

# Cloud Native with Spring Boot and Kubernetes

**From development to production**

---

Thomas Vitale  
GOTO Copenhagen  
Nov 10th, 2021

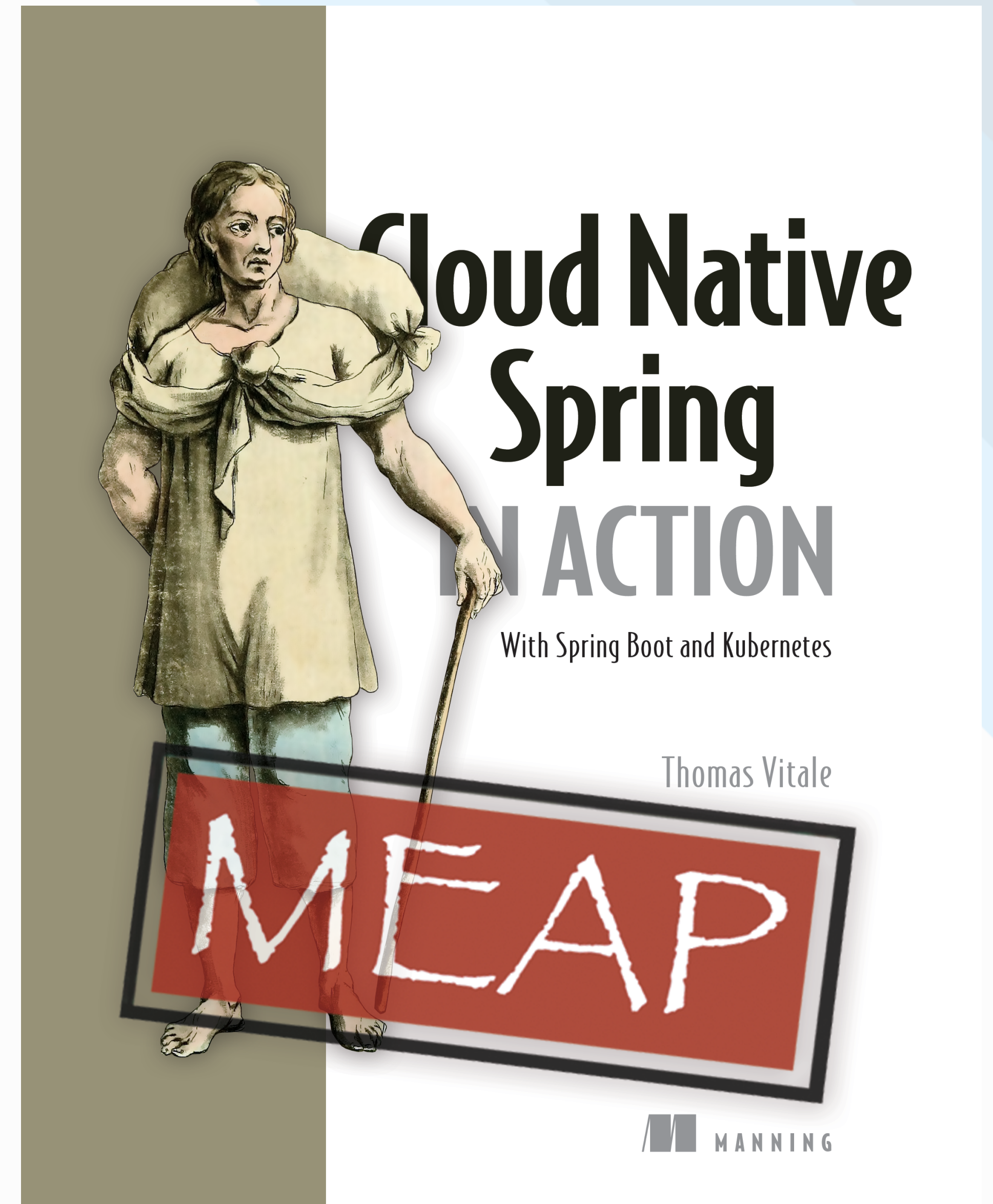
@vitaletomas

# About Me

## Thomas Vitale

- **Senior Software Engineer** at Systematic, Denmark.
- Author of “**Cloud Native Spring in Action**” (Manning).
- **Spring Security** and **Spring Cloud** contributor.

