Inhaltsverzeichnis

[Allgemein: 2](#_Toc134538602)

[Helm: 2](#_Toc134538603)

[Helm Charts 101 2](#_Toc134538604)

[Commands: 2](#_Toc134538605)

[Chart.yaml example: 2](#_Toc134538606)

[Ingress example: 2](#_Toc134538607)

[Python: 2](#_Toc134538608)

[App starten: 2](#_Toc134538609)

[FastAPI example App: 2](#_Toc134538610)

[Docker: 3](#_Toc134538611)

[Commands: 3](#_Toc134538612)

[Dockerfilebeispiel: 3](#_Toc134538613)

[Argocd: 3](#_Toc134538614)

[Helm Chart: 3](#_Toc134538615)

[Publish: 3](#_Toc134538616)

# Allgemein:

* VPN immer an haben

# Helm:

## Helm Charts 101

Values.yaml:

name: python-redis

deployment.yaml:

name: {{ Values.name }}

## Commands:

helm create <name>

helm upgrade --install <name> ./Ordnername -n <n>

helm uninstall <name>

## Chart.yaml example:

apiVersion: v2

name: python-redis

description: A Helm chart for Kubernetes

type: application

version: 0.1.0

appVersion: "1.0.0" (Die Dockerimage-Version)

## Ingress example:

Chart.yaml:

* appVersion: “1.0.0”

values.yaml:

* image ohne Version
* service.type: ClusterIP (Standard)
* service.port: Variabel
* ingress.enabled: true
* ingress**.**hosts: <ip ohne http://> Bsp.: pokerapp.127.0.0.1.nip.io

deployment.yaml:

* liveness und readinessprobe einstellen und in Pythonapp denken
* liveness/readinessProbe.initialDelaySeconds: ggf. hochschrauben

# Python:

## App starten:

if \_\_name\_\_ == "\_\_main\_\_":

## FastAPI example App:

from fastapi import FastAPI

import uvicorn

app = FastAPI()

@app.get('/health')

def health():

return {'healthy': True}

if \_\_name\_\_ == "\_\_main\_\_":

uvicorn.run(app, host="0.0.0.0", port=8080)

# Docker:

## Commands:

docker login

docker build -t schwadse/<image>:<Version>

docker push schwadse/<image>:<Version>

## Dockerfilebeispiel:

FROM python:3.11

ADD main.py .

RUN pip install poker

CMD [ "python", "./main.py" ]

# Argocd:

## Helm Chart:

values.yaml:

server:

insecure: true

extraArgs:

- --insecure

ingress:

enabled: true

hosts:

- argocd.127.0.0.1.nip.io

configs:

secret:

argocdServerAdminPassword: "$2a$12$KVGWO.6Cb1fyKlKKdLT/7uIPlTdztkP8p9ci.izLPI7SuDlj50FLS"

argocdServerAdminPasswordMtime: "2023-01-01T00:00:00Z"

## Publish:

kubectl port-forward service/myargo-argocd-server -n myargo 8081:443

python-redis:

app deployt

kubectl port-forward service/python-redis -n redis 1909:1909

Healthcheck hat geklappt (http://python-redis.127.0.0.1.nip.io:1909/health)

helm install redis bitnami/redis -n redis

<http://python-redis.127.0.0.1.nip.io:1909/increase_counter>

Internal Server Error

kubectl port-forward --namespace redis svc/my-redis-master 6379:6379 &

REDISCLI\_AUTH="$REDIS\_PASSWORD" redis-cli -h 127.0.0.1 -p 6379

[1] 16047

Could not connect to Redis at 127.0.0.1:6379: Connection refused

not connected> Forwarding from 127.0.0.1:6379 -> 6379

Forwarding from [::1]:6379 -> 6379

todo

CICD: https://docs.github.com/en/actions/learn-github-actions

K8s sidecars?

Kafka

ansible