



## 02. Os primeiros comandos

|           |  |
|-----------|--|
| Date      | @21/02/2023                              |
| Categoria | Docker                                   |
| Curso     | Docker: criando e gerenciando containers |

### Tópicos

- Conhecendo o Docker Hub
- O comando docker run
- Fluxo da criação de containers
- Etapas do run
- Outros comandos importantes
- Run vs Exec
- Mapeando portas
- Visualizando mapeamentos
- Faça como eu fiz: Acessando portas externamente
- O que aprendemos?

### O comando docker run

Já aprendemos que ao utilizar o Docker precisamos conhecer e saber para que servem alguns comandos para fazer ações quando trabalhamos com containers, sendo um deles o `docker run`.

Escolha a alternativa que mostra como a definição do comportamento do Docker ao executarmos o comando.

- O comando `docker run` cria uma cópia de um container já em execução.

- O `docker run` é responsável pela execução de containers. Seu papel não é criar cópias.
  - O comando `docker run` é responsável por executar um container em nosso host.
    - Através deste comando, o docker irá executar o container da maneira esperada.
  - O comando `docker run` é responsável por iniciar o Docker dentro do sistema.
    - O `docker run` é responsável pela execução de novos containers.
- 

## Etapas do run

Quando queremos executar um container e usamos o comando `docker run`, ocorre uma série de etapas ordenadas até que a execução seja feita efetivamente.

Selecione a alternativa que melhor descreve este fluxo.

- Baixa a imagem -> Valida o hash da imagem -> Executa o container.
  - Procura a imagem localmente -> Baixa a imagem caso não encontre localmente -> Valida o hash da imagem -> Executa o container.
  - Procura a imagem localmente -> Baixa a imagem caso não encontre localmente -> Executa o container.
- 

## Run vs Exec

Recentemente, vimos sobre o comando `docker run` e `docker exec`. Sabemos que ambos os comandos envolvem o fluxo de inicialização e execução de comandos em containers, porém em contextos diferentes.

Selecione a alternativa com a diferença do funcionamento entre esses comandos.

- O `docker exec` cria um novo container e o executa. O `docker run` permite executar um comando em um container que já está em execução.
- Os dois geram o mesmo resultado.
- O `docker run` cria um novo container e o executa. O `docker exec` permite executar um comando em um container que já está em execução.

---

## Visualizando mapeamentos

As flags `-p` e `-P` são úteis quando queremos fazer o mapeamento de portas entre nosso container e o nosso host. Existe um comando responsável pela visualização de como o mapeamento de portas de um container está sendo feito.

Selecione a alternativa que corresponde a este comando.

- `docker port`
  - Este comando é responsável por exibir como o mapeamento de portas de um container está sendo feito.
- `docker inspect`
  - Este comando é responsável por inspecionar detalhes sobre determinado recurso.
- `docker ps`
  - Este comando é responsável por listar containers em execução.
- `docker network`
  - Este comando está relacionado com as redes do Docker.

---

## Faça como eu fiz: Acessando portas externamente

Agora nossa proposta é executar um container e validar que, através da flag `-p`, é possível acessá-lo diretamente a partir do host. Lembrando que já sabemos como executar containers, porém, devido ao isolamento, ainda não conseguimos validar o funcionamento deles. Vamos fazer essa execução e validação?

Inicialmente, execute o comando `docker run -d dockersamples/static-site` para que o seu host baixe a imagem e execute o container em seguida.

A seguir, verifique se o seu container está em execução sem problemas através do comando `docker ps` ou `docker container ls`. Repare na coluna `PORTS` e veja que seu container está informando que possui uma aplicação que pode ser exposta tanto na porta 80 quanto na 443.

Obtendo esta informação, remova o container recém-criado com o comando `docker container rm <id-do-container> --force`.

Seu objetivo agora será criar um novo container fazendo o devido mapeamento de portas. Para isso, execute o comando `docker run -d -p 8080:80 dockersamples/static-site`. Repare que através da flag `-p`, estamos informando que a porta 8080 de nosso host irá refletir na porta 80 do container.

Por fim, acesse em seu navegador `localhost:8080` e veja a aplicação sendo carregada dentro de seu container.

Com a utilização da flag `-p`, é possível fazer uma ponte entre a aplicação dentro do container e o nosso host. Este cenário é muito interessante quando queremos testar o funcionamento e como uma aplicação está interagindo com outras.

---

## O que aprendemos?

Nessa aula aprendemos:

- O Docker Hub é um grande repositório de imagens que podemos utilizar;
- A base dos containers são as imagens;
- Como utilizar comandos acerca do ciclo de vida dos containers, como:
  - `docker start`, para iniciar um container que esteja parado;
  - `docker stop`, para parar um que esteja rodando;
  - `docker pause`, para pausar um container;
  - `docker unpause` para iniciar um container pausado;
- Conseguimos mapear portas de um container com as flags `-p` e `-P`.