enterjs, 22.06.2023

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

https://www.sandra-parsick.de

https://ready-for-review.dev


WHAT THE H*LL IS GOING ON HERE?


Ready for Review


A challenge.


NORTHERN JUSTICE

App K8s-ready machen

Versionierung

Debugging

Container Images

CI

Konfiguration

Wie sehe ich was im Cluster los ist?

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung

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

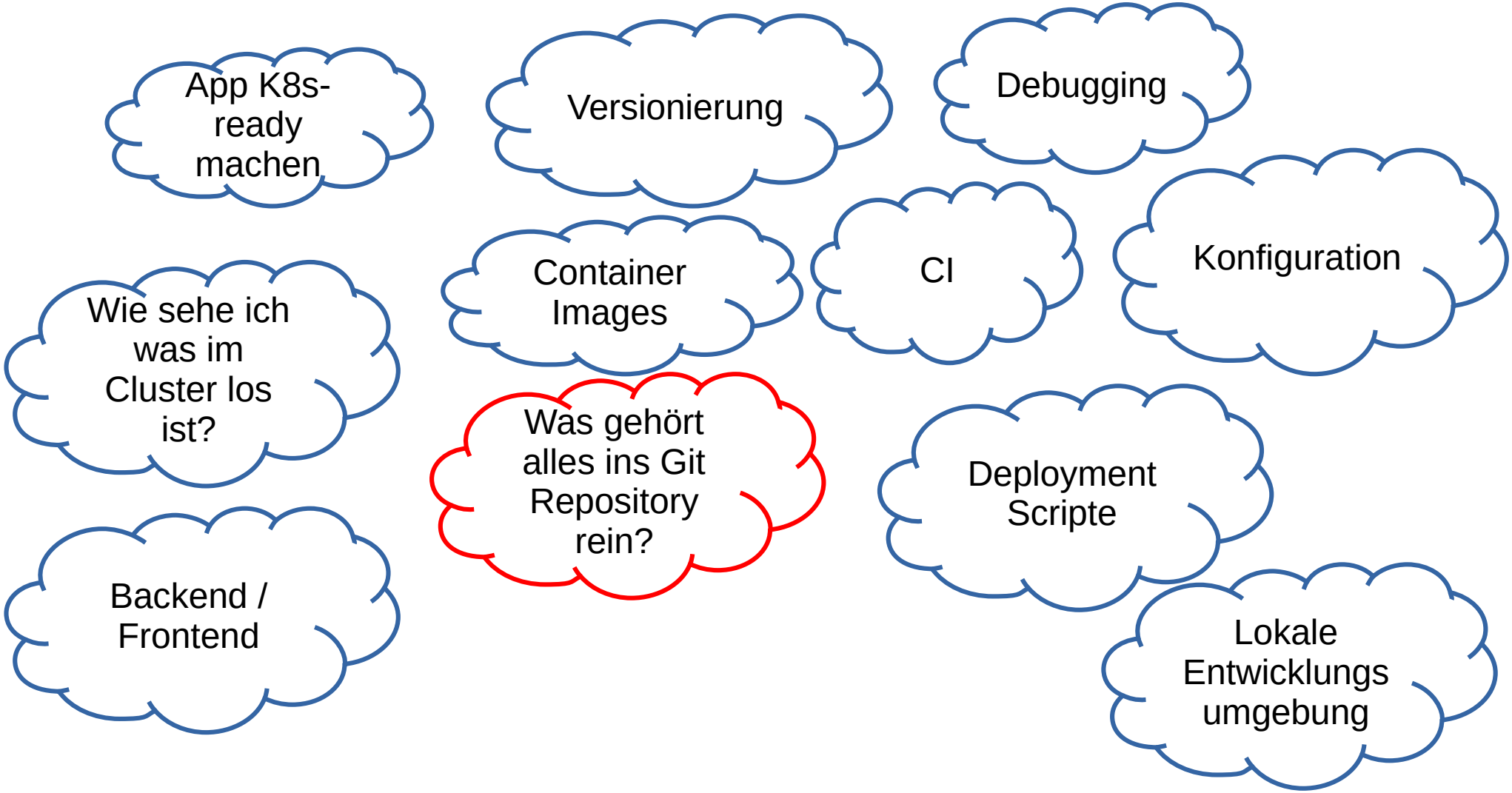
VIII. Concurrency
Scale out via the process model

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

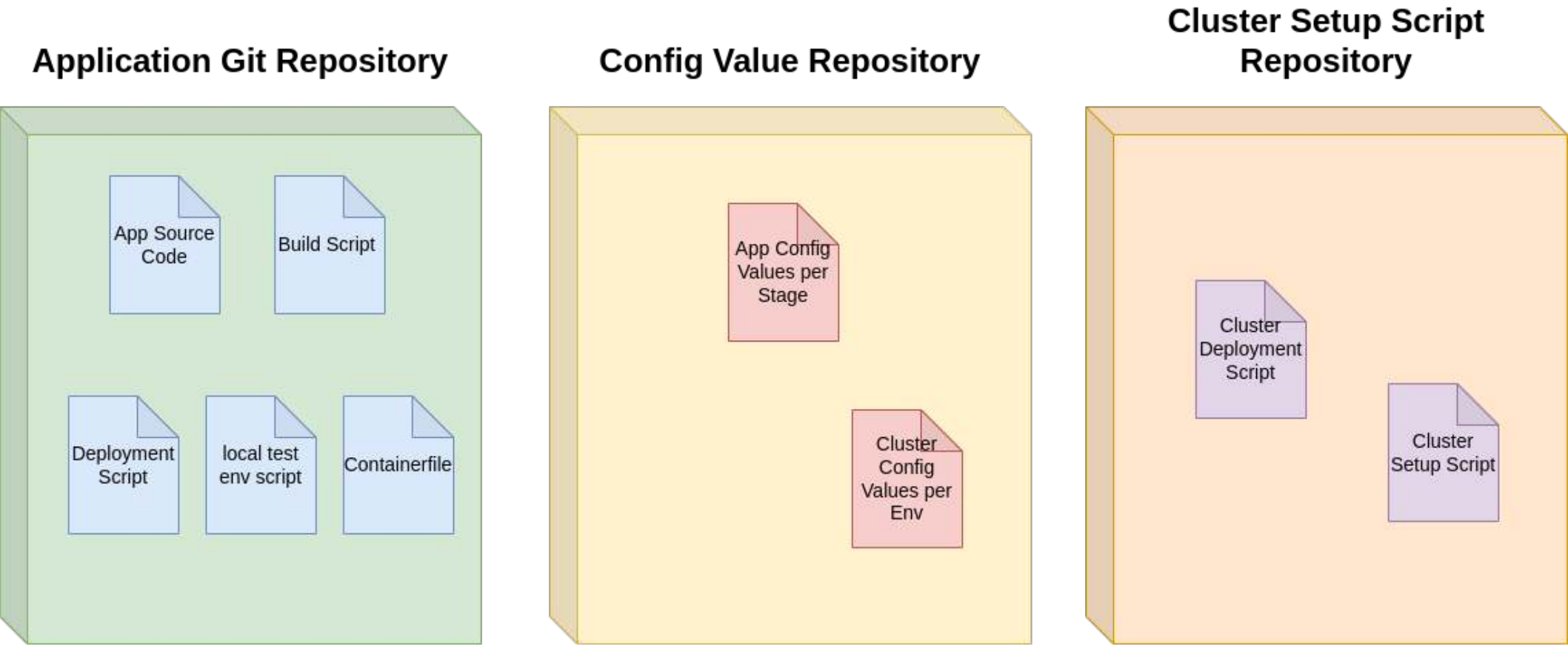
XII. Admin processes
Run admin/management tasks as one-off processes


App K8s-ready machen
Versionierung
Debugging
Container Images
CI
Konfiguration
Wie sehe ich was im Cluster los ist?
Was gehört alles ins Git Repository rein?
Deployment Scripte
Backend / Frontend
Lokale Entwicklungs umgebung

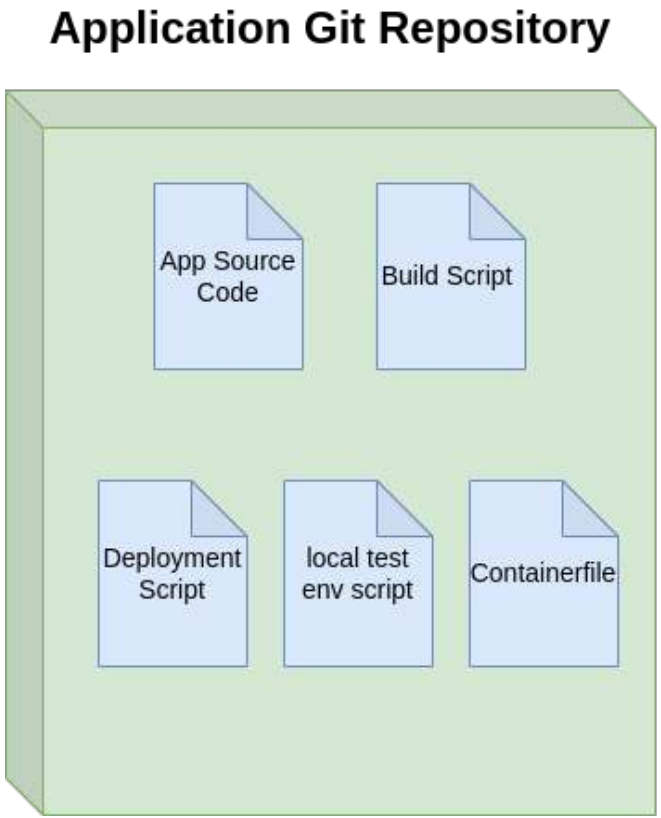
Kurzform: ALLES

Eigentliche Fragestellung:
Wieviele Repositories?

