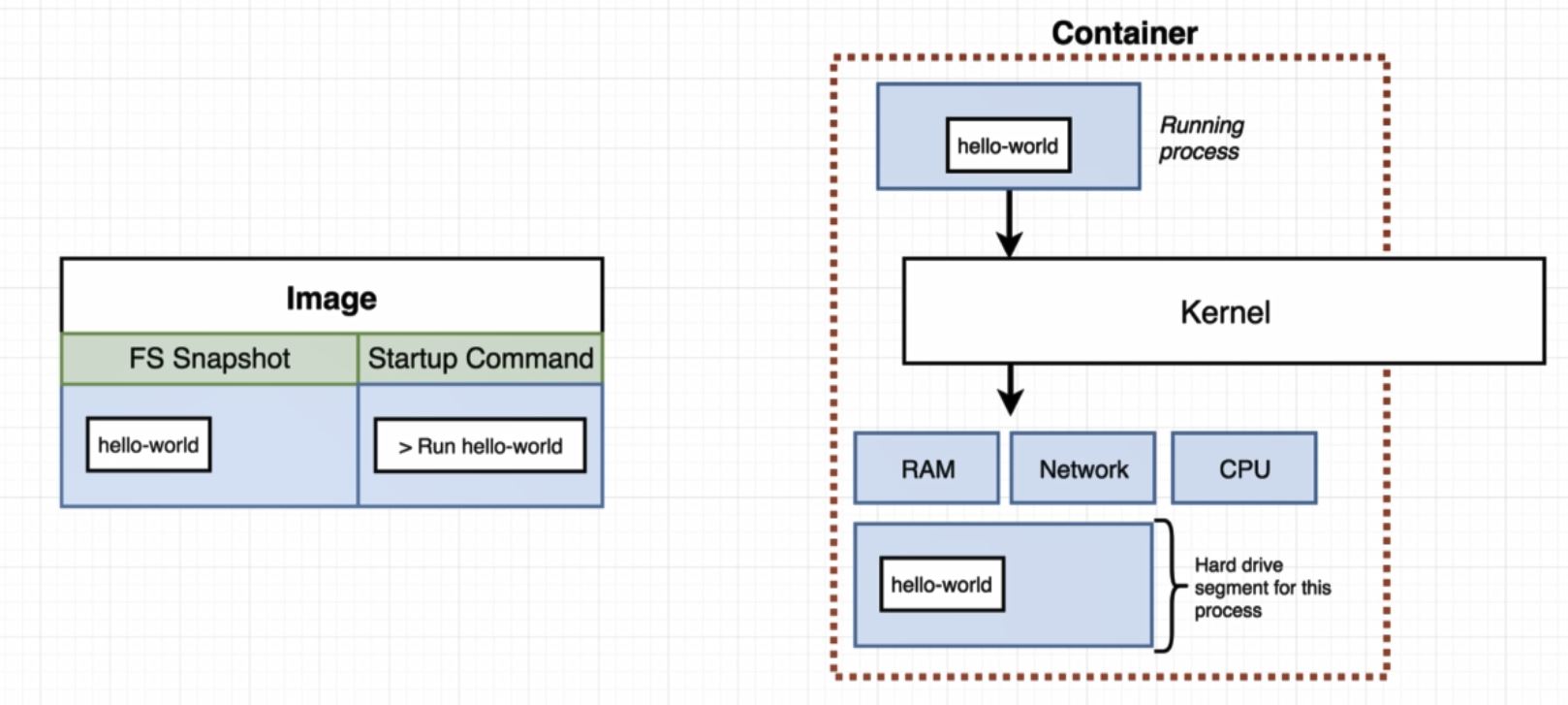
**Docker Cli Commands**

* docker run hello-world

Download image from hub if is not exist on local and save it on docker cache, copy Fs Stapshot from image inside container and run start up command from image inside container (2-1.jpg)



