

Crafting the Ultimate Docker Image for Spring Applications



bellsoft

whoami



whoami

- Pasha Finkelshteyn



whoami

- Pasha Finkelshteyn
- Dev  at BellSoft



whoami

- Pasha Finkelshteyn
- Dev  at BellSoft
- ≈10 years in JVM. Mostly  and 



whoami

- Pasha Finkelshteyn
- Dev  at BellSoft
- ≈10 years in JVM. Mostly  and 
- And 



whoami

- Pasha Finkelshteyn
- Dev  at BellSoft
- ≈10 years in JVM. Mostly  and 
- And 
-  asm0di0



whoami

- Pasha Finkelshteyn
- Dev  at BellSoft
- ≈10 years in JVM. Mostly  and 
- And 
-  asm0di0
-  @asm0dey@fosstodon.org



BellSoft

- Vendor of Liberica JDK
- Contributor to the OpenJDK
- Author of ARM32 support in JDK
- Own base images
- Own Linux: Alpaquita

Liberica is the JDK officially recommended
by 



BellSoft

- Vendor of Liberica JDK
- Contributor to the OpenJDK
- Author of ARM32 support in JDK
- Own base images
- Own Linux: Alpaquita

Liberica is the JDK officially recommended
by



We know our stuff!



So, what is ultimate?

So, what is ultimate?

- Smallest

So, what is ultimate?

- Smallest
- Fatsest startup

So, what is ultimate?

- Smallest
- Fatsest startup
- Something inbetween

So, let's look at all of
them!

Common ground

1. Each directive creates a layer of the image.
2. Layers are *immutable*
3. Some layers are zero-sized

Common ground

1. Each directive creates a layer of the image.
2. Layers are *immutable*
3. Some layers are zero-sized

```
1 FROM bash  
2  
3 COPY . /app  
4 RUN rm -rf /app
```

Common ground

1. Each directive creates a layer of the image.
2. Layers are *immutable*
3. Some layers are zero-sized

```
1  FROM bash  
2  
3  COPY . /app  
4  RUN rm -rf /app
```

Common ground

1. Each directive creates a layer of the image.
2. Layers are *immutable*
3. Some layers are zero-sized (for example RUN rm -rf /app)

```
1 FROM bash
2
3 COPY . /app
4 RUN rm -rf /app
```

Common ground

1. Each directive creates a layer of the image.
2. Layers are *immutable*
3. Some layers are zero-sized (for example `RUN rm -rf /app`)

```
1 FROM bash
2
3 COPY . /app
4 RUN rm -rf /app
```

4. We would like image to be light, but it's not :(

Let's start simple

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Let's start simple

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Let's start simple

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Let's start simple

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Let's start simple

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Result

```
13 <missing>                                0B      ARG BUNDLE_TYPE
14 <missing>                                0B      ARG BUNDLE_TYPE
15 <missing>                                0B      ARG BUNDLE_TYPE
16 <missing>                                0B      ARG JAVA_RELEASEN
17 <missing>                                7.49MiB  CMD [ "/bin/sh" ]
18 <missing>                                0B      |9 APK_REPOS= D
19 <missing>                                0B      ADD file:a4d77f
20 <missing>                                0B      LABEL org.opend
21 <missing>                                0B      LABEL org.opend
22 <missing>                                0B      LABEL maintaine
23 <missing>                                0B      ARG APK_REPOS D
24 <missing>                                0B      ARG APK_REPOS L
25 <missing>                                0B      ARG APK_REPOS L
26 <missing>                                0B      ARG APK_REPOS L
27 <missing>                                0B      ARG LIBC MINIRO
28 <missing>                                0B      ARG MINIROOTFS
```

Result

9	<missing>	0B	LABEL org.openc
10	<missing>	0B	LABEL maintaine
11	<missing>	0B	ARG BUNDLE_TYPE
12	<missing>	0B	ARG BUNDLE_TYPE
13	<missing>	0B	ARG BUNDLE_TYPE
14	<missing>	0B	ARG BUNDLE_TYPE
15	<missing>	0B	ARG BUNDLE_TYPE
16	<missing>	0B	ARG JAVA_RELEASE
17	<missing>	7.49MiB	CMD ["/bin/sh"]
18	<missing>	0B	9 APK_REPOS= D
19	<missing>	0B	ADD file:a4d77f
20	<missing>	0B	LABEL org.openc
21	<missing>	0B	LABEL org.openc
22	<missing>	0B	LABEL maintaine
23	<missing>	0B	ARG APK_REPOS D
24	<missing>	0B	ARG APK_REPOS L
25	<missing>	0B	ARG APK_REPOS L

Result

3	74a0788411	512.03MiB	/bin/sh -c cd /
4	7b82eb08ec	67.41MiB	COPY dir:ef251b
5	e4bcf83d29 bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE
6	<missing>	0B	ENV JAVA_HOME="
7	<missing>	0B	ENV LANG=en_US.
8	<missing>	0B	LABEL org.openc
9	<missing>	0B	LABEL org.openc
10	<missing>	0B	LABEL maintaine
11	<missing>	0B	ARG BUNDLE_TYPE
12	<missing>	0B	ARG BUNDLE_TYPE
13	<missing>	0B	ARG BUNDLE_TYPE
14	<missing>	0B	ARG BUNDLE_TYPE
15	<missing>	0B	ARG BUNDLE_TYPE
16	<missing>	0B	ARG JAVA_RELEASE
17	<missing>	7.49MiB	CMD ["/bin/sh"]
18	<missing>	0B	9 APK_REPOS= D
19	<missing>	0B	ADD file:a4d77f

Result

	ID	TAG	SIZE	COMMAND
1	cda5201e70	dumb:latest	0B	CMD ["/bin/sh"
2	74a0788411		512.03MiB	/bin/sh -c cd /
3	7b82eb08ec		67.41MiB	COPY dir:ef251b
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE
5	<missing>		0B	ENV JAVA_HOME="
6	<missing>		0B	ENV LANG=en_US.
7	<missing>		0B	LABEL org.openc
8	<missing>		0B	LABEL org.openc
9	<missing>		0B	LABEL maintaine
10	<missing>		0B	ARG BUNDLE_TYPE
11	<missing>		0B	ARG BUNDLE_TYPE
12	<missing>		0B	ARG BUNDLE_TYPE
13	<missing>		0B	ARG BUNDLE_TYPE
14	<missing>		0B	ARG BUNDLE_TYPE
15	<missing>		0B	ARG BUNDLE_TYPE
16	<missing>		0B	ARG JAVA_RELEASE

