Jib 打包 Java 應用的那些事
心得分享與擴充攻略

Matt Ho
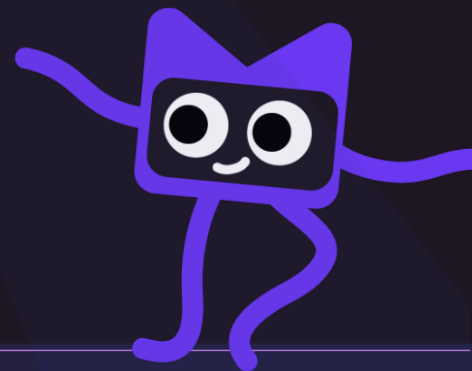
Hi, I'm Matt 👋

- 松凌科技 Architect / R&D

- JCConf Speaker

- in/methodho

- github.com/shihyuho

Lab

github.com/shihyuho/jcconf2024-jib

今天講什麼?

Java Code → build → Container Image → run → Container (Container Runtime)
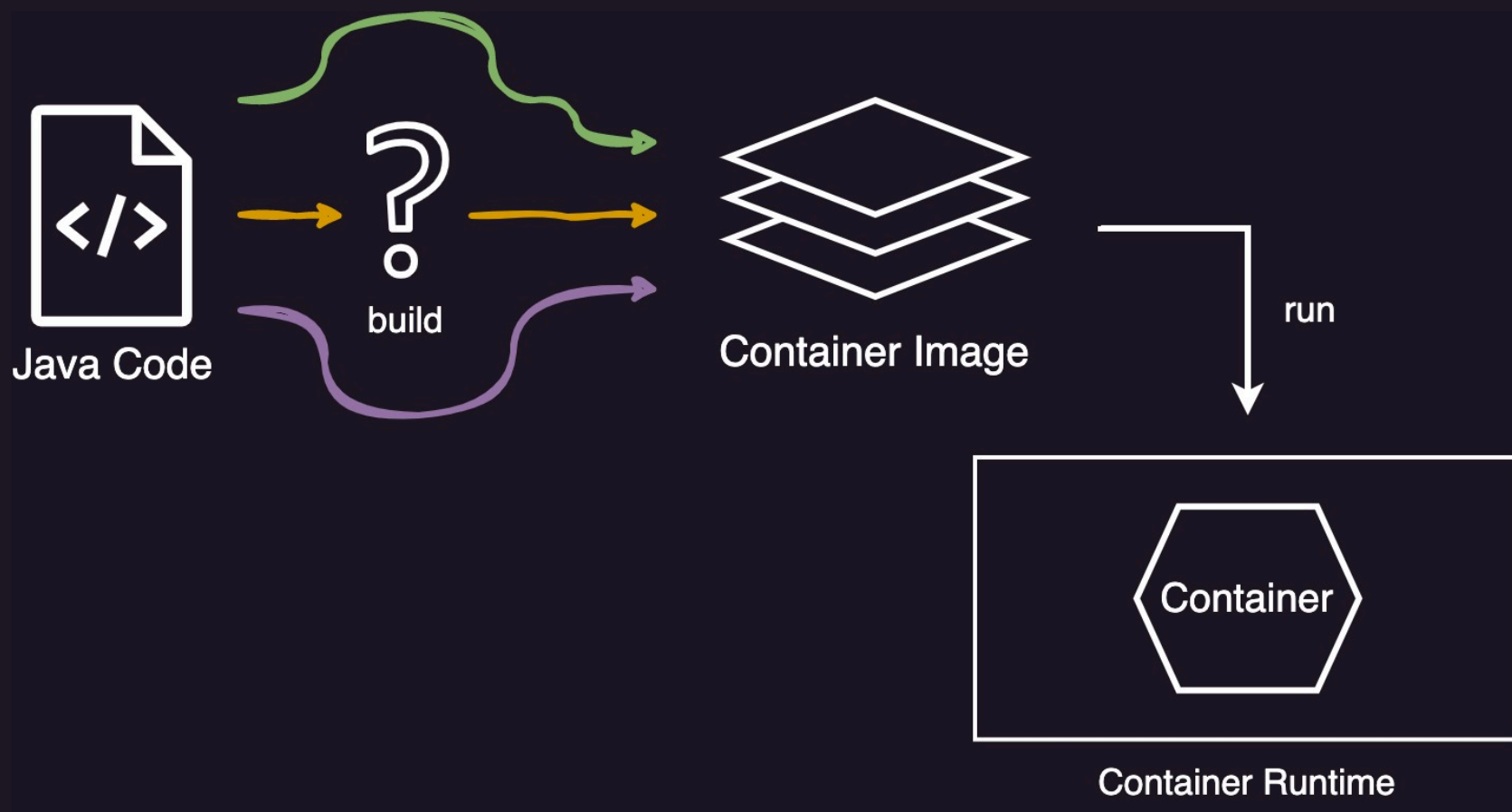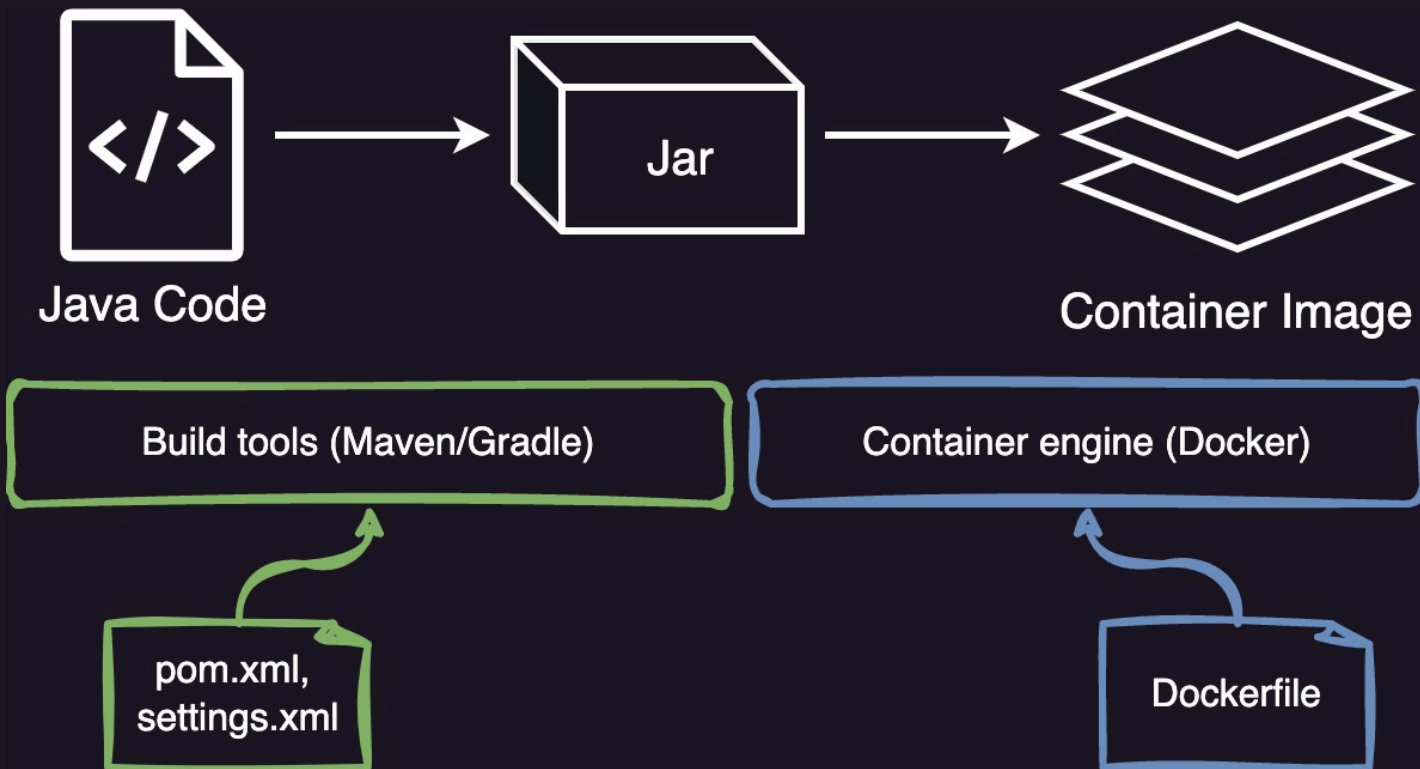