Documentação da Aplicação e Artefatos Kubernetes

1. Descrição Geral da Aplicação

A aplicação é composta por dois principais componentes: um backend que serve como API para o processamento de dados e um frontend que fornece a interface com o usuário. Ambos os componentes são empacotados como imagens Docker e implantados em um cluster Kubernetes utilizando Minikube.

2. Componentes da Aplicação

2.1 Backend

O backend é responsável por fornecer uma API que gerencia as regras de negócios da aplicação. Desenvolvido usando Node.js (ou outra linguagem de sua escolha), ele contém as principais lógicas de autenticação, manipulação de dados e resposta às requisições feitas pelo frontend.

O backend é empacotado em uma imagem Docker, que inclui o código da aplicação, suas dependências e todas as configurações necessárias para sua execução. Essa imagem é armazenada localmente no Docker do Minikube, facilitando a orquestração pelo Kubernetes.

O backend escuta na porta 3000 dentro do container e, por meio de um artefato Kubernetes chamado Service, é acessado pelos outros componentes da aplicação no cluster.

A imagem gerada pelo backend tem o nome de devops/back

2.2 Frontend

O frontend é a interface da aplicação, responsável por permitir que os usuários interajam com os dados processados pelo backend. Desenvolvido usando React (ou outra tecnologia frontend), o frontend consome as APIs fornecidas pelo backend para exibir e manipular as informações no navegador.

Semelhante ao backend, o frontend é empacotado em uma imagem Docker, que inclui o código da interface, suas dependências e as configurações necessárias. O frontend é servido na porta 80, o que é padrão para aplicações web.

Os arquivos estáticos são construídos e servidos pelo container, garantindo que a aplicação seja facilmente acessível pelo navegador. A comunicação entre o frontend e o backend é intermediada pelo Kubernetes, que facilita o roteamento das requisições.

A imagem gerada pelo frontend é chamada de devops/front

3. Artefatos Kubernetes Utilizados

3.1 Deployment

Um Deployment é um artefato Kubernetes utilizado para gerenciar e manter a execução dos containers no cluster. Ele garante que uma quantidade especificada de réplicas dos containers esteja sempre rodando. Se um container falhar ou for removido, o Deployment se encarrega de criar um novo.

Na aplicação, há dois Deployments: um para o backend e outro para o frontend. Cada Deployment define:

- A imagem Docker a ser utilizada (no caso, as imagens do backend e frontend).
- A quantidade de réplicas que devem ser executadas.
- Configurações para garantir que os containers sejam corretamente monitorados e reiniciados em caso de falhas.

3.2 Service

O Service é um artefato Kubernetes que expõe os containers dentro do cluster, permitindo que outros serviços ou pods comuniquem-se com eles. No caso da minha aplicação, existem dois Services:

- O Service do backend, que expõe a API do backend na porta 3000.
- O Service do frontend, que expõe a interface do frontend na porta 80.

Esses Services permitem que as requisições feitas pelo frontend sejam roteadas corretamente para o backend e também permitem que o backend seja acessado dentro do cluster.

3.3 Ingress

O Ingress é um artefato Kubernetes que gerencia o acesso externo à aplicação dentro do cluster. Ele oferece uma forma de expor múltiplos serviços Kubernetes por meio de um único endpoint, definindo regras de roteamento baseadas no hostname e no caminho da URL.

Na minha aplicação, o Ingress é usado para:

- Roteamento de requisições ao frontend, acessível pela URL frontend.k8s.local.
- Roteamento de requisições ao backend, acessível pela URL backend.k8s.local.

Isso permite que os serviços da aplicação sejam acessíveis externamente de forma organizada e eficiente.

4. Scripts para Automação

4.1 Script build-and-export.sh

O script build-and-export.sh foi desenvolvido para automatizar a construção das imagens Docker da aplicação, tanto para o backend quanto para o frontend, e garantir que essas imagens sejam registradas no ambiente Docker do Minikube.

4.2 Script deploy-to-minikube.sh

O script deploy-to-minikube.sh é responsável por automatizar a implantação da aplicação no cluster Kubernetes, aplicando os arquivos YAML correspondentes ao backend, frontend e Ingress.

5. Conclusão

Esta documentação descreve os principais componentes da aplicação e os artefatos Kubernetes utilizados para sua implantação. A aplicação é organizada em dois containers, backend e frontend, que são orquestrados pelo Kubernetes por meio de Deployments, Services e Ingress.

Os scripts bash fornecidos automatizam o processo de build e implantação, garantindo que a aplicação possa ser implantada no ambiente do Minikube de forma eficiente e consistente. Dessa maneira, tanto o desenvolvimento quanto a implantação ficam mais simplificados e escaláveis.