

Deploying Using Docker



Bogdan Sucaciu

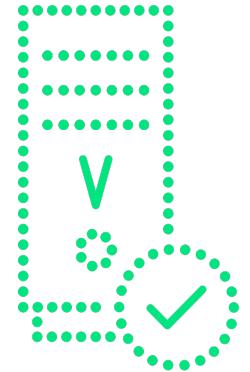
Tech Lead

@bsucaciu | bsucaciu.com

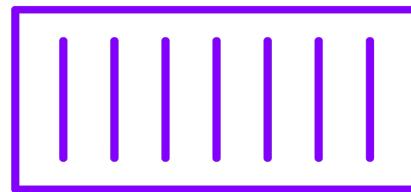
Deploying Spring Framework 6 Applications Playbook



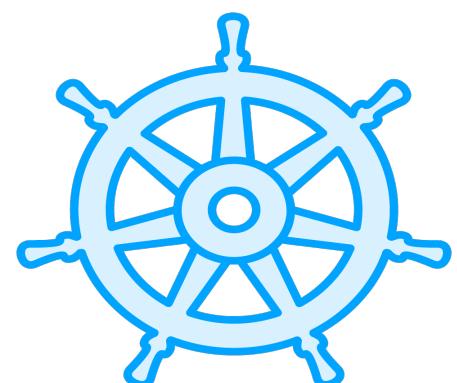
Independent Course Modules



Utilizing Virtual Machines



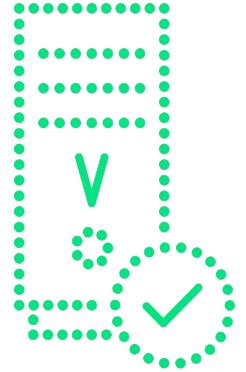
Deploying using Docker



Orchestrating using Kubernetes

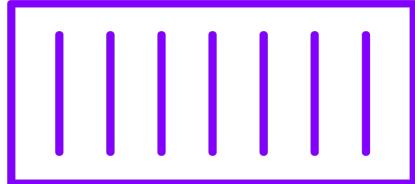


Prerequisites



Java 17

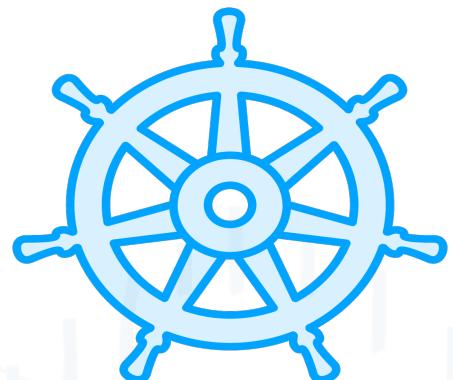
Spring
Framework 6



Java 17

Spring
Framework 6

Docker



Java 17