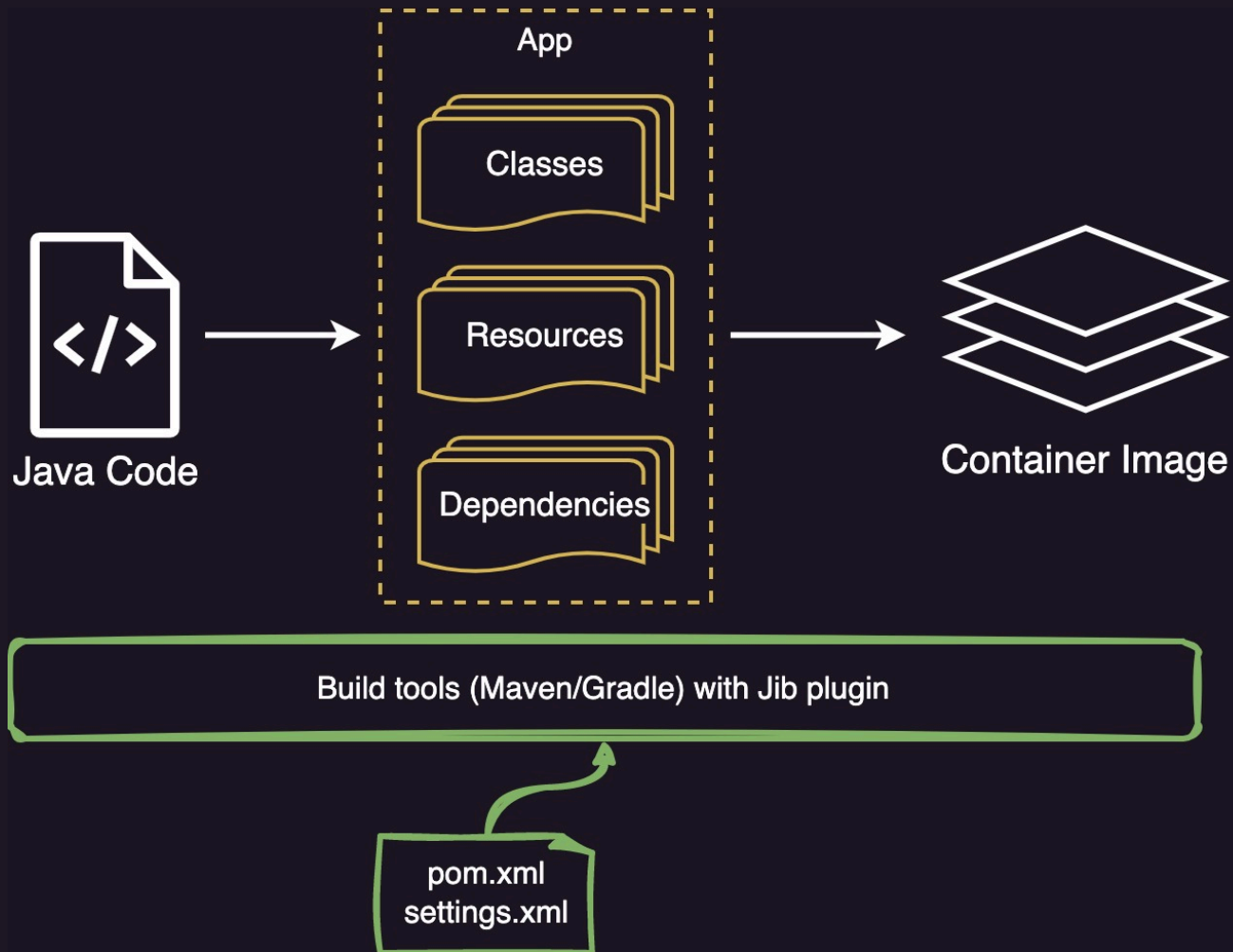
Image 裡有什麼
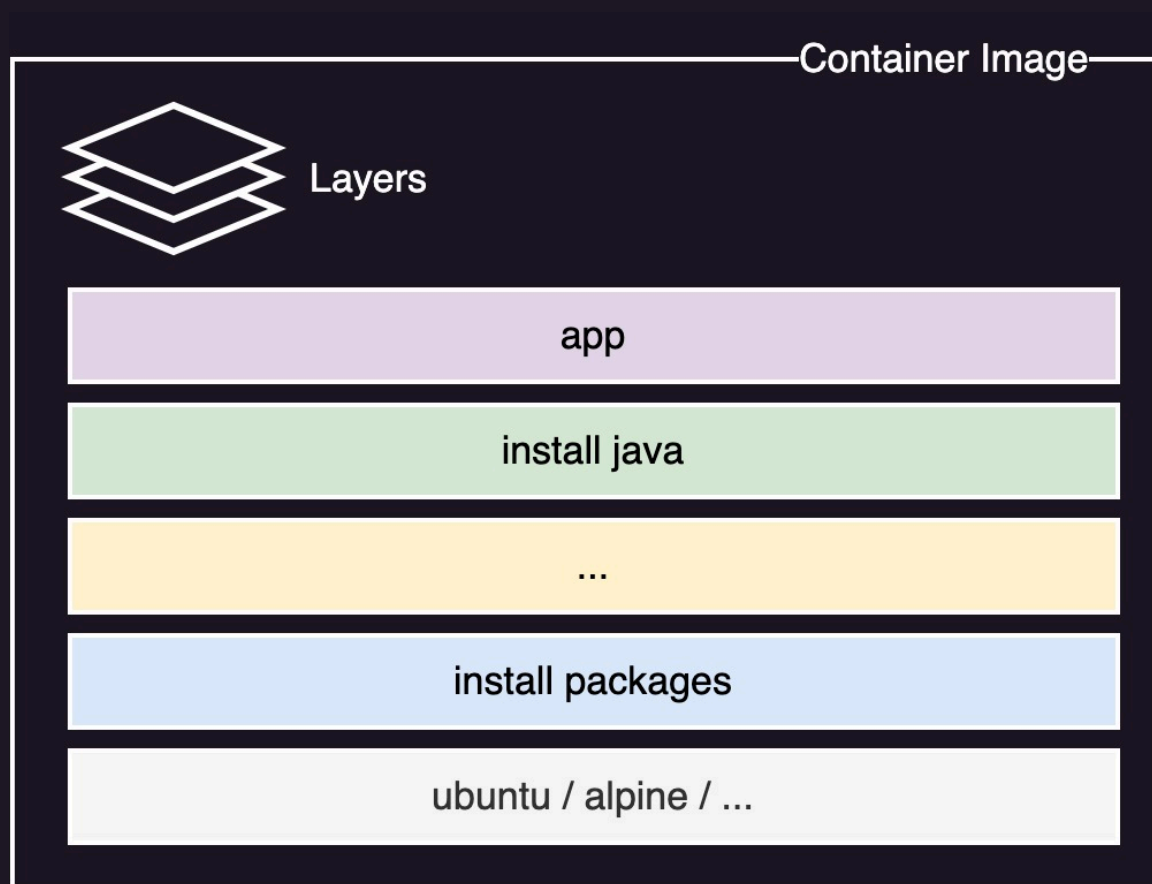
Fat Jar

