Installation

Mac - <https://docs.docker.com/desktop/install/mac-install/>

Ubuntu - <https://docs.docker.com/engine/install/ubuntu/>

Windows - <https://docs.docker.com/desktop/install/windows-install/>

Docker Playground

<https://labs.play-with-docker.com/>

Commands

Docker --help important

Docker --version important

Docker version --format ‘{{json .}}’ important -day to day

Docker info

Docker pull --help

Docker pull ubuntu:20.04 (uses a specific ‘tag’)

Docker pull redis (doesn’t use a ‘tag’, so the latest version is pulled by default)

Docker image ls

Docker images

Docker run redis

Docker ps - to view the container

Docker ps -a - its shows the running the container

Docker run -it redis - to run the container

Docker run -d redis - to detach(which mean run bg process all time)

Docker run -it -d redis - to detach(which means run bg)

Docker run -it  --name=akhilredis -d redis - ro naming the container

Docker stats - cpu usage mem usage

Docker search redis - it go to the docker hub which display available on that website

Docker search --filter=stars=3 --no-trunc redis - we see best rated imgs

Docker search --filter=stars=3 --no-trunc --limit 10 redis - return only top 10 images

Docker start a8217c4c56 (also try name instead of ID here)

Docker stop a8217c4c56 (also try name instead of ID here)

Docker restart a8217c4c56 (also try name instead of ID here)

Docker pause a8217c4c56 (also try name instead of ID here)

Docker unpause a8217c4c56 (also try name instead of ID here)

Docker logs a8217c4c56 (also try name instead of ID here) - when container get error this cmd will useful -important - day to day life

Docker exec -it a8217c4c56 bash (start bash inside the container, type exit to exit the bash) - to go inside the container which run bash script like linux terminal

Docker attach (containername) - interactive bash shell like linux terminal.

The above cmd same as this cmd.

Docker run -it --name=akhilredis -d redis /bin/bash - create a new container -day to day - important -basic

Docker exec 023828e786e0 apt-get update

Docker rename oldcontainername newcontainer (renames the container, container can be running or stopped)

Docker rm test (you have to stop the container before removing it, also try this with container ID)

Docker stop $(docker ps -a -q) (Stops all running containers)

Docker rm -f $(sudo docker ps -a -q) (removes all stopped containers) - warning cmd.

Docker inspect happy\_faraday (also works with ID) - info about container

Docker kill happy\_faraday (same as stop)

Docker kill $(docker ps -q) (stops all running containers)

Docker system prune  - warning cmd remove all

 To see dangling images in action

Cat > Dockerfile

FROM ubuntu:latest

CMD ["echo", "Hello World"]

Ctrl + D (close the file)

(docker build -t my-image .) //builds image from Dockerfile

docker images

//open up the editor in docker playground n make changes

FROM ubuntu:latest

CMD ["echo", "Hello World!"] (the exclamation is extra)

docker build -t my-image . //same command with same image name

docker images //now you will see dangling images

docker image prune -a // specifically works for dangling images

Docker attach <container\_name> (opposite of -d, we will see this in action in the next step)

To work with cp command -

Docker run -i -t --name=akhilredis -d redis /bin/bash

Touch test\_file (creates a test file in the current directory)

Docker cp . akhilredis:/data (copies everything from current directory to akhilredis container)

Docker attach akhilredis (attaches to akhilredis container)

Ls (you will see the file here)

List of processes running in a container - Docker top <container\_name>

 Docker events (run this command in a terminal and in a different terminal, run operations like starting a container etc. u will get all events in the first terminal)

Docker container prune (works specifically for containers which remove all container)

Docker volume create new-vol

Docker volume ls

Docker volume inspect new-vol

 Docker volume rm new-vol (volumes can only be removed if the container they’re attached to is stopped)

Docker volume prune -remove all volume including default

Let’s see an example of attaching vol to a container

Create a new vol -> docker volume create new-vol

Docker run -d --name redisvol --mount source=new-vol,target=/app redis

Now if we try to delete the volume - docker volume rm new-vol

It won’t work as it’s assigned to a container, we can stop container and then delete the volume

attaching a volume in read-only capacity

Docker run -d --name redisvol2 --mount source=new-vol3,target=/app redis,readonly

Task:

SERVICE, NETWORK, COMPOSE, 1 golang and 1 nodejs project

ADDITIONAL QUESTIONS:

Docker - imgs and containers

Kubernets - vms

Dangling imgs - create duplicate with same name it will dangling with duplicate