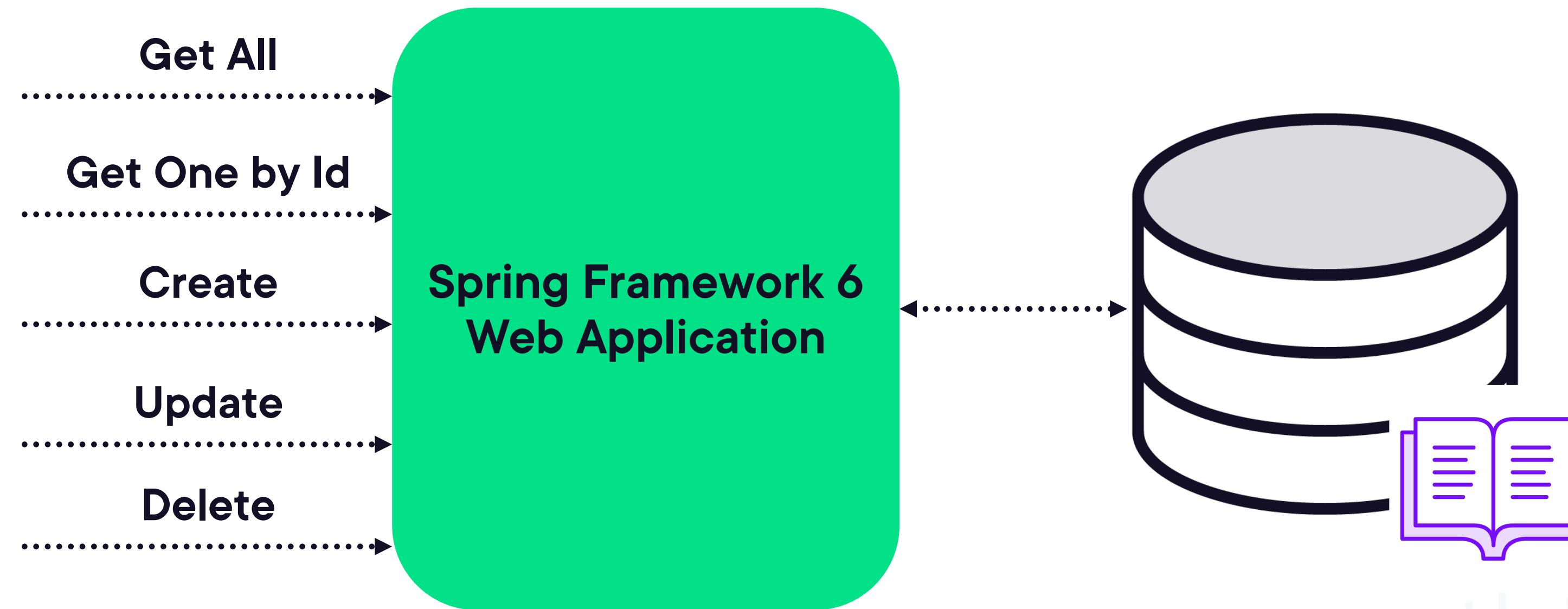
Spring
Framework 6

Docker

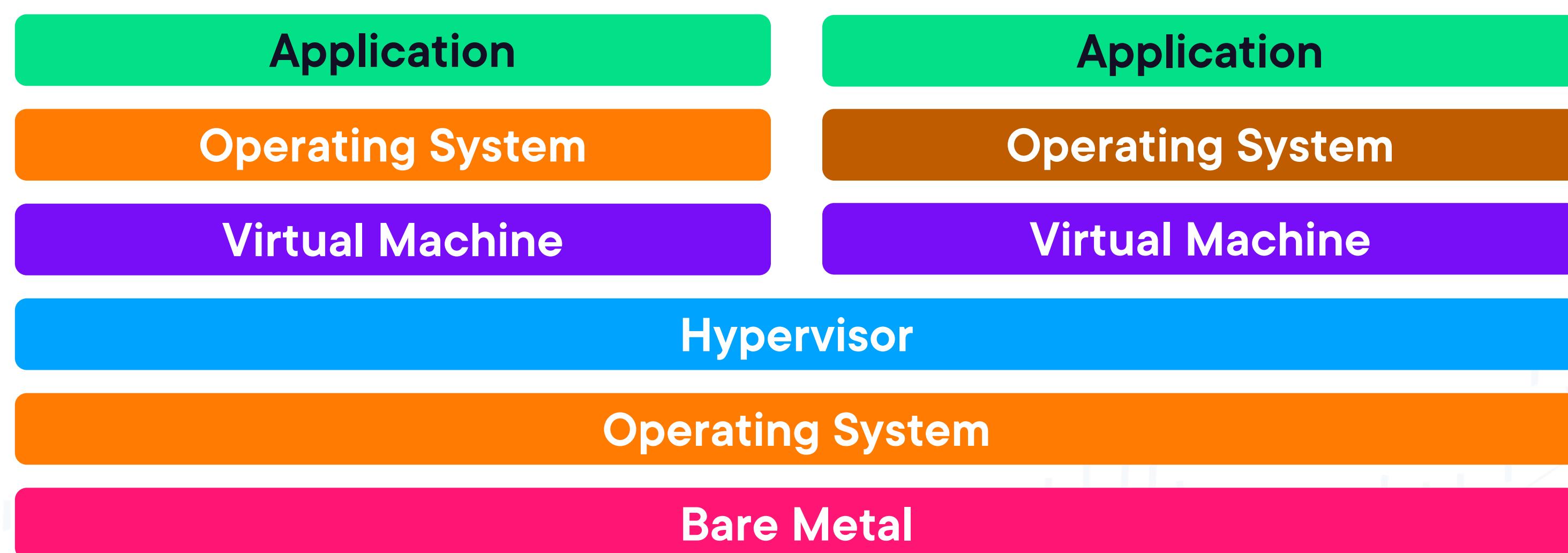
Kubernetes



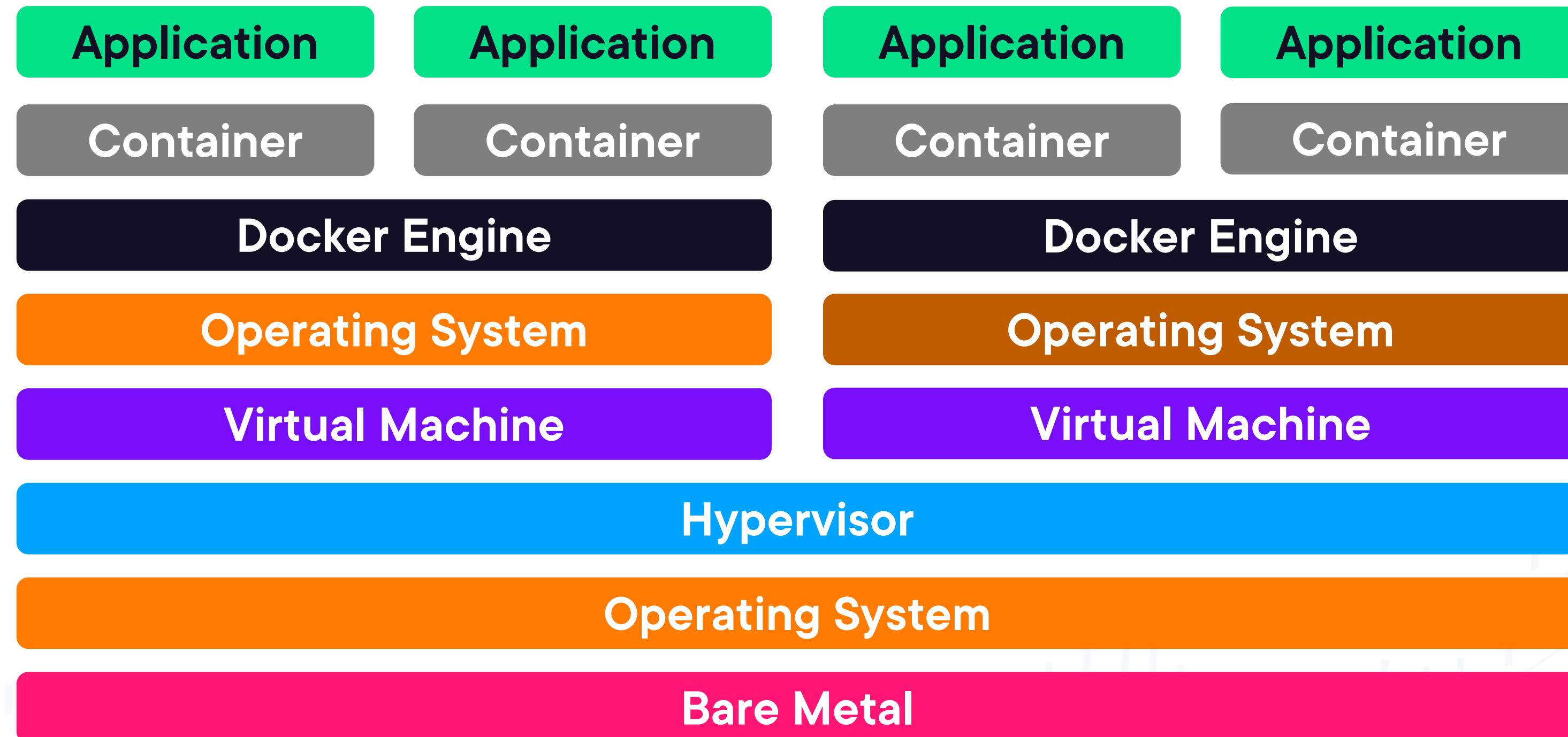
Spring Framework 6 Application



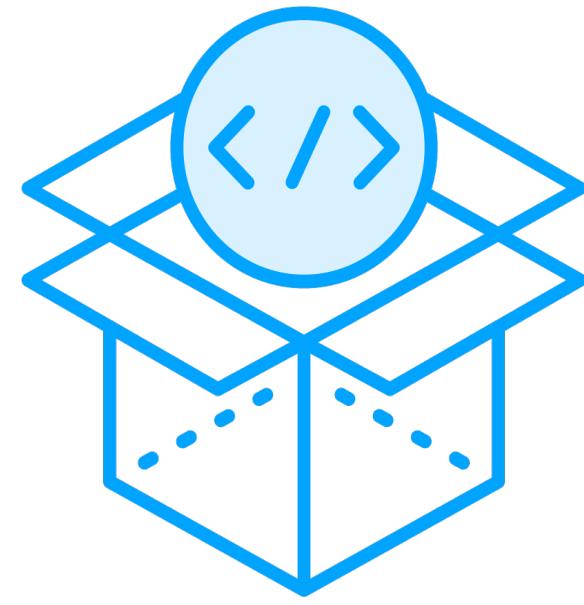
Virtual Machines



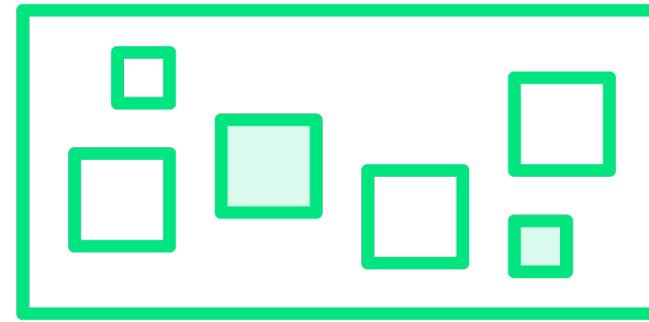
Containerized Applications



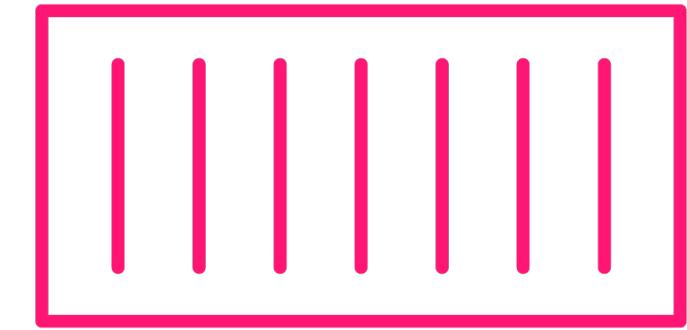
Containerized Applications



Package to JAR



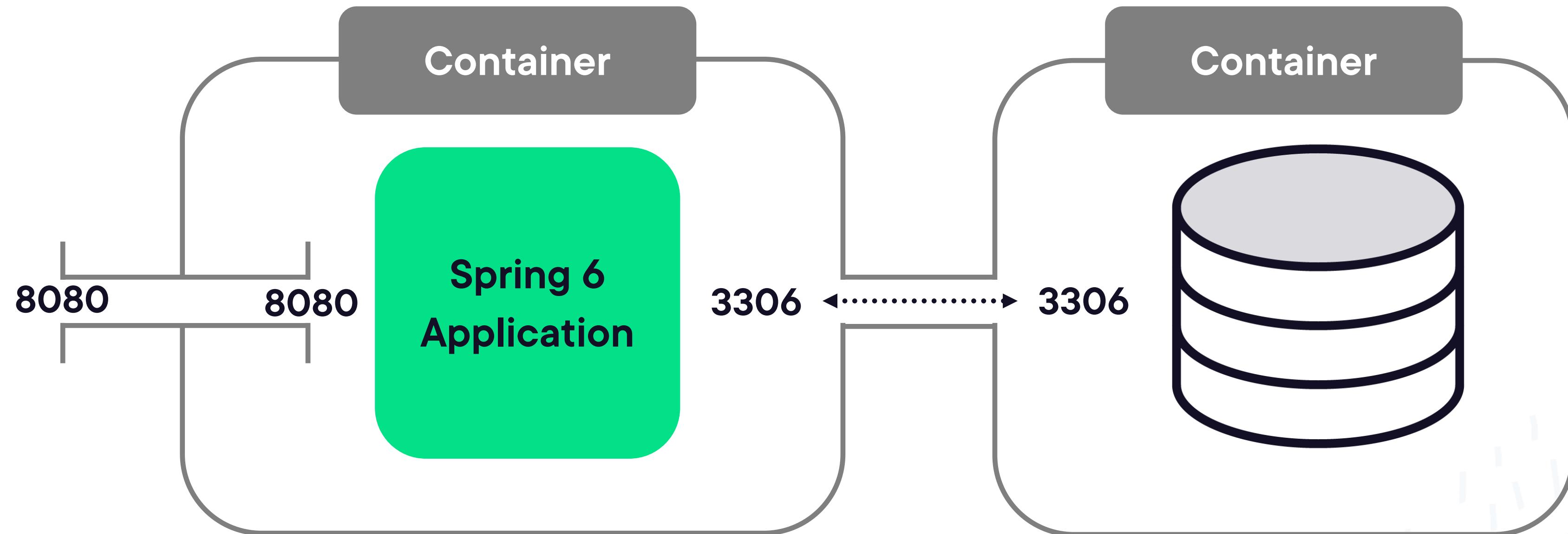
Create Docker image



Run container



Containerized Applications



Demo



Use Maven Spring Boot Plugin

Create Dockerfile

Build Docker Image



