

Paketo Buildpacks: the Spring Boot way to build images, and more!



📷 presentation
content and
references 📷



 Anthony Dahanne, Software developer [@Herodevs](https://twitter.com/Herodevs)
[@anthony.dahannet.net](https://www.anthonydahannet.net)
framapiaf.org/@anthonydahanne

 July
17

April 15th 2026

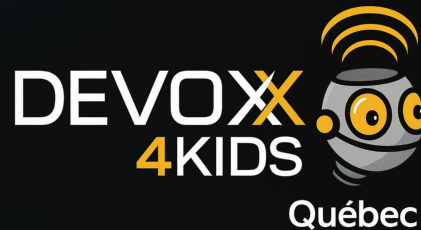
 Spring I/O Barcelona

Your presenter for this session

☕ Java and Go developer, Cloud Architect, Devops, etc. but also community leader 🤝



paketo
buildpacks



OSS Maintainer

Non Profit Orgs. President

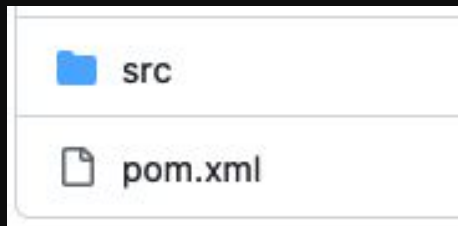


Agenda

- What are buildpacks?
- Compare some use cases with Dockerfiles
- Spring Boot optimizations
- Buildpacks usage

What are buildpacks?

Detect and build!



Java Buildpack 



```
java -jar my-app.jar
```

==> BUILDING

[builder] Paketo Buildpack for BellSoft Liberica

[builder] Downloading from <https://github.com/bell-sw/etc/bellsoft-jdk21.0.6+10-linux-aarch64.tar.gz>

[builder] Paketo Buildpack for Maven

[builder] Executing `mvnw --batch-mode -Dmaven.test.skip=true --no-transfer-progress package`

[builder] Paketo Buildpack for Spring Boot

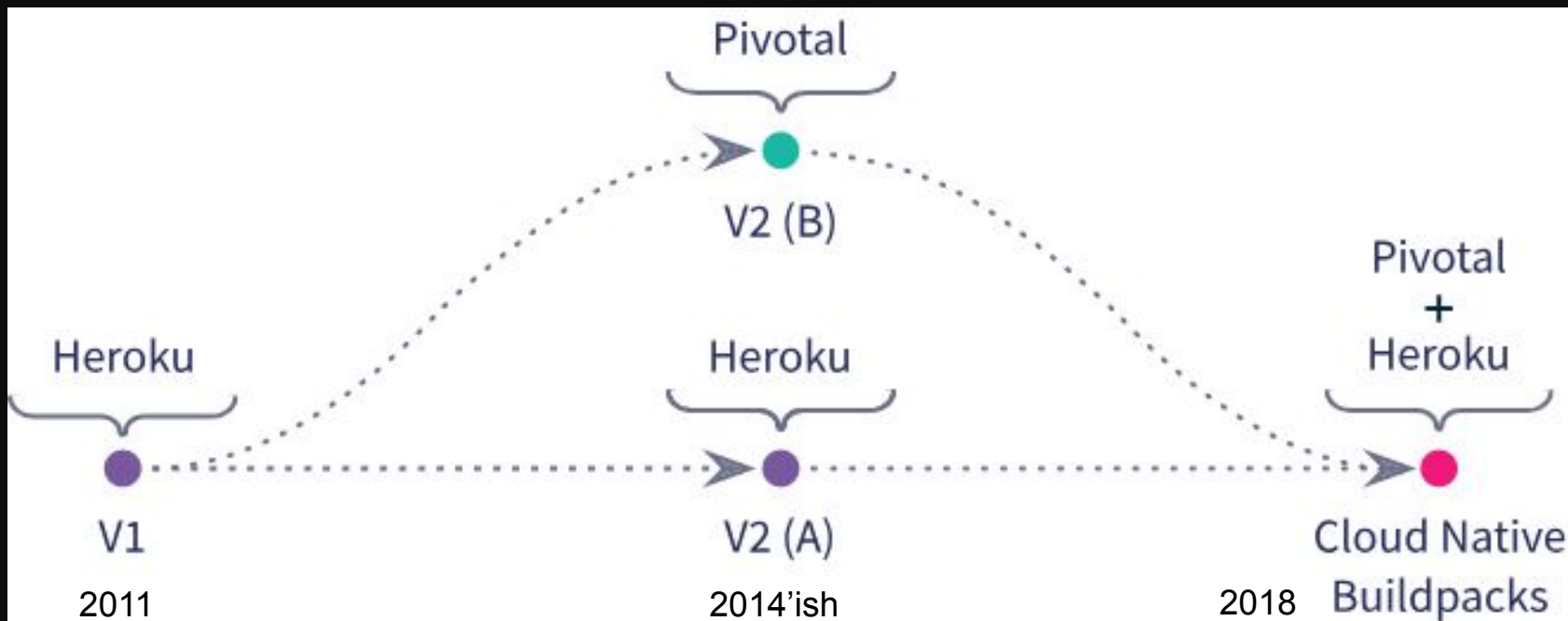
[builder] Downloading from <https://repo1.maven.org/maven2/org/springframework/cloud/spring-cloud-bindings/2.0.4/spring-cloud-bindings-2.0.4.jar>