[thomasvitale.com](http://thomasvitale.com)



# Cloud Native

---

[thomasvitale.com](https://thomasvitale.com)

[@vitalethomas](https://twitter.com/vitalethomas)

# Why Cloud Native?

## Speed

Faster and flexible delivery

## Resilience

Availability and stability

## Scale

Elasticity and dynamic scaling

## Cost

Efficiency and cost optimisation

# The Three P's of Cloud Native Applications

## Place

Private Cloud  
Public Cloud  
Hybrid Cloud

## Properties

Scalability  
Loose Coupling  
Resilience  
Manageability  
Observability  
Security

## Practices

Automation  
Continuous Delivery  
DevOps

# From Development to Production

Cloud native journey in less than 45 minutes

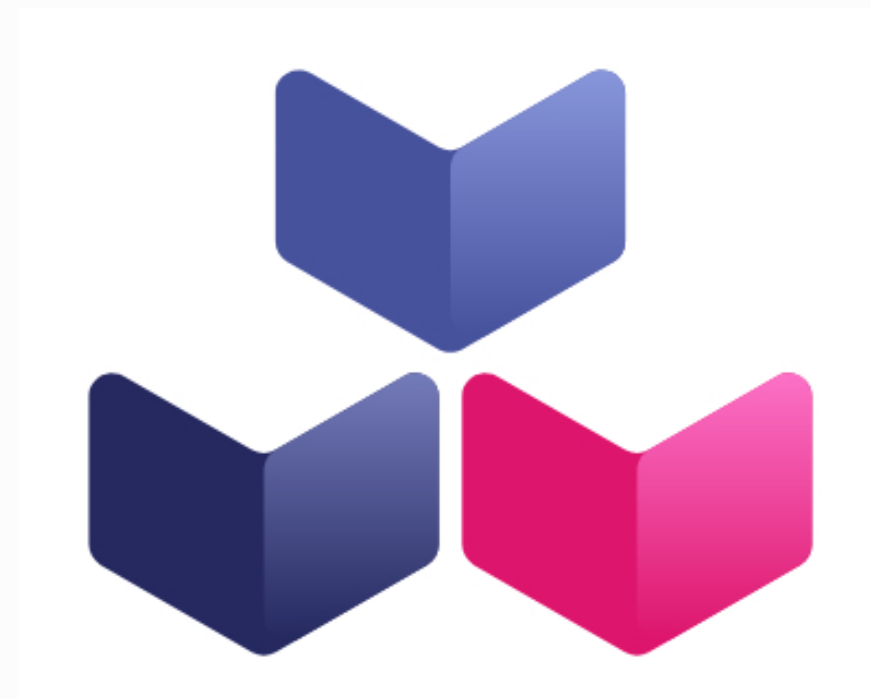
Development



Spring Boot



Containerization



Cloud Native  
Buildpacks



Deployment



Kubernetes

# Cloud Native Development

---



# Cloud Native Development

## Development principles with Spring Boot



- **Self-contained application**
  - Embedded server
  - No external dependencies
  - JAR packaging (“uber-JAR”)
- **Externalized configuration**
  - Property files for default values
  - JVM system variables
  - Environment variables



# Packaging Spring Boot

## Standalone JAR



JAR

Gradle

`bootJar`

Maven

`spring-boot:repackage`

# Containerization

---

- 1 Don't use fat JARs
- 2 Optimize build/runtime performance
- 3 Don't run as *root* or include secrets

# Containerize Java Applications

Dockerfiles

Jib

Cloud Native Buildpacks



Buildpacks.io

[Features](#)

[Community](#)

[Blog](#)

[Registry](#)

[Docs](#)

[GitHub](#)

**Cloud Native Buildpacks**  
transform your application source code into  
images that can run on any cloud.

[Get Started](#)

## Why Cloud Native Buildpacks?

### Control



Balanced control between App Devs  
and Operators.

### Compliance



Ensure apps meet security and  
compliance requirements.

### Maintainability



Perform upgrades with minimal  
effort and intervention.

