Helm Chart Erstellung

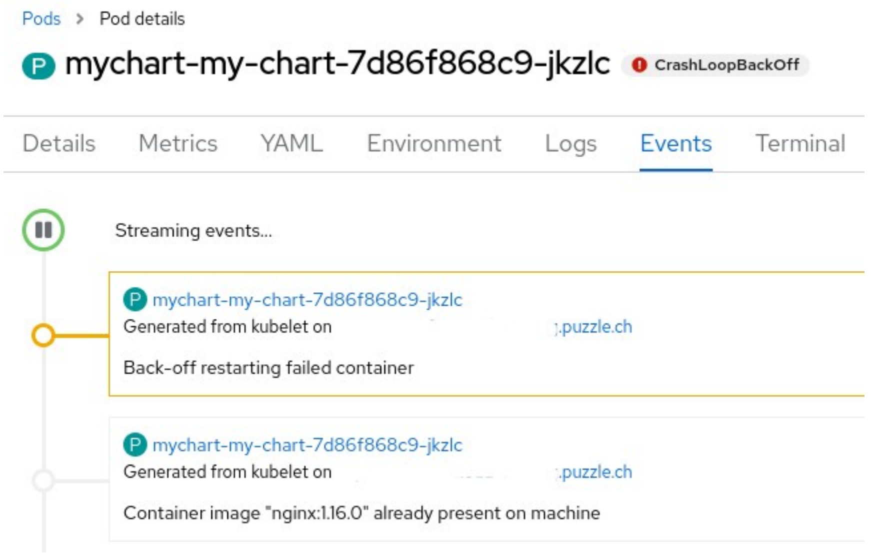
Gebastel oder Entwicklung?

Christoph Raaflaub
Middleware Engineer
raaflaub@puzzle.ch

PUZZLE ITC
changing IT for the better

Probleme mit Helm Charts

```
$ helm template <releaseName> my-chart/

---

# Source: my-chart/templates/serviceaccount.yaml

apiVersion: v1

kind: ServiceAccount

...

   ~/

$ helm install <releaseName> my-chart/

Error: INSTALLATION FAILED: Deployment.apps "mychart1-my-
chart" is invalid:
spec.template.spec.containers[0].imagePullPolicy:
Unsupported value: "Immer": supported values: "Always",
"IfNotPresent", "Never"

✘-1 ~/
```



Pods > Pod details

P mychart-my-chart-7d86f868c9-jkzlc CrashLoopBackOff

Details Metrics YAML Environment Logs Events Terminal

Streaming events...

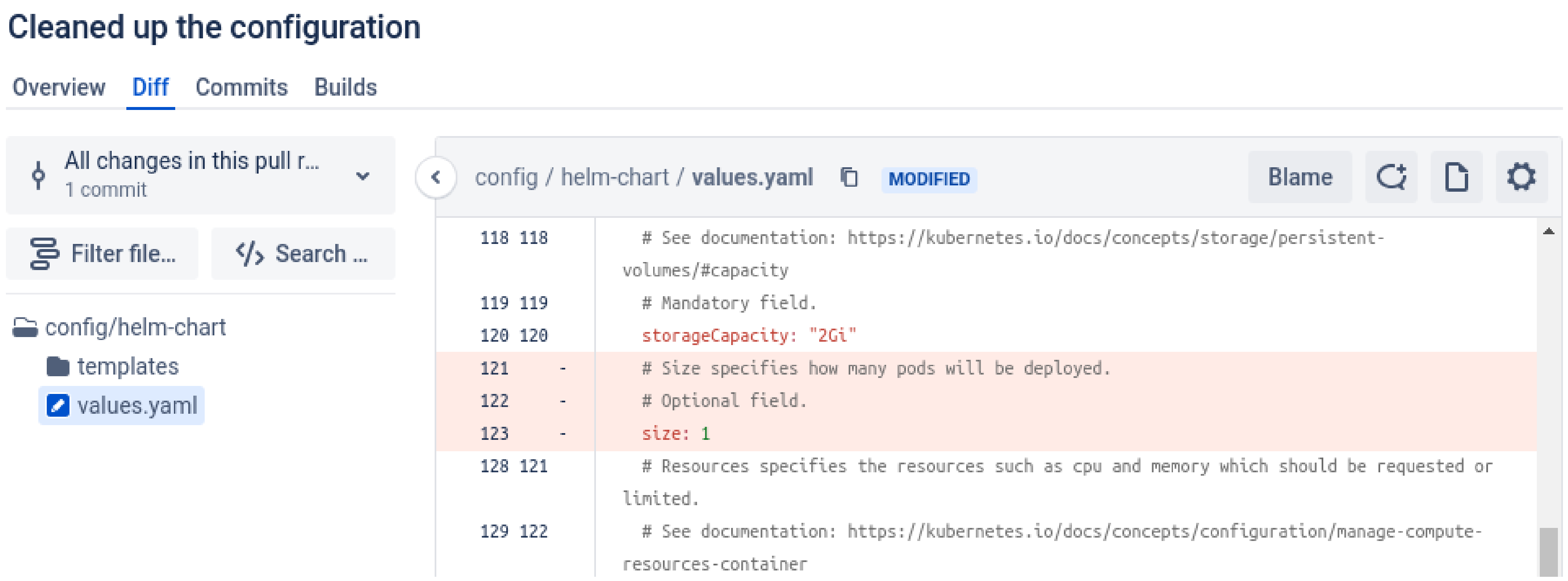
P mychart-my-chart-7d86f868c9-jkzlc
Generated from kubelet on .puzzle.ch

