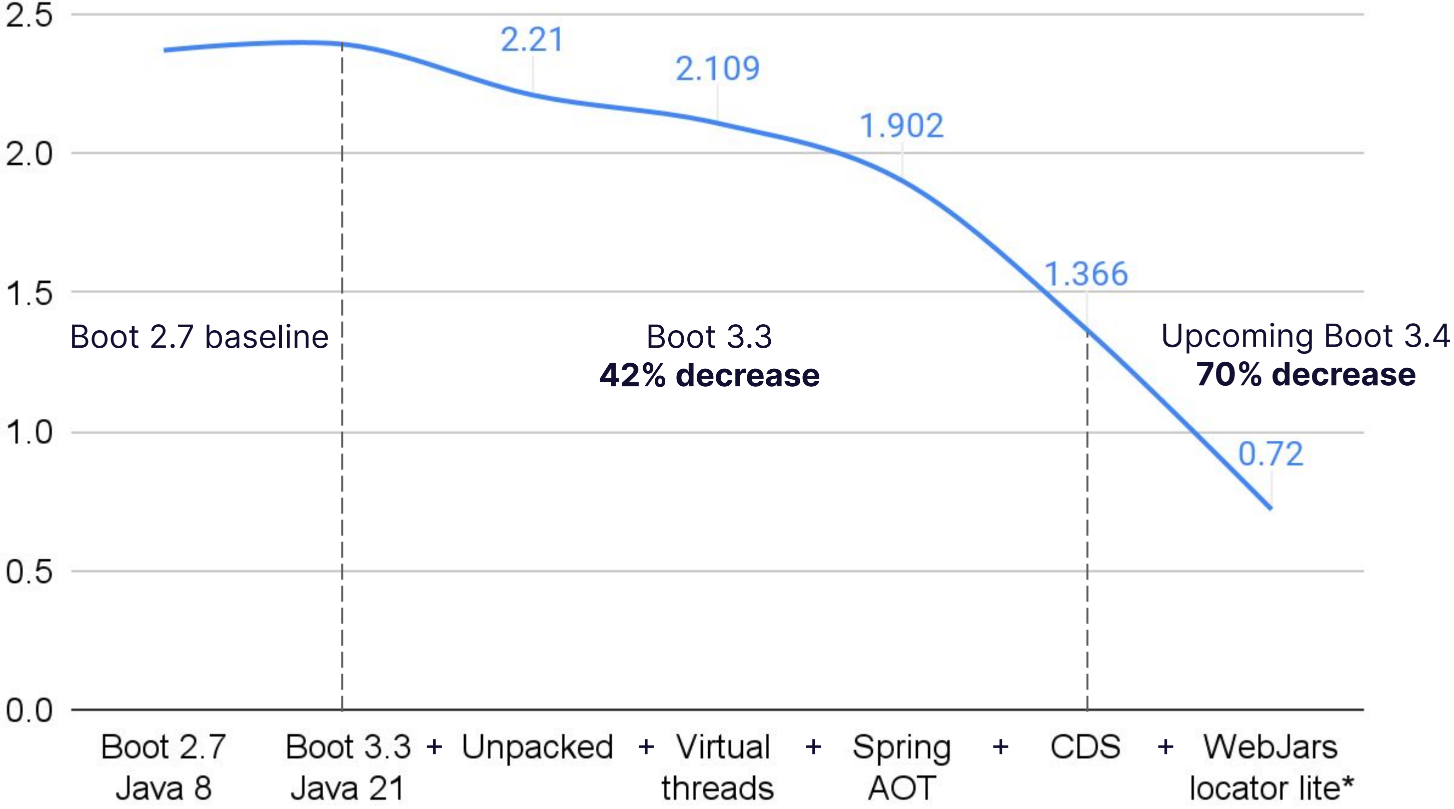
https://github.com/spring-projects/spring-framework/issues/27619



Spring Boot 3 efficient container

Petclinic JDBC application startup time (s)