Beispiel für eine Aufteilung



Für Devs am wichtigsten



Ausgangspunkt eine nodejs App

```
.
├── app.js
├── bin
│   └── www
├── LICENSE
├── package.json
├── package-lock.json
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── README.md
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.pug
    ├── index.pug
    └── layout.pug
```

<u>Technologiestack</u>:

- Node.js
- Expressjs
- Npm

App K8s-ready machen

Versionierung

Debugging

Container Images

CI

Konfiguration

Wie sehe ich was im Cluster los ist?

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung

Basis: Container

```
.
├── app.js
├── bin
│   └── www
├── Dockerfile
├── .dockerignore
├── LICENSE
├── package.json
├── package-lock.json
├── public
│   ├── images
│   ├── javascripts
│   └── stylesheets
│       └── style.css
├── README.md
├── routes
│   ├── index.js
│   └── users.js
└── views
    ├── error.pug
    ├── index.pug
    └── layout.pug
```

Basis: Container

```
# Dockerfile
FROM node:20
WORKDIR /app
COPY package*.json ./
RUN npm ci --omit=dev
COPY . .

EXPOSE 3000
CMD ["node", "./bin/www"]

# .dockerignore
node_modules
npm-debug.log
.idea
.git
.gitignore
```

```
"scripts": {
  "start": "node ./bin/www",
  "docker": "docker build -t expressjs-demo .",
}
```

Alternativen

- Buildpacks
- JIB
- 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

```dockerfile
# Dockerfile
FROM node:20
WORKDIR /app
COPY package*.json ./
RUN npm ci --omit=dev
COPY . .

EXPOSE 3000
CMD ["node", "./bin/www"]
```

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 expressjs-demo:latest
2023-06-21T12:17:23.080+0200    INFO    Vulnerability scanning is enabled
2023-06-21T12:17:23.081+0200    INFO    Secret scanning is enabled
2023-06-21T12:17:23.081+0200    INFO    If your scanning is slow, please try '--scanners vuln' to
disable secret scanning
2023-06-21T12:17:23.081+0200    INFO    Please see also https://aquasecurity.github.io/trivy/v0.42
/docs/secret/scanning/#recommendation for faster secret detection
2023-06-21T12:17:39.936+0200    INFO    Detected OS: debian
2023-06-21T12:17:39.936+0200    INFO    Detecting Debian vulnerabilities...
2023-06-21T12:17:40.170+0200    INFO    Number of language-specific files: 1
2023-06-21T12:17:40.170+0200    INFO    Detecting node-pkg vulnerabilities...

expressjs-demo:latest (debian 12.0)

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

2023-06-21T12:17:40.346+0200    INFO    Table result includes only package filenames. Use '--format
json' option to get the full path to the package file.

Node.js (node-pkg)

Total: 1 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 1, CRITICAL: 0)
```

Vulnerability-Scans (Bsp.: Trivy)

Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
pug (package.json)	CVE-2021-21353	HIGH	2.0.4	3.0.1	pug: user provided objects as input to pug templates can achieve remote... https://avd.aquasec.com/nvd/cve-2021-21353

Vulnerability-Scans Weitergedacht

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

App K8s-ready machen

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung

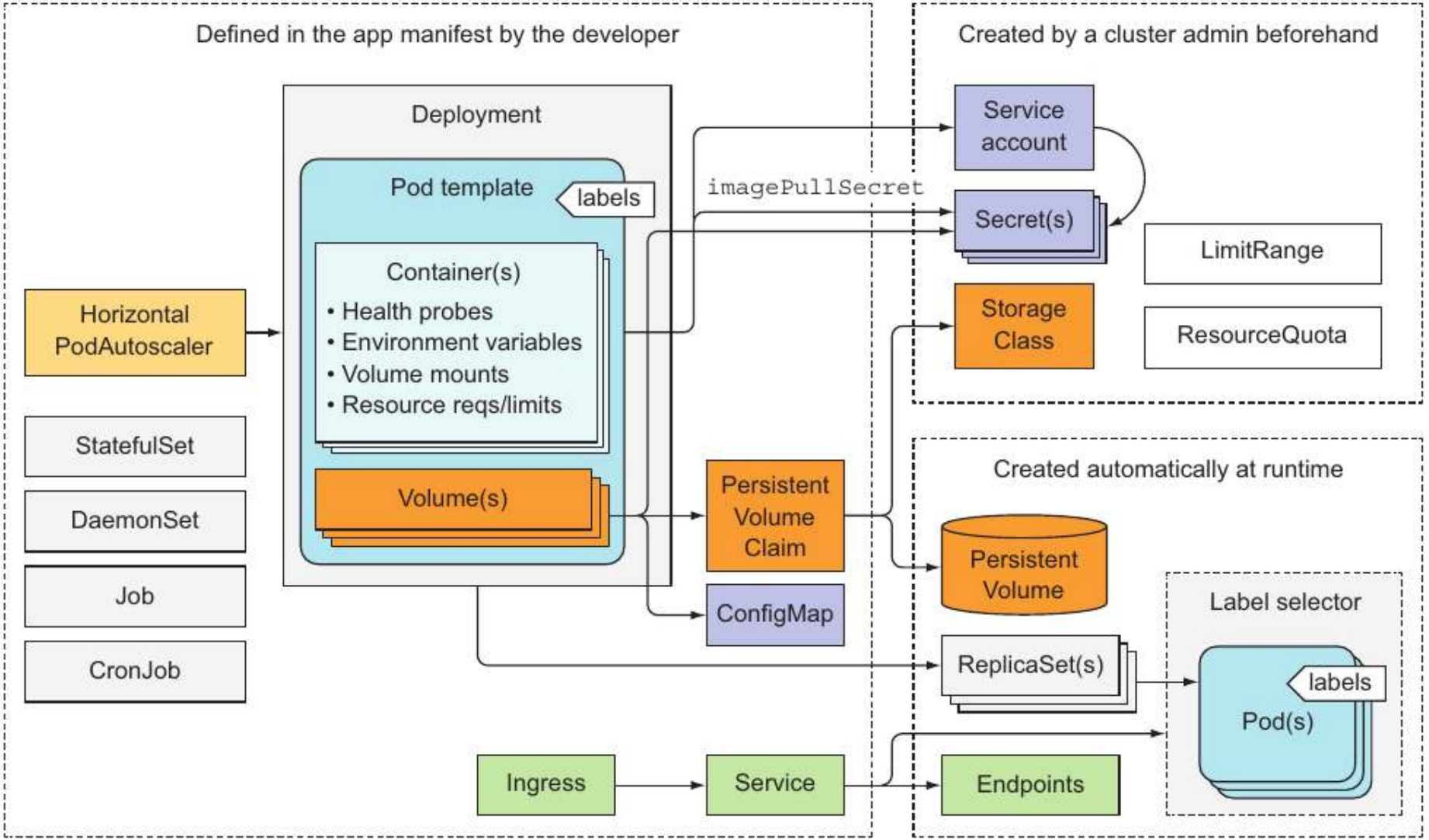
Next Step: Helm Charts

