1. Downloading the images.

docker pull <imageName:Tag>

Note: By default the image take latest tag if none provided

1. List images in machine

* docker images

1. running container from docker image

* docker run -<arguments> <imagename:tag>

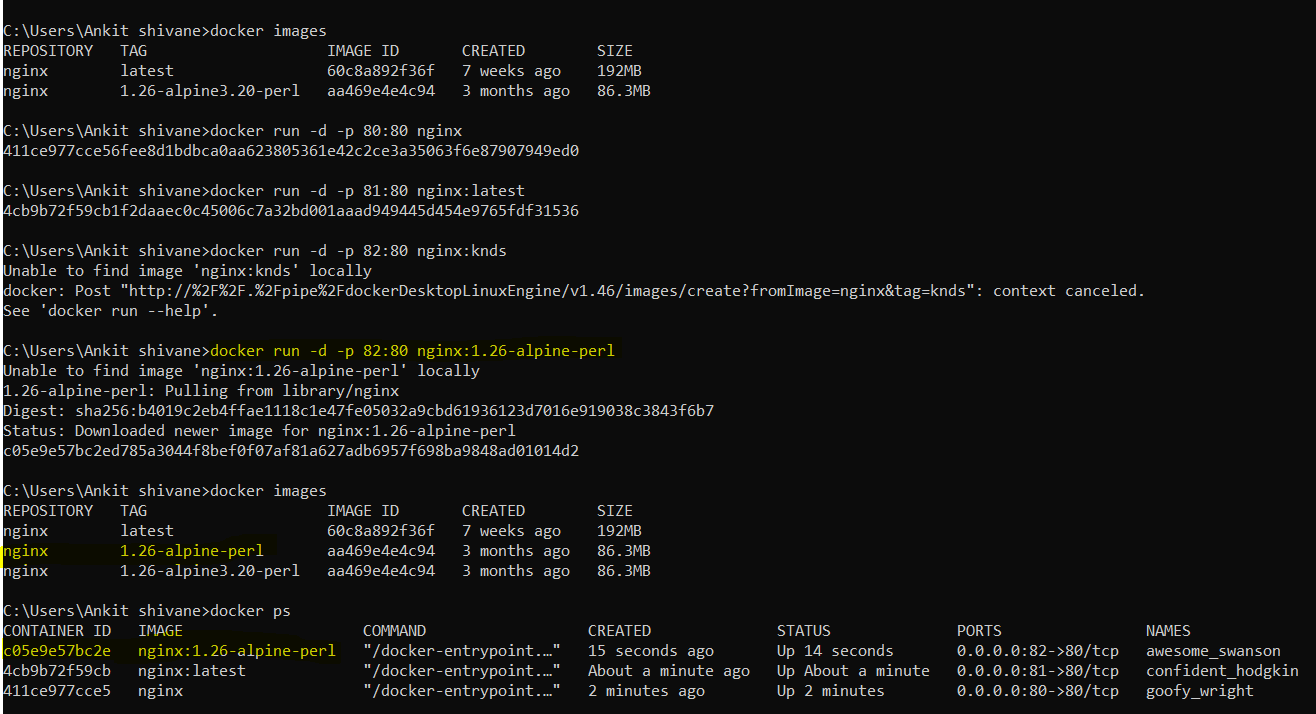
Note:

* 1. if none of the tag provided then it will take the latest tag

For example: docker run -dt -p 80:80 nginx

To verify in browser type localhost:80 as output it will load the nginx page

* 1. we can not run the docker run command for the same image on same port for example if we already started the nginx container on 80 port the we can not use it to start the same container again we need to use other port for example 60:80 we can take but not 80:80
  2. If we use docker run command with image tag that does not available in our local machine then it will connect to docker hub and first download the image and then run the container from that specific tag for example in below example the the nginx image the given tag is not available in the machine so it downloaded and then ran the container “1.26-alpine-perl”
* docker run -d -p 82:80 nginx:1.26-alpine-perl



* 1. If we run a docker container command without any tag then it will run the container for the latest tag
  2. OPTIONS:
     1. By default the docker will give any random name to a docker container if we do not specify so to give meaning full name use –name option in docker command to specify the name for example: docker run –name mynginx -d -p 90:80 nginx Note: we can not run a docker container with same name or also same port even if the container name is kept different.

