“reconcile it” - Kubernetes und die Macht der Operatoren

About me

- 9 Jahre in der IT
- Focus auf SRE und Developer Productivity Engineering (DPE)
- Home Automation
- Sport und Outdoor begeistert
- LinkedIn: https://www.linkedin.com/in/rico-pahlisch-013925232/

About GrayC

- Erfahrungen & Schwerpunkte:
  - Digitale Produktentwicklung als Ganzheitlicher Ansatz
    - Product Visioning & Ownership
    - SW-Entwicklung & -Architektur
    - Aufbau und Management von skalierfähiger Cloud Infrastruktur
    - IoT-Lösungen und IoT Flottenmanagement
    - cross funktionale Teams
    - lernende Organisationen
- Mission - Bringing Humans and IT into Harmony:
  - **Wir unterstützen unsere Kunden und Partner mit unserer Erfahrung beim Aufbau und der Entwicklung nachhaltiger, digitaler Produkte.**

Inhaltsverzeichnis

These

Die Anzahl der Tools und Technologien, mit denen sich Entwickler auseinandersetzen müssen, hat sich in den letzten Jahren deutlich erhöht. Neben der reinen Entwicklung müssen sich Entwickler deutlich mehr mit Infrastruktur auseinandersetzen.

Entwickler Alltag

- Arbeit auf den Host Rechner
- Entwickler Rechner
- Entwickeln in einer VM
  - Vorkonfiguriert mit allen Abhängigkeiten
  - einfach zu verteilen
- Docker
  - externe Abhängigkeiten
  - zum Verpacken der Anwendung

Neue Technologien

- Microservice
- Datenbanktechnologien
  - Relationale Datenbanken
    - MySQL,
    - Postgres
  - Document Store
    - Elastic Search
    - MongoDB
  - Caches
    - Redis
    - KeyDB
- Message Queues
  - Kafka
  - RabbitMQ
- etc.

https://www.okteto.com/blog/okteto-hearts-content/

Infrastrukturmanagement

- Terraform / Pulumi
- Ansible
- Puppet
- Chef
- Custom Scripts
- etc.

The Way to Production

- Anträge ausfüllen
- Berechtigungen
- Zugangsdaten für meine Application
  - wo kommen die Secrets her und wie kommen die sicher zu meiner Application
- wo bekomme ich meine Datenbank her
  - backups etc


Wir Gallier würden gerne eine Tagung machen...
Passierschein A38 gibt es drei Treppen tiefer, Abteilung Hygiene...

Kubernetes (k8s)

Kubernetes, auch k8s genannt (beginnt mit k, hat 8 Zeichen, endet mit s), oder kurz „kube“ ist eine Open Source-Plattform, die den Betrieb von Linux-Containern automatisiert. Dabei werden viele der manuellen Prozesse, die mit der Bereitstellung und Skalierung von containerisierten Anwendungen einhergehen, eliminiert. *

