[**1. SECTION – 1 : Pull Image using Docker Hub** 1](#_Toc171864308)

[**2. SECTION – 2 : Create Image for Console Application** 3](#_Toc171864309)

[**3. SECTION – 2.1 : Create Image for .Net Core Application** 6](#_Toc171864310)

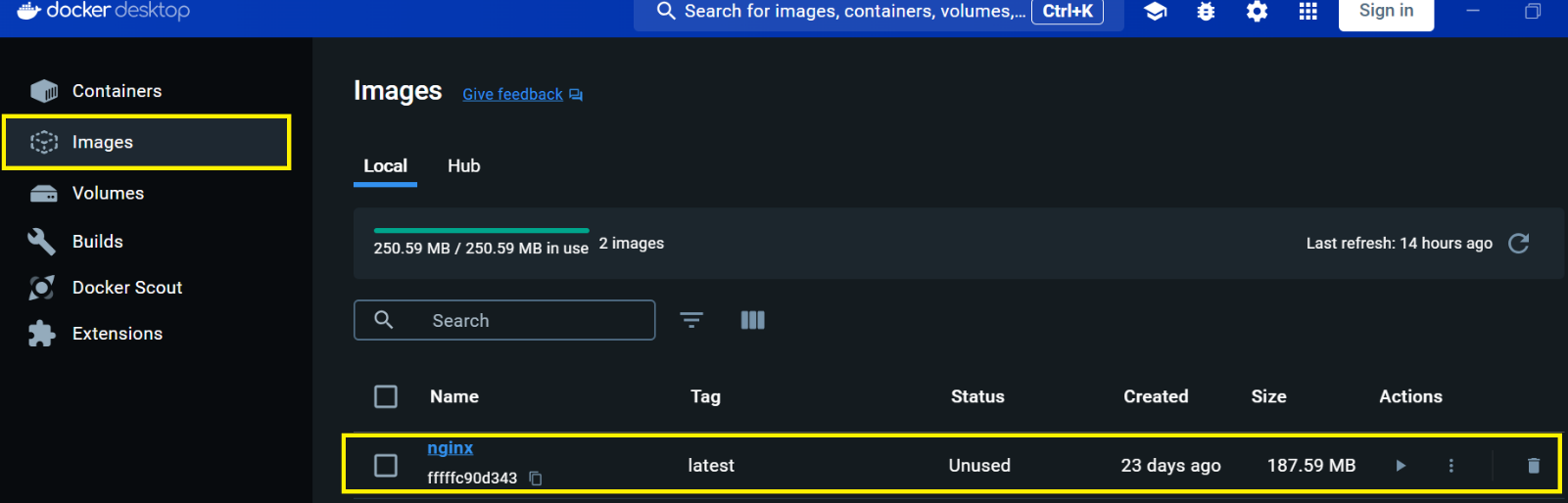
# **1. SECTION – 1 : Pull Image using Docker Hub**

1. Open Docker Desktop.
2. Example => For nginx image, Pull the nginx image from <https://hub.docker.com/>
3. Open Command prompt and execute the below command.

A screenshot of a computer program

Description automatically generated

1. Lates downloaded pulled images we are now able to see in Docker Desktop.



1. Run the below command for container which contains inside images (Publish Code) .

i.e, container contains the running instance of image.



Where nginx is the Web Server of Linux, same as IIS Web Server of Windows

**--name nginx =>** --name flag is for specify container name and nginx is the Name of the container.

**-p =>** Flag for port

**8080 =>** Port on local machine

**80 =>** Port on Web Server (By default port on web server is 80)

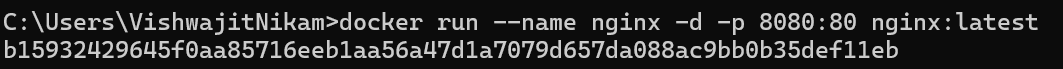
1. Now the container is running on container section of Docker Desktop

