



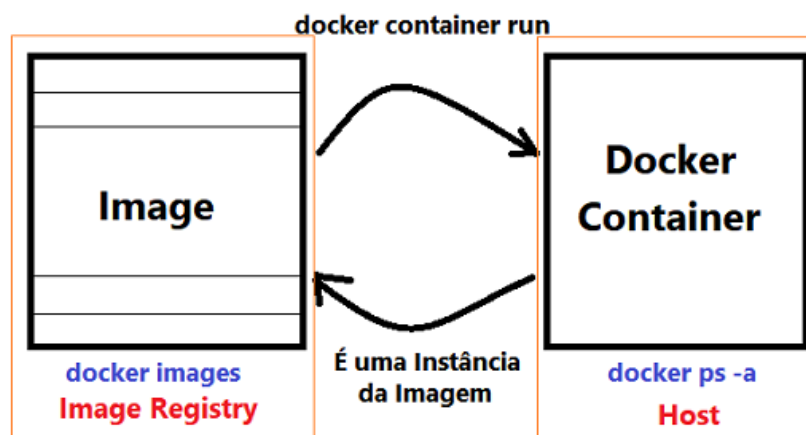
Neste artigo vou apresentar os conceitos básicos relativos ao Docker sob o ponto de vista de um desenvolvedor .NET.



**Imagens** são modelos que são usados para **criar contêineres** e que contêm um sistema de arquivos com todos os arquivos que a aplicação no contêiner requer. ([artigo anterior](#))



Assim um contêiner é uma instância de uma imagem:



Quando testamos o Docker no [segundo artigo](#), usamos um comando que instruiu o Docker para usar uma **imagem** chamada **hello-world**, que esta publicada no repositório público de imagens, conhecido como o [Docker Hub](#).

Dessa forma existem **repositórios de imagens** prontas para facilitar a nossa vida. Se você precisar criar um contêiner com o Banco de dados **MySQL**, você teria que montar a sua imagem e isso demandaria tempo. Para poupar o seu trabalho basta usar a imagem oficial do MySQL que é mantido pela Docker no repositório [Docker Hub](https://hub.docker.com/_/mysql/): [https://hub.docker.com/\\_/mysql/](https://hub.docker.com/_/mysql/)

A imagem **hello-world** contém todos os arquivos exigidos por um aplicativo que imprime uma simples mensagem de saudação, fornecendo uma maneira independente de distribuir os arquivos para que eles possam ser usados para executar o aplicativo.

Veremos a seguir como podemos tratar imagens.



## Trabalhando com imagens

Vamos agora exibir as imagens que temos na máquina local digitando o comando:

**docker images**

```
root@linux: /home/macoratti

Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             dbfc48660aeb       2 weeks ago        109MB
debian              latest             be2868bebaba       2 weeks ago        101MB
microsoft/dotnet-samples latest             481daa42f326       3 weeks ago        180MB
hello-world          latest             4ab4c602aa5e       8 weeks ago        1.84kB

root@linux:/home/macoratti#
```

A resposta do comando exibe uma relação de imagens disponíveis na máquina local com as seguintes informações:

<b>REPOSITORY</b>	O nome da imagem no repositório
<b>TAG</b>	A versão da imagem
<b>IMAGE ID</b>	A identificação da imagem (um hash atribuído pelo docker)
<b>CREATED</b>	A data de criação da imagem
<b>SIZE</b>	O tamanho ocupado pela imagem

Dessa forma quando criamos contêineres o Docker faz o download da imagem para a máquina local para agilizar tarefas futuras.

Para apenas baixar uma imagem de um repositório digite o comando: **docker pull <imagem>**

Onde <imagem> é o nome da imagem. Vamos baixar uma imagem da [Asp .Net Core](#) digitando:

**docker pull microsoft/aspnetcore**

```
root@linux: /home/macoratti

Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker pull microsoft/aspnetcore
Using default tag: latest
latest: Pulling from microsoft/aspnetcore
55cbf04beb70: Pull complete
2ad9fb8b9d3d: Pull complete
7debd121b442: Pull complete
76fbad01d0bd: Pull complete
1cba46fa0c00: Pull complete
Digest: sha256:8bd7808981568a92b7f1d0aa4bab07ef811e6417898847d9caf698a4a788ab11
Status: Downloaded newer image for microsoft/aspnetcore:latest
root@linux:/home/macoratti#
```

