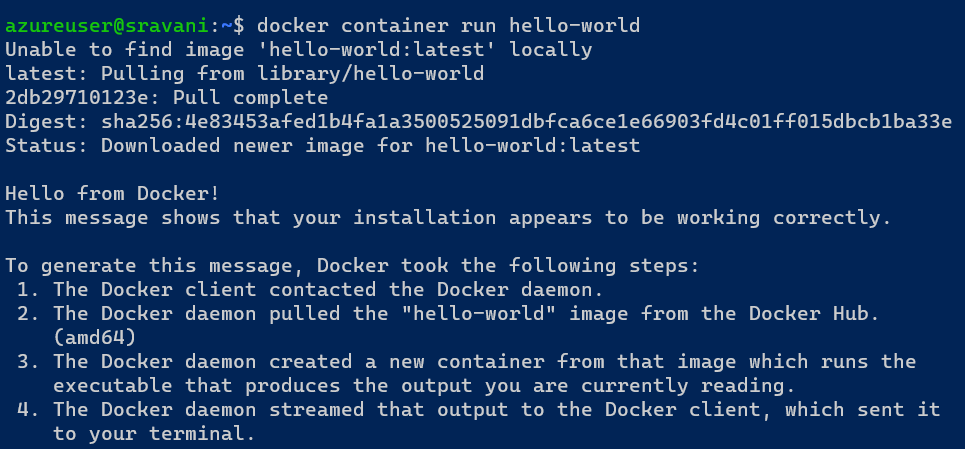
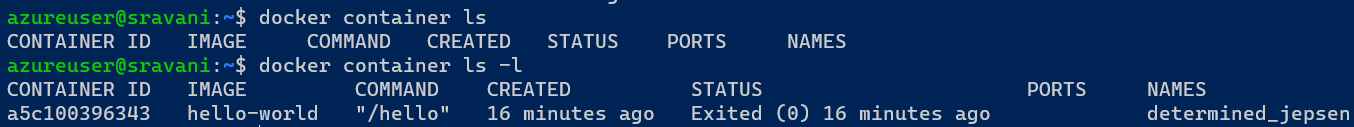
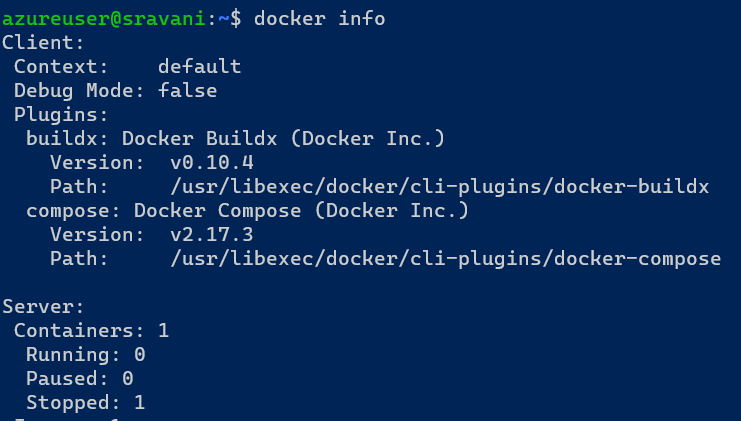
1. Run hello-world docker container and observe the container status

