

JUG Schweiz, 12.06.2023

# Kubernetes Developer Survival Kit

Sandra Parsick

@SandraParsick

@sparsick@mastodon.social

mail@sandra-parsick.de

# Wer bin ich?

- Sandra Parsick
- Freiberuflicher Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Trainings
- Workshops

✉️ mail@sandra-parsick.de

🐦 @SandraParsick

Ⓜ️ @sparsick@mastodon.social

RSS https://www.sandra-parsick.de

🎧 https://ready-for-review.dev







NORTHERN JUSTICE  
MADEIRA

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Deployment Scripte

Konfiguration

Lokale Entwicklungsumgebung



# Friendly Reminder: 12 Factor App

## I. Codebase

One codebase tracked in revision control, many deploys

## II. Dependencies

Explicitly declare and isolate dependencies

## III. Config

Store config in the environment

## IV. Backing services

Treat backing services as attached resources

## V. Build, release, run

Strictly separate build and run stages

## VI. Processes

Execute the app as one or more stateless processes

# Friendly Reminder: 12 Factor App

## VII. Port binding

Export services via port binding

## VIII. Concurrency

Scale out via the process model

## IX. Disposability

Maximize robustness with fast startup and graceful shutdown

## X. Dev/prod parity

Keep development, staging, and production as similar as possible

## XI. Logs

Treat logs as event streams

## XII. Admin processes

Run admin/management tasks as one-off processes

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Debugging

CI

Konfiguration

Deployment Scripte

Was gehört alles ins Git Repository rein?

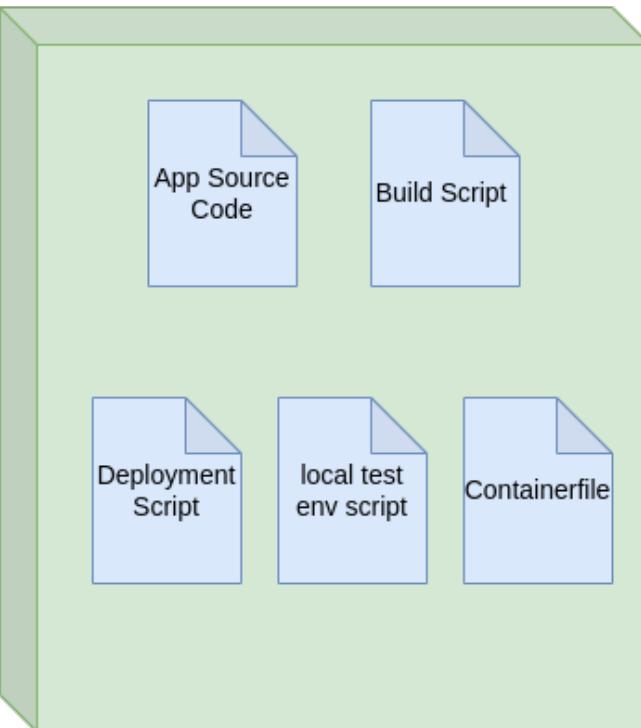
Lokale Entwicklungs umgebung

Kurzform: ALLES

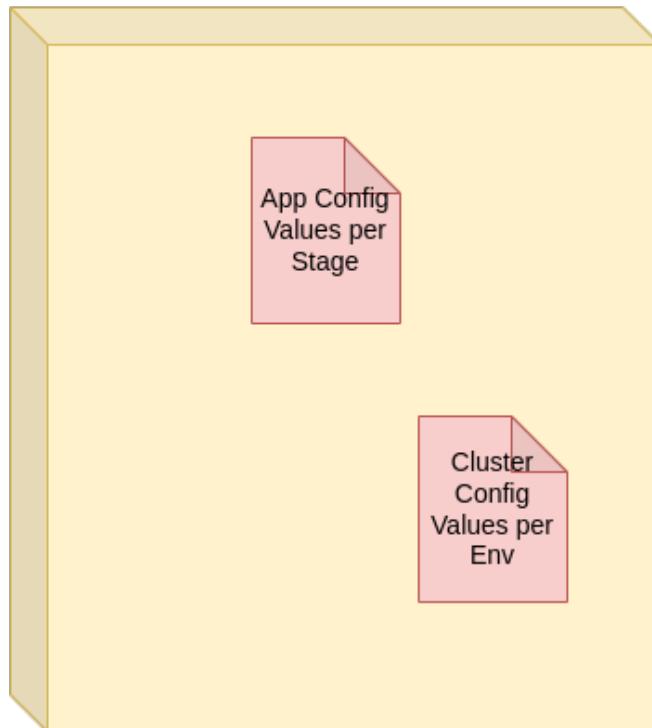
Eigentliche Fragestellung:  
Wieviele Repositories?

# Beispiel für eine Aufteilung

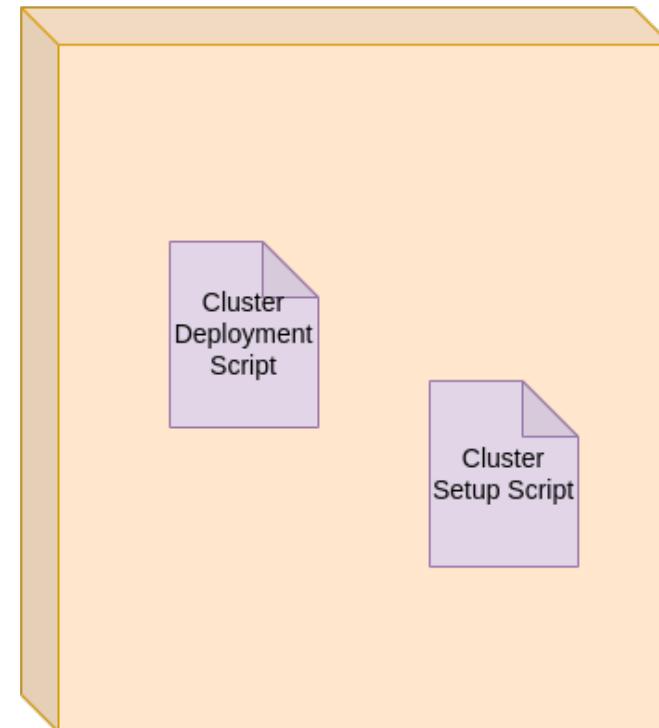
**Application Git Repository**



**Config Value Repository**

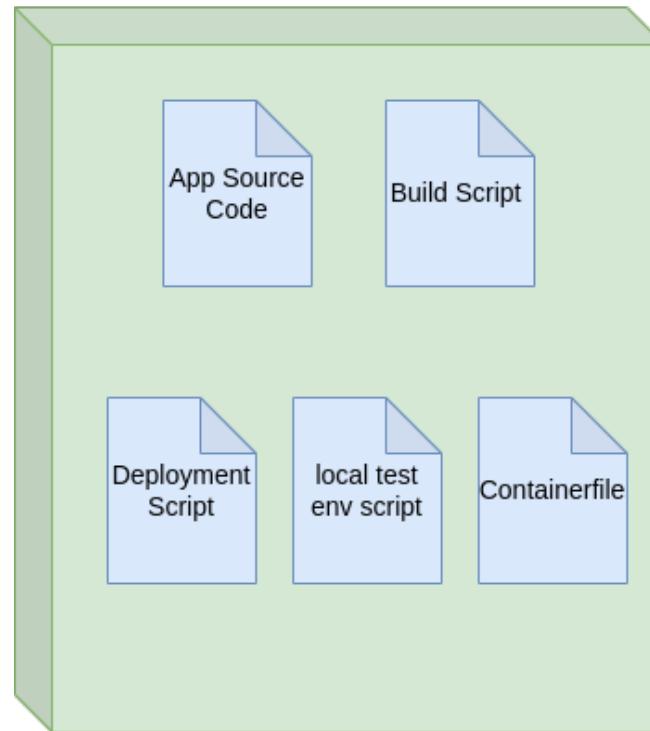


**Cluster Setup Script Repository**

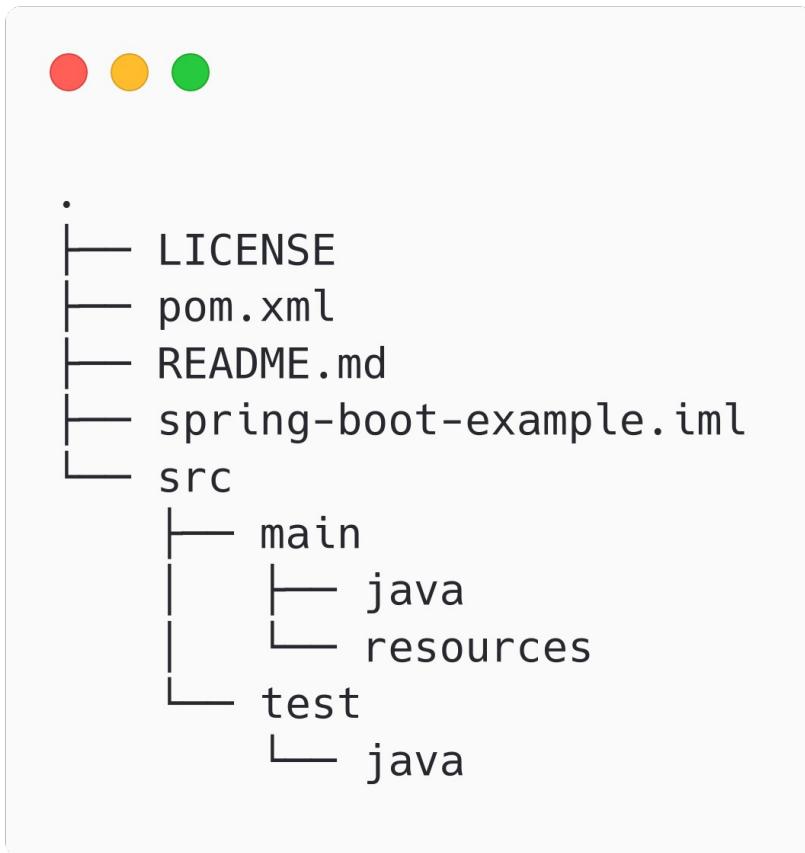


# Für Devs am wichtigsten

## Application Git Repository



# Ausgangspunkt einer Java App



## Technologiestack:

- Java 17
- Spring Boot 3.1.x
- Thymeleaf
- Apache Maven

App K8s-ready machen

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

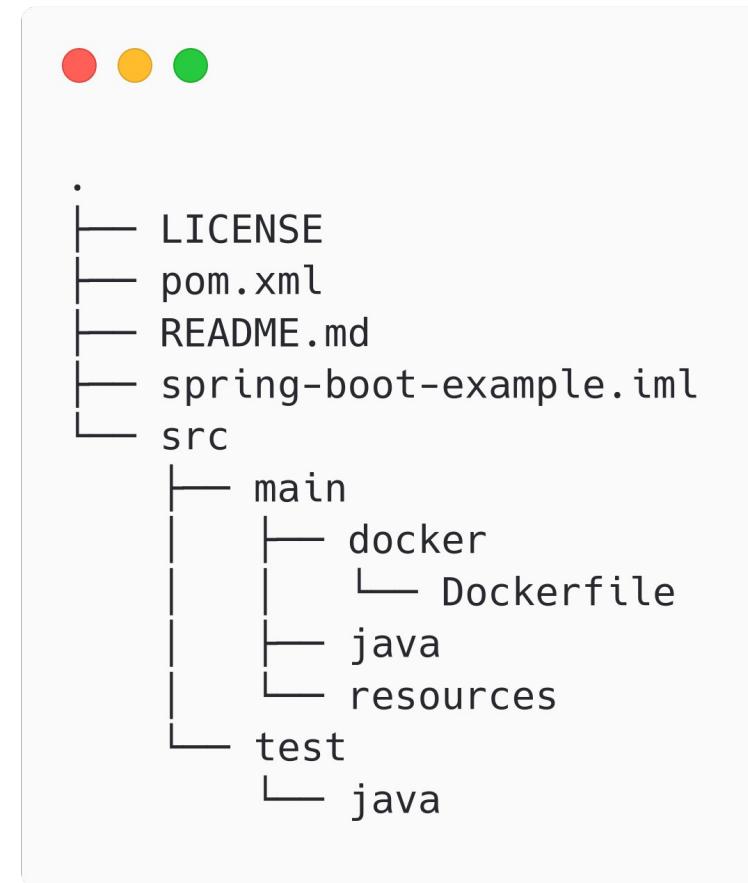
Backend / Frontend

Was gehört alles ins Git Repository rein?

