Aplicação Asp.net Core

Publicando aplicação Asp.Net Core.  
*Criando app .net core, seu arquivo Dockerfile e publicando no container.*

Esse vai ser um tutorial básico de como criar um aplicativo .net core 5.0, definir seu Dockerfile para criação de uma imagem, criar e executar o container para que seja possível acessar aplicação na web.

1. Criar aplicação.
2. Publicar aplicação.
3. Criar o arquivo Dockerfile.
4. Criar a imagem.
5. Criar o contêiner.

# Criando Aplicação .Net Core

**App:** Criado normalmente no visual studio

* Criado uma aplicação .net core 5.0 mvc.



# Publicar aplicação

**--configuration Release:** Indica que estamos usando o modo Release que é o modo usado na produção.

**--output dist:** Especifica que o projeto compilado será copiado para uma pasta dist.

1. Abrir o prompt no projeto raiz.
2. dotnet publish --configuration Release --output dist.
3. O arquivo de ponto de entrada vai ser app.dll

# Criar arquivo Dockerfile

**dockerfile:** Arquivo deve ser criado na raiz do projeto.

* Definir uma imagem base.
* Definir informações para a imagem.
* Definir a pasta de trabalho **(/app)**
* Copiar os arquivos da pasta **dist** para uma pasta no contêiner **(/app)**
* Expor a porta do contêiner e definir em qual porta o servidor vai atender.
* Definir a ponte de entrada para aplicação (app.dll) .

FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base

LABEL version="1.0.1" description="Aplicacao Asp .Net Core"

WORKDIR /app

COPY dist /app

EXPOSE 80/tcp

ENTRYPOINT ["dotnet","AppMVC.dll"]

**dockerfile:** Arquivo deve ser criado na raiz do projeto.

# Criar a Imagem

**Executar após o Dockerfile está configurado:** docker build -t aspnetcoremvc:1.0 .

Desta forma a imagem irá ser criada baseado no arquivo Dockerfile.

Imagem foi criada?

1. Docker build – o comando.
2. -t – Parâmetro usado para informar que a imagem pertence ao usuário.
3. aspnetcoremvc:1.0 – nome da imagem e a tag atribuída.
4. . o diretório atual (a pasta que contém o dockerfile)

# Criar o Container

**Criando container:** docker container create -p 3000:80 –name mvcprodutos aspnetcoremvc:1.0

Desta forma a nossa aplicação estará rodando no container do docker.

1. docker container create – comando para criar o contêiner.
2. -p 3000:80 – mapeia a porta 3000 do host para a porta 80 do contêiner.
3. --name mvcprodutos – define o nome do contêiner.
4. aspnetcoremvc:1.0 - nome da imagem usada.
5. docker container start mvcprodutos

É possível acessar aplicação no docker mediante o link

<http://localhost:3000/>

# Publicar no DockerHub

1. Imagem criada e container testados, agora vamos publicar.
2. docker image tag idimagem userDockerHub/tagnovaimagem:1.0
3. docker push userDockerHub/tagnovaimagem:1.0