Fat Jar - Dockerfile

```
FROM eclipse-temurin:21-jre-alpine
COPY app.jar /app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

Fat Jar - Layers


```
FROM eclipse-temurin:21-jre-alpine
COPY app.jar /app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```
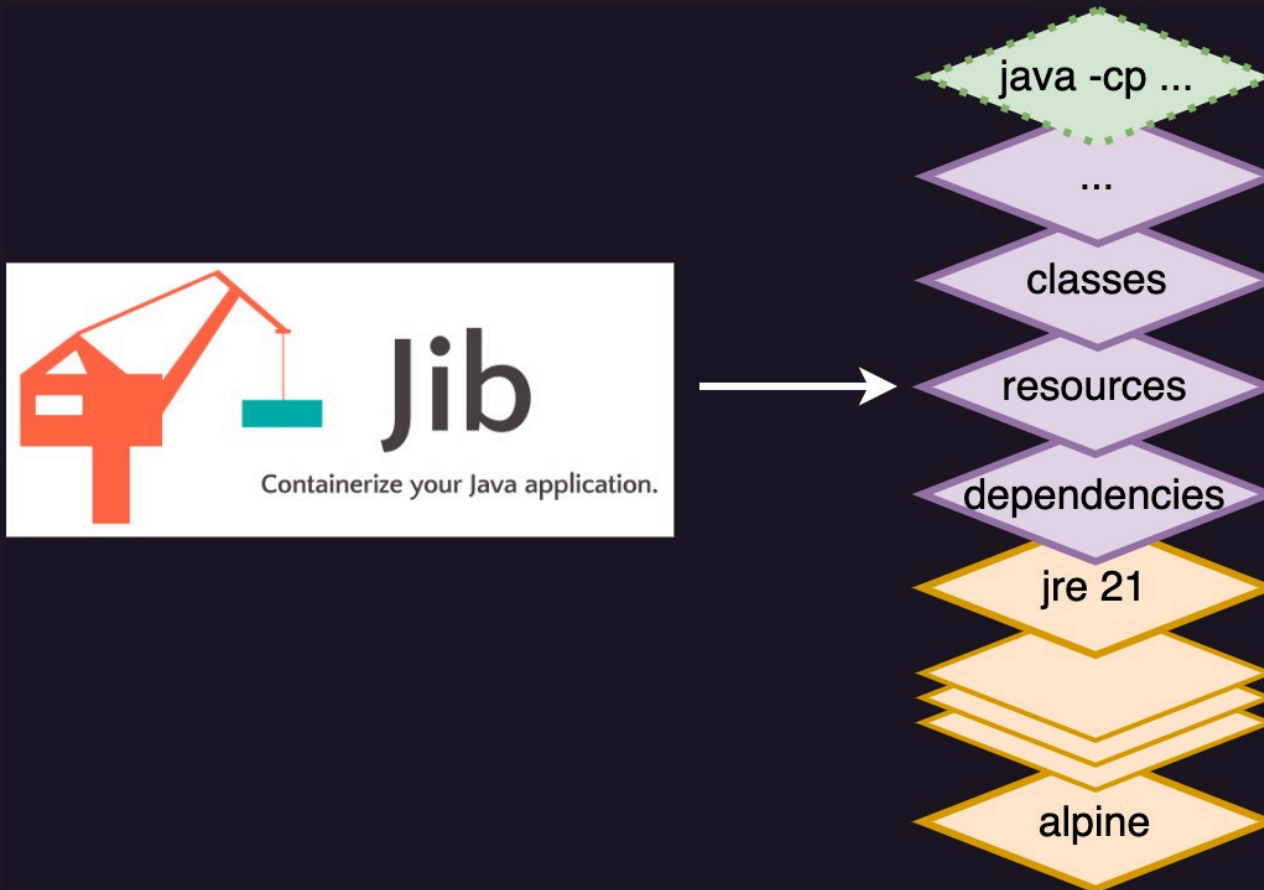
java -jar ...
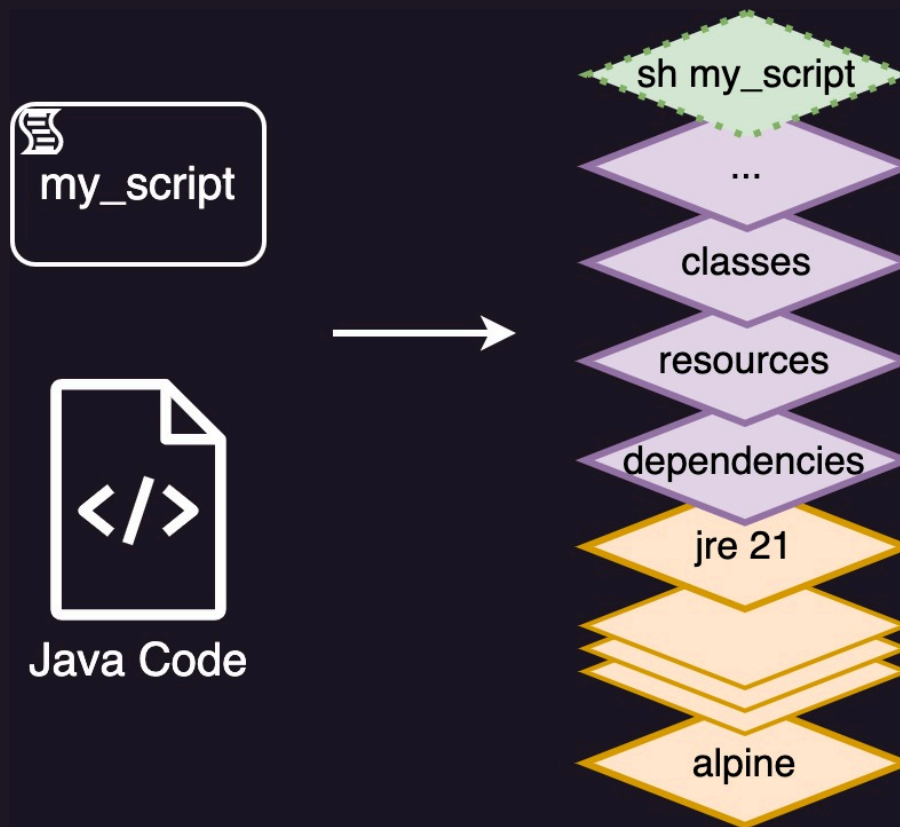
app.jar

