1- check the docker version installed

* docker version

2- run docker container for hello-world

* sudo docker run hello-world

3- run docker container redis in detached mode

* docker run -d redis

4- try to stop the running redis container

* docker stop 0f537848bed7

4- check the present container o the host

* docker inspect cfa7b1da8b87

5- check the ID of the redis container

* docker ps |grep redis

6- try to run a container from nginx:alpine and delete image

* docker run nginx:alpine
* docker rmi nginx:alpine

7- delete the image redis

* docker rmi 3358aea34e8c

8- pull image from nginx:1.14-alpine

Run an instance of the ubuntu image to run the sleep 1000 command at

startup.

Exec into the container and touch a file called test-file

* docker pull nginx:1.14-alpine
* docker run nginx:1.14-alpine sleep 1000 &
* docker exec -it ba11f888c3e8 /bin/sh
* # touch test-file
* # ls

9- Run a container with the nginx:alpine image and name it web

10- delete all the images from the host

11- Run an instance of nginx:alpine with a name nginx and map port

8080 on the container to 38282 on the host.

12- create ubuntu image and check the size of it

13- Run a container named blue-app using image kodekloud/simplewebapp

and set the environment variable APP\_COLOR to blue. Make the

application available on port 38282 on the host. The application

listens on port 8080.

14- Deploy a mysql database using the mysql image and name it mysqldb

Set the database password to use db\_pass123 then inspect it to

check the value

15- pull the code from https://github.com/sabreensalama/simpleflask-

app/tree/main and create a docker file for this flask app

16- Create a volume called mysql\_data, Run a mysql container again,

but this time map a volume to the container so that the data stored

by the container is stored at /opt/data on the host.

Use the same name : mysql-db and same password: db\_pass123 as

before. Mysql stores data at /var/lib/mysql inside the container.