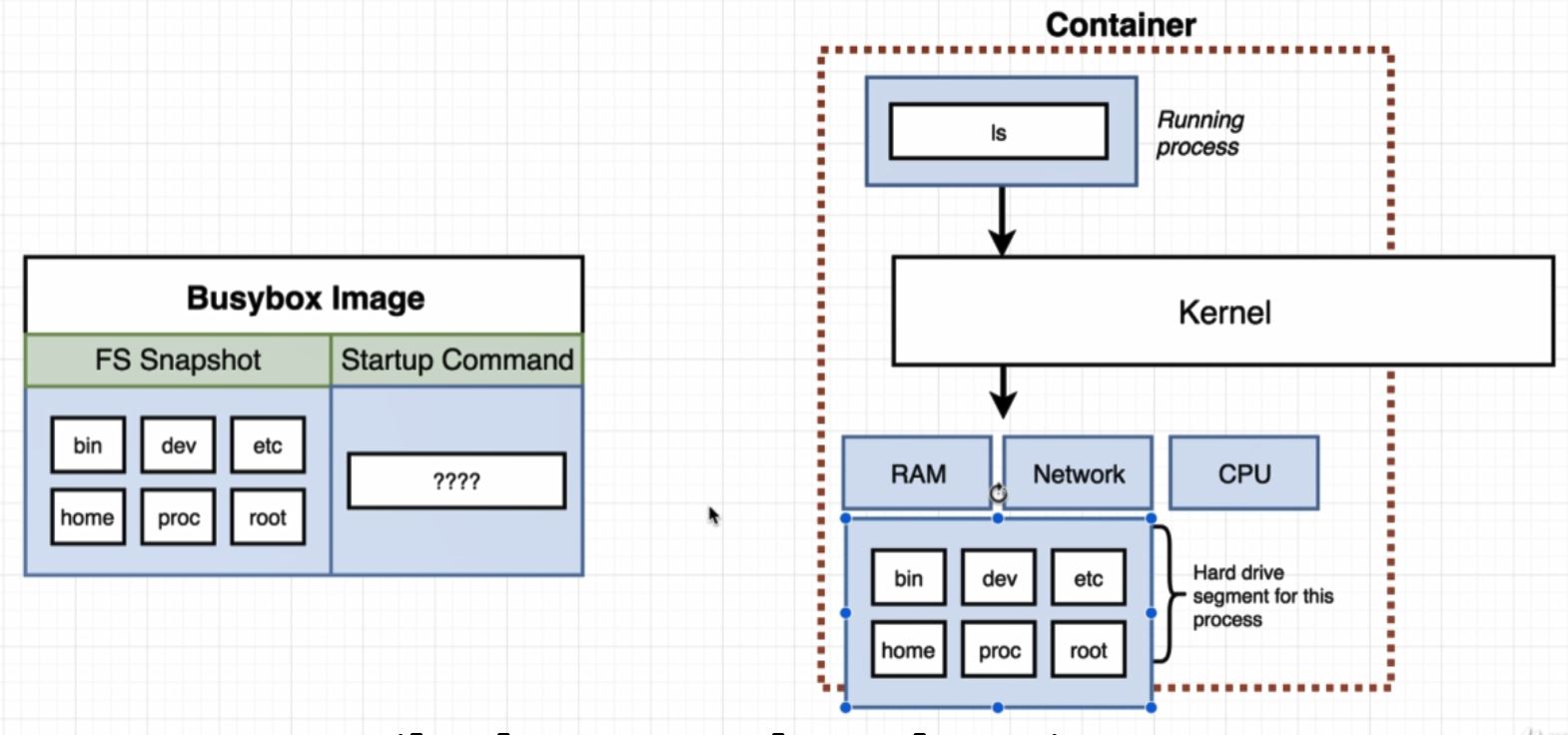
\*\*To override start up command from image:

* docker run busybox echo hi there

\*\* "echo hi there" command run after container started

* docker run busybox ls (2-2.jpg)

\* ls or echo command exist on busybox image, we cannot run these command on hello-world if does not contain these command



\* This command will list all the different running container that are currently on your machine

* -docker ps

\*\* this command will display null list because docker run hello-world quickly run and close container

\*\* to run container for more time we can use :