[buildpacks.io](https://buildpacks.io)



[docs](#) [community](#) [blog](#) [github](#)

Your app,  
in your favorite language,  
ready to run in the cloud



**GraalVM**



**NGINX**

**node**



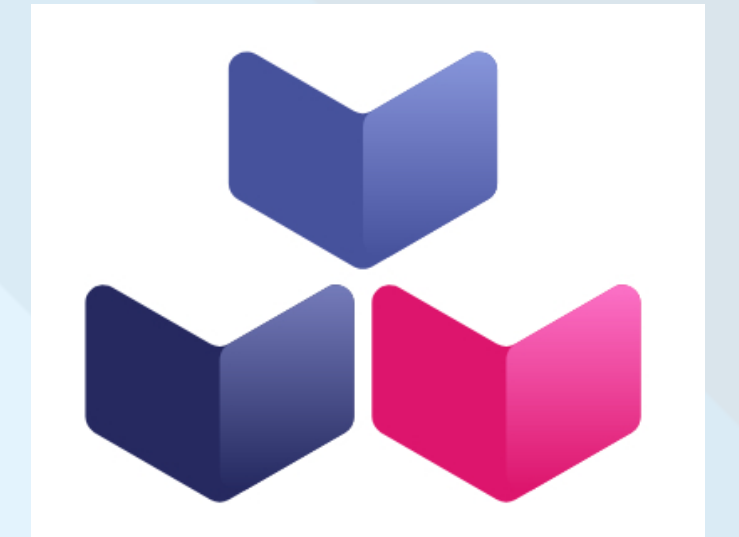
**python**

Get Started

[paketo.io](https://paketo.io)

# Packaging Spring Boot

## JAR & Container Image



JAR

Gradle

`bootJar`

Maven

`spring-boot:repackage`

Container Image

Gradle

`bootBuildImage`

Maven

`spring-boot:build-image`



# Spring Boot on Kubernetes

---

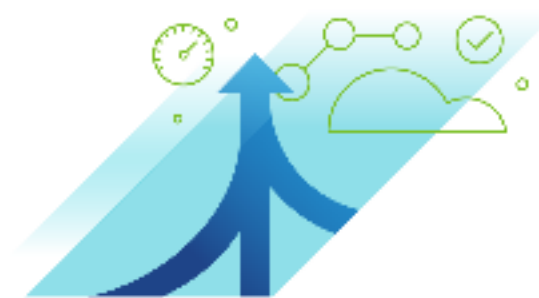


## Experience VMware Tanzu now

VMware Tanzu Community Edition is a full-featured, easy-to-manage Kubernetes platform for learners and users, especially those working in small-scale or preproduction environments

TRY IT OUT

DOWNLOAD



### Take on new challenges confidently

Count on the same open source software used in Tanzu commercial editions to take you wherever you need to go.



### Explore independently


Enjoy free Tanzu software, top-notch education resources, and community support.



### Get up and running in minutes

Install and configure quickly on your local workstation or favorite cloud.

[tanzucommunityedition.io](https://tanzucommunityedition.io)

 Knative v0.26 ▾

Search

GitHub  
☆ 3k 👤 968

HomeGetting startedAdministration guideDeveloper guideKnative ServingKnative EventingClientBlogReference


Join the Community ➞

# Enterprise-grade Serverless on your own terms.

Kubernetes-based platform to deploy and manage modern serverless workloads.

Developer? Get Started Here!

Cluster Administrator? Get Started Here!



## Highlights

- ✓ **Stand up scalable, secure, stateless services in seconds.**
- ✓ **Focused API** with higher level abstractions for common app use-cases.
- ✓ **Pluggable components** let you bring your own logging and monitoring, networking, and service mesh.
- ✓ Run Knative anywhere Kubernetes runs, **never worry about vendor lock-in.**
- ✓ **Seamless developer experience**, supports GitOps, DockerOps, ManualOps, etc..
- ✓ **Supports many common tools and frameworks** such as Django, Ruby on Rails, Spring, and many more.

[knative.dev](https://knative.dev)

# Configuring Resources

## CPU & Memory



CPU

Compressible  
Resource

When limit hit  
Throttle

Memory

Non  
Compressible  
Resource

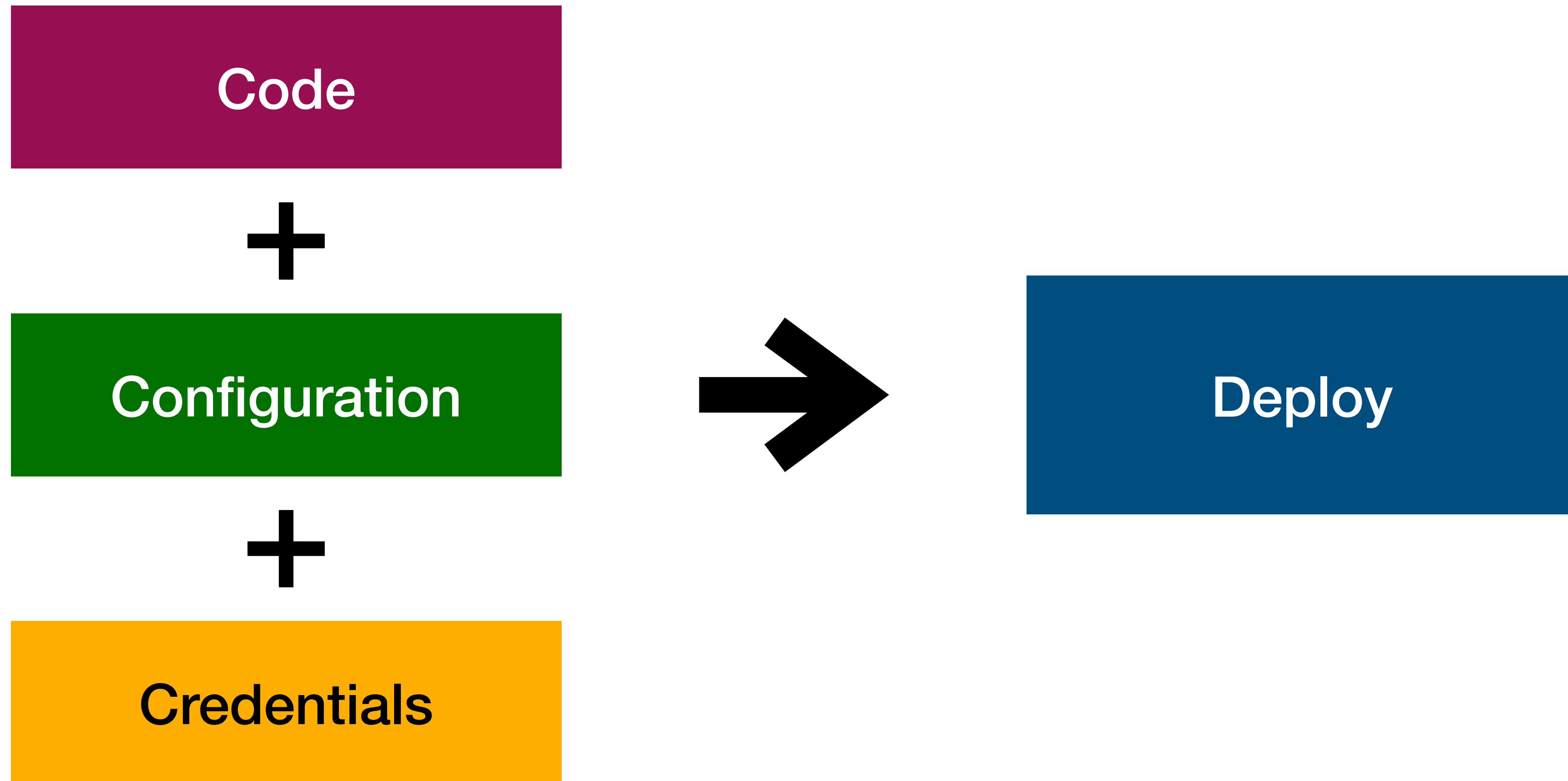
When limit hit  
OOMKilled

```
containers:
- name: book-service
  image: book-service:1.0.0
  resources:
    requests:
      memory: "512Mi"
      cpu: "0.5"
    limits:
      memory: "512Mi"
```

