Technical University  
of Applied Sciences  
Würzburg-Schweinfurt

Faculty of Computer Science and Business Information Systems

Würzburg, on  
04.05.2023

# Cloud Native Computing

**Exercise: Kubernetes**

**Exercise 1: Check Your Understanding (multiple-choice)**

What is not the advantage of Kubernetes?

* ( ) High availability
* ( ) Easy to manage
* ( ) Automation
* ( ) Self-Healing

What is the purpose of cluster context? (2 answers)

* ( ) Define how the cluster should be built
* ( ) Define user credential
* ( ) Define the cluster endpoint
* ( ) Define the capacity of the cluster

Which command is needed to see the IP of the pod? (more than one answer)

* ( ) kubectl get pod
* ( ) kubectl get pod --all-namespaces
* ( ) kubectl describe pod
* ( ) kubectl get pod -o wide

What happens if we delete all pods created by replicaset?

* ( ) We receive a notification from Replicaset
* ( ) Replicaset recreates all pods with the same name as before
* ( ) Replicaset recreates all pods with the different name as before
* ( ) The replicaset deletes itself.

What is not the purpose of namespace?

* ( ) Group the workload
* ( ) Limit resource usage in the namespace
* ( ) Increase security
* ( ) Specify a set of names to call the workload

**Exercise 2: Hands-On: Cluster Context**

**2.1 Basics Cluster Context**

Run the following commands and observe its result to learn more about - cluster context

-- Recreate the cluster --

Display the contents of the kubeconfig.yaml for cluster.

**kubectl config view --flatten**

View the nodes running in the cluster

**kubectl get node**

View information about a specific node in the cluster

**kubectl describe node <node-name>**

Check which cluster you are currently working in

**kubectl config get-contexts**

**kubectl config current-context**

**Exercise 3: Hands-On: Kubernetes Instances**

**3.1 Basics Kubernetes Instances: Pod**

Run the following commands and observe its result to learn more about - Pod

Show pods running in the cluster (in default namespace)

**kubectl get pods**

Show pods running in the cluster (in all namespaces)

**kubectl get pods --all-namespaces**

Create a pod named nginx with an image nginx.

**kubectl run nginx --image nginx**

Redisplay pods running in the cluster now (in default namespace)

**kubectl get pods**

Show the detail of nginx pod

**kubectl describe pods nginx**

View the node running the nginx pod and the IP address of the nginx pod

**kubectl get pod nginx -o wide**

Forward port 80 of the nginx container to port 8080 on your local machine

**kubectl port-forward nginx 8080:80**

Open "localhost:8080" in a browser to see the response from the nginx container.

**localhost:8080**

-- Press Ctrl + C to cancel the command port-forward --

Delete the "nginx" pod

**kubectl delete pod nginx**

**3.2 Basics Kubernetes Instances: YAML-file**

Run the following commands and observe its result to learn more about – YAML-file

**-- Recreate the cluster --**

Create a YAML file named "nginx-pod.yaml" that creates "nginx" pod with nginx:1.17.1 image.

**apiVersion: v1**

**kind: Pod**

**metadata:**

**labels:**

**run: nginx**

**name: nginx**

**spec:**

**containers:**

**- image: nginx:1.17.1**

**name: nginx**

Apply nginx-pod.yaml to start "nginx" pod

**kubectl apply -f nginx-pod.yaml**

Show how YAML file of "nginx" pod will look like after the pod is created

**kubectl get pod nginx -o yaml**

Delete the nginx pod with nginx-pod.yaml

**kubectl delete -f nginx-pod.yaml**

**3.3 Basics Kubernetes Instances: replicaset**

Run the following commands and observe its result to learn more about – replicaset

**-- Recreate the cluster --**

Create a YAML file named "nginx-replicaset.yaml" that create the replicaset with YAML configuration here and apply it.

**apiVersion: apps/v1**

**kind: ReplicaSet**

**metadata:**

**name: nginx-replicaset**

**labels:**

**app: nginx**

**tier: frontend**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**tier: frontend**

**template:**

**metadata:**

**labels:**

**tier: frontend**

**spec:**

**containers:**

**- name: nginx**

**image: nginx**

Apply nginx-replicaset.yaml to start "nginx" replicaset

**kubectl apply -f nginx-replicaset.yaml**

View the replicasets in the cluster

**kubectl get replicaset**

Display the details of an nginx-replicaset replicaset.

