

T2 DevOps

Eduardo Minoru Takeda - 776857

Kevin Vinicius Carvalho de Brito - 769769

Aplicação

A aplicação escolhida é composta por backend (NodeJS), frontend (ReactJS) e banco de dados (MongoDB). O backend possui dois métodos: GET e POST, e, após completar o formulário, a api salva as informações no MongoDB e posteriormente são extraídos os dados via GET em `/api/todos/`.

Contêineres Adicionais

Foram adicionados 2 contêineres adicionais para atingir o número mínimo pedido do trabalho. O Nginx e o Redis. O Nginx serve para atuar como intermediário entre o cliente (usuário) e o backend de forma segura. Enquanto que o Redis tem a função de cache.

Artefatos Kubernetes

Os artefatos são basicamente deployments e services de cada container, com o Banco de Dados sendo o único com arquivos a mais para configuração de volumes e variáveis de login. Há também o arquivo do Ingress que servirá de chave para acessar a api. Os deployments são responsáveis por manter os pods ativos e subir a configuração (portas, nomes, labels, conexões entre os pods) para o Minikube, enquanto o service faz o papel de conector. Todos os pods são do tipo ClusterIP, para que sejam visíveis somente entre si, sendo a única exportação via Ingress.

Instalação

Para instalar o trabalho basta clonar o git em que está arquivado

(<https://github.com/Krayth/DevOps-T2>), entrar no diretório

“docker-frontend-backend-db-master” e digitar no terminal “minikube-windows-up.bat” ou

“chmod +x minikube-linux-up.sh && ./minikube-linux-up.sh” dependendo da SO da sua máquina. É necessário ter o Docker e Minikube instalados. Se possível, reserve pelo menos 5 cpus no minikube start. Após a conclusão do script, a api deve ser acessada pelo link:

<http://todo.k8s.local>

Importante

Para executar os scripts é importante que esteja na pasta

“docker-frontend-backend-db-master” para que o script não se perca nos diretórios. Tanto os scripts ups quanto os downs.