https://www.redhat.com/de/topics/containers/what-is-kubernetes

Kubernetes Don’ts

https://www.globalnerdy.com/2020/07/17/how-you-deploy-a-container-on-kubernetes/

Kubernetes 1x1 Bausteine

- Pod
- Configmap
- Secret
- Service
- Ingress

- Deployment
- Statefulset
- Daemonset
- Replicaset
- Job/Cronjob

Kubernetes 1x1 Komponenten

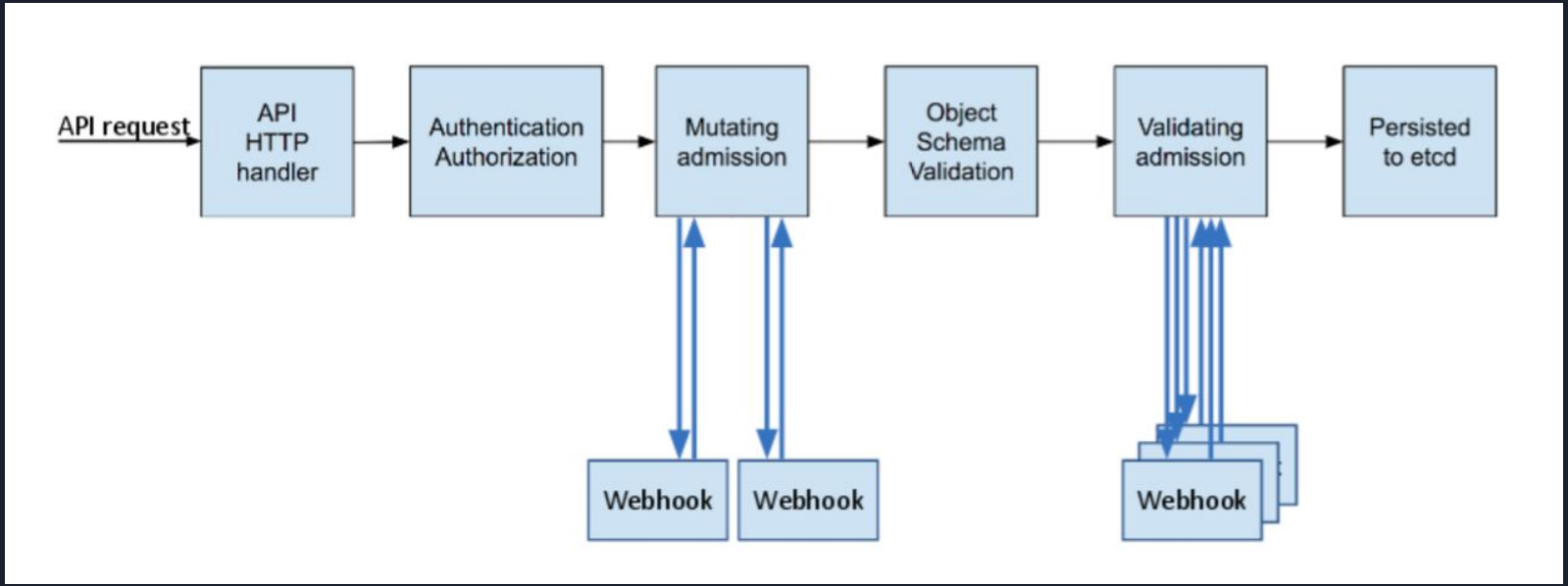
- kube-api-server
- kube-controller-manager
  - Operator für alle Standard Ressourcen
- kube-scheduler
- kubelet -> Node Agent
- kube-proxy -> Network
- etcd -> Store

https://kubernetes.io/docs/reference/command-line-tools-reference/

Kubernetes local

- minikube
- k3s / k3d
- micro k8s
- kind - Kubernetes in Docker

Kubernetes Webhooks Request Flow



https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/

k8s Webhooks

- validating
  - parallel
  - keine Veränderung des Payloads
  - kann den Request fehl schlagen lassen oder nur zu einem Warning führen
- mutating
  - Veränderung des Payloads
  - z.b. Versions Migration v1beta1 -> v1

k8s Authorization Webhook

- https://kubernetes.io/docs/reference/access-authn-authz/webhook/
- config des API Servers

```
kube-apiserver
--authorization-mode=RBAC,Webhook
 --authorization-webhook-version=v1
 --authorization-webhook-config-file=/etc/kubernetes/api-server/authz-webhook.yaml
 --authorization-webhook-cache-authorized-ttl=120s
 --authorization-webhook-cache-unauthorized-ttl=30s
```

