**Docker Assignment 3**

1. What is the way to establish communication between the docker and Linux hosts?

* First of all we need see the networking/ethernet is enabled or not by using command **ipconfig** or **ifconfig** on the docker host. This command ensuresa that an ethernet adapter is created as long as the docker is present in the host
* Now let's take a scenarios two containers can communicate to each other and there are two ways
* **Communicating through networking**: Containers are usually isolated from each other. But they can send and receive requests to other applications using networking.
* eg. a web server container can expose a port so that it can receive requests on port say lets say 80.
* **Sharing files on disk**: Some applications communicate by read/write to a file operations. applications can communicate by writing their files into a volume and then it can be shared with other containers.
* eg. let say we have data processing application which write a file to a shared volume which contains customer data, which is then read by another application container. or multiple containers can read from that location.
* let see it from below eg. now you want to mount the disk or keep data external for eg. you have mysql as app inside container that will keep all data in containers path only like /var/lib/mysql so you want to keep them in external outside of continainer or host then can do that

**syntax:**

* **docker run -v <path of external host>: <path of/inside container> app\_name/image\_name**
* **docker run -v /home/ubuntu/data:/var/lib/mysql mysql**

1. Differentiate between COPY and ADD commands that are used in a Dockerfile?

* In Dockerfile it provides two ways to copy files from the source system into an image: the COPY and ADD
* COPY and ADD are both Dockerfile instructions that are almost same purposes. with this we can copy files from a specific location into a Docker image.
* COPY takes in a source and destination. It only allows you copy in a local or directory from your host (the machine building the Docker image) into the Docker image itself.
* ADD does that same job, but it also supports 2 other sources.
* First you can use a **URL** instead of a **local file/directory**.
* Secondly you can **extract tar** from the source directory into the destination.

1. Can you tell what the functionality of a hypervisor is?

* A hypervisor is a basically a software that allows to run virtual machines on a given host. A hypervisor allows one host to support multiple VMs by virtually sharing its resources, such as memory and processing. In virtual machine we can create and run multiple OS virtually. for eg. using vmware or VirtualBox only windows OS we can install these software and on top of that we can create VM and run Linux. this is a classic eg which almost all IT/software engineer might have used in there college days or might still be using that same.
* There will be Hardware, on top of that will be OS(eg. Hypervisor/vexsi), on top of that will be multiple virtual manchine. Now this virtual machines will now again has it's own OS in each virtual machines. Now on top of the VM's OS containers will be running
* features of VM are as follow:Resouce utlization will be more with this kind of setup
* Since in each VM there willbe vitual OS so Disk utlization will be more ( In GB's)
* Slow boot time
* Deployement is lenghty
* Hypervisors are of 2 types:  
  1. **Native Hypervisor:** This type is also called a Bare-metal Hypervisor and runs directly on the underlying host system which also ensures direct access to the host hardware which is why it does not require base OS.  
  2. **Hosted Hypervisor:** This type makes use of the underlying host operating system which has the existing OS installed.

1. Can you tell me something about the docker namespace?

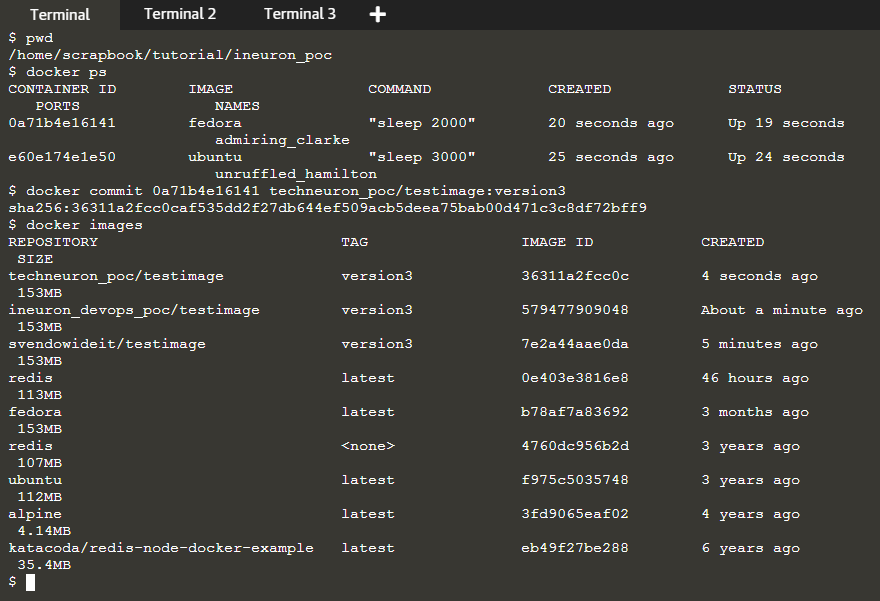
* Whenever a container is run then Docker will create a set of isolated workspaces for the container called namespaces.
* Namespace adds a layer of isolation in containers. Each container runs in a separate namespace and its access is limited to that namespace.
* This forms the core concept behind containerization as namespaces introduce a layer of isolation amongst the containers.
* Docker provides various namespaces in order to stay portable and not affect the underlying host system. Few namespace types supported by Docker – PID, Mount, IPC, User, Network