A screenshot of a computer

Description automatically generated

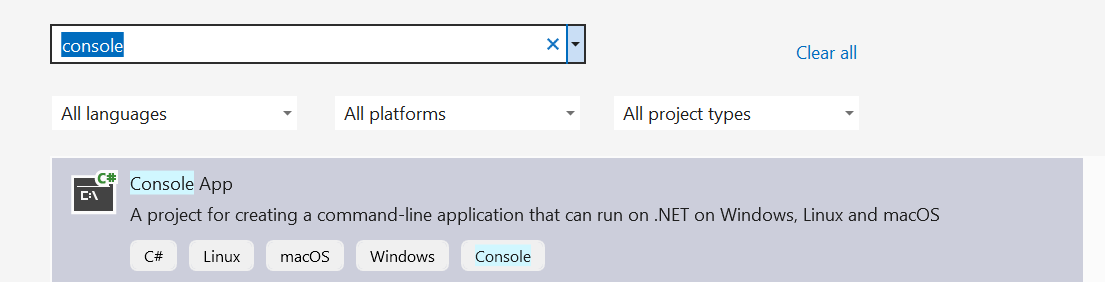
1. We can run the container using detach mode (-d flag used). It creates the container but will not attach to terminal of the running container, which doesn’t show any shell (Logs) in the console. But is show the shell in docker desktop behind the scene.

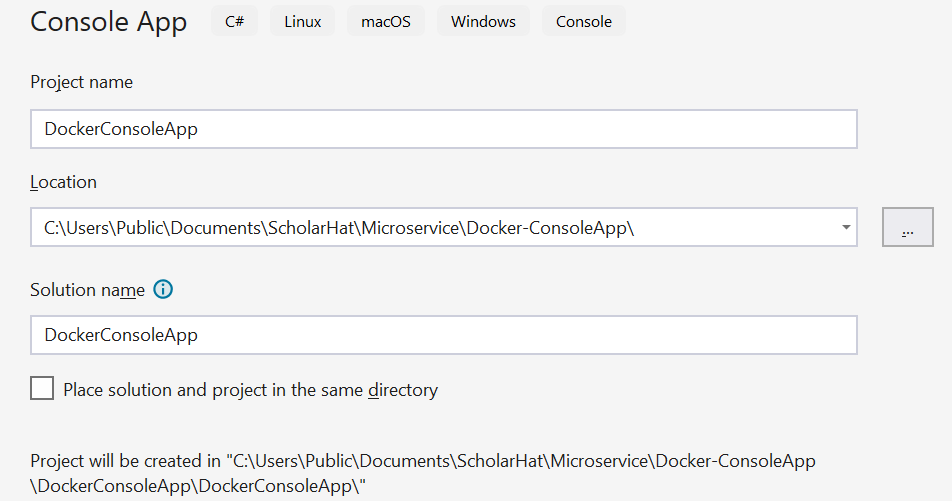
So, delete the existing container which running the inginx image from docker desktop.



# **2. SECTION – 2 : Create Image for Console Application**

1. Create Project 🡪 Select the console application template

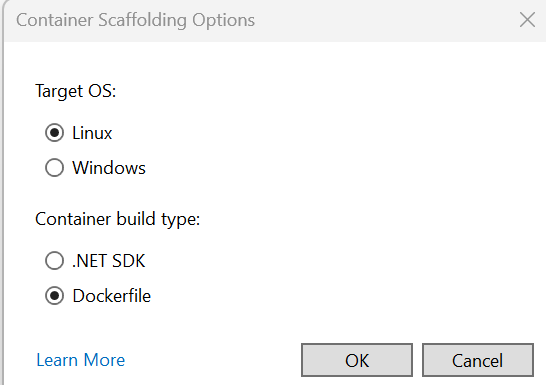




A screenshot of a computer

Description automatically generated

1. Right click on ConsoleApp 🡪 Add 🡪 Docker Support 🡪 choose the below options



1. In Docker file mention the below commands.

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS Build

WORKDIR /app

COPY . .

ENTRYPOINT ["dotnet", "run"]

Where 1st line is base image from Microsoft Container Registry

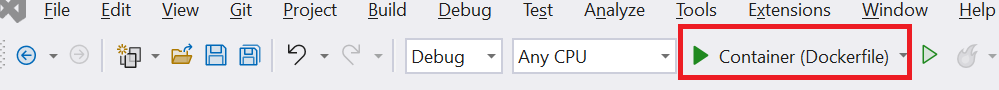
2nd line is Working directory where application code will copy

3rd line of first dot (.) => Current directory folder structure

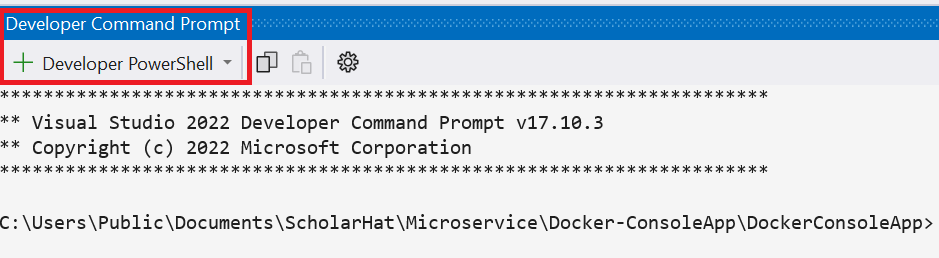
And second dot (.) is the destination folder where the all the files will copy. Here files will copy in app folder working directory.

