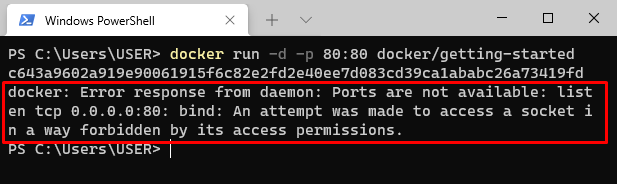
**Docker Documentation**

Getting started:

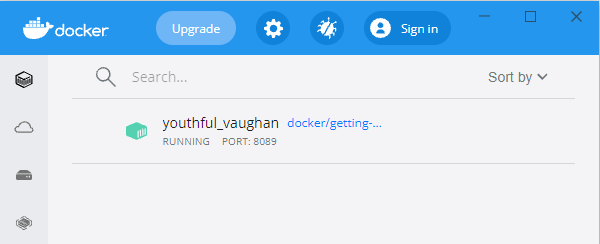
* Download and install docker desktop from [Docker.com](https://www.docker.com/)
* Test application
* Run this command on PowerShell

$ docker run -d -p 80:80 docker/getting-started

* -d - run the container in detached mode (in the background)
* -p 80:80 - map port 80 of the host to port 80 in the container
* docker/getting-started - the image to use

If it shows error like this, please change host port and try again.

* See the container list from Docker desktop



* Open your browser and try to access [http://localhost:8089](http://localhost:8089/)
* Change the port [8089] according to what you have entered

Networks:

* Useful command for networks
* List of networks

$ docker network ls

* Create new network

$ docker network create -d bridge mynet

* + -d - run the network in detached mode (in the background)
  + bridge – Network driver
  + mynet – Network name
* Network details

$ docker network inspect mynet

* + mynet – Network name
* Remove network

$ docker network rm mynet

* + mynet – Network name

Volumes:

* Useful command for Volumes
* List of volumes

$ docker volume ls

* Create new volumes

$ docker volume create myvol

* + myvol – Volume name
* Volumes details

$ docker volume inspect myvol

* + myvol – Volume name
* Remove volumes

$ docker volume rm myvol

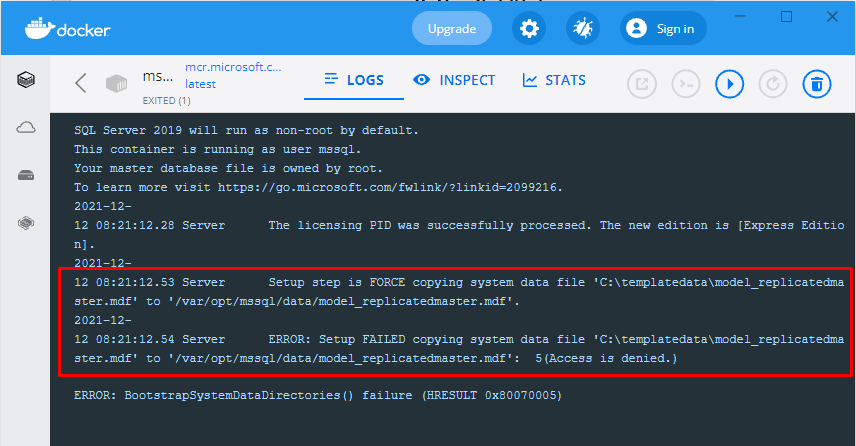
* + myvol – Volume name

Create mssql express container:

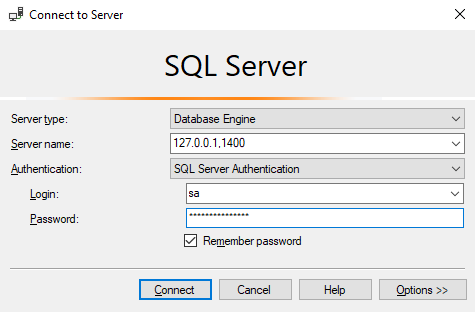
* Open PowerShell and run this command for create mssql container

$ docker run --name=mssqlexpres --net mynet -v mssqlvol:/var/opt/mssql/data -e "ACCEPT\_EULA=Y" -e "SA\_PASSWORD=express!2019" -e "MSSQL\_PID=Express" -p 1401:1433 -d mcr.microsoft.com/mssql/server:2019-latest

* --name – Container name
* mssqlvol – is volume name if this volume doesn’t exist docker will create one
* --e – Set the environ variables for mssql
* --p – [1st one is host computer port] : [2nd one is container port], for mssql set the container post 1433
* -d - run the network in detached mode (in the background)
* “mcr.microsoft.com/mssql/server:2019-latest” – is the image name for mssql
* If error throw like this add new property on command “--user root”, it will run mssql as root user.