Deployment Scripte

Lokale Entwicklungs umgebung

# Basis: Container



# Basis: Container



```
FROM docker.io/eclipse-temurin:17.0.1_12-jre as builder
WORKDIR /application
COPY maven/*.jar application.jar
RUN java -Djarmode=layer-tools -jar application.jar extract

FROM gcr.io/distroless/java17-debian11
WORKDIR /application
EXPOSE 8080
COPY --from=builder /application/dependencies/ ./
COPY --from=builder /application/spring-boot-loader/ ./
COPY --from=builder /application/snapshot-dependencies/ ./
COPY --from=builder /application/application/ ./
ENTRYPOINT ["java", "org.springframework.boot.loader.JarLauncher"]
```

```
<plugin>
    <groupId>io.fabric8</groupId>
    <artifactId>docker-maven-plugin</artifactId>
    <version>0.40.0</version>
    <executions>
        <execution>
            <id>docker-build</id>
            <goals>
                <goal>build</goal>
                <goal>push</goal>
            </goals>
        </execution>
    </executions>
    <configuration>
        <images>
            <image>
                <name>spring-boot-demo:latest</name>
                <build>
                    <dockerFile>Dockerfile</dockerFile>
                    <assembly>
                        <descriptorRef>artifact</descriptorRef>
                    </assembly>
                </build>
            </image>
        </images>
        <pushRegistry>localhost:6000</pushRegistry>
    </configuration>
</plugin>
```

# Alternativen

- Buildpacks (spring-boot-maven-plugin)
- JIB (jib-maven-plugin)
- Buildah
- Podman
- Weitere Infos im Artikel „Container-Images Deep Dive“ auf Informatik Aktuell

Container-Image-Bau ist Teil des Buildprozess  
und lokal ausführbar

# Good Practises Container Image Build

- unnötige Tools aus dem Image entfernen
- nur ein Service pro Image verpacken
- kleine Image bauen
- Build-Cache optimieren

- Eigene Container Registry benutzen
- Tags beim Releasen nur einmal verwenden

- Vulnerability-Scans der Container Images

# Optimierter Container Image



```
FROM docker.io/eclipse-temurin:17.0.1_12-jre as builder
WORKDIR /application
COPY maven/*.jar application.jar
RUN java -Djarmode=layer-tools -jar application.jar extract

FROM gcr.io/distroless/java17-debian11
WORKDIR /application
EXPOSE 8080
COPY --from=builder /application/dependencies/ ./
COPY --from=builder /application/spring-boot-loader/ ./
COPY --from=builder /application/snapshot-dependencies/ ./
COPY --from=builder /application/application/ ./
ENTRYPOINT ["java", "org.springframework.boot.loader.JarLauncher"]
```

# Container Registry

- Cloud Provider:
  - Azure Container Registry
  - AWS Elastic Container Registry
  - Google Container Registry
- On Premise:
  - JFrog Container Registry
  - Red Hat Quay
  - Harbor
  - Artifactory
  - Sonatype Nexus

# Vulnerability-Scans (Bsp.: Trivy)



```
→ trivy i --ignore-unfixed -o result spring-boot-demo:latest
2022-06-23T09:55:56.244+0200 INFO Vulnerability scanning is enabled
2022-06-23T09:55:56.245+0200 INFO Secret scanning is enabled
2022-06-23T09:55:56.245+0200 INFO If your scanning is slow, please try '--security-checks vuln' to disable secret scanning
2022-06-23T09:55:56.245+0200 INFO Please see also https://aquasecurity.github.io/trivy/v0.29.2/docs/secret/scanning/#recommendation for faster secret detection
2022-06-23T09:55:56.254+0200 INFO Detected OS: debian
2022-06-23T09:55:56.255+0200 INFO Detecting Debian vulnerabilities...
2022-06-23T09:55:56.265+0200 INFO Number of language-specific files: 1
2022-06-23T09:55:56.265+0200 INFO Detecting jar vulnerabilities...
```

spring-boot-demo:latest (debian 11.2)

=====

Total: 23 (UNKNOWN: 1, LOW: 2, MEDIUM: 6, HIGH: 6, CRITICAL: 8)

Java (jar)

=====

Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)

# Vulnerability-Scans (Bsp.: Trivy)



Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
libc6	CVE-2021-33574	CRITICAL	2.31-13+deb11u2	2.31-13+deb11u3	glibc: mq_notify does not handle separately allocated thread attributes <a href="https://avd.aquasec.com/nvd/cve-2021-33574">https://avd.aquasec.com/nvd/cve-2021-33574</a>
	CVE-2022-23218				glibc: Stack-based buffer overflow in svcunix_create via long pathnames <a href="https://avd.aquasec.com/nvd/cve-2022-23218">https://avd.aquasec.com/nvd/cve-2022-23218</a>
	CVE-2022-23219				glibc: Stack-based buffer overflow in sunrpc clnt_create via a long pathname <a href="https://avd.aquasec.com/nvd/cve-2022-23219">https://avd.aquasec.com/nvd/cve-2022-23219</a>
	CVE-2021-43396	LOW			glibc: conversion from ISO-2022-JP-3 with iconv may emit spurious NUL character on... <a href="https://avd.aquasec.com/nvd/cve-2021-43396">https://avd.aquasec.com/nvd/cve-2021-43396</a>



# Vulnerability-Scans Weitergedacht

- Was ist mit
  - Container in der Registry
  - Container, die schon im Cluster laufen

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

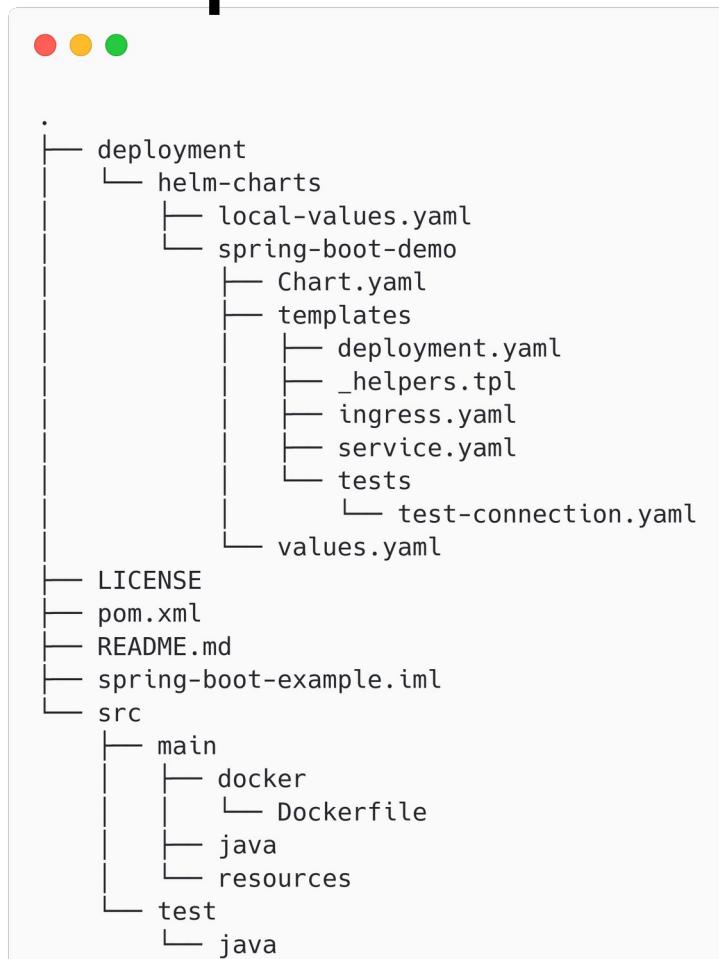
CI

Konfiguration

Deployment Scripte

Lokale Entwicklungsumgebung

# Next Step: Helm Charts

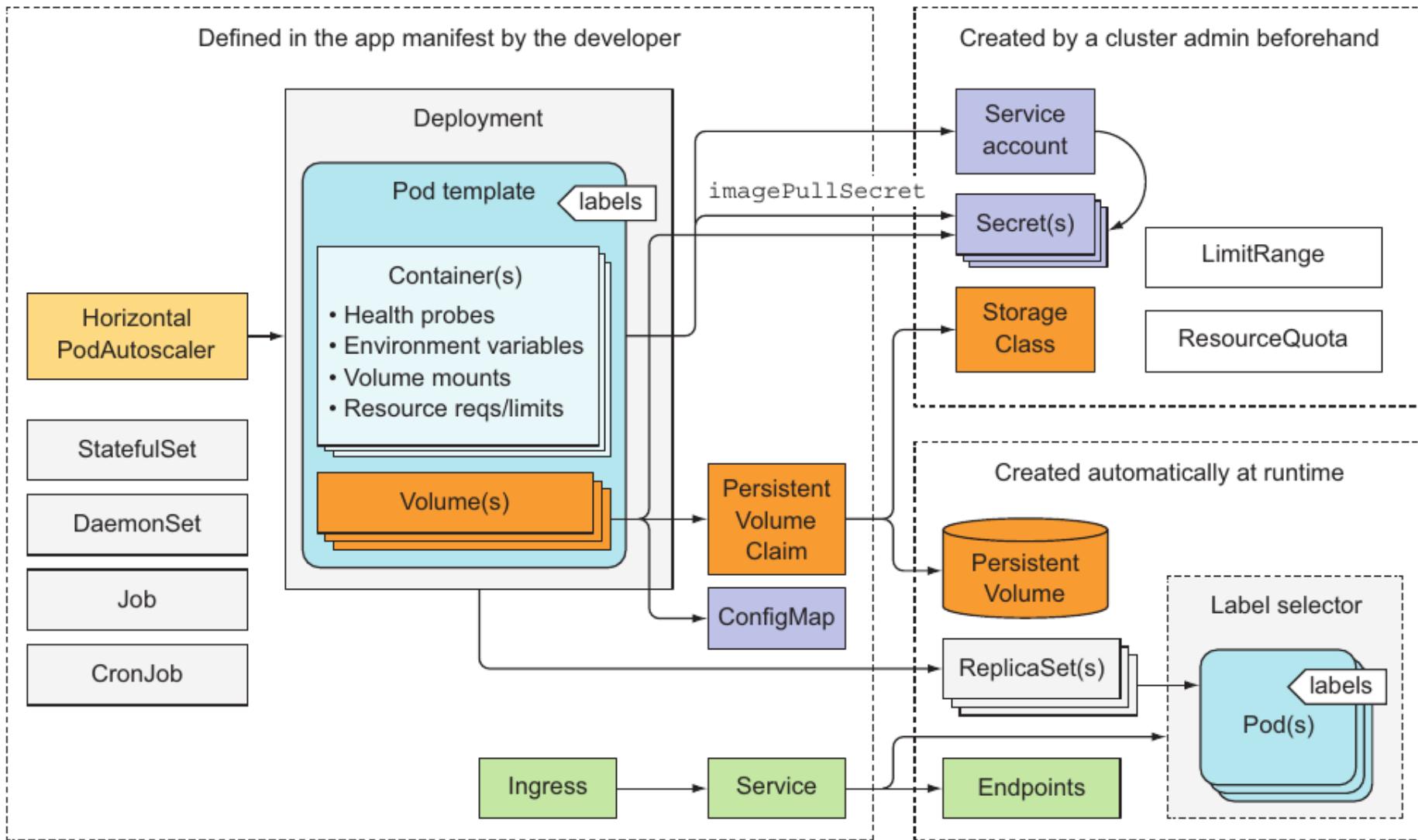


# Auszug: Service Definition



```
apiVersion: v1
kind: Service
metadata:
  name: {{ include "spring-boot-demo.fullname" . }}
  namespace: {{ include "spring-boot-demo.namespaceName" . }}
  labels:
    {{- include "spring-boot-demo.labels" . | nindent 4 }}
spec:
  type: {{ .Values.service.type }}
  ports:
    - port: {{ .Values.service.port }}
      targetPort: 8080
      protocol: TCP
      name: http
  selector:
    {{- include "spring-boot-demo.selectorLabels" . | nindent 4 }}
```

Um welche K8s Resource soll ich mich als Dev  
kümmern?





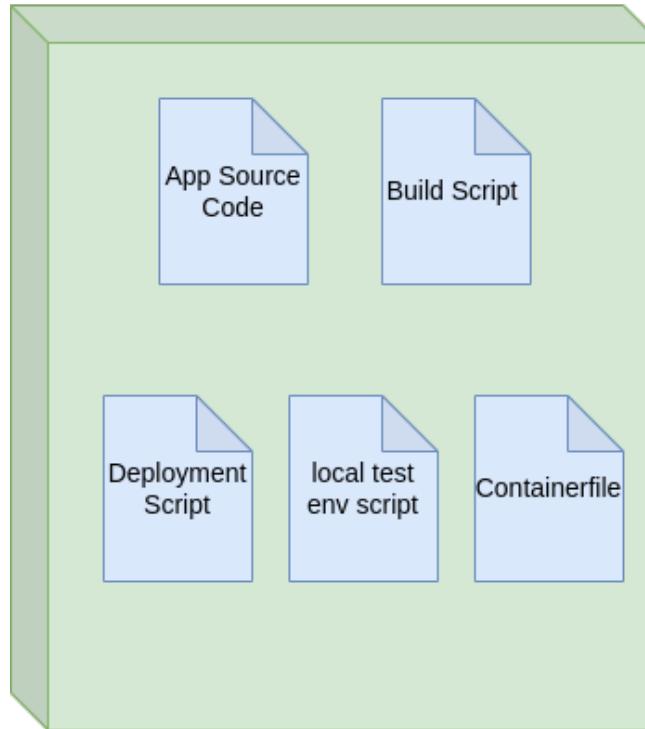
```
<plugin>
    <groupId>io.kokuwa.maven</groupId>
    <artifactId>helm-maven-plugin</artifactId>
    <version>6.3.0</version>
    <configuration>
        <chartDirectory>${project.basedir}/deployment/helm-charts</chartDirectory>
        <chartVersion>${project.version}</chartVersion>
        <helmVersion>3.8.1</helmVersion>
    </configuration>
    <executions>
        <execution>
            <id>build-chart</id>
            <phase>package</phase>
            <goals>
                <goal>package</goal>
            </goals>
        </execution>
        <execution>
            <id>upload-chart</id>
            <phase>deploy</phase>
            <goals>
                <goal>upload</goal>
            </goals>
        </execution>
    </executions>
</plugin>
```

