Arquitetura-de-microsserviços

Aqui está o passo a passo completo para configurar e executar o projeto de exemplo usando Python, Kubernetes e PostgreSQL:

Projeto de Exemplo - Microsserviços com Kubernetes e PostgreSQL

Este projeto tem como objetivo demonstrar como criar um aplicativo usando a arquitetura de microsserviços com Kubernetes e banco de dados PostgreSQL.

Pré-requisitos

Antes de iniciar, verifique se você tem os seguintes pré-requisitos:

- Python 3.6 ou superior instalado em sua máquina local
- Docker instalado em sua máquina local
- Minikube instalado em sua máquina local
- kubectl instalado em sua máquina local
- Um ambiente de desenvolvimento Python configurado em sua máquina local (opcional)

1. Criando o banco de dados PostgreSQL

Antes de prosseguir, vamos criar um cluster PostgreSQL usando o Docker. Abra um terminal e execute o seguinte comando:

```
docker run --name postgres -e POSTGRES_PASSWORD=mysecretpassword -d postgres
```

Esse comando cria um contêiner PostgreSQL com o nome "postgres" e define uma senha para o usuário "postgres".

2. Criando a tabela de dados

Agora, crie uma tabela "pessoas" no banco de dados PostgreSQL usando o seguinte comando:

```
docker exec -it postgres psql -U postgres -c "CREATE TABLE pessoas (nome text, idade integ
er);"
```

Esse comando cria uma tabela "pessoas" no banco de dados PostgreSQL.

3. Criando o primeiro microsserviço

Agora, crie um diretório para o primeiro microsserviço usando o seguinte comando:

```
mkdir inserir_dados && cd inserir_dados
```

Dentro do diretório, crie um arquivo chamado `Dockerfile` com o seguinte conteúdo:

```
FROM python:3.8-slim-buster

RUN mkdir /app
WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["python", "app.py"]
```

Este Dockerfile define a imagem base como Python 3.8, instala as dependências do Python e define o comando de inicialização do aplicativo.

Crie um arquivo `requirements.txt` com o seguinte conteúdo:

```
flask==2.0.1
psycopg2-binary==2.9.1
```

Este arquivo lista as dependências do Python para o microsserviço.

Crie um arquivo `app.py` com o seguinte conteúdo:

```
from flask import Flask, request
import psycopg2

app = Flask(__name__)

@app.route('/inserir', methods=['POST'])
def inserir():
    conn = psycopg2.connect(
        dbname="postgres",
        user="postgres",
        password="mysecretpassword",
        host="localhost"
    )
    cursor = conn.cursor()
    cursor.execute(
        "INSERT INTO pessoas (nome, idade) VALUES (%s, %s)",
        (request.form['nome'], request.form['idade'])
    )
    conn.commit()
    cursor.close()
    conn.close()
    return "Dados inseridos com sucesso!"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Este arquivo contém o código do microsserviço responsável por inserir os dados no banco.