Result

	ID	TAG	SIZE	COMMAND
1	cda5201e70	dumb:latest	0B	CMD ["/bin/sh"
2	74a0788411		512.03MiB	/bin/sh -c cd /
3	7b82eb08ec		67.41MiB	COPY dir:ef251b
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE
5	<missing>		0B	ENV JAVA_HOME="
6	<missing>		0B	ENV LANG=en_US.
7	<missing>		0B	LABEL org.openc
8	<missing>		0B	LABEL org.openc
9	<missing>		0B	LABEL maintaine
10	<missing>		0B	ARG BUNDLE_TYPE
11	<missing>		0B	ARG BUNDLE_TYPE
12	<missing>		0B	ARG BUNDLE_TYPE
13	<missing>		0B	ARG BUNDLE_TYPE
14	<missing>		0B	ARG BUNDLE_TYPE
15	<missing>		0B	ARG BUNDLE_TYPE
16	<missing>		0B	ARG JAVA_RELEASE

Result

	ID	TAG	SIZE	COMMAND
1	cda5201e70	dumb:latest	0B	CMD ["/bin/sh"
2	74a0788411		512.03MiB	/bin/sh -c cd /
3	7b82eb08ec		67.41MiB	COPY dir:ef251b
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE
5	<missing>		0B	ENV JAVA_HOME="
6	<missing>		0B	ENV LANG=en_US.
7	<missing>		0B	LABEL org.openc
8	<missing>		0B	LABEL org.openc
9	<missing>		0B	LABEL maintaine
10	<missing>		0B	ARG BUNDLE_TYPE
11	<missing>		0B	ARG BUNDLE_TYPE
12	<missing>		0B	ARG BUNDLE_TYPE
13	<missing>		0B	ARG BUNDLE_TYPE
14	<missing>		0B	ARG BUNDLE_TYPE
15	<missing>		0B	ARG BUNDLE_TYPE
16	<missing>		0B	ARG JAVA_RELEASE

Result

	ID	TAG	SIZE	COMMAND
1	cda5201e70	dumb:latest	0B	CMD ["/bin/sh"
2	74a0788411		512.03MiB	/bin/sh -c cd /
3	7b82eb08ec		67.41MiB	COPY dir:ef251b
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE
5	<missing>		0B	ENV JAVA_HOME="
6	<missing>		0B	ENV LANG=en_US.
7	<missing>		0B	LABEL org.openc
8	<missing>		0B	LABEL org.openc
9	<missing>		0B	LABEL maintaine
10	<missing>		0B	ARG BUNDLE_TYPE
11	<missing>		0B	ARG BUNDLE_TYPE
12	<missing>		0B	ARG BUNDLE_TYPE
13	<missing>		0B	ARG BUNDLE_TYPE
14	<missing>		0B	ARG BUNDLE_TYPE
15	<missing>		0B	ARG BUNDLE_TYPE
16	<missing>		0B	ARG JAVA_RELEASE

579.44 MB are changed on every build!

Why do we care? We care because:

1. `push` takes longer time to start
(update is longer)
2. `pull` takes longer
(update takes longer & scaling takes
longer)

Also, more disk pspace is inefficiently used



Optimizing. Round 1.

Let's build it outside of container

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5 ENTRYPOINT java -jar /app/build/libs/spring-petclinic*.jar
```

Optimizing. Round 1.

Let's build it outside of container

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY build/libs/spring-petclinic-3.3.0.jar /app/app.jar
4 CMD java -jar /app/app.jar
```

Optimizing. Round 1.

Let's build it outside of container

```
1 FROM bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl
2
3 COPY build/libs/spring-petclinic-3.3.0.jar /app/app.jar
4 CMD java -jar /app/app.jar
```

Just copying the prebuilt `jar` file

Results

	ID	TAG	SIZE	COMMAND
1	cda5201e70	dumb:latest	0B	CMD ["/bin/sh" "-c" "/app
2	74a0788411		512.03MiB	/bin/sh -c cd /app && ./g
3	7b82eb08ec		67.41MiB	COPY dir:ef251bd1e17ac0af
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE=jdk CDS=n
5	<missing>		0B	ENV JAVA_HOME="/usr/lib/j
6	<missing>		0B	ENV LANG=en_US.UTF-8 LANG
7	<missing>		0B	LABEL org.opencontainers.
8	<missing>		0B	LABEL org.opencontainers.
9	<missing>		0B	LABEL maintainer="\$MAINTA
10	<missing>		0B	NER
11	<missing>		0B	ARG BUNDLE_TYPE CDS DESCRI
12	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_
13	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_
14	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_
15	<missing>		0B	ARG BUNDLE_TYPE JAVA_REL
16	<missing>		0B	ARG JAVA_RELEASE
17	<missing>		7.49MiB	CMD ["/bin/sh"

Results

1	ID	TAG	SIZE	COMMAND
2	cefb30e21d	smarter:latest	0B	CMD ["/bin/sh" "-c" "java
3	46d7594000		58.80MiB	COPY file:8d96bee95ae5ce43
4	e4bcf83d29	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE=jdk CDS=no
5	<missing>		0B	ENV JAVA_HOME="/usr/lib/jv
6	<missing>		0B	ENV LANG=en_US.UTF-8 LANGU
7	<missing>		0B	LABEL org.opencontainers.i
8	<missing>		0B	LABEL org.opencontainers.i
9	<missing>		0B	LABEL maintainer="\$MAINTAI
10	<missing>		0B	ARG BUNDLE_TYPE CDS DESCRI
11	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_R
12	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_R
13	<missing>		0B	ARG BUNDLE_TYPE CDS JAVA_R
14	<missing>		0B	ARG BUNDLE_TYPE JAVA_RELEASE
15	<missing>		0B	ARG JAVA_RELEASE
16	<missing>		7.49MiB	CMD ["/bin/sh"]

Saved 500+ MB

But the build is not clean now :(

Saved 500+ MB

But the build is not clean now :(

We have to build everything outside, what if environment affects the build?