**kubectl describe replicaset "nginx-replicaset"**

Delete a pod created by replicaset. What happened?

**kubectl delete pod <pod-name>**

Scale the number of pods in the replicaset to 4

**kubectl scale replicaset nginx-replicaset --replicas 4**

Delete the replicaset "nginx-replicaset"

**kubectl delete replicaset nginx-replicaset**

**3.4 Basics Kubernetes Instances: deployment**

Run the following commands and observe its result to learn more about – deployment

**-- Recreate the cluster --**

Create a YAML file named "nginx-deployment.yaml" that create the deployment with YAML configuration here and apply it.

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: nginx-deployment**

**labels:**

**app: nginx**

**tier: frontend**

**spec:**

**replicas: 3**

**selector:**

**matchLabels:**

**tier: frontend**

**template:**

**metadata:**

**labels:**

**tier: frontend**

**spec:**

**containers:**

**- name: nginx**

**image: nginx**

Apply nginx-deployment.yaml to start "nginx" deployment

**kubectl apply -f nginx-deployment.yaml**

Show the list of the created deployment

**kubectl get deployment**

Show the details of deployments "nginx-deployment".

**kubectl describe deployment nginx-deployment**

Display the list of replicasets in the cluster. You will see a replicaset created by deployment "nginx-deployment".

**kubectl get replicaset**

Show all pods with the label "tier=frontend". What are the pods called? How many pods does the "tier=frontend" label have?

**kubectl get pod -l tier=frontend**

Scale the pods for deployment "nginx-deployment" to 100

**kubectl scale deployment nginx-deployment --replicas 100**

Delete "nginx-deployment" deployment

**kubectl delete deployment nginx-deployment**

**3.5 Basics Kubernetes Instances: configmap**

Run the following commands and observe its result to learn more about – configmap

**-- Recreate the cluster --**

Create a YAML file named "nginx-configmap.yaml" that create the configmap with YAML configuration here and apply it.

**apiVersion: v1**

**kind: ConfigMap**

**metadata:**

**name: nginx-config**

**namespace: default**

**data:**

**index.html: |**

**"Welcome to Course: Kubernetes !"**

Apply nginx-configmap.yaml to start "nginx" deployment

**kubectl apply -f nginx-configmap.yaml**

Show all configmaps in the cluster

**kubectl get configmap**

Display the value in the configmap we created

**kubectl describe configmap nginx-config**

**kubectl get configmap nginx-config -o yaml**

Create a pod using our newly created configmap with this YAML configuration. Save it under the file "nginx-pod-with-configmap.yaml" and apply it.

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: nginx-pod**

**spec:**

**containers:**

**- name: nginx-container**

**image: nginx**

**volumeMounts:**

**- name: nginx-page**

**mountPath: /usr/share/nginx/html/**

**volumes:**

**- name: nginx-page**

**configMap:**

**name: nginx-config**

**restartPolicy: Never**

Apply nginx-pod-with-configmap.yaml to start "nginx" deployment

**kubectl apply -f nginx-pod-with-configmap.yaml**

Check if the Nginx pod is already started

**kubectl get pod nginx-pod**

Run "kubectl port-forward" to map port 80 of "nginx-pod" on our localhost at port 8080. Then use the URL "localhost:8080" in your web browser to see the response from nginx

**kubectl port-forward nginx-pod 8080:80**

**127.0.0.1:8080**

**3.6 Basics Kubernetes Instances: service**

Run the following commands and observe its result to learn more about – service

**-- Recreate the cluster --**

Create a deployment with nginx:alpine image named nginx-depl with 3 pods

**kubectl create deployment nginx-depl --image nginx:alpine --replicas 3**

Generate a ClusterIP service for deployment named "nginx-svc" that has an IP address of 8080 and is mapped to port 80 creating a YAML file named "nginx-service.yaml" with YAML configuration here and apply it.

**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**

Apply nginx-service.yaml to start "nginx" service

**kubectl apply -f nginx-service.yaml**

View all services in the cluster

**kubectl get service**

Display a detail of the nginx-svc service

**kubectl describe service nginx-svc**

Run the kubectl exec command in the nginx-depl deployment pod and run the following curl command to see the output of Nginx (names from the pod). Now we show the response from the DNS and IP address of the service

**kubectl exec -it <pod-name> -- sh**

**curl nginx-svc:8080**

**curl <Service-IP>:8080**

Delete the nginx-svc service

**kubectl delete service nginx-svc**