Demo



Deploy MySQL using Docker
Connect application to Database



Demo



Dockerfile best practices

Dockerfile best practices

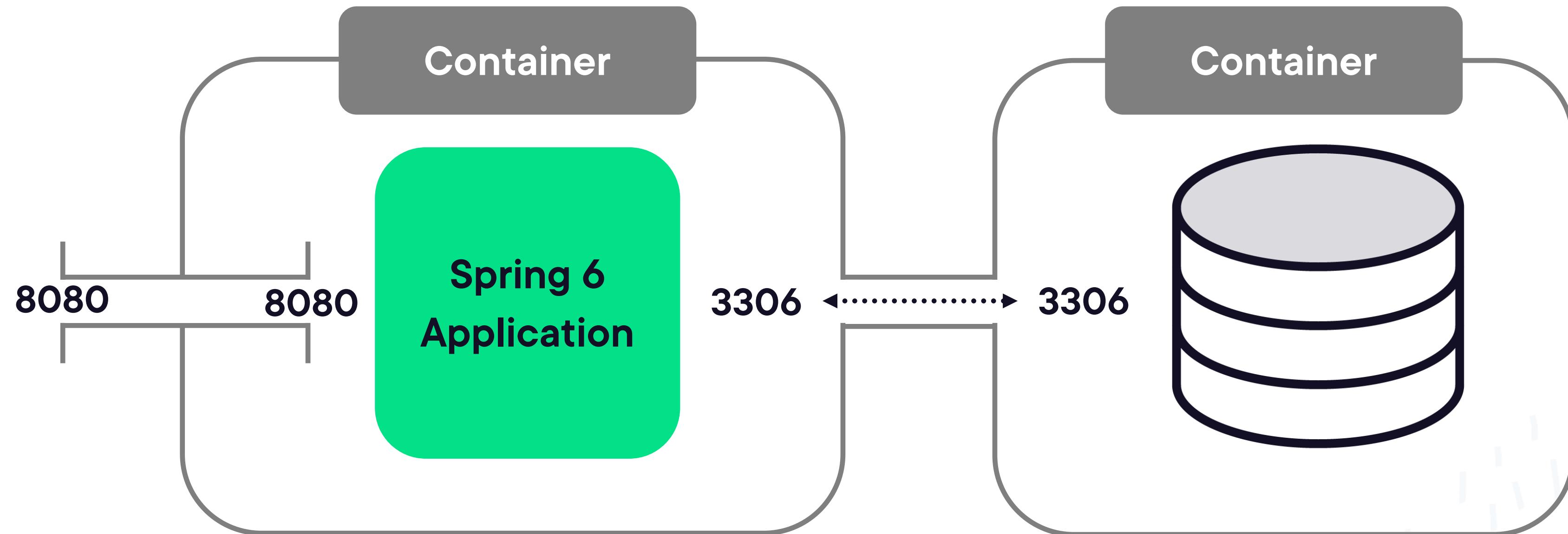
- Set workdir
- Add non-root user
- Add healthcheck



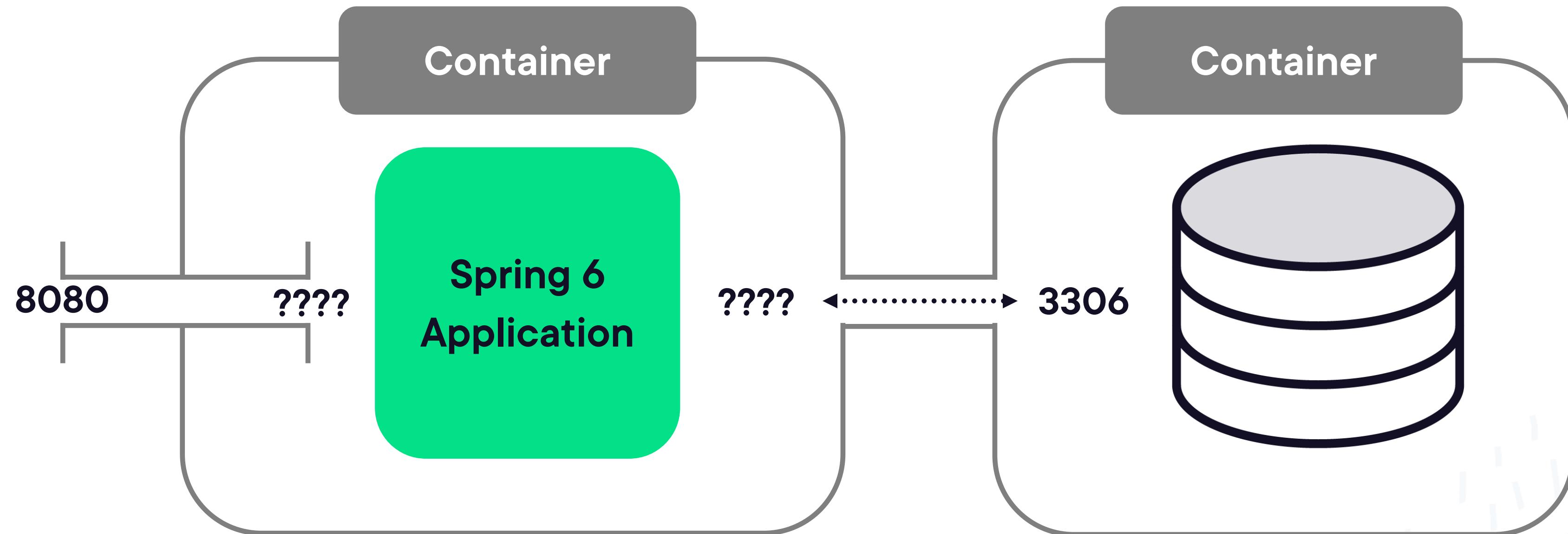
Environment Variables



Environment Variables



Environment Variables



Providing Properties

application.yml

```
server:  
  port: 8080  
  
spring:  
  datasource:  
    url: jdbc:mysql://mysql-server:3306/pluralsight  
    username: root  
    password: password  
    driverClassName: com.mysql.cj.jdbc.Driver
```



Externalized Configuration

Command Line Arguments

Java System properties

OS Environment Variables

@PropertySource annotations

Application properties outside of your packaged jar

Application properties packaged inside your jar

Default Properties



Externalized Configuration

Command Line Arguments

Java System properties

OS Environment Variables

@PropertySource annotations

Application properties outside of your packaged jar

Application properties packaged inside your jar

Default Properties



Binding from Environment Variables

Replace dots (.)
with underscores (_)

Remove any
dashes (_)

Convert to
uppercase



Binding from Environment Variables

Spring Properties

VS

Environment Variables

server

SERVER

server.port

SERVER_PORT

client.userName

CLIENT_USER_NAME

client.userName

CLIENT_USERNAME

spring.jpa.show-sql

SPRING_JPA_SHOW_SQL

spring.jpa.show-sql

SPRING_JPA_SHOWSQL



Demo



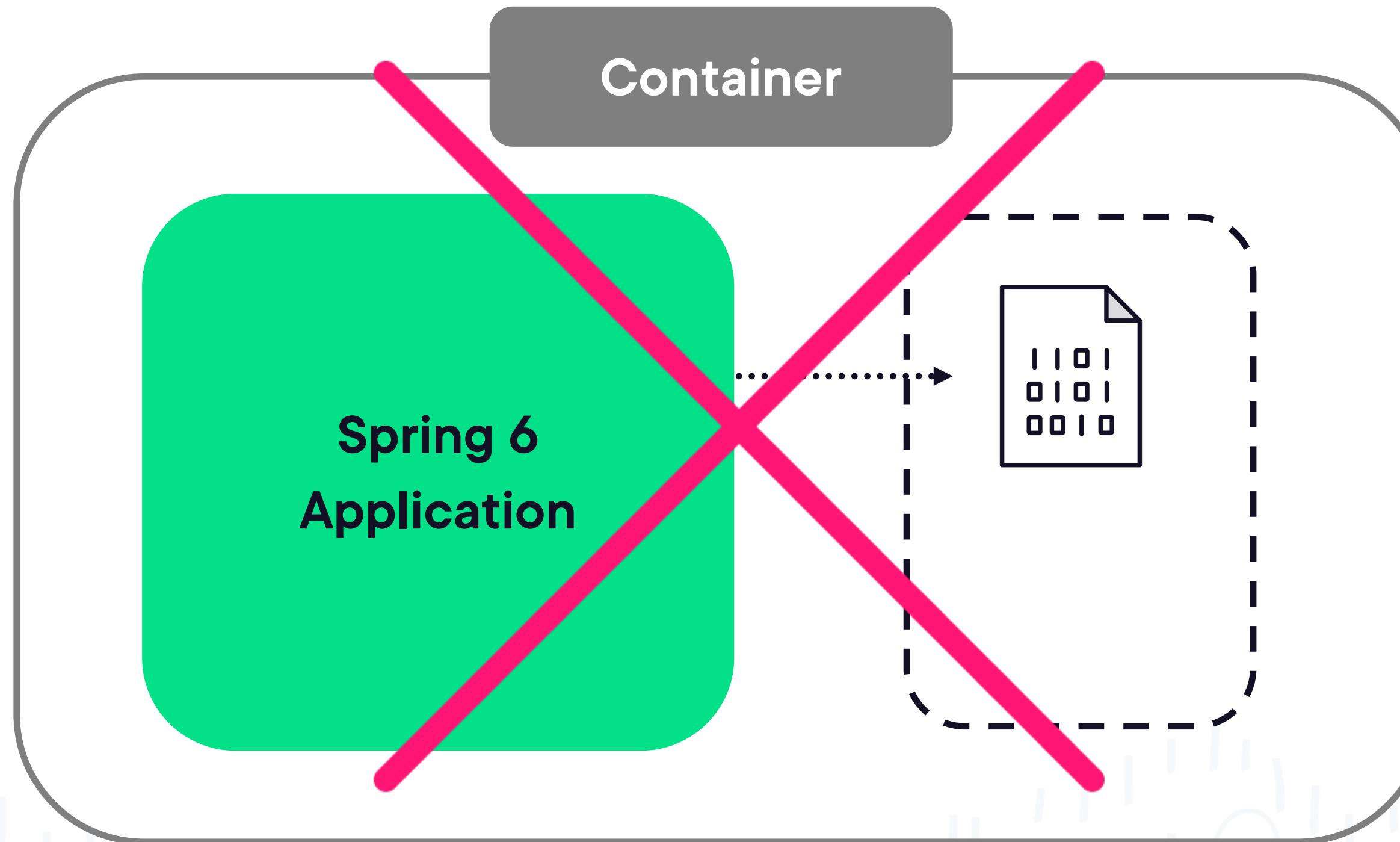
Set Environment variables using Docker



