**Docker Assignment 2**

1. Difference between virtualization and containerization?

|  |  |
| --- | --- |
| Containers | Virtual Machine |
| There will be Hardware, on top of that will be OS, on top of that will be docker, on top of that will be multiple containers. Inside containers will be some librataies and dependencies. | There will be Hardware, on top of that will be OS(eg. Hypervisor/vexsi), on top of that will be multiple virtual manchine. Now this virtual machines will now again has it's own OS in each virtual machines. No on top of the VM's OS containers will be running |
| In side this containers Resouce utlization is less | Resouce utlization will be more |
| Disk utlization will be less ( in MB's) | Since in each VM there willbe vitual OS so Disk utlization will be more ( In GB's) |
| Fast boot time | Slow boot time |
| Uses exceution engine | Use hypervisor |
| Deployement is easy | Deployement is lenghty |

1. What is a Docker hub?

* Docker Hub is docker registry where we save our images.
* Docker Hub is a public registry that anyone can use, and Docker is configured to look for images on Docker Hub by default. You can even run your own private registry.
* Docker Hub is a service provided by Docker for finding and sharing container images with anyone may be with your team. It is the world’s largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers..

1. Tell us something about docker-compose?

* [Docker Compose](https://docs.docker.com/compose/) is a tool that was developed to help define and share multi-container applications. With Compose, we can create a YAML file to define the services and with a single command,
* eg.. docker compose up
* let's say in our stack we have four containers running

First continer which has flask application, second is DB-cassandra,

Third conatiner has Messaging app IBM MQ, fourth as Ansible

To run all these we will use commands

docker run *webflask\_app*

docker run *cassandra\_db*

docker run *IBMMQ\_messaging*

docker run *Ansible\_automation*

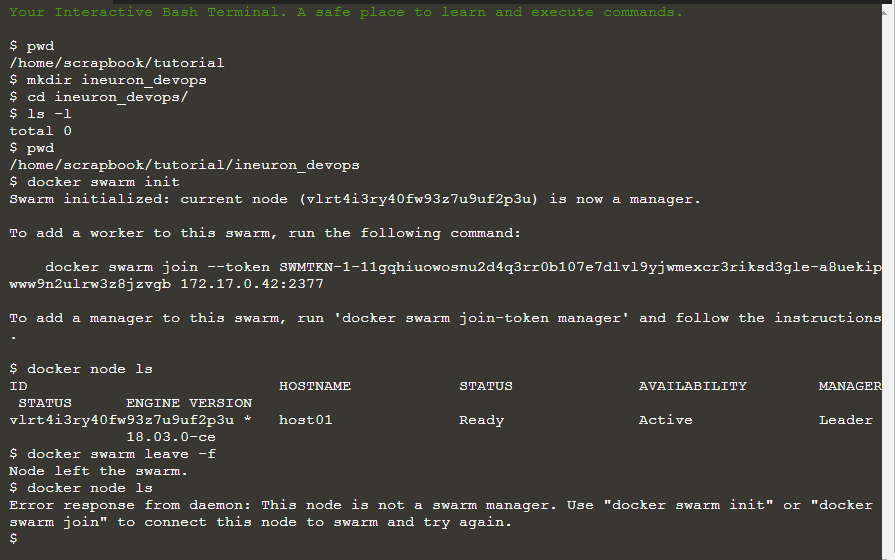
* Now in docker-compose.yml file we can mention all above container name and which we just want to run in one go with this .yml file.  
  Inside below **docker-compose.yml** you can add below content. Note - pls take care for indentation as yml we have to follow there indentation else can face some issue.

|  |
| --- |
| **services**:  **Webapp**:  image:"ronak/mywebapp"  **db**:  image:"cassandra"  **message**:  image:"IBM\_MQ"  **Orchestration**:  image:"ansible" |

* There are 3 versions in docker compose
* Docker compose V1, Docker compose V2, Docker compose V3
* The compose file are yaml file defining services, networks and volume for a docker application.
* Version 1 is already deprecated.

