

# Containers com *Docker*

Professor: José Eurípedes Ferreira de Jesus Filho  
*jeferreirajf@gmail.com*

Universidade Federal de Jataí – UFJ

# Aula anterior

- **Dockerfile:**

- Descrição passo a passo de como uma imagem deve ser
  - ✓ Geralmente uma imagem sempre é derivada de alguma imagem base.
  - ✓ A partir da imagem base, executa-se comandos e personalizações.
  - ✓ É possível definir uma série de características da imagem.
- Diversos comandos:
  1. **Workdir**
  2. **Run**
  3. **Cmd**
  4. **Entrypoint**
  5. **Copy**

Hoje falaremos sobre...

**Networks!**

# Introdução

- Já vimos no curso como fazemos para redirecionar uma porta do nosso computador para uma porta do nosso **container**.
- Vimos ainda como expor uma porta do nosso **container** para o mundo.
- Mas como essas coisas se conectam?

# Network

- Existem vários tipos básicos de redes no **Docker**:
  1. **Bridge**

Rede padrão do **Docker**. Utilizada quando você precisa que um **container** se comunique com outro.
  2. **Host**

Essa rede consegue mesclar a rede do **Docker** com a rede do computador que está hospedando o **Docker**.
  3. **Overlay**

Esse driver consegue criar uma “rede virtual” que coloca diferentes hosts de **Docker** na mesma rede. Muito comum para quando queremos utilizar **Docker** em uma arquitetura distribuída.
  4. **Macvlan**

Esse driver cava mais fundo nas camadas da rede e simula um **mac address** para cada **container**. Útil para simular a conexão com a camada física da rede.
  5. **None**

Nenhuma rede especificada.

# Network

- Podemos listar as redes do **Docker** com o seguinte comando:

```
docker network ls
```

# Network

- Podemos criar uma rede no **Docker** com o seguinte comando:

```
docker network create --driver <driver> <network_name>
```

# Network

- Podemos inspecionar uma rede no **Docker** com o seguinte comando:

```
docker network inspect <network_name>
```



# Network

- Podemos rodar um **container** e especificar a qual rede ele pertence:

```
docker run --network <network_name> <image>
```

# Network

- Podemos conectar um **container** em execução em uma rede existente:

```
docker network connect <network_name> <container>
```

# Network

- Podemos acessar a máquina **host** do **Docker** um **container** de dentro do container.
  - Basta apontar o endereço para `http://host.docker.internal:<port>`

**Hora da prática!**

# Exercício

- Quais são os tipos de redes do **Docker**? O que muda entre elas?

# Exercício

- Crie uma **imagem Docker** chamada **nginx-app** utilizando um **Dockerfile** a partir de um **nginx:1.25.2**. Modifique a página inicial do **nginx** para aparecer “**Hello world**”. Agora crie um **container** da imagem **nginx-app**. Crie ainda um **container** a partir da imagem do **bash:5.2.15** e entre nela. Acesse página inicial do **nginx-app** utilizando o **curl**.

# Exercício

- Crie uma **imagem Docker** chamada **nginx-app** utilizando um **Dockerfile** a partir de um **nginx:1.25.2**. Modifique a página inicial do **nginx** para aparecer “**Hello world**”. Agora crie um **container** da imagem **nginx-app** e coloque-o em uma rede chamada **nginx-network**. Crie ainda um **container** a partir da imagem do **bash:5.2.15** e coloque-o em uma rede chamada **bash-network**. De dentro do container, acesse página inicial do **nginx-app** utilizando o **curl**.