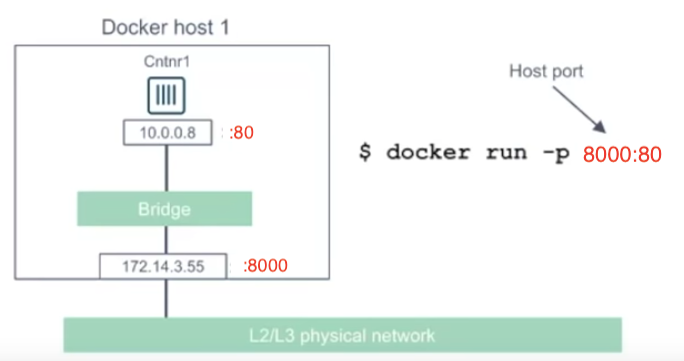
Sharing ports and disk

Mapping ports

Default behaviour of docker container Docker container run it own environment have its own ports, network config, disk space etc. We can link, share ports, disk etc between docker and host machine so that we can access docker container resources in or through host machine.

Ex. If docker container is running webserver and serving index.html file on port 80. If we have to access that file from host machine then we can map host machines port 8080 to dockers port 80 so that all the requests coming on hosts machine port 8080 will be forwarded to dockers port 80

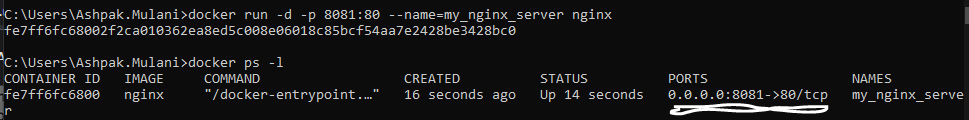
To map ports, we can use -p 8080:80 option. This means port of host machine 8080 mapped to docker contains port 80



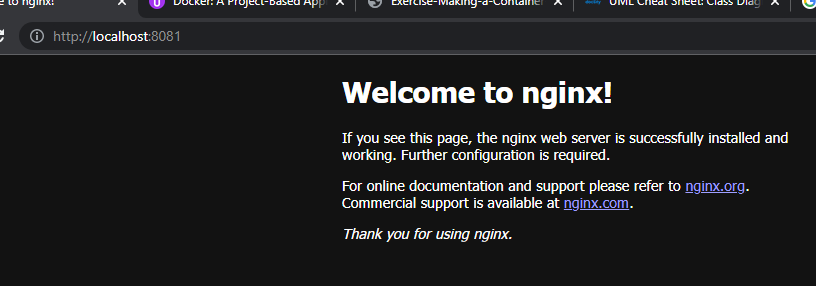
Full command will look like this if we have to run nginx server and map ports of 8081 of host with 80 of docker container.

docker run -d -p 8081:80 --name=my\_nginx\_server nginx

if we check docker details using ps -l command then we can see the ports mapping in returned results as shown below .



Dock now if we hist localhost:8081 in browser it will eventually going to call docker containers 80 port and request for its contents. On same line if we hit localhost:8081 in browser we get following results from nginx server.



# Mapping disk folder

Like network we can also map disk folders between host and docker container

In below command we are running a container, mapping port and also creating an virtual drive/mapped folder inside container using **-v** which maps ‘/user/share/nginx/html’ location from docker container to <<current dir>>/webpage directory on host machine. **:ro** option says its read only means docker container cannot modify contents present in virtual drive.

docker run -p 8082:80 -d --name=my\_nginx\_container -v %cd%/webpage:/usr/share/nginx/html:ro nginx

on local host we have created our own index.html file so when we visit <http://localhost:8082> then it’s going to show contents from our custom from <<current dir>>/webpage/index.html file rather than original nginx index.html