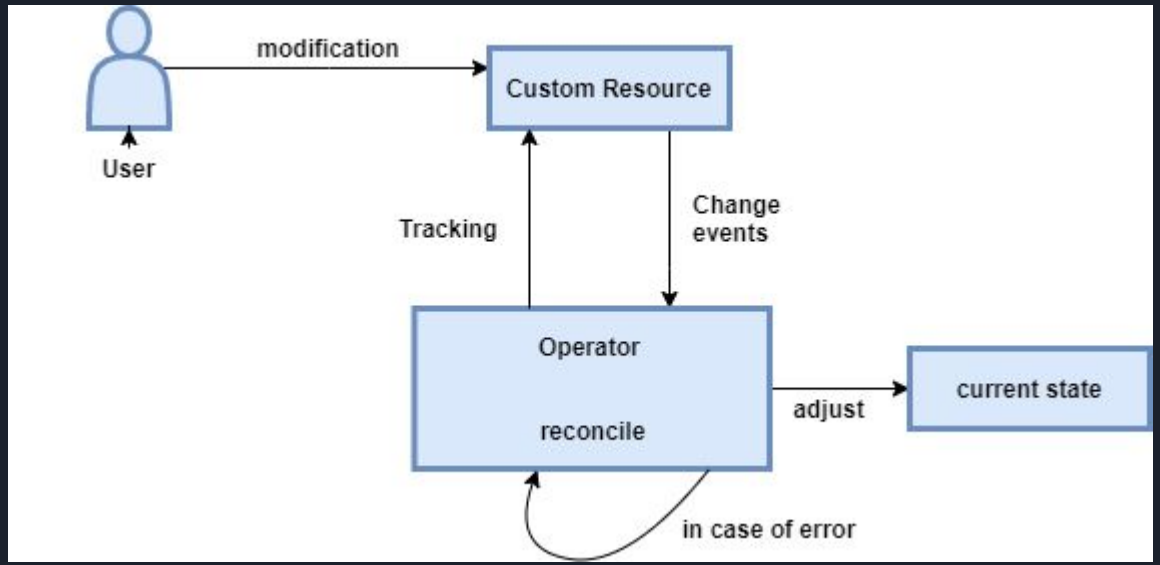
Und wie hilft mir das jetzt?

Custom Resource Definition

“Custom resources are extensions of the Kubernetes API.”

```yaml
apiVersion: jugsaxony.org/v1beta1
kind: Vortrag
metadata:
 name: reconcile-it
spec:
 thema: 'RECONCILE IT - KUBERNETES UND DIE MACHT DER OPERATOREN'
status:
 phase: IN_PROGRESS
```

Kubernetes Reconciliation Loops



https://www.kubermatic.com/blog/kubernetes-operators-automating-complex-application-lifecycles/

Operator Frameworks

- Kubebuilder
  - https://book.kubebuilder.io/introduction.html
- Operator SDK
  - https://sdk.operatorframework.io/
- Java Operator SDK
  - https://javaoperatorsdk.io/

Operator Entwicklung Tipps

- Finalizer
  - verhindern löschen
  - https://kubernetes.io/docs/concepts/overview/working-with-objects/finalizers/
- Annotations
  - zur Steuerung von custom verhalten
- Owner Reference
  - dient z.b. zum hierarchischen löschen
  - https://kubernetes.io/docs/concepts/overview/working-with-objects/owners-dependents/

Operator Frameworks best practice

- deklaratives API design (kein imperatives)
- der Anwender gibt den gewünschten Zustand an
- nur ein Operator für eine CRD (ein Owner)
- ein Controller pro Komponente
  - kafka
    - zookeeper
    - kafka

https://sdk.operatorframework.io/docs/best-practices/best-practices/

Demo

https://github.com/rpahli/jugsaxony-operator

Take-aways

- eine API für “alles”
- einfache Erweiterbarkeit
- viele Möglichkeiten über Annotationen
- Zugriffssteuerung über RBAC
- es ist genau definiert was möglich ist und was nicht
- Aggregated APIs zur Erweiterung
    - https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/apiserver-aggregation/
- https://operatorhub.io/
- https://github.com/kokuwaio/k3s-maven-plugin

k8s Erweiterungen

- https://www.telepresence.io/
- https://tekton.dev/
- https://fluxcd.io/
- Service Meshes
  - Zentrale Probleme auslagern

Danke

Fragen
?

ENDE