The background image depicts a grand, ornate interior of a theater or opera house. The room is filled with rich, dark wood paneling and gold-colored decorations. A large, ornate chandelier hangs from the ceiling above a grand stage. The stage is framed by heavy, red velvet curtains with gold fringe. Several gold-colored armchairs are arranged on the stage, and a small, ornate sofa sits in front of them. The floor is made of polished wood, and a large, patterned rug lies on the floor in the foreground. The lighting is warm and dramatic, coming from multiple chandeliers and spotlights.

Enter build stages

Multi-stage builds

Holy grail of pure builds

- Allow clean builds
- Allow optimal packaging
- Allow different base images

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Staged example

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
```

Result

	ID	TAG	SIZE	COMMAND
1	6ea7ad11af	smarter:latest	0B	CMD ["/bin/sh" "-c"]
2	d6eb71a5df		58.80MiB	COPY file:8d96beef
3	c6ede8dac2	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE=jdk
4	<missing>		0B	ENV JAVA_HOME="/usr
5	<missing>		0B	ENV LANG=en_US.UTF
6	<missing>		0B	LABEL org.opencont
7	<missing>		0B	LABEL org.opencont
8	<missing>		0B	LABEL maintainer=''
9	<missing>		0B	ARG BUNDLE_TYPE CD
10	<missing>		0B	ARG BUNDLE_TYPE CD
11	<missing>		0B	ARG BUNDLE_TYPE CD
12	<missing>		0B	ARG BUNDLE_TYPE CD
13	<missing>		0B	ARG BUNDLE_TYPE CD
14	<missing>		0B	ARG BUNDLE_TYPE JA
15	<missing>		0B	ARG JAVA_RELEASE
16	<missing>		7.49MiB	CMD ["/bin/sh"]

Result

1	ID	TAG	SIZE	COMMAND
2	6ea7ad11af	smarter:latest	0B	CMD ["/bin/sh" "-c"]
3	d6eb71a5df		58.80MiB	COPY file:8d96beef
4	c6ede8dac2	bellsoft/liberica-runtime-container:jdk-21.0.3_10-musl	98.32MiB	15 BUNDLE_TYPE=jdk
5	<missing>		0B	ENV JAVA_HOME="/usr
6	<missing>		0B	ENV LANG=en_US.UTF
7	<missing>		0B	LABEL org.opencont
8	<missing>		0B	LABEL org.opencont
9	<missing>		0B	LABEL maintainer=''
10	<missing>		0B	ARG BUNDLE_TYPE CD
11	<missing>		0B	ARG BUNDLE_TYPE CD
12	<missing>		0B	ARG BUNDLE_TYPE CD
13	<missing>		0B	ARG BUNDLE_TYPE CD
14	<missing>		0B	ARG BUNDLE_TYPE JA
15	<missing>		0B	ARG JAVA_RELEASE
16	<missing>		7.49MiB	CMD ["/bin/sh"]

That's all folks!

That's all folks!

Or is it?

What's these 59 MiB?

We have to pull 59 MiB of petclinic on every small commit!

Is there a way to optimize it?

Layers!

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
```

- Build image

Layers!

```
1  FROM bellsoft/liberica-runtime-container:jre-slim-musl as builder
2
3  COPY . /app
4  RUN cd /app && ./gradlew build -xtest
5
6  FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8  COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9  WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
```

- Build image
- Introduce new "optimizer" stage

Layers!

```
4 RUN cd /app/ && ./gradlew build -x test
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14 ENTRYPOINT ["java", "-jar", "springframework-boot-launcher.jar"]
```

- Build image
- Introduce new "optimizer" stage

Layers!

```
5
6   FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8   COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9   WORKDIR /app
10  RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12  FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14  ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15  COPY --from=optimizer /app/app/dependencies/ /
```

- Build image
- Introduce new "optimizer" stage
- Extract the jar to layered structure

Layers!

```
/  
8  COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar  
9  WORKDIR /app  
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher  
11  
12 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner  
13  
14 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]  
15 COPY --from=optimizer /app/app/dependencies/ ./  
16 COPY --from=optimizer /app/app/spring-boot-loader/ ./  
17 COPY --from=optimizer /app/app/snapshot_dependencies/ ./
```

- Build image
- Introduce new "optimizer" stage
- Extract the jar to layered structure

Layers!

```
❯ WORKDIR /app
10  RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12  FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14  ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15  COPY --from=optimizer /app/app/dependencies/ ./
16  COPY --from=optimizer /app/app/spring-boot-loader/ ./
17  COPY --from=optimizer /app/app/snapshot-dependencies/ ./
18  COPY --from=optimizer /app/app/application/ ./
```

- Build image
- Introduce new "optimizer" stage
- Extract the jar to layered structure
- Copy layers

Layers!

```
7   WORKDIR /app
10  RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12  FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14  ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15  COPY --from=optimizer /app/app/dependencies/ ./
16  COPY --from=optimizer /app/app/spring-boot-loader/ ./
17  COPY --from=optimizer /app/app/snapshot-dependencies/ ./
18  COPY --from=optimizer /app/app/application/ ./
```

- Build image
- Introduce new "optimizer" stage
- Extract the jar to layered structure
- Copy layers

Layers

Because Spring Boot jar is complex!

```
1 Usage:  
2     java -Djarmode=tools -jar my-app.jar  
3  
4 Available commands:  
5     extract      Extract the contents from the jar  
6     list-layers  List layers from the jar that can be extracted
```

Layers

```
1 app
2   └── application
3     |   └── BOOT-INF
4     |     └── classes
5     |       ├── application-mysql.properties
6     |       ├── application-postgres.properties
7     |       ├── application.properties
8     |       ├── banner.txt
9     |       └── db
10    |         └── ...
11    |           └── org
```

Layers

```
1 app
2   └── application
3     |   └── BOOT-INF
4     |     └── classes
5     |       └── application-mysql.properties
6     |       └── application-postgres.properties
7     |       └── application.properties
8     |       └── banner.txt
9     |       └── db
10    |       |   └── ...
11    |       └── org
```

