CONTAINER AND IMAGE COMMAND OPTIONS
Search / filter
docker search ubuntu
docker search --filter stars=100 ubuntu [when we give = it means "> or = " so anything above 100 will also be listed.]
docker search -f stars=7 --filter is-official=true ubuntu [here only "official" images are listed]
docker search --limit 10 ubuntu [lists only the first 10 items]
Tagging an image
docker tag ubuntu:bionic myubuntu:bionic
Using CLI commands to manage images
docker image history <image:version>
docker image save image:ver > image_name.custom.tar
tar tvf image_name.custom.tar [tvf option is to list all the contents of the tar file verbosely]
docker import image_name.custom.tar myubuntu:V2 [where myubuntu:v2 is the image name and version name]
docker image prune [This will remove all the dangling images, images that dont have a connection]
docker iamge prune -a [will remove ALL the images of those that does not have atleast one container.]
Inspect images
docker image inspect image:version > image.output
docker image inspect ubuntu:bionic --format '{{.GraphDriver}}'
docker image inspect ubuntu:bionic --format '{{json .GraphDriver}}' [This gives a better formatting]
docker image inspect ubuntu:bionic --format '{{.RepoTags}}' [This is to identify all the tags associated with a particular image . in case we dont have any tagged images just go ahead create {ex. docker tag ubuntu:bionic anotherrepo/ubuntu:V2 }] output will be : [anotherrepo/ubuntu:V2 ubuntu:bionic myrepo/ubuntu:V1]
Dynamic / Run time variable assignment
docker run -it --rm --env MYVAR=testvar --name kalyan_new3 ubuntu:bionic /bin/bash
docker run -it --rm --env MYVAR=testvar --env MYVAR2=testvar2 --name kalyan_new3 ubuntu:bionic /bin/bash