Back-off restarting failed container

P mychart-my-chart-7d86f868c9-jkzlc
Generated from kubelet on .puzzle.ch

Container image "nginx:1.16.0" already present on machine

Pull Request Prüfung


Cleaned up the configuration

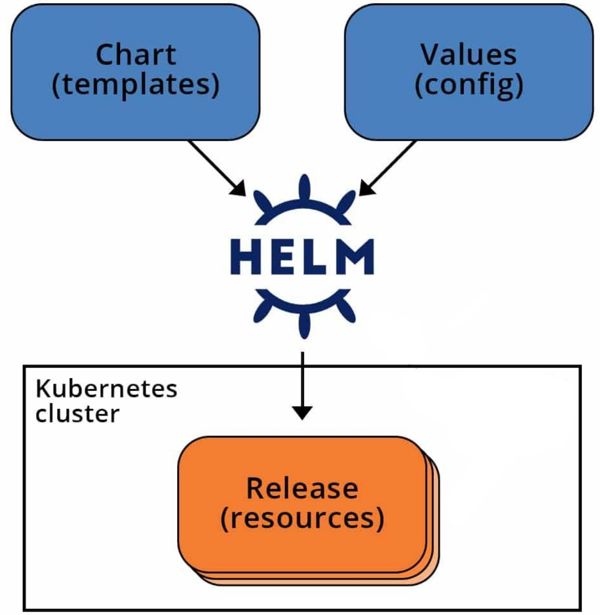
```
118 118     # See documentation: https://kubernetes.io/docs/concepts/storage/persistent-volumes/#capacity
119 119     # Mandatory field.
120 120     storageCapacity: "2Gi"
121      -  # Size specifies how many pods will be deployed.
122      -  # Optional field.
123      -  size: 1
128 121     # Resources specifies the resources such as cpu and memory which should be requested or limited.
129 122     # See documentation: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container
```

Agenda

- Helm / Helm Charts
- Helm Chart Erstellung
- Statische Überprüfungen
- Testen
- Automatisierung
- Schlussfolgerung