Layers

```
2   |   └── application
3   |       |   └── BOOT-INF
4   |       |       |   └── classes
5   |       |       |       └── application-mysql.properties
6   |       |       |       └── application-postgres.properties
7   |       |       |       └── application.properties
8   |       |       |       └── banner.txt
9   |       |       |       └── db
10  |       |       |           |   └── ...
11  |       |       |       └── org
12  |       |       |               └── springframework
```

Layers

```
9   |   |   |   └── db
10  |   |   |       └── ...
11  |   |   └── org
12  |   |       └── springframework
13  |   |           └── aop
14  |   |               └── aspectj
15  |   |                   └── annotation
16  |   |                       └── AnnotationAwareAspectJAutoProxyCreator__BeanDefinitions
17  |   |                           ├── com/github/asm0dey/ ...
18  |   |                           └── static
19  |   |       ....
```

Layers

```
12 | | | | └ springframework
13 | | | |   └ aop
14 | | | |     └ aspectj
15 | | | |       └ annotation
16 | | | |         └ AnnotationAwareAspectJAutoProxyCreator__BeanDefinitions
17 | | | |           └ com/github/asm0dey/ ...
18 | | | └ static
19 | | | ....
20 | | └ classpath.idx
21 | └ layers.idx
22 └ META-INF
```

Layers

```
16 | | | | |           └ AnnotationAwareAspectJAutoProxyCreator__BeanDefinitions  
17 | | | | └ com/github/asm0dey/ ...  
18 | | | └ static  
19 | | | ....  
20 | | └ classpath.idx  
21 | | └ layers.idx  
22 └ └ META-INF  
23 |   └ MANIFEST.MF  
24 |   └ native-image  
25 |     └ ch.qos.logback  
26 |       └ logback-classic
```

Layers

```
17 |   |   |   |   ┌── com/github/asm0dey/ ...
18 |   |   |   |   └── static
19 |   |   |   ....
20 |   |   └── classpath.idx
21 |   |   └── layers.idx
22 |   └── META-INF
23 |       ├── MANIFEST.MF
24 |       ├── native-image
25 |       |   └── ch.qos.logback
26 |       |   └── logback-classic
27 |           └── 1.5.6
```

Layers

```
22 |   └── META-INF
23 |     ├── MANIFEST.MF
24 |     ├── native-image
25 |     |   ├── ch.qos.logback
26 |     |   |   └── logback-classic
27 |     |   |       └── 1.5.6
28 |     |   |           ├── reflect-config.json
29 |     |   |           └── resource-config.json
30 |     |   └── ...
31 |     └── sbom
32 |         └── application.cdx.json
```

Layers

```
27 |           |           └── 1.5.6
28 |           |               ├── reflect-config.json
29 |           |               └── resource-config.json
30 |           |               ...
31 |           └── sbom
32 |               └── application.cdx.json
33 |           └── services
34 |               └── java.nio.file.spi.FileSystemProvider
35 └── dependencies
36     └── BOOT-INF
37         └── lib
```

Layers

```
30      |      |      └ ...
31      |      └ sbom
32      |          └ application.cdx.json
33      └ services
34          └ java.nio.file.spi.FileSystemProvider
35 └ dependencies
36     └ BOOT-INF
37         └ lib
38             └ angus-activation-2.0.2.jar
39                 └ ...
40 └ snapshot-dependencies
```

Layers

```
33      └── services
34          └── java.nio.file.spi.FileSystemProvider
35 └── dependencies
36     └── BOOT-INF
37         └── lib
38             ├── angus-activation-2.0.2.jar
39             ├── ...
40     └── snapshot-dependencies
41     └── spring-boot-loader
42         └── org
43             └── springframework
```

Layers

```
35 └── dependencies
36   └── BOOT-INF
37     └── lib
38       ├── angus-activation-2.0.2.jar
39       └── ...
40 └── snapshot-dependencies
41   └── spring-boot-loader
42     └── org
43       └── springframework
44         └── boot
45           └── loader
```

Layers

```
36 |   └── BOOT-INF
37 |     └── lib
38 |       ├── angus-activation-2.0.2.jar
39 |       ├── ...
40 |   └── snapshot-dependencies
41 |   └── spring-boot-loader
42 |     └── org
43 |       └── springframework
44 |         └── boot
45 |           └── loader
46 |             └── jar
```

Layers

```
145     └── ZipContent$Loader.class  
146     └── ZipContent$Source.class  
147     └── ZipContent.class  
148     └── ZipDataDescriptorRecord.class  
149     └── ZipEndOfCentralDirectoryRecord$Located.class  
150     └── ZipEndOfCentralDirectoryRecord.class  
151     └── ZipLocalFileHeaderRecord.class  
152     └── ZipString$CompareType.class  
153         └── ZipString.class  
154  
155 212 directories, 525 files
```

Layered image structure

1	ID	TAG	SIZE	COMMAND
2	618743f6a2	layers:latest	2.96MiB	COPY dir:e0faa63b96547
3	9cca3273f0		0B	COPY dir:acd0d0ac1f7d1
4	fe123ee16e		382.56kiB	COPY dir:01225d3c4ef6c
5	dede9bac3d		57.69MiB	COPY dir:755815d928fd9
6	65e3fb4cbf		0B	ENTRYPOINT ["java" "or
7	7130fa5864	bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTION
8	<missing>		0B	ENV JAVA_HOME="/usr/li
9	<missing>		0B	ENV LANG=en_US.UTF-8 L
10	<missing>		0B	LABEL org.opencontainers

Layered image structure

1	ID	TAG	SIZE	COMMAND
2	618743f6a2	layers:latest	2.96MiB	COPY dir:e0faa63b96547
3	9cca3273f0		0B	COPY dir:acd0d0ac1f7d1
4	fe123ee16e		382.56kiB	COPY dir:01225d3c4ef6c
5	dede9bac3d		57.69MiB	COPY dir:755815d928fd9
6	65e3fb4cbf		0B	ENTRYPOINT ["java" "or
7	7130fa5864	bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTION
8	<missing>		0B	ENV JAVA_HOME="/usr/li
9	<missing>		0B	ENV LANG=en_US.UTF-8 L
10	<missing>		0B	LABEL org.opencontainers

