# IT Tage, 11.12.2024

# Kubernetes, das unbekannte Wesen Schnelleinstieg für Entwicklerinnen

### Sandra Parsick

@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

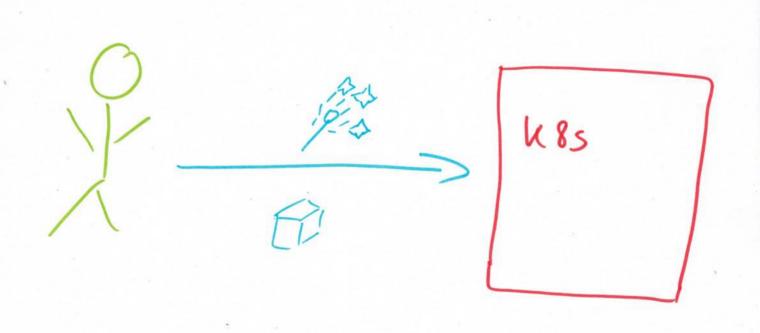
- mail@sandra-parsick.de
- @sparsick@mastodon.social
- https://www.sandra-parsick.de
- https://ready-for-review.dev







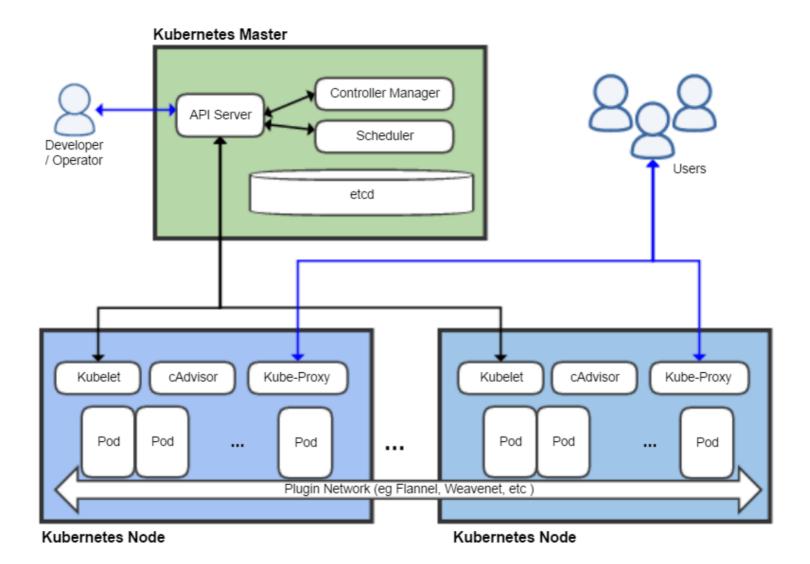


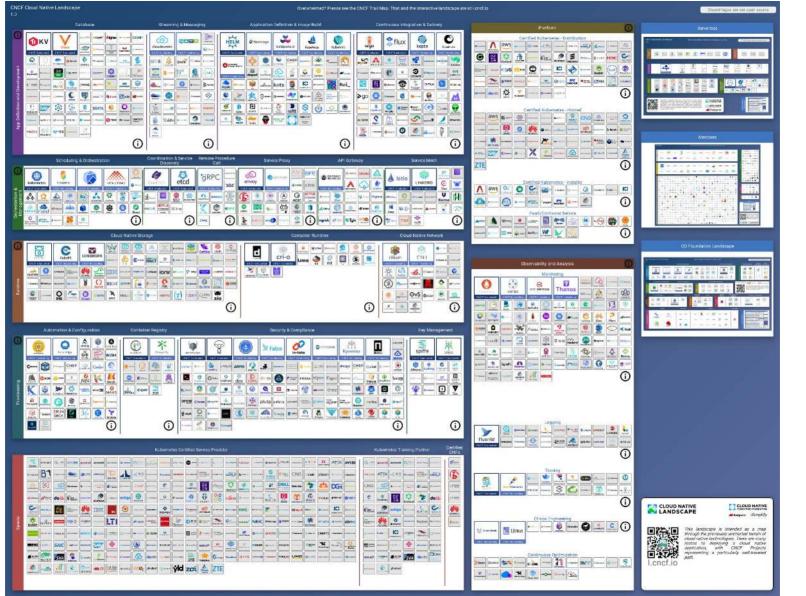


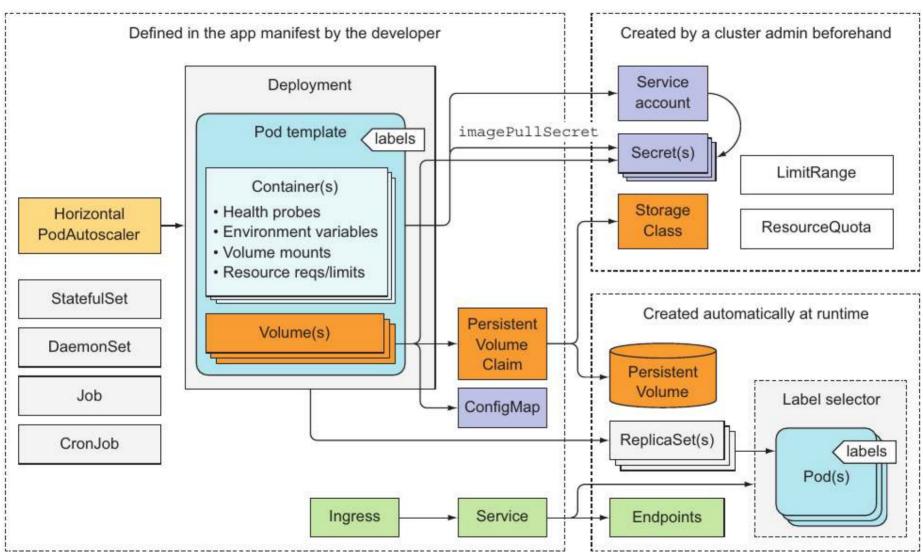
Was ist eigentlich dieses Kubernetes?

**Kubernetes** is an open-source container orchestration system for automating software deployment, scaling, and management

(Wikipedia)

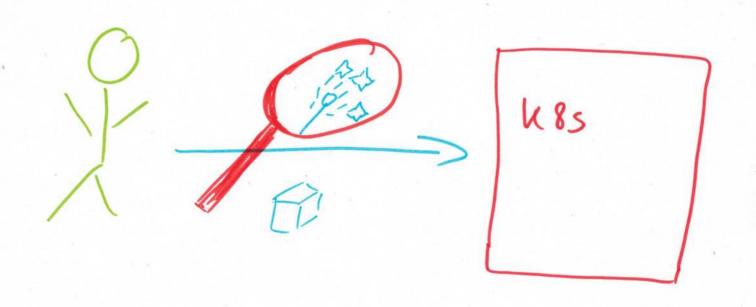






Aus: Kubernetes in Action, 1st Edition

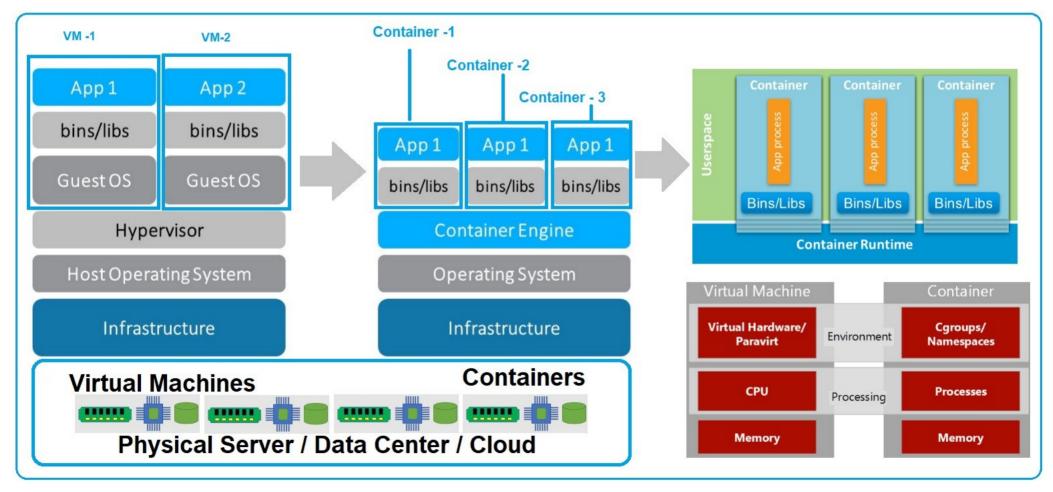
### Ein Begriffsbild für Kubernetes Das folgende Bild zeigt aus Sicht der Anwendungsentwicklung zentrale Begriffe von k8s im Zusammenspiel. Legende Allgemeines Konzept Clusterweite K8s-Obiekte **CONTAINER JOB DEPLOYMENT Organisiert in Namespaces** Kann Teil einer selbst-ge-Möglichkeit. Pods im Bietet deklarative Aktualisie-Batch-Stil