

MATHEUS PAVANI

POSTECH

SOFTWARE ARCHITECTURE

DOCKERIZAÇÃO

AULA 06

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS.....	5
O QUE VOCÊ VIU NESTA AULA?	10
REFERÊNCIAS.....	11
PALAVRAS-CHAVE	12

EMSE

O QUE VEM POR AÍ?

Nesta aula, vamos aprender como escalar uma aplicação em um container Docker, trabalhando com algo que, no mercado, pode ser bem-visto, como é o caso do Elastic Search.

Ao final da disciplina, teremos um desafio que faz parte da sua jornada de aprendizado, e você pode acessá-lo na área de atividades da plataforma FIAP ON.

Este desafio vai te ajudar a praticar os conhecimentos adquiridos durante as aulas e te preparar para o projeto da fase!

HANDS ON

Agora veremos, na prática, como subir a imagem do Elastic Search e fazer isso ser o mais próximo do produtivo, sem termos tanto trabalho manual.

A seguir, temos uma ideia do código que foi desenvolvido na videoaula, mas é importante que, ao olhar para ele, você consiga se desenvolver e trabalhar de maneira livre. Isso significa que é interessante mexer no código e praticar, explorar as funcionalidades e parâmetros. Por isso as documentações são tão importantes!

EXEMPLO

SAIBA MAIS

Após nossa caminhada pela jornada do conhecimento em Docker, passando pelos conceitos iniciais (mais teóricos) e entendendo na prática como funciona essa ferramenta incrível, agora vamos para mais aplicações bacanas.

Vamos pensar juntos: com essa ferramenta poderosa, não é possível que fiquemos ali só comunicando com códigos simples, não é mesmo?!

Agora, vou te transportar para uma situação corriqueira na tarefa de desenvolvimento: imagine que, neste momento, eu e você fomos convocados pela empresa em que trabalhamos, para assumir um grande projeto dentro da startup unicórnio “FIAP&Alura Technologies”. Esse projeto consiste em carregar em tempo real, um grande volume de dados (big data) de forma escalonada para uma equipe de Advanced Analytics analisar e criar algoritmos. Nossa missão aqui é fazer isso de maneira escalonada utilizando Docker. É de fato um desafio e tanto né? Mas e se nós dissemos que existe uma ferramenta muito bacana pra isso? E que ela se chama: Elastic Search?!

O Elastic Search é um motor de busca distribuído, de código aberto, e que permite indexar, pesquisar e analisar grandes volumes de dados de maneira rápida e eficiente. É uma ferramenta poderosa que foi projetada para lidar com grandes quantidades de informações de forma escalável e ágil, tornando-a ideal para aplicações empresariais e de análise de dados.

Ele foi criado com a intenção de fornecer uma solução de busca e análise de dados fácil de usar e escalável, e que permita aos usuários realizar pesquisas complexas em segundos. É baseado em tecnologias de clusterização, o que significa que ele pode escalar horizontalmente para acompanhar o crescimento do volume de dados. Além disso, o Elastic Search oferece uma ampla gama de recursos de pesquisa, análise e agregação de dados, permitindo aos usuários obter insights valiosos a partir de seus dados.

A principal aplicação do Elastic Search é a busca em grandes quantidades de dados. É amplamente utilizado por empresas para indexar e pesquisar grandes quantidades de informações, como registros de vendas, dados de log de aplicativos, dados de monitoramento de mídia social e muito mais. O Elastic Search pode ser

integrado a uma ampla gama de fontes de dados, incluindo bancos de dados relacionais, arquivos CSV, logs de aplicativos e muito mais, tornando-o uma ferramenta versátil e flexível para as empresas.

Além da busca, ele também é amplamente utilizado para análise de dados e geração de relatórios. Ele permite aos usuários agregar dados de várias fontes, aplicar filtros e realizar análises avançadas para obter insights valiosos sobre suas operações. Isso inclui aplicações como monitoramento de desempenho de aplicativos, análise de tendências de vendas, monitoramento de sentimentos em mídia social e muito mais.

Outra aplicação popular dessa ferramenta é a criação de aplicativos de visualização de dados. Ele permite aos desenvolvedores criar dashboards interativos e visualizações de dados para ajudar as equipes a tomar decisões informadas com base em dados. Isso inclui aplicações como painéis de monitoramento de rede, dashboards de análise de vendas e muito mais. Além disso, o Elastic Search é compatível com uma ampla gama de ferramentas de visualização de dados, incluindo o Kibana, que é uma ferramenta de análise e visualização de dados desenvolvida pela mesma equipe do Elastic Search.