- Dependencies: ~57.7 MiB

Layered image structure

1	ID	TAG	SIZE	COMMAND
2	618743f6a2	layers:latest	2.96MiB	COPY dir:e0faa63b96547
3	9cca3273f0		0B	COPY dir:acd0d0ac1f7df
4	fe123ee16e		382.56kiB	COPY dir:01225d3c4ef6c
5	dede9bac3d		57.69MiB	COPY dir:755815d928fd9
6	65e3fb4cbf		0B	ENTRYPOINT ["java" "or
7	7130fa5864	bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTION
8	<missing>		0B	ENV JAVA_HOME="/usr/li
9	<missing>		0B	ENV LANG=en_US.UTF-8 L
10	<missing>		0B	LABEL org.opencontainers

- Dependencies: ~57.7 MiB
- Launcher: 382.56kiB

Layered image structure

1	ID	TAG	SIZE	COMMAND
2	618743f6a2	layers:latest	2.96MiB	COPY dir:e0faa63b96547
3	9cca3273f0		0B	COPY dir:acd0d0ac1f7d1
4	fe123ee16e		382.56kiB	COPY dir:01225d3c4ef6c
5	dede9bac3d		57.69MiB	COPY dir:755815d928fd9
6	65e3fb4cbf		0B	ENTRYPOINT ["java" "or
7	7130fa5864	bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTION
8	<missing>		0B	ENV JAVA_HOME="/usr/li
9	<missing>		0B	ENV LANG=en_US.UTF-8 L
10	<missing>		0B	LABEL org.opencontainers

- Dependencies: ~57.7 MiB
- Launcher: 382.56kiB

Layered image structure

1	ID	TAG	SIZE	COMMAND
2	618743f6a2	layers:latest	2.96MiB	COPY dir:e0faa63b96547
3	9cca3273f0		0B	COPY dir:acd0d0ac1f7df
4	fe123ee16e		382.56kiB	COPY dir:01225d3c4ef6c
5	dede9bac3d		57.69MiB	COPY dir:755815d928fd9
6	65e3fb4cbf		0B	ENTRYPOINT ["java" "or
7	7130fa5864	bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTION
8	<missing>		0B	ENV JAVA_HOME="/usr/li
9	<missing>		0B	ENV LANG=en_US.UTF-8 L
10	<missing>		0B	LABEL org.opencontainers

- Dependencies: ~57.7 MiB
- Launcher: 382.56kiB
- Application: ~3MiB

Layered image structure

ID	TAG	SIZE	COMMAND
1	618743f6a2 layers:latest	2.96MiB	COPY dir:e0faa63b9654
2	9cca3273f0	0B	COPY dir:acd0d0ac1f7c
3	fe123ee16e	382.56kiB	COPY dir:01225d3c4ef6
4	dede9bac3d	57.69MiB	COPY dir:755815d928fc
5	65e3fb4cbf	0B	ENTRYPOINT ["java" "j
6	7130fa5864 bellsoft/liberica-runtime-container:jre-slim-musl	126.80MiB	18 CDS=no DESCRIPTIO
7	<missing>	0B	ENV JAVA_HOME="/usr/t
8	<missing>	0B	ENV LANG=en_US.UTF-8
9	<missing>	0B	LABEL org.opencontai
10	<missing>	0B	

- Dependencies: ~57.7 MiB
- Launcher: 382.56kiB
- Application: ~3MiB

Together: ~61MiB

61.0MiB > 58.8Mib!



What are we optimizing?

Pull size!

Pull size here is usually only around 3MiB!

We just reinvented how the BellSoft's buildpack works!

And it is amazing

Diversion: buildpacks

<https://paketo.io/>

Trivial usage:

```
1 /usr/sbin/pack build petclinic \
2   --builder bellsoft/buildpacks.builder \
3   --path . \
4   -e BP_JVM_VERSION=21
```

Diversion: buildpacks

<https://paketo.io/>

Trivial usage:

```
1 /usr/sbin/pack build petclinic \
2   --builder bellsoft/buildpacks.builder \
3   --path . \
4   -e BP_JVM_VERSION=21
```

Diversion: buildpacks

<https://paketo.io/>

Trivial usage:

```
1 /usr/sbin/pack build petclinic \
2   --builder bellsoft/buildpacks.builder \
3   --path . \
4   -e BP_JVM_VERSION=21
```

Diversion: buildpacks

<https://paketo.io/>

Trivial usage:

```
1 /usr/sbin/pack build petclinic \
2   --builder bellsoft/buildpacks.builder \
3   --path . \
4   -e BP_JVM_VERSION=21
```

Result

	ID	TAG	SIZE	COMMAND
1	2774e3f214	petclinic:latest	0B	Buildpacks Process Types
2	<missing>		1.44kiB	Buildpacks Launcher Config
3	<missing>		2.44MiB	Buildpacks Application Launcher
4	<missing>		59.17MiB	Application Layer
5	<missing>		729.21kiB	Software Bill-of-Materials
6	<missing>		97.87MiB	Layer: 'jre', Created by buildpack: bellsoft/buildpack
7	<missing>		200.00B	Layer: 'java-security-properties', Created by buildpac
8	<missing>		4.27MiB	Layer: 'helper', Created by buildpack: bellsoft/buildp
9	<missing>		2.97kiB	
10	<missing>		7.48MiB	
11	<missing>			

Result

	ID	TAG	SIZE	COMMAND
1	2774e3f214	petclinic:latest	0B	Buildpacks Process Types
2	<missing>		1.44kiB	Buildpacks Launcher Config
3	<missing>		2.44MiB	Buildpacks Application Launcher
4	<missing>		59.17MiB	Application Layer
5	<missing>		729.21kiB	Software Bill-of-Materials
6	<missing>		97.87MiB	Layer: 'jre', Created by buildpack: bellsoft/buildpack
7	<missing>		200.00B	Layer: 'java-security-properties', Created by buildpac
8	<missing>		4.27MiB	Layer: 'helper', Created by buildpack: bellsoft/buildp
9	<missing>		2.97kiB	
10	<missing>		7.48MiB	
11	<missing>			