auszuführen. rungen (z. B. Image-Version, oder auch Middleware. Ggf. auch parallel. Anzahl Repliken) CONFIGMAP mounted Entkopplung der umgebungsnutzt spezifischen Konfiguration beinhaltet kann ausführen VOLUME von Container-Images REPLICASET Speichermedium, auf das Stellt Ausführung einer Container zugreifen können ist spezielles bestimmter Anzahl von Podz. B. PersistentVolume stellt Anzahl Replikaten sicher. sicher ist ähnlich zu POD referenziert Die kleinste verteilbare Ein-**SECRET SERVICE** heit in K8s (Pod == engl. für Enthält sensible Daten wie ermöglicht Bietet einen Endpunkt für Kapsel). Zugriff auf ein Passwort, ein Token oder eine Menge von Pods. einen Schlüssel. gehört zu läuft auf macht von außen gehört zu zugreifbar **NAMESPACE** NODE struktu-Konzept um Ordnung im besteht **CLUSTER** Physischer oder virtueller riert Cluster zu halten, K8s-Obaus Rechner (VM), der die Arbeit Einstiegspunkt. Wenn Sie **INGRESS** jekte sind in der Regel einem macht (Worker). sich mit K8s "kubectl" verbin-Steuert Zugriffe von oder zugeordnet. den, tun Sie dies mit einem nach außen (Egress) Cluster.