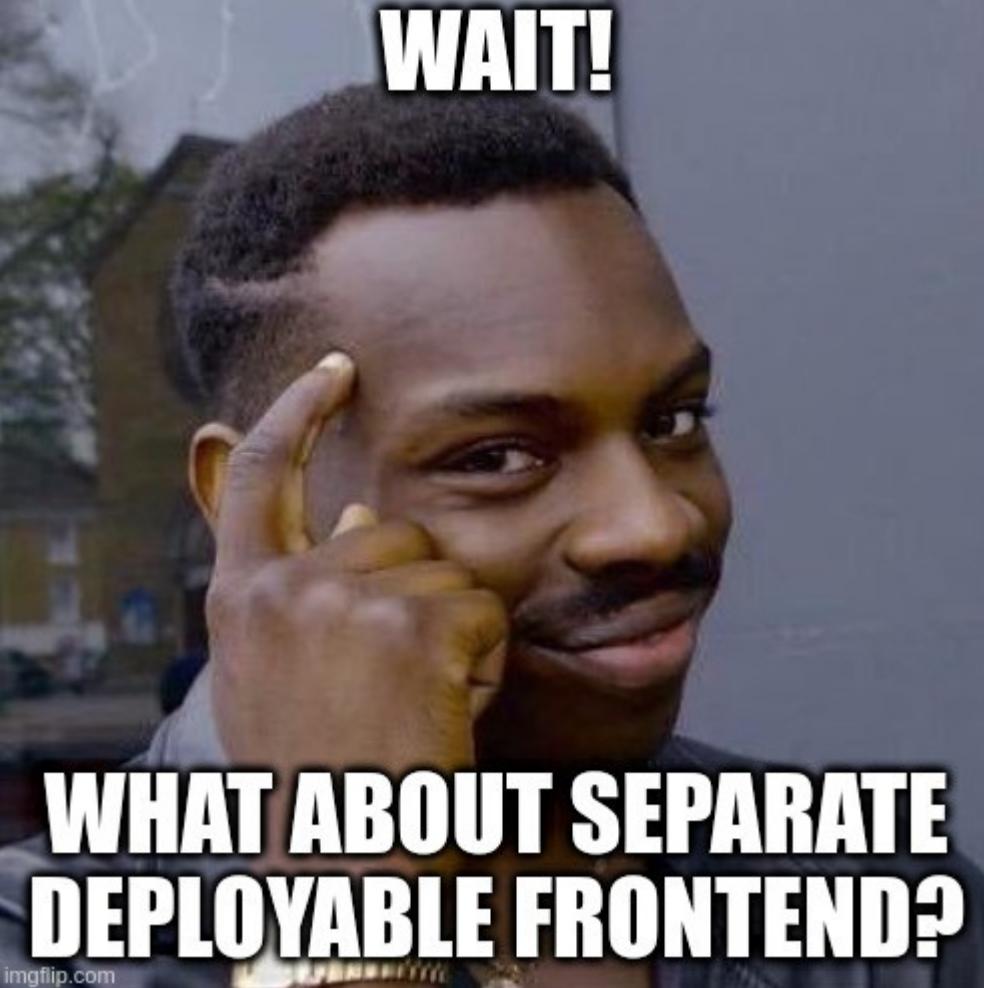
# Helm Charts Paketierung Teil des Build Prozesses

# Helm Chart Repository

- Allgemein:
  - Jede Container Registry kann dafür genutzt werden
- Darauf spezialisiert:
  - Chartmuseum
  - JFrog Container Registry
  - Artifactory
  - Sonatype Nexus

## **Application Git Repository**





**WAIT!**

**WHAT ABOUT SEPARATE  
DEPLOYABLE FRONTEND?**

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

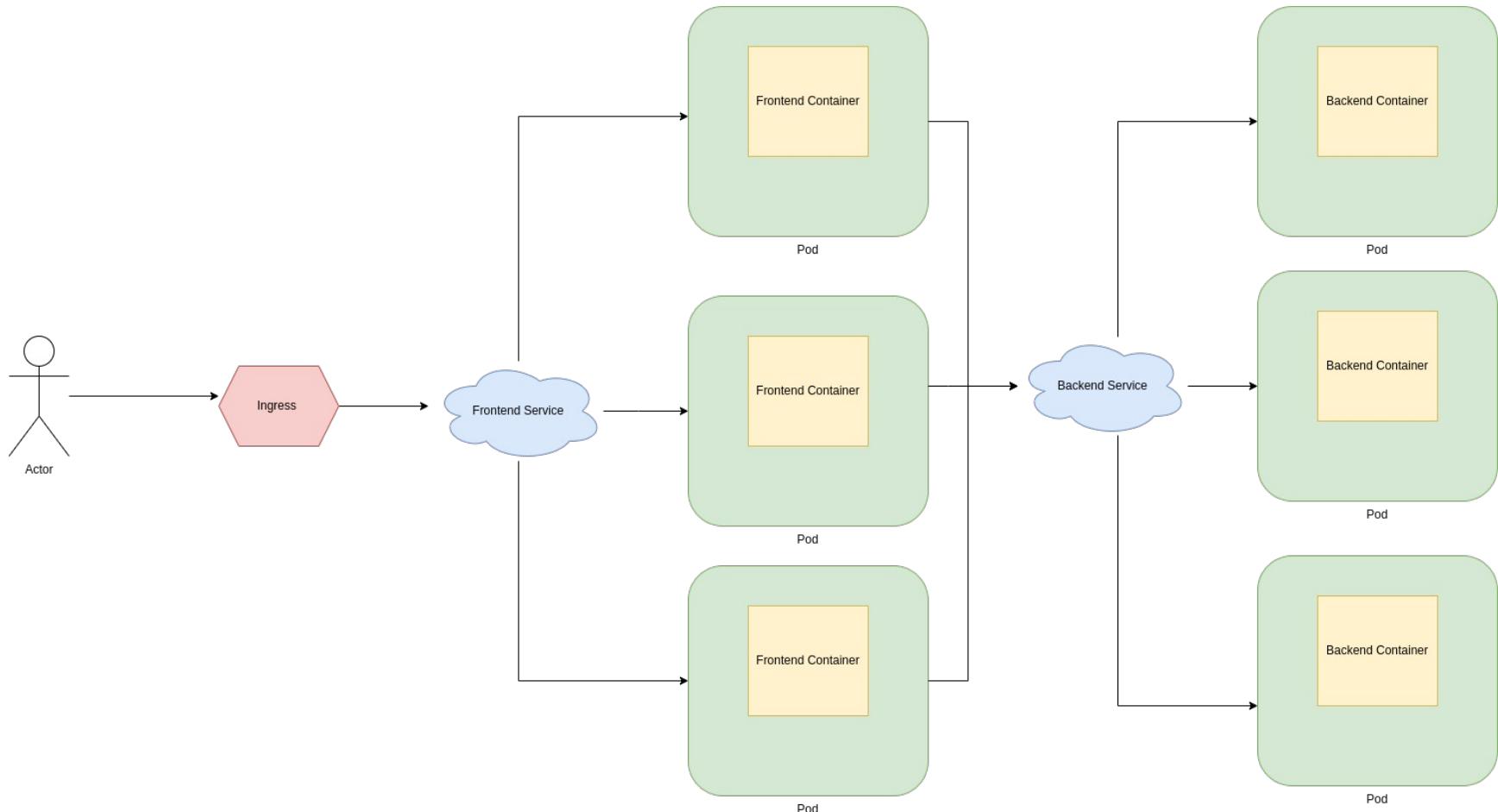
CI

Konfiguration

Deployment Scripte

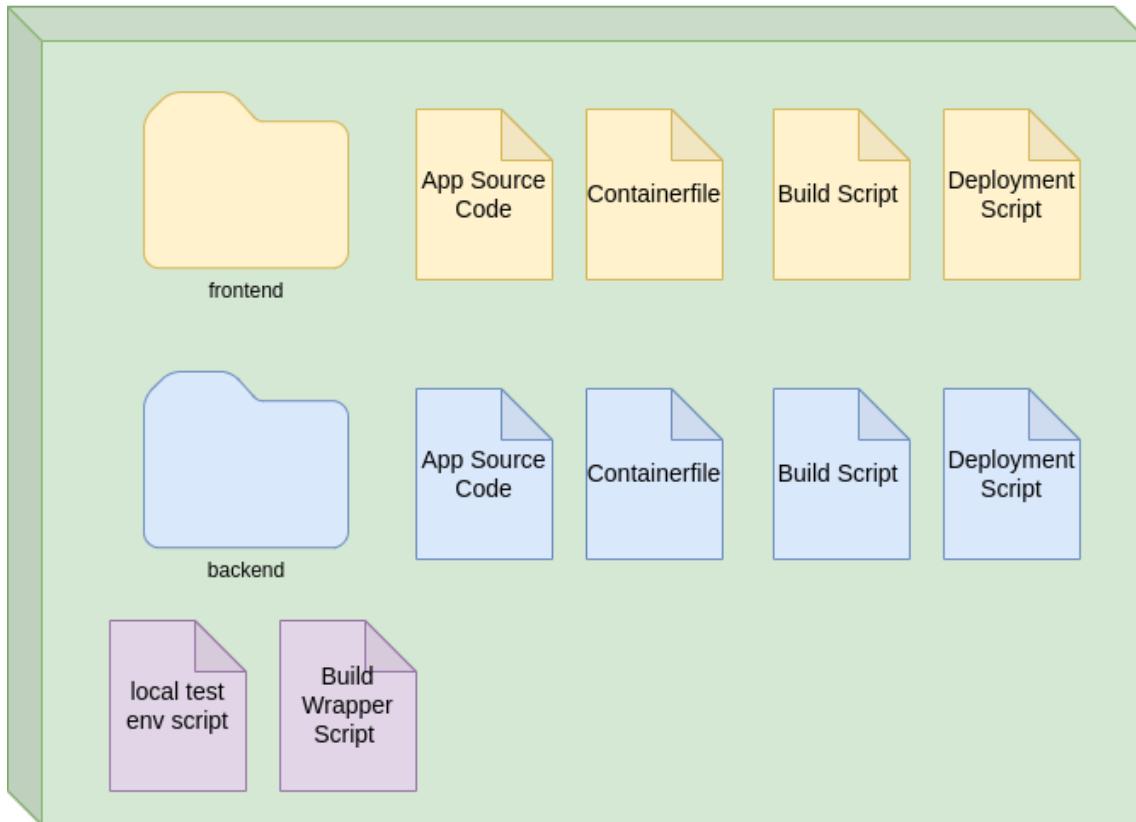
Lokale Entwicklungsumgebung

# Frontend und Backend in K8s



# Git Repository Struktur

## Application Git Repository



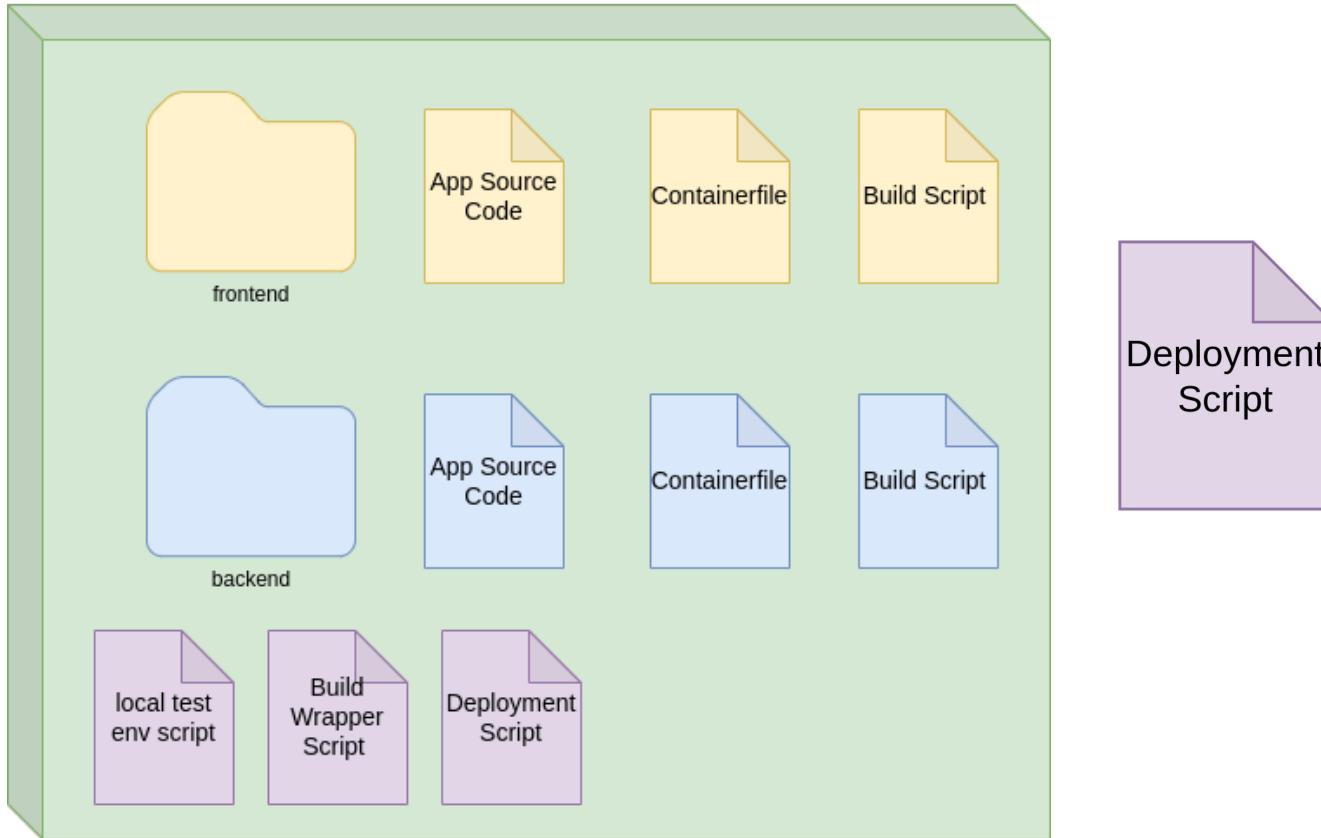
- Ingress
- Frontend Service
- Frontend Deployment



- Backend Service
- Backend Deployment

# Git Repository Struktur

## Application Git Repository



- Ingress
- Frontend Service
- Frontend Deployment
- Backend Service
- Backend Deployment

