

**Universidade Federal de São Carlos**

**DevOps Trabalho 2 - Prática com Kubernetes**

**Catálogo de Filmes**

**Disponível em:**

**<https://github.com/KY-U/trabalho-devops-kubernetes>**

**Integrantes: Caio Ueda Sampaio 802215**

- **Aplicação:**

- **Descrição geral:**

- A aplicação se trata de um simples sistema de catálogo de filmes, em que é possível tanto adicionar quanto remover filmes do catálogo. Cada serviço possui seu próprio container.

- **Database:**

- O sistema utiliza um banco mysql para guardar o id, título, ano de lançamento e descrição do filme. O script de inicialização e povoamento inicial do b.d. encontra-se em /database.

- **Backend:**

- O backend é implementado como um app flask, que realiza as queries mysql de listagem, adição ou remoção de filmes baseado na requisição http recebida.
    - As requisições seguem o padrão R.E.S.T.. Para listar os filmes, recebe GET em /movies, para adicionar um filme, recebe POST em /movies com um body contendo as informações do filme, e para deletar um filme, recebe DELETE em /movies/<id do filme>

- **FrontEnd:**

- O frontend se trata de uma simples interface implementada com HTML, CSS e JS, que realiza as requisições baseado no input do usuário.

- **Proxy Reverso:**

- Apenas o endereço "k8s.local" fica exposto ao usuário. Todas as requisições em "/" são redirecionadas para o FrontEnd. Já as requisições em "/api" são redirecionadas para o BackEnd. Esse redirecionamento é realizado pelo Ingress.

- **Artefatos Kubernetes:**

- **Descrição Geral:**

- Todos os artefatos .yaml estão presentes em /k8
    - A automatização do build, deploy e undeploy é fornecida pelos scripts .sh

- **DataBase:**

- O banco de dados é o único que apresenta 3 artefatos .yaml, pois além do documento de deployment do pod, "mysql-deployment.yaml", e serviço, "mysql-service.yaml", existe também um arquivo para o povoamento do banco de dados, "mysql-init-config.yaml"
    - A imagem mysql utilizada é a imagem mysql oficial disponível no dockerhub.

- **BackEnd:**

- O backend apresenta um documento de deployment, "backend-deployment.yaml", e de serviço, "backend-service.yaml".
    - A imagem utilizada está disponível no repositório caiou/backend

- **Frontend:**

- O frontend apresenta um documento de deployment, "frontend-deployment.yaml", e de serviço, "frontend-service.yaml".
    - A imagem utilizada está disponível no repositório caiou/frontend

- **Ingress:**

- O Ingress é configurado com o arquivo "ingress.yaml", disponível por definir o host "k8s.local" e o redirecionamento das requisições.

- **Rodando o projeto:**

- **Inicialização:**

- Como as imagens já foram previamente carregadas no dockerhub, basta executar o script "**minikube-up**" com o docker daemon rodando e a aplicação minikube instalada.

- Para fins educacionais, o script de build e exportação das imagens está disponível em “**build.sh**”.
- Como o script configura a utilização do docker daemon do minikube, não é necessário realizar a exportação das imagens para o minikube manualmente com “minikube image load <image>”

- **Encerramento:**

- Para encerrar o serviço, basta executar o script “minikube-down”, que, por boas práticas, encerra os artefatos kubernetes na ordem inversa à que foram criados.