

Microsserviço Spring Boot no Kubernetes com Balanceamento de Carga

Este é um exemplo de microsserviço construído com Spring Boot e implantado em um cluster Kubernetes local usando Minikube. O serviço está configurado com balanceamento de carga para distribuir as requisições entre várias réplicas, demonstrando uma arquitetura escalável e resiliente.

Pré-requisitos

- [Docker](#)
- [Kubectcl](#)
- [Minikube](#)

Passos para Implantação

1. Configurar o Minikube para Usar o Docker Local

1. **Iniciar o Minikube:** Se o Minikube ainda não estiver iniciado, execute:

```
minikube start
```

2. **Configurar o Docker Local do Minikube:** Para usar a imagem Docker local no Minikube:

```
eval $(minikube docker-env)
```

2. Preparar a Imagem Docker do Microsserviço

1. **Criar o Dockerfile:** Certifique-se de que há um [Dockerfile](#) na raiz do projeto Spring Boot.

Exemplo de [Dockerfile](#):

```
FROM openjdk:17-jdk-alpine
COPY target/microsservico.jar microsservico.jar
ENTRYPOINT ["java", "-jar", "/microsservico.jar"]
```

2. **Construir a Imagem Docker:**

No diretório raiz do projeto, execute:

```
docker build -t microsservico:v1 .
```

3. Verificar a Imagem no Docker do Minikube:

```
docker images
```

Certifique-se de que a imagem `microservico:v1` está listada.

3. Criar os Arquivos de Manifesto Kubernetes

1. Criar o `deployment.yaml` no diretório `kubernetes/`:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: microservico-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: microservico
  template:
    metadata:
      labels:
        app: microservico
    spec:
      containers:
        - name: microservico
          image: microservico:v1
          ports:
            - containerPort: 8080
```

2. Criar o `service.yaml` no diretório `kubernetes/`:

```
apiVersion: v1
kind: Service
metadata:
  name: microservico-service
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 8080
  selector:
    app: microservico
```

4. Aplicar os Arquivos de Manifesto no Cluster Kubernetes

1. Criar o Deployment e o Service:

```
kubectl apply -f kubernetes/deployment.yaml  
kubectl apply -f kubernetes/service.yaml
```

2. Verificar os Pods:

```
kubectl get pods
```

Certifique-se de que todos os pods estão em **Running**.

5. Acessar o Microserviço Externamente

1. Acessar o Microserviço:

Use o comando abaixo para acessar o serviço:

```
minikube service microservico-service
```

6. Parar ou Excluir o Minikube (Quando Não for Mais Necessário)

1. Parar o Minikube:

```
minikube stop
```

2. Excluir o Minikube (para liberar recursos):

```
minikube delete
```

3. Retornar para o Docker local:

```
eval $(minikube docker-env -u)
```

Observações

- Caso você faça mudanças na aplicação, lembre-se de reconstruir a imagem Docker com `docker build -t microservico:v1 .` e aplicar os manifestos novamente.
- Certifique-se de que o ambiente Docker está corretamente configurado para o Minikube antes de criar e aplicar os manifestos.

Problemas Comuns

- **Erro de `ImagePullBackOff`:** Certifique-se de que a imagem está corretamente construída e disponível no ambiente Docker do Minikube.
- **Problemas de Acesso ao Serviço:** Verifique se o IP do Minikube e a porta exposta estão corretos.

Contribuições

Contribuições e melhorias são bem-vindas! Sinta-se à vontade para abrir issues ou pull requests.

Este projeto é apenas para fins educacionais e de demonstração. Ajustes podem ser necessários dependendo do seu ambiente e requisitos específicos. Se precisar de mais ajustes ou informações, é só avisar!