1. list/check containers

* docker ps

Note: this command will give the list of all the running status

If we want to check what all are the containers which are running, stopped or kiled then use below command with ‘-a’ argument

* docker ps -a

1. stop running container

* docker stop <container\_name or dontainer id>

Note: if we try to stop a already stopped container then it wont throw any error it will just return the same container id as result

1. to kill a running container

* docker kill <containerId or container name>

Note:

* 1. if we try to kill already stopped or killed container it will throw error for example as below:

C:\Users\Ankit shivane>docker kill 772571aa7f8d

Error response from daemon: cannot kill container: 772571aa7f8d: container 772571aa7f8dafe0dec9ddadbd831fc0e19adb93cbb2cdaaef5c24fb07ba2bbe is not running

* 1. we cannot start a already killed container using **docker start or docker run command**
  2. we can not kill the container which is in stop state using docker stop command.

1. to delete a container, container can be only deleted if its not running

* docker rm <container name or container id>

Note: if we try to delete a container which is already running then we will get below error C:\Users\Ankit shivane>docker rm 00fe534c8555

Error response from daemon: cannot remove container "/great\_hawking": container is running: stop the container before removing or force remove

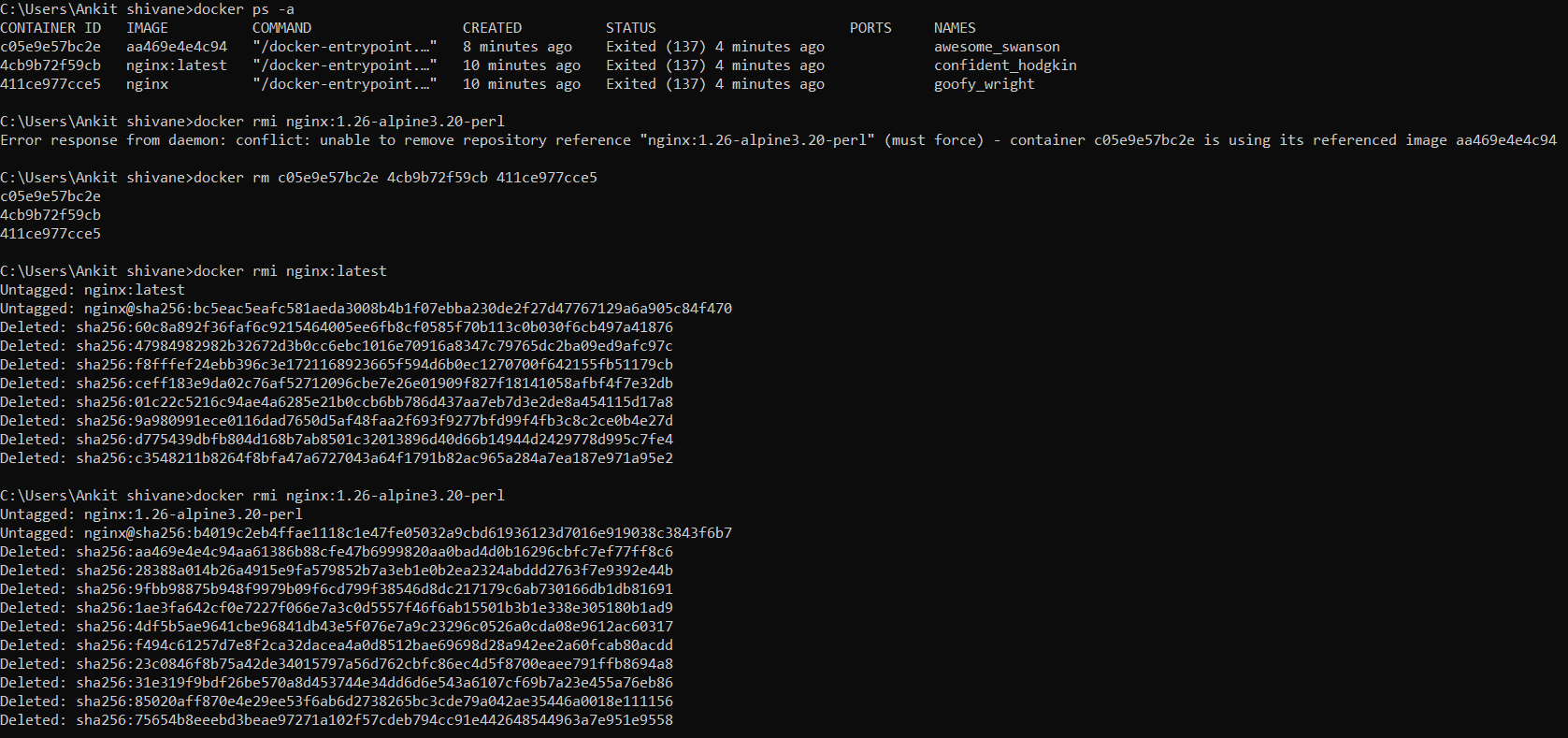
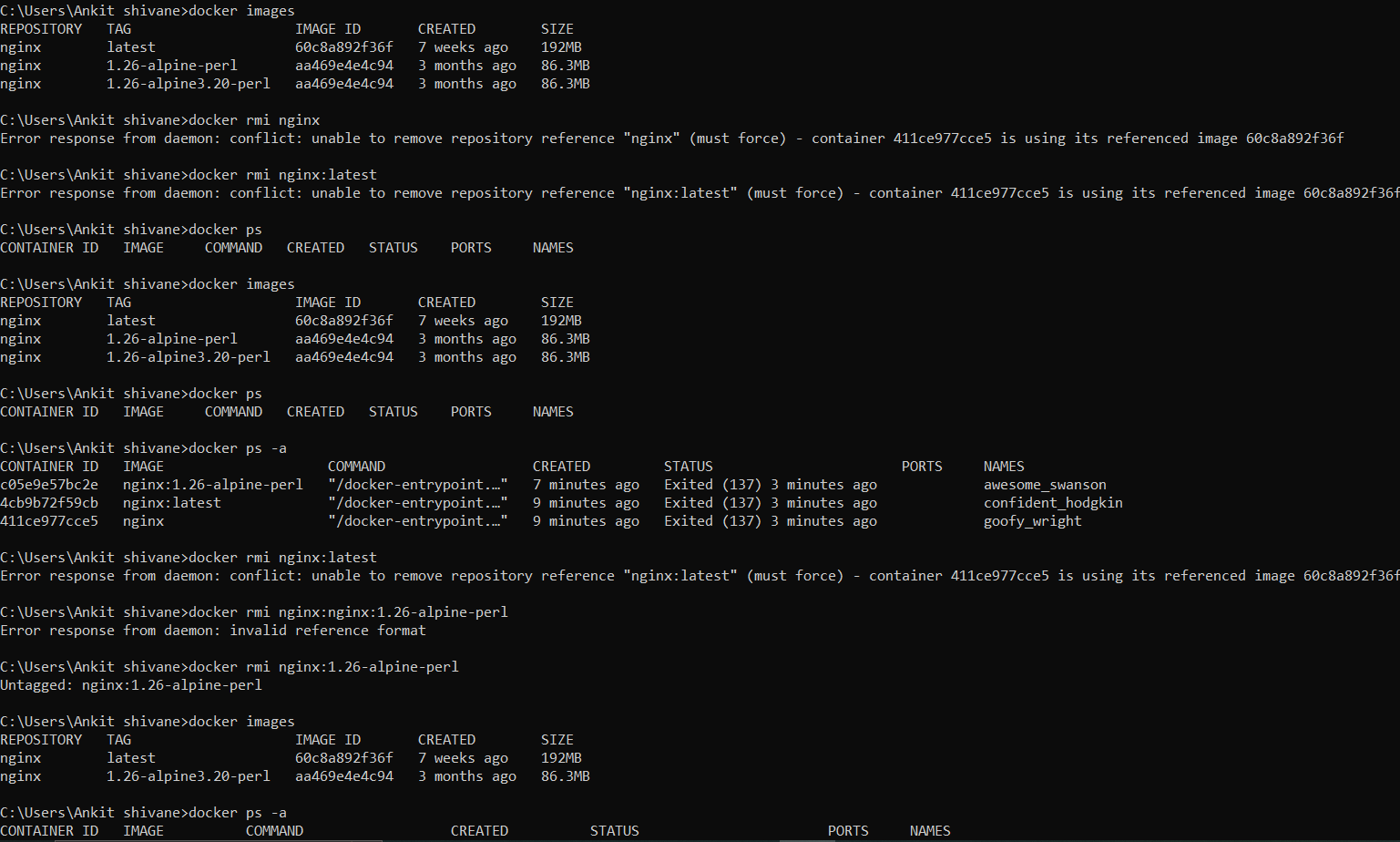
* we can even pass multiple container ids even in single commnds separated by space so that it will delete all at once For example: docker rm cid1 cid2 cid2