What are buildpacks?

Many iterations across the years!

Other kind of buildpacks exist too!

- [Nixpacks](#)
- Custom shell scripts!



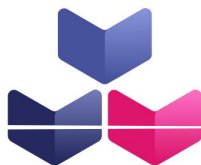
<https://buildpacks.io>

Where do they come from?

A spec and CLI under the CNCF umbrella



vmware



Buildpacks.io

<https://buildpacks.io>

<https://github.com/buildpacks>

 **pack** Public

CLI for building apps using Cloud Native Buildpacks

● Go ☆ 2.3k 🍴 254

 **spec** Public

Specification for Cloud Native Buildpacks

● Shell ☆ 243 🍴 69

 **lifecycle** Public

Reference implementation of the Cloud Native Buildpacks lifecycle

● Go ☆ 176 🍴 91

What's a Paketo?

OSS buildpacks implementations



vmware



<https://paketo.io>

<https://github.com/paketo-buildpacks>

Java

Node.js

.NET Core

Go

Web Servers

Python

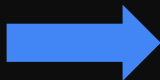
PHP

Ruby

Why Paketo buildpacks over ... the Dockerfile?

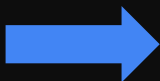
Benefit from maintainers expertise

```
FROM openjdk
COPY target/*runner.jar /app/app-runner.jar
WORKDIR /app
EXPOSE 8080
ENTRYPOINT [ "java", "-jar" ]
CMD ["app-runner.jar"]
```



`java -jar app-runner.jar`

```
pack build my-app
--builder      paketobuildpacks/builder-noble-java
-tiny
```



```
Setting Active Processor Count to 5
Calculating JVM memory based on 15788052K available memory
Calculated JVM Memory Configuration: -XX:MaxDirectMemorySize=10M
-Xmx15397353K -XX:MaxMetaspaceSize=83498K
-XX:ReservedCodeCacheSize=240M -Xss1M (Total Memory: 15788052K,
Thread Count: 50, Loaded Class Count: 12328, Headroom: 0%)
Enabling Java Native Memory Tracking
Adding 137 container CA certificates to JVM truststore
Spring Cloud Bindings Enabled
Picked up JAVA_TOOL_OPTIONS:
-Djava.security.properties=/layers/paketo-buildpacks_bellsoft-libe
rica/java-security-properties/java-security.properties
-XX:+ExitOnOutOfMemoryError -XX:ActiveProcessorCount=5
-XX:MaxDirectMemorySize=10M -Xmx15397353K
-XX:MaxMetaspaceSize=83498K -XX:ReservedCodeCacheSize=240M -Xss1M
-XX:+UnlockDiagnosticVMOptions -XX:NativeMemoryTracking=summary
-XX:+PrintNMTStatistics
-Dorg.springframework.cloud.bindings.boot.enable=true
```

