docker useful commands

```
what are Docker Images:
=======================
An image is a package that consists of an application and all of its
dependencies to run the application in a container.
containers and images use a layered file system. Each layer consists
only the differences from the previous layer.
the image consists of one or more read only layers, while container
adds on one writable layer on top of image layers.


The layered file system allows multiple images and containers to share
the same layers.
this would result in:
  smaller overall storage footprint
  faster image transfer
  faster image build

how to build docker image
=========================
two ways
      1) MANAUL
      2) AUTOMATED

   Manual Process
   ==============
     choose a base image
     run it as cotnainer in interactive mode
     make neccessary changes inside the container
     come out of container safely ( ctrl pq )
     freeze the changes made inside container to covert it as image (
use docker commit command )
     ex:
        docker pull ubuntu
        docker run -it ubuntu bash
        < inside contianer changes >
             apt-get update
             apt-get install -y nginx
             apt-get install -y vim
             mkdir /home/configuration
             touch /home/configuration/db.props
             vi /var/www/html/index.html ( edit & save file )
             ctrl pq ( to comeout of contianer safely )

      docker commit -m "install nginx" -c 'CMD /usr/sbin/nginx -g
"daemon off;"' -c 'EXPOSE 80' <contid> <new image name>

      while commiting the changes made to a container below are
```

```
mandatory
          -m -- a message what changes are made
          CMD -- a command that would start a process inside the
container
          EXPOSE -- port number on which process inside container runs
always
          cont id -- in which the changes are made will have to be
freezed / saved
          new image name -- anyname for your new image ( ex: mynginx )

   Automated process ( real time practice )
   =========================================
      create a simple text file & write the all the instructions build
an image

      vi mydockerfile
          FROM ubuntu
          RUN apt-get udpate
          RUN apt-get install -y nginx vim
          RUN mkdir /home/config/
          RUN touch /home/config/db.props
          EXPOSE 80
          CMD /usr/sbin/nginx -g "daemon off;"
      save&quit

      docker build -f /path/to/mydockerfile -t <new image name> .
(context "." (currnet directory))


how to push docker images into docker hub
==========================================
        create an account in hub.docker.com
                create a reportsitory after logged into the account
(ex: myapp)

                on docker host
                docker login
                        username: docker hub username
                        password: docker hub passwd
                ensure you get "Login Succeeded" message

                docker tag local-image:tagname new-repo:tagname
                    docker tag myapp:v1 lerndevops/myapp:v1
                docker push new-repo:tagname
                     docker push lermdevops/myapp:v1


how to push images to private repo ---- DTR ( docker trusted registry )
=======================================================================
          goto mydockerrepo.com & create an acct
          inside the acct create a repo
          come back to the server & login to the private repo ( docker
login mydockerrepo.com ) provide uid/pwd
          docker tag local-image:tagname mydockerrepo.com/acctname/repo:
tagname
                    docker tag myapp-mytomcat:v1 mydockerrepo.com
/lerndevops/myapp:v1
          docker push mydockerrepo.com/acctname/repo:tagname
                     docker push mydockerrepo.com/lerndevops/myapp:v1


how to push images to private repo ( registry container by docker )
===================================================================
          docker run -d -p 5000:5000 --restart always --name registry
registry:2
          local registry container address -- localhost:5000
          docker tag local-image:tagname new-repo:tagname
                    docker tag myapp-mytomcat:v1 localhost:5000/myapp:v1
          docker push new-repo:tagname
                     docker push localhost:5000/myapp:v1


how to push images offline ( docker save & docker load )
========================================================
    docker save -o mycentos.tgz mycentos:v1
    scp mycentos.tgz to target machine / server
    docker load < mycentos.tgz

Advanced image concepts
=======================
   find dangling images
       docker images -f dangling=true
       docker image prune --dangling=true

   Inspect image metadata:
       docker image inspect nginx:1.14.0
       docker image inspect nginx:1.14.0 --format "{{.Architecture}}"
       docker image inspect nginx:1.14.0 --format "{{.Architecture}} {{.
Os}}"
```