Result

	ID	TAG	SIZE	COMMAND
1	2774e3f214	petclinic:latest	0B	Buildpacks Process Types
2	<missing>		1.44kiB	Buildpacks Launcher Config
3	<missing>		2.44MiB	Buildpacks Application Launcher
4	<missing>		59.17MiB	Application Layer
5	<missing>		729.21kiB	Software Bill-of-Materials
6	<missing>		97.87MiB	Layer: 'jre', Created by buildpack: bellsoft/buildpack
7	<missing>		200.00B	Layer: 'java-security-properties', Created by buildpac
8	<missing>		4.27MiB	Layer: 'helper', Created by buildpack: bellsoft/buildp
9	<missing>		2.97kiB	
10	<missing>		7.48MiB	
11	<missing>			

Result

1	ID	TAG	SIZE	COMMAND
2	2774e3f214	petclinic:latest	0B	Buildpacks Process Types
3	<missing>		1.44kiB	Buildpacks Launcher Config
4	<missing>		2.44MiB	Buildpacks Application Launcher
5	<missing>		59.17MiB	Application Layer
6	<missing>		729.21kiB	Software Bill-of-Materials
7	<missing>		97.87MiB	Layer: 'jre', Created by buildpack: bellsoft/buildpack
8	<missing>		200.00B	Layer: 'java-security-properties', Created by buildpac
9	<missing>		4.27MiB	Layer: 'helper', Created by buildpack: bellsoft/buildp
10	<missing>		2.97kiB	
11	<missing>		7.48MiB	

Buildpacks

Why do we need them?

- Buildpacks transform your application source code into container images
- The Paketo open source project provides production-ready buildpacks for the most popular languages and frameworks
- Use Paketo Buildpacks to easily build your apps and keep them updated
- Reminds of s2i images
- Build better than default
- Spring-aware

Is this all?

We've optimized pull/push size, right?

Optimization is multidimensional

When startup time is more important...

There are several solutions for the Spring application startup time

1. Class Data Sharing (CDS)
2. Coordinated Restore at Checkpoint
3. Native Image

Class Data Sharing (CDS)

- When: Java 5 (!)
- Why: reduce the startup and memory footprint of multiple JVM instances running on the same host
- How: stores data in an archive

How does it help us?

Class Data Sharing (CDS)

- When: Java 5 (!)
- Why: reduce the startup and memory footprint of multiple JVM instances running on the same host
- How: stores data in an archive

How does it help us?

It does not! 

Class Data Sharing (CDS)

- When: Java 5 (!)
- Why: reduce the startup and memory footprint of multiple JVM instances running on the same host
- How: stores data in an archive

How does it help us?

It does not!  But

Application Class Data Sharing

- When: Java 10
- Why: add application classes to the archive

And then:

- When: Java 13, JEP 350
- Why: allow addition of classes into the archive upon app exit!

And this helps! How?

How to use AppCDS with Spring?

- `-XX:ArchiveClassesAtExit=application.jsa` to create an archive
- `-Dspring.context.exit=onRefresh` to start and immediately exit the application

NB:

1. Use the same JDK
2. Use the same arguments

And even better!

Spring AOT

Add `-Dspring.aot.enabled=true`

Even more classes!!!

Practice

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15 COPY --from=optimizer /app/app/dependencies/ ./
16 COPY --from=optimizer /app/app/spring-boot-loader/ ./
17 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
18 COPY --from=optimizer /app/app/application/ ./
```

Practice

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12 FROM bellsoft/liberica-runtime-container:jre-slim-musl as runner
13
14 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15 COPY --from=optimizer /app/app/dependencies/ ../
16 COPY --from=optimizer /app/app/spring-boot-loader/ ../
17 COPY --from=optimizer /app/app/snapshot-dependencies/ ../
18 COPY --from=optimizer /app/app/application/ ../
```

Practice

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
13
14 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15 COPY --from=optimizer /app/app/dependencies/ ../
16 COPY --from=optimizer /app/app/spring-boot-loader/ ../
17 COPY --from=optimizer /app/app/snapshot-dependencies/ ../
18 COPY --from=optimizer /app/app/application/ ../
```

Practice

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-slim-musl as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Djarmode=tools -jar /app/app.jar extract --layers --launcher
11
12 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
13
14 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
15 COPY --from=optimizer /app/app/dependencies/ ./
16 COPY --from=optimizer /app/app/spring-boot-loader/ ./
17 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
18 COPY --from=optimizer /app/app/application/ ./
```

Practice

```
1 #omitted
2 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
3
4 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
5 COPY --from=optimizer /app/app/dependencies/ ./
6 COPY --from=optimizer /app/app/spring-boot-loader/ ./
7 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
8 COPY --from=optimizer /app/app/application/ ./
9 RUN java -Dspring.aot.enabled=true \
10   -XX:ArchiveClassesAtExit=./application.jsa \
11   -Dspring.context.exit=onRefresh \
12   org.springframework.boot.loader.launch.JarLauncher
```

Practice

```
1 #omitted
2 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
3
4 ENTRYPOINT ["java", "org.springframework.boot.loader.launch.JarLauncher"]
5 COPY --from=optimizer /app/app/dependencies/ ./
6 COPY --from=optimizer /app/app/spring-boot-loader/ ./
7 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
8 COPY --from=optimizer /app/app/application/ ./
9 RUN java -Dspring.aot.enabled=true \
10   -XX:ArchiveClassesAtExit=./application.jsa \
11   -Dspring.context.exit=onRefresh \
12   org.springframework.boot.loader.launch.JarLauncher
```

Practice

```
1 #omitted
2 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
3
4 ENTRYPOINT ["java", \
5             "-Dspring.aot.enabled=true", \
6             "-XX:SharedArchiveFile=application.jsa", \
7             "org.springframework.boot.loader.launch.JarLauncher"]
8 COPY --from=optimizer /app/app/dependencies / ./
9 COPY --from=optimizer /app/app/spring-boot-loader/ ./
10 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
11 COPY --from=optimizer /app/app/application/ ./
12 RUN java -Dspring.aot.enabled=true \
13     -XX:ArchiveClassesAtExit=../application.jsa \
14     -Dspring.context.exit=onRefresh \
15     org.springframework.boot.loader.launch.JarLauncher
```

Practice

```
1 #omitted
2 FROM bellsoft/liberica-runtime-container:jre-cds-slim-musl as runner
3
4 ENTRYPOINT ["java", \
5             "-Dspring.aot.enabled=true", \
6             "-XX:SharedArchiveFile=application.jsa", \
7             "org.springframework.boot.loader.launch.JarLauncher"]
8 COPY --from=optimizer /app/app/dependencies / ./
9 COPY --from=optimizer /app/app/spring-boot-loader/ ./
10 COPY --from=optimizer /app/app/snapshot-dependencies/ ./
11 COPY --from=optimizer /app/app/application/ ./
12 RUN java -Dspring.aot.enabled=true \
13     -XX:ArchiveClassesAtExit=./application.jsa \
14     -Dspring.context.exit=onRefresh \
15     org.springframework.boot.loader.launch.JarLauncher
```

What does it cost ?

```
1 -r-- r-- r-- 0:0 81 MB ┊ application.jsa
```

Which is not small at all!