Helm

The package manager for Kubernetes



https://helm.sh/

Helm Charts

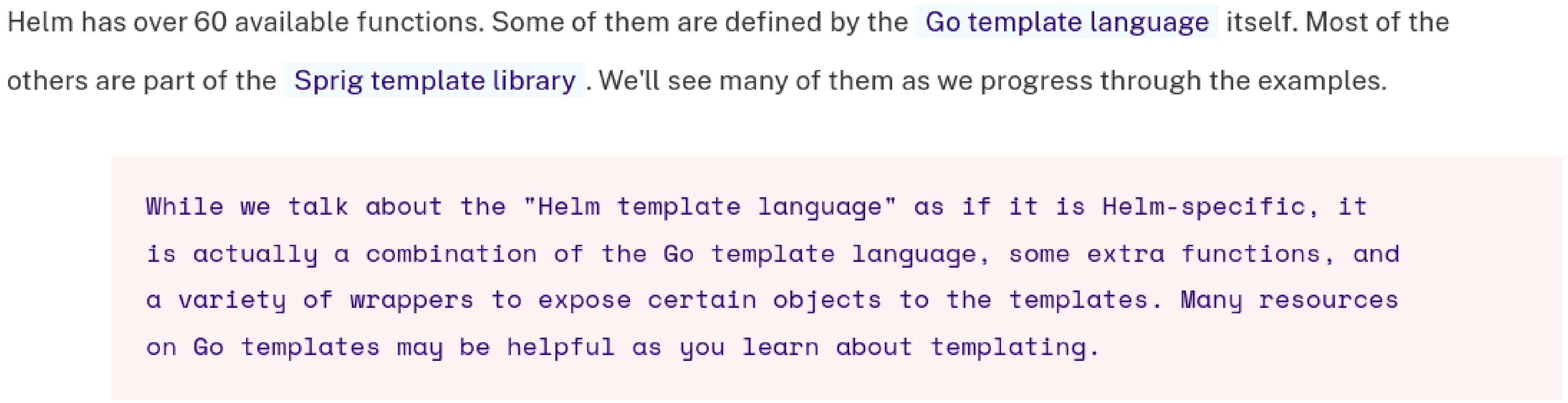
```
wordpress/
  Chart.yaml          # A YAML file containing information about the chart
  LICENSE             # OPTIONAL: A plain text file containing the license for the chart
  README.md           # OPTIONAL: A human-readable README file
  values.yaml         # The default configuration values for this chart
  values.schema.json  # OPTIONAL: A JSON Schema for imposing a structure on the values.yaml file
  charts/             # A directory containing any charts upon which this chart depends.
  crds/               # Custom Resource Definitions
  templates/          # A directory of templates that, when combined with values,
                      # will generate valid Kubernetes manifest files.
  templates/NOTES.txt # OPTIONAL: A plain text file containing short usage notes
```

https://helm.sh/docs/topics/charts/

Template Beispiel

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ include "my-chart.fullname" . }}
data:
  myValue: "Hello {{ .Values.group }}"
  drink: {{ quote .Values.favorite.drink }}
  food: {{ .Values.favorite.food | upper | quote }}
  {{ if eq .Values.favorite.drink "coffee" }}
  mug: "true"
  {{ end }}
softDrink: {{ .Values.favorite.softDrink | default (printf "%s-water" (include "fullname" .)) }}
```

Helm Template Funktionen



Helm has over 60 available functions. Some of them are defined by the Go template language itself. Most of the others are part of the Sprig template library. We'll see many of them as we progress through the examples.

> While we talk about the "Helm template language" as if it is Helm-specific, it is actually a combination of the Go template language, some extra functions, and a variety of wrappers to expose certain objects to the templates. Many resources on Go templates may be helpful as you learn about templating.

Quelle

Helm Chart Erstellung

Kann rasch komplex und unübersichtlich werden!

Helm Chart Erstellung

Wie
entwickle ich
gute
Helm Charts ?

- Statische Überprüfungen
- Lokales Testen
- Unittests schreiben und automatisiert prüfen
- Tests auf weiteren Teststufen hinzufügen
- Pair Programming und oder Code Reviews
- Dokumentation

Statische Überprüfungen

```
$ helm lint my-chart/
==> Linting my-chart/

1 chart(s) linted, 0 chart(s) failed
```

- Yaml Syntax der Templates
- Schema Files für Input Validierung erstellen
  - Struktur
  - Inhalt
  - Werte (Regex)

Eingabewerte Prüfen

```yaml
resources:
  requests:
    cpu: "100m"
    memory: "250Mi"
  limits:
    cpu: "200m"
    memory: "500Mi"
```

values.yaml

```json
{
  "$schema": "http://json-schema.org/draft-07/schema#", "$ref": "#/definitions/Schema",
  "definitions": {
    "ResourcesDefinition": {
      "type": "object",
      "properties": {
        "cpu": {
          "type": "string",
          "pattern": "^[0-9]{1,5}m$"
        },
        "memory": {
          "type": "string",
          "pattern": "^[0-9]{1,5}[M,G]i$"
        }
      },
      "required": [
        "cpu",
        "memory"
      ...
```

values.schema.json

Lokal Testen

```
$ helm template --debug <releaseName> ./my-chart/
---
# Source: my-chart/templates/serviceaccount.yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  name: release42-my-chart
...
```