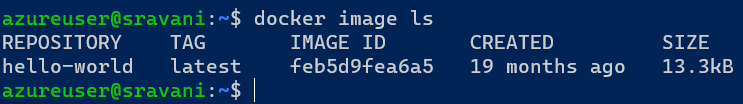


Docker container status

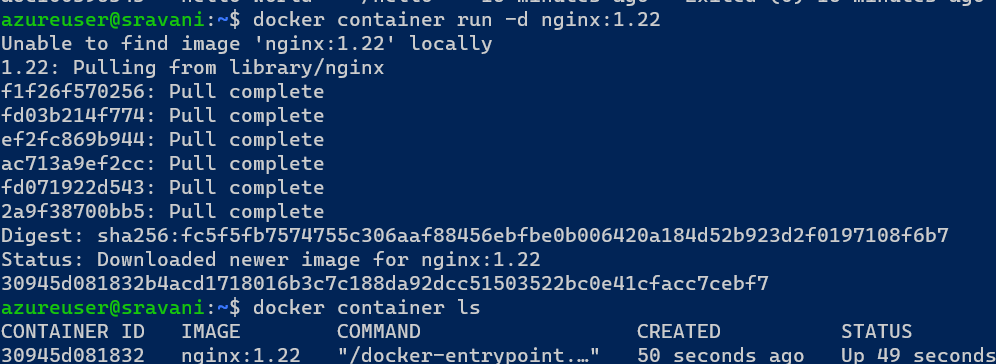


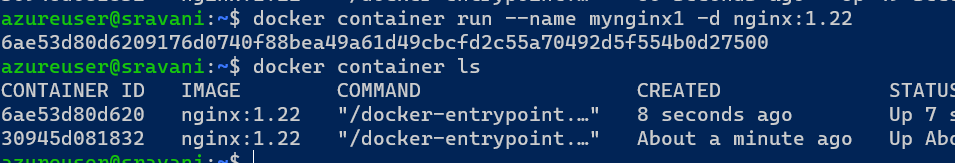


1. Check the docker images and also write down the size of hello-world image

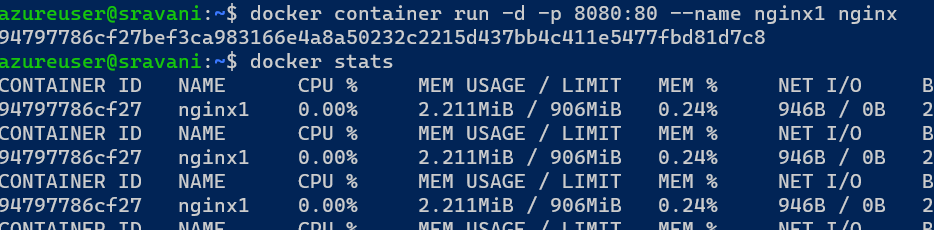


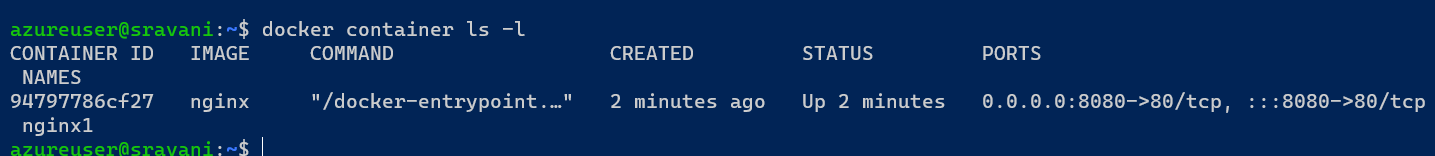
1. Run the nginx container with name as nginx1 and expose it on 8080 port on docker host





expose it on 8080 port on docker host





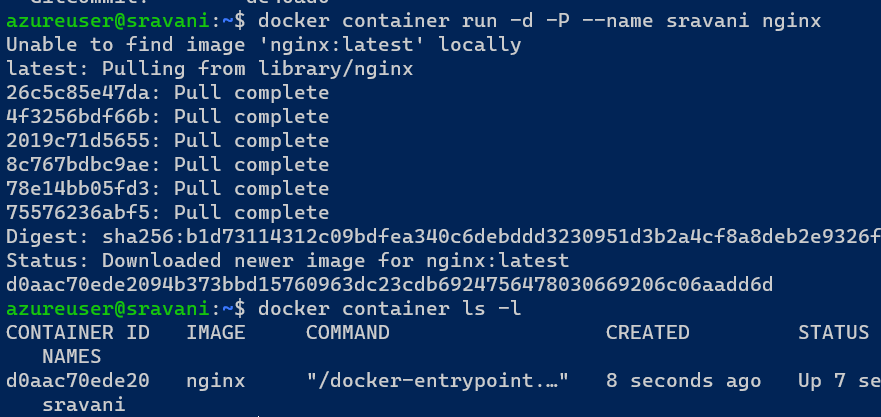
1. Explain docker container lifecycle

Docker lifecycle has 6 states they are

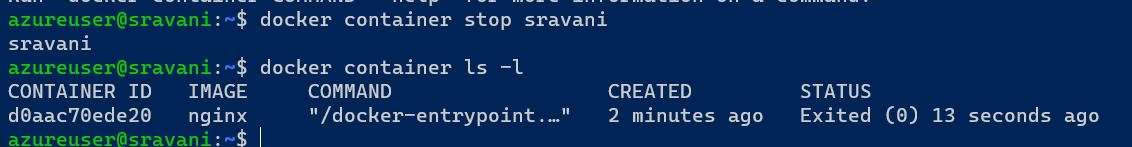
1. Create
2. Run
3. Stop
4. Pause
5. Unpause
6. Delete