### Container



# Exkurs: Container vs VMs



Repository myapp

Tag

LAYER 17612

LAYER 17612

Container Image myapp: 2.80

LAYER 675411

LAYER 675411

# **Basis: Container**

```
FROM docker.io/eclipse-temurin:17.0.1 12-jre as builder
WORKDIR /application
COPY maven/*.jar application.jar
RUN java -Djarmode=layertools -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"]
```

## Basis: Container



### Demo

# Container Builder

- Docker
- Buildpacks
- JIB
- Buildah
- Podman
- Weitere Infos im Artikel "Container-Images Deep Dive" auf Informatik Aktuell

**Exkurs: Lokales Kubernetes** 



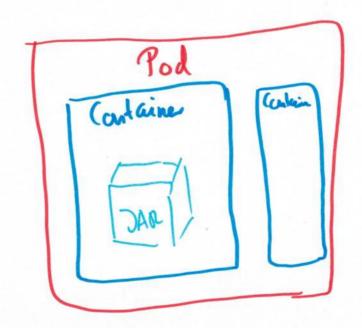
# minikube

### Demo

# Alternativen zu Minikube

- k3s
- k3d
- kind
- microk8s
- k0s

### Pod



apiVersion: v1 kind: Pod metadata: name: spring-boot-demo spec: containers: - name: hero-app image: sparsick/spring-boot-demo:1.5.0 ports: - containerPort: 8080



kubectl apply -f pod.yaml

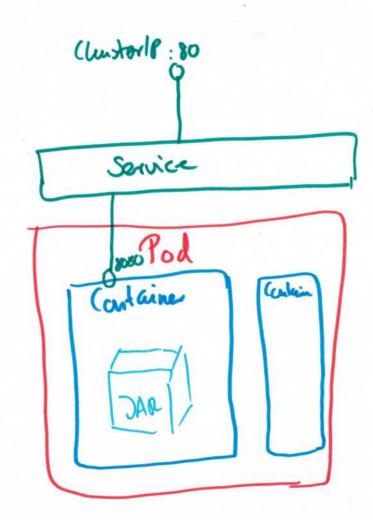
### Demo

```
. .
apiVersion: v1
kind: Pod
metadata:
  name: spring-boot-demo
spec:
  containers:
  - name: hero-app
    image: sparsick/spring-boot-demo:1.5.0
    ports:
    - name: container-http
      containerPort: 8080
      protocol: TCP
    livenessProbe:
      httpGet:
        path: /actuator/health/liveness
        port: container-http
      initialDelaySeconds: 15
      periodSeconds: 10
      timeoutSeconds: 30
    readinessProbe:
      httpGet:
        path: /actuator/health/readiness
        port: container-http
      initialDelaySeconds: 15
      periodSeconds: 10
      timeoutSeconds: 30
```

