Technical University  
of Applied Sciences  
Würzburg-Schweinfurt

Faculty of Computer Science and Business Information Systems

Würzburg, on  
17.05.2023

# Cloud Native Computing

**Exercise: Kubernetes Templating**

**Kustomize**

**-- Recreate the cluster --**

Create a folder name “kustomize” in your exercise folder and create a folder named “base” inside.

**mkdir kustomize**

**cd kustomize**

**mkdir base**

Create a YAML file named "nginx-deployment.yaml" and inside “base” folder that create the deployment using the following configuration.

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**labels:**

**app: nginx-depl**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**app: nginx-depl**

**template:**

**metadata:**

**labels:**

**app: nginx-depl**

**spec:**

**containers:**

**- name: nginx**

**image: nginx**

Create a YAML file named "nginx-svc.yaml" and inside “base” folder that create the service using the following configuration.

**apiVersion: v1**

**kind: Service**

**metadata:**

**labels:**

**app: nginx-depl**

**name: nginx-svc**

**spec:**

**ports:**

**- port: 8080**

**protocol: TCP**

**targetPort: 80**

**selector:**

**app: nginx-depl**

**type: ClusterIP**

You can apply many YAML files with one command by use cd into folder contain YAML files and run this command. Apply every YAML files in the “base” folder

**kubectl apply -f .**

**kubectl apply -f <path-to-YAML-folder>**

You can also delete all instances created by these YAML files with the delete command.

**kubectl delete -f .**

**kubectl delete -f <path-to-YAML-folder>**

*--Pause--*

You can apply all these YAML files using Kustomize. Since Kustomize is Kubernetes native configuration management, you don’t need to install separate tool.

First create kustomization.yaml in **base** folder and add this configuration in.

**resources:**

**- nginx-deployment.yaml**

**- nginx-svc.yaml**

The folder structure should look like this

Ein Bild, das Text, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

To apply YAML files using kustomize, cd to **base** folder and run this command

**kubectl apply -k .**

You can also delete all instances created by kustomization.yaml using this command

**kubectl delete -k .**

If we want to apply another version of YAML files, we need to put the values we want to injected in YAML file in **overlay** folder.

Create an overlay folder under kustomize folder

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Create “nginx-deployment-v2.yaml” YAML file with this configuration. This include the name of deployment and values which will be injected into base configuration.

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**spec:**

**replicas: 10**

Create another kustomization.yaml in overlay folder with this content

**bases:**

**- ../base**

**patches:**

**- nginx-deployment-v2.yaml**

Cd in to **overla**y folder and run this command to apply all YAML files in base folder with values injected from YAML files in overlay folder using kustomize

**kubectl apply -k .**

Delete the instances created by kustomize by running this command

**kubectl delete -k .**

*--Pause--*

**Helm**

**-- Recreate the cluster --**

Since Helm is a separate templating tool, we need to install helm command first. Use this link

**https://helm.sh/docs/intro/install/**

Use the following command only if you still have problem on installation using helm documents above.

curl -LO https://get.helm.sh/helm-v3.8.2-linux-amd64.tar.gz

tar -C /tmp/ -zxvf helm-v3.8.2-linux-amd64.tar.gz

rm helm-v3.8.2-linux-amd64.tar.gz

sudo mv /tmp/linux-amd64/helm /usr/local/bin/helm

sudo chmod +x /usr/local/bin/helm

Check helm command tool

**helm version**

First create a folder for storing helm chart by running this command. The name of folder will be name of the chart. The command will create another example component in the folder.

**helm create my-first-chart**

You will see the files and folder in my-first-chart folder. Explore it.

Ein Bild, das Text, Schrift, Screenshot, Design enthält.

Automatisch generierte Beschreibung

We will create our first chart. First you need to delete everything in template folder and all content in values.yaml

…

Create nginx-deployment.yaml in template folder with this configuration

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**labels:**

**app: nginx**

**tier: frontend**

**spec:**

**# modify replicas according to your case**

**replicas: 3**

**selector:**

**matchLabels:**

**tier: frontend**

**template:**

**metadata:**

**labels:**

**tier: frontend**

**spec:**

**containers:**

**- name: nginx**

**image: nginx:1.16.1**

Create nginx-svc.yaml in template folder with this configuration

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: nginx-svc**

**labels:**

**app: nginx-depl**

**spec:**

**ports:**

**- port: 8080**

**protocol: TCP**

**targetPort: 80**

**selector:**

**app: nginx-depl**

Now in our chart contain 2 YAML files that can be deployed. We can check the end result of YAML files after deploying them using helm by this command

**helm template my-first-chart my-first-chart**

Deploy all YAML files in template folder using helm

**helm install my-first-chart my-first-chart**

See all chart which is currently deployed. You will see the our chart with version 1.

**helm ls**

We can uninstall our chart using this command

**helm uninstall my-first-chart my-first-chart**

*--Pause--*

We will create our chart again, but this time with values from values.yaml injected.

First add a new value of replicas into values.yaml

**deployment:**

**replicas: 10**

Then replace the number of replicas in deployment with this placeholder

**replicas: {{ .Values.deployment.replicas }}**

Check the final version of YAML files again

**helm template my-first-chart my-first-chart**

Run helm install again to install the chart

**helm install my-first-chart my-first-chart**

Check whether all Pods are created

**kubeclt get pod**

You can **try to change number of replicas** to other value in values.yaml and run this command to apply the change

**deployment:**

**replicas: 2**

You can redeploy helm chart using the new values with this command

**helm upgrade my-first-chart my-first-chart**

Check whether all Pods are created

**kubeclt get pod**

Run helm ls again. Now you can see that it shows the second version of your chart

**helm ls**

We can undo our deployment

**helm rollback my-first-chart**

Run helm ls again. Which version can you see ?

**helm ls**

Check how many pods are there

**kubeclt get pod**

We can create second version of chart by create second version of values.yaml inside folder my-first-chart like this

**touch ./my-first-chart/values-v2.yaml**

Add this config into values-v2.yaml

**deployment:**

**replicas: 100**

Use this command to deploy our chart using values-v2.yaml

**helm upgrade my-first-chart my-first-chart --values ./my-first-chart/values-v2.yaml**

Check whether all Pods are created

**kubeclt get pod**

You can injected values on the fly for small change

**helm upgrade my-first-chart my-first-chart --set deployment.replicas=5**

Your exercise ! Instead of injecting replicas, also try to with other properties of YAML files. You can also **add another YAML files** into this too.

*--Pause--*

Now create a new chart with example chart files again using another name and explore those example chart files again. Can you understand more ?

**helm create my-second-chart**

**-- Recreate the cluster --**

You can find many available helm chart for many application online. One of the current famous chart repository is:

**https://artifacthub.io/**

We will try with available helm chart for "Nginx".

**https://artifacthub.io/packages/helm/bitnami/nginx**

First clone nginx chart repository to your machine

**helm repo add bitnami https://charts.bitnami.com/bitnami**

Then just install it with this command

**helm install my-release bitnami/nginx**

This charts will create nginx deployment and LoadBalancer service for you. You can directly access it with node IP in your Browser.

Your exercise ! If Nginx is boring for you, try out with other helm chart by yourself.

Your exercise ! If you still have more time, try to deploy Fluentd, Elasticsearch and Kibana for EFK Stack. We will do it together next lecture.