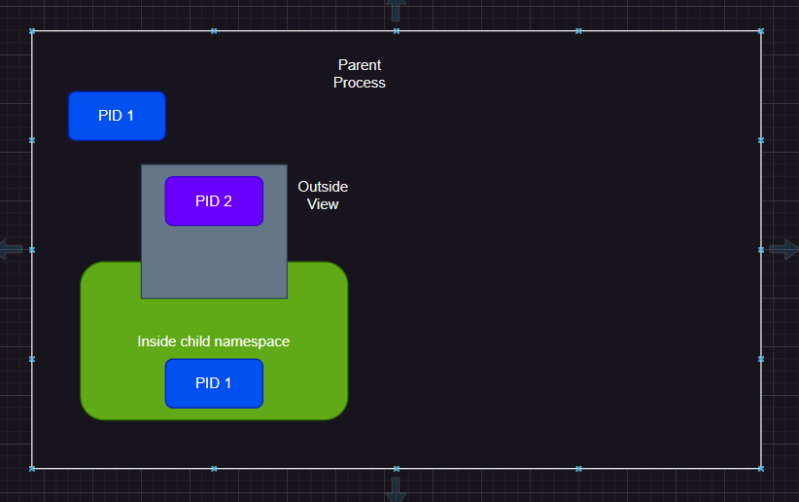
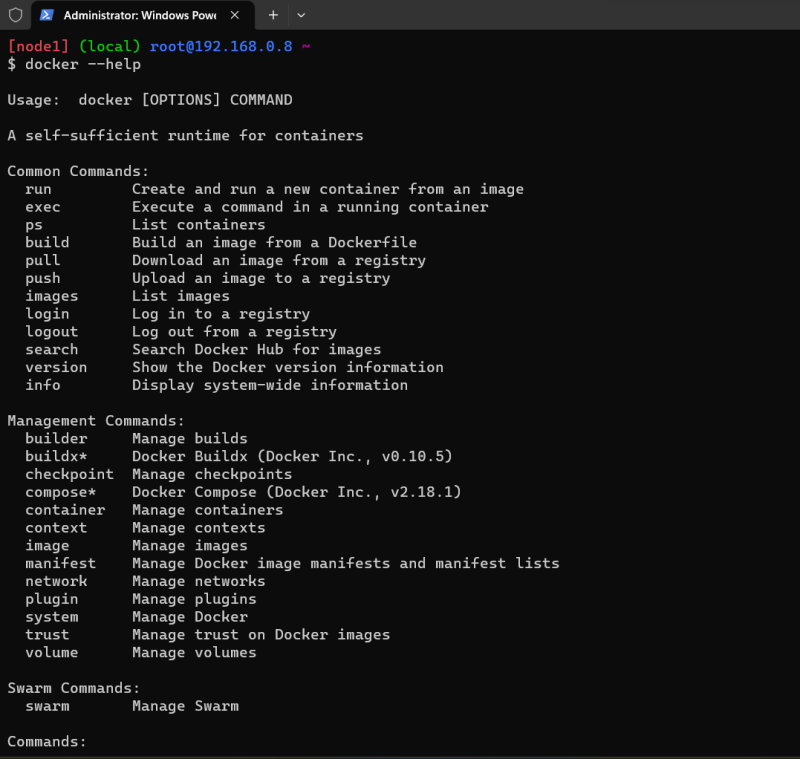
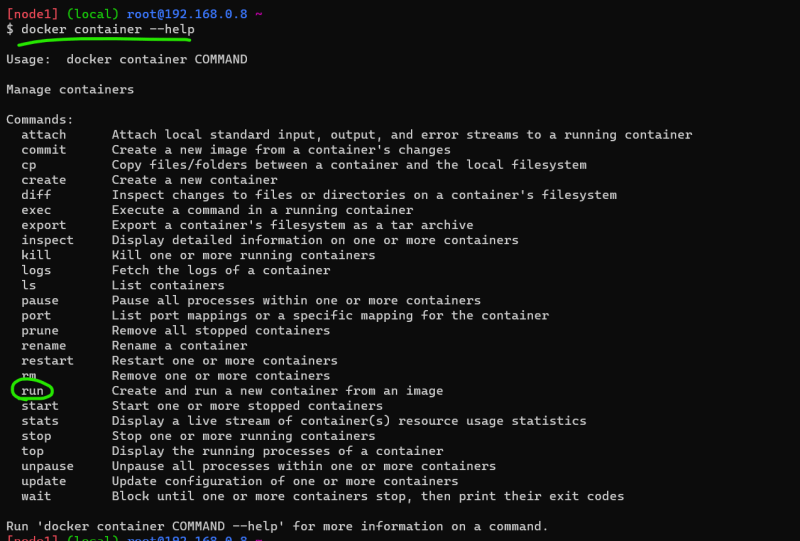
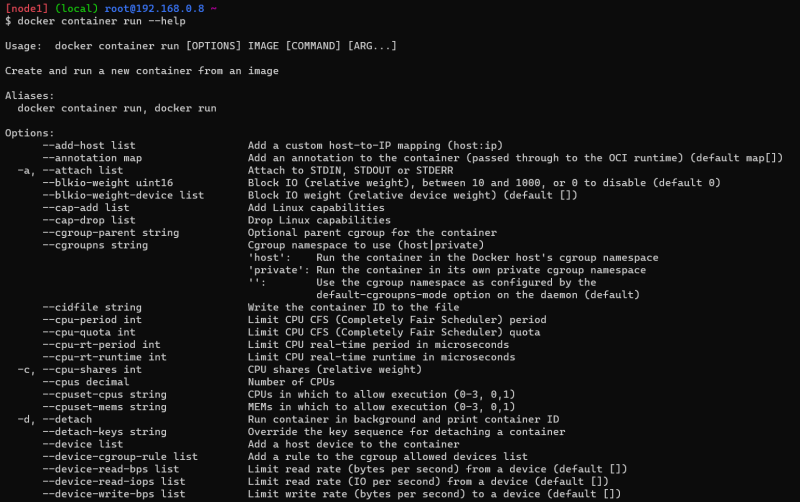
**Namespaces in Linux**