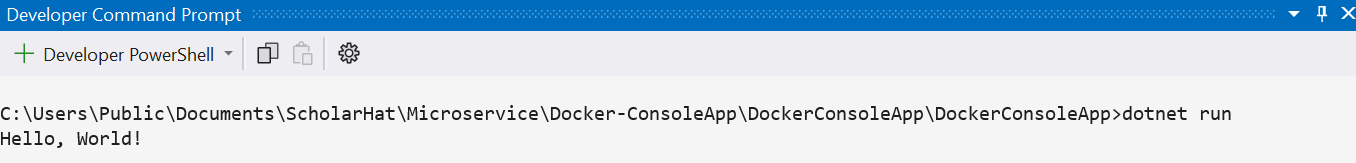
4th line => Entry point to execute the application.

We can run the Docker file using VS



Run the Docker file using Developer Command Prompt ( Right Click on Project 🡪 Open in Terminal and Select the Developer Command Prompt option from Developer PowerShell





Visual Studio by default create the image. We can check image and container in Docker Desktop as well.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

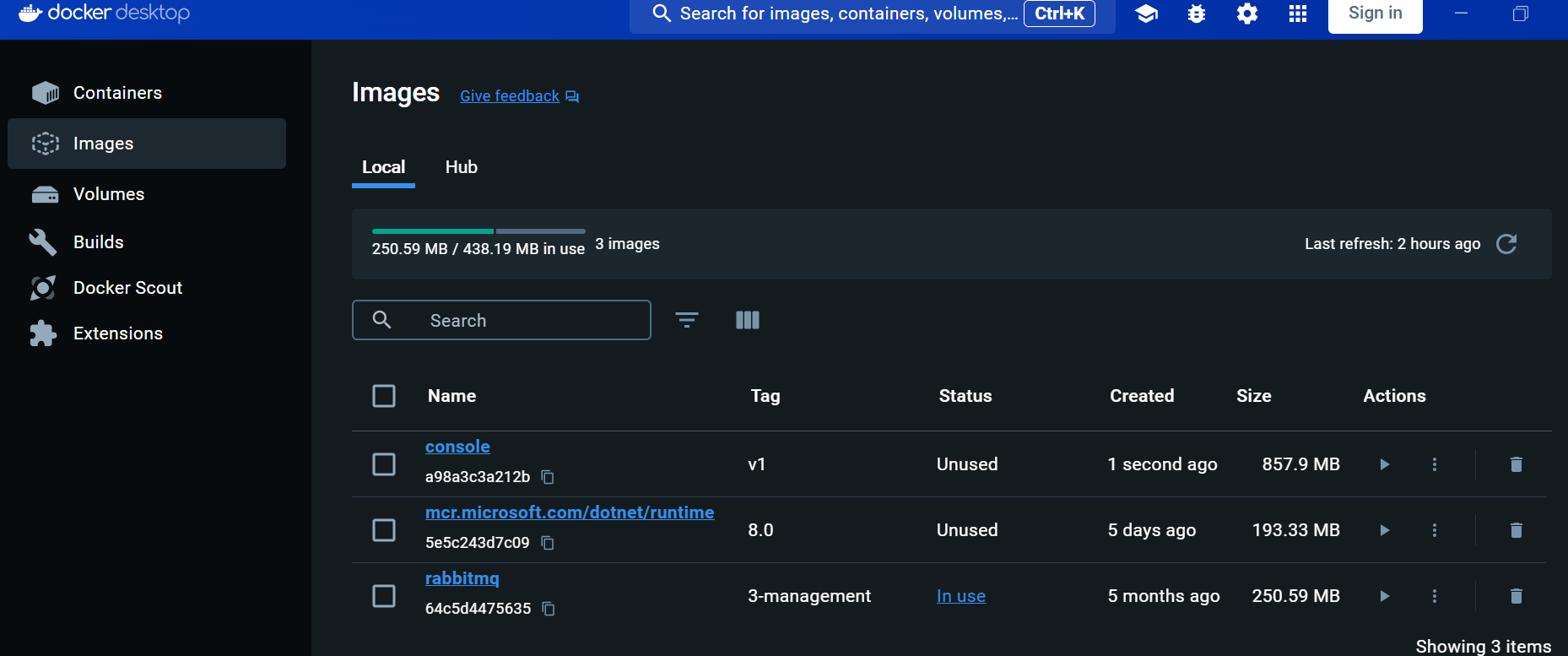
Delete the image and container from Docker Desktop. We can create (build) image and container our self.

