Install docker from docker hub

<https://hub.docker.com/>

Login in to docker using Docker ID and Pass. If you don’t have a Docker ID, create one on hub.docker.com

Create a Web API Application using .NET Core 3.1 with Docker support

Update docker file with following copy

**Docker file for single solution with single project**

FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS build-env

WORKDIR /app

# Copy csproj and restore as distinct layers

COPY \*.csproj ./

RUN dotnet restore

# Copy everything else and build

COPY . ./

RUN dotnet publish -c Release -o out

# Build runtime image

FROM mcr.microsoft.com/dotnet/core/aspnet:3.1

WORKDIR /app

COPY --from=build-env /app/out .

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

WebApiOnDocker is project name

**Open command prompt and navigate to project directory where your .csproj is located and run following command to build a docker image for your application.**

**> cd C:\dev\practice\WebApiOnDocker\WebApiOnDocker**

> docker build -t webapi-image . // build an image with name webapi-image from the Dockerfile located in current directory ( . ) , -t for tag latest is the default

Execute following command to deploy your docker image to run the application

> docker run -d -p 8080:80 –name webapi-container webapi-image

-d Run detached (live a service). Without --detach option (-d) we will start seeing app console output from container.

-p port mapping container outbound port 80, host machine port 8080

Go to <http://localhost:8080/weatherforecast> to access your api in a web browser.

> docker container stop container-name

> docker container rm container-name

> docker image rm image-name

> docker container ls

> docker images or > docker image ls

> docker build -t webapi-image .

> docker run -d -p 8080:80 –name webapi-container webapi-image

Reference

<https://www.softwaredeveloper.blog/multi-project-dotnet-core-solution-in-docker-image>

**Docker file for solution with multiple projects**

FROM mcr.microsoft.com/dotnet/core/sdk:3.1 AS build-env

WORKDIR /app

# Copy everything else and build

COPY . ./

RUN dotnet publish WebApiOnDocker -c Release -o out

# Build runtime image

FROM mcr.microsoft.com/dotnet/core/aspnet:3.1

WORKDIR /app

COPY --from=build-env /app/out .

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

**Open command prompt and navigate to root solution directory and run following command to build a docker image for your application.**

**> cd C:\dev\practice\WebApiOnDocker**

> docker build -f WebApiOnDocker/Dockerfile -t webapi-image .

> docker run -d -p 8080:80 –name webapi-container webapi-image