* Namespaces are a feature of Linux Kernel, that partitions kernel resources such that one set of processes sees one set of resources in other words they isolate processes from each other
* Types of Namespaces
  + user namespace: creates its own set of user ids and group IDS in process
  + Process (PID) namespace: Assigns a set of PIDS to process that are independent from parent process  
    
  + network namespace: Create its own private routing table, set of ip address etc
  + mount namespace: Creates independent list of mount points seen within process
  + IPC (interprocess communication) namespace: Creates its own IPCs
* Namespaces are one of the technologies that containers are built on, they are used to enforce segregration of resources. Using container technologies like Docker, rkt, podman makes much easier to create namespaces on your behalf

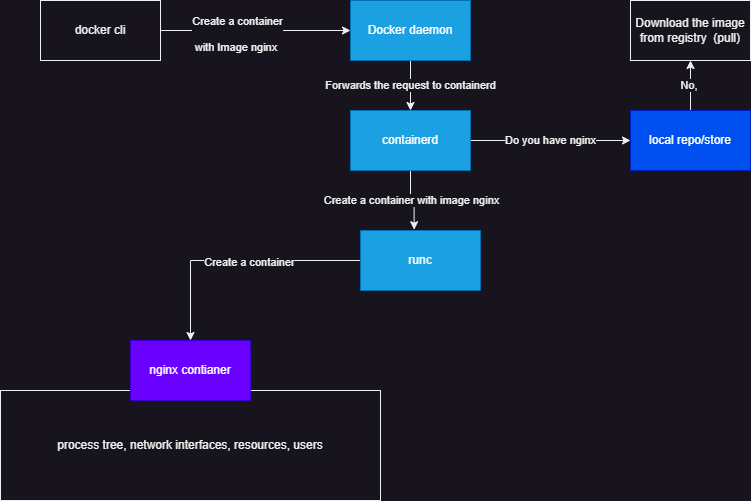
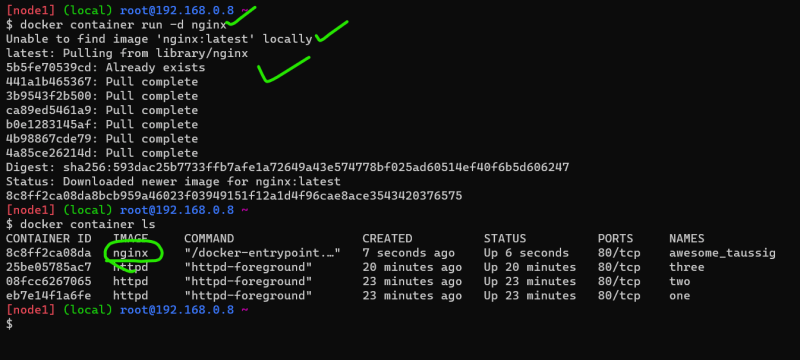
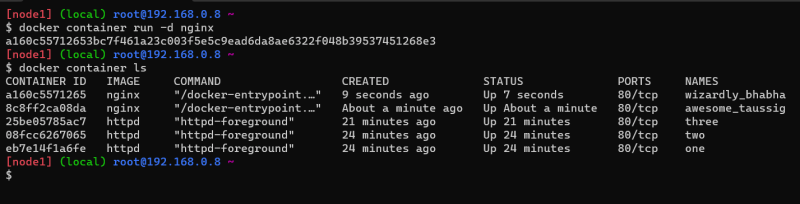
**cgroups (Control groups)**

