Technische Hochschule Rosenheim
Technical University of Applied Sciences

MICROSERVICES

RELEASE MANAGEMENT IN KUBERNTES (K8S)

CONTENT

- Considerations
- Artifacts
- Artifact repositories
- Dependencies

CONSIDERATIONS

- How to deploy, update, rollback?
- How to manage artifacts?
- How to manage multiple environments?
  - Develop
  - Staging
  - Production

ARTIFACTS

BEFORE DOCKER

- .jar/.war
- .dll
- .so/.a

DOCKER/K8S

- Container images
- Helm packages
- Forge
- Draft

ARTIFACT REPOSITORIES

BEFORE DOCKER

- Maven Central/jcenter
- NuGet/MyGet
- npm
- ...

DOCKER/K8S

- Docker Hub/Quay image repository
- Helm charts/Quay helm repository

DEPENDENCIES

BEFORE DOCKER

- Maven - POM/BOM
- Gradle - build.gradle
- Node.js - packages.json
- ...

DOCKER/K8S

- Helm's `requirements.yaml`

Technische Hochschule Rosenheim
Technical University of Applied Sciences

HELM INTRODUCTION

Technische Hochschule Rosenheim
Technical University of Applied Sciences

HELM CLI - BASICS

CMD	Explanation
`helm init`	Initialize Helm client and server
`helm create`	Create a new chart
`helm dependency`	Manage charts dependencies
`helm history`	List history of a release
`helm list`	List all releases

Technische Hochschule Rosenheim
Technical University of Applied Sciences

HELM CLI - TEMPLATING & MANAGEMENT

CMD	Explanation
`helm lint`	Lint chart for templating issues
`helm template`	Render chart and print it to STDOUT
`helm install`	Initially deploy a chart
`helm upgrade --install`	Initially deploy or upgrade a chart

DIRECTORY STRUCTURE

```
wordpress/
  Chart.yaml          # A YAML file containing information about
  LICENSE             # OPTIONAL: A plain text file containing th
  README.md           # OPTIONAL: A human-readable README file
  requirements.yaml   # OPTIONAL: A YAML file listing dependencie
  values.yaml         # The default configuration values for this
  charts/             # A directory containing any charts upon wh
  templates/          # A directory of templates that, when combi
                      # will generate valid Kubernetes manifest f
  templates/NOTES.txt # OPTIONAL: A plain text file containing sh
```

Source

Chart.yaml

```yaml
apiVersion: v1
name: hello-chart
version: 1.2.3
kubeVersion: 1.12.0
description: My sample chart description
keywords:
  - sample
home: https://ssample.github.io/chart-sample
sources:
  - https://github.com/ssample/chart-sample
maintainers: # (optional)
  - name: Simon Sample
    email: ssample@gmail.com
    url: https://www.linkedin.com/in/ssample-2349234
engine: gotpl # The name of the template engine (optional, defaul
icon: https://github.com/ssample/chart-sample/draft/blob/master/a
```

requirements.yaml SAMPLE

```yaml
# requirements.yaml
dependencies:
- name: nginx
  version: "1.2.3"
  repository: "https://example.com/charts"
- name: memcached
  version: "3.2.1"
  repository: "https://another.example.com/charts"
```

Source

TEMPLATING - BASICS

- Templating is based on the Golang templating engine
- it ships with a few built-in objects
- it ships with a few built-in pipelines
- it can be extended with custom helper functions

TEMPLATING - BUILT-IN OBJECTS

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hello World"
  drink: {{ .Values.favoriteDrink }}
```

TEMPLATING - PIPELINES

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hello World"
  drink: {{ .Values.favorite.drink | quote }}
  food: {{ .Values.favorite.food | upper | quote }}
```

Technische Hochschule Rosenheim
Technical University of Applied Sciences

TEMPLATING - FLOW CONTROL

Control element	Explanation
`if/else`	used to create conditional blocks
`with`	to override current scope
`range`	loop over values

FLOW CONTROL - SAMPLE

```
{{ if PIPELINE }}
  # Do something
{{ else if OTHER PIPELINE }}
  # Do something else
{{ else }}
  # Default case
{{ end }}
```