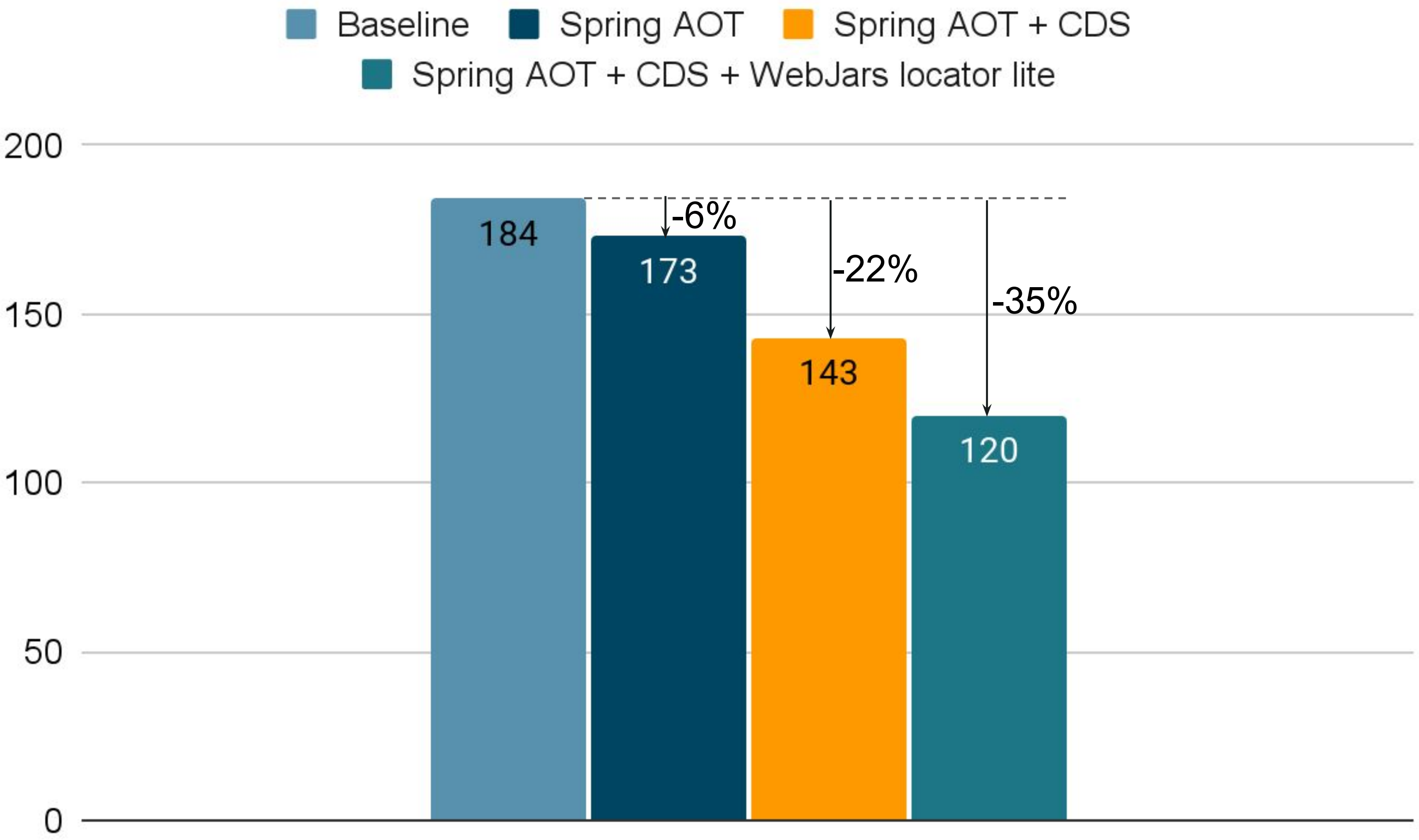
Lower is better - 2 CPUs 2G RAM



* See spring-boot#40146 “Add support for webjars-locator-lite” currently planned for Spring Boot 3.4.x (based on Spring Framework 6.2)

Petclinic JDBC Memory consumption (MB)

