Dockerfile Components

Steps: Create a container
: update the data make them into image
: new/from

To find difference,
docker diff <container name>

To create container of image,
docker commit new container update image
name of container name of image

DOCKERFILE

- A text file which contains set of instruction
- Automation of docker image creation

COMPONENTS

FROM- Top command, it tells base image
RUN - To execute
MAINTAINER - Author/owner
COPY - Copy from local host
ADD - Copy from internet also extract
EXPOSE - To expose port number
WORKDIR - To set working directory for container
CMD - execute command during creation of container
ENTRYPOINT - high priority than CMD
ENV - Environment variable.
ARG -

To create image from Dockerfile
docker build -t myimage .

Docker Volume

- Volume can be shared.
- A directory inside our container
- Even if it container is stopped its accesible.
- Once you create image of a container and from that image create another container Then old container is not sharing with new container.

* Create Dockerfile
  FROM ubuntu
  VOLUME ["/myvolume"]

* To create image from Dockerfile
  docker build -t myimage .

* Share vol with another Container
  docker run -it --name container2 --privileged=true
  --volume from container1 ubuntu /bin/bash

* Now try to create volume by using commad.
  docker run -it --name container3 -v volume2 ubuntu /bin/bash

To share,
docker run it --name container4 --privileged=true
--volume=from container3 ubuntu /bin/bash

Extra Commands,

docker volume	ls	
//	create	<volume name>
//	rm	
//	prune	(removed unused docker vo
//	inspect	

docker container inspect <container name>

Docker Port mapping & Expose

1) docker run -itd --name <name of container> -p 80:80 <image name>
   port "container

Though expose only internal connection btw containers can take place but though -P it can be done globally

To check if a port is expose on container,
docker port <container name>

docker exec -it <container name> /bin/bash

- -P has priority over expose

Docker Hub

- Create an image from container then publish into dockerhub.
- other person will pull that image and create container from it.

* To make image of a container
docker commit <container id> image id

To login into dockerhub account use,
- docker login

1) To stop all running containers: docker stop $(docker ps -a -q)
2) delete all stopped containers: docker rm "
3) delete all image: docker rm -f $(docker images -q)