* docker run busybox ping google.com

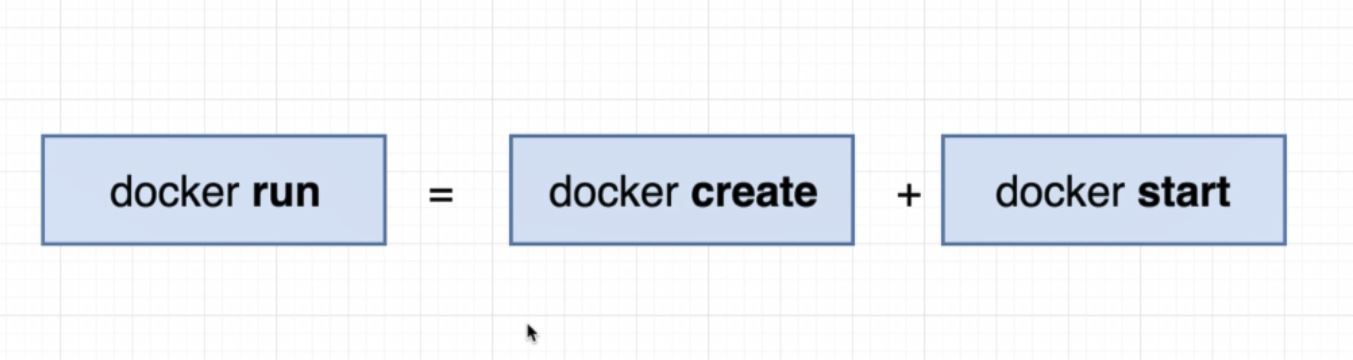
\* in this case by runing docker ps on another window we can see container with status running

\* This command is used to display all container that have ever been created on our machine