1. delete images from docker

* docker rmi <imageName>

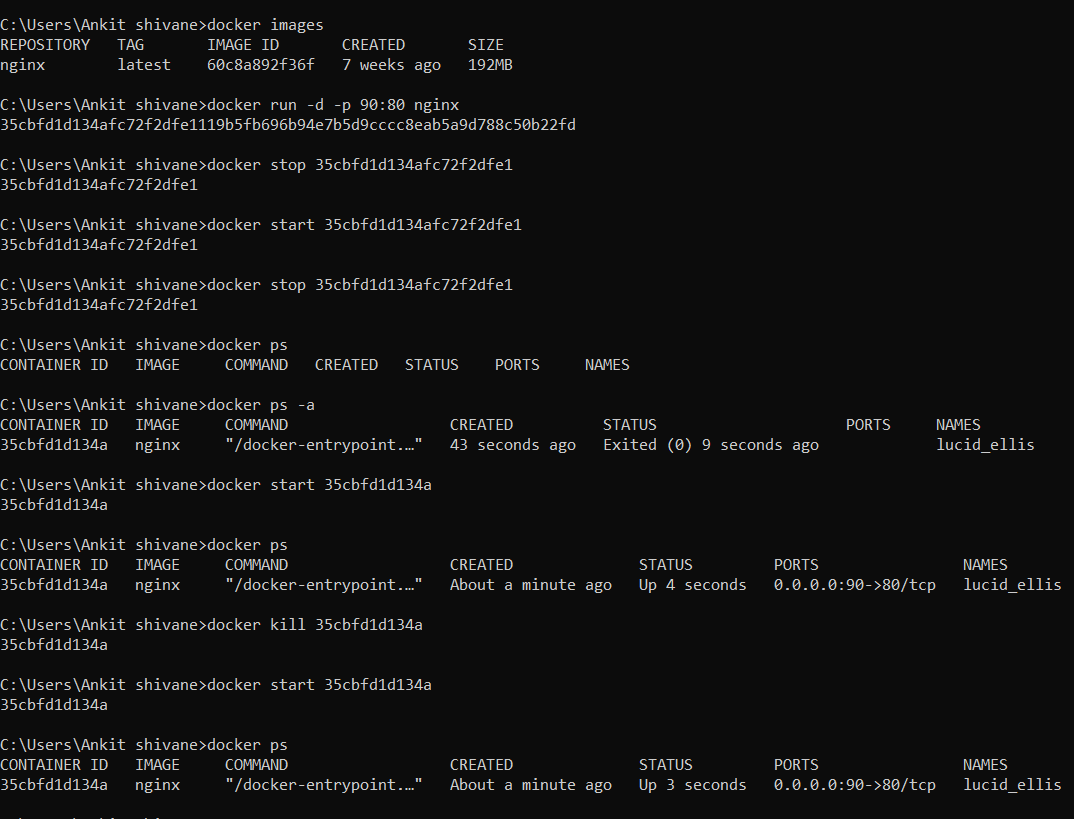
Note:

* 1. if there are any containers even if in the killed state nothing running still docker wont allow to delete the images until you delete the stopped or killed container.



1. starting the stopped or killed container using below command

* docker start <containerId or container Name>

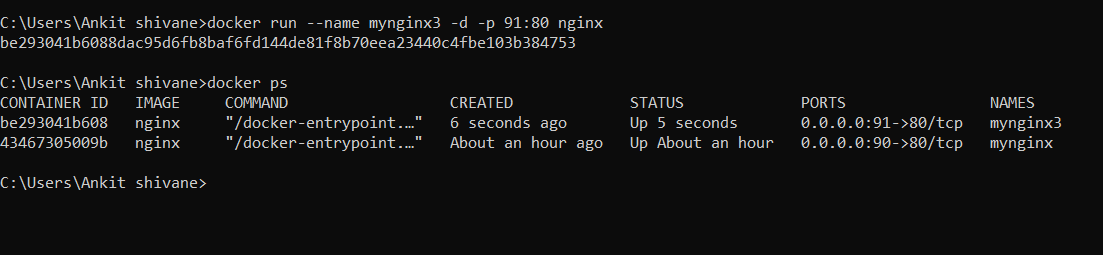


how to start a pause container? What is the different state of docker container?

Difference between docker kill vs docker stop?

