Docker Introduction

Install docker engine, instructions are given in the docker website

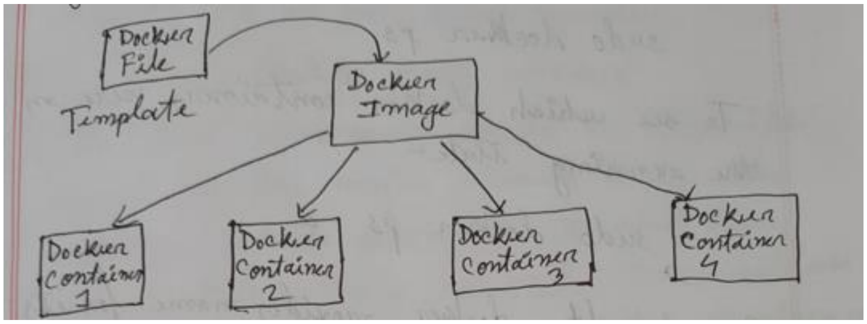
Check docker version

```
sudo docker version
```

launch an nginx server

```
sudo docker run nginx
```

The docker run command will first search in the host machine. If it is unable to found in the host, it will pull image from docker hub/ registry.



When the application is not in running state, it is called 'image'. Image contains everything to run the program.

When the application is in the running state, it is called container.

List the number of docker images

```
sudo docker images
```

Custom images can be created and can be pushed to the docker hub.

To know the number of processes for docker:

```
sudo docker ps
```

To see which docker containers are in the running state:

```
sudo docker ps -a
```

By default docker provides name for its containers but custom name can be given.

```
sudo docker run --name second nginx
```

To delete the a docker container:

```
sudo docker rm ‘container name’
```

To run an existing docker container start:

```
sudo docker start second
```

To stop a running container

```
   sudo docker stop second
```

To immediately halt stop a container use 'kill'. It is not recommended.

```
   sudo docker kill second
```

[The 'kill' command sends a SIGKILL signal to the main process inside the container, which forces it to shut down immediately. This abrupt termination does not allow the process any time to clean up or save its current state.]

Before, I have deleted a container now I shall delete an image. I will delete a redis image.

```
   sudo docker rmi redis
```

If it needs to be accessed from outside, for example the nginx container, I have to enable port forwarding: The port forwarding in this case, the external port of the nginx is to be bound with a host's port.

```
sudo docker run -p host port:container port nginx
sudo docker run -p 5000:80 nginx
```

Now in a browser of the host machine,

192.168.0.10:5000

When the page opens up, there are some texts which I want to modify.

First copy the container id from ‘docker ps’ command.

```
sudo docker exec -it container id bash
sudo dockesr exec -it 9cfe1d6f2a5d bash
root@9cfe1d6f2a5d:/#
```

-i or --interactive: This flag keeps STDIN open even if not attached which is necessary to interact with the container.

-t or --tty: This flag allocates a pseudo-TTY or terminal which connects the terminal with the standard input and output of the container.

So ‘docker exec -it’ allows to start an interactive session with the container.

Now use 'ls' command inside the container

```
ls
```

Output:

```
bin  docker-entrypoint.d  home  media  ...
boot  ...   lib ...
```

Similar to 'the host', there is a file system. Now go to the directory

```
cd /usr/share/nginx/html
vim index.html
```

output: bash: vim: command not found

‘vim’ is not available inside the container, so install it.

```
apt update -y
apt install vim
```

Now edit the 'index.html' for necessary changes.