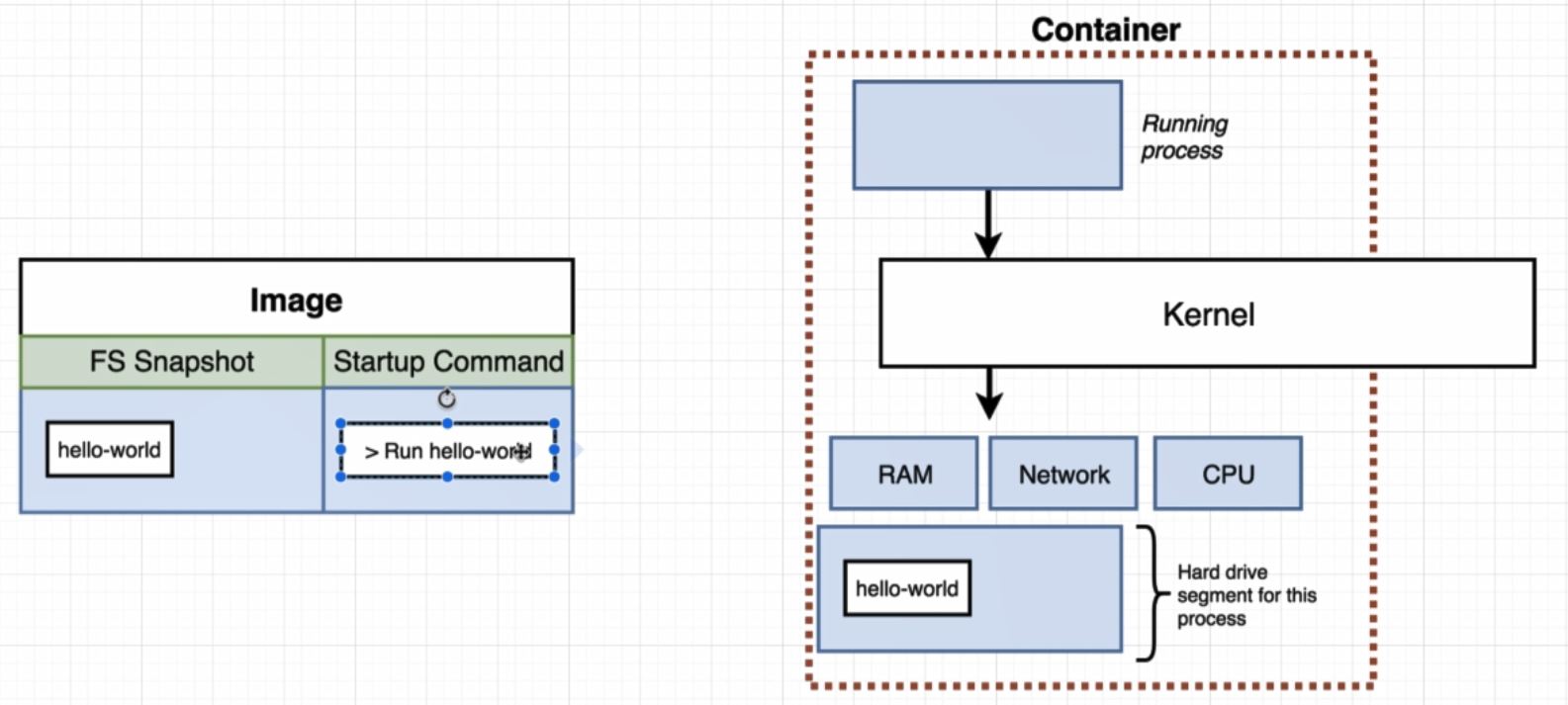
- docker ps –all

\* docker run = docker create + docker start

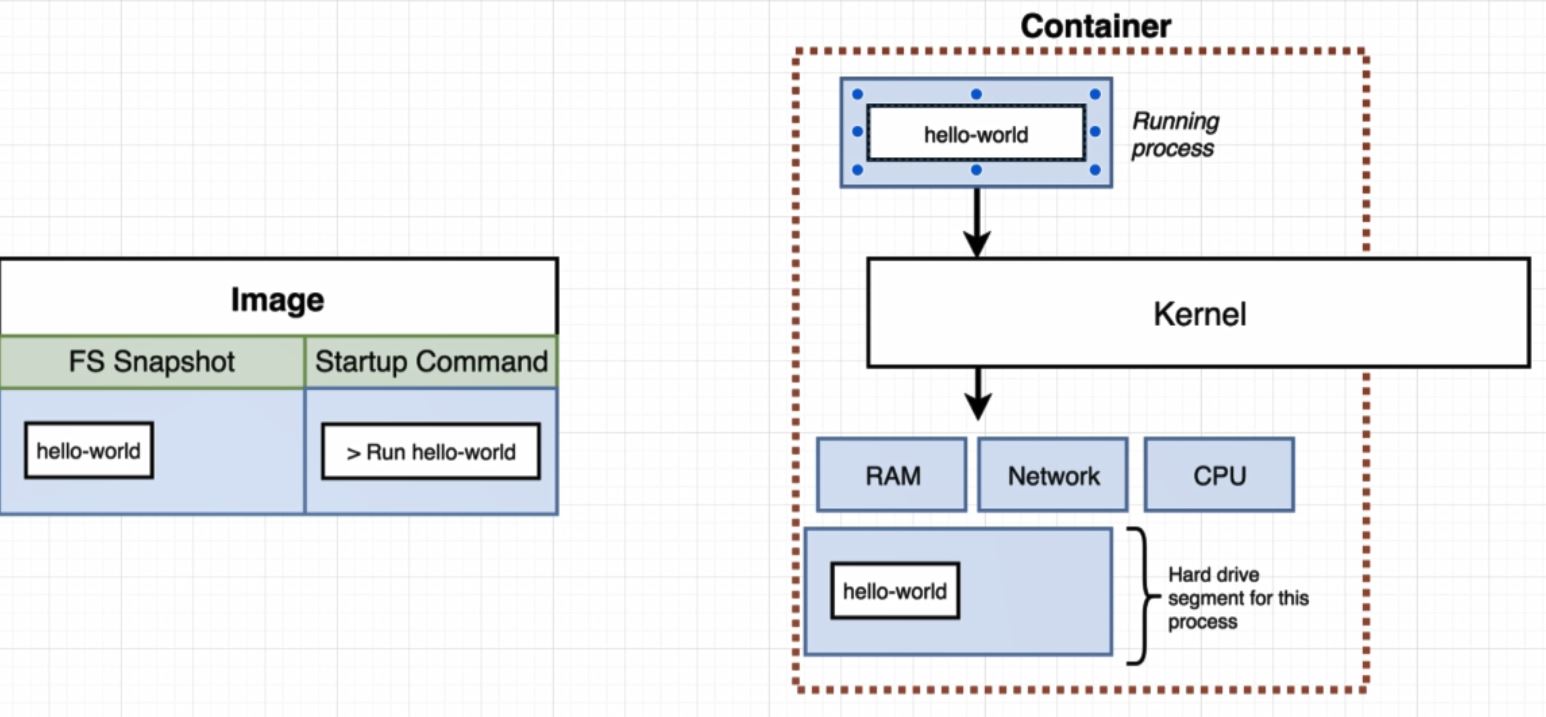
* docker create hello-world (return ID)
* docker start -a ID
* (-a is to make docker actually watch for output from the container and print it out to your terminal)



docker create hello-world



docker start –a ID



\*\* After execute docker ps --all , in column Status if a container Exited we can still start it back up (Exited dose not mean that container dead )

\*\* We cannot override command on docker start command

* docker start -a ID echo bye there (error)

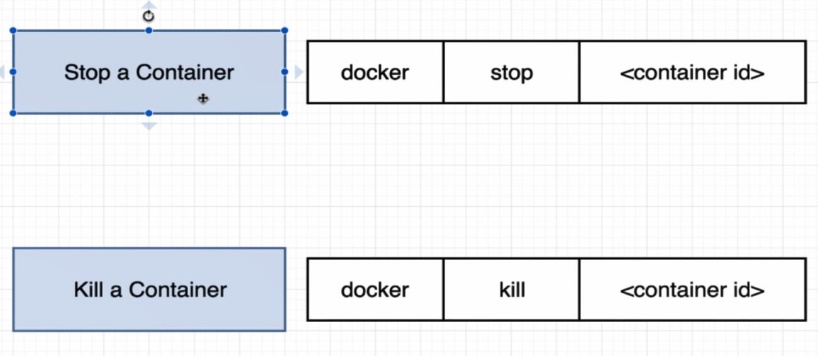
\*\* After execute docker ps --all , in column Status if a container Exited we can still start it back up (Exited dose not mean that container dead )

\* To delete all container we can use this command

* docker system prune

\* To go back to that stopped container or container which still runnig and get all the logs we can use command bellow. If a container start without –a flag then we can use command bellow to display output of the running or runned container

* docker logs ID



\* To stop runnng container:

* docker stop ID

\* it will try to stop container for 10 seconds, if cannot stop after 10 seconds it will kill it

\* to kill container:

* docker kill ID

\* To start up redis on my machine using command bellow:

* docker run redis

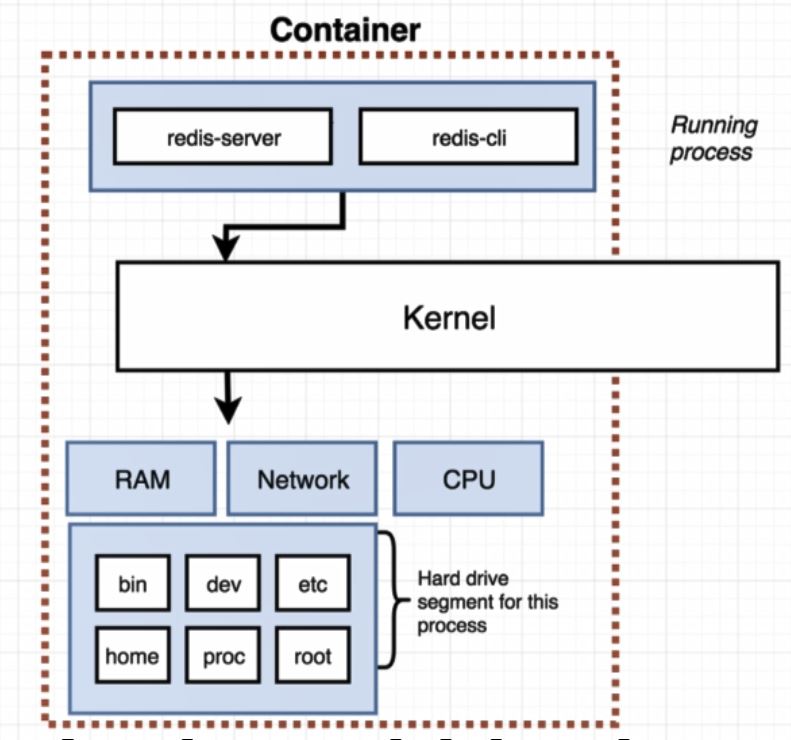
\*\* By running command above, in background docker cli reached out to docker hub and download a single file called image(a file contain all the dependencies and configurations required to run a program).

