docker –version => version check of installed docker

docker images => show all images

docker ps -a => all containers

docker run hello-world => if hello-world image is not found in my local , this command pull hello- world image pull docker hub and make it a container and run it

docker build -t name of image . => create an image based on docker file (-t to tag /named the image (can give name I want) , .(dot) for directory of dockerfile )

docker run --name hello-docker-container -p 8080:80 hello-docker:1.0.0

Give container name for a docker image

Image name cmnd

Docker run cmnd for a image

Define port cmnd. Where 8080 in our local machine and 80 is container port. Listing fro 80 and maped to 8080

docker stop 78e

Container Id

Docker stop cmnd for a container

docker start 78e

Container Id

Docker start cmnd for a container

docker restart 78e

Container Id

Docker restart cmnd for a container

docker rm 1be = > remove one container

Container Id

docker rm 1be -F = > remove one container by force when it is running

docker ps -a -q => cmnd for get all container id

docker rm $(docker ps -a -q) => remove all container

docker rmi 168 => remove one image

image Id

docker images -q => get all images id

docker rmi $(docker images -q) => remove all images

docker exec -it a8c bash/sh => execute bash or shell into a container with interactive terminal mode

* - it =>for interactive terminal mode
* a8c => last 3-character container id

docker inspect 932 => for inspecting container information

Container Id

docker kill 932 => for stopping/killing container instant by force

Container Id

docker kill $(docker ps -q) => kill all containers

**docker buid command with options:**

**docker build -f D:\Self\Programming\Docker\dockerfile --label "zakaria'sImage" --label "Builtfors3innovate" -q --rm=false -t new-image:latest -t new-image:1.0.0 -t hello-world:lastest .**

* **-f D:\Self\Programming\Docker\dockerfile => file location of docker file**
* **--label "zakaria'sImage" --label "Builtfors3innovate" => Set metadata for an image**
* **-q => Suppress the build output and print image ID on success**
* **--rm=false => Remove intermediate containers after a successful build (by default true)**
* **-t new-image:latest -t new-image:1.0.0 -t hello-world:lastest => Name and optionally a tag in the 'name:tag' format**
* **. (dot)** => **directory of dockerfile**

**docker run command with options:**

**docker run -p 4080:80 -p 4443:443 -v D:\Self\Programming\Docker\vol:/root/docker -e TestVar=1234 --name=new-container** **--rm -h myhostname --label “this is zakaria’s container” new-image:latest**

* **-p 4080:80 -p 4443:443** => Publish a container's port(s) to the host (80 => http, 433 => https)
* **-v D:\Self\Programming\Docker\vol:/root/docker =>** Bind mount a volume
* **-e TestVar=1234 =>** Set environment variables
* **--name=new-container =>** Assign a name to the container
* **-d =>** Run container in background and print container ID
* **-i =>** Keep STDIN open even if not attached
* **-t =>** Allocate a pseudo-TTY
* **--restart =>** Restart policy to apply when a container exits (default no)
  + **–restart=on-failure:5 (container try to restart 5 times when something unexpected/unhandled things occurred )**
  + **–restart=unless-stoped (container)**
  + **–restart = awalys**
* **--rm =>** Automatically remove the container when it exits
* **-h =myhostname =>** Container host name
* **--label “This is zakaria’s container” =>** Set meta data on a container
* **-P =>** Publish all exposed ports to random ports
* **new-image:latest =>** image name

**docker volume command with options:**

docker volume ls => for list of local volumes

docker volume inspect 0c9e6c74e525a6cfb8e55556d0e6d6e8a52236c4f6555eb667a69490be921e7d

=> list of local volumes

volume name

docker volume rm 0c9e6c74e525a6cfb8e55556d0e6d6e8a52236c4f6555eb667a69490be921e7d

=> for removing volume (one/more)

volume name

docker volume prune => remove all unused local volumes