Repetindo o comando para exibir as imagens : **docker images**

```
root@linux: /home/macoratti

Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             dbfc48660aeb       2 weeks ago        109MB
debian              latest             be2868bebaba       2 weeks ago        101MB
microsoft/dotnet-samples latest             481daa42f326       3 weeks ago        180MB
hello-world          latest             4ab4c602aa5e       8 weeks ago        1.84kB
microsoft/aspnetcore latest             db030c19e94b       2 months ago        347MB

root@linux:/home/macoratti#
```

As imagens de repositório podem ser versionadas, permitindo que diferentes versões de uma imagem coexistam e garantindo que você obtenha a versão correta de uma imagem quando a baixar.

Uma **tag** é especificada ao anexar um dois pontos (:) para o nome da imagem, seguido pela **tag**.

Vamos agora baixar a imagem que contém a **ASP .NET Core e os runtimes do .NET Core** e bibliotecas otimizadas para rodar a ASP .NET Core em produção.

## **docker pull microsoft/dotnet:2.1-aspnetcore-runtime**

```
root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker pull microsoft/dotnet:2.1-aspnetcore-runtime
2.1-aspnetcore-runtime: Pulling from microsoft/dotnet
f17d81b4b692: Already exists
b89e58f928f1: Pull complete
ed484e612149: Pull complete
e354f96b5b0e: Pull complete
Digest: sha256:2f52f2ea349384f0f2adbef756f68414eb2fba4857c140f98e8eaa4961eded5a
Status: Downloaded newer image for microsoft/dotnet:2.1-aspnetcore-runtime
root@linux:/home/macoratti#
```

O comando usa a tag **:2.1** indicando que a imagem possui a versão **2.1** da **ASP .NET Core**.

Exibindo as imagens teremos: **docker images**

```
root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
microsoft/dotnet	2.1-aspnetcore-runtime	1fe6774e5e9e	2 weeks ago	255MB
nginx	latest	dbfc48660aeb	2 weeks ago	109MB
debian	latest	be2868bebaba	2 weeks ago	101MB
microsoft/dotnet-samples	latest	481daa42f326	3 weeks ago	180MB
hello-world	latest	4ab4c602aa5e	8 weeks ago	1.84kB
microsoft/aspnetcore	latest	db030c19e94b	2 months ago	347MB

```
root@linux:/home/macoratti#
```

Omitir a tag da imagem equivale a solicitar a imagem marcada como mais recente usando a tag **latest**.

Assim os comandos abaixo são equivalentes:

## **docker pull microsoft/aspnetcore e docker pull microsoft/aspnetcore:latest**

Para ver detalhes sobre as **tags** consulte o repositório: <https://hub.docker.com/r/microsoft/aspnetcore/>

Para deletar uma imagem da máquina local podemos usar o comando **rmi** e indicar o identificador da imagem ou o nome da imagem.

Vamos apagar a imagem **microsoft/aspnetcore** identificada pelo **IMAGE ID db030c19e94b** :

## **docker rmi db030c19e94b**

```
root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker rmi db030c19e94b
Untagged: microsoft/aspnetcore:latest
Untagged: microsoft/aspnetcore@sha256:8bd7808981568a92b7f1d0aa4bab07ef811e6417898847d9caf698a4a788ab11
Deleted: sha256:db030c19e94bce26158ce2a8afa99948fe04ac06c6d06019a3e65eb54f7e2130
Deleted: sha256:e323984dc026ae12ce5ceb89c82eed9514db5f32105d97e1f60421a1013425d1
Deleted: sha256:65c5a208a786db16c2b2933a65ab194e0e6420af50065be4738790eae63900e
Deleted: sha256:b6b1610980ee9032e95ec610fba7986feca8a78225b66aeb54a91327c16be44c
Deleted: sha256:18061986490a5984e6bc4119a71373b79d695bdd75b238f52827046fa0e87313
Deleted: sha256:3b10514a95bec77489a57d6e2fbfdbb7ddfdb643907470ce5de0f1b05c603706
root@linux:/home/macoratti#
```

Para obter o mesmo resultado poderíamos ter usando o comando informando o nome da imagem:

```
docker rmi microsoft/aspnetcore
```

Exibindo as imagens veremos que a imagem não aparece mais na listagem:

```
docker images
```

```
root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
microsoft/dotnet    2.1-aspnetcore-runtime 1fe6774e5e9e       2 weeks ago        255MB
nginx               latest             dbfc48660aeb       2 weeks ago        109MB
debian              latest             be2868bebaba       2 weeks ago        101MB
microsoft/dotnet-samples latest             481daa42f326       3 weeks ago        180MB
hello-world         latest             4ab4c602aa5e       8 weeks ago        1.84kB
root@linux:/home/macoratti#
```

Para remover uma imagem que estiver sendo usada por um contêiner usamos o parâmetro -f :

```
docker rmi -f db030c19e94b
```

Para apagar todas as imagens de uma vez usamos o comando:

```
docker rmi -f $(docker images -q)
```

O argumento -q especifica o modo silencioso(quiet) e retorna os valores de **IMAGE ID** para o comando e assim apaga todas as imagens.

Assim `docker images -q` retorna todos os valores de **IMAGE ID**:

```
docker images -q
```

```
root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker images -q
1fe6774e5e9e
dbfc48660aeb
be2868bebaba
481daa42f326
4ab4c602aa5e
root@linux:/home/macoratti#
```

Para inspecionar uma imagem e exibir detalhes de configuração usamos o comando:

```
docker inspect <NOME OU ID DA IMAGEM>
```

onde podemos usar o valor de **REPOSITORY** ou o valor de **IMAGE ID**:

## docker inspect 1fe6774e5e9e ou docker inspect microsoft/dotnet

```

root@linux: /home/macoratti
Arquivo Editar Ver Pesquisar Terminal Ajuda
root@linux:/home/macoratti# docker inspect 1fe6774e5e9e
[
  {
    "Id": "sha256:1fe6774e5e9e4be6904e2f0b61658c67225aa36b65f76b2949e386e8da124cde",
    "RepoTags": [
      "microsoft/dotnet:2.1-aspnetcore-runtime"
    ],
    "RepoDigests": [
      "microsoft/dotnet@sha256:2f52f2ea349384f0f2adbef756f68414eb2fba4857c140f98e8aaa4961eded5a"
    ],
    "Parent": "",
    "Comment": "",
    "Created": "2018-10-17T14:09:44.7302838Z",
    "Container": "9a10dea828ea675ad424b69dfd95b4fca41bf223d423151ff5261e7bf29ddffd",
    "ContainerConfig": {
      "Hostname": "",
      "Domainname": "",
      "User": "",
      "AttachStdin": false,
      "AttachStdout": false,
      "AttachStderr": false,
      "Tty": false,
    }
  }
]

```

Esse comando é útil para inspecionar informações detalhadas das imagens existentes e criadas.

Em todos os comandos mostrados não foi usado o identificador [image](#) que especifica que o comando esta atuando sobre uma imagem. Embora os comandos funcionem usar o identificador é a sintaxe recomendada.

Abaixo segue um resumo dos principais comandos para tratar imagens:

<a href="#">docker image build</a>	Constrói (Build) uma imagem a partir de um <a href="#">Dockerfile</a> .
<a href="#">docker image history</a>	Exibe o histórico de uma imagem
<a href="#">docker image import</a>	Importa o conteúdo a partir de um tarball p/criar imagem de sistema de arquivos
<a href="#">docker image inspect</a>	Exibe informação detalhada da imagem
<a href="#">docker image load</a>	Carrega um imagem a partir de um arquivo tar ou <a href="#">STDIN</a>
<a href="#">docker image ls</a>	Lista Imagens locais.
<a href="#">docker image prune</a>	Remove uma imagem não usada.
<a href="#">docker image pull</a>	Baixa uma imagem ou um repositório de um registro.
<a href="#">docker image push</a>	Envia uma imagem para um repositório. <i>(Tem que criar conta e fazer login)</i>
<a href="#">docker image rm</a>	Remove uma ou mais imagens. Remove tudo: <b>docker rmi \$(docker images -q)</b>
<a href="#">docker image save</a>	Salva uma ou mais imagem para um arquivo <a href="#">tar</a>
<a href="#">docker image tag</a>	Cria uma tag <a href="#">TARGET_IMAGE</a> que se refere a <a href="#">SOURCE_IMAGE</a>

Na [próxima aula](#) veremos como criar nossas próprias imagens.