* This is also a linux kernel feature that limmits, accounts form and isolates the resource usage (CPU,memory, disk I/O, network …)

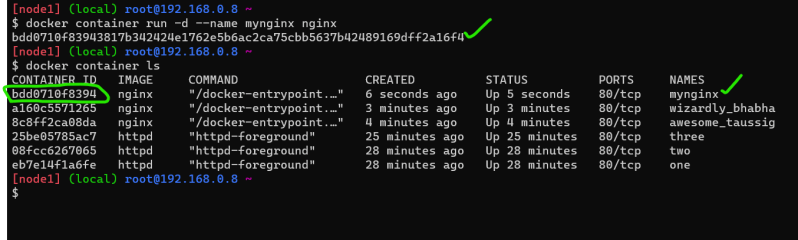
**Play with Docker CLI**

* Docker command line:
  + Use cheatsheet
  + Use help
* Using Help  
    
    
  

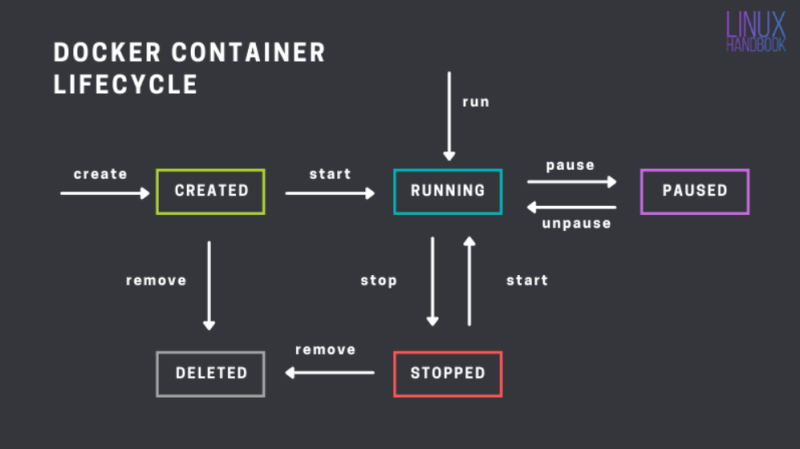
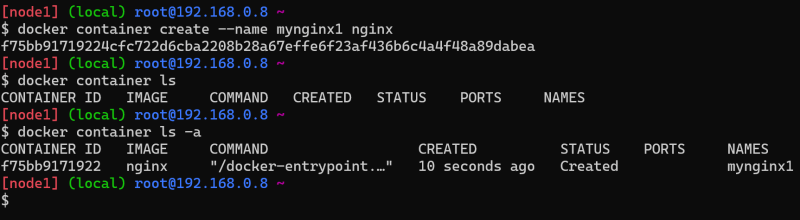
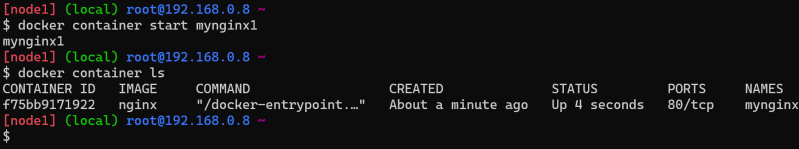
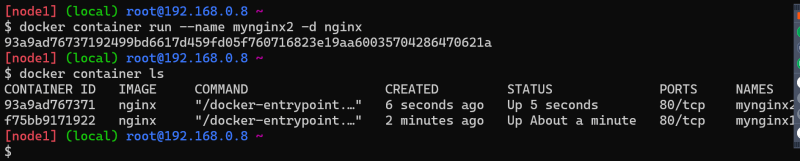
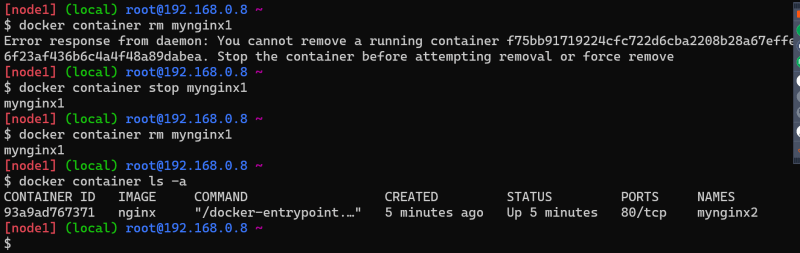
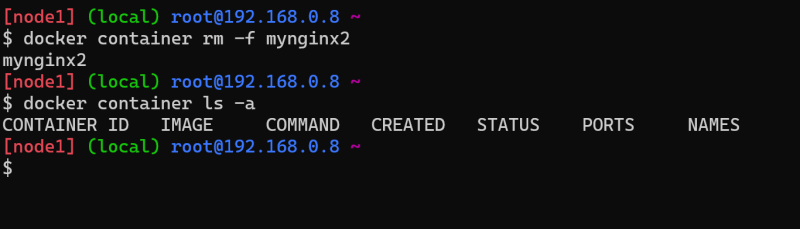
**Creating my first container**