### Demo

Tontall Hachine Conhellers A31-Server Soduler Luebecty excd Node Machine Container Pleintine

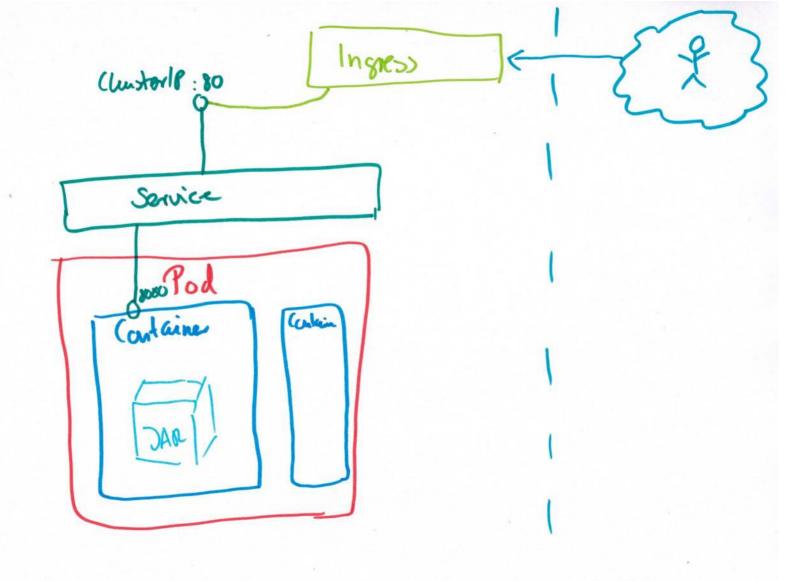
### Service



```
. . .
apiVersion: v1
kind: Pod
metadata:
 name: spring-boot-demo
 labels:
    app.kubernetes.io/name: hero
spec:
 containers:
  - name: hero-app
    image: sparsick/spring-boot-demo:1.5.0
   ports:
   - name: container-http
     containerPort: 8080
     protocol: TCP
apiVersion: v1
kind: Service
metadata:
 name: spring-boot-demo-srv
spec:
 type: ClusterIP
 selector:
   app.kubernetes.io/name: hero
 ports:
  - name: hero-http
   protocol: TCP
   port: 80
    targetPort: 8080
```

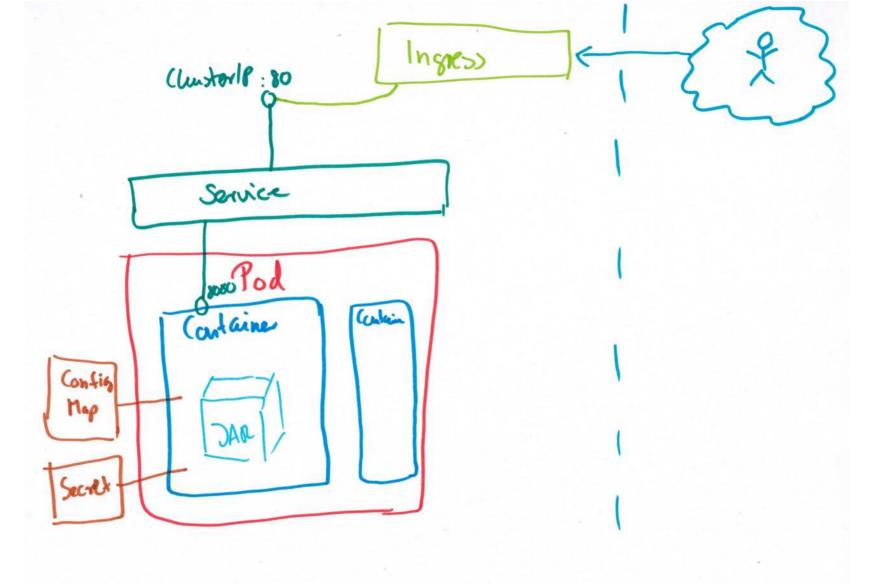
### Demo

# Ingress



```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: spring-demo-ingress
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    nginx.ingress.kubernetes.io/x-forwarded-prefix: "/"
spec:
  rules:
    host: spring-boot-demo.local
      http:
        paths:
          - path: /
            pathType: Prefix
           backend:
              service:
                name: spring-boot-demo-srv
                port:
                 number: 80
```