What does it cost ?

```
1 -r-- r-- r-- 0:0 81 MB — application.jsa
```

Which is not small at all!

But you're trading pull speed for startup speed

How much faster?

Up to 50-60%

Pushing further with CRaC

CRaC: Coordinated Restore at Checkpoint

The CRaC (Coordinated Restore at Checkpoint) Project researches coordination of Java programs with mechanisms to checkpoint (make an image of, snapshot) a Java instance while it is executing. Restoring from the image could be a solution to some of the problems with the start-up and warm-up times. The primary aim of the Project is to develop a new standard mechanism-agnostic API to notify Java programs about the checkpoint and restore events. Other research activities will include, but will not be limited to, integration with existing checkpoint/restore mechanisms and development of new ones, changes to JVM and JDK to make images smaller and ensure they are correct.

<https://openjdk.org/projects/crac/>

In a perfect world it should be:

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5
6 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
11
12 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
13
14 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
15 COPY --from=optimizer /app/app.jar /app/app.jar
16 COPY --from=optimizer /checkpoint /checkpoint
```

In a perfect world it should be:

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5
6 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
11
12 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
13
14 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
15 COPY --from=optimizer /app/app.jar /app/app.jar
16 COPY --from=optimizer /checkpoint /checkpoint
```

In a perfect world it should be:

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build
5
6 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
11
12 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
13
14 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
15 COPY --from=optimizer /app/app.jar /app/app.jar
16 COPY --from=optimizer /checkpoint /checkpoint
```

But in reality

```
1 #12 6.051      Suppressed: java.lang.RuntimeException: Native checkpoint failed.  
2 #12 6.051          at java.base/jdk/cr.ac.Core.translateJVMExceptions(Core.java:114) ~[r  
3 #12 6.051          at java.base/jdk/cr.ac.Core.checkpointRestore1(Core.java:192) ~[na:na:  
4 #12 6.051          at java.base/jdk/cr.ac.Core.checkpointRestore(Core.java:299) ~[na:na:  
5 #12 6.051          at java.base/jdk/cr.ac.Core.checkpointRestore(Core.java:278) ~[na:na:  
6 #12 6.051          at java.base/javax/cr.ac.Core.checkpointRestore(Core.java:73) ~[na:na:  
7 #12 6.051          at java.base/jdk/internal/reflect.DirectMethodHandleAccessor.invoke(  
8 #12 6.051          at java.base/java/lang/reflect.Method.invoke(Method.java:580) ~[na:r  
9 #12 6.051          at org.cr.ac.Core$Compat.checkpointRestore(Core.java:141) ~[crac-1.4.  
10 #12 6.051         ... 17 common frames omitted
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
2
3 COPY . /app
4 RUN cd /app && ./gradlew build -xtest
5
6 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
7
8 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
9 WORKDIR /app
10 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
11
12 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
13
14 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
15 COPY --from=optimizer /app/app.jar /app/app.jar
16 COPY --from=optimizer /checkpoint /checkpoint
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 # syntax=docker/dockerfile:1-labs
2 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
3
4 COPY . /app
5 RUN cd /app && ./gradlew build -xtest
6
7 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
8
9 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
10 WORKDIR /app
11 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
12
13 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
14
15 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
16 COPY --from=optimizer /app/app.jar /app/app.jar
17 COPY --from=optimizer /checkpoint /checkpoint
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 # syntax=docker/dockerfile:1-labs
2 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
3
4 COPY . /app
5 RUN cd /app && ./gradlew build -xtest
6
7 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
8
9 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
10 WORKDIR /app
11 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
12
13 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
14
15 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
16 COPY --from=optimizer /app/app.jar /app/app.jar
17 COPY --from=optimizer /checkpoint /checkpoint
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 # syntax=docker/dockerfile:1-labs
2 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
3
4 COPY . /app
5 RUN cd /app && ./gradlew build -xtest
6
7 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
8
9 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
10 WORKDIR /app
11 RUN java -Dspring.context.checkpoint=onRefresh -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
12
13 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
14
15 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
16 COPY --from=optimizer /app/app.jar /app/app.jar
17 COPY --from=optimizer /checkpoint /checkpoint
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 # syntax=docker/dockerfile:1-labs
2 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
3
4 COPY . /app
5 RUN cd /app && ./gradlew build -xtest
6
7 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
8
9 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
10 WORKDIR /app
11 RUN --security=insecure java -Dspring.context.checkpoint=onRefresh \
12     -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar
13
14 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
15
16 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
```

CRaC is hard!

Let's try to fix it with arcane magic

```
1 # syntax=docker/dockerfile:1-labs
2 FROM bellsoft/liberica-runtime-container:jdk-musl as builder
3
4 COPY . /app
5 RUN cd /app && ./gradlew build -xtest
6
7 FROM bellsoft/liberica-runtime-container:jre-crac-slim as optimizer
8
9 COPY --from=builder /app/build/libs/spring-petclinic-3.3.0.jar /app/app.jar
10 WORKDIR /app
11 RUN --security=insecure java -Dspring.context.checkpoint=onRefresh \
12     -XX:CRaCCheckpointTo=/checkpoint -jar /app/app.jar || true
13
14 FROM bellsoft/liberica-runtime-container:jre-crac-slim as runner
15
16 ENTRYPOINT java -XX:CRaCRestoreFrom=/checkpoint
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx create --buildkitd-flags \  
2     '--allow-insecure-entitlement security.insecure' \  
3     --name insecure-builder
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx create --buildkitd-flags \  
2     '--allow-insecure-entitlement security.insecure' \  
3     --name insecure-builder
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx use insecure-builder
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx build \
2     --allow security.insecure \
3     -f Dockerfile.crac \
4     -t pet-crack --output type=docker .
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx build \
2     --allow security.insecure \
3     -f Dockerfile.crac \
4     -t pet-crack --output type=docker .
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker buildx build \
2     --allow security.insecure \
3     -f Dockerfile.crac \
4     -t pet-crack --output type=docker .
```

And this is not all!

Did you hear about `buildx` ?

