



Imagens – Exercícios resolvidos

1 – Você criou uma aplicação Console chamada **appnetcore** no .NET Core usando o comando **dotnet new console** que imprime 'Ola Mundo!' no console.

Resposta:

```
mkdir appnetcore
cd appnetcore
dotnet new console
```

Altere o código da classe Program.cs conforme abaixo:

```
using System;

namespace appnetcorex1
{
    class Program
    {
        static void Main(string[] args)
        {
            if (args == null)
            {
                Console.WriteLine("Ola Mundo !");
            }
            else
            {
                Console.WriteLine(args[0].ToString());
            }
        }
    }
}
```

Agora o programa pode aceitar argumentos na linha de comando.

Agora você precisa compilar e executar essa mesma aplicação em um contêiner Docker. **Resposta:**

Para compilar e executar a aplicação no contêiner temos que usar a imagem do SDK (**microsoft/dotnet:2.1-sdk**)



Imagens – Exercícios resolvidos

a-) Defina um arquivo **Dockerfile** para criar uma imagem chamada **appnetcore:dev** para esta aplicação. **Resposta:**

```
# define a imagem base
FROM microsoft/dotnet:2.1-sdk

# define o diretório de trabalho no contêiner
WORKDIR /app

# copia o arquivo csproj e restaura as camadas
COPY *.csproj ./

# restaura as dependências do projeto
RUN dotnet restore

# copia tudo da pasta atual para o contêiner
COPY . ./

# executa o comando que publica o projeto na pasta out
RUN dotnet publish -c Release -o out

# executa o comando dotnet no arquivo do projeto na pasta out
ENTRYPOINT ["dotnet", "out/appnetcorex1.dll"]
```

Arquivo **Dockerfile** criado na raiz do projeto.

b-) Explique os comandos usados no arquivo Dockerfile. **Resposta:**
A explicação dos comandos está nos comentários do arquivo Dockerfile

c-) Qual o comando usado para criar a imagem a partir do Dockerfile
Resposta:

docker built -t appnetcore:dev .

Obs: O nome do arquivo Dockerfile não foi informado pois estamos usando o nome padrão. Se tivéssemos usado outro nome como **Dockerfile.dev**, poderíamos definir o comando assim:



Imagens – Exercícios resolvidos

`docker built -t appnetcore:dev -f Dockerfile.dev .`

Nota: O argumento `-f` permite informar uma localização e nome alternativos para o Dockerfile

d-) Indique o comando usado para criar um contêiner chamado **testeapp** com a imagem criada para testar sua aplicação

Resposta:

`docker container run --name testeapp appnetcore:dev 'Curso Docker'`

2 - Após testar sua imagem você precisa enviá-la para o repositório de imagens para disponibilizar a imagem para os membros da sua equipe. Quais os procedimentos para subir a imagem para o Docker Hub ? **Resposta:**

Para enviar uma imagem para o repositório de imagens você deve se cadastrar no Docker.hub e criar uma conta.(<http://hub.Docker.com>)

Após isso prepare a imagem, faça o login na sua conta e envie a imagem.

`docker image tag imagem <imagem>`

`docker login`

`docker image push <imagem>`

3 – Indique os comandos Docker que realiza as seguintes operações com imagens:

Resposta:

a - baixar uma imagem do repositório

`docker pull <imagem>`

b - exibir as imagens existentes

`docker images` ou `docker image ls`

c - associar um nome a uma imagem

`docker image tag image_atual <imagem>`



Imagens – Exercícios resolvidos

d - publicar uma imagem em um repositório

docker push <imagem>

e - processar o arquivo Dockerfile e criar uma imagem

docker build -t <nome_imagem> .

f - remover uma imagem do sistema local

docker image rm <imagem_id>

g- inspecionar uma imagem

docker image inspect <imagem>

h – apagar uma image

docker image rm <id> ou <nome>

i – apagar todas as imagens

docker image prune

j – exibir apenas os ids das imagens

docker images -q