jre 21

alpine

Jib

Jib - Layers

喜歡 Jib 的原因

1. Daemonless
2. No more Dockerfiles
3. Well integrated with build tools (Maven/Gradle)
4. Optimized for improved layer caching
5. Extensible (等等~ 這啥?)

有時候我們會需要自定義 entrypoint

原來是 Jib 忽略了 <jvmFlags>

| entrypoint | list | *None* | The command to start the container with (similar to Docker's ENTRYPOINT instruction). If set, then `jvmFlags`, `mainClass`, `extraClasspath`, and `expandClasspathDependencies` are ignored. You may also set `<entrypoint>INHERIT</entrypoint>` (`<entrypoint><entry>INHERIT</entry></entrypoint>` in old Maven versions) to indicate that the `entrypoint` and `args` should be inherited from the base image.* |

Jib 的擴充

softleader/jib-jvm-flags-extension-maven

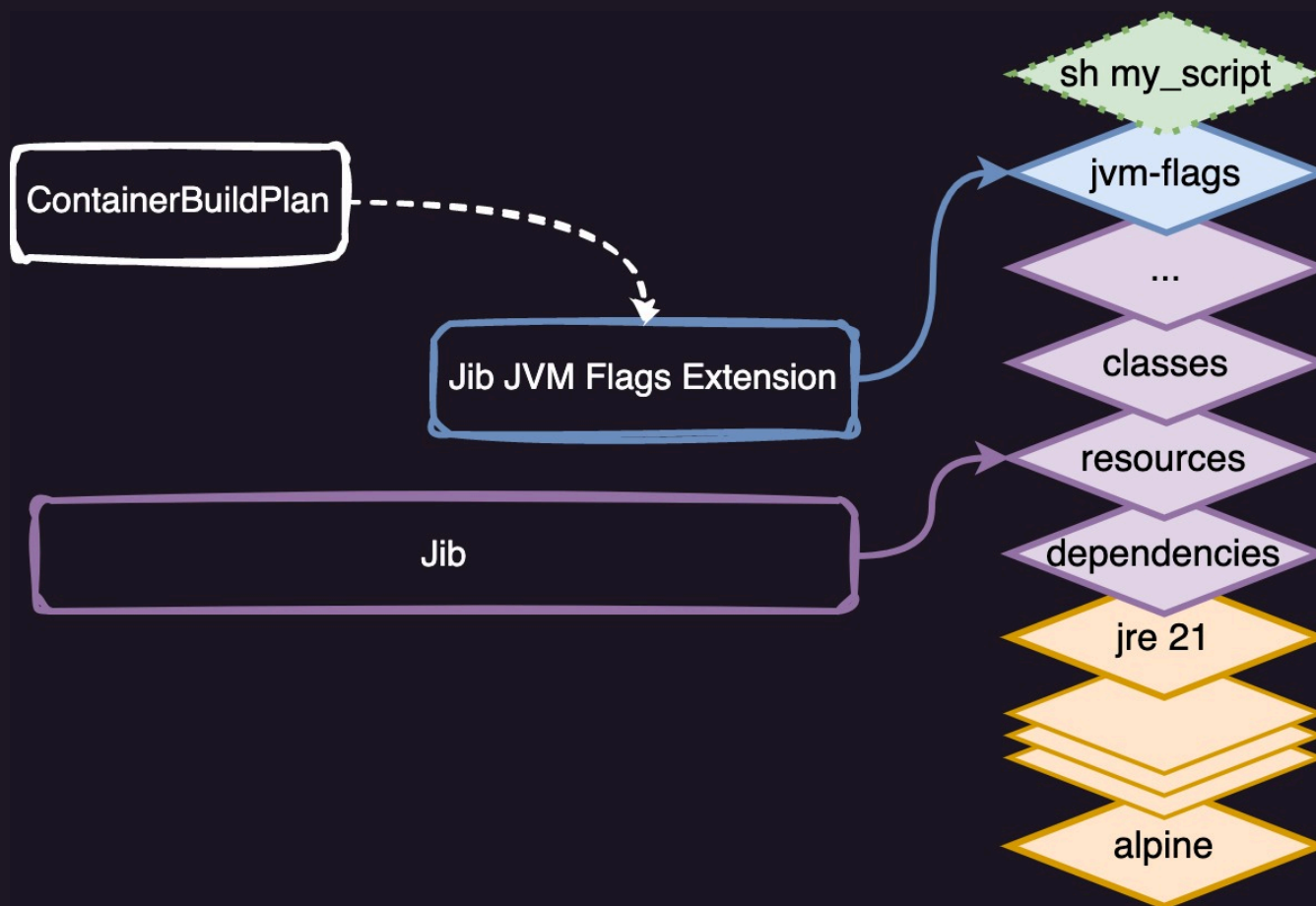A Jib maven extension outputs the configured <jvmFlags> into the /app/jib-jvm-flags-file file, allowing a custom entrypoint to access these flags.

