

MATHEUS PAVANI

POS TECH

SOFTWARE ARCHITECTURE

DOCKERIZAÇÃO

AULA 05

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

EMANIP

O QUE VEM POR AÍ?

Nesta aula, vamos aprender a trabalhar com docker network e verificar as possibilidades que podemos abordar com o tema, além de entender como funciona uma rede compartilhada (bridge).

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

Este desafio vai te ajudar a praticar os conhecimentos adquiridos durante as aulas, assim como na preparação para o projeto da fase!

HANDS ON

Agora chegou o momento de ver, na prática, como começar a criar nossas próprias redes bridge do Docker, e como levar isso de maneira ampla e produtiva.

A seguir, temos uma ideia de código que foi desenvolvido no vídeo da aula, 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!

SAIBA MAIS

É comum que ao longo de nossa trajetória em tecnologia, seja na área de dev, ou seja como cientista de dados, nós falemos aquela frase clássica:

“Nossa! Mas na minha máquina funciona!”, ou até mesmo: “Impossível fazer esses containers se comunicarem na rede!”.

É nesse contexto, jovem padawan, em que vamos mergulhar! Se já entendemos que os containers são uma porta de entrada para separar aplicações e suas dependências, vamos imaginar o seguinte caso:

Você, uma pessoa da área de tecnologia que aprendeu a usar containers Docker e vai aplicando isso em seu dia a dia, se depara com um abismo: suas aplicações precisam, necessariamente, estar na mesma rede, e seu trabalho é fazer isso de uma maneira fácil e em poucas linhas de código.

O subsistema de rede do Docker é conectável usando drivers. Vários drivers existem por padrão e fornecem funcionalidade básica de rede:

Bridge: o driver de rede padrão. Se você não especificar um driver, este é o tipo de rede que você está criando. As redes de ponte geralmente são usadas quando seus aplicativos são executados em containers independentes que precisam se comunicar. Consulte redes de pontes.

Host: para containers autônomos, remova o isolamento de rede entre o container e o host do Docker, e use a rede do host diretamente. Consulte usar a rede do host.

Overlay: as redes de sobreposição conectam vários daemons do Docker e permitem que os serviços de enxame se comuniquem entre si. Você também pode usar redes de sobreposição para facilitar a comunicação entre um serviço de enxame e um container autônomo, ou entre dois containers autônomos em diferentes daemons do Docker. Essa estratégia elimina a necessidade de roteamento no nível do sistema operacional entre esses containers. Consulte redes de sobreposição.

Ipvlan: as redes Ipvlan oferecem aos usuários controle total sobre o endereçamento IPv4 e IPv6. O driver VLAN se baseia nisso, dando às operadoras controle completo da marcação VLAN da camada 2 e até mesmo do roteamento Ipvlan L3 para usuários interessados na integração de rede subjacente. Veja redes Ipvlan.

Macvlan: as redes Macvlan permitem que você atribua um endereço MAC a um container, fazendo com que ele apareça como um dispositivo físico em sua rede. O daemon do Docker roteia o tráfego para containers por seus endereços MAC. Às vezes, usar o Macvlan driver é a melhor escolha ao lidar com aplicativos herdados que esperam ser conectados diretamente à rede física, em vez de roteados pela pilha de rede do host Docker. Veja redes Macvlan.

None: para este container, desative todas as redes. Geralmente usado em conjunto com um driver de rede personalizado, none não está disponível para serviços de enxame. Consulte desabilitar a rede de containers.

Plug-ins de rede: você pode instalar e usar plug-ins de rede de terceiros com o Docker. Esses plug-ins estão disponíveis no Docker Hub ou em fornecedores terceirizados. Consulte a documentação do fornecedor para instalar e usar um determinado plug-in de rede.

Imagine agora um cenário onde nossa equipe de desenvolvimento solicitou que colocássemos em produção 3 aplicações distintas em containers Docker separados. Porém, é necessário que além deles estarem separados, a rede entre eles seja compartilhada.

Logo, o que estamos procurando é utilizar uma rede do tipo bridge.

Bridge Network (Redes de ponte)

As redes de ponte se aplicam a containers em execução no mesmo host de daemon do Docker. Para comunicação entre containers em execução em diferentes hosts de daemon do Docker, você pode gerenciar o roteamento no nível do sistema operacional ou pode usar uma rede de sobreposição.

Quando você inicia o Docker, uma rede de ponte padrão (também chamada de bridge) é criada automaticamente e os containers recém-iniciados se conectam a ela,

a menos que especificado de outra forma. Você também pode criar redes de ponte personalizadas definidas pelo usuário. As redes de ponte definidas pelo usuário são superiores à rede padrão bridge.

Em termos de rede, uma rede de ponte é um dispositivo de camada de enlace que encaminha o tráfego entre segmentos de rede. Uma ponte pode ser um dispositivo de hardware ou um dispositivo de software executado no kernel de uma máquina host.

Em termos de Docker, uma rede de ponte usa uma ponte de software que permite que containers conectados à mesma rede de ponte se comuniquem, enquanto fornece isolamento de containers que não estão conectados à essa rede de ponte. O driver de ponte do Docker instala regras automaticamente na máquina host para que os containers em diferentes redes de ponte não possam se comunicar diretamente uns com os outros.

Use o comando `docker network create` para criar uma rede de ponte definida pelo usuário.

```
$ docker network create my-net
```

Você pode especificar a sub-rede, o intervalo de endereços IP, o gateway e outras opções. Consulte a referência de criação de rede docker ou a saída de `docker network create --help` para obter detalhes.

Para conectar um container em execução a uma ponte definida pelo usuário existente, use o `docker network connect`. O seguinte comando conecta um container, `meu-container`, já em execução à uma rede já existente, `minha-rede`:

```
$ docker network connect minha-rede meu-container
```

A ideia é que com a teoria e comandos acima, tudo isso seja feito diretamente no terminal em nossa máquina virtual, para que você possa encarar o processo de disponibilização de rede entre os containers.

Nesse contexto, podemos realizar os passos a seguir codando ao vivo e vendo o quão prático é essa comunicação:

- a. Criar uma rede:

```
docker network create shared-network
```

- b. Iniciar o container de banco de dados postgres:

```
docker run --name database --network shared-network -e  
POSTGRES_PASSWORD=password -d postgres
```

- c. Iniciar um container Python que vai ter acesso ao banco de dados:

```
docker run --name python-app --network shared-network -d  
python:3.8 bash
```

- d. Acessar o container Python e instalar as bibliotecas necessárias:

```
docker exec -it python-app bash  
pip install psycopg2-binary
```

- e. Escrever um código Python:

```
import psycopg2  
  
conn = psycopg2.connect(  
    host="database",  
    database="postgres",  
    user="postgres",  
    password="password")  
  
cursor = conn.cursor()  
cursor.execute("SELECT * FROM <table_name>")  
rows = cursor.fetchall()  
  
for row in rows:  
    print(row)  
  
conn.close()
```


- f. Execute o código dentro do container:

python <nomedoarquivo>.py

EXEMPLO

O QUE VOCÊ VIU NESTA AULA?

Nesta aula vimos como criar nossas primeiras redes compartilhadas para que um ou mais containers possam acessá-las.

Daqui para a frente, é importante que você replique os conhecimentos adquiridos para fortalecer mais suas bases e conhecimentos, já que para ter excelência como cientista de dados não basta apenas ser uma enciclopédia humana, mas também saber 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 só nesta jornada! Te esperamos no Discord e nas *lives* com os nossos 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

Palavras-chave: Docker. Redes. Bridge.

EMSE



POSTECH