Why Paketo buildpacks over ... the Dockerfile?

Benefit from the ecosystem for common use cases : custom CA certificate

```
FROM openjdk
WORKDIR /app
COPY custom-ca.crt /app
RUN keytool -importcert -file custom-ca.crt
EXPOSE 8080
ENTRYPOINT [ "java", "-jar" ]
CMD ["app-runner.jar"]
```



```
pack build my-app
  --builder
  paketo/buildpacks/builder-jammy-base
  --volume
  $PWD/ca-certs:/bindings/my-certs
```

```
FROM golang:1.15 AS builder
RUN apk update
RUN apk add -U ca-certificates && update-ca-certificates
WORKDIR /GreetingAPI
COPY . /greeting
WORKDIR /greeting
ENV GO111MODULE=on
RUN CGO_ENABLED=0 GOOS=linux go build -o greeting
```

```
FROM scratch
COPY --from=builder /etc/ssl/certs/ca-certificates.crt
/etc/ssl/certs/
COPY --from=builder /greeting .
CMD ["/greeting"]
```



```
pack build my-app
  --builder
  paketo/buildpacks/builder-jammy-base
  --volume
  $PWD/ca-certs:/bindings/my-certs
```

Why Paketo buildpacks over ... the Dockerfile?

SBOM generation out of the box!

```
pack sbom download myapp --output-dir ./app-sbom
```

```
{
  "id": "1a3e0473bf22b7d6",
  "name": "jackson-databind",
  "version": "2.14.1",
  "type": "java-archive",
  "foundBy": "sbom-cataloger",
  "locations": [
    {
      "path": "layers/sbom/launch/paketo-buildpacks_executable-jar/sbom.cdx.json"
    }
  ],
  "licenses": [],
  "language": "java",
  "cpes": [
    "cpe:2.3:a:jackson-databind:jackson-databind:2.14.1:*:*:*:*:*:*:*",
    "cpe:2.3:a:jackson-databind:jackson_databind:2.14.1:*:*:*:*:*:*:*",
  ]
}
```

Why Paketo buildpacks over ... the Dockerfile?

Third party agents integration!

- At build time, the agent is included

```
pack build apps/maven -e BP_DATADOG_ENABLED=true
```

- At runtime, the agent is configured

```
docker run apps/maven -e BPL_JMX_ENABLED=true  
                        -e BPL_JMX_PORT=8000  
                        -e DD_ENV=paketo-prod  
                        -e DD_SERVICE=paketo-maven  
                        -e DD_VERSION=1.0  
                        -e DD_PROFILING_ENABLED=true
```

Demo time!

Let's build some Java apps!

- Create a project from <https://start.spring.io>
 - Use the pack CLI to build the container image
 - `pack build my-app -B builder-noble-java-tiny`
 - `docker run my-app`
 - Use Spring Boot plugin to build the container image
 - `./mvnw spring-boot:build-image`
 - `docker run demo:0.0.1-SNAPSHOT`
 - Done!

Java Paketo Buildpacks goodies

Building java container images, the best way

- Choose your Java version easily!