* Now you can connect this mssql container from SQL Server Management Studio like this



* Add this mssql connection in appsettings like this

Data Source=127.0.0.1,1400; Initial Catalog=DB\_NAME; User Id=sa; Password=express!2019; Integrated Security=false;

Create image from Dockerfile for Asp.net core publish file:

* Create a file named “Dockerfile” without extension in the root folder of publish files.
* This file will like this

FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS base

COPY . app/source

WORKDIR /app/source

ENTRYPOINT ["dotnet", "DETS.API.dll"]

* “mcr.microsoft.com/dotnet/aspnet:6.0” – change this according to your dotnet version.
* “app/source” is where the files are copy in container
* WORKDIR must be same as copy directory.
* “DETS.API.dll” is your startup project dll name.
* Open PowerShell in root folder of publish files and run this command for create new image from publish files.
* Before build check EXPOSE port on Dockerfile, expose port and container port must be same

$ docker build -t detsapi .

* + detsapi – Image name
* For show create image list

$ docker images

Create a container from newly created image:

* Open PowerShell and run this command for create new container based on previously created image

$ docker run --rm --name dets -v detsapivol:/usr/src/app --net mynet -p 3000:3000 detsapi

* + --rm – If you want to delete this container automatically after container stopped set this, otherwise not.
  + detsapivol – is volume name if this volume doesn’t exist docker will create one

Deploy .net project with mssql server in docker using docker compose:

* Create a file named “docker-compose.yml” in root folder of publish files.
* This file will like this

version: "3.9"

services:

    web:

        container\_name: web

        image: webimage

        build: .

        ports:

            - "8000:80"

        networks:

            - webnet

        volumes:

            - webvol:/app/source

        depends\_on:

            - db

    db:

        container\_name: db

        image: "mcr.microsoft.com/mssql/server:2019-latest"

        user: root

        environment:

            SA\_PASSWORD: "express!2019"

            ACCEPT\_EULA: "Y"

            MSSQL\_PID: "Express"

        ports:

            - "1400:1433"

        networks:

            - webnet

        volumes:

            - sqlsevervol:/var/opt/mssql/data

networks:

    webnet:

        name: webnet

        driver: bridge

volumes:

    webvol:

        name: webvol

    sqlsevervol:

        name: sqlsevervol

* Update appsettings like

Server=db; Database=DB\_NAME; User=sa; Password=express!2019;

* + db – is the mssql docker container name
* Open PowerShell in root folder of publish files and run these commands
* $ docker-compose build
* It will build the images on docker
* $ docker-compose up
* It will build the networks, volumes and containers according to docker-compose file on docker
* $ docker-compose down
* It will remove networks and container created previously
* For use host computer directory as volume, change docker-compose file like this

version: "3.9"

services:

    web:

        container\_name: web

        image: webimage

        build: .

        ports:

            - "8000:80"

        networks:

            - webnet

        volumes:

            - E:/Docker/data/web:/app/source

        depends\_on:

            - db

    db:

        container\_name: db

        image: "mcr.microsoft.com/mssql/server:2019-latest"

user: root

        environment:

            SA\_PASSWORD: "express!2019"

            ACCEPT\_EULA: "Y"

            MSSQL\_PID: "Express"

        ports:

            - "1400:1433"

        networks:

            - webnet

        volumes:

            - E:/Docker/data/db:/var/opt/mssql/data

networks:

    webnet:

        name: webnet

        driver: bridge

* Use a local directory path as volume name and remove volume section from docker-compose file.
* Copy and paste the publish files with Dockerfile and docker-compose.yml files in volume host directory in this case it is: “E:/Docker/data/web”.
* Open PowerShell in host directory and run command like previous.