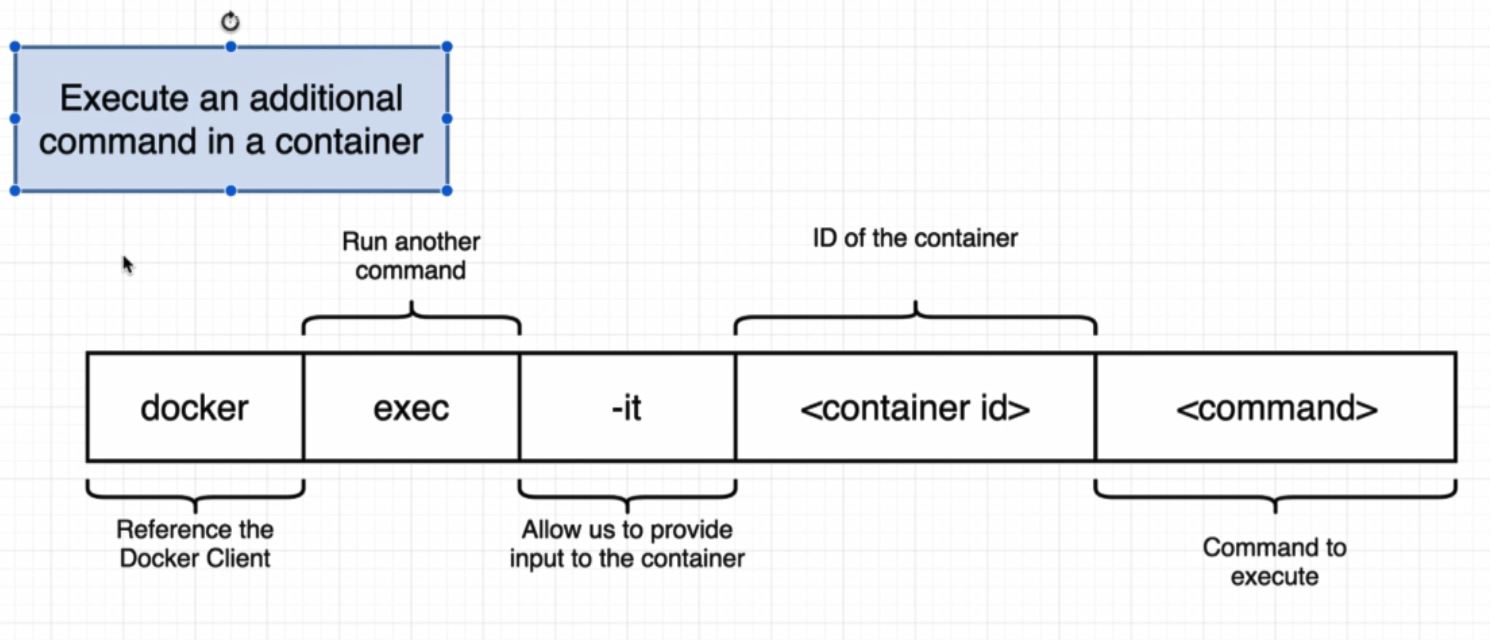
\*\* After download a image we can use this image to create container.(Container is a instance of image and has own little space on memory and has space on networking, hard drive space and so on)

\*\* Outside of the container we have no access to anything that going on inside there.

\*\* For example after create and start redis , we cannot start redis cls from another cmd outside of container.

\*\* If you want to start up the redis cli, you need to get into this container and execute a second command inside container like command bellow(2-7.jpg)





* docker exec –it ID radis-cli

\* By using code bellow we can start up redis CLI inside of latest container(docker run redis).

* docker exec -it <container id> <command>

\* (-it): Allow us to provide input to the container

\* (-it = -i -t): -i say to container to get input if need, -t used when a command get input show user usefull information

* docker exec -it b4c8af03474e redis-cli

\* To execute a command inside container we should run (docker exec) for each command but there is a better way to do this is to run command shell inside docker and run any command you need

* docker exec -it b4c8af03474e sh

\* sh is a program that is being executed inside that container (a program that allow us to write command in and execute it inside container)

\* We can now start to type out typical commands that we could expect to be able to run in UNIX environment. (cd, ls, echo vs)

\*\* to exit command shell we can use CTRL+d

\*\* we can run sh program inside container when we just create it like command bellow:

* docker run –it busybox sh

\*\*\* between two containers they do not automatically share their file system