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

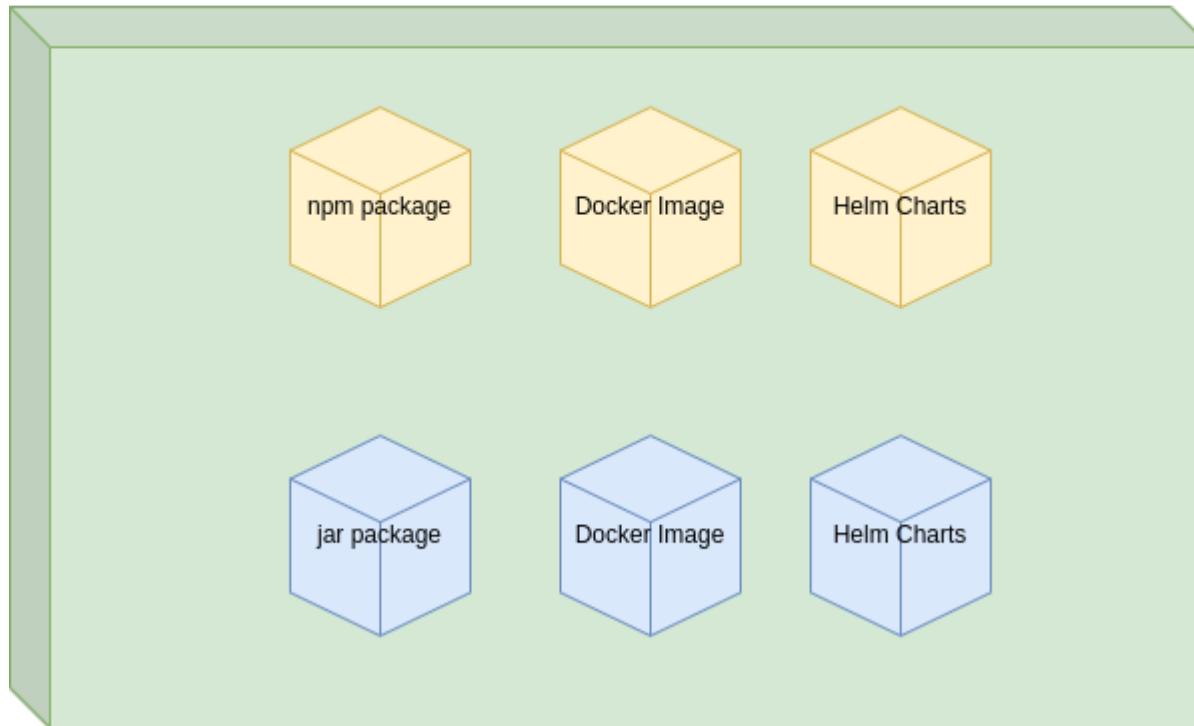
Konfiguration

Deployment Scripte

Lokale Entwicklungs umgebung

Fangt einfach an:  
Eine Versionsnummer über alle Artifakte

# Application Artifacts



App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Konfiguration

Deployment Scripte

Lokale Entwicklungs umgebung

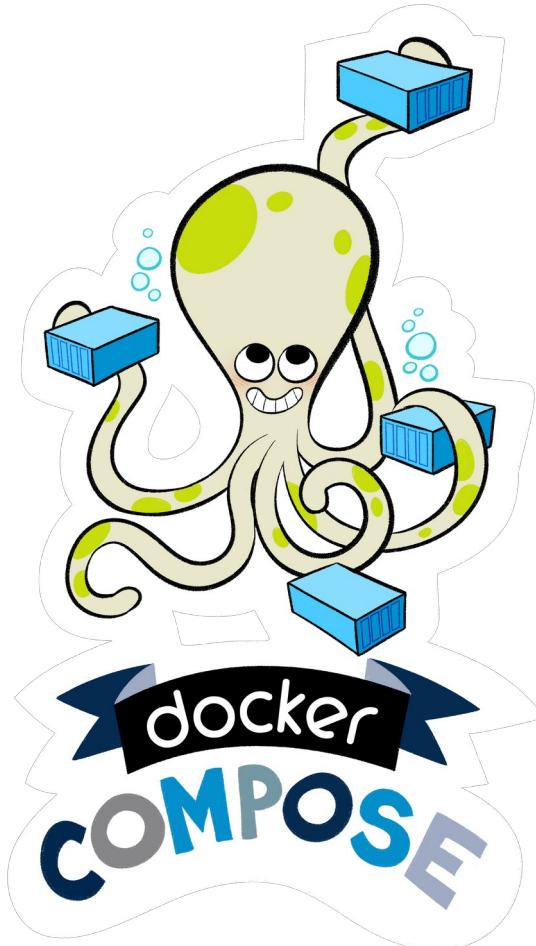


Applikation  
lokal testen



Deployment  
Skripte lokal  
entwickeln

# Applikation lokal testen



Spring Boot Maven Plugin

# Applikation lokal testen



```
version: "3.9"
services:
  database:
    image: mongo:4.2.21
    restart: always
    ports:
      - 27017:27017
  environment:
    MONGO_INITDB_ROOT_USERNAME: root
    MONGO_INITDB_ROOT_PASSWORD: root123
  volumes:
    - ./local-env/:/dockerentrypoint-initdb.d/
```



```
<plugin>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-maven-plugin</artifactId>
  <configuration>
    <layers>
      <enabled>true</enabled>
    </layers>
    <environmentVariables>
      <MONGODB_ENABLED>true</MONGODB_ENABLED>
      <MONGODB_URI>mongodb://test:test123@localhost/test</MONGODB_URI>
    </environmentVariables>
  </configuration>
</plugin>
```



```
version: "3.9"
services:
  demo-app:
    image: spring-boot-demo:latest
    restart: always
    ports:
      - 80:8080
    environment:
      MONGODB_ENABLED: "true"
      MONGODB_URI: mongodb://test:test123@database:27017/test
    depends_on:
      - database

  database:
    image: mongo:4.2.21
    restart: always
    ports:
      - 27017:27017
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: root123
    volumes:
      - ./local-env/:/docker-entrypoint-initdb.d/
```

# Spring Boot Docker Compose Support

```
<!-- Since Spring Boot 3.1 -->
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-docker-compose</artifactId>
        <optional>true</optional>
    </dependency>
</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
            <configuration>
                <layers>
                    <enabled>true</enabled>
                </layers>
            </configuration>
        </plugin>
    </plugins>
</build>
```



```
# application.properties
# default: compose.yml
spring.docker.compose.file=docker-compose.yml
```



Andere  
Abhängigkeiten?

# Mocking

- <https://www.mock-server.com>
- <https://github.com/navikt/mock-oauth2-server>



```
version: "3.9"
services:
  mockserver:
    image: mockserver/mockserver:latest
    restart: always
    ports:
      - 1080:1080
    environment:
      MOCKSERVER_INITIALIZATION_JSON_PATH: /config/expectation.json
    volumes:
      - ./local-env/mockserver:/config
```



```
[  
  {  
    "httpRequest": {  
      "path": "/success"  
    },  
    "httpResponse": {  
      "body": "Successful!"  
    }  
  },  
  {  
    "httpRequest": {  
      "path": "/fail"  
    },  
    "httpResponse": {  
      "statusCode": 400  
    }  
  }  
]
```

# Deployment Skripte lokal entwickeln



minikube

# Alternativen zu Minikube

- k3s
- k3d
- kind
- microk8s
- k0s

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Deployment Scripte

Konfiguration

Lokale Entwicklungs umgebung

# 12 Factor App: Die Konfiguration in Umgebungsvariablen ablegen

# Applikation vorbereiten



# snippet application.properties

```
spring.data.mongodb.uri=${MONGODB_URI:mongodb://localhost/test}
```

```
mongodb.enabled=${MONGODB_ENABLED:false}
```

# Helm Charts anpassen



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "spring-boot-demo.fullname" . }}-config
  namespace: {{ include "spring-boot-demo.namespaceName" . }}
  labels:
    {{- include "spring-boot-demo.labels" . | nindent 4 }}
data:
  MONGODB_URI: "{{ .Values.mongodb.uri }}"
  MONGODB_ENABLED: "{{ .Values.mongodb.enabled }}"
```

# Helm Charts anpassen



```
# code snippet with the important part
apiVersion: apps/v1
kind: Deployment
# ...
spec:
  template:
    metadata:
      annotations:
        checksum/config: {{ include (print $.Template.BasePath "/config.yaml") . | sha256sum }}
  spec:
    containers:
      - name: {{ .Chart.Name }}
        image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
        args: [{{ .Values.spring_boot_demo_chart.container_args }}]
        imagePullPolicy: {{ .Values.image.pullPolicy }}
    envFrom:
      - configMapRef:
          name: {{ include "spring-boot-demo.fullname" . }}-config
```

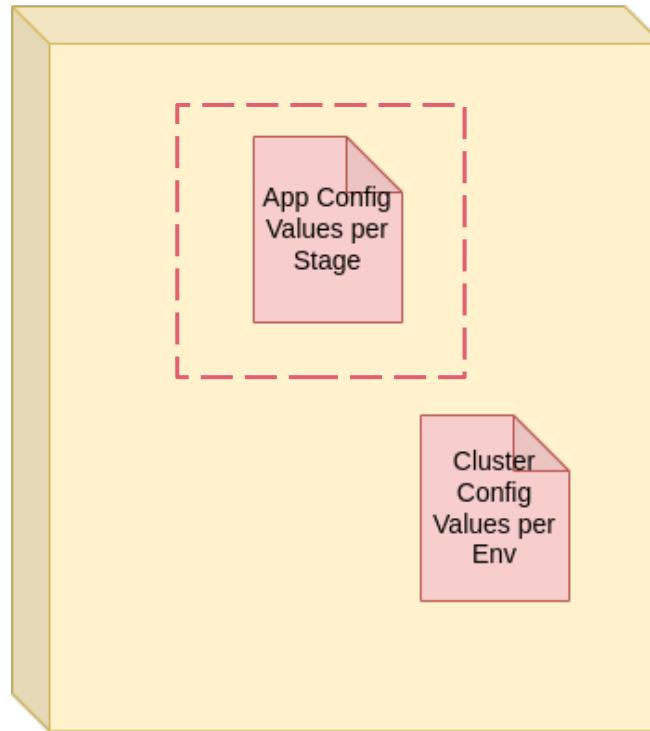
# Helm Charts anpassen



```
# code snippet with the important part from value.yaml
mongodb:
  enabled: false
  uri: mongodb://test:test@localhost/test
```

# Konfiguration verwalten

## Config Value Repository



# Konfiguration verwalten



config-value-repo on ✘ dev  
→ tree

```
.
```

- └── namespace-a
  - └── app1.yml
  - └── registry.yaml

→ git branch  
\* dev  
 pre-prod  
 prod

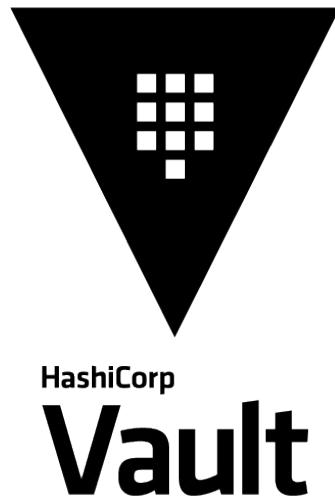


flat-config-value-repo on ✘ master  
→ tree

```
.
```

- └── dev
  - └── namespace-a
    - └── app1.yml
    - └── registry.yaml
- └── pre-prod
  - └── namespace-a
    - └── app1.yml
    - └── registry.yaml
- └── prod
  - └── namespace-a
    - └── app1.yml
    - └── registry.yaml

# Secrets



Cloud Lösungen (Bsp):

- Google Secret Manager
- AWS Secrets & Configuration Provider
- Azure Key Vault Provider

# Helm Secret Plugin



```
→ helm plugin install https://github.com/jkroepke/helm-secrets --version v3.12.0  
→ helm secrets help
```

Secrets encryption `in` Helm Charts

This plugin provides ability to encrypt/decrypt secrets files to store `in` less secure places, before they are installed using Helm.

For more information, see the README at [github.com/jkroepke/helm-secrets](https://github.com/jkroepke/helm-secrets)

To decrypt/encrypt/edit you need to initialize/first encrypt secrets with sops - <https://github.com/mozilla/sops>

# Helm Secret Plugin



```
// sops must be configured
→ helm secrets enc examples/sops/secrets.yaml
Encrypting examples/sops/secrets.yaml
Encrypted examples/sops/secrets.yaml
→ helm upgrade name . -f secrets://examples/sops/secrets.yaml value.yaml
```

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Deployment Scripte

Konfiguration

Lokale Entwicklungs umgebung

# Good Practices für Anwendungen in Container

- Nur ein Anwendungsprozess pro Container
- Ausführung als root vermeiden
- Privilegierte Container vermeiden
- Zustandslose Anwendungen bevorzugen
- Log-Nachrichten auf stdout
- Anwendungsüberwachung bedenken
- Robust hoch- und runterfahren können

# Log-Nachrichten auf stdout



Tipp: Nutzt Spring Default Logging Settings

# Anwendungsüberwachung



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
<dependency>
    <groupId>io.micrometer</groupId>
    <artifactId>micrometer-registry-prometheus</artifactId>
</dependency>
```



```
management.metrics.export.prometheus.enabled=true
management.metrics.web.server.request.autotime.enabled=true
management.endpoints.web.exposure.include=prometheus
```

# Robust hoch- und runterfahren



```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```



```
management.endpoints.web.exposure.include=info,health
```

# Robust hoch- und runterfahren

```
# code snippet with the important part
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag }}"
          ports:
            - name: container-http
              containerPort: 8080
              protocol: TCP
      livenessProbe:
        httpGet:
          path: /actuator/health/liveness
          port: container-http
        initialDelaySeconds: {{ .Values.livenessProbe.initialDelaySeconds }}
        periodSeconds: {{ .Values.livenessProbe.periodSeconds }}
        timeoutSeconds: {{ .Values.livenessProbe.timeoutSeconds }}
      readinessProbe:
        httpGet:
          path: /actuator/health/readiness
          port: container-http
        initialDelaySeconds: {{ .Values.readinessProbe.initialDelaySeconds }}
        periodSeconds: {{ .Values.readinessProbe.periodSeconds }}
        timeoutSeconds: {{ .Values.readinessProbe.timeoutSeconds }}
```

# Robust hoch- und runterfahren

Wichtig:  
Sichert diese Endpunkte nach außen ab!

# Robust hoch- und runterfahren

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ include "spring-boot-demo.fullname" . }}
  namespace: {{ include "spring-boot-demo.namespaceName" . }}
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
    nginx.ingress.kubernetes.io/x-forwarded-prefix: "/"
    nginx.ingress.kubernetes.io/server-snippet: |
      location ~* "^/actuator/" {
        deny all;
        return 403;
      }
spec:
  rules:
    - host: {{ .Values.ingress.host }}
      http:
        paths:
          - path: /(.*)
            pathType: Prefix
            backend:
              service:
                name: {{ include "spring-boot-demo.fullname" . }}
                port:
                  number: 8080
```

# JVM und Kubernetes

- JVM Default Einstellung (welcher GC oder default max Heap Size) abh. von Umgebung und Java Version
- Lesson Learnt: Immer die JVM konfigurieren
- Don't worry: Es gibt Empfehlungen