1. In what circumstances will you lose data stored in a container?

* As soon as you delete container data will be lost.
* The data remains of a container in it until and unless you delete the container

1. Can you use a container, edit it, and update it? Also, how do you make it new and store it on the local system?

* Yes you can use a container and can edit it and update it with below command
* To update configuartion of docker comamnd is
* **docker container update OPTIONS container\_name**
* To create a new image from a container’s changes : docker container commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
* It can be useful to commit a container’s file changes or settings into a new image.
* **docker commit conatainer\_id username/imagename**
* Below is the referencee poc



PTO for next question

1. How to delete an image from the local storage system?

* To check what all images are there on host use command

**docker image ls** or **docker images**

* Comamnd to remove image is **docker rmi image\_id**
* To focefully remove we can use the -f option. f means focefully
* If you use the -f flag and specify the images then this will untags and removes all images that matchs the specified id.
* **docker rmi -f image\_id**
* Note that for a running contmainer image either we an stop and remove or we can remove forcefully with -f option

1. How to build a Dockerfile?

* After writing a docker file which is a set of instruction then you need to build it to create an image with those specifications. Use the following command to build a Dockerfile:
* **docker build *path\_to\_ docker\_file***
* Let us see that in depth with one example. For eg.We can create a docker file with following

FROM ubuntu

RUN apt-get update

RUN apt-get install-u python3 pythin3-pip

RUN pip3 install flask

RUN mkdir /pot/app

WORKDIR /opt/app

COPY . /opt/app

ENTRYPOINT FLASK\_APP=/opt/app/app.py flash run --host=0.0.0.0

Now we need to build with below commands open terminal and execute below commands

docker build . -t calc -> to build a image of above commands on top of ubuntu image and tag

docker images -> to check the image and there should be image with name as calc

docker run -p 8080:5000 cal -> to do a port mapping to allow access external to container and run

-> from this step output we get the IP which we can check in browser with port 8080

If you need to push it somewhere then first login

docker login -provide user id and password

docker push calc - Now you can sign into your docker hub and check it repository from there

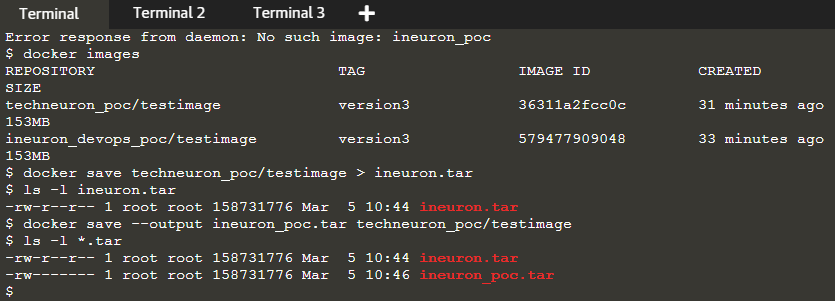
docker ps -> to check the existing active containers

1. Do you know why the docker system prune is used? What does it do?

* The below command is used to remove all the stopped containers, all the networks that are not used, all dangling images and all build caches. It’s one of the most useful docker commands. **docker system prune**

1. What command can you run to export a docker image as an archive?

* To save one or mutliple images into a tar archieve can be done woithe below comamnd and the syntax is **docker save -o exported\_name.tar container-name**
* -o means write a file to output/stdout
* **docker save image\_name > image\_name.tar**
* below is some examples
* **docker save *techneuron\_poc/testimage* > *ineuron\_poc*.tar**
* docker save --output ineuron\_poc.tar techneuron\_poc/testimage
* docker save -o ubuntu-all.tar fedora
* docker save -o ubuntu-latest.tar fedora:latest
* You can use gzip to save the image file and make the backup smaller.
* **docker save image\_name:latest | gzip > image\_name\_latest.tar.gz**

****