Também é amplamente utilizado em aplicações de segurança de TI, como análise de log de segurança, detecção de ameaças e monitoramento de segurança. Ele permite aos profissionais de segurança indexar, pesquisar e analisar grandes quantidades de log de segurança e dados de ameaças em tempo real, o que os ajuda a detectar e responder rapidamente a possíveis ameaças.



Figura 1 – Elastic Search
Fonte: elastic.co/pt/ (2023)

Agora que já fomos apresentados a essa ferramenta... e o Docker? Onde ele entra nessa jogada?

Primeiro de tudo, vamos pensar naquela solução bacana da aula passada (onde criamos dois containers que se comunicavam)? Lá, o exemplo era um que servia como banco de dados e outro que consumia os dados do primeiro. Partindo da mesma ideia, faremos assim:

Vamos criar um Dockerfile contendo o que precisamos para criar o container com Elastic Search com esses dados:

Use a imagem oficial do Elasticsearch

FROM docker.elastic.co/elasticsearch/elasticsearch:7.10.2

Configure o cluster

ENV discovery.type single-node

Configure a porta exposta

EXPOSE 9200 9300

Criaremos uma rede que será compartilhada entre os containers:

```
docker network create my-network
```

Vamos construir a imagem com o seguinte comando no terminal:

```
docker build -t elasticsearch .
```

E, em seguida, iniciar o container do Elastic Search:

```
docker run --name elasticsearch --net my-network -p 9200:9200 -p 9300:9300 -d elasticsearch
```

Já devemos ter criado o código de aplicação app.py que será colocado dentro do container com o seguinte código:

```
import requests  
  
# URL de acesso ao Elasticsearch  
elasticsearch_url = "http://elasticsearch:9200"  
  
# Consulta Elasticsearch para obter todos os dados  
response = requests.get(elasticsearch_url + "/_search")  
  
# Verifique se a resposta é válida  
if response.status_code == 200:  
    # Imprima os dados retornados pelo Elasticsearch  
    print(response.json())  
else:  
    # Imprima uma mensagem de erro  
    print("Erro ao obter dados do Elasticsearch")
```

Caso você queira utilizar mais bibliotecas para outras ações, é interessante que utilize um arquivo requirements.txt contendo as bibliotecas. Mas mesmo com uma

biblioteca no código, é interessante já aderir a essa boa prática. Vamos ao Dockerfile do código Python:

```
# Use a imagem oficial do Python  
FROM python:3.9  
  
# Crie o diretório de trabalho  
WORKDIR /app  
  
# Copie o arquivo requirements.txt para o diretório de trabalho  
COPY requirements.txt .  
  
# Instale as dependências  
RUN pip install -r requirements.txt  
  
# Copie o código da aplicação para o diretório de trabalho  
COPY . .  
  
# Configure a porta exposta  
EXPOSE 5000  
  
# Defina o comando a ser executado quando o container for iniciado  
CMD [ "python", "app.py" ]
```

Vamos construir a imagem do segundo container:

```
docker build -t app .
```

Agora, devemos iniciar o segundo container, especificando a dependência com o primeiro container:

```
docker run --name app --net my-network --link elasticsearch:elasticsearch -p 5000:5000 -d app
```

O QUE VOCÊ VIU NESTA AULA?

Nesta aula vimos como trazer uma aplicação de mercado para dentro de um container Docker.

Daqui para a frente, é importante que você replique os conhecimentos adquiridos para fortalecer ainda mais as suas bases e conhecimentos, já que um bom (ou uma boa) cientista de dados não é somente aquele(a) que é uma enciclopédia humana, mas sim aquele(a) que sabe ler um problema e atuar com eficácia.

IMPORTANTE: não esqueça de praticar com o desafio da disciplina, para que assim você possa aprimorar os seus conhecimentos!

Você não está sozinho(a) nesta jornada! Te esperamos no Discord e nas *lives* com os professores especialistas, onde você poderá tirar dúvidas, compartilhar conhecimentos e estabelecer conexões!

REFERÊNCIAS

POULTON, N. **Docker Deep Dive**. [s.l.]: Independently published, 2017.

VITALINO, J. F. N. **Descomplicando o Docker**. [s.l.]: Brasport. 2018.

EMEND

PALAVRAS-CHAVE

Docker. Elasticsearch. Network.

EMEND



POSTECH