**docker -t <ImageName>: version\_number <current working directory path i.e .>**

We can create the image as below command.



The image (console) is created as below screenshot



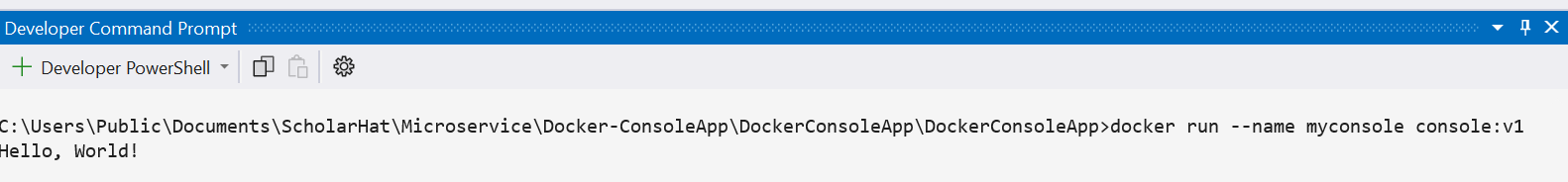
To see the list of image use below command

**docker image ls**

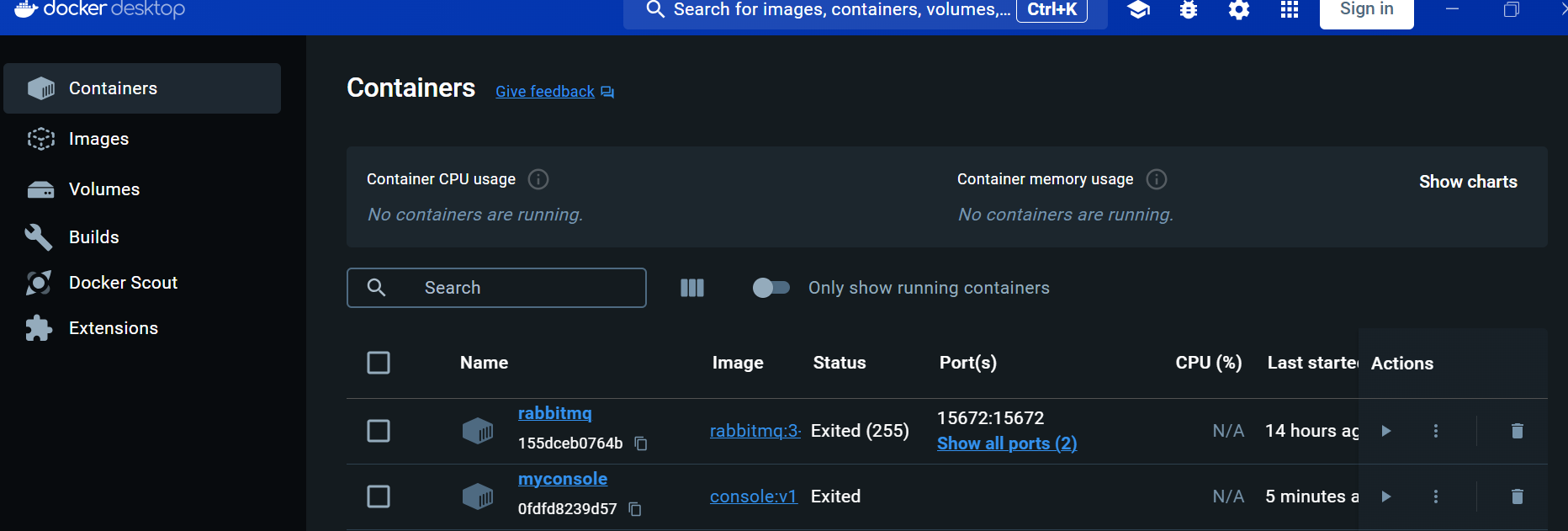
Now run the images through container by using below command

