

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

DevOps GRAD_1001544_A_SAO_CARLOS_2024_1

github.com/LucasMBalheiro/DevOps-Kubernetes

Prática Kubernetes

Prof. Delano Medeiros Beder

Aluno:

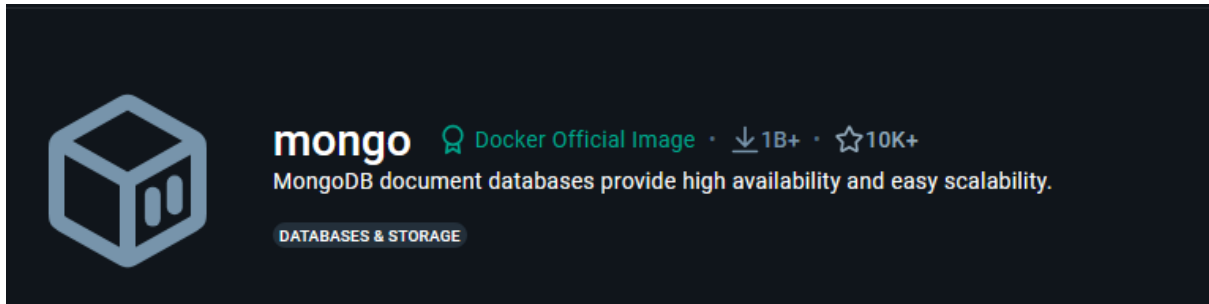
Lucas Maciel Balieiro, 800534, BCC

12 de Setembro de 2024 - São Carlos/SP

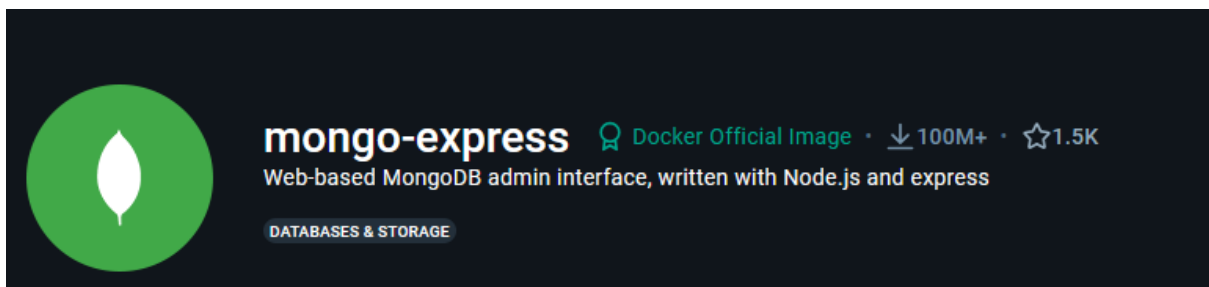
Descrição da aplicação

A aplicação utiliza 3 containers em seu funcionamento, sendo eles:

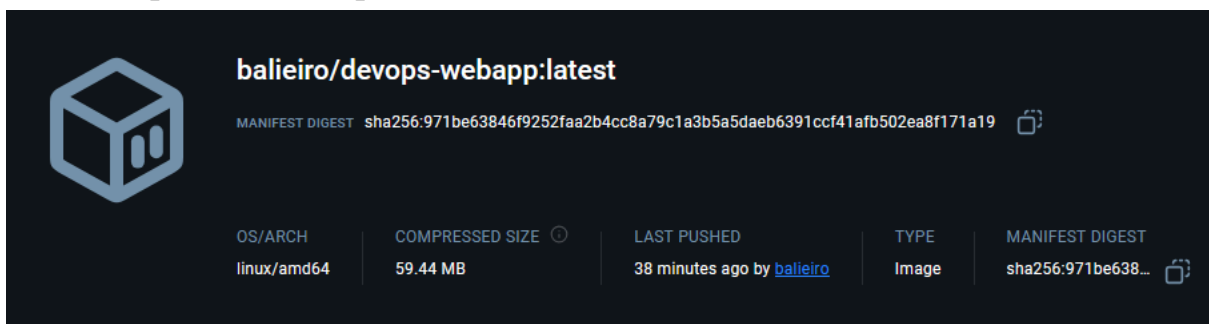
- A database escolhida, sendo ela o MongoDB



- Uma API para trabalhar com o banco de dados, Mongo-Express



- A aplicação web para visualização



A imagem do web app, neste caso, é a build da entrega do primeiro trabalho da matéria.

Arquivos do MongoDB (/database)

ConfigMap - O configmap da database simplesmente expõe a url de entrada do seu service, para que outros pods possam acessá-la.

Deployment - A criação desse pod expõe a porta 27017 (a padrão do banco de dados), além disso, possui variáveis de ambiente necessárias para o ingresso do usuário no banco de dados: MONGO_INITDB_ROOT_USERNAME e MONGO_INITDB_ROOT_PASSWORD

Secret - No secret, temos os valores das variáveis de ambiente usadas no deployment, nesse caso, a criptografia delas é somente base64.

Service - Simplesmente expõe a porta 27017 do Mongo para o seu uso dentre os outros serviços.

