1. how to search a docker image in hub.docker.com

docker search httpd

1. Download a docker image from hub.docker.com

docker image pull <image\_name>:<image\_version/tag>

1. List out docker images from your local system

docker image ls

1. Create/run/start a docker container from image

docker run -d --name <container\_Name> <image\_name>:<image\_version/tag>

d - run your container in back ground (detached)

1. Expose your application to host server

docker run -d -p <host\_port>:<container\_port> --name <container\_Name> <image\_name>:<Image\_version/tag>

docker run -d --name httpd\_server -p 8080:80 httpd:2.2

1. List out running containers

docker ps

1. List out all docker container (running, stpooed, terminated, etc...)

docker ps -a

1. run a OS based container which interactive mode (nothing but login to container after it is up and running)

docker run -i -t --name centos\_server centos:latest

i - interactive

t - Terminal

1. Stop a container

docker stop <container\_id>

1. Start a container which is in stopped or exit state

docker start <container\_id>

1. Remove a container

docker rm <container\_id>

1. login to a docker container

docker exec -it <container\_Name> /bin/bash

**Docker Hub**

Docker Hub is a cloud-based registry service which allows you to link to code repositories, build your images and test them, stores manually pushed images, and links to Docker Cloud so you can deploy images to your hosts. It provides a centralized resource for container image discovery, distribution and change management, user and team collaboration, and workflow automation throughout the development pipeline.

**Ref URL :**[**https://docs.docker.com/docker-hub/**](https://docs.docker.com/docker-hub/)

1. Create a docker hub account in <https://hub.docker.com/>
2. Pull a docker image

docker pull ubuntu

1. pull a docker image with the old version

docker pull ubuntu:16.04

1. create a custom tag to the docker image

docker tag ubuntu:latest valaxy/ubuntu:demo

1. login to your docker hub registry
2. docker login

docker push valaxy/ubuntu:demo

**testing**

1. Remove all images in docker server

docker image rm -f <Image\_id>

1. Pull your custom image from your docker account

docker pull valaxy/ubuntu:demo

DockerFile

1. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.
2. The docker build command builds an image from a Dockerfile and a context.

docker build .

docker build -f /path/to/a/Dockerfile .

docker build -t shykes/myapp .

1. The Docker daemon runs the instructions in the Dockerfile one-by-one, committing the result of each instruction to a new image if necessary, before finally outputting the ID of your new image.
2. Whenever possible, Docker uses a build-cache to accelerate the docker build process significantly.
3. When you’re done with your build, you’re ready to look into scanning your image with docker scan

docker scan hello-world

**Docker file creation**

Docker file format

# Comment

INSTRUCTION arguments

1. A Dockerfile must begin with a FROM instruction.
2. Docker distributes official versions of the images that can be used for building Dockerfiles under docker/dockerfile repository on Docker Hub.

* FROM - A valid Dockerfile must start with a FROM instruction.
* RUN - The RUN instruction will execute any commands in a new layer on top of the current image and commit the results. The resulting committed image will be used for the next step in the Dockerfile.
* CMD - The main purpose of a CMD is to provide defaults for an executing container.

Note: There can only be one CMD instruction in a Dockerfile. If you list more than one CMD then only the last CMD will take effect.

Note - Do not confuse RUN with CMD. RUN actually runs a command and commits the result; CMD does not execute anything at build time, but specifies the intended command for the image.

* LABEL -The LABEL instruction adds metadata to an image. A LABEL is a key-value pair. T
* EXPOSE
  1. The EXPOSE instruction informs Docker that the container listens on the specified network ports at runtime.
  2. To set up port redirection on the host system, see using the -P flag.
  3. The docker network command supports creating networks for communication among containers without the need to expose or publish specific ports, because the containers connected to the network can communicate with each other over any port.
* ENV - The ENV instruction sets the environment variable to the value .
* ADD - The ADD instruction copies new files, directories or remote file URLs from and adds them to the filesystem of the image at the path .

If is a local tar archive in a recognized compression format (identity, gzip, bzip2 or xz) then it is unpacked as a directory.

* COPY - The COPY instruction copies new files or directories from and adds them to the filesystem of the container at the path .
* ENTRYPOINT - An ENTRYPOINT allows you to configure a container that will run as an executable.
  1. ENTRYPOINT will override all elements specified using CMD
  2. Both CMD and ENTRYPOINT instructions define what command gets executed when running a container. There are few rules that describe their co-operation.
  3. Dockerfile should specify at least one of CMD or ENTRYPOINT commands.
  4. ENTRYPOINT should be defined when using the container as an executable.
  5. CMD should be used as a way of defining default arguments for an ENTRYPOINT command or for executing an ad-hoc command in a container.
  6. CMD will be overridden when running the container with alternative arguments.
* VOLUME - The VOLUME instruction creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.
* USER - The USER instruction sets the user name to use when running the image
* WORKDIR - The WORKDIR instruction sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD instructions
* ARG - The ARG instruction defines a variable that users can pass at build-time to the builder with the docker build command using the --build-arg = flag.
* SHELL - The SHELL instruction allows the default shell used for the shell form of commands to be overridd

Example

FROM ubuntu:12.04

MAINTAINER abc@gmail.com

LABEL version="1.1.0" \

app\_name="Training application"

RUN apt-get update && apt-get install -y apache2 && apt-get clean

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

EXPOSE 80

COPY index.html /var/www/html

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]