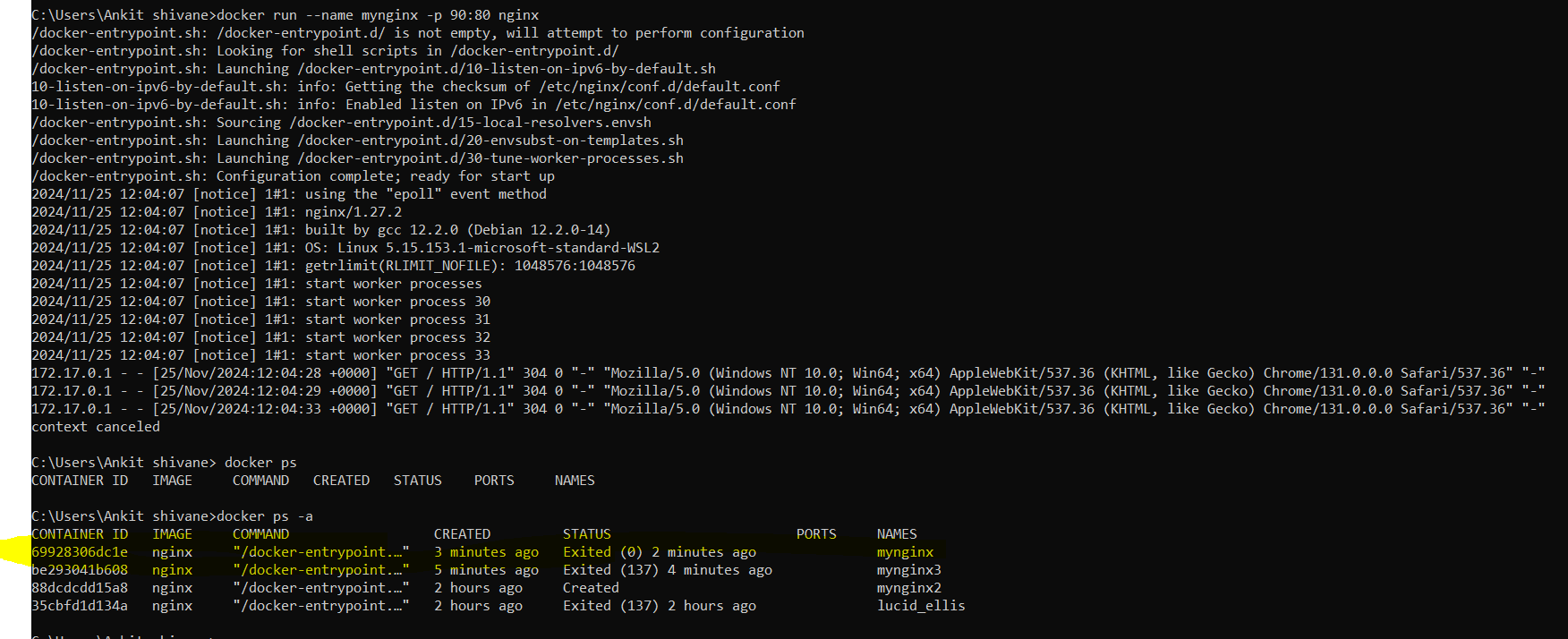
**PORT Binding:**

1. By default, Docker containers can make connections to the outside world, but the outside world can not connect to the containers.
2. If we want containers to accept incoming connection from the world,you will have to bind it to a host port.