```
$ helm template <releaseName> ./my-chart/ \
   | kubectl create \
       --dry-run=client \
       --validate \
       -f -
serviceaccount/mychart-my-chart created (dry run)
service/mychart-my-chart created (dry run)
deployment.apps/mychart-my-chart created (dry run)
```

```
$ helm install mychart --dry-run --debug my-chart/
install.go:194: [debug] Original chart version: ""
install.go:211: [debug] CHART PATH: ./my-chart

NAME: mychart
LAST DEPLOYED: Tue May 23 06:57:06 2023
NAMESPACE: my-ns
STATUS: pending-install
REVISION: 1
TEST SUITE: None
USER-SUPPLIED VALUES:
{}

...
```

Helm Unittest Plugin

```yaml
suite: test deployment
templates:
  - deployment.yaml
tests:
  - it: should work
    set:
      image.tag: latest
    asserts:
      - isKind:
          of: Deployment
      - matchRegex:
          path: metadata.name
          pattern: -my-chart$
      - equal:
          path: spec.template.spec.containers[0].image
          value: nginx:latest
```

Links:

- Plugin Dokumentation
- Assert Funktionen

Unittests Schreiben

```yaml
{{- if .Values.administrators }}
apiVersion: v1
kind: Secret
metadata:
  name: {{ .Release.Name }}-administrators
  labels:
    app: {{ .Values.appName }}
stringData:
  {{- range $k, $v := .Values.administrators }}
  {{ upper $k }}_FULLNAME: {{ $v.fullName }}
  {{ upper $k }}_PASSWORD: {{ $v.password }}
  {{ upper $k }}_EMAIL: {{ $v.email }}
  {{- end }}
{{- end }}
```

admin-secret.yaml

```yaml
tests:
  - it: should not exist if not enabled (default)
    asserts:
      - hasDocuments:
          count: 0
  - it: should be created and of kind secret
    set:
      administrators:
        devops:
          fullName: "Dev Ops"
          password: "mySecret"
          email: devops@localhost.com
    asserts:
      - hasDocuments:
          count: 1
      - isKind:
          of: Secret
```

admin-secret_test.yaml

JSON Path

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: myapp
  labels:
    statefulset.kubernetes.io/pod-name: {{ .Values.appName }}
```

```yaml
suite: test configuration
tests:
...
    asserts:
      - equal:
          path: spec.selector["statefulset.kubernetes.io/pod-name"]
          value: "myApp"
```

jsonpath-support

Arrays Testen

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "my-chart.fullname" . }}
spec:
  template:
    spec:
      containers:
        - name: nginx
          image: "nginx:{{ .Values.image.tag }}"
```

```yaml
suite: test deployment
templates:
  - deployment.yaml
tests:
...
      - equal:
          path: spec.template.spec.containers[0].image
          value: nginx:latest
```

```yaml
...
      - equal:
          path: spec.template.spec.containers[?(@.name=='nginx')].image
          value: nginx:latest
```

Listen Testen

```
...
spec:
  configs:
    - name: config1
    {{- if .Values.configA }}
    - name: {{- if .Values.configA }}
    {{- end }}
    {{- if .Values.configB }}
    - name: {{- if .Values.configB }}
    {{- end }}
```

```
suite: test configuration
tests:
...
      - contains:
          path: spec.configs
          content:
            name: config1
          count: 1
          any: true
      - notContains:
          path: spec.configs
          content:
            name: configuration-beta
```

Fehler vermeiden (Flaky-Tests)

```yaml
  - it: should configure object store connection corretly
    set:
      objectStore.enabled: true
      objectStore.connection:
        host: "my-host"
        port: 1234
        region: "eu-central-1"
        credentials: {}
```

```yaml
- it: should configure object store connection corretly
  set:
    objectStore:
      enabled: true
      connection:
        host: "my-host"
        port: 1234
        region: "eu-central-1"
        credentials: {}
```

CI/CD Integration

```
$ helm unittest \
  -v tests/values/testValues.yaml \
  -t "JUnit" \
  -o "helm-test-result.xml" \
  my-chart/
```