Jib 怎麼擴充
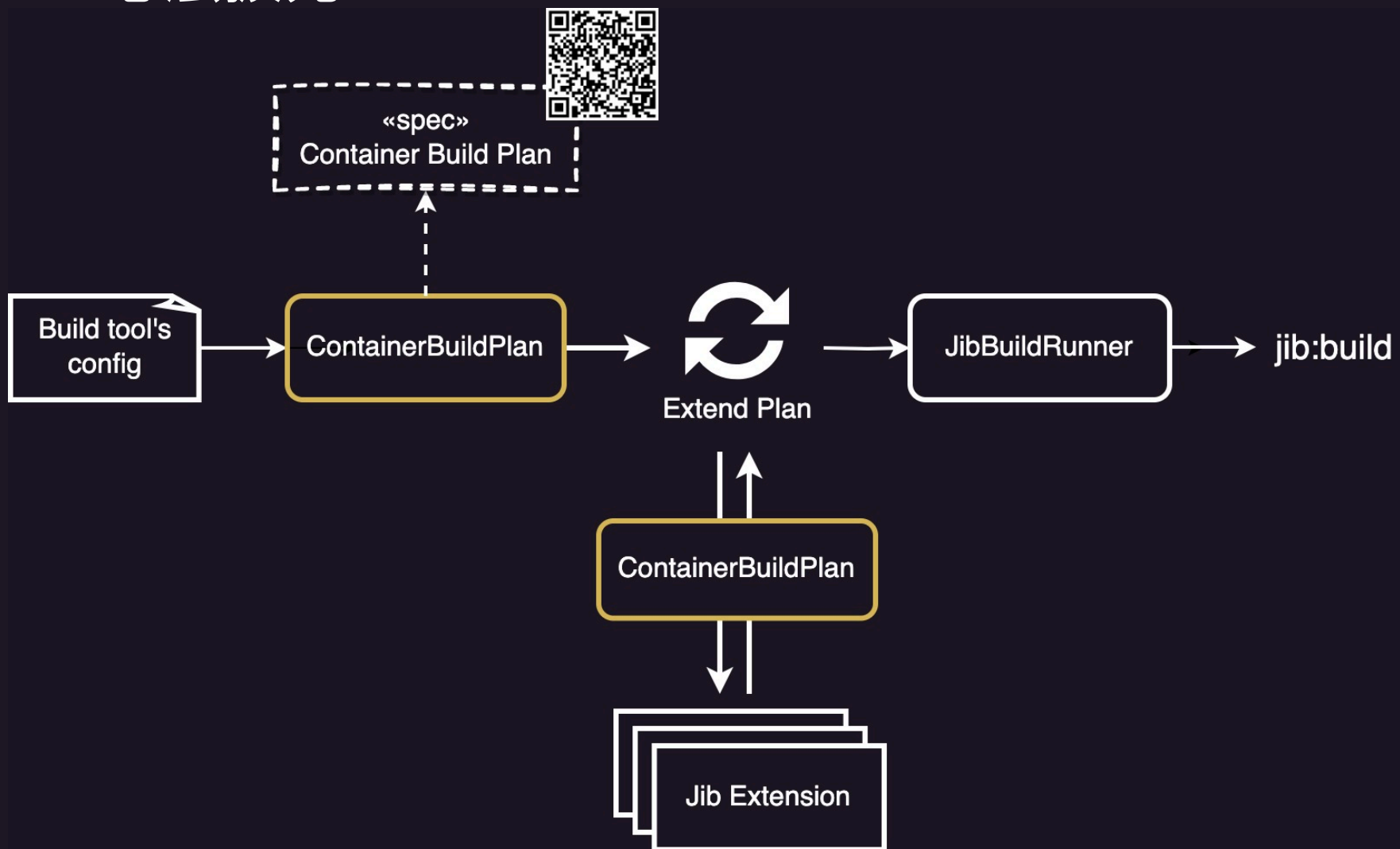
多了一個 Layer

現在的 entrypoint 長這樣

```bash
#!/bin/bash
set -e

# Perform any necessary steps before starting the JVM,
# such as setting JVM options or preparing the environment

exec java $(cat /app/jib-jvm-flags-file) \
  -cp $(cat /app/jib-classpath-file) \
  $(cat /app/jib-main-class-file) \
  "$@"
```

更多的應用

softleader/memory-calculator

基於 paketo-buildpacks/libjvm 的 JVM 記憶體計算工具

Recap

- Image 的組成

- 打包成 Image 的幾種方式

- 喜歡 Jib 的原因

- Jib 的擴充及延伸應用

Thank you 🙌