# JVM und Kubernetes

- Topologie des Kubernetes Cluster ist abhängig von deiner Anwendung
  - Kleine JVMs + viele Replicas vs Große JVMs + wenige Replicas
- Messen, Messen, Messen

# Weitere Infos zu JVM und Kubernetes

- Secrets of Performance Tuning Java on Kubernetes von Bruno Borges
- We Moved one Java Product to Kubernetes and This Is What We Learned von Carlos Sanchez
- Tuning and Optimizing Java Garbage Collection von Monica Beckwith

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

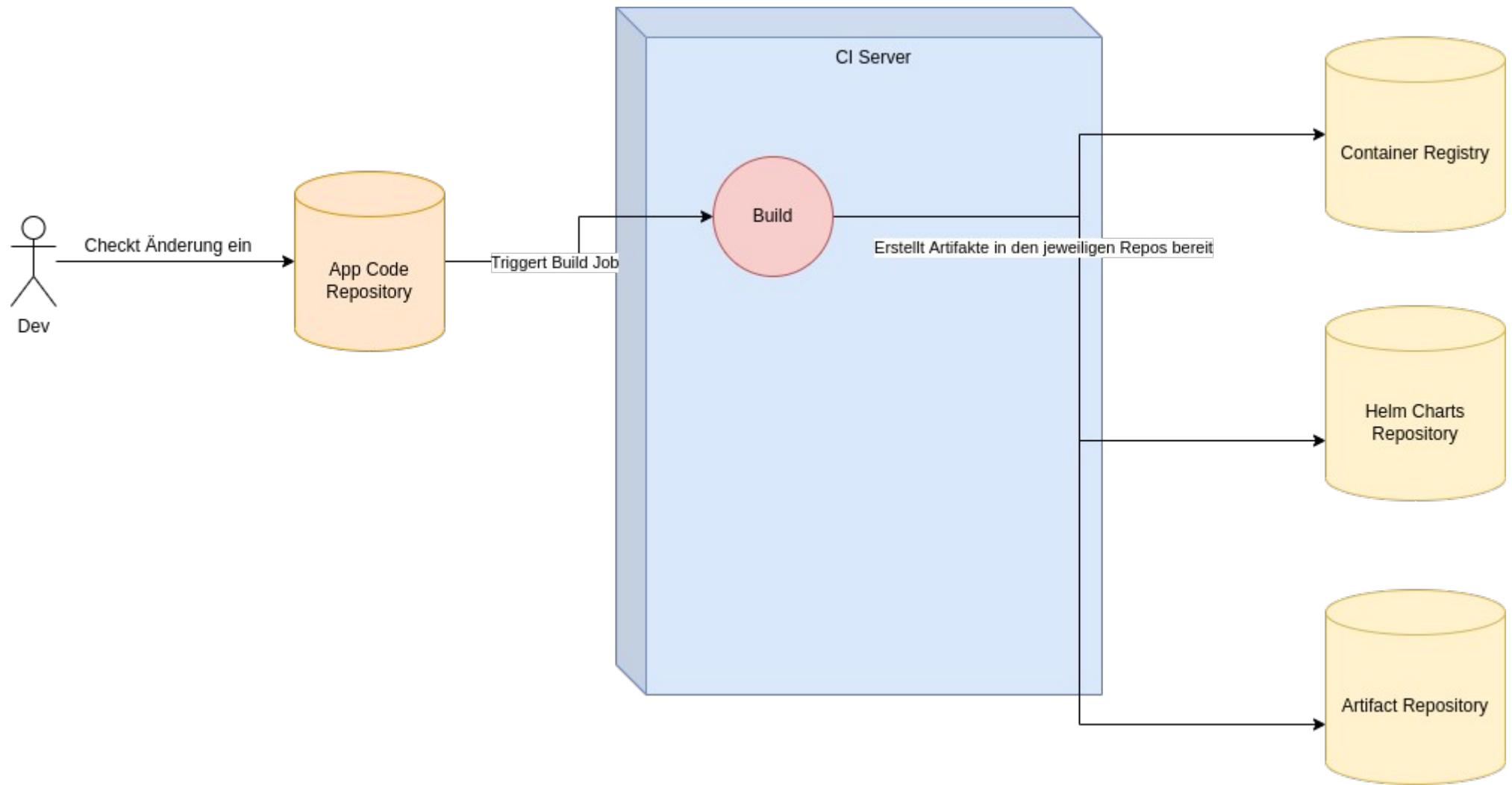
Deployment Scripte

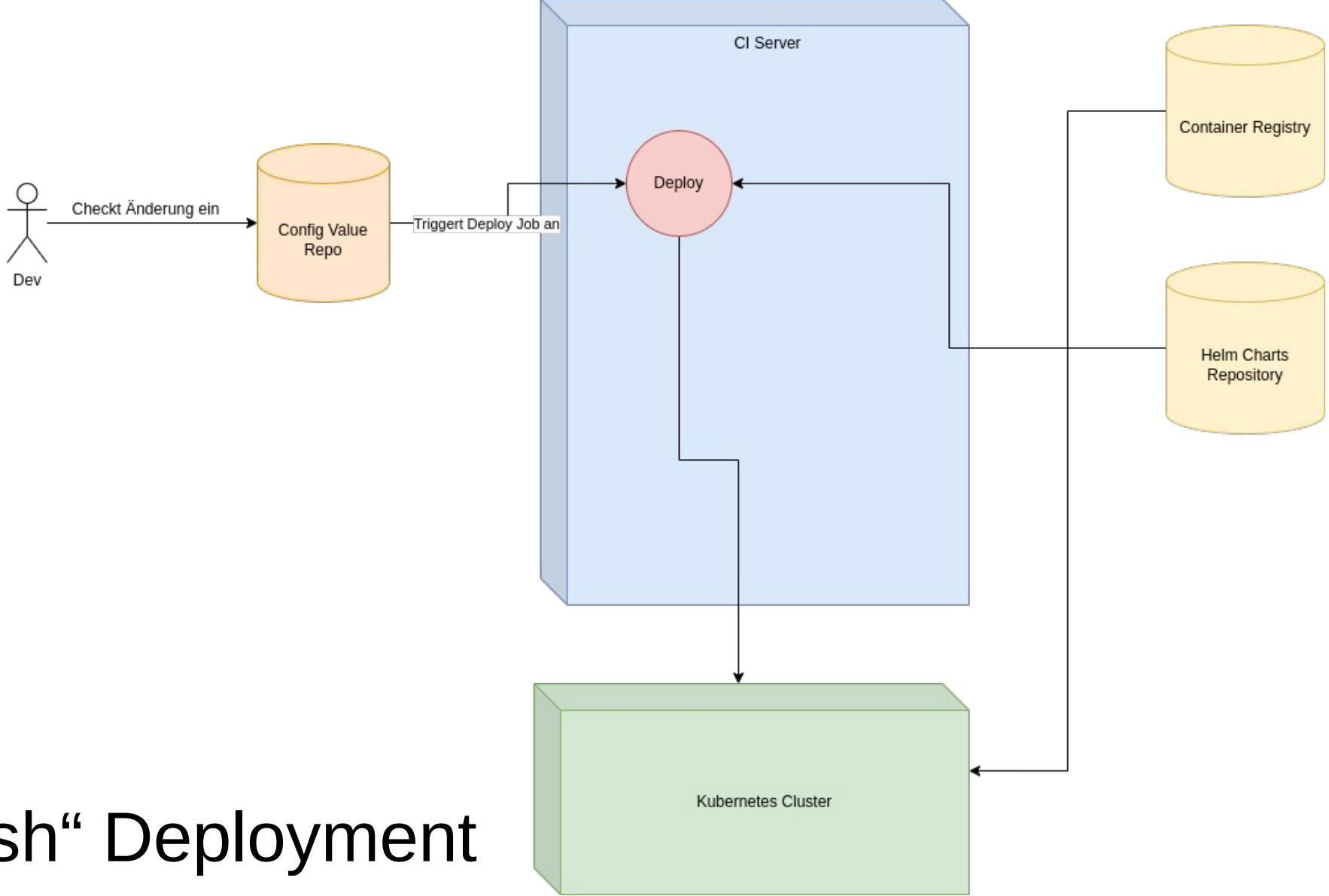
Konfiguration

Lokale Entwicklungs umgebung

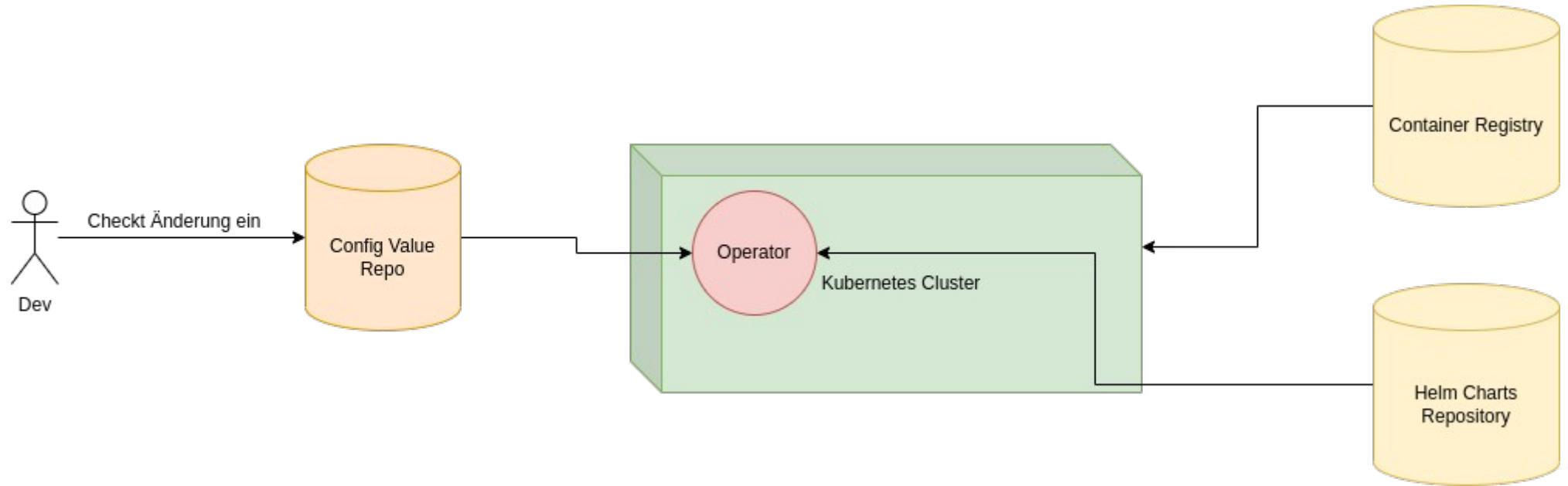
12 Factor App:  
Build- und Run-Phase strikt trennen

# Build





# „Push“ Deployment



# „Pull“ Deployment

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Deployment Scripte

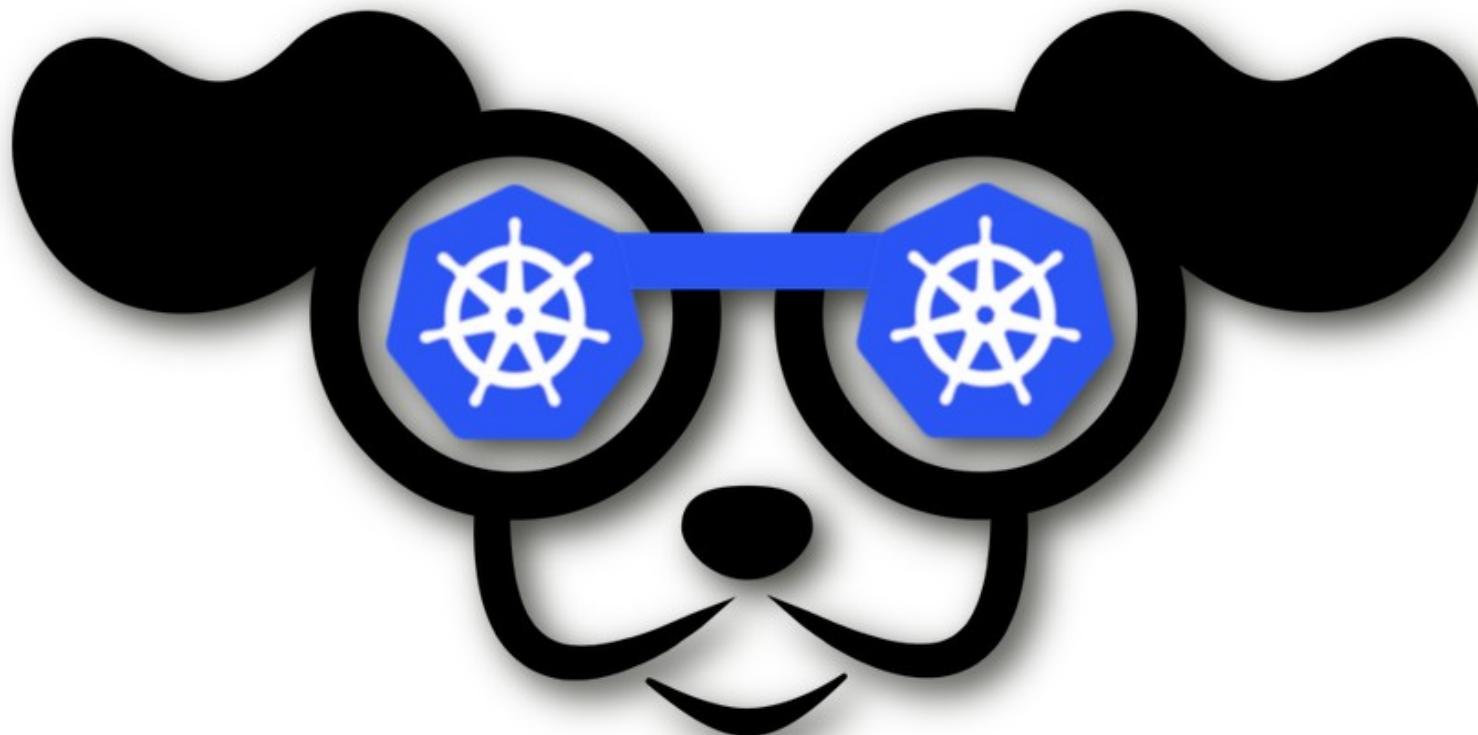
Konfiguration

Lokale Entwicklungs umgebung

kubectl

# k9s

*Kubernetes CLI To Manage Your Clusters In Style!*



K8s Lens / Open Lens

The screenshot shows a user interface for managing clusters, likely in a Kubernetes environment. The top right corner displays the user's profile information: SP @sparsick Personal Space. The main area is titled "Clusters" and shows one item: "minikube". The table columns are "Name", "Source", "Labels", and "Status". The "Name" column lists "minikube". The "Source" column shows "local". The "Labels" column contains "file=~/kube/config". The "Status" column indicates "connected". A search bar at the top right allows for filtering the results. On the left, a sidebar titled "Catalog" contains a "Browse" section and a "CATEGORIES" section. The "CATEGORIES" section is expanded, showing "General", "Clusters" (which is selected and highlighted in blue), and "Web Links". Below these are "Dev Clusters" and a "NEW" button. At the bottom right of the main area is a blue circular button with a white plus sign. The bottom of the screen features a blue navigation bar with icons for back, forward, and other system functions.