* To create a docker container we need image or docker image
* Docker image is a packaging format which is OCI compliant has all the necessary stuff included to run your application in container
* Docker Hub <https://hub.docker.com/> is a default registry which is collection of docker images. Each image is stored as repository
* Registry is collection of docker image repository. Each repository represents some application and tag of the image represents some version
* Let’s try to run nginx <https://hub.docker.com/_/nginx>
* What happens when we try to run a container  
    
    
  

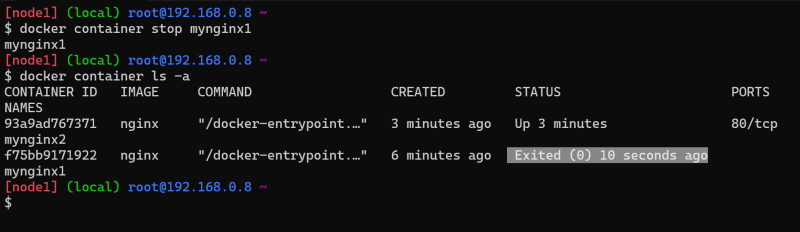
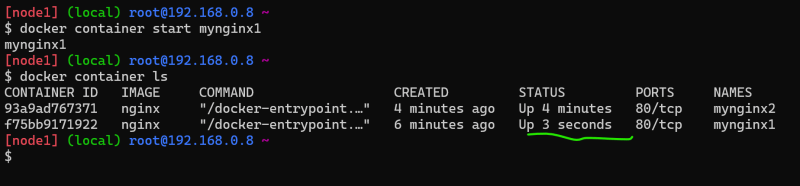
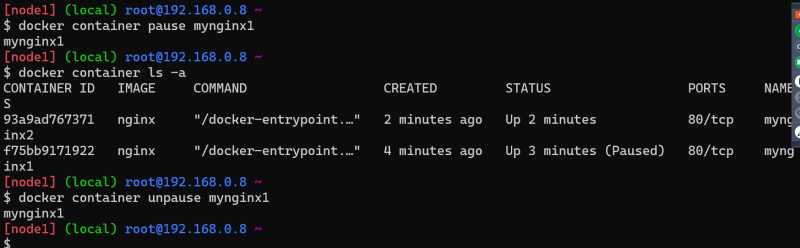
Commands:

* + docker container run (create a container)
  + docker image ls (list all the local images)
  + docker container ls (running containers)
* Every container that is created will have unique id and name. id is generated by docker, where as name is generated only when you dont pass it  
  
* To interact with containers we need id or name

**Docker Container states**

* Refer the below from Linux Handbook  
  
* Create a container and then start it  
    
  
* Run the container to create and start  
  
* Running container can be
  + stopped
  + paused
  + deleted by using force  
      
    

Stopped container can be

* + started
  + deleted  
      
    
* Paused container can be
  + unpaused
  + deleted  
    

**Microservice-Design Problem**

* We have discussed about publish and subscribe (an intro)  
  