Arquivos do Mongo Express (/express)

Deployment - Ele expõe a porta 8081 para acesso da API pelo sistema, além disso, ele possui algumas variáveis necessárias para seu funcionamento:

- ME_CONFIG_MONGODB_ADMINUSERNAME = o mesmo usado para fazer o login na database, ele retira esse dado do secret demonstrado anteriormente.
- ME_CONFIG_MONGODB_ADMINPASSWORD = o mesmo usado para fazer o login na database, ele retira esse dado do secret demonstrado anteriormente.
- ME_CONFIG_MONGODB_SERVER = esta variável de ambiente é utilizada para comunicar o Mongo Express com o service do MongoDB, caso este dado não seja atribuído, a API não terá como ser utilizada.

Service - Além de expor a porta 8081 para ser utilizada, este service é do tipo de LoadBalancer e expõe a porta node número 30000. Isto é utilizado caso necessite acessar o serviço fora do cluster, ou seja, expondo para fora do Kubernetes.

Arquivos da aplicação web (/web)

Deployment - O deployment expõe a porta 3000 para uso, puxando os dados da imagem criada no DockerHub. As variáveis de ambiente são similares as antes aqui comentadas:

- USER_NAME = nome de usuário para conectar na database, buscado no database/secret
- USER_PWD = senha do usuário para conectar na database, buscado no database/secret
- DB_URL = url da database, retirada do configmap, necessária para fazer as requisições do banco.

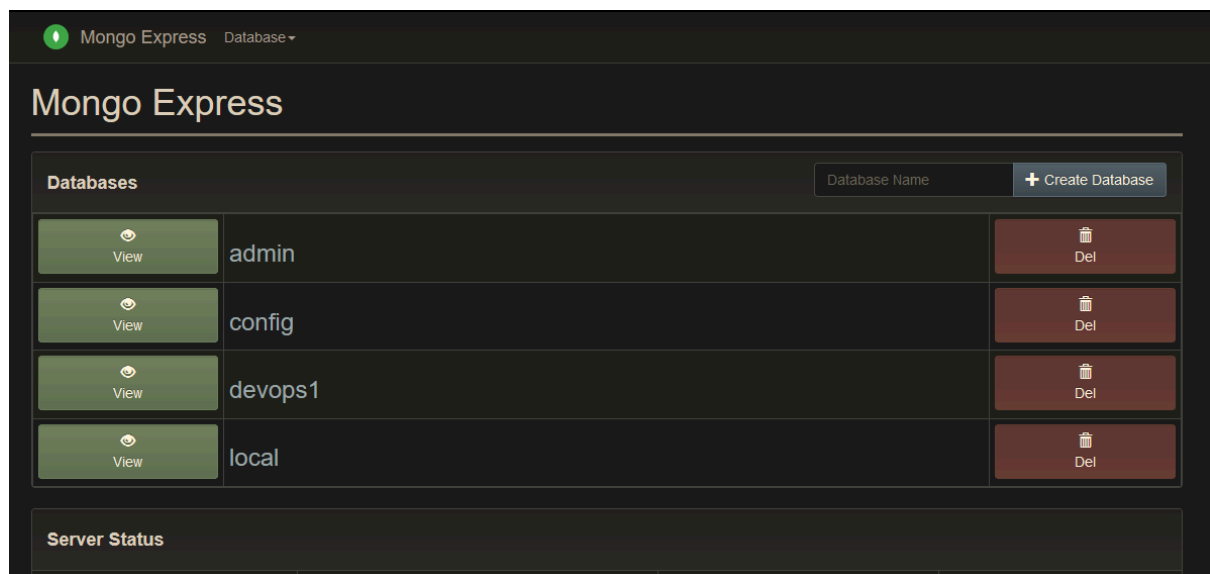
Service - A principal diferença deste serviço é que ele é do tipo NodePort, atribuindo uma porta estática para fora do cluster. O tráfego do serviço é exposto na porta 3000, com um nodePort na porta 30100.

Ingress

O ingress está hosteando a porta k8s.local, expondo a página web, com o MongoExpress estando hosteado no caminho /mongo. Não consegui fazer o ingress funcionar corretamente na minha máquina, então não posso garantir o funcionamento dos endereços.

Aplicação rodando

```
Administrator: Windows PowerShell
PS D:\Github\DevOps-Kubernetes\web> minikube service mongo-express
-----|-----|-----|-----|
| NAMESPACE | NAME       | TARGET PORT | URL                               |
|-----|-----|-----|-----|
| default    | mongo-express | 8081        | http://192.168.49.2:30000       |
|-----|-----|-----|-----|
* Starting tunnel for service mongo-express.
|-----|-----|-----|-----|
| NAMESPACE | NAME       | TARGET PORT | URL                               |
|-----|-----|-----|-----|
| default    | mongo-express |            | http://127.0.0.1:58621         |
|-----|-----|-----|-----|
* Opening service default/mongo-express in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run i
t.
```



```
PS D:\Github\DevOps-Kubernetes> minikube service webapp-service
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | webapp-service | 3000 | http://192.168.49.2:30100 |
|-----|-----|-----|-----|
* Starting tunnel for service webapp-service.
|-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
|-----|-----|-----|-----|
| default | webapp-service | | http://127.0.0.1:58723 |
|-----|-----|-----|-----|
* Opening service default/webapp-service in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run i
t.
```

O (não mais) melhor site apresentado na materia de DevOps...

Porque rodar o ingress pelo Windows é tão difícil? **Eu tentei muito ;-;**

Como poderia solucionar este problema?

O (não mais) melhor site apresentado na materia de DevOps...

Porque rodar o ingress pelo Windows é tão difícil? **Eu tentei muito ;-;**

Como poderia solucionar este problema? **Usando Linux :)**