Name	Source	Labels	Status
minikube	local	file=~/kube/config	connected

# Monokle Desktop

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

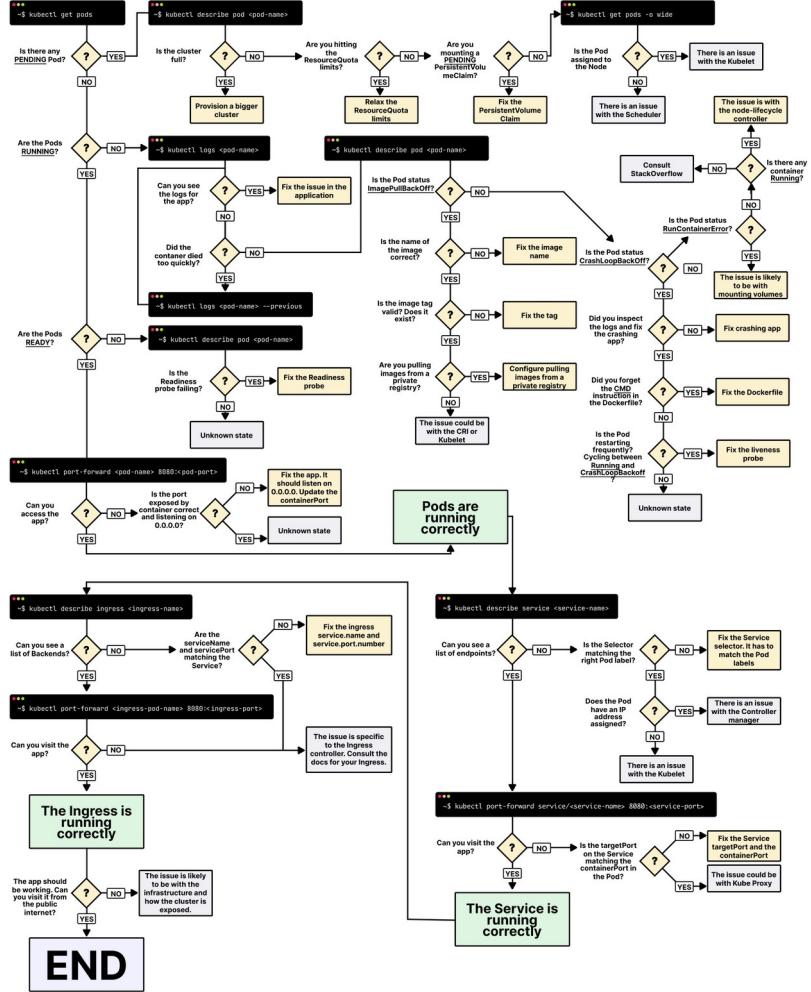
CI

Deployment Scripte

Konfiguration

Lokale Entwicklungs umgebung

# START



<https://learnk8s.io/troubleshooting-deployments>

# Troubleshooting Applications

This doc contains a set of resources for fixing issues with containerized applications. It covers things like common issues with Kubernetes resources (like Pods, Services, or StatefulSets), advice on making sense of container termination messages, and ways to debug running containers.

---

[Debug Pods](#)

[Debug Services](#)

[Debug a StatefulSet](#)

[Debug Init Containers](#)

[Debug Running Pods](#)

[Determine the Reason for Pod Failure](#)

<https://kubernetes.io/docs/tasks/debug/debug-application/>

[Get a Shell to a Running Container](#)

# debug container (K8s v1.23)



```
$ kubectl run ephemeral-demo --image=k8s.gcr.io/pause:3.1 --restart=Never
$ kubectl exec -it ephemeral-demo -- sh
OCI runtime exec failed: exec failed: container_linux.go:346: starting container process caused "exec:
\"sh\": executable file not found in $PATH": unknown

$ kubectl debug -it ephemeral-demo --image=busybox:1.28 --target=ephemeral-demo
Defaulting debug container name to debugger-8xzrl.
If you don't see a command prompt, try pressing enter.
/ #
```

App K8s-ready machen

Wie sehe ich was im Cluster los ist?

Backend / Frontend

Versionierung

Container Images

Was gehört alles ins Git Repository rein?

Debugging

CI

Konfiguration

Deployment Scripte

Lokale Entwicklungs umgebung

Fragen?

[mail@sandra-parsick.de](mailto:mail@sandra-parsick.de)

@SandraParsick

@sparsick@mastodon.social

<https://github.com/sparsick/k8s-dev-survival-kit-talk>

**architektur SPICKER**

Übersichten für die konzeptionelle Seite der Softwareentwicklung

**embarc Software Consulting GmbH**

**SIGS DATACOM**

# Container-Anwendungen entwickeln

Einsatz von Containern verspricht die immer größer werdende Komplexität der Anwendungslandschaft besser zu beherrschen. Dieser Spicker erklärt, wie Sie und Ihr Team Applikationen in Containern bauen und was Sie beachten müssen, um diese zu betreiben.

**IN DIESER AUSGABE**

- Kleine Stilkunde für Anwendungen in Containern
- Images bauen und Container betreiben
- Kubernetes (k8s) und Co.
- Migrationshilfe für bestehende Anwendungen

**Worum geht's?**

- Sie wollen Anwendungen in Containern betreiben. Welche Vorteile bringen diese bei welchen Architekturstil mit?
- Alte Anwendungen verharren noch in monolithischen Deployments oder schwergewichtigen Applikationsservern. Wie kriegen Sie diese schmerzarm in die „schöne neue Welt“?
- Um den richtigen Containerwechsel wird schon lange viel diskutiert. Wie setzen Sie Container heute richtig ein?
- Der Markt stellt soviel Orchestrierungslösungen bereit. Welche passt auf Ihre Situation am besten?

**Anwendungsentwurf, Container und Orchestrierung**

**Die Zusammenhänge, Fragen und Antworten**

Anwendungen in Containern sind ohne Hilfsmittel schwierig zu betreiben. Wie hängen die zentralen Begriffe zusammen? Was motiviert den Einsatz von Docker, Kubernetes und Co.?

**1. Worum geht es bei Anwendungen?**

Ohne Zerlegung ließe sich z. B. nicht mit mehreren Teams gleichzeitig daran arbeiten.

**2. Was haben Microservices damit zu tun?**

Die Zerlegung einer komplexen Anwendung macht sie barrierefrei und bewirtschaftbar.

**3. Warum sollten wir Anwendungen in verteilten Prozessen laufen lassen?**

Microservices sind ein Architekturstil, also eine grundlegende Art Anwendungen zu bauen.

**4. Und was sind Container?**

Container sind vergleichbar mit VMs, aber kleiner.

**5. Wie helfen Container bei verteilten Anwendungen?**

Container beinhalten jeweils nur eine Anwendung und alles, was es dazu braucht.

**6. Wie kommen Orchestrierungsstrategien ins Spiel?**

Mit Orchestrierungs-Lösungen lassen sich Anwendungen einfacher definieren, strukturiert und konfigurieren.

**Zusatzinformationen**

- Zu Architekturstilen siehe Seite 2
- Zu Microservices siehe Spicker #3
- Mehr zu Container ab Seite 3
- Mehr zu Kubernetes ab Seite 4

<https://architektur-spicker.de>

**architektur SPICKER**

Übersichten für die konzeptionelle Seite der Softwareentwicklung

**embarc Software Consulting GmbH**

**SIGS DATACOM**

# Continuous Delivery

Zeitmäßige Techniken aus Continuous Integration (CI) und Continuous Delivery (CD) unterstützen wichtige Architekturziele wie Stabilität und Reaktionsfähigkeit. Dieser Spicker zeigt den Aufbau einer passenden CI/CD-Kette.

**IN DIESER AUSGABE**

- Welchen Nutzen stiften die Elemente einer CI/CD-Kette?
- Welche Prinzipien und Praktiken haben sich bewährt?
- Wie profitiert Architekturarbeit?
- Wie startet man den Aufbau einer CI-CD-Kette?

**Worum geht's? (Herausforderungen/Ziele)**

- Neue Features in einer Lösung zu integrieren ist aufwändig und fehleranfällig. Wie minimiert ihr dieses Risiko?
- Moderne Architekturentsätze wie Microservices haben hohe Anforderungen bzgl. Integration und Verteilung. Welche Wechselwirkungen bestehen zwischen Architekturstil und CI/CD?
- Manuelle, wiederkehrende Tätigkeiten binden Kräfte und lassen sich nicht in gleichbleibender Qualität wiederholen. Wie eliminiert das Team diese monotonen Aufgaben?
- Auswirkungen von Änderungen in Quelltext, Technologie und Konfiguration werden erst spät im Entwicklungsprozess erkannt. Wie erhältet ihr rasch Feedback?

