Level-2-Kubernetes

1. Kubernetes Shared Volumes

Ein Bild, das Text, Screenshot, Software enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Seklinde bir yaml dosyasi olusturyoruz. Dosya file adi

#### 🡺 volume-share-devops.yaml

Seklinde olacak.

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

Automatisch generierte Beschreibung

Daha sonra sirasi ile

🡺 kubectl apply -f volume-share-devops.yaml

🡺 kubectl exec -it volume-share-devops -c volume-container-devops-1 -- /bin/bash

🡺daha sonra /tmp/media/media.txt olusturuyoruz. İcerisine 🡺This is a test file yaziyoruz

🡺 kubectl exec -it volume-share-devops -c volume-container-devops-2 -- /bin/bash

🡺 cat /tmp/demo/media.txt

🡺exit komutlari ile kontrol ediyoruz.

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

Automatisch generierte Beschreibung2-Kubernetes Sidecar Containers

Asagidaki komutlar ile bir yml file olusturup .

🡺 kubectl apply -f webserver.yaml komutunu calistiriyoruz.

Bu YAML dosyası, webserver adında bir Pod oluşturur ve içinde iki farklı konteyner barındırır: nginx-container ve sidecar-container. shared-logs adında bir boş bir emptyDir tipinde bir volume tanımlanmıştır ve bu volume her iki konteynerde de /var/log/nginx konumuna bağlanmıştır.

apiVersion: v1

kind: Pod

metadata:

name: webserver

spec:

volumes:

- name: shared-logs

emptyDir: {}

containers:

- name: nginx-container

image: nginx:latest

volumeMounts:

- name: shared-logs

mountPath: /var/log/nginx

- name: sidecar-container

image: ubuntu:latest

command: ["sh", "-c", "while true; do cat /var/log/nginx/access.log /var/log/nginx/error.log; sleep 30; done"]

volumeMounts:

- name: shared-logs

mountPath: /var/log/nginx

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

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung3- Deploy Nginx Web Server on Kubernetes Cluster

Seklinde bir tane nginx-deployment.yaml

File olusturuyoruz ve

🡺 kubectl apply -f nginx-deployment.yaml calistiriyoruz.

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

Automatisch generierte BeschreibungArkasindan nginx-service.yaml file olusturup

🡺 kubectl apply -f nginx-service.yaml komutunu calistiriyoruz

4- Print Environment Variables

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung Ein Bild, das Text, Screenshot, Software, Schrift enthält.

Automatisch generierte BeschreibungSeklinde bir yaml file olusturup. Yaml file ismi önemli

🡺kubectl apply -f file adin ardindan 🡺kubectl get pods iel pod calisiyor mu kontrol

🡺kubectl logs pod adi ile de ciktiyi aliyorsunuz.

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

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung5- Rolling Updates And Rolling Back Deployments in Kubernetes

Seklinde yaml file olustur. İlkinde hata aldim. Bu farkli degerleri degistir.

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

Automatisch generierte Beschreibung🡺 kubectl apply -f httpd-deployment.yaml komut ile calistir

Seklinde yaml file olustur. 🡺 kubectl apply -f httpd-service.yaml komutu ile valistir.

İlk yaml file da image versiyonunu degistir.

🡺 kubectl apply -f httpd-deployment.yaml komutunu calistir.

En son eski surume dönmek icin

🡺 kubectl rollout undo deployment httpd-deploy -n datacenter komutunu calistir.

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

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung6- Deploy Jenkins on Kubernetes

Seklinde bir tane

jenkins-namespace.yaml

file olusturuyoruz.

Ein Bild, das Text, Elektronik, Screenshot, Software enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Elektronik, Screenshot, Multimedia enthält.

Automatisch generierte Beschreibung

Seklinde jenkins-service.yaml file olusturuyoruz.

Daha sonra sagdaki gibi jenkins-deployment.yaml file olusturuyoruz.

kubectl apply -f jenkins-namespace.yaml

kubectl apply -f jenkins-service.yaml

kubectl apply -f jenkins-deployment.yaml

komutlari ile file lari apply yapiyoruz. Artik jenkins serverine ulasabiliriz.