

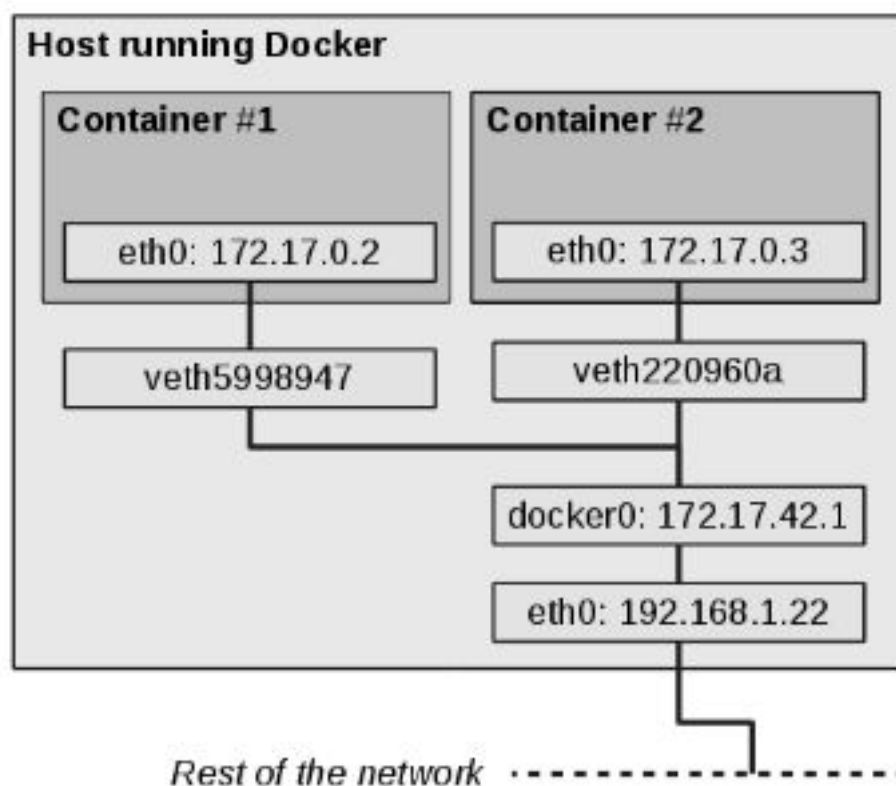
Seção 5 - Introdução a aplicações com Docker

1. Containers como microserviços

- Microserviço = Uma aplicação por container. Aplicação com uma única função;
- **Containers** ⇒ empacotados, leves e projetados para serem executados em qualquer lugar;
- 1. Ambientes de execução mais enxutos;
- 2. Melhor isolamento de recursos;
- 3. Inicialização e execução mais rápidas;

2. Básico de 'Container Networking'

- *Nenhum container é acessado diretamente pelo seu endereço IP entregue pela bridge0*



- Entre containers;
- Host hospedeiro;
- Resto do mundo*;