**Die Continuous Delivery Perlenkette**

CD automatisiert die Integrations- und Verteilungsprozesse von der Codierung bis zur lauffähigen Software, um schnell und verlässlich zu liefern. Die folgende Abbildung fädelt die wesentlichen Elemente zu einer CD-Kette zusammen:

```

graph TD
    1[1. Version Control System] --> 2[2. Continuous Build]
    2 --> 3[3. Continuous Testing]
    3 --> 4[4. Continuous Integration]
    4 --> 5[5. Configuration Management]
    5 --> 6[6. Automated Deployment]
    6 --> 7[7. Continuous Monitoring]
    7 --> 8[8. Continuous Delivery]
    8 --> 9[9. Continuous Deployment]
    9 --> 10[10. Continuous Delivery Pipeline]
    10 --> 11[11. Continuous Delivery Pipeline]
    11 --> 12[12. Continuous Delivery Pipeline]
    12 --> 13[13. Continuous Delivery Pipeline]
    13 --> 14[14. Continuous Delivery Pipeline]
    14 --> 15[15. Continuous Delivery Pipeline]
    15 --> 16[16. Continuous Delivery Pipeline]
    16 --> 17[17. Continuous Delivery Pipeline]
    17 --> 18[18. Continuous Delivery Pipeline]
    18 --> 19[19. Continuous Delivery Pipeline]
    19 --> 20[20. Continuous Delivery Pipeline]
    20 --> 21[21. Continuous Delivery Pipeline]
    21 --> 22[22. Continuous Delivery Pipeline]
    22 --> 23[23. Continuous Delivery Pipeline]
    23 --> 24[24. Continuous Delivery Pipeline]
    24 --> 25[25. Continuous Delivery Pipeline]
    25 --> 26[26. Continuous Delivery Pipeline]
    26 --> 27[27. Continuous Delivery Pipeline]
    27 --> 28[28. Continuous Delivery Pipeline]
    28 --> 29[29. Continuous Delivery Pipeline]
    29 --> 30[30. Continuous Delivery Pipeline]
    30 --> 31[31. Continuous Delivery Pipeline]
    31 --> 32[32. Continuous Delivery Pipeline]
    32 --> 33[33. Continuous Delivery Pipeline]
    33 --> 34[34. Continuous Delivery Pipeline]
    34 --> 35[35. Continuous Delivery Pipeline]
    35 --> 36[36. Continuous Delivery Pipeline]
    36 --> 37[37. Continuous Delivery Pipeline]
    37 --> 38[38. Continuous Delivery Pipeline]
    38 --> 39[39. Continuous Delivery Pipeline]
    39 --> 40[40. Continuous Delivery Pipeline]
    40 --> 41[41. Continuous Delivery Pipeline]
    41 --> 42[42. Continuous Delivery Pipeline]
    42 --> 43[43. Continuous Delivery Pipeline]
    43 --> 44[44. Continuous Delivery Pipeline]
    44 --> 45[45. Continuous Delivery Pipeline]
    45 --> 46[46. Continuous Delivery Pipeline]
    46 --> 47[47. Continuous Delivery Pipeline]
    47 --> 48[48. Continuous Delivery Pipeline]
    48 --> 49[49. Continuous Delivery Pipeline]
    49 --> 50[50. Continuous Delivery Pipeline]
    50 --> 51[51. Continuous Delivery Pipeline]
    51 --> 52[52. Continuous Delivery Pipeline]
    52 --> 53[53. Continuous Delivery Pipeline]
    53 --> 54[54. Continuous Delivery Pipeline]
    54 --> 55[55. Continuous Delivery Pipeline]
    55 --> 56[56. Continuous Delivery Pipeline]
    56 --> 57[57. Continuous Delivery Pipeline]
    57 --> 58[58. Continuous Delivery Pipeline]
    58 --> 59[59. Continuous Delivery Pipeline]
    59 --> 60[60. Continuous Delivery Pipeline]
    60 --> 61[61. Continuous Delivery Pipeline]
    61 --> 62[62. Continuous Delivery Pipeline]
    62 --> 63[63. Continuous Delivery Pipeline]
    63 --> 64[64. Continuous Delivery Pipeline]
    64 --> 65[65. Continuous Delivery Pipeline]
    65 --> 66[66. Continuous Delivery Pipeline]
    66 --> 67[67. Continuous Delivery Pipeline]
    67 --> 68[68. Continuous Delivery Pipeline]
    68 --> 69[69. Continuous Delivery Pipeline]
    69 --> 70[70. Continuous Delivery Pipeline]
    70 --> 71[71. Continuous Delivery Pipeline]
    71 --> 72[72. Continuous Delivery Pipeline]
    72 --> 73[73. Continuous Delivery Pipeline]
    73 --> 74[74. Continuous Delivery Pipeline]
    74 --> 75[75. Continuous Delivery Pipeline]
    75 --> 76[76. Continuous Delivery Pipeline]
    76 --> 77[77. Continuous Delivery Pipeline]
    77 --> 78[78. Continuous Delivery Pipeline]
    78 --> 79[79. Continuous Delivery Pipeline]
    79 --> 80[80. Continuous Delivery Pipeline]
    80 --> 81[81. Continuous Delivery Pipeline]
    81 --> 82[82. Continuous Delivery Pipeline]
    82 --> 83[83. Continuous Delivery Pipeline]
    83 --> 84[84. Continuous Delivery Pipeline]
    84 --> 85[85. Continuous Delivery Pipeline]
    85 --> 86[86. Continuous Delivery Pipeline]
    86 --> 87[87. Continuous Delivery Pipeline]
    87 --> 88[88. Continuous Delivery Pipeline]
    88 --> 89[89. Continuous Delivery Pipeline]
    89 --> 90[90. Continuous Delivery Pipeline]
    90 --> 91[91. Continuous Delivery Pipeline]
    91 --> 92[92. Continuous Delivery Pipeline]
    92 --> 93[93. Continuous Delivery Pipeline]
    93 --> 94[94. Continuous Delivery Pipeline]
    94 --> 95[95. Continuous Delivery Pipeline]
    95 --> 96[96. Continuous Delivery Pipeline]
    96 --> 97[97. Continuous Delivery Pipeline]
    97 --> 98[98. Continuous Delivery Pipeline]
    98 --> 99[99. Continuous Delivery Pipeline]
    99 --> 100[100. Continuous Delivery Pipeline]
    100 --> 101[101. Continuous Delivery Pipeline]
    101 --> 102[102. Continuous Delivery Pipeline]
    102 --> 103[103. Continuous Delivery Pipeline]
    103 --> 104[104. Continuous Delivery Pipeline]
    104 --> 105[105. Continuous Delivery Pipeline]
    105 --> 106[106. Continuous Delivery Pipeline]
    106 --> 107[107. Continuous Delivery Pipeline]
    107 --> 108[108. Continuous Delivery Pipeline]
    108 --> 109[109. Continuous Delivery Pipeline]
    109 --> 110[110. Continuous Delivery Pipeline]
    110 --> 111[111. Continuous Delivery Pipeline]
    111 --> 112[112. Continuous Delivery Pipeline]
    112 --> 113[113. Continuous Delivery Pipeline]
    113 --> 114[114. Continuous Delivery Pipeline]
    114 --> 115[115. Continuous Delivery Pipeline]
    115 --> 116[116. Continuous Delivery Pipeline]
    116 --> 117[117. Continuous Delivery Pipeline]
    117 --> 118[118. Continuous Delivery Pipeline]
    118 --> 119[119. Continuous Delivery Pipeline]
    119 --> 120[120. Continuous Delivery Pipeline]
    120 --> 121[121. Continuous Delivery Pipeline]
    121 --> 122[122. Continuous Delivery Pipeline]
    122 --> 123[123. Continuous Delivery Pipeline]
    123 --> 124[124. Continuous Delivery Pipeline]
    124 --> 125[125. Continuous Delivery Pipeline]
    125 --> 126[126. Continuous Delivery Pipeline]
    126 --> 127[127. Continuous Delivery Pipeline]
    127 --> 128[128. Continuous Delivery Pipeline]
    128 --> 129[129. Continuous Delivery Pipeline]
    129 --> 130[130. Continuous Delivery Pipeline]
    130 --> 131[131. Continuous Delivery Pipeline]
    131 --> 132[132. Continuous Delivery Pipeline]
    132 --> 133[133. Continuous Delivery Pipeline]
    133 --> 134[134. Continuous Delivery Pipeline]
    134 --> 135[135. Continuous Delivery Pipeline]
    135 --> 136[136. Continuous Delivery Pipeline]
    136 --> 137[137. Continuous Delivery Pipeline]
    137 --> 138[138. Continuous Delivery Pipeline]
    138 --> 139[139. Continuous Delivery Pipeline]
    139 --> 140[140. Continuous Delivery Pipeline]
    140 --> 141[141. Continuous Delivery Pipeline]
    141 --> 142[142. Continuous Delivery Pipeline]
    142 --> 143[143. Continuous Delivery Pipeline]
    143 --> 144[144. Continuous Delivery Pipeline]
    144 --> 145[145. Continuous Delivery Pipeline]
    145 --> 146[146. Continuous Delivery Pipeline]
    146 --> 147[147. Continuous Delivery Pipeline]
    147 --> 148[148. Continuous Delivery Pipeline]
    148 --> 149[149. Continuous Delivery Pipeline]
    149 --> 150[150. Continuous Delivery Pipeline]
    150 --> 151[151. Continuous Delivery Pipeline]
    151 --> 152[152. Continuous Delivery Pipeline]
    152 --> 153[153. Continuous Delivery Pipeline]
    153 --> 154[154. Continuous Delivery Pipeline]
    154 --> 155[155. Continuous Delivery Pipeline]
    155 --> 156[156. Continuous Delivery Pipeline]
    156 --> 157[157. Continuous Delivery Pipeline]
    157 --> 158[158. Continuous Delivery Pipeline]
    158 --> 159[159. Continuous Delivery Pipeline]
    159 --> 160[160. Continuous Delivery Pipeline]
    160 --> 161[161. Continuous Delivery Pipeline]
    161 --> 162[162. Continuous Delivery Pipeline]
    162 --> 163[163. Continuous Delivery Pipeline]
    163 --> 164[164. Continuous Delivery Pipeline]
    164 --> 165[165. Continuous Delivery Pipeline]
    165 --> 166[166. Continuous Delivery Pipeline]
    166 --> 167[167. Continuous Delivery Pipeline]
    167 --> 168[168. Continuous Delivery Pipeline]
    168 --> 169[169. Continuous Delivery Pipeline]
    169 --> 170[170. Continuous Delivery Pipeline]
    170 --> 171[171. Continuous Delivery Pipeline]
    171 --> 172[172. Continuous Delivery Pipeline]
    172 --> 173[173. Continuous Delivery Pipeline]
    173 --> 174[174. Continuous Delivery Pipeline]
    174 --> 175[175. Continuous Delivery Pipeline]
    175 --> 176[176. Continuous Delivery Pipeline]
    176 --> 177[177. Continuous Delivery Pipeline]
    177 --> 178[178. Continuous Delivery Pipeline]
    178 --> 179[179. Continuous Delivery Pipeline]
    179 --> 180[180. Continuous Delivery Pipeline]
    180 --> 181[181. Continuous Delivery Pipeline]
    181 --> 182[182. Continuous Delivery Pipeline]
    182 --> 183[183. Continuous Delivery Pipeline]
    183 --> 184[184. Continuous Delivery Pipeline]
    184 --> 185[185. Continuous Delivery Pipeline]
    185 --> 186[186. Continuous Delivery Pipeline]
    186 --> 187[187. Continuous Delivery Pipeline]
    187 --> 188[188. Continuous Delivery Pipeline]
    188 --> 189[189. Continuous Delivery Pipeline]
    189 --> 190[190. Continuous Delivery Pipeline]
    190 --> 191[191. Continuous Delivery Pipeline]
    191 --> 192[192. Continuous Delivery Pipeline]
    192 --> 193[193. Continuous Delivery Pipeline]
    193 --> 194[194. Continuous Delivery Pipeline]
    194 --> 195[195. Continuous Delivery Pipeline]
    195 --> 196[196. Continuous Delivery Pipeline]
    196 --> 197[197. Continuous Delivery Pipeline]
    197 --> 198[198. Continuous Delivery Pipeline]
    198 --> 199[199. Continuous Delivery Pipeline]
    199 --> 200[200. Continuous Delivery Pipeline]
    200 --> 201[201. Continuous Delivery Pipeline]
    201 --> 202[202. Continuous Delivery Pipeline]
    202 --> 203[203. Continuous Delivery Pipeline]
    203 --> 204[204. Continuous Delivery Pipeline]
    204 --> 205[205. Continuous Delivery Pipeline]
    205 --> 206[206. Continuous Delivery Pipeline]
    206 --> 207[207. Continuous Delivery Pipeline]
    207 --> 208[208. Continuous Delivery Pipeline]
    208 --> 209[209. Continuous Delivery Pipeline]
    209 --> 210[210. Continuous Delivery Pipeline]
    210 --> 211[211. Continuous Delivery Pipeline]
    211 --> 212[212. Continuous Delivery Pipeline]
    212 --> 213[213. Continuous Delivery Pipeline]
    213 --> 214[214. Continuous Delivery Pipeline]
    214 --> 215[215. Continuous Delivery Pipeline]
    215 --> 216[216. Continuous Delivery Pipeline]
    216 --> 217[217. Continuous Delivery Pipeline]
    217 --> 218[218. Continuous Delivery Pipeline]
    218 --> 219[219. Continuous Delivery Pipeline]
    219 --> 220[220. Continuous Delivery Pipeline]
    220 --> 221[221. Continuous Delivery Pipeline]
    221 --> 222[222. Continuous Delivery Pipeline]
    222 --> 223[223. Continuous Delivery Pipeline]
    223 --> 224[224. Continuous Delivery Pipeline]
    224 --> 225[225. Continuous Delivery Pipeline]
    225 --> 226[226. Continuous Delivery Pipeline]
    226 --> 227[227. Continuous Delivery Pipeline]
    227 --> 228[228. Continuous Delivery Pipeline]
    228 --> 229[229. Continuous Delivery Pipeline]
    229 --> 230[230. Continuous Delivery Pipeline]
    230 --> 231[231. Continuous Delivery Pipeline]
    231 --> 232[232. Continuous Delivery Pipeline]
    232 --> 233[233. Continuous Delivery Pipeline]
    233 --> 234[234. Continuous Delivery Pipeline]
    234 --> 235[235. Continuous Delivery Pipeline]
    235 --> 236[236. Continuous Delivery Pipeline]
    236 --> 237[237. Continuous Delivery Pipeline]
    237 --> 238[238. Continuous Delivery Pipeline]
    238 --> 239[239. Continuous Delivery Pipeline]
    239 --> 240[240. Continuous Delivery Pipeline]
    240 --> 241[241. Continuous Delivery Pipeline]
    241 --> 242[242. Continuous Delivery Pipeline]
    242 --> 243[243. Continuous Delivery Pipeline]
    243 --> 244[244. Continuous Delivery Pipeline]
    244 --> 245[245. Continuous Delivery Pipeline]
    245 --> 246[246. Continuous Delivery Pipeline]
    246 --> 247[247. Continuous Delivery Pipeline]
    247 --> 248[248. Continuous Delivery Pipeline]
    248 --> 249[249. Continuous Delivery Pipeline]
    249 --> 250[250. Continuous Delivery Pipeline]
    250 --> 251[251. Continuous Delivery Pipeline]
    251 --> 252[252. Continuous Delivery Pipeline]
    252 --> 253[253. Continuous Delivery Pipeline]
    253 --> 254[254. Continuous Delivery Pipeline]
    254 --> 255[255. Continuous Delivery Pipeline]
    255 --> 256[256. Continuous Delivery Pipeline]
    256 --> 257[257. Continuous Delivery Pipeline]
    257 --> 258[258. Continuous Delivery Pipeline]
    258 --> 259[259. Continuous Delivery Pipeline]
    259 --> 260[260. Continuous Delivery Pipeline]
    260 --> 261[261. Continuous Delivery Pipeline]
    261 --> 262[262. Continuous Delivery Pipeline]
    262 --> 263[263. Continuous Delivery Pipeline]
    263 --> 264[264. Continuous Delivery Pipeline]
    264 --> 265[265. Continuous Delivery Pipeline]
    265 --> 266[266. Continuous Delivery Pipeline]
    266 --> 267[267. Continuous Delivery Pipeline]
    267 --> 268[268. Continuous Delivery Pipeline]