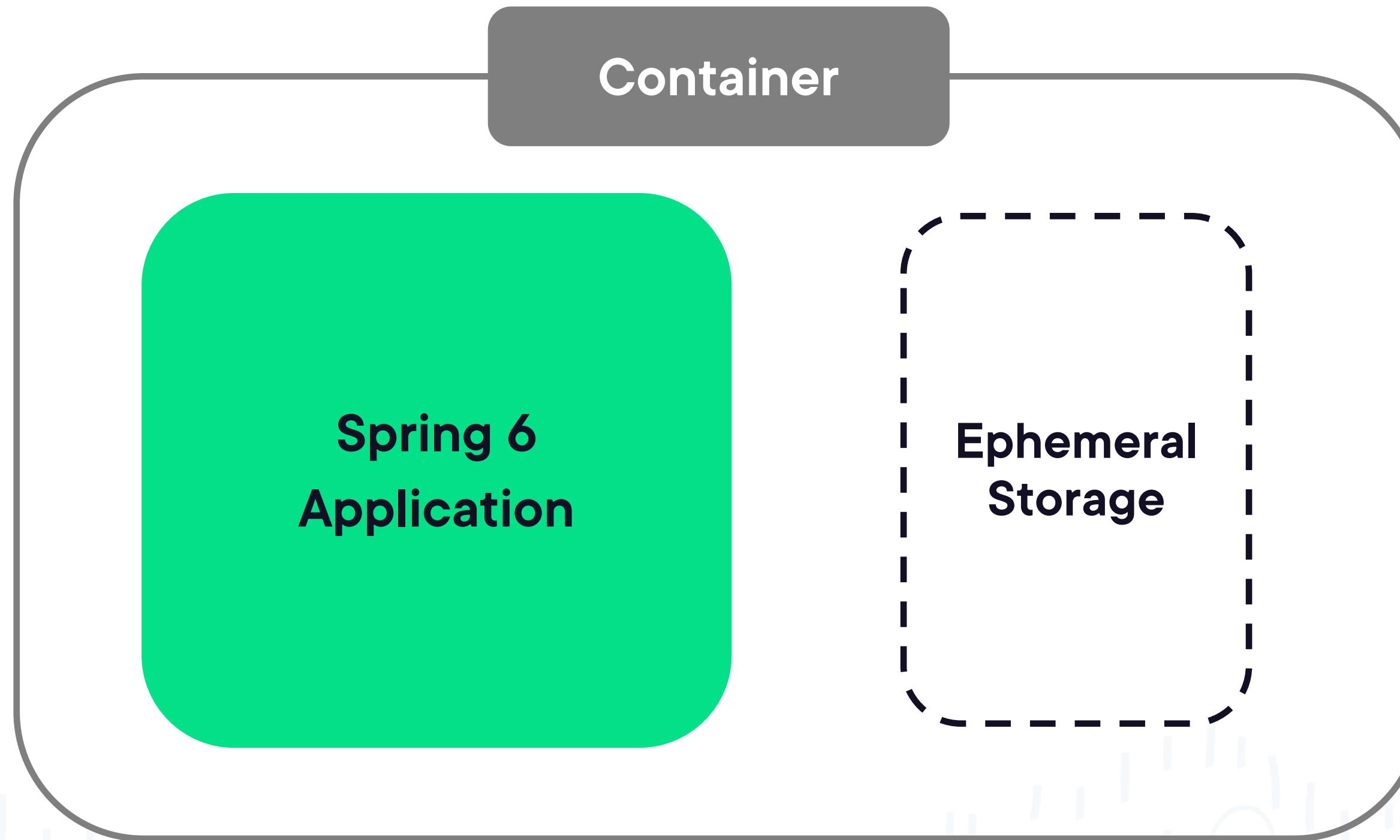
Persisting Storage



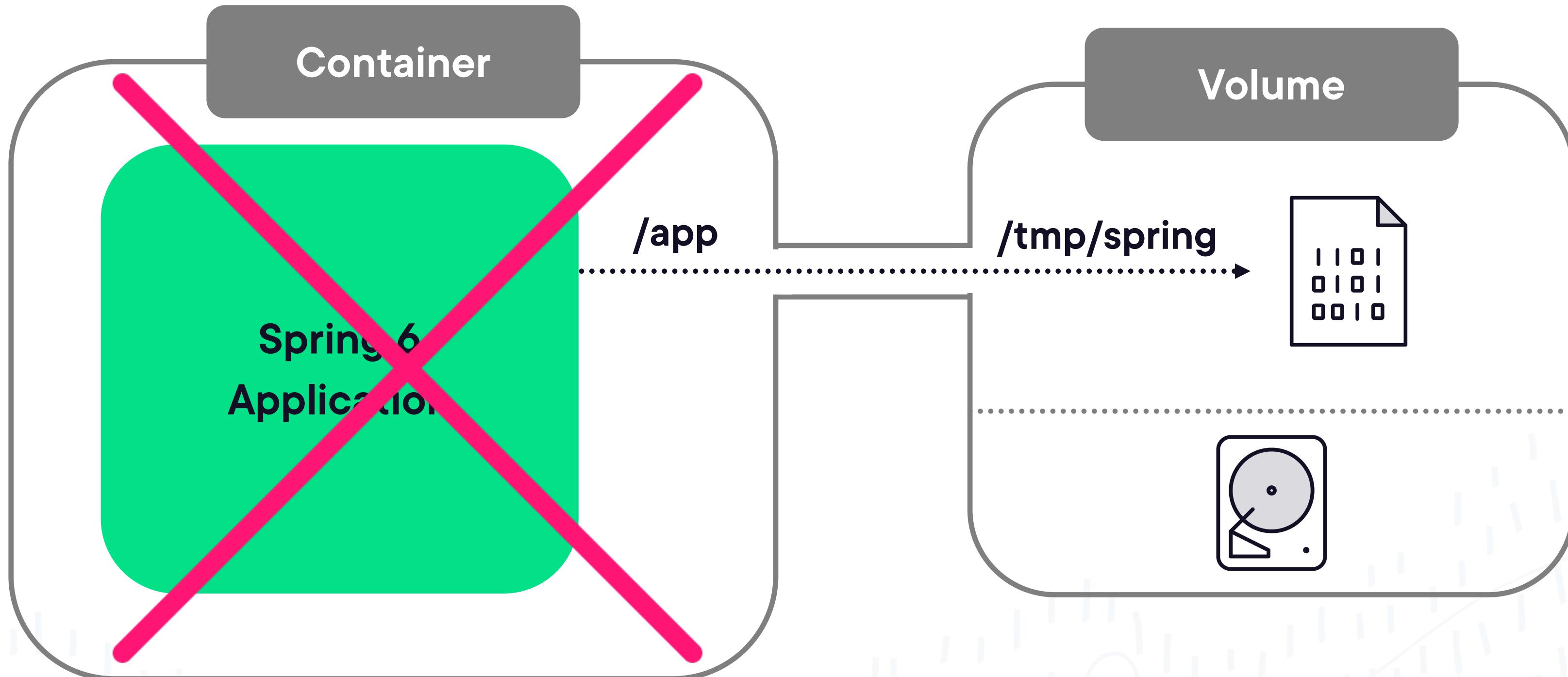
Persisting Storage



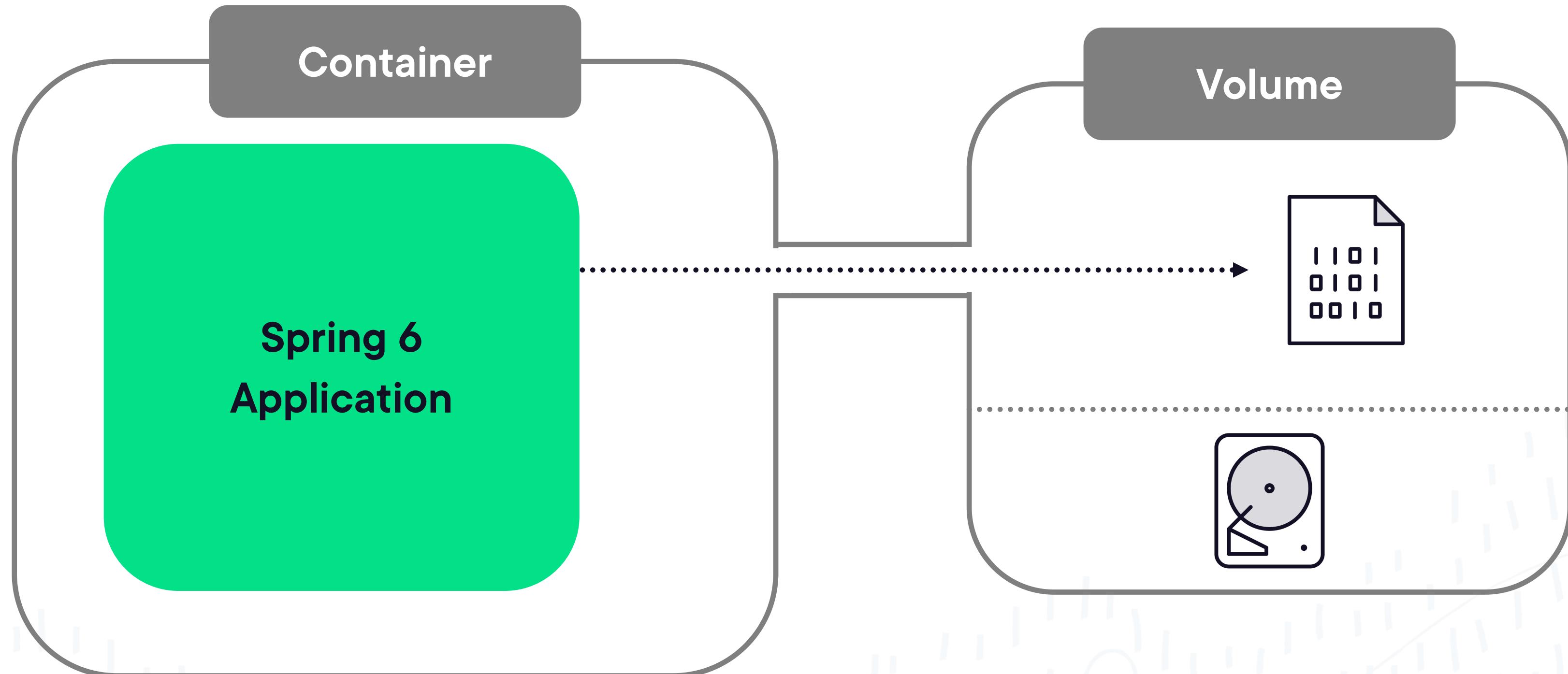
Persisting Storage



Persisting Storage



Persisting Storage



Demo



Use Docker Volumes to persist files



Docker compose



Deploying with Docker

1. Create Docker Network

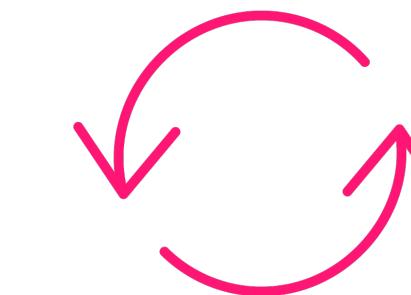
2. Deploy backing services (MySQL)