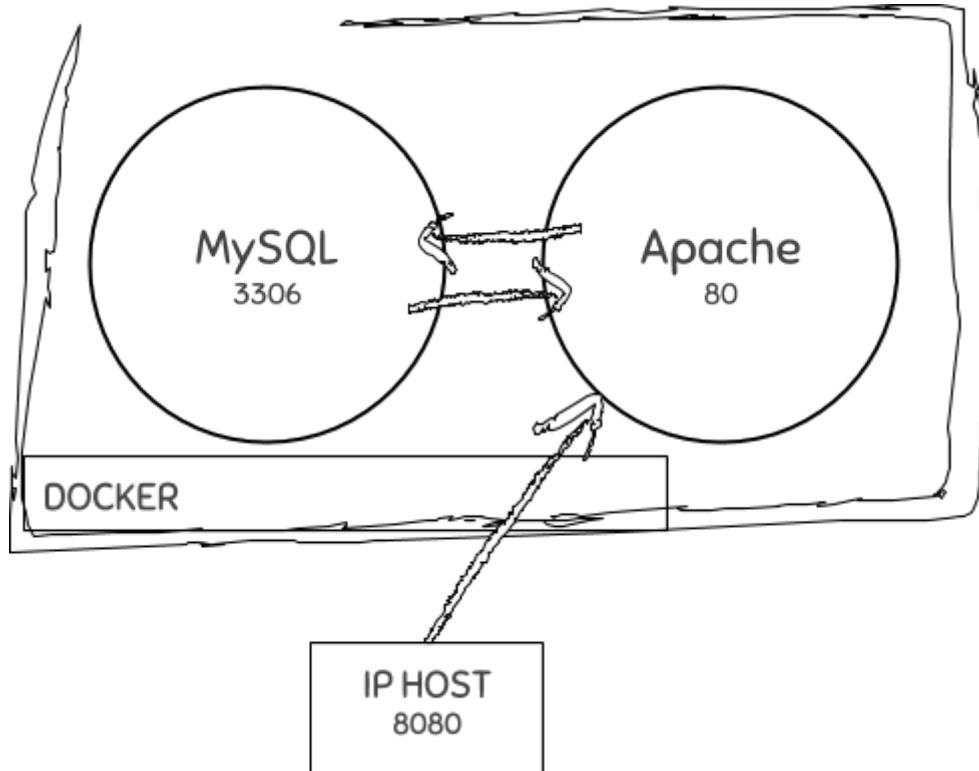
DOCKER0 (BRIDGE)

- “Ponte” entre o host e os containers;
- Criada no momento da instalação da Engine Docker;
- Aleatoriamente é criada uma subnet PRIVADA;

VIRTUAL ETHERNET (VETH)

- Camada que permite determinado container ter sua própria “eth”;
- Cada container recebe, automaticamente, endereço IP do segmento da “bridge0”;
- Interface virtual é criada no host;

3. Acesso externo ao container



MAPEAMENTO PORTAS

- Serviço do container precisa ser acessado externamente
- 1 porta do host $\Leftarrow \Rightarrow$ 1 porta do container;
- Mapeamento Manual ou Automático;
- Parâmetros opcionais “-p” ou “-P” no comando “docker run”

```
# docker run -d -p 8080:80 httpd
```

```
# docker run -d -P httpd
```

“Mão na massa”

No host, exibir e verificar as interfaces de redes criadas:

```
sudo ifconfig
```

** Observe as interfaces docker0 e veth. Além disso, encontre o segmento de rede PRIVADA criada na instalação da Engine Docker*

Verificar se o container com seu nome está ativo (iniciado), caso contrário iniciar e capturar o endereço IP que ele recebeu:

```
sudo docker ps  
sudo docker exec -it [CONTAINER NAME] ip a
```

** Observe que o endereço IP atribuído a esse container será do mesmo segmento da interface bridge “docker0”*



Acessar um terminal bash no container com seu nome:

```
sudo docker exec -it [CONTAINER NAME] bash
```

Verificar a conectividade com o “mundo exterior” pingando para algum host da internet:

```
ping 8.8.8.8  
ping google.com  
exit
```

Ou diretamente, pelo comando “docker exec”:

```
sudo docker exec -it [CONTAINER NAME] ping google.com
```

Iniciar um container que ofereça uma aplicação web, com mapeamento de portas manual

```
sudo docker run -itd --name web_80 -p 80:80 httpd
```

** Observe o parâmetro -p*

** Depois de iniciado, prossiga para o webbrowser do seu computador e acesse
http://localhost*

Iniciar um container que ofereça uma aplicação web, com mapeamento de portas automático

```
sudo docker run -itd --name web_random -P httpd
```

** Observe o parâmetro -P*

** Mapeamento automático gera um porta aleatória. Para descobri-la, execute:*

```
sudo docker port [CONTAINER_NAME]
```

** Depois de iniciado e descoberto a porta, prossiga para o webbrowser do seu computador e acesse http://localhost:PORTA*



Listar todos os containers ativos (iniciados):

`sudo docker ps`

- * *Observar as colunas COMMAND e PORTS de cada container*
- * *COMMAND → comando será executado na inicialização do container*
- * *PORTS → portas mapeadas com o host*

Referências

- ▶ <https://docs.docker.com/engine/userguide/networking/>
- ▶ https://docs.docker.com/engine/userguide/networking/default_network/
- ▶ <https://docs.docker.com/engine/reference/run/>