**docker run - - name <container-name> <image-name>: <version>**

**Note:** Console application don’t have port, so not mentioned the port.



Running container can see in Docker Desktop as below. (Container is a running instance of an image)



**Note :** We can create multiple container for the same image.

# **3. SECTION – 2.1 : Create Image for .Net Core Application**

1. Create New Project 🡪 Select ASP.NET Core Web App (Model-View-Controller) template.

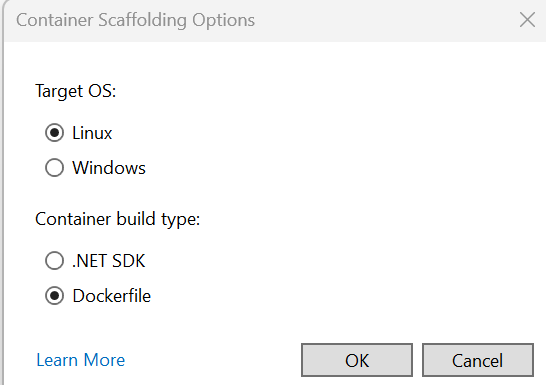
A screenshot of a computer

Description automatically generated

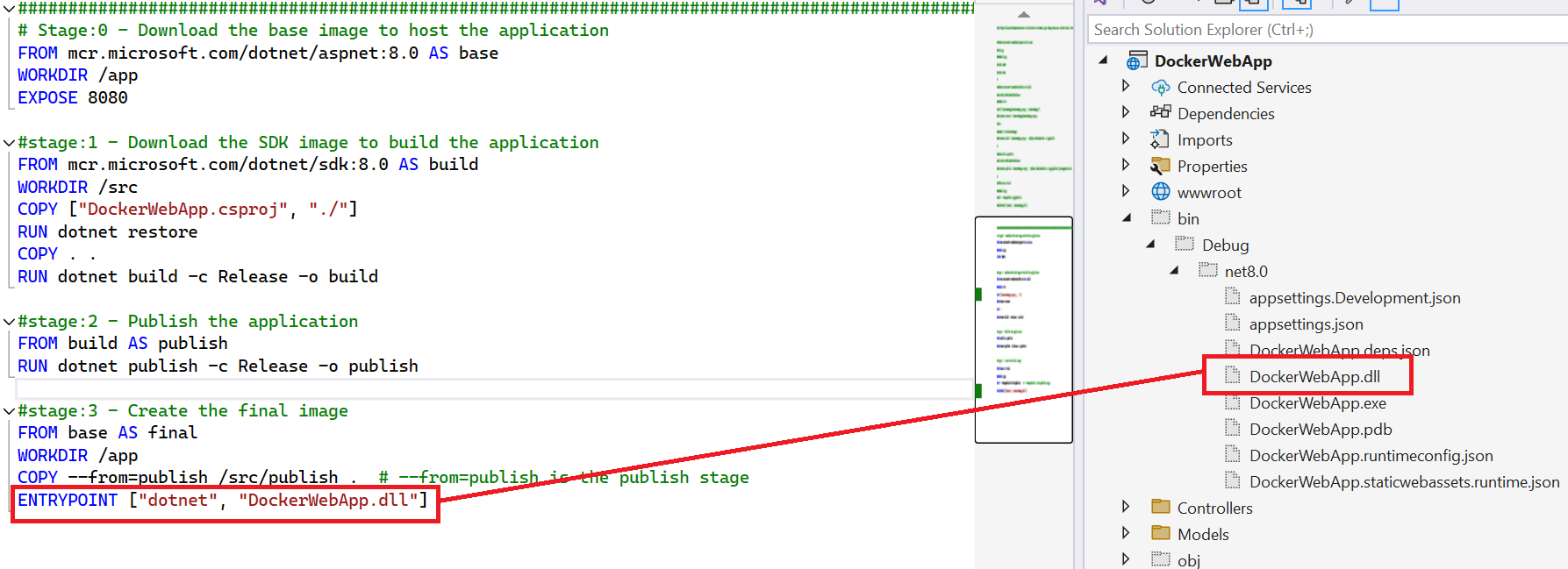
A screenshot of a computer

Description automatically generated

1. Right click on Project 🡪 Add 🡪 Docker Support 🡪 select the below option is will create the Docker file with default commands



Replace the default commands with the below command.



