**Docker Concepts that I tried to cover initially.** [**3 min read** ]  
  
 Reference:

* From docker site docs & guide
* From Several tutorials
* From Blog and Instruction manual
* I had prior knowledge about Docker File – Deployment – Image – Desktop Docker etc

**Exposure of work around / learning:**

1. **What is Docker**, why it is to say simplify building, running & shipping apps, what are the **benefits** that developers attracted to docker.
2. **Role** of **images**, **Container**. [Little bit elaborate]
   1. Like Image is a read only template composed of layered file system to create container.
   2. Container instances are created from images that can be run, stopped, moved, deleted.
3. **Where** does **docker run**
   1. like upon Windows server 2016 Docker Daemon can collaborate with client.
   2. Brief of dockers benefit and varieties.
4. Knowledge about **Docker Tools**. (As Docker Toolbox is legacy, I tried to cover the topics around **Docker Desktop** (run on Windows 10+)
   1. Docker Desktop provides Image, container tools, Hyper-V to run VM etc in brief
   2. try to research around and understand.
   3. I write a document on Docker for my further learning.
5. **Docker Desktop Tools** in brief & elaborate
   1. **Docker Client**:
      1. Interact with Docker Engine
      2. Build & manage Image
      3. Run and Manage Containers
6. **Installed Docker Desktop** in my Windows OS. Tried the **initial command** to play around
   1. - docker pull [Image]
   2. - docker run [image]
   3. - docker ps
   4. - docker ps –a
   5. - docker stop [ContainerID]
   6. - docker rmi [ImageID]

\* **Skiped docker machine** seems docker machine is not for Docker Desktop.

8. **Docker Compose**

9. **Docker Kitematic** :

1. GUI visual way to handle image & container
2. Visually search for Docker Image.
3. Actually, behind the scenes it works with commands, but the user can interact in a visual way.
4. have installed and tried to see actions like stop, run, logs, delete, exec etc.

10. **Tried to Hooking source code into container**

1. **Creating a container** volume that points to the source code.
2. Add source code into a **custom Image** that is used to create a container

11. Learned briefly about **Layered file system**. A brief work around about Image, Container and File Layers  
  
12. About **Docker Volumes**

a. like Special type of directory mentioned Data Volume

b. Can share or reuse by Containers

c. how Data volume are persistent etc

13. **Tutorials** and work around like

a. **Creating Data volume** (like Docker run –p 8080:3000 createvolume/container volume

14. **Locating** volume or **Customizing** volumes  
15. **Hooking a volume to ASP.NET Core Source Code**  
16. **Removing** Containers and volumes

17. How to hook Source Code into Container

a. Create a container volume that points to the source code

b. Add Your source code into a custom Image that is used to create the container

18. **Understanding with Visual Studio**

a. Adding New Project Docker support

b. Existing project D**ocker support**.

19. **Understanding the Docker File**

|  |  |  |
| --- | --- | --- |
| **From** | **Label** | **RUN** |
| **Copy** | **ENTRYPOINT** | **WORKDIR** |
| **Expose** | **ENV** | **VOLUME** |

20. **Trying with Building a Custom Image**

Like Docker build –t (short for tag) <username>/aspnetcore . (for build context)

21. **Publishing** an Image to **Docker hub**

Like Docker push <your username>/node

**NOTE**: By the time you sort out with other things. I will be continuing (But will not log time before next week)