1. What is the docker swarm?

* Let us see first see container orchestration this will basically give us an idea of what docker swarm is
* orchestration tool is to help us deploy our application on a large number of containers
* orchestration tool will help us to launch all of those containers in one go all we have to define is what do we want in those containers what kind of image we need to be creating
* Docker swarm is a container orchestration tool part of the docker engine.
* Docker Swarm is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host.
* With it developers and IT administrators can deploy and manage cluster of docker nodes as a single virtual system
* docker swarm you can use it to deploy and manage a cluster of docker nodes now these docker nodes will basically be docker containers these nodes will be the servers of their own and you can manage all of these things together as a single virtual System
* Architecure would be like something like it will have a **single manager node** and you have different **multiple worker nodes** which are connected to the manager node and each of these will have a particular task to perform. worker nodes will have tasks like you know implementing in new features or something and the manager node will have the special task of managing all of these worker nodes and this manager will be handled by services which in turn is managed by a docker engine
* Few of the Feature are as below
* Decentralized design
* Declarative service model
* Scaling
* Desired state reconciliation
* Multi-host networking
* Service discovery
* Load balancing
* Secure by default
* Rolling updates

A Small poc as in below snap  


1. Explain the lifecycle of the docker container?

Docker container lifecycle consits of below multiple stages

* 1. **Create phase**

To create a docker but not start

**docker create - -name container\_name image\_name**

* 1. **Start Container**

To start a stopped container, we can use the docker start command.  
 **docker start container\_name**

* 1. **Running phase**

A container running with all its processes

**docker run -it - -name container\_name image\_name**

* 1. **Paused phase/unpause phase**

A container whose processes have been paused/unpaused

**docker pause container\_name**

**docker unpause container\_name**

* 1. **Stopped phase**

To stop any container

**docker stop container\_name**

**docker stop $(docker container ls –aq)**

* 1. **Remove/Delete phase**

A container needs to be first stop then remove/delve

**docker rm <container name>**

* 1. **kill phase**

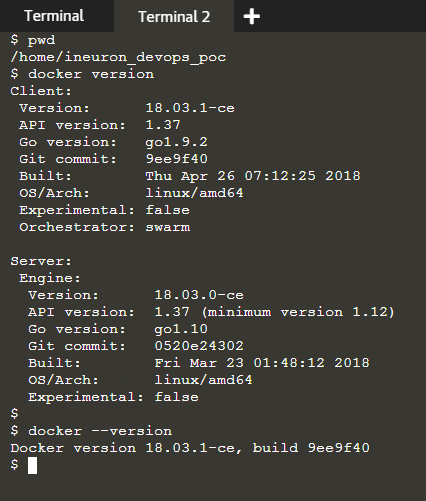
We can kill containers with below comamnd

**docker kill <container name>**

1. How to check the docker client and docker server version?

Use command : **docker version** to get below information we can also use

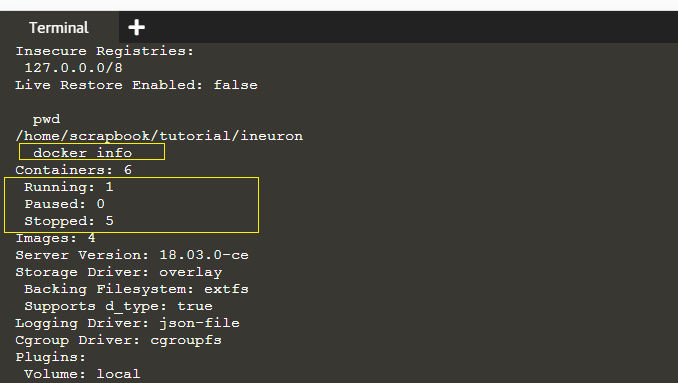
**docker --version**



1. How do you get the number of containers running, paused, and stopped?

we can use **docker info** we can get the number of containers and its state

just for additional info - if we only want all state of conainer try docker ps -a