- BP_JVM_VERSION=26

- Choose your Java vendor easily!

```
<buildpacks>
  <buildpack>paketo/buildpacks/azul-zulu:latest</buildpack>
  <buildpack>urn:cnb:builder:paketo-buildpacks/java:latest</buildpack>
</buildpacks>
```

- Rely on Memory Calculator to customize your JVM arguments
- Spring Boot optimizations
 - spring-cloud-bindings
 - Spring generation
 - AOTCache and other performance optimizations
- Let's see those in action, with the `spring-petclinic`
 - <https://github.com/spring-projects/spring-petclinic>

How to integrate them to my existing pipelines?

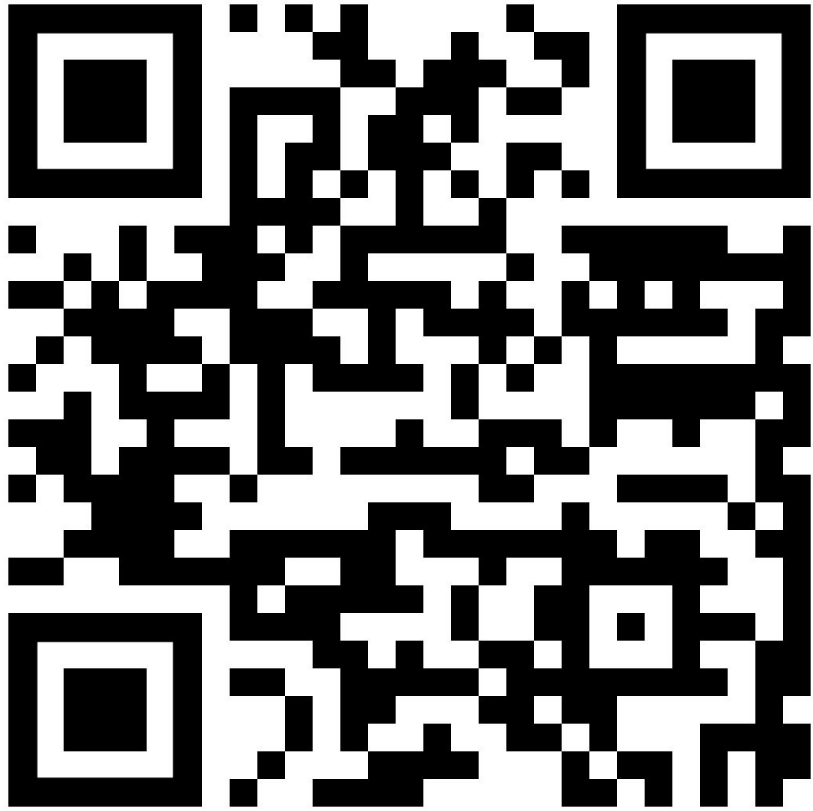
Are there some tools that can run buildpacks for me?

- [Spring Boot Maven / Gradle plugins](#)
 - [Quarkus too!](#)
- Building images natively from Kubernetes, using [Epinio](#)
- From your favorite CI tool
 - [Github Actions](#)
 - [Gitlab CI/CD](#)
 - [Jenkins](#)
- From your favorite PaaS
 - Fly.io, Heroku, Porter
- Or cloud provider
 - [GCP Cloud Run](#), [Azure Container Apps](#)

What's in for the future?

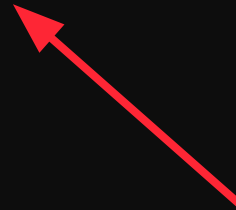
We asked the community, and we compiled the results:

- Distroless builders
 - Lightweight base images with the minimum to run your app - often with no CVEs
 - Bellsoft Alpaquita, Chainguard Wolfi, Docker Hardened images, etc.
- SBOM with CycloneDX
 - Currently it's syft based, which is not as used as CycloneDX
 - Better SBOM integration means better component transparency
- Cross-stack package manager buildpack
 - Right now we have an **apt** buildpack for Debian based images
 - What about Alpine, UBI, etc. users who want to install browsers, etc.



<https://linktr.ee/buildpacks>

Thank you!



 presentation
content and
references 