```
1 docker run --rm -it --privileged pet-crac
```

And this is not all!

Did you hear about `buildx` ?

- 1 Restarting Spring-managed lifecycle beans after JVM restore
- 2 HikariPool-1 - Thread starvation or clock leap detected (housekeeper delta=1d19h37m42s102ms798μs806ns)
- 3 Tomcat started on port 8080 (http) with context path ''
- 4 Restored PetClinicApplication in 0.186 seconds (process running for 0.19)

And this is not all!

Did you hear about `buildx` ?

- 1 Restarting Spring-managed lifecycle beans after JVM restore
- 2 HikariPool-1 - Thread starvation or clock leap detected (housekeeper delta=1d19h37m42s102ms798μs806ns)
- 3 Tomcat started on port 8080 (http) with context path '/'
- 4 Restored PetClinicApplication in 0.186 seconds (process running for 0.19)

Is it a good solution?

It depends

Is it a good solution?

It depends

- If "It Works" is enough for you

Is it a good solution?

It depends

- If "It Works" is enough for you -> YES
- If you need more predictable and maintainable thing

Is it a good solution?

It depends

- If "It Works" is enough for you -> YES
- If you need more predictable and maintainable thing

Is it a good solution?

It depends

- If "It Works" is enough for you -> YES
- If you need more predictable and maintainable thing -> NO

How to make it better?

1. Build JAR (in docker or not)
- 2.

```
1 docker run -d --name pre_crac -it -p8080:8080 \
2   --cap-add CAP_SYS_PTRACE --cap-add CAP_CHECKPOINT_RESTORE\
3   pre_crac
4 docker exec -it pre_crac jcmd 127 JDK.checkpoint
```

3. Container will run and stop
4. Commit the container like

```
docker commit pre_crac post_crac
```

Pros and Cons

Pros:

1. Does not require arcane magic
2. Works more predictably
3. Does not depend on unstable features
4. Does not require privileged containers in

Cons:

1. Organization-specific
2. Requires more steps

And of course native image!

```
1 FROM bellsoft/liberica-native-image-kit-container:jdk-21-nik-23.1.3-stream-musl as builder
2
3 WORKDIR /app
4 ADD . /app/
5 RUN cd /app && ./gradlew nativeCompile -x test
6
7
8 FROM bellsoft/alpaquita-linux-base:stream-musl
9
10 WORKDIR /app
11 EXPOSE 8081
12 ENTRYPOINT ["/app/petclinic-native"]
13 COPY --from=builder /app/spring-petclinic-main/target/spring-petclinic /app/petclinic-native
```

And of course native image!

```
1  FROM bellsoft/liberica-native-image-kit-container:jdk-21-nik-23.1.3-stream-musl as builder
2
3  WORKDIR /app
4  ADD . /app/
5  RUN cd /app && ./gradlew nativeCompile -x test
6
7
8  FROM bellsoft/alpaquita-linux-base:stream-musl
9
10 WORKDIR /app
11 EXPOSE 8081
12 ENTRYPOINT ["/app/petclinic-native"]
13 COPY --from=builder /app/spring-petclinic-main/target/spring-petclinic /app/petclinic-native
```

Might fail on a huge application

And of course native image!

```
1 FROM bellsoft/liberica-native-image-kit-container:jdk-21-nik-23.1.3-stream-musl as builder
2
3 WORKDIR /app
4 ADD . /app/
5 RUN cd /app && ./gradlew nativeCompile -x test
6
7
8 FROM bellsoft/alpaquita-linux-base:stream-musl
9
10 WORKDIR /app
11 EXPOSE 8081
12 ENTRYPOINT ["/app/petclinic-native"]
13 COPY --from=builder /app/spring-petclinic-main/target/spring-petclinic /app/petclinic-native
```

Might fail on a huge application

And of course native image!

```
1  FROM bellsoft/liberica-native-image-kit-container:jdk-21-nik-23.1.3-stream-musl as builder
2
3  WORKDIR /app
4  ADD . /app/
5  RUN cd /app && ./gradlew nativeCompile -x test
6
7
8  FROM bellsoft/alpaquita-linux-base:stream-musl
9
10 WORKDIR /app
11 EXPOSE 8081
12 ENTRYPOINT ["/app/petclinic-native"]
13 COPY --from=builder /app/spring-petclinic-main/target/spring-petclinic /app/petclinic-native
```

Might fail on a huge application

And of course native image!

```
1  FROM bellsoft/liberica-native-image-kit-container:jdk-21-nik-23.1.3-stream-musl as builder
2
3  WORKDIR /app
4  ADD . /app/
5  RUN cd /app && ./gradlew nativeCompile -x test
6
7
8  FROM bellsoft/alpaquita-linux-base:stream-musl
9
10 WORKDIR /app
11 EXPOSE 8081
12 ENTRYPOINT ["/app/petclinic-native"]
13 COPY --from=builder /app/spring-petclinic-main/target/spring-petclinic /app/petclinic-native
```

Might fail on a huge application

And now we have all
flavours of ultimate
docker images!

Quick summary?

1. Use layers for faster deployment
2. Use CDS for faster startup without many compromises
3. Use CRaC for a *lightning-fast* startup (with compromises)
4. Use Liberica NIK for *lightning-fast* startup (when your application is small-ish and NI-compatible)

Thank you!

Find me @

-  asm0di0
-  @asm0dey@fosstodon.org
-  me@asm0dey.site
-  asm0dey
-  asm0dey
-  asm0dey
-  asm0dey
-  asm0dey



END