

4525 – Curso de Infraestrutura Ágil com Práticas DevOps

Desafio - Administrando e Operando o Docker

Neste desafio iremos utilizar o conhecimento que tivemos na aula **Administrando e Operando o Docker** para evoluir a configuração do nossos containers do Dockers. A ideia é criar dois containers em redes específicas (homologação e produção) compartilhando diretórios com a máquina física.

Tema: Gerenciando networks, imagens e volumes entre os containers

1. Criar duas redes adicionais no Docker Engine que representarão Homologação e Produção:

Opções	REDE 1	REDE 2
NOME	dexter-production	dexter-homolog
ENDEREÇO	172.30.0.0/16	172.31.0.0/16

\$ docker network create --subnet 172.30.0.0/16 dexter-production

\$ docker network create --subnet 172.31.0.0/16 dexter-homolog

2. Criar dois novos containers onde o primeiro será de homologação e o segundo de produção, porém o container de produção deverá ser criado a partir de uma imagem do servidor de homologação. Um ponto de atenção **ANTES** de criar o container é que nesse ambiente, iremos compartilhar o diretório /var/log do contaneir para um diretório específico na máquina física, como por exemplo: /logs/container-a /logs/container-b /logs/container-c

OBS: Não se esqueça de criar o diretório /logs na máquina física!

Isso irá nos ajudar no gerenciamento de problemas e criação de relatório com ferramentas de estatísticas, pois o acesso ao logs dos containers estarão todos centralizados na máquina física.

Para criar o containers de homologação siga as instruções abaixo:

Crie um container com a rede dexter-homolog seguindo as especificações abaixo e após a criação do container você precisará conectar o mesmo a rede padrão criada na instalação do docker-engine, nome: bridge.

- **Nome:** dns-homolog
- **Hostname:** nshomolog
- **Endereço de IP fixo:** 172.31.0.10
- **Imagem:** CentOS
- **Compartilhe:** Diretório /var/log com o diretório /logs/dns-homolog da máquina física.

\$ mkdir /logs

\$ docker run -it --name dns-homolog --hostname nshomolog --net dexter-homolog --ip 172.31.0.10 -v /logs/dns-homolog:/var/log centos /bin/bash

Agora que o container foi criado, realizar a instalação dos seguintes pacotes:

- vim
- nano
- curl
- bind
- bind-utils
- tcpdump
- iproute

\$ yum install vim nano curl bind bind-utils tcpdump iproute

Inserir no arquivo /etc/vimrc as seguintes linhas:

```
" Habilitar linha numeradas no vim
set number
" Habilitar palavras em negritos no VIM.
```

```
set bg=dark
" Desabilitar case-sensitive nas buscas por texto.
set ic
" Destacar palavras que estão sendo procuradas.
set hlsearch
" Habilitar sistema de identificação automática
set smartindent
```

\$ vim /etc/vimrc

```
" Habilitar linha numeradas no vim
set number
" Habilitar palavras em negritos no VIM.
set bg=dark
" Desabilitar case-sensitive nas buscas por texto.
set ic
" Destacar palavras que estão sendo procuradas.
set hlsearch
" Habilitar sistema de identificação automática
set smartindent
```

conectar a rede de gerenciamento da Dexter (Rede padrão do Docker Engine)

\$ docker network ls

\$ docker network connect brigde dns-homolog

\$ docker exec dns-homolog ip a

3. Gerar uma imagem a partir do container criado no exercício anterior com o nome **dns**. Essa imagem será usada para criarmos nosso segundo container que será de produção.

\$ docker commit dns-homolog dns

4. Crie um container com a rede dexter-production seguindo as especificações abaixo e

pós a criação do container você precisará conectar o mesmo a rede padrão criada na instalação do docker-engine, nome: **brigde**,

- **Nome:** dns-master
- **Hostname:** ns1
- **Endereço de IP fixo:** 172.30.0.10
- **Imagem:** dns
- **Compartilhe:** Diretório /var/log com o diretório /logs/dns-master da máquina física.

IMPORTANTE: Por padrão, os containers do Docker possuem uma política para NÃO iniciar o containers caso a máquina física seja reiniciada. Para garantir que os nossos containers sejam iniciados (ligados) quando a máquina física for ligada ou reiniciada, você deve alterar a politica padrão dos containers no momento de sua criação. É esperado que você faça esse ajuste na criação do container de produção.

Dica: Utilize o comando 'man docker-run' para identificar desta opção.

```
$ docker run -it --name dns-master --hostname ns1 --net dexter-production --ip 172.30.0.10 -v /logs/dns-master:/var/log --restart=always dns /bin/bash
```

Conectando o container na rede de gerenciamento da Dexter (Rede padrão do Docker Engine)

```
$ docker network connect brigde dns-master
```

```
$ docker exec dns-master ip a
```

Após a realização do passos acima o seu desafio está completo.