Lower is better - 2 CPUs 1G RAM

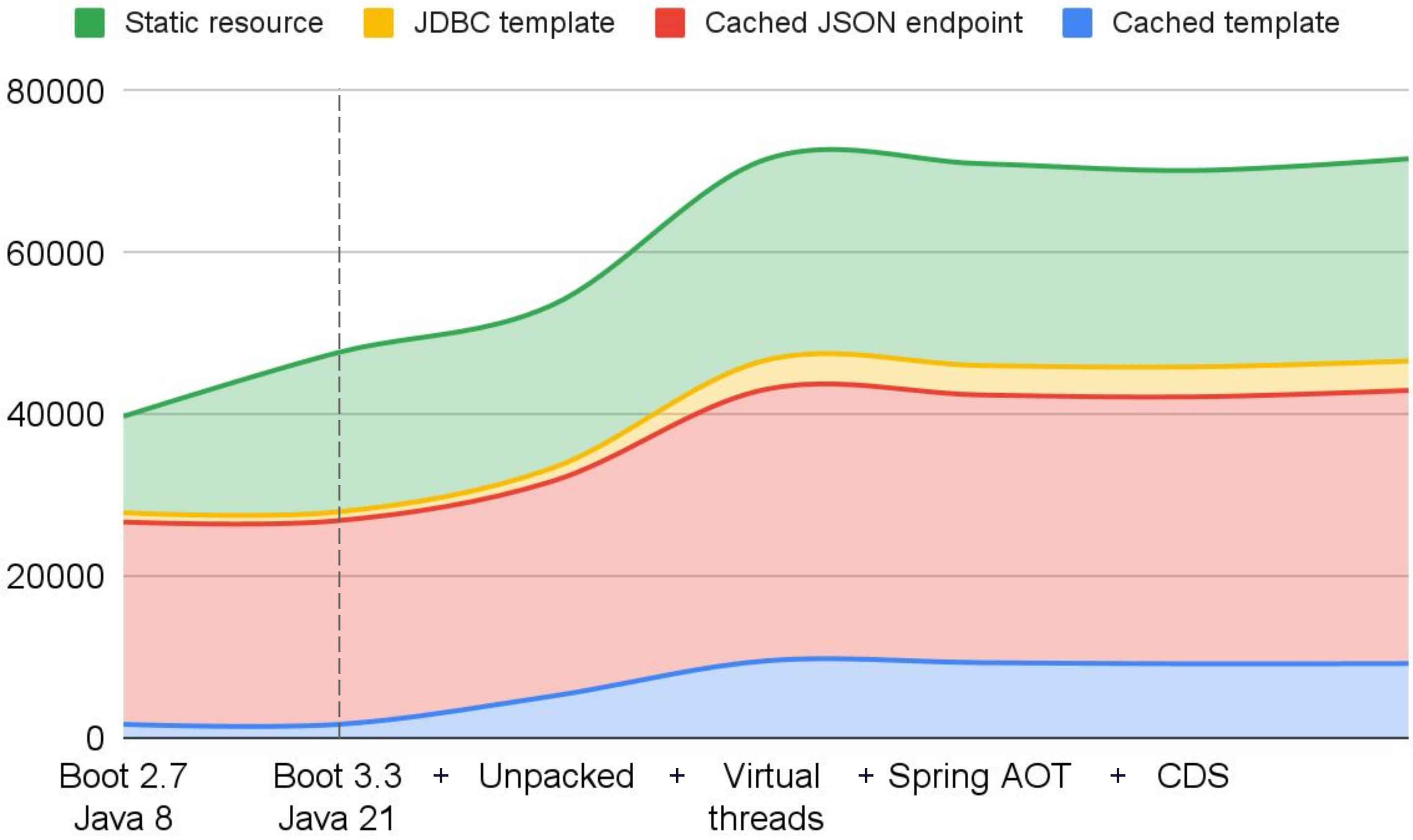


Benchmarking with oha: warmup

Benchmarking with oha: peak performance

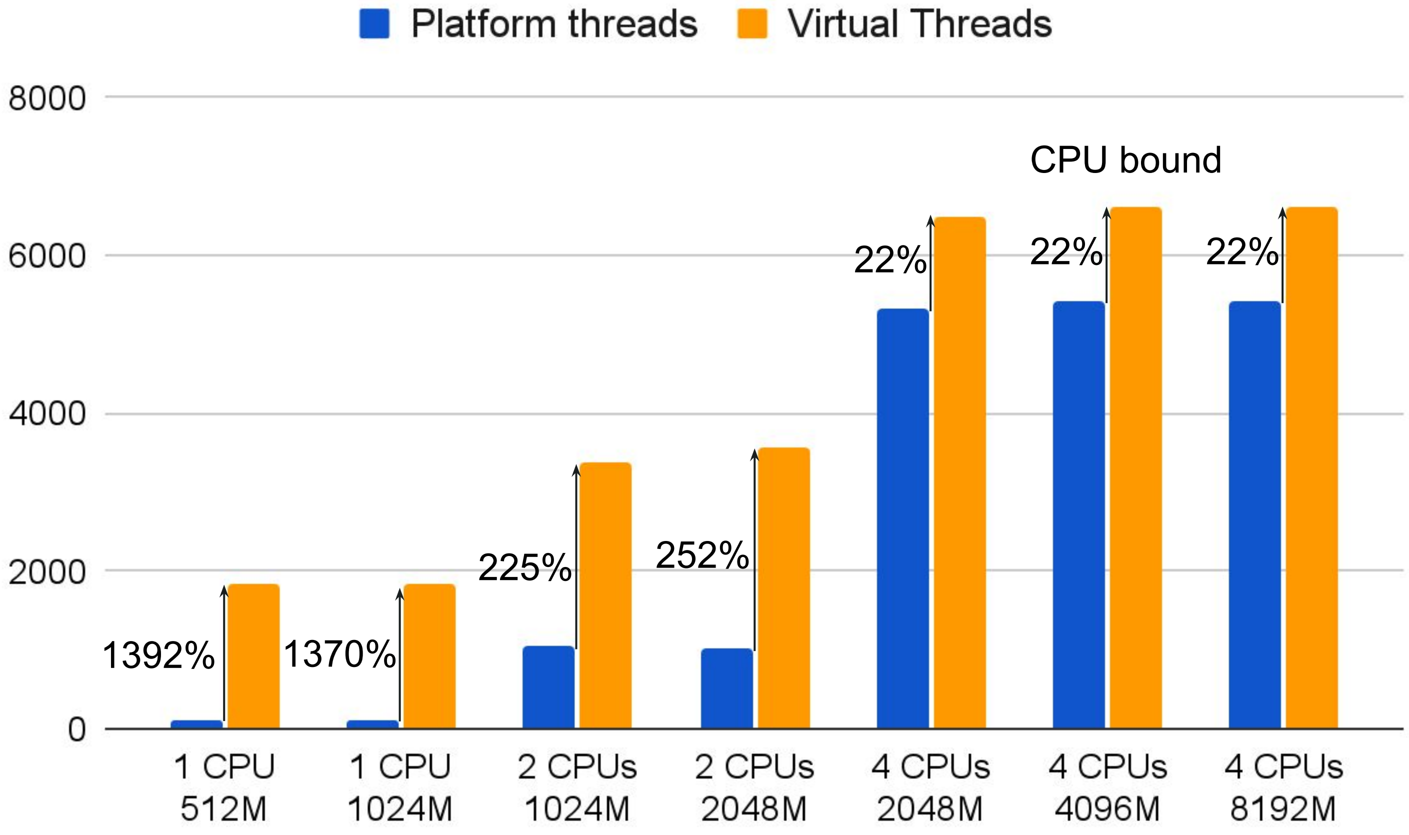
Petclinic JDBC throughput (req/s)

Higher is better - 2 CPUs 2G RAM



Petclinic JDBC template throughput (req/s)

Higher is better



Spring Boot 3.3, Java 21, virtual threads and CDS allow you to use smaller and cheaper instances with better performance.

Collaboration between Java Platform and Spring teams on Project Leyden for even faster startup and warm-up with smaller containers!

```
Started PetClinicApplication in 0.298 seconds
(process running for 0.411)
```

Thanks!

Sébastien Deleuze

https://seb.deleuze.fr

https://github.com/sdeleuze/petclinic-efficient-container

THE SPRING IO
DEVELOPER CONFERENCE