

Universidade Federal de São Carlos
Departamento de Computação

Projeto com Docker e Kubernetes

Integrantes

Lucas Abbiati, 801572

São Carlos, 16 de julho de 2024

1. Aplicação

1.1. Frontend

O frontend da aplicação foi desenvolvido utilizando JavaScript react para fazer uma interface simples de um sistema onde se cadastra livros.

1.2. Backend

O backend foi feito em Python, utilizando FastAPI e Uvicorn. No backend também está presente o banco de dados que, no caso, é um arquivo .json (o banco de dados é mockado pois a finalidade é fazer o estudo do Docker e do Kubernetes e não da aplicação em si)

2. Componentes Docker e Kubernetes

2.1. Docker

2.1.1. Aplicação

A aplicação conta com um arquivo (docker-compose.yml) para ser executada de maneira simples, além de um script (docker-up.sh) para facilitar sua execução e um para parar sua execução (docker-down.sh)

2.1.2. Backend

O backend conta com um arquivo Dockerfile, onde são executados comando para criar uma imagem do backend dentro de um container docker, instalando suas dependências, arquivos necessários e rodando o código main.py

2.1.3. Frontend

Igualmente, o frontend contém um arquivo Dockerfile que é utilizado para criar a imagem dentro do container docker, instalar suas dependências, executar a aplicação frontend e instanciar o nginx .

2.2. Kubernetes

2.2.1. Aplicação

A aplicação contém uma pasta chamada k8s, onde estão os arquivos de deployment do backend e do frontend, os arquivos de service do frontend e do backend e o arquivo para criar um ingress para a aplicação, além de dois scripts, um para iniciar a aplicação(minikube-up.sh) e outro para

encerrar a aplicação (minikube-down.sh),

2.2.2. Backend

O possui um arquivo chamado backend-deployment.yaml que é utilizado para criar um pod do backend no minikube utilizando uma imagem que está disponível no dockerhub (<https://hub.docker.com/r/lucasabbi/backend>).

Além disso, também possui um arquivo chamado backend-service.yaml para que, após o deployment ser criado, possa ser criado o service do backend.

2.2.3. Frontend

Igualmente, o frontend possui um arquivo chamado frontend-deployment.yaml que é utilizado para criar um pod do frontend no minikube utilizando uma imagem que está disponível no dockerhub

(<https://hub.docker.com/r/lucasabbi/frontend>).

Além disso, também possui um arquivo chamado frontend-service.yaml para que, após o deployment ser criado, possa ser criado o service do frontend.

3. Execução

3.1. Com o Docker

Em um repositório a sua escolha, utilize o comando git clone <https://github.com/LucasAbbiati/DevOps.git> para clonar o repositório e, na raiz do projeto, execute o comando ./docker-up.sh e, depois acesse <http://localhost:80> para ter acesso a aplicação e, para encerrar, utilize o comando ./docker-down.sh

3.2. Com o Kubernetes

Em um repositório a sua escolha, utilize o comando git clone <https://github.com/LucasAbbiati/DevOps.git> para clonar o repositório e, na raiz do projeto, execute o comando ./minikube-up.sh e, depois utilize o comando *minikube service frontend* para ter acesso a aplicação, caso deseje acessar pelo ingress (k8s.local), no arquivo /etc/hosts, insira a linha 192.168.49.2 k8s.local, salve e após isso, acesso o link. Para encerrar, utilize o comando ./minikube-down.sh encerrar a aplicação.