    268 --> 269[269. Continuous Delivery Pipeline]
    269 --> 270[270. Continuous Delivery Pipeline]
    270 --> 271[271. Continuous Delivery Pipeline]
    271 --> 272[272. Continuous Delivery Pipeline]
    272 --> 273[273. Continuous Delivery Pipeline]
    273 --> 274[274. Continuous Delivery Pipeline]
    274 --> 275[275. Continuous Delivery Pipeline]
    275 --> 276[276. Continuous Delivery Pipeline]
    276 --> 277[277. Continuous Delivery Pipeline]
    277 --> 278[278. Continuous Delivery Pipeline]
    278 --> 279[279. Continuous Delivery Pipeline]
    279 --> 280[280. Continuous Delivery Pipeline]
    280 --> 281[281. Continuous Delivery Pipeline]
    281 --> 282[282. Continuous Delivery Pipeline]
    282 --> 283[283. Continuous Delivery Pipeline]
    283 --> 284[284. Continuous Delivery Pipeline]
    284 --> 285[285. Continuous Delivery Pipeline]
    285 --> 286[286. Continuous Delivery Pipeline]
    286 --> 287[287. Continuous Delivery Pipeline]
    287 --> 288[288. Continuous Delivery Pipeline]
    288 --> 289[289. Continuous Delivery Pipeline]
    289 --> 290[290. Continuous Delivery Pipeline]
    290 --> 291[291. Continuous Delivery Pipeline]
    291 --> 292[292. Continuous Delivery Pipeline]
    292 --> 293[293. Continuous Delivery Pipeline]
    293 --> 294[294. Continuous Delivery Pipeline]
    294 --> 295[295. Continuous Delivery Pipeline]
    295 --> 296[296. Continuous Delivery Pipeline]
    296 --> 297[297. Continuous Delivery Pipeline]
    297 --> 298[298. Continuous Delivery Pipeline]
    298 --> 299[299. Continuous Delivery Pipeline]
    299 --> 300[300. Continuous Delivery Pipeline]
    300 --> 301[301. Continuous Delivery Pipeline]
    301 --> 302[302. Continuous Delivery Pipeline]
    302 --> 303[303. Continuous Delivery Pipeline]
    303 --> 304[304. Continuous Delivery Pipeline]
    304 --> 305[305. Continuous Delivery Pipeline]
    305 --> 306[306. Continuous Delivery Pipeline]
    306 --> 307[307. Continuous Delivery Pipeline]
    307 --> 308[308. Continuous Delivery Pipeline]
    308 --> 309[309. Continuous Delivery Pipeline]
    309 --> 310[310. Continuous Delivery Pipeline]
    310 --> 311[311. Continuous Delivery Pipeline]
    311 --> 312[312. Continuous Delivery Pipeline]
    312 --> 313[313. Continuous Delivery Pipeline]
    313 --> 314[314. Continuous Delivery Pipeline]
    314 --> 315[315. Continuous Delivery Pipeline]
    315 --> 316[316. Continuous Delivery Pipeline]
    316 --> 317[317. Continuous Delivery Pipeline]
    317 --> 318[318. Continuous Delivery Pipeline]
    318 --> 319[319. Continuous Delivery Pipeline]
    319 --> 320[320. Continuous Delivery Pipeline]
    320 --> 321[321. Continuous Delivery Pipeline]
    321 --> 322[322. Continuous Delivery Pipeline]
    322 --> 323[323. Continuous Delivery Pipeline]
    323 --> 324[324. Continuous Delivery Pipeline]
    324 --> 325[325. Continuous Delivery Pipeline]
    325 --> 326[326. Continuous Delivery Pipeline]
    326 --> 327[327. Continuous Delivery Pipeline]
    327 --> 328[328. Continuous Delivery Pipeline]
    328 --> 329[329. Continuous Delivery Pipeline]
    329 --> 330[330. Continuous Delivery Pipeline]
    330 --> 331[331. Continuous Delivery Pipeline]
    331 --> 332[332. Continuous Delivery Pipeline]
    332 --> 333[333. Continuous Delivery Pipeline]
    333 --> 334[334. Continuous Delivery Pipeline]
    334 --> 335[335. Continuous Delivery Pipeline]
    335 --> 336[336. Continuous Delivery Pipeline]
    336 --> 337[337. Continuous Delivery Pipeline]
    337 --> 338[338. Continuous Delivery Pipeline]
    338 --> 339[339. Continuous Delivery Pipeline]
    339 --> 340[340. Continuous Delivery Pipeline]
    340 --> 341[341. Continuous Delivery Pipeline]
    341 --> 342[342. Continuous Delivery Pipeline]
    342 --> 343[343. Continuous Delivery Pipeline]
    343 --> 344[344. Continuous Delivery Pipeline]
    344 --> 345[345. Continuous Delivery Pipeline]
    345 --> 346[346. Continuous Delivery Pipeline]
    346 --> 347[347. Continuous Delivery Pipeline]
    347 --> 348[348. Continuous Delivery Pipeline]
    348 --> 349[349. Continuous Delivery Pipeline]
    349 --> 350[350. Continuous Delivery Pipeline]
    350 --> 351[351. Continuous Delivery Pipeline]
    351 --> 352[352. Continuous Delivery Pipeline]
    352 --> 353[353. Continuous Delivery Pipeline]
    353 --> 354[354. Continuous Delivery Pipeline]
    354 --> 355[355. Continuous Delivery Pipeline]
    355 --> 356[356. Continuous Delivery Pipeline]
    356 --> 357[357. Continuous Delivery Pipeline]
    357 --> 358[358. Continuous Delivery Pipeline]
    358 --> 359[359. Continuous Delivery Pipeline]
    359 --> 360[360. Continuous Delivery Pipeline]
    360 --> 361[361. Continuous Delivery Pipeline]
    361 --> 362[362. Continuous Delivery Pipeline]
    362 --> 363[363. Continuous Delivery Pipeline]
    363 --> 364[364. Continuous Delivery Pipeline]
    364 --> 365[365. Continuous Delivery Pipeline]
    365 --> 366[366. Continuous Delivery Pipeline]
    366 --> 367[367. Continuous Delivery Pipeline]
    367 --> 368[368. Continuous Delivery Pipeline]
    368 --> 369[369. Continuous Delivery Pipeline]
    369 --> 370[370. Continuous Delivery Pipeline]
    370 --> 371[371. Continuous Delivery Pipeline]
    371 --> 372[372. Continuous Delivery Pipeline]
    372 --> 373[373. Continuous Delivery Pipeline]
    373 --> 374[374. Continuous Delivery Pipeline]
    374 --> 375[375. Continuous Delivery Pipeline]
    375 --> 376[376. Continuous Delivery Pipeline]
    376 --> 377[377. Continuous Delivery Pipeline]
    377 --> 378[378. Continuous Delivery Pipeline]
    378 --> 379[379. Continuous Delivery Pipeline]
    379 --> 380[380. Continuous Delivery Pipeline]
    380 --> 381[381. Continuous Delivery Pipeline]
    381 --> 382[382. Continuous Delivery Pipeline]
    382 --> 383[383. Continuous Delivery Pipeline]
    383 --> 384[384. Continuous Delivery Pipeline]
    384 --> 385[385. Continuous Delivery Pipeline]
    385 --> 386[386. Continuous Delivery Pipeline]
    386 --> 387[387. Continuous Delivery Pipeline]
    387 --> 388[388. Continuous Delivery Pipeline]
    388 --> 389[389. Continuous Delivery Pipeline]
    389 --> 390[390. Continuous Delivery Pipeline]
    390 --> 391[391. Continuous Delivery Pipeline]
    391 --> 392[392. Continuous Delivery Pipeline]
    392 --> 393[393. Continuous Delivery Pipeline]
    393 --> 394[394. Continuous Delivery Pipeline]
    394 --> 395[395. Continuous Delivery Pipeline]
    395 --> 396[396. Continuous Delivery Pipeline]
    396 --> 397[397. Continuous Delivery Pipeline]
    397 --> 398[398. Continuous Delivery Pipeline]
    398 --> 399[399. Continuous Delivery Pipeline]
    399 --> 400[400. Continuous Delivery Pipeline]
    400 --> 401[401. Continuous Delivery Pipeline]
    401 --> 402[402. Continuous Delivery Pipeline]
    402 --> 403[403. Continuous Delivery Pipeline]
    403 --> 404[404. Continuous Delivery Pipeline]
    404 --> 405[405. Continuous Delivery Pipeline]
    405 --> 406[406. Continuous Delivery Pipeline]
    406 --> 407[407. Continuous Delivery Pipeline]
    407 --> 408[408. Continuous Delivery Pipeline]
    408 --> 409[409. Continuous Delivery Pipeline]
    409 --> 410[410. Continuous Delivery Pipeline]
    410 --> 411[411. Continuous Delivery Pipeline]
    411 --> 412[412. Continuous Delivery Pipeline]
    412 --> 413[413. Continuous Delivery Pipeline]
    413 --> 414[414. Continuous Delivery Pipeline]
    414 --> 415[415. Continuous Delivery Pipeline]
    415 --> 416[416. Continuous Delivery Pipeline]
    416 --> 417[417. Continuous Delivery Pipeline]
    417 --> 418[418. Continuous Delivery Pipeline]
    418 --> 419[419. Continuous Delivery Pipeline]
    419 --> 420[420. Continuous Delivery Pipeline]
    420 --> 421[421. Continuous Delivery Pipeline]
    421 --> 422[422. Continuous Delivery Pipeline]
    422 --> 423[423. Continuous Delivery Pipeline]
    423 --> 424[424. Continuous Delivery Pipeline]
    424 --> 425[425. Continuous Delivery Pipeline]
    425 --> 426[426. Continuous Delivery Pipeline]
    426 --> 427[427. Continuous Delivery Pipeline]
    427 --> 428[428. Continuous Delivery Pipeline]
    428 --> 429[429. Continuous Delivery Pipeline]
    429 --> 430[430. Continuous Delivery Pipeline]
    430 --> 431[431. Continuous Delivery Pipeline]
    431 --> 432[432. Continuous Delivery Pipeline]
    432 --> 433[433. Continuous Delivery Pipeline]
    433 --> 434[434. Continuous Delivery Pipeline]
    434 --> 435[435. Continuous Delivery Pipeline]
    435 --> 436[436. Continuous Delivery Pipeline]
    436 --> 437[437. Continuous Delivery Pipeline]
    437 --> 438[438. Continuous Delivery Pipeline]
    438 --> 439[439. Continuous Delivery Pipeline]
    439 --> 440[440. Continuous Delivery Pipeline]
    440 --> 441[441. Continuous Delivery Pipeline]
    441 --> 442[442. Continuous Delivery Pipeline]
    442 --> 443[443. Continuous Delivery Pipeline]
    443 --> 444[444. Continuous Delivery Pipeline]
    444 --> 445[445. Continuous Delivery Pipeline]
    445 --> 446[446. Continuous Delivery Pipeline]
    446 --> 447[447. Continuous Delivery Pipeline]
    447 --> 448[448. Continuous Delivery Pipeline]
    448 --> 449[449. Continuous Delivery Pipeline]
    449 --> 450[450. Continuous Delivery Pipeline]
    450 --> 451[451. Continuous Delivery Pipeline]
    451 --> 452[452. Continuous Delivery Pipeline]
    452 --> 453[453. Continuous Delivery Pipeline]
    453 --> 454[454. Continuous Delivery Pipeline]
    454 --> 455[455. Continuous Delivery Pipeline]
    455 --> 456[456. Continuous Delivery Pipeline]
    456 --> 457[457. Continuous Delivery Pipeline]
    457 --> 458[458. Continuous Delivery Pipeline]
    458 --> 459[459. Continuous Delivery Pipeline]
    459 --> 460[460. Continuous Delivery Pipeline]
    460 --> 461[461. Continuous Delivery Pipeline]
    461 --> 462[462. Continuous Delivery Pipeline]
    462 --> 463[463. Continuous Delivery Pipeline]
    463 --> 464[464. Continuous Delivery Pipeline]
    464 --> 465[465. Continuous Delivery Pipeline]
    465 --> 466[466. Continuous Delivery Pipeline]
    466 --> 467[467. Continuous Delivery Pipeline]
    467 --> 468[468. Continuous Delivery Pipeline]
    468 --> 469[469. Continuous Delivery Pipeline]
    469 --> 470[470. Continuous Delivery Pipeline]
    470 --> 471[471. Continuous Delivery Pipeline]
    471 --> 472[472. Continuous Delivery Pipeline]
    472 --> 473[473. Continuous Delivery Pipeline]
    473 --> 474[474. Continuous Delivery Pipeline]
    474 --> 475[475. Continuous Delivery Pipeline]
    475 --> 476[476. Continuous Delivery Pipeline]
    476 --> 477[477. Continuous Delivery Pipeline]
    477 --> 478[478. Continuous Delivery Pipeline]
    478 --> 479[479. Continuous Delivery Pipeline]
    479 --> 480[480. Continuous Delivery Pipeline]
    480 --> 481[481. Continuous Delivery Pipeline]
    481 --> 482[482. Continuous Delivery Pipeline]
    482 --> 483[483. Continuous Delivery Pipeline]
    483 --> 484[484. Continuous Delivery Pipeline]
    484 --> 485[485. Continuous Delivery Pipeline]
    485 --> 486[486. Continuous Delivery Pipeline]
    486 --> 487[487. Continuous Delivery Pipeline]
    487 --> 488[488. Continuous Delivery Pipeline]
    488 --> 489[489. Continuous Delivery Pipeline]
    489 --> 490[490. Continuous Delivery Pipeline]
    490 --> 491[491. Continuous Delivery Pipeline]
    491 --> 492[492. Continuous Delivery Pipeline]
    492 --> 493[493. Continuous Delivery Pipeline]
    493 --> 494[494. Continuous Delivery Pipeline]
    494 --> 495[495. Continuous Delivery Pipeline]
    495 --> 496[496. Continuous Delivery Pipeline]
    496 --> 497[497. Continuous Delivery Pipeline]
    497 --> 498[498. Continuous Delivery Pipeline]
    498 --> 499[499. Continuous Delivery Pipeline]
    499 --> 500[500. Continuous Delivery Pipeline]
    500 --> 501[501. Continuous Delivery Pipeline]
    501 --> 502[502. Continuous Delivery Pipeline]
    502 --> 503[503. Continuous Delivery Pipeline]
    503 --> 504[504. Continuous Delivery Pipeline]
    504 --> 505[505. Continuous Delivery Pipeline]
    505 --> 506[506. Continuous Delivery Pipeline]
    506 --> 507[507. Continuous Delivery Pipeline]
    507 --> 508[508. Continuous Delivery Pipeline]
    508 --> 509[509. Continuous Delivery Pipeline]
    509 --> 510[510. Continuous Delivery Pipeline]
    510 --> 511[511. Continuous Delivery Pipeline]
    511 --> 512
```

# Weitere gute Vorträge zum Thema

- Vortrag „Wenn ich das nur vorher gewusst hätte: Kubernetes für Entwickler“ von Stefan Schlott
- Vortrag „Kubernetes-Lektionen aus der Wolke“ von Jochen Mader
- Vortrag „What's going on in my cluster?“ von Matthias Häussler

# Weitere Informationen

- <https://www.informatik-aktuell.de/entwicklung/methoden/container-images-deep-dive-101-wege-zum-bauen-und-bereitstellen.html>
- „Kubernetes in Action“ von Marko Lukša
- „Docker in Action“ von Jeff Nickoloff, Stephen Kuenzli
- „Container-Anwendungen entwickeln“  
<https://www.architektur-spicker.de/>
- „Continuous Delivery“ <https://www.architektur-spicker.de/>

# Bildnachweise

- [https://unsplash.com/photos/RfwGg5ZZh4Q?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditShareLink](https://unsplash.com/photos/RfwGg5ZZh4Q?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink)
- [https://unsplash.com/photos/CpsTAUPoScw?utm\\_source=unsplash&utm\\_medium=referral&utm\\_content=creditShareLink](https://unsplash.com/photos/CpsTAUPoScw?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink)