Docker Operation commands:

#Docker -version

#Docker login

Which will help us to login to Docker repository /registry(either from cloud/on prem) to pull images

#Docker search

Which will display images available in registry

#Docker pull <nginx>

Which will download the images from Docker hub registry

#Docker inspect image <image name> which will inspect image specified

#Docker images which will display all the images in local repository

#Docker create <image name> which will create container with default name and network and ports

#Docker start <container ID /Name> which will start/poweron the container

#Docker ps -a : which will display all containers along with exited and running

#Docker stop <container ID /Name> which will stop/poweroff the container gracefully

#Docker create --name <custom name of the container> <imagename> which will create contaner with custome name

#Docker rmi <imagename> whch will delete image specified however if the image is assoicated with any contaner , it will not delete

#Docker rm <container name/id) : which will delete container but not running container

#Docker rm <container name/id) --force : which will delete container forcefullyand it is not recomended

#Docker kill <container name/id> : which will stop container forcefully

#Docker run -d -p 8080:80/tcp --name <my-container> <myapp-image> :which would create container with custom name given and ports, protocals and run the container . -d is for detached mode. Without this it auto login and will not get out

#Docker exec -it <container name> /bin/bash : which will login to container specified "-it" is interactive terminal

Note: Container support very limited commands

#Docker inspect network bridge which will display bridge network details and containers connected to that network

#Docker inspect image <imagename> which will give more details about specific image and its version, layers and other details

Docker Network:

Custom Image Build:

Docker File: includes following entries to build custom image

From: to specify version of base image to pull from docker hub to build custom image during the build image

Add: to specify the command/source and destination path to copy the file/folders from URL /local path from the host. Also extract the zip/tar file from host to container during the image build process

Copy: to specify the command/source and destination path to copy the file/folders from local host to image during the build process

Cmd: is used to specify parameters/arguments required to run along with Entry point apps

EntryPoint: to specify the path of the programs/apps which should be started along with container

ENV: is to set the environment variables

Maintainer: is to specify author of the container like label

Expose: to specify the port on which app need to be run in container

Run: is to specify the command to update the container or any other commands during the image build

Volume: to mount the local directory into Container

User: which user to be allowed to login to container

WorkDirectory:

Example Dockerfile:

From ubuntu:18.04

Label maintainer="vijay.08123@gmail.com"

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

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apachectl"]

CMD ["-D", "FOREGROUND"]

From ubuntu:18.04

Label maintainer="vijay.08123@gmail.com"

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

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apachectl"]

CMD ["-D", "FOREGROUND"]

COPY index.html /var/www/html/index.html

VOLUME /var/www/html

#Docker build –t <vijay/Image Name > .

This command will build a custom image as per inputs given in dockerfile from current working directory

#Docker tag <image Name> <Image name:<tagname usually version> >: to add tag for the image

#Docker push

Volumes

To Create Docker Volume:

#Docker volume create testvolume1

To mount volume while creating container:

#Docker run –it –name <servername> --mount source=testvolume1,Destination=/data centos

This will mount the previously created volume to the container path /data

#Docker run –it --name srv04 –v testvolume1:/data centos

This will create container named srv04 and mount testvolume1 to the path /data in container

#Docker run –it --volumes -from srv01 --name srv02 centos /bin/bash

This would create a container mounting volumes from srv01 to srv02

#Docker run –it --name srv05 –v “$(pwd)”:/data1 centos

This will mount current directory as source and /data1 is as destination directory

Here srv05 os server name, $(pwd) is current directory, /data1 is destination directory to mount and centos os image name

#Docker volume inspect testvolume1

This will give details like physical path of the volume in Host

#Docker volume rm <volumename/ID>

#Docker volume rm ${docker ps –a –q}

This will remove all volumes using output specified in {}

#Docker volume prune

This will remove all un-used volumes

Docker Compose:

Advanced topics:

Custom Image build

How to pull the images from self-repository instead of going to public repository?

Which volumes will be mounted default one or all externally created volumes?

How to mount existing volumes to existing running container?

How to modify port number internal and external in running container?

How to copy data from host to running container?

How to change/detach/connect the network for running container?

log analysis

certificate installation in containers

implement kuberneties