**Lessson17 Containerizing a .Net app**

**Notes: -**



**1-we docker file you can create custom docker images which docker file represent sequence of commands to tell how to create custom image**

**2-docker file contains the following**

**A-base docker image which is represetn the Asp.net core SDK v3**

**(each container has its own file system , and own virtual network interface (has its completed packaged solution that working at runtime) (just like the VM))**

**B-we declare the working directory from where the custom docker image reading from**

**C-we copy the content of the publish folder into the working directory**

**D-we initalize the port at 5000**

**E-we set the dll who will be executable**

**(so in another word with docker file you can configure the port no plus the dll it execute and the base image and working directory that represent the path of the publish folder to publish your project via the docker container)**

**Steps:-**

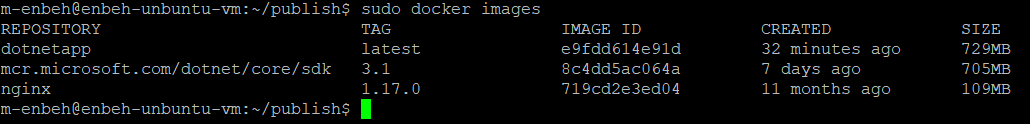
**1-we have the docker file and paste it inside the pbulish folder**

**2-we execute the following command on PUTTY command line**

**//to create custom image called dotnetapp**

**sudo docker build -t dotnetapp .**

**//we see that there is built in custom docker SDK 3.1 and nginx for hosting and the //custom docker image which depend on the SDK 3.1 on executing**



**//to run this custom image under port 5000 and uploaded to docker hub**

**sudo docker run -d -p 5000:5000 dotnetapp**

**(so in the previous lesson , we can say all steps is requried except that with this docker file you can create docker images and publish it seperated with demand of the base image docker which is .net core SDK 3.1)**

**sudo docker ps //it will show all running custom images**



**sudo docker start f2**

**//it will start the custom image (can write the first characters of his ID)**

**sudo docker stop f2**

**//it will stop the custom image (can write the first characters of his ID)**