Above states explained with example below

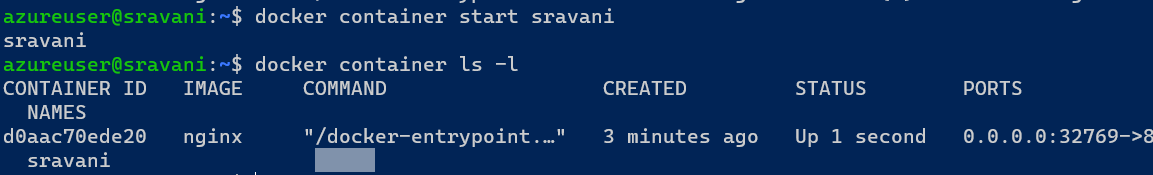
To create and run



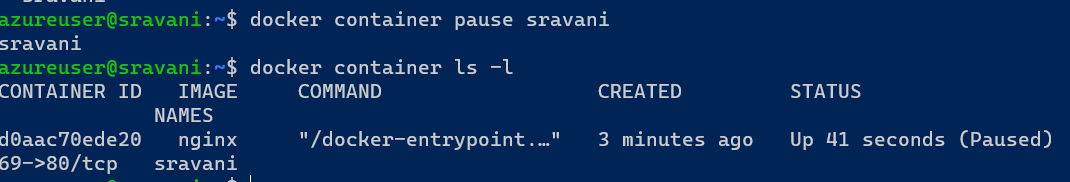
To stop



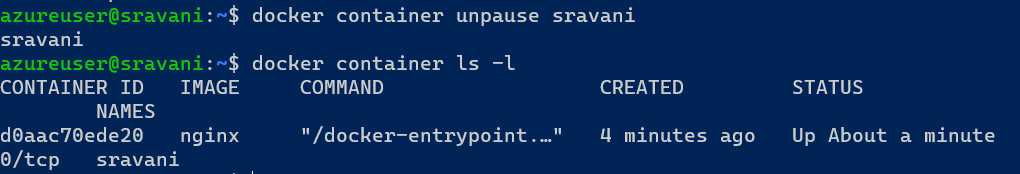
To start



To pause

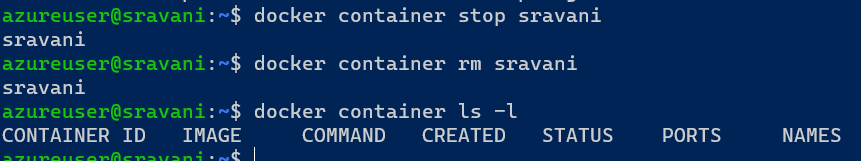


To Unpause



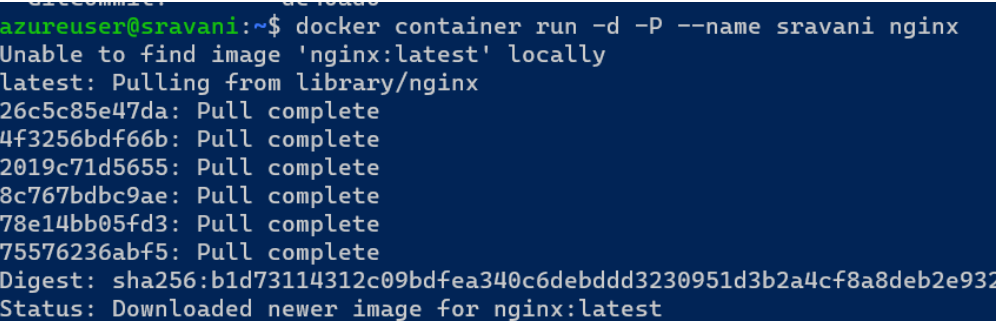
To delete

If container I running it is not possible to delete so we will stop the container and delete



1. Explain what happens when you run the docker container

The docker run command creates running container by using docker images and can run commands inside them. When using the docker run command, a container can run a action.



1. Explain the Docker Architecture

Docker architecture is also called as client architecture. Docker architecture talks about docker daemon. Means heavy lifting buildings running and distributing your docker containers. There are three parts in architecture

* Docker host
* Docker network and storage compound
* Docker hub

In docker architecture will be explained in three generations

Generation 1:

Docker client send the request to docker daemon were commands interact with docker.It has to be relay on to create lxc (Linux)container.this internally uses name spaces and c groups.It is inbuilt kernel is Linux.

Generation 2:

In this generation updating kernels and frequently creates lib containers.lib container is docker component. It has own code to run the applications.Libc directly speak with kernel.

Generation 3:

In this generation docker client specks with docker daemon.Docker daemon expose api and it speaks with containerd. This containerd creates container by any container technology that is OCI. The containerd speaks with runc.