ConfigMap / Secret



```
apiVersion: v1
kind: ConfigMap
metadata:
  name: spring-boot-demo-config
data:
  MONGODB_ENABLED: "false"
apiVersion: v1
kind: Secret
metadata:
  name: secret-sample
stringData:
  passwort: geheim
```

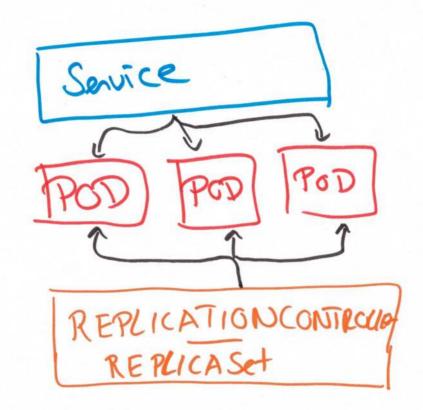
```
apiVersion: v1
kind: Pod
metadata:
 name: spring-boot-demo
spec:
  containers:
  - name: hero-app
    image: sparsick/spring-boot-demo:1.5.0
    ports:
    - name: container-http
      containerPort: 8080
      protocol: TCP
    envFrom:
    - configMapRef:
        name: spring-boot-demo-config
```

# Namespace

I CLUSTER Namespace A Namespoce B Sevice

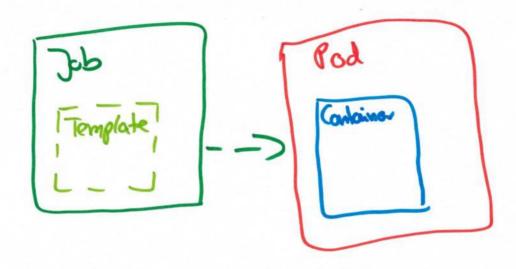
```
apiVersion: v1
kind: Namespace
metadata:
 name: spring-boot-demo-namespace
apiVersion: v1
kind: Pod
metadata:
 name: spring-boot-demo
  namespace: spring-boot-demo-namespace
spec:
  containers:
  - name: hero-app
    image: sparsick/spring-boot-demo:1.5.0
    ports:
    - containerPort: 8080
```

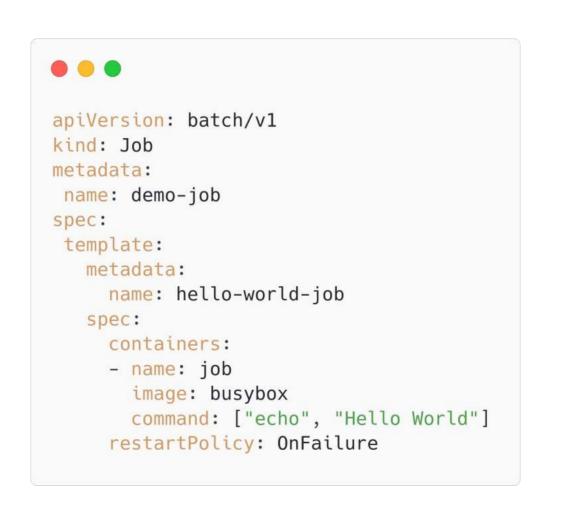
# Deployment

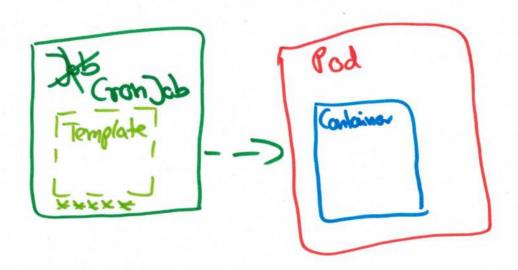


```
. .
apiVersion: apps/v1
kind: Deployment
metadata:
 name: spring-boot-demo-deploy
 namespace: spring-boot-demo-namespace
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hero-app
  template:
    metadata:
      labels:
        app: hero-app
    spec:
      containers:
      - name: hero-app
        image: sparsick/spring-boot-demo:1.5.0
        ports:
        - name: container-http
          containerPort: 8080
          protocol: TCP
        envFrom:
        - configMapRef:
            name: spring-boot-demo-config
```