# Externalized Configuration

---

# Code, Configuration, Credentials



# Configuration Options

## Spring Boot Properties

- Property files
- Command line arguments
- Environment variables

## Cloud Platform Services

- Spring Cloud Alibaba
- Spring Cloud AWS
- Spring Cloud Azure
- Spring Cloud GCP

## Configuration Services

- Spring Cloud Config Server
- Spring Cloud Consul Config
- Spring Cloud Vault
- Spring Cloud Zookeeper Config

## Kubernetes Platform

- ConfigMaps
- Secrets
- Environment variables



# ConfigMaps & Secrets



Mount the volume to the /config path, where Spring Boot will automatically read property files.

```
containers:
- name: book-service
  image: book-service:0.0.1-SNAPSHOT
  volumeMounts:
  - name: book-service-volume
    mountPath: /workspace/config
volumes:
- name: book-service-volume
  configMap:
    name: book-service
```

Load the ConfigMap  
In a volume

# Graceful Shutdown

---

# Graceful shutdown

## Spring Boot and Kubernetes

- **Spring Boot**

- Enable graceful shutdown
- Define a grace period

- **Kubernetes**

- Add pre-stop hook
- Define a grace period

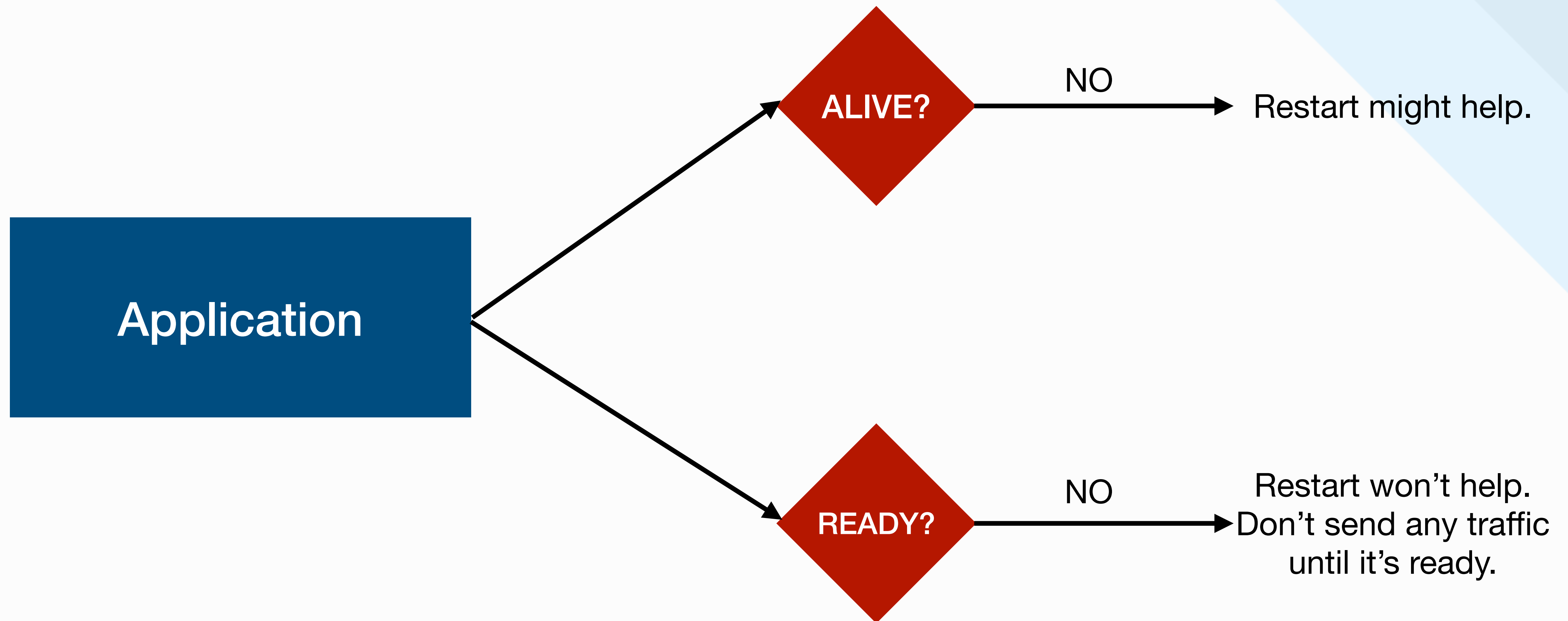
# Health Probes

---

[thomasvitale.com](https://thomasvitale.com)

