

# Aufgabe 2: Praktische Aufgabe

## 1. Speicherung des Deployments als yaml Datei

- Wir haben im Praxis Teil des Kurses kubectl create genutzt um das Deployment zu erstellen.
- Jetzt wollen wir dieses aber nachträglich ändern und benötigen die YAML Konfigurationsdatei.
- Finde einen Weg die aktuelle Version des Deployments bei dir als Yaml Datei abzuspeichern.

(Tipp): Schaue dir die Doku für kubectl get an.

## 2. Aktualisieren des nginx-Deployments zur Nutzung der ConfigMap

- Öffne die YAML-Datei für das nginx-Deployment, die du in der ersten Aufgabe erstellt hast.
- Füge die ConfigMap als Volume zum Deployment hinzu und mounte sie im nginx-Container.
- Folge hierfür folgendem Schema oder schaue dir hier die Doku an  
<https://kubernetes.io/docs/tasks/configure-pod-container/configure-pod-configmap/#add-configmap-data-to-a->

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx
          volumeMounts:
            - name: config-volume
              mountPath: /usr/share/nginx/html
      volumes:
        - name: config-volume
          configMap:
            name: nginx-index
```

- Wende die Änderungen an mit  
kubectl apply -f DEIN\_DEPLOYMENT\_YAML\_DATEINAME.yaml.

## 2. Überprüfen des Deployments und des Mounts

- Stelle sicher, dass das Deployment aktualisiert wurde:  
kubectl rollout status deployment/nginx-deployment.
- Überprüfe, ob die Pods neu gestartet wurden: kubectl get pods.

### 3. Zugriff auf den aktualisierten nginx-Webserver

- Stelle eine Verbindung zu einem der Pods des Deployment her, um auf die Startseite zuzugreifen.
- Nutze hierfür entweder k9s oder kubectl um ein Port-Forwarding einzurichten.
- Du solltest nun die benutzerdefinierte Startseite aus der ConfigMap sehen.