#commiting

Saylani A.I Batch-I Docker Class  
Dated:17-Dec-17  
Note (Bricket use for description commands)  
======Docker Installation  
First up, pre-requisites. Docker for Windows requires:  
Windows 10 Pro | Enterprise | Education (1511 November update, Build  
10586 or later)  
Must be 64-bit  
The Hyper-V and Containers features must be enabled in Windows  
Hardware virtualization support must be enabled in your system’s BIOS  
  
======Check docker installed  
go to powerShall  
1) docker version  
2) docker image ls OR docker images  
  
======Install New   
1) docker image pull alpine:latest  
2) docker image ls  
  
======Run Container  
1) docker run -it --name TASK1 alpine:latest (--name Task1 custome name for container without this attribute also run container)  
->ls  
->ps  
->touch abc.txt  
->vi abc.txt (Now press insertButton and write any text for exit esc->:x->enter)  
2) control+q+p (escape without terminate the runing container)  
3) docker container ls  
  
4) docker run -it --name NewContainerName alpine:latest /bash/sh   
5) contrl+Q+P  
6) docker container ls  
7) docker container exec -it TASK1  
->ls  
->vi abc.txt  
->exit (exit with terminate container)  
8) docker container ls -a (Show all created containers)  
9) docker contaienr stop ContainerName  
  
10) docker container rm ContainerName (before this command you have to stop container)  
 docker contaienr rm ContainerName -f (-f force if contaienr runing)  
 docker container rm ContainerId  
  
======Build Image  
Download code  
The Linux app can be located from: <https://github.com/nigelpoulton/psweb.git>  
  
$ git clone <https://github.com/nigelpoulton/psweb.git>  
$ cd psweb  
$ ls -l  
$ $ cat Dockerfile  
FROM alpine  
LABEL maintainer="nigelpoulton@hotmail.com"  
RUN apk add --update nodejs nodejs-npm  
COPY . /src  
WORKDIR /src  
RUN npm install  
EXPOSE 8080  
ENTRYPOINT ["node", "./app.js"]  
  
==>create Image command with Dockerfile   
1) docker image build -t NewImageName:latest .  
2) docker images  
3) docker container run -d --name web1 -p 8080:8080 NewImageName:latest  
  
Contratulation you have successfully build your own image and also run container.. Wow  
  
======Push Image on Container Registery (hub.docker.com)  
<https://hub.docker.com/>  
Sign Up  
Create New Repository  
  
1) docker login  
 UserName :  
 Password  
 successfully login  
  
1) docker tag NewImageName:latest   
2) docker image ls  
3) docker push UserName/NewImageName:latest  
Woowwww.... Again you have Created and Push your image on Remote server now widely all over the world every one can use your image.  
  
4) docker image pull UserIdName/NewImageName:latest  
  
======Map your Host directory with container directory  
  
1) docker container run -it -v d:\demo1:/hello1 --name xyz alpine:latest  
  
(Now we have created demo1 folder in D:/ drive where we will keep which file that we have to run in container enviornment)  
stop all containers:  
docker kill $(docker ps -q)  
  
remove all containers  
docker rm $(docker ps -a -q)  
  
remove all docker images  
docker rmi $(docker images -q)