3. Create Volume

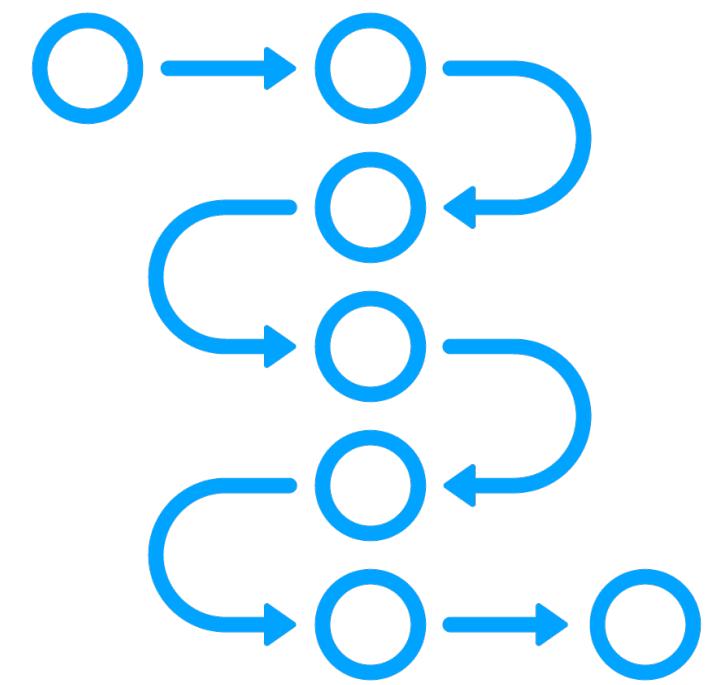
4. Set up environment variables

5. Define ports

6. Deploy Spring application



Docker Compose



`docker-compose.yml`



`.env`



Demo



**Describe deployment using Docker
compose**

Use .env files



Summary



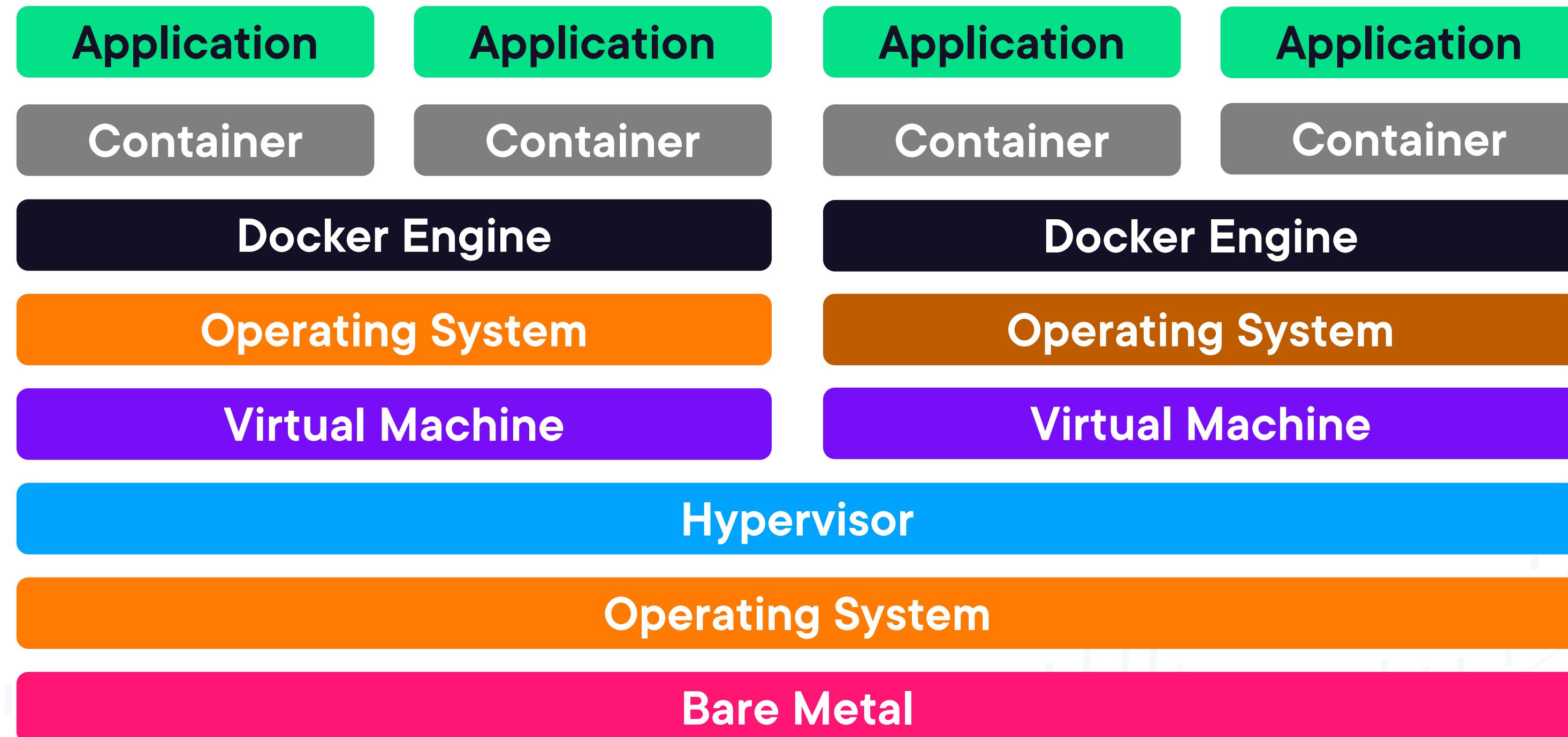
Package to Docker images

Deploy Docker containers

Docker compose



Containerized Applications



Externalized Configuration

Command Line Arguments

Java System properties

OS Environment Variables

@PropertySource annotations

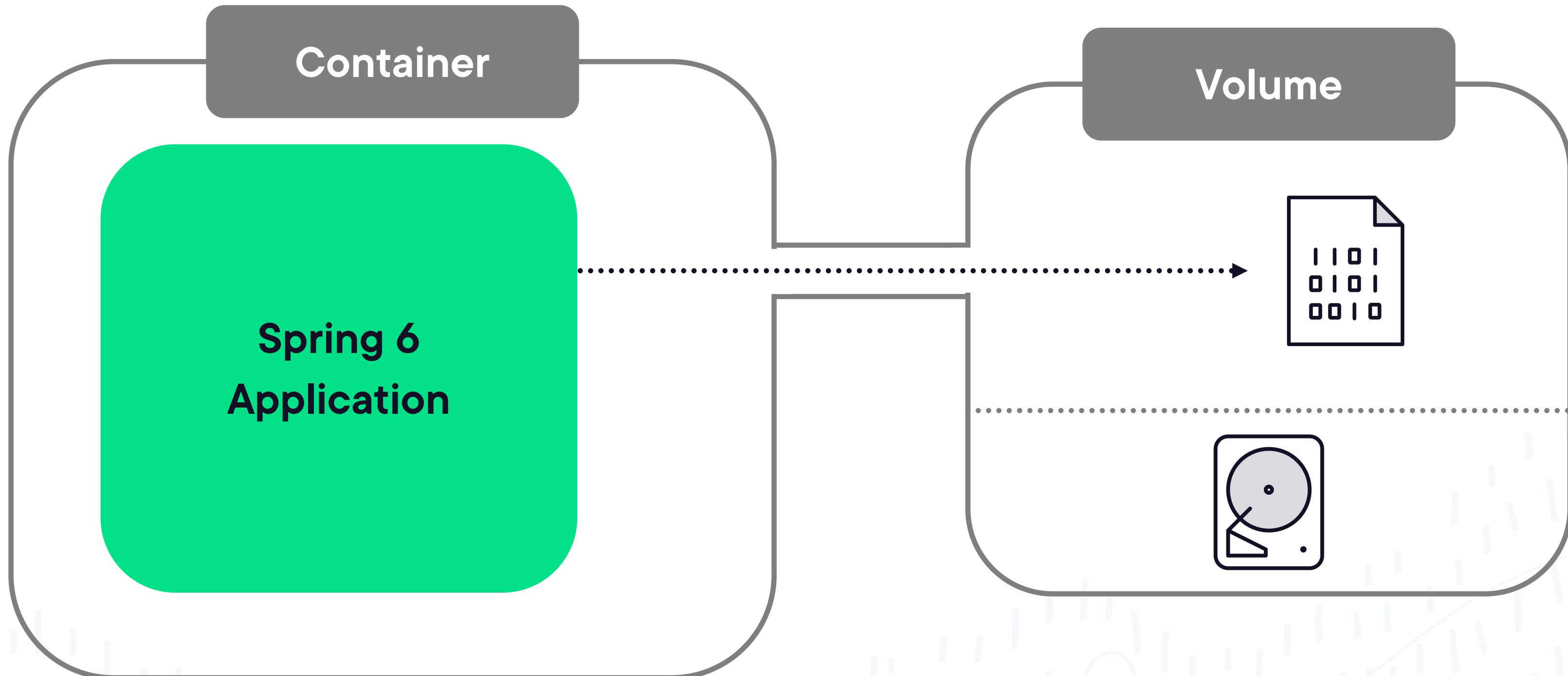
Application properties outside of your packaged jar

Application properties packaged inside your jar

Default Properties



Persisting Storage



Deploying with Docker

1. Create Docker Network

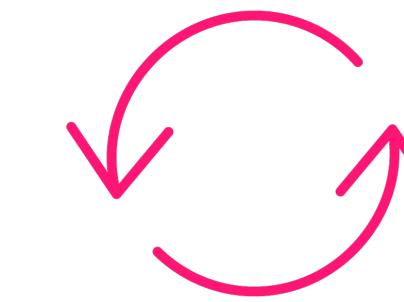
2. Deploy backing services (MySQL)

3. Create Volume

4. Set up environment variables

5. Define ports

6. Deploy Spring application



Up Next:

Orchestrating using Kubernetes



Get in Touch with Me



@BSucaciu

[linkedin.com/in/bogdan-sucaciu](https://www.linkedin.com/in/bogdan-sucaciu)

bsucaciu.com



Join the Spring Community

spring.io/events