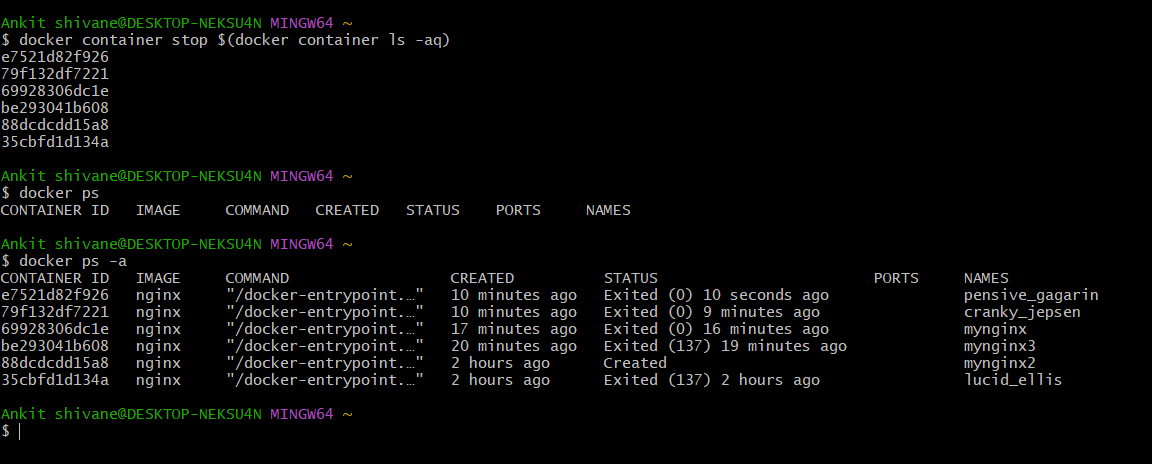
**Detached mode:**

1. Here if we do not specify -d option in docker run command then it will run the container in attach mode it will show all the logs in the same console

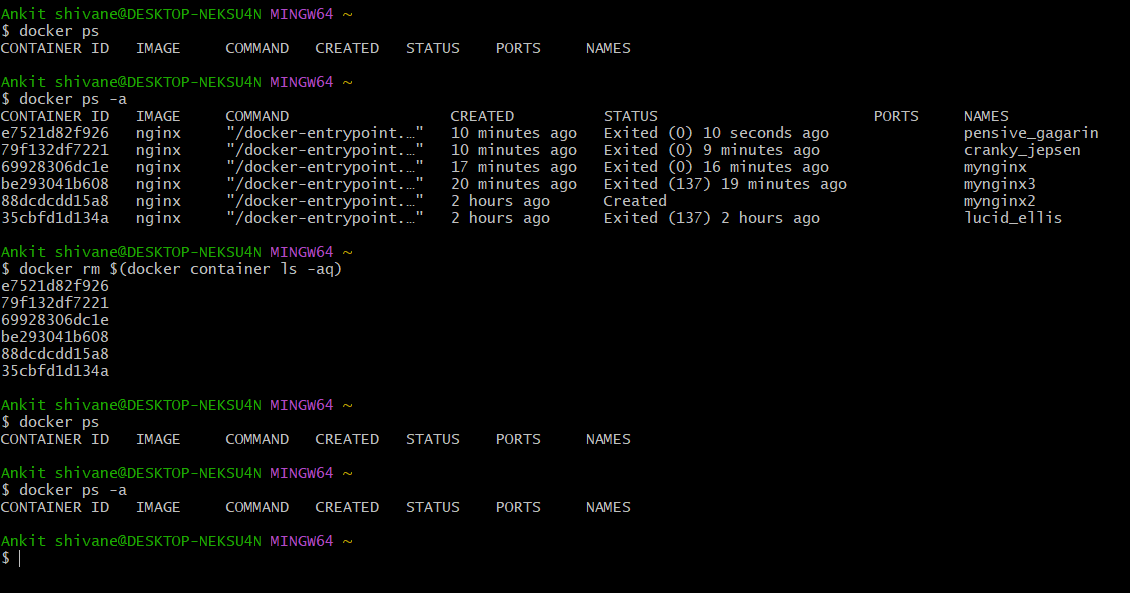


And if you press ctrl+C then it will stop running the container

* Stop all running docker container using scripted command like shown below (NOTE: In windows below command won’t work so use either Linux based OS or in windows use git bash)



* Remove all container from machine using below command



* Exec command:

The exec command allows to run any command inside docker container only thing is the container has to be in running status and not restarted.

* 1. For example: below we are using docker exec command to get access of bash shell of container so we can execute command inside it.
* docker container exec -it 75ce481f4702 bash

syntax: docker container exec -it <CONTAINER\_ID OR CONTAINER\_NAME> <SHELL\_NAME>

* 1. If we don’t want to login into the container shell but still want to execute any command then instead of bash provide the command name to be executed inside the container
* docker container exec -it 75ce481f4702 netstat -ntlp

Note: if we are trying to execute any command whose binary is not available in the container then it will throw us an error as we should keep remember that the container does not contain full blown operating system it only contains bare minimum OS. And also not always the bash will be the default shell inside the container so if bash doesn’t work then try to find the different shell available inside the container.

Importance of IT flag in docker exec command:

* here the IT are the important argument of docker exec command because ‘I’ helps us to send the command to the interactive standard input in linux OS of container and ‘t’ helps us to show or make available the tty means the terminal if we don’t use ‘I’ flag in command then it wont allow us to execute command and if we don’t use ‘t’ flag in the terminal then it wont show us the terminal though we can execute the terminal so its important to use -it with exec argument in docker container command.
* We can use this flag even separately for example -I -t