```
post {
    always {
        junit allowEmptyResults: true, testResults: "**/helm-test-result.xml"
    }
}
```

Integrationtests

- Helm test
  - In Helm CLI integriert
  - Testet Helm Release
    - (Test Pod wird gestartet)
  - Oft für Smoketests verwendet
- chart-testing CLI
- Automatisierte Integrationstests mit Terratest
  - Blog

GitHub Action

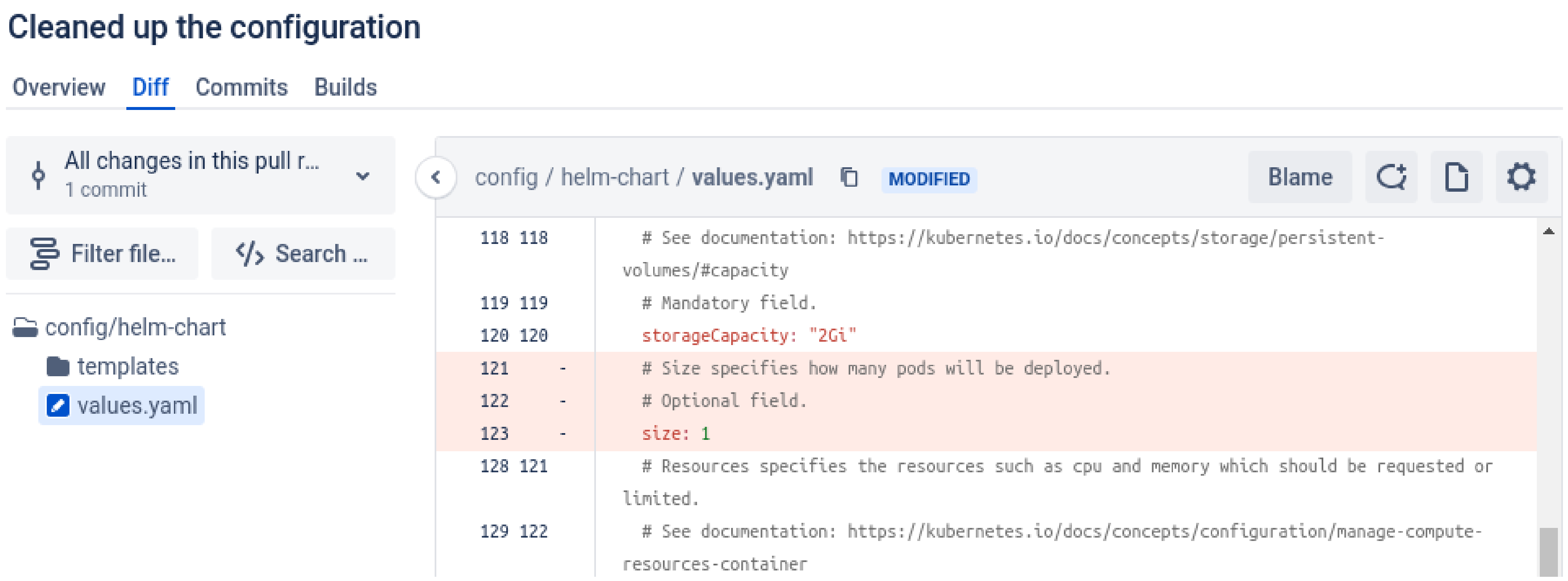
```yaml
name: Lint and Test Charts
on: pull_request
jobs:
...
   - name: Run chart-testing (lint)
     if: steps.list-changed.outputs.changed == 'true'
     run: ct lint —target-branch
       ${{ github.event.repository.default_branch }}
   - name: Create kind cluster
     if: steps.list-changed.outputs.changed == 'true'
     uses: helm/kind-action@v1.4.0
   - name: Run chart-testing (install)
     if: steps.list-changed.outputs.changed == 'true'
     run: ct install —target-branch
```

Security

- Trivy Misconfiguration Scanning
- Helm security and best practices Blog

Schlussfolgerung

Pull Request Prüfung



```
118 118     # See documentation: https://kubernetes.io/docs/concepts/storage/persistent-
            volumes/#capacity
119 119     # Mandatory field.
120 120     storageCapacity: "2Gi"
121     -   # Size specifies how many pods will be deployed.
122     -   # Optional field.
123     -   size: 1
128 121     # Resources specifies the resources such as cpu and memory which should be requested or
            limited.
129 122     # See documentation: https://kubernetes.io/docs/concepts/configuration/manage-compute-
            resources-container
```

We contribute

Fragen ?

Merci!

PUZZLE ITC
changing IT for the better

Christoph Raaflaub
Middleware Engineer
raaflaub@puzzle.ch