```
.
├── app.js
├── bin
│   └── www
├── deployment
│   └── helm-charts
│       └── expressjs-demo
│           ├── charts
│           ├── Chart.yaml
│           ├── local-values.yaml
│           ├── templates
│           │   ├── deployment.yaml
│           │   ├── _helpers.tpl
│           │   ├── ingress.yaml
│           │   ├── NOTES.txt
│           │   ├── service.yaml
│           │   └── tests
│           │       └── test-connection.yaml
│           └── values.yaml
├── Dockerfile
├── LICENSE
├── package.json
├── package-lock.json
├── public
├── README.md
├── routes
└── views
```

Auszug: Service Definition

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

Um welche K8s Resource soll ich mich als Dev kümmern?



Aus: Kubernetes in Action, 1st Edition

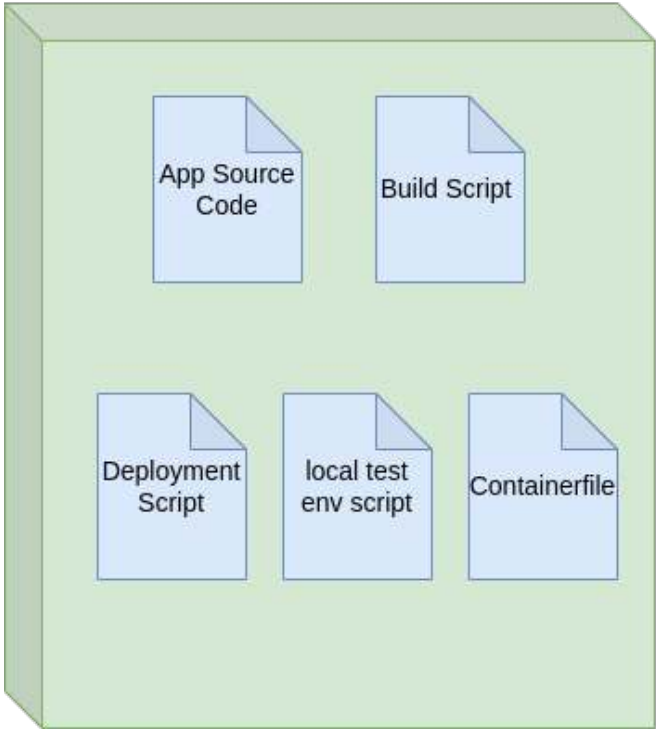
```
"scripts": {
  "start": "node ./bin/www",
  "docker": "docker build -t expressjs-demo .",
  "helm": "helm package ./deployment/helm-charts/expressjs-demo -d dist"
}
```

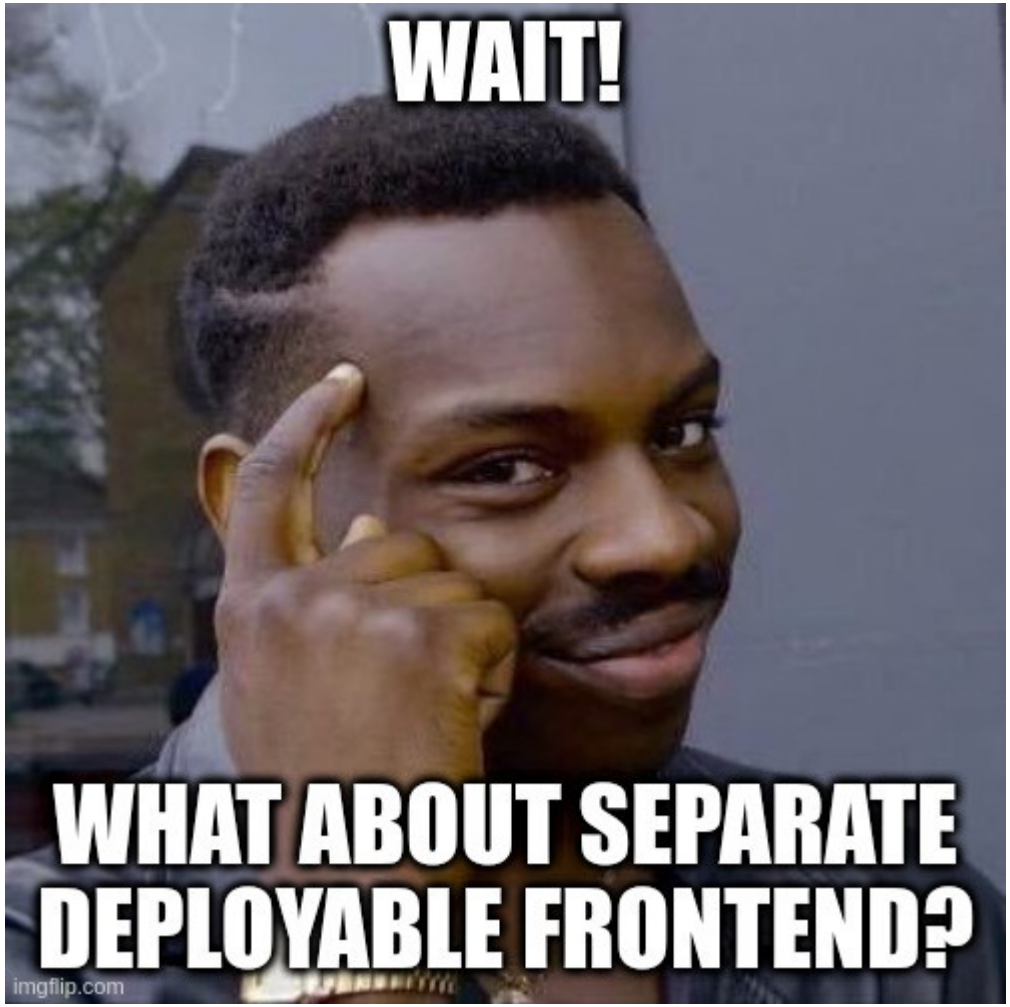
Helm Charts Paketierung Teil des Build Prozesses

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?
imgflip.com

App K8s-ready machen

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

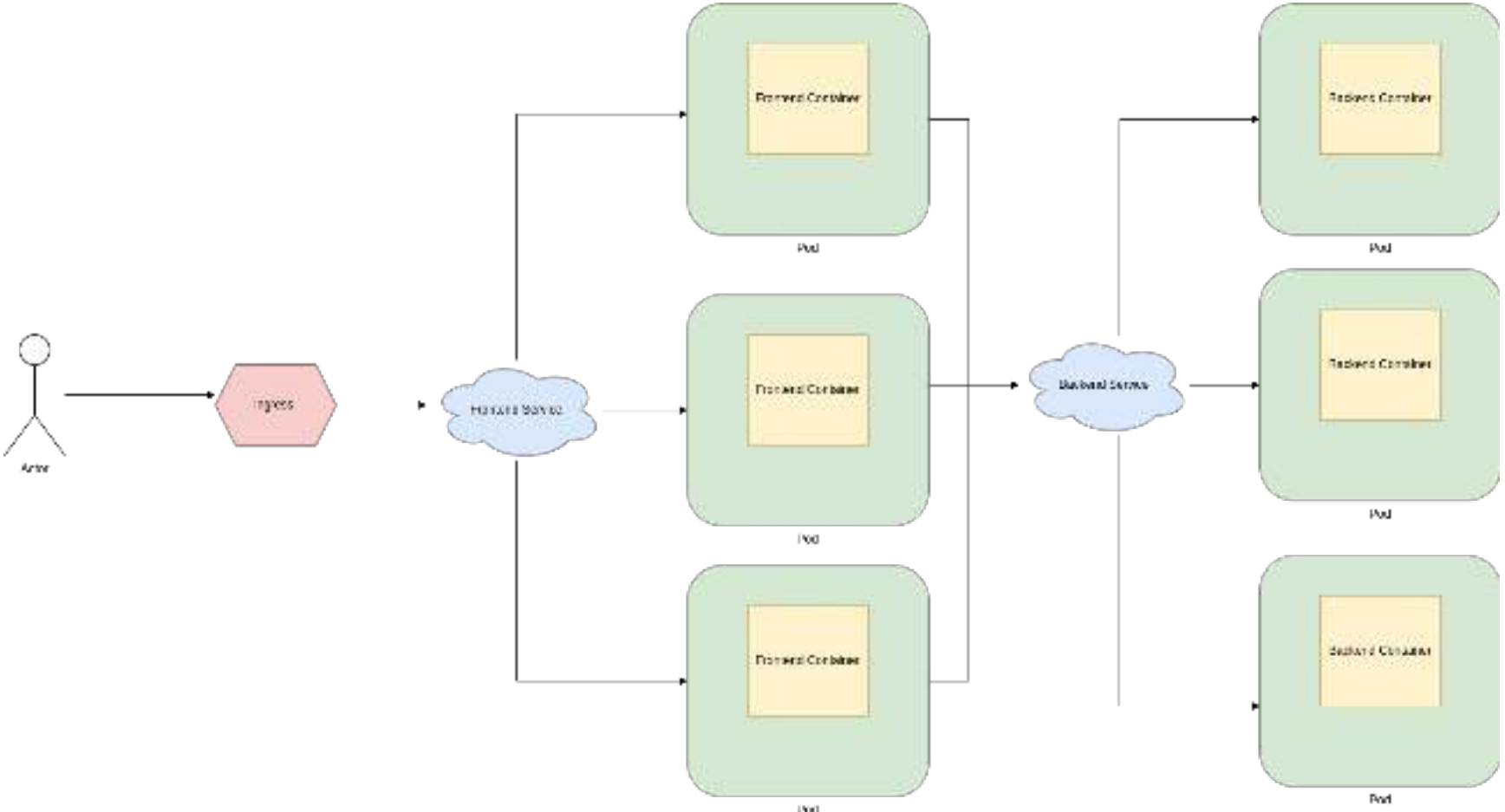
Was gehört alles ins Git Repository rein?

Deployment Scripte

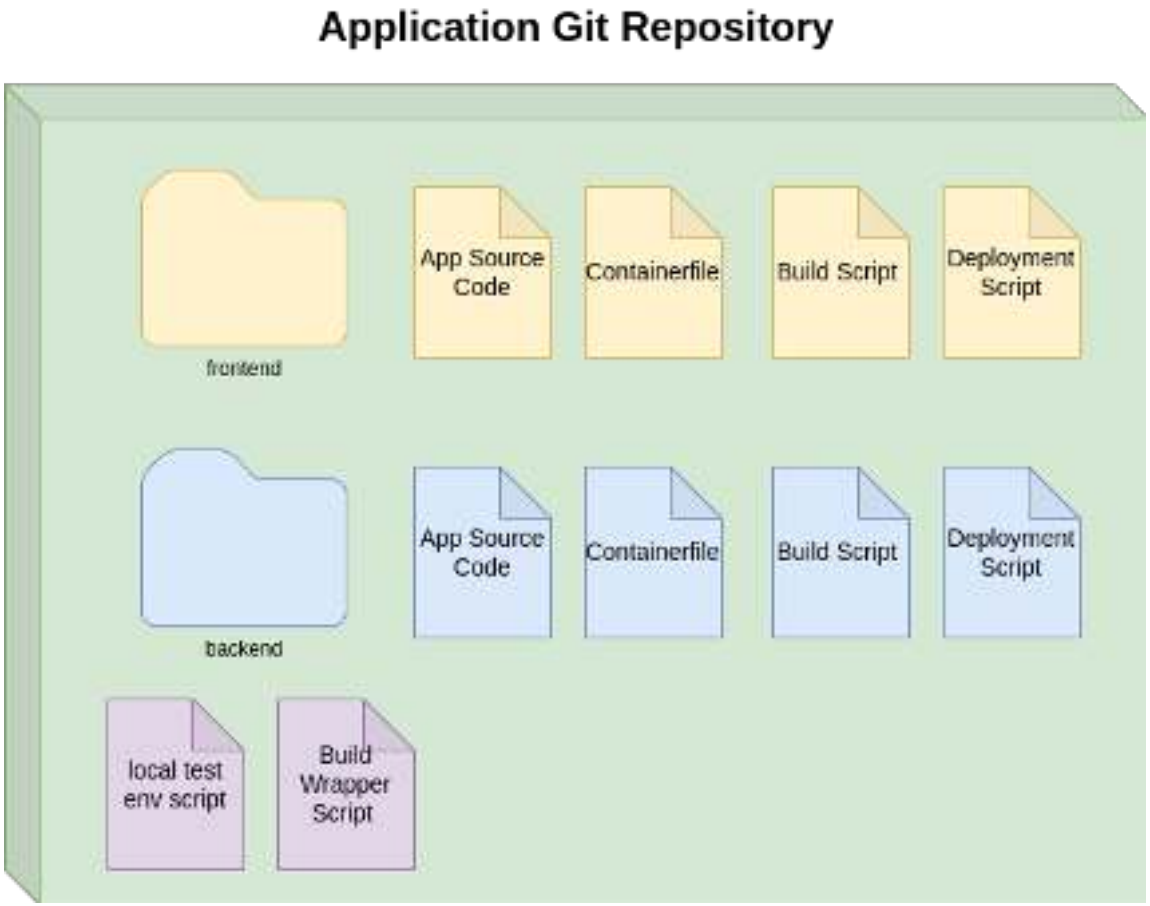
Backend / Frontend

Lokale Entwicklungs umgebung

Frontend und Backend in K8s



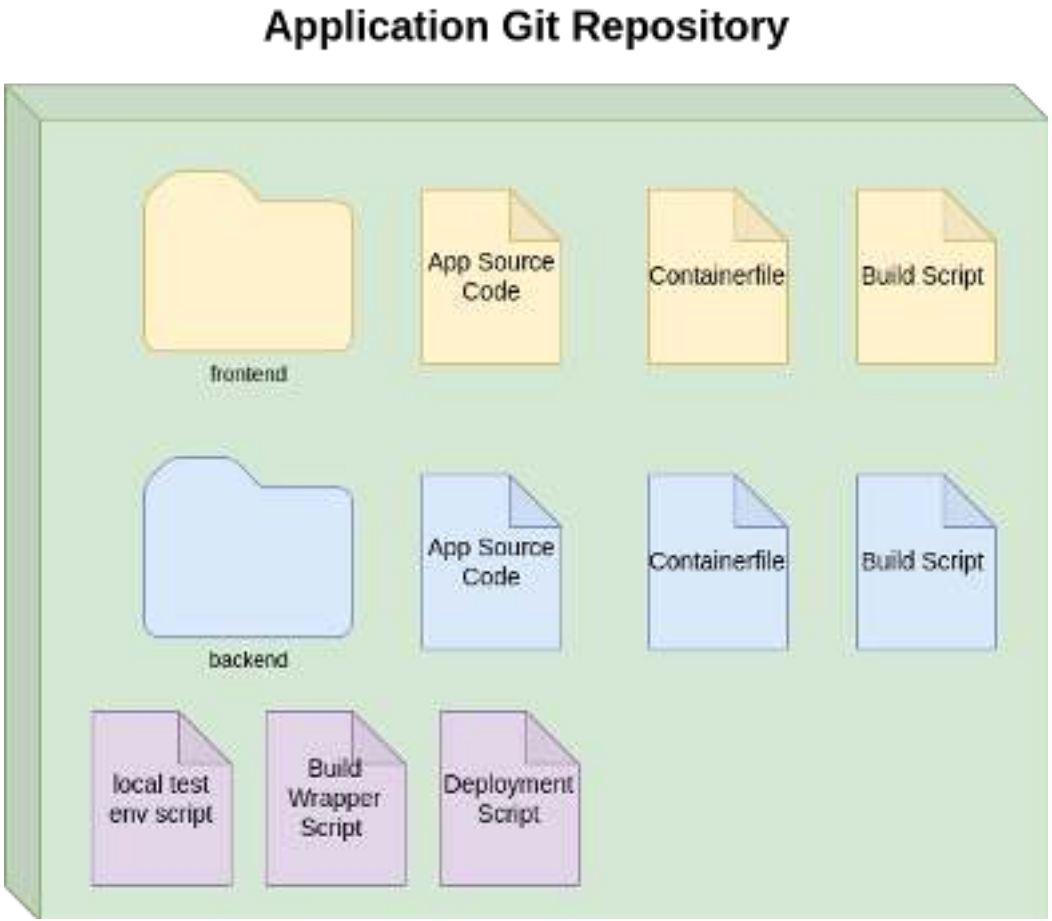
Git Repository Struktur

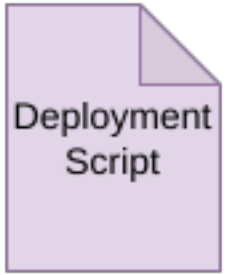


- Ingress
- Frontend Service
- Frontend Deployment

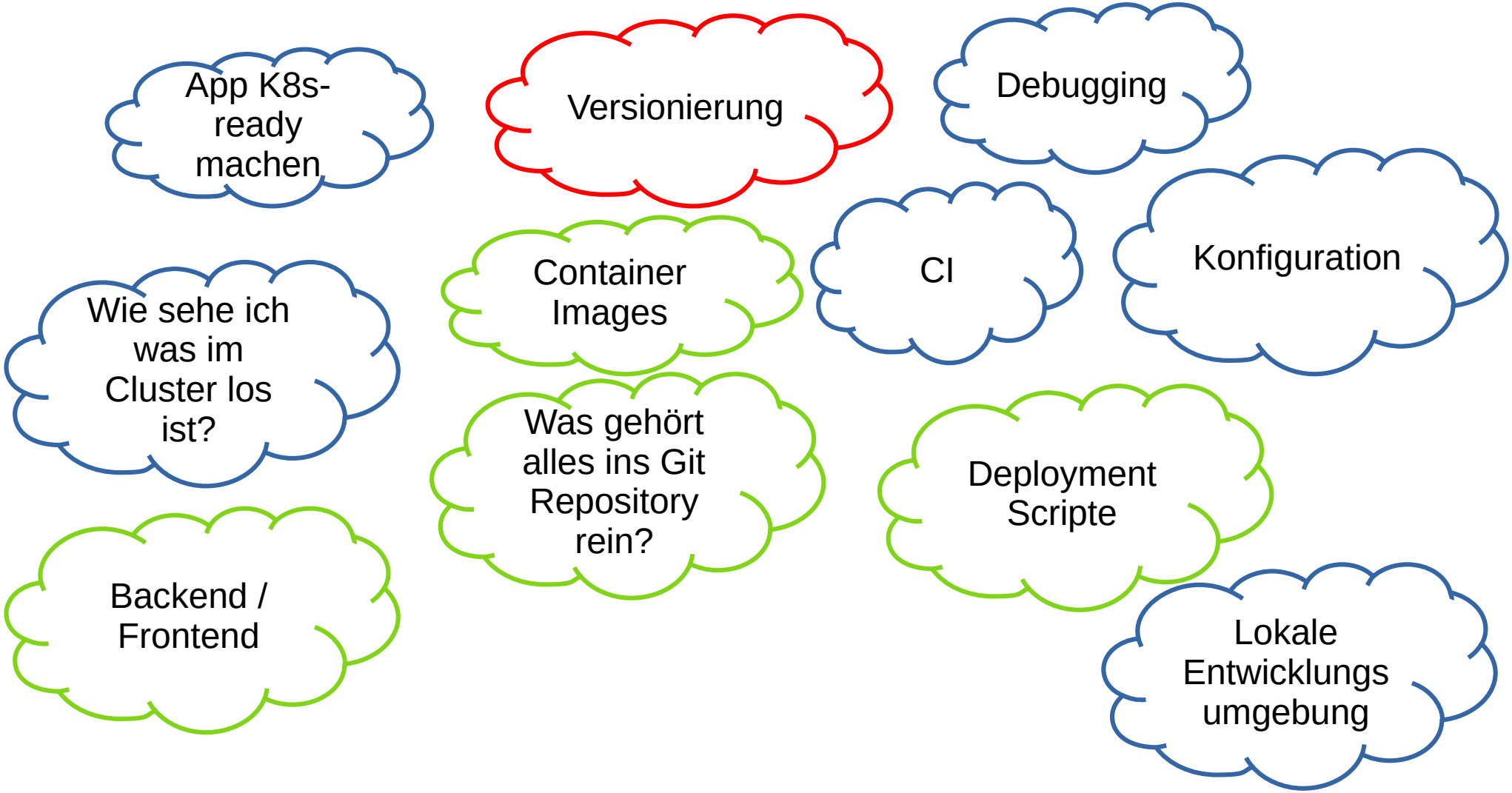
- Backend Service
- Backend Deployment

Git Repository Struktur



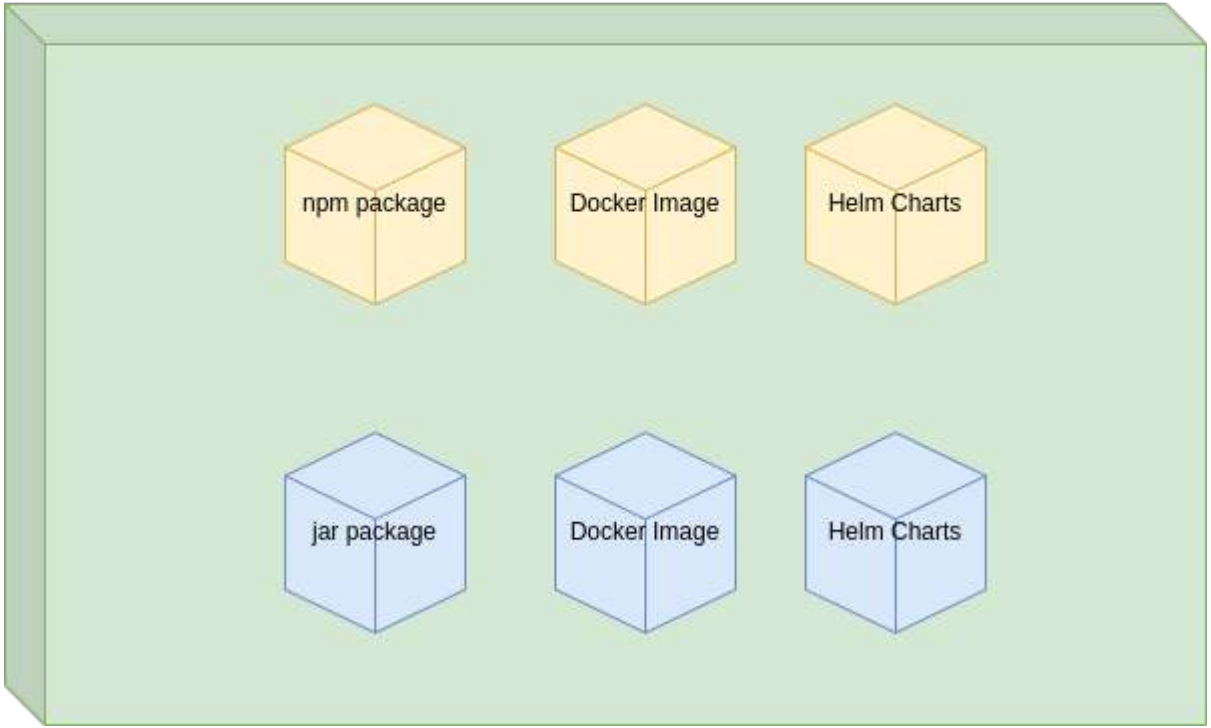


- Ingress
- Frontend Service
- Frontend Deployment
- Backend Service
- Backend Deployment


App K8s-ready machen
Versionierung
Debugging
Container Images
CI
Konfiguration
Wie sehe ich was im Cluster los ist?
Was gehört alles ins Git Repository rein?
Deployment Scripte
Backend / Frontend
Lokale Entwicklungs umgebung

Fangt einfach an:
Eine Versionsnummer über alle Artifakte

Application Artifacts



App K8s-ready machen

Versionierung

Debugging

Container Images

CI

Konfiguration

Wie sehe ich was im Cluster los ist?

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung

Applikation
lokal testen

Deployment
Skripte lokal
entwickeln

Applikation lokal testen

Applikation lokal testen

```yaml
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/
```

```
# .env
MONGODB_URI: mongodb://test:test123@localhost:27017/test


# package.json
"scripts": {
    "start": "node ./bin/www"
  }

→ npm start
```

Andere
Abhängigkeiten?

Mocking

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

```yaml
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
```

```json
[
  {
    "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

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung

12 Factor App:
Die Konfiguration in Umgebungsvariablen ablegen

Applikation vorbereiten

```
→ npm install dotenv --save
```

```
require('dotenv').config()

process.env.MONGODB_URI
```

Helm Charts anpassen

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "expressjs-demo.fullname" . }}
data:
  MONGODB_URI: {{ .Values.expressjs.mongoDBUri }}
  ENV: {{ .Values.expressjs.env }}
```

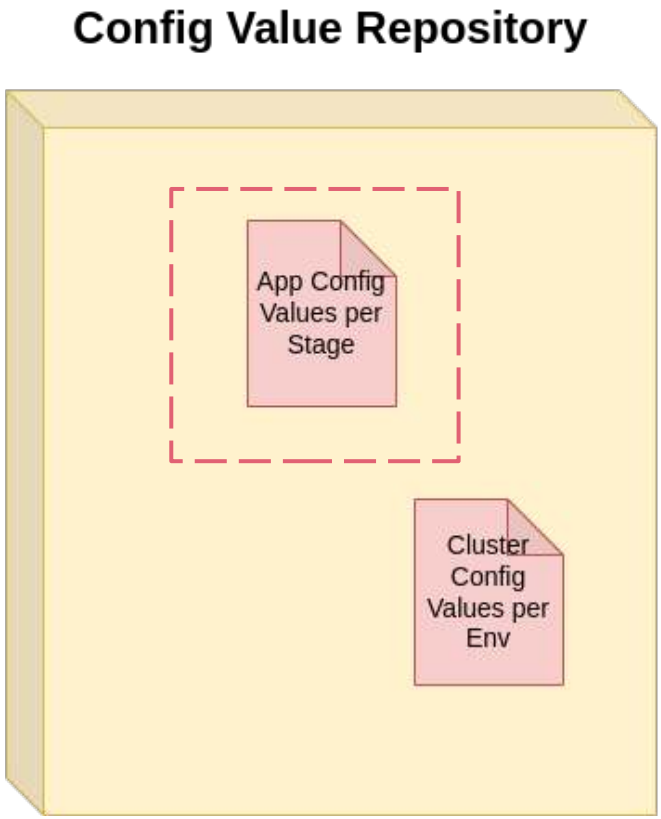
Helm Charts anpassen

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "expressjs-demo.fullname" . }}
spec:
  template:
    metadata:
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag | default .Chart.AppVersion }}"
          imagePullPolicy: {{ .Values.image.pullPolicy }}
          envFrom:
            - configMapRef:
                name:  {{ include "expressjs-demo.fullname" . }}
```

Helm Charts anpassen

```yaml
# code snippet from values.yaml
expressjs:
  mongoDBUri: mongodb://mongo:27017/expressjs-demo
```

Konfiguration verwalten



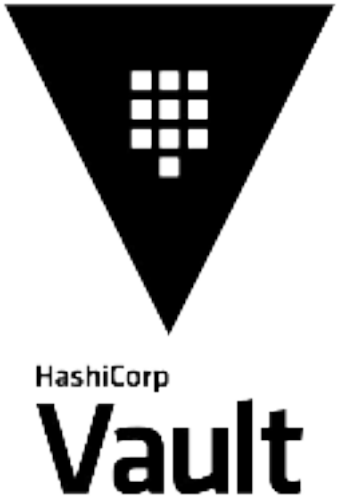
Konfiguration verwalten

```
config-value-repo on ᚠ dev
→ tree
.
├── namespace-a
│   └── app1.yml
└── registry.yaml

→ git branch
* dev
  pre-prod
  prod
```

```
flat-config-value-repo on ᚠ 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

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

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

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

```
→ npm install winston --save
```

```
var winston = require('winston');

var logger = winston.createLogger({
    transports: [
        new winston.transports.Console({
            level: 'info',
            handleExceptions: true,
            json: true,
            colorize: true
        })
    ],
    exitOnError: false
});

module.exports = logger;
module.exports.stream = {
    write: function(message, encoding){
        logger.info(message);
    }
}
```

Log-Nachrichten auf stdout

```
var logger = require("./utils/logger");

logger.info("Hello World")

app.use(require("morgan")("combined", { stream: logger.stream }));
```

Anwendungsüberwachung

```
const express = require('express');
const app = express();
const promBundle = require("express-prom-bundle");

// Add the options to the prometheus middleware most option are for http_request_duration_seconds
histogram metric
const metricsMiddleware = promBundle({
    includeMethod: true,
    includePath: true,
    includeStatusCode: true,
    includeUp: true,
    customLabels: {project_name: 'hello_world', project_type: 'test_metrics_labels'},
    promClient: {
        collectDefaultMetrics: {
        }
      }
});
// add the prometheus middleware to all routes
app.use(metricsMiddleware)

// curl http://localhost:3000/metrics
```

```
npm install prom-client express-prom-bundle --save
```

Robust hoch- und runterfahren

```
npm install express-actuator --save
```

```
const actuator = require('express-actuator');
const app = express();

app.use(actuator());
```

Robust hoch- und runterfahren

- Weitere Express JS Module:
  - Terminus
  - Lightship
  - http-terminator
- https://expressjs.com/en/advanced/healthcheck-graceful-shutdown.html

Robust hoch- und runterfahren

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "expressjs-demo.fullname" . }}
spec:
  template:
    metadata:
    spec:
      containers:
        - name: {{ .Chart.Name }}
          image: "{{ .Values.image.repository }}:{{ .Values.image.tag | default .Chart.AppVersion }}"
          ports:
            - name: http
              containerPort: 3000
              protocol: TCP
          livenessProbe:
            httpGet:
              path: /health
              port: http
          readinessProbe:
            httpGet:
              path: /health
              port: http
```

Robust hoch- und runterfahren

Wichtig:
Sichert diese Endpunkte nach außen ab!

Robust hoch- und runterfahren

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ include "expressjs-demo.fullname" . }}
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /$1
    nginx.ingress.kubernetes.io/x-forwarded-prefix: "/"
    nginx.ingress.kubernetes.io/server-snippet: |
      location ~* "^/health/" {
          deny all;
          return 404;
        }
```

App K8s-ready machen

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

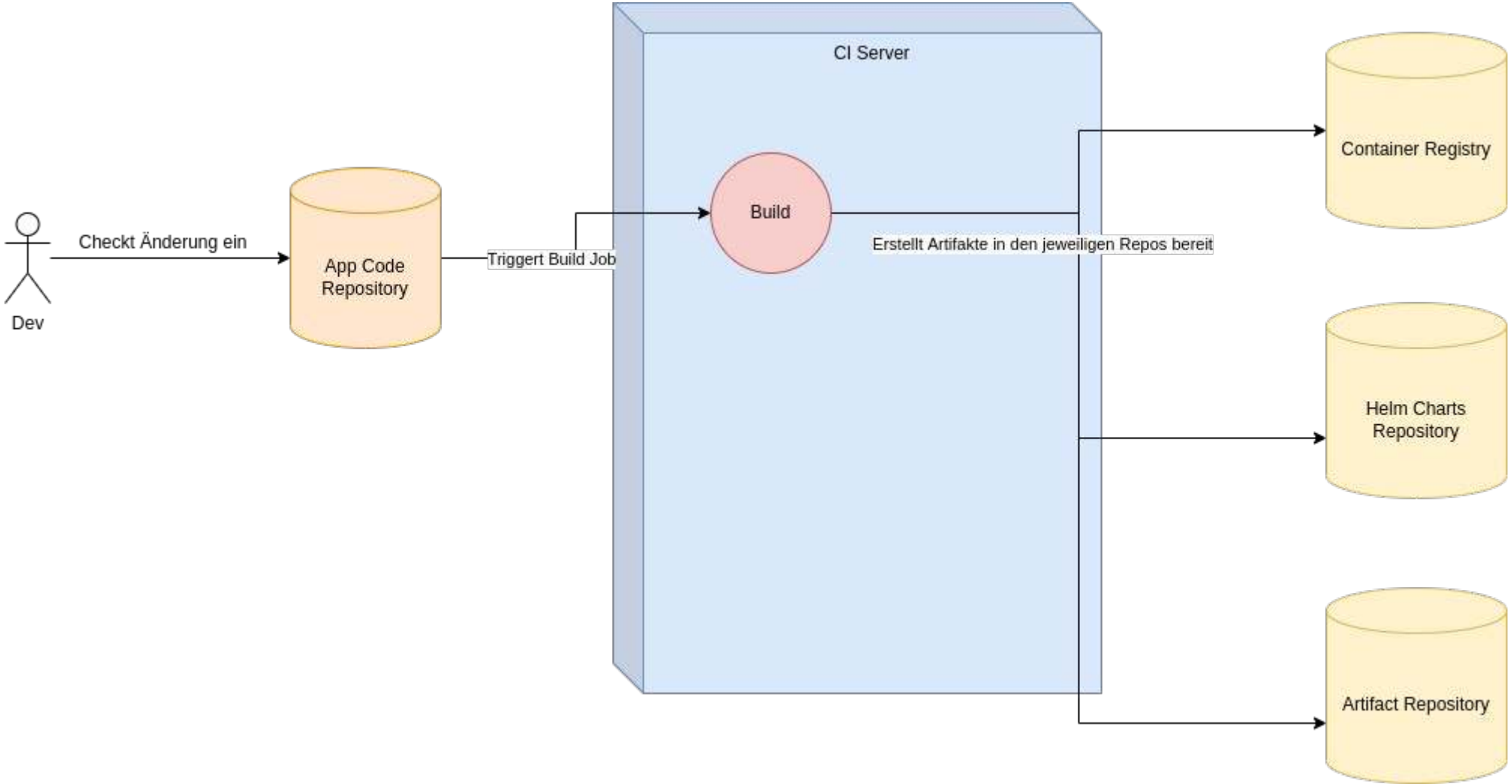
Deployment Scripte

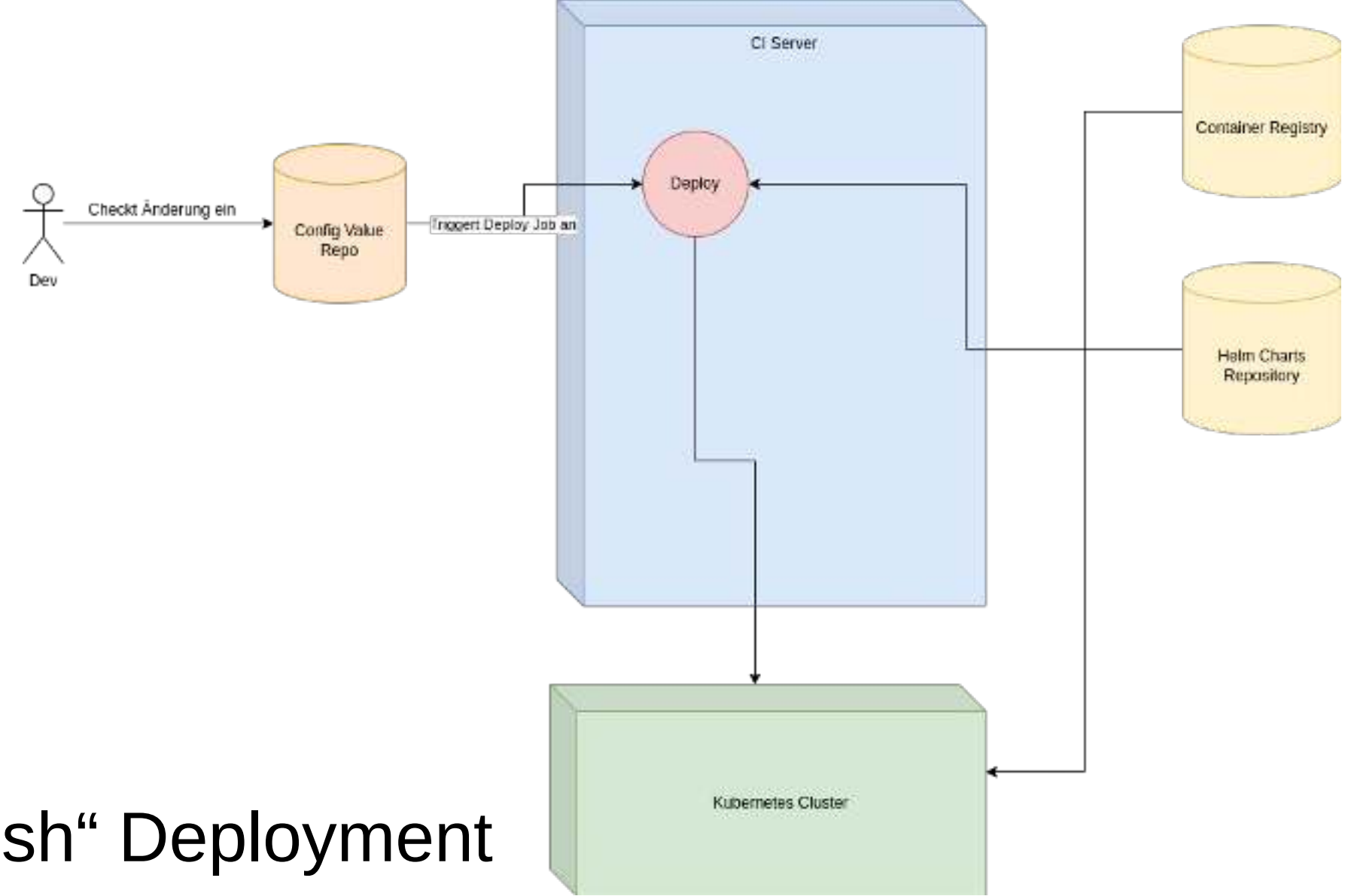
Backend / Frontend

Lokale Entwicklungs umgebung

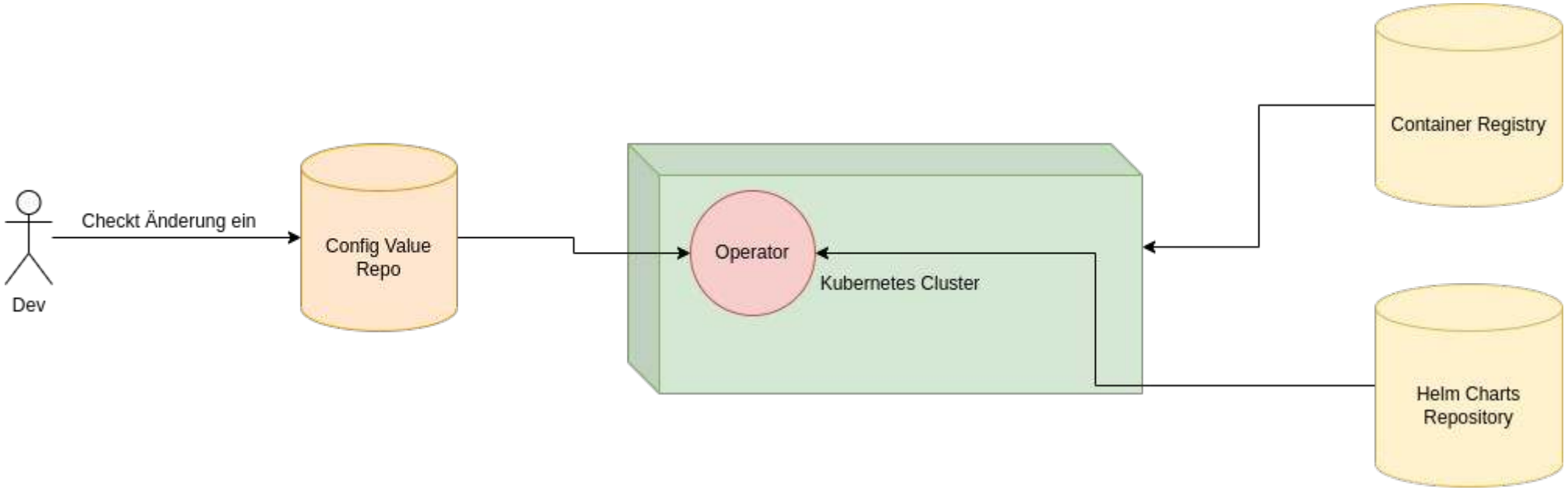
12 Factor App:
Build- und Run-Phase strikt trennen

Build


CI Server
Dev
Checkt Änderung ein
App Code Repository
Triggert Build Job
Build
Erstellt Artifakte in den jeweiligen Repos bereit
Container Registry
Helm Charts Repository
Artifact Repository



„Push“ Deployment



„Pull“ Deployment

App K8s-ready machen

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

Deployment Scripte

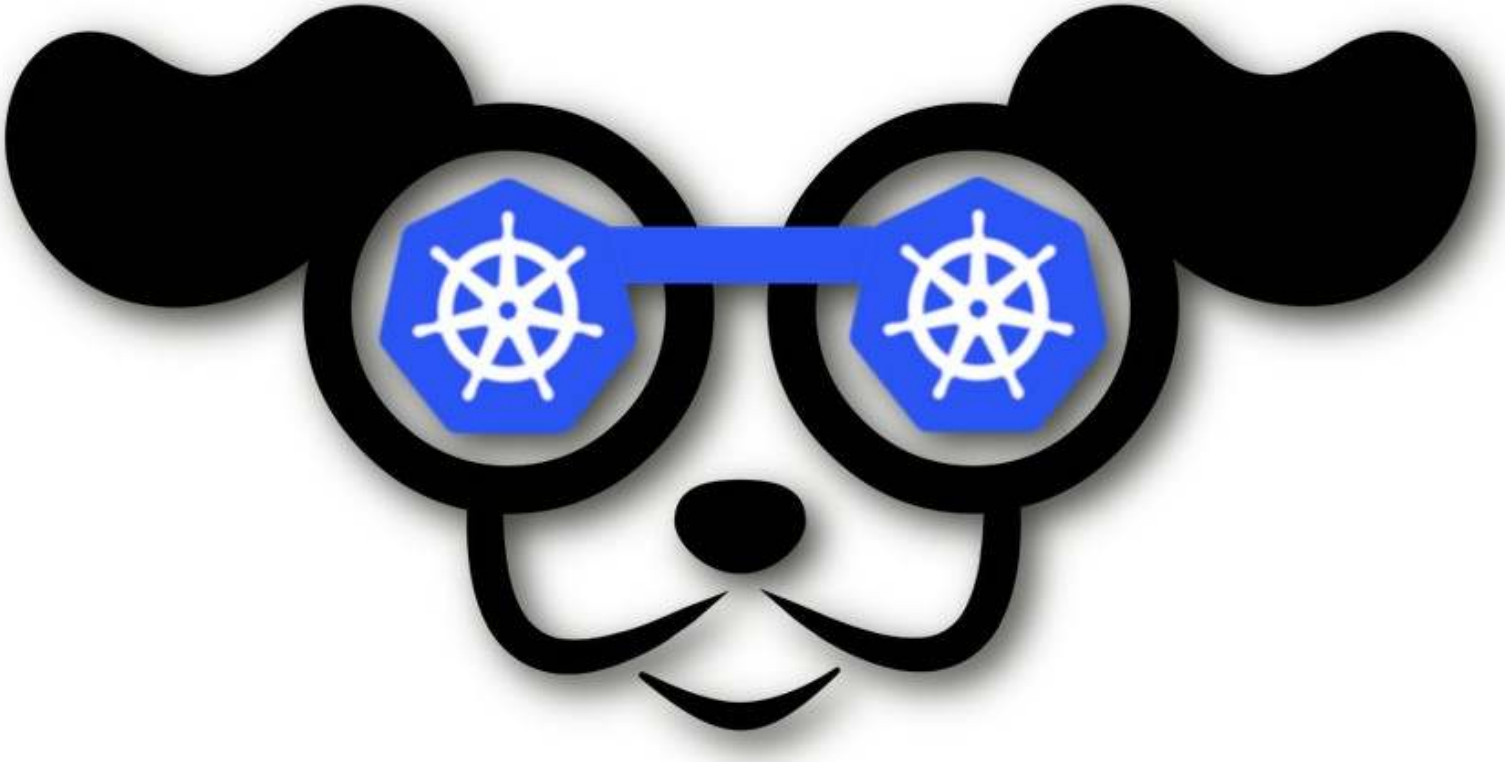
Backend / Frontend

Lokale Entwicklungs umgebung

kubectl

k9s

Kubernetes CLI To Manage Your Clusters In Style!



K8s Lens / Open Lens

Monokle Desktop

App K8s-ready machen

Versionierung

Debugging

Container Images

CI

Konfiguration

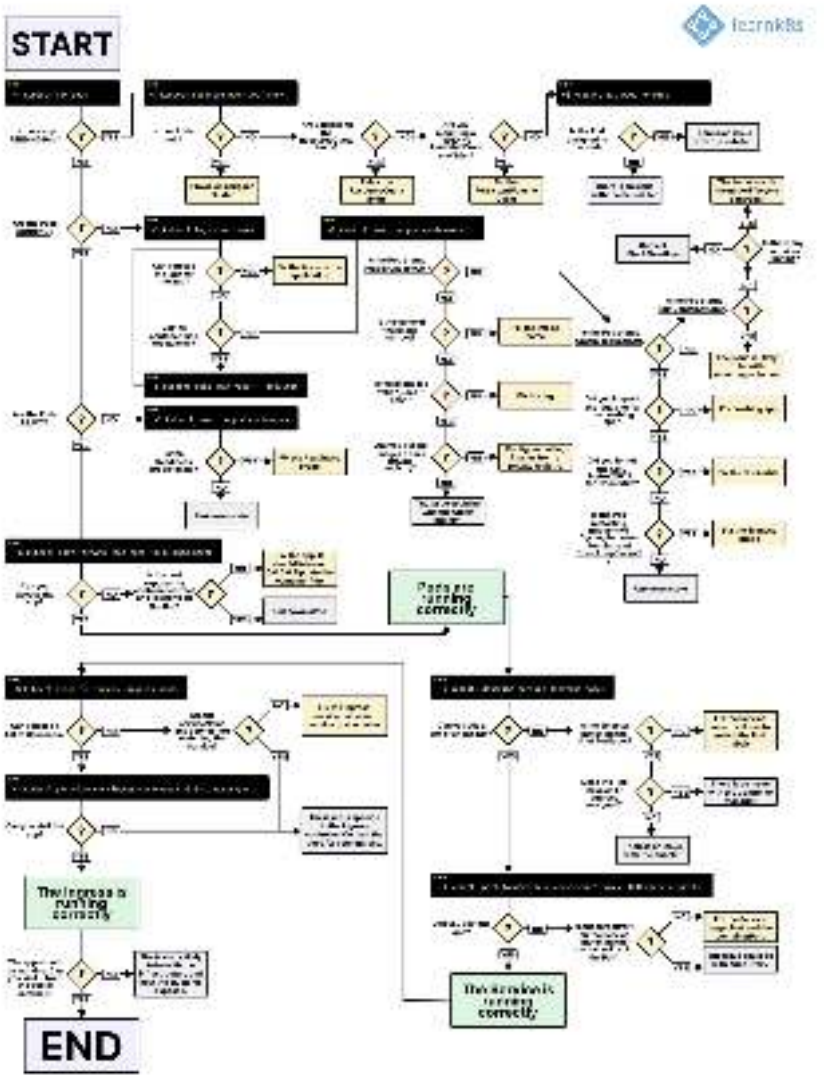
Wie sehe ich was im Cluster los ist?

Was gehört alles ins Git Repository rein?

Deployment Scripte

Backend / Frontend

Lokale Entwicklungs umgebung



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.

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

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

Versionierung

Debugging

Wie sehe ich was im Cluster los ist?

Container Images

CI

Konfiguration

Was gehört alles ins Git Repository rein?

Deployment Scripte

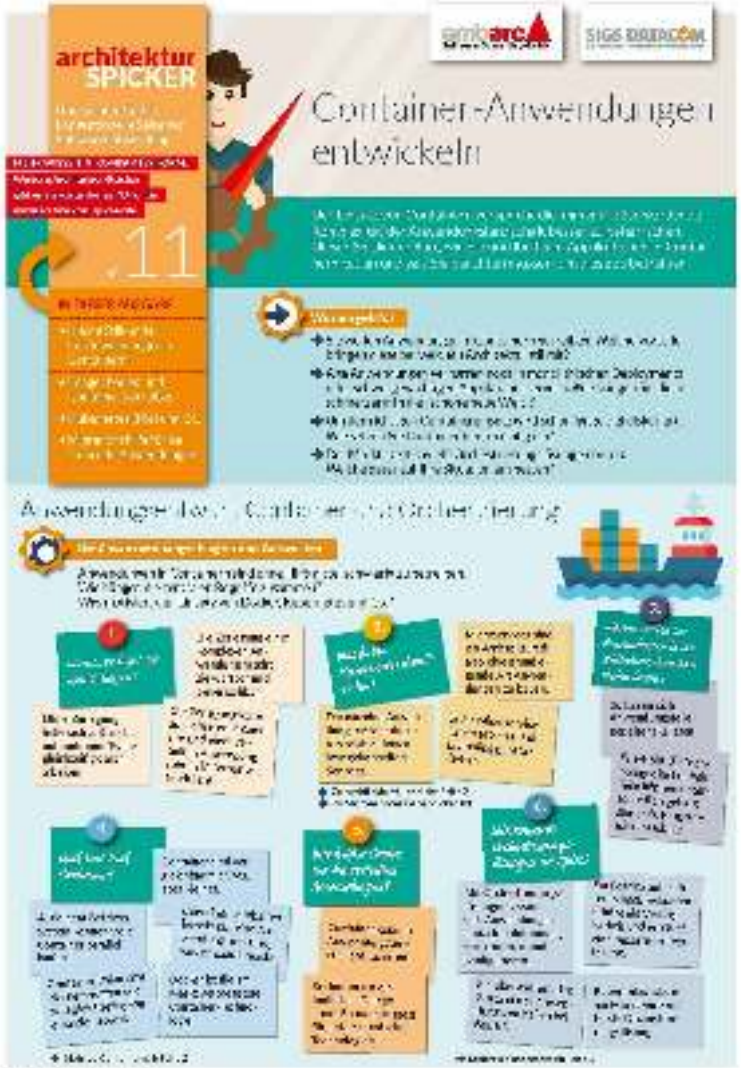
Backend / Frontend

Lokale Entwicklungs umgebung

Fragen?

mail@sandra-parsick.de
@SandraParsick
@sparsick@mastodon.social

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


architektur SPICKER
.11
Container-Anwendungen entwickeln


architektur SPICKER
.7
Continuous Delivery

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/CpsTAUPoScw?utm_source=unsplash&utm_medium=referral&utm_content=creditShareLink