1. Right click on Project 🡪 Open in Terminal 🡪 Select the Developer Command Prompt option from Developer PowerShell

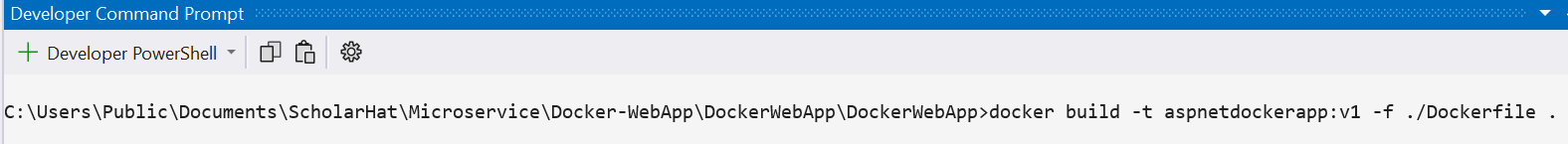
If the docker file in the same location, then use below command.

**docker -t <ImageName>: version\_number <current working directory path i.e .>**

**or**

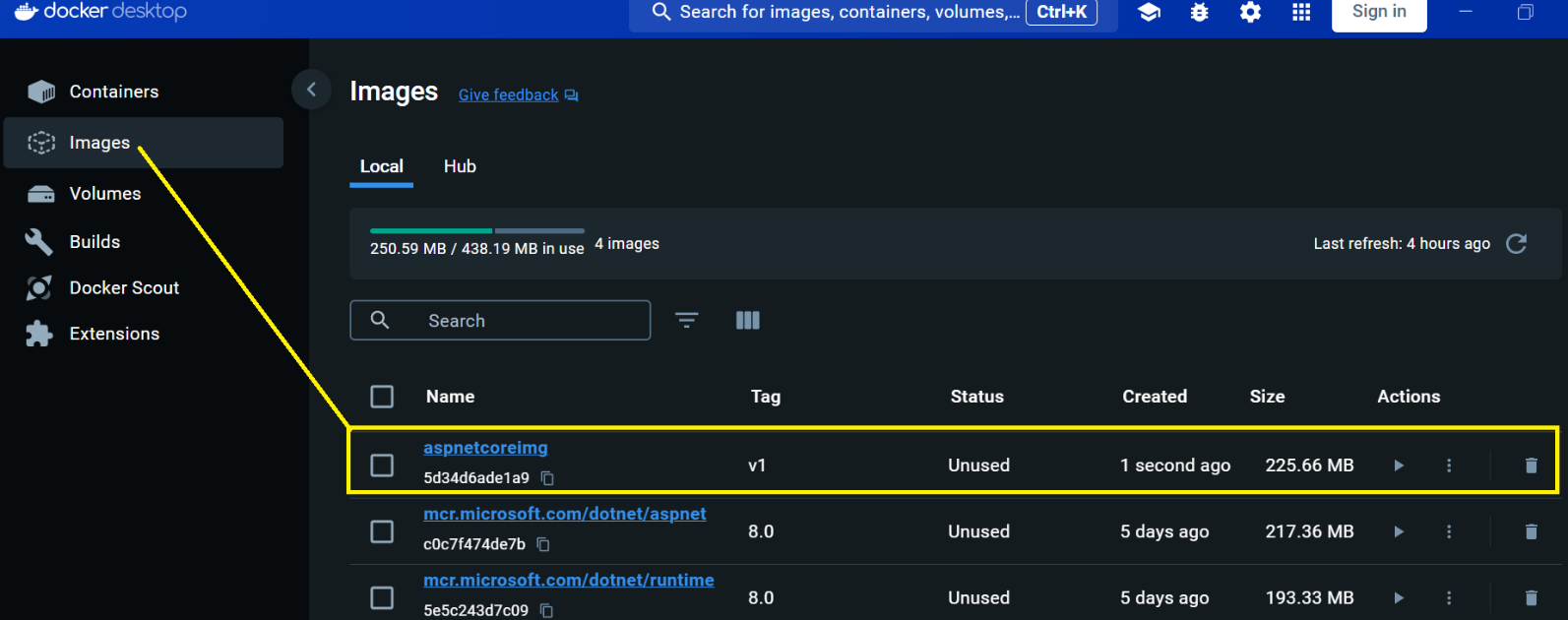
(If docker file is in some other location, then specify the path as follows) Build the image from some other location where we don’t have docker file.

**docker -t <ImageName>: version\_number -f ./Dockerfile <current working directory path i.e .>**

****

Here simply used the below command





1. Run the below command for container run

