Exercise - 1

docker search	docker search ubuntu #Search the Docker Hub for image docker search --stars=1000 --no-trunc ubuntu #Above command displays images with a name containing 'ubuntu', at least 100 stars and the description isn't truncated in the output:
Docker pull ubuntu	Docker pull ubuntu #will contact Docker public registry and download the image(s). docker pull ubuntu:latest docker pull ubuntu:14.04 #Given without any tag, docker pulls the latest image.
docker run	docker run [OPTIONS] IMAGE[:TAG| [COMMAND] [ARG...] docker run --name "myubuntu" -ti ubuntu :14.04 /bin/bash #Starts the container and takes us to the bash of the container. #-t attaches a terminal #i stands for interactive (standard input opens) #Ctrl + p + Q to quit the container without stopping it.
docker attach	docker attach <Container ID | name> docker attach myubuntu To attach the running container
docker top	docker top <Container ID | name> docker myubuntu #Tells what applications are running in the given container
	cd /var/lib/docker/containers #It will list all the containers with long Ids…. #cd <long container Id> and see the list…it shows the list of files
docker stop	docker stop <Container ID | name> #Stops one or more running containers docker stop myubuntu #Stops the container
Docker start	docker start<Container ID | name> #Start one or more stopped containers docker start myubuntu
Persistent Storage for Docker	
1. Start a container 2. Bash into the container 3. Create a file(s) 4. Exit container safely 5. Go to the default volume folder in the host machine 6. Create some files in the host system 7. Bash into the container again 8. See the new files there	docker run -ti -v /data --name myubuntu1 ubuntu:latest /bin/bash cd /data touch file1 Ctrl + p + q # to come out of the current container cd /var/lib/docker/volumes/<Container ID> #to see the attached volumes cd _data touch file2 file3 docker attach myubuntu1 cd data ls

Attaching a data from host 1. Create a folder in the host system 2. Change to the folder 3. Create a sub folder 4. Start docker container and mount the folder path into container 5. Create few files 6. Exit container safely 7. See the files that are created in the container existing in the host system	 mkdir /dockervol cd dockervol mkdir ubuntu3 cd ubuntu3 docker run -ti -v /dockervol/ubuntu3:/data --name=ubuntu3 ubuntu:14.04 /bin/bash cd /data touch file1 file2 file3 Ctrl + p + q cd /dockervol/ubuntu3 ls
Another example	docker run -it --name ub1 -p 1234:80 -v $(pwd)/.:/home/test_dir ubuntu:16.04
Creating a read only mount	docker run -it --name ub1 -p 8080:80 -v $(pwd)/.:/home/test_dir:ro ubuntu:16.04
docker kill	docker kill $(docker ps -q)

Container with network connection

1. Pull a tomcat container 2. Start tomcat container 3. Browse using http://<ip>	docker pull tomcat docker run --name mytom -d -p 80:8080 tomcat
	Some other example with Maven image
Create a maven container with required group and artifactId	docker run --rm -it -v $(pwd):/myproj -w /myproj maven mvn archetype:generate \ -DgroupId=hello.docker \ -DartifactId=HelloDocker \ -DarchetypeArtifactId=maven-archetype-webapp \ -DinteractiveMode=false
Package the java application with mvn:package command	docker run --rm -it -v $(pwd):/project -w /project maven mvn package