## Job

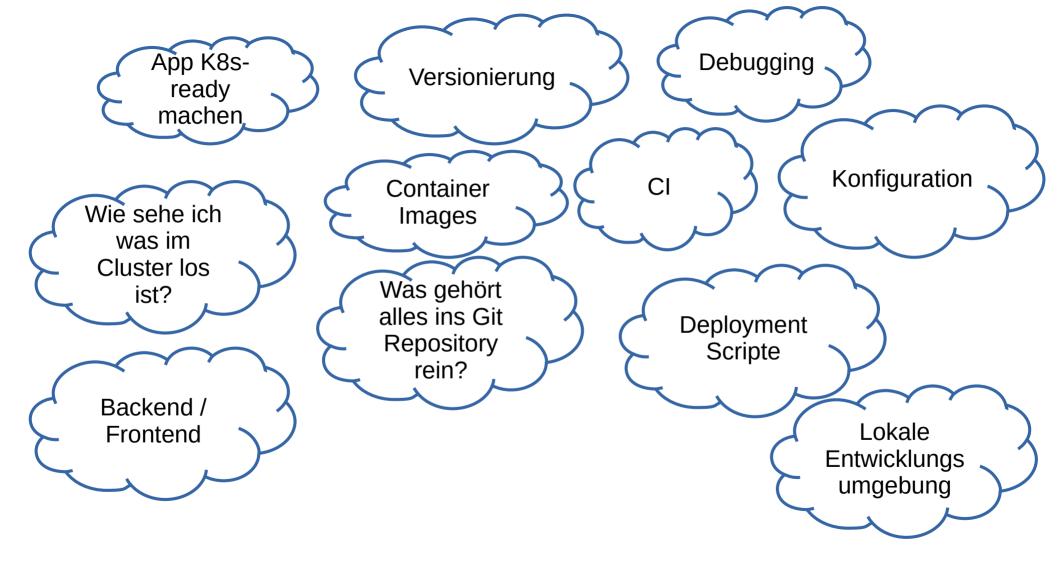






```
apiVersion: batch/v1
kind: CronJob
metadata:
  name: cronjob-demo
spec:
  schedule: "* * * * *"
  jobTemplate:
    spec:
      template:
        spec:
          containers:
          - name: hello-world-cronjob
            image: busybox
            imagePullPolicy: IfNotPresent
            command:
            - /bin/sh
            - -c
            - date; echo "Hello World, again"
          restartPolicy: OnFailure
```

#### Ein Begriffsbild für Kubernetes Das folgende Bild zeigt aus Sicht der Anwendungsentwicklung zentrale Begriffe von k8s im Zusammenspiel. Legende Allgemeines Konzept Clusterweite K8s-Obiekte CONTAINER **JOB DEPLOYMENT Organisiert in Namespaces** Kann Teil einer selbst-ge-Möglichkeit. Pods im Bietet deklarative Aktualisie-Batch-Stil auszuführen. rungen (z. B. Image-Version, oder auch Middleware. Ggf. auch parallel. Anzahl Repliken) CONFIGMAP mounted Entkopplung der umgebungsnutzt spezifischen Konfiguration beinhaltet kann ausführen VOLUME von Container-Images REPLICASET Speichermedium, auf das Stellt Ausführung einer Container zugreifen können ist spezielles bestimmter Anzahl von Podz. B. PersistentVolume stellt Anzahl Replikaten sicher. sicher ist ähnlich zu POD referenziert Die kleinste verteilbare Ein-**SECRET SERVICE** heit in K8s (Pod == engl. für Enthält sensible Daten wie ermöglicht Bietet einen Endpunkt für Kapsel). Zugriff auf ein Passwort, ein Token oder eine Menge von Pods. einen Schlüssel. gehört zu läuft auf macht von außen gehört zu zugreifbar **NAMESPACE** NODE struktu-Konzept um Ordnung im besteht **CLUSTER** Physischer oder virtueller riert Cluster zu halten, K8s-Obaus Rechner (VM), der die Arbeit Einstiegspunkt. Wenn Sie **INGRESS** jekte sind in der Regel einem macht (Worker). sich mit K8s "kubectl" verbin-Steuert Zugriffe von oder zugeordnet. den, tun Sie dies mit einem nach außen (Egress) Cluster.



Fragen?
mail@sandra-parsick.de

@sparsick@mastodon.social

https://github.com/sparsick/k8s-intro-talk

## Weitere Informationen

- https://www.informatik-aktuell.de/entwicklung/methode n/container-images-deep-dive-101-wege-zum-bauen-u nd-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/

## Bildnachweisweise

 https://dzone.com/articles/docker-containersand-kubernetes-an-architectural