Wie man Kubernetes verwendet

Kubernetes (oft als K8s abgekürzt) ist eine Open-Source-Plattform zur Automatisierung des Deployments, der Skalierung und des Betriebs von containerisierten Anwendungen. Hier ist eine Schritt-für-Schritt-Anleitung zur effektiven Nutzung von Kubernetes.

---

1. Einrichten eines Kubernetes-Clusters

Bevor Sie Anwendungen deployen können, benötigen Sie ein Kubernetes-Cluster –eine Gruppe von Maschinen (Knoten), die Ihre containerisierten Workloads ausführen und von einer Steuerungsebene verwaltet werden.

- **Für die lokale Entwicklung:**
  - Verwenden Sie Minikube oder Docker Desktop, um ein Single-Node-Cluster auf Ihrem lokalen Rechner einzurichten.
  - Beispiel mit Minikube:

    ```
    minikube start
    ```

- **Für die Produktion:**
  - Verwenden Sie verwaltete Dienste wie Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS) oder Azure Kubernetes Service (AKS).
  - Alternativ können Sie ein Cluster manuell mit Kubeadm einrichten.
  - Beispiel mit einem verwalteten Dienst (z.B. GKE):

    ```
    gcloud container clusters create my-cluster
    ```

---

2. Erstellen eines Docker-Images Ihrer Anwendung

Kubernetes verwaltet containerisierte Anwendungen, typischerweise mit Docker-Containern.

- Schreiben Sie eine `Dockerfile`, um die Umgebung Ihrer Anwendung zu definieren. Beispiel:

```
FROM node:16
WORKDIR /app
COPY . .
RUN npm install
CMD ["npm", "start"]
```

- Bauen Sie das Docker-Image:

  ```
  docker build -t your-image-name:latest .
  ```

- Schieben Sie das Image in ein Container-Registry (z.B. Docker Hub):

  ```
  docker push your-image-name:latest
  ```

---

3. Definieren von Kubernetes-Objekten

Kubernetes verwendet YAML-Dateien, um Ressourcen wie Pods, Services und Deployments zu definieren.

- **Pod:** Die kleinste deploybare Einheit, die einen oder mehrere Container enthält.
- **Service:** Stellt Ihre Anwendung dem Netzwerk zur Verfügung.
- **Deployment:** Verwalten von Pods, stellt sicher, dass die gewünschte Anzahl läuft und aktualisiert.

Beispiel einer `Deployment` YAML-Datei (`my-app-deployment.yaml`):

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3  # Anzahl der Pod-Instanzen
  selector:
    matchLabels:
      app: my-app
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-container
        image: your-image-name:latest
```

---

4. Deployen der Anwendung

Verwenden Sie das `kubectl`-Befehlszeilen-Tool, um mit Ihrem Cluster zu interagieren und Ihre Anwendung zu deployen.

- Wenden Sie die YAML-Datei auf das Cluster an:

```
kubectl apply -f my-app-deployment.yaml
```

- Überprüfen Sie das Deployment:

```
kubectl get deployments
kubectl get pods
```

---

5. Verwalten der Anwendung

`kubectl` bietet Befehle zur Überwachung und Verwaltung Ihrer Anwendung:

- **Skalieren der Anwendung:**

```
kubectl scale deployment my-app --replicas=5
```

- **Überprüfen des Pod-Status:**

```
kubectl get pods
```

- **Protokolle anzeigen:**

```
kubectl logs <pod-name>
```

- **Auf einen Container zugreifen:**

```
kubectl exec -it <pod-name> -- /bin/bash
```

Wichtige Konzepte zur weiteren Erforschung

- **Namespaces:** Organisieren von Ressourcen und Verwaltung des Zugriffs (z.B. `kubectl create namespace my-namespace`).
- **ConfigMaps:** Speichern von Konfigurationsdaten getrennt von der Anwendung.
- **Secrets:** Verwaltung sensibler Informationen wie Passwörter.
- **Ingress:** Exponieren von Diensten extern mit einem einzigen Einstiegspunkt.

Tipps zum Einstieg

Kubernetes ist ein leistungsfähiges, aber komplexes System. Beginnen Sie mit diesen Grundlagen –Einrichten eines Clusters, Deployen einer einfachen Anwendung und Verwenden von `kubectl` –und erkunden Sie dann bei Bedarf fortgeschrittene Funktionen. Für praktische Übungen probieren Sie Tutorials wie die Kubernetes Basics auf der offiziellen Website aus.

Mit dieser Grundlage können Sie Kubernetes effektiv nutzen, um containerisierte Anwendungen zu deployen und zu verwalten!