[@vitalethomas](https://twitter.com/vitalethomas)

# Liveness and Readiness Probes



# Liveness and Readiness Probes



When not available,  
Kubernetes will restart  
the container

```
containers:
- name: book-service
  image: book-service:1.0.0
  livenessProbe:
    httpGet:
      path: /actuator/health/liveness
      port: 8080
    initialDelaySeconds: 10
    periodSeconds: 5
  readinessProbe:
    httpGet:
      path: /actuator/health/readiness
      port: 8080
    initialDelaySeconds: 5
    periodSeconds: 15
```

When not ready,  
Kubernetes will not send  
any traffic to the container

# Spring Native

---

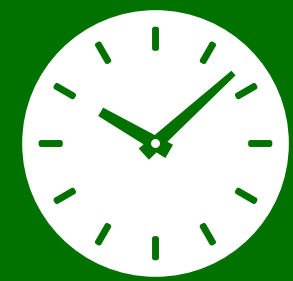
[thomasvitale.com](http://thomasvitale.com)

[@vitalethomas](https://twitter.com/vitalethomas)



# Spring Native

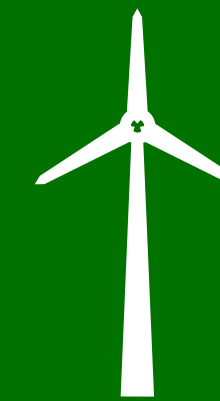
## Native executables with GraalVM



Instant  
Startup



Instant  
Peak  
Performance



Reduced  
Memory  
Consumption



Slower  
Heavier  
Build



Fewer  
Runtime  
Optimizations

# From Development to Production

Cloud native journey in less than 45 minutes

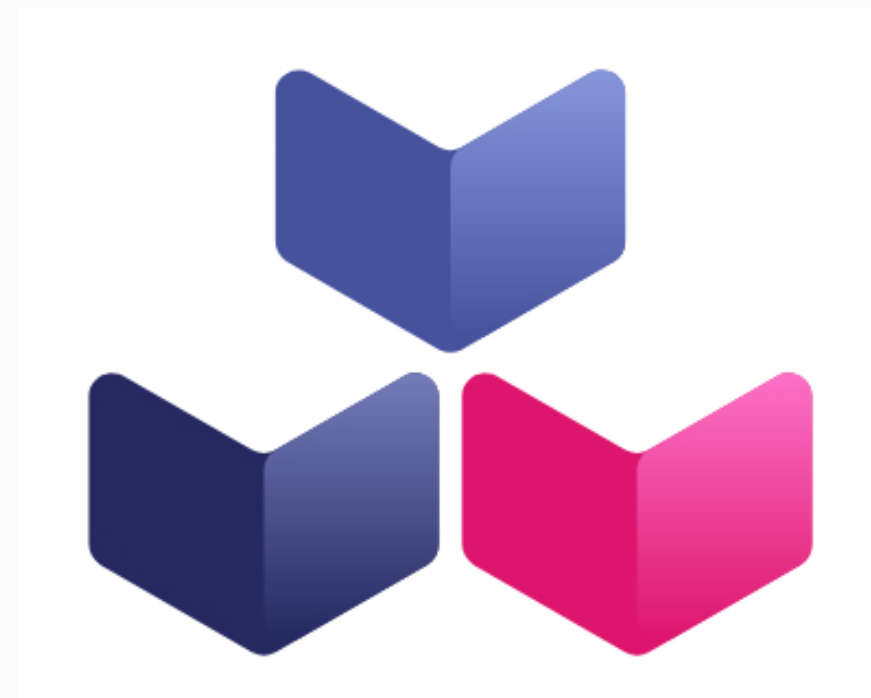
Development



Spring Boot



Containerization



Cloud Native  
Buildpacks



Deployment



Kubernetes

# Cloud Native with Spring Boot and Kubernetes

**From development to production**

---

Thomas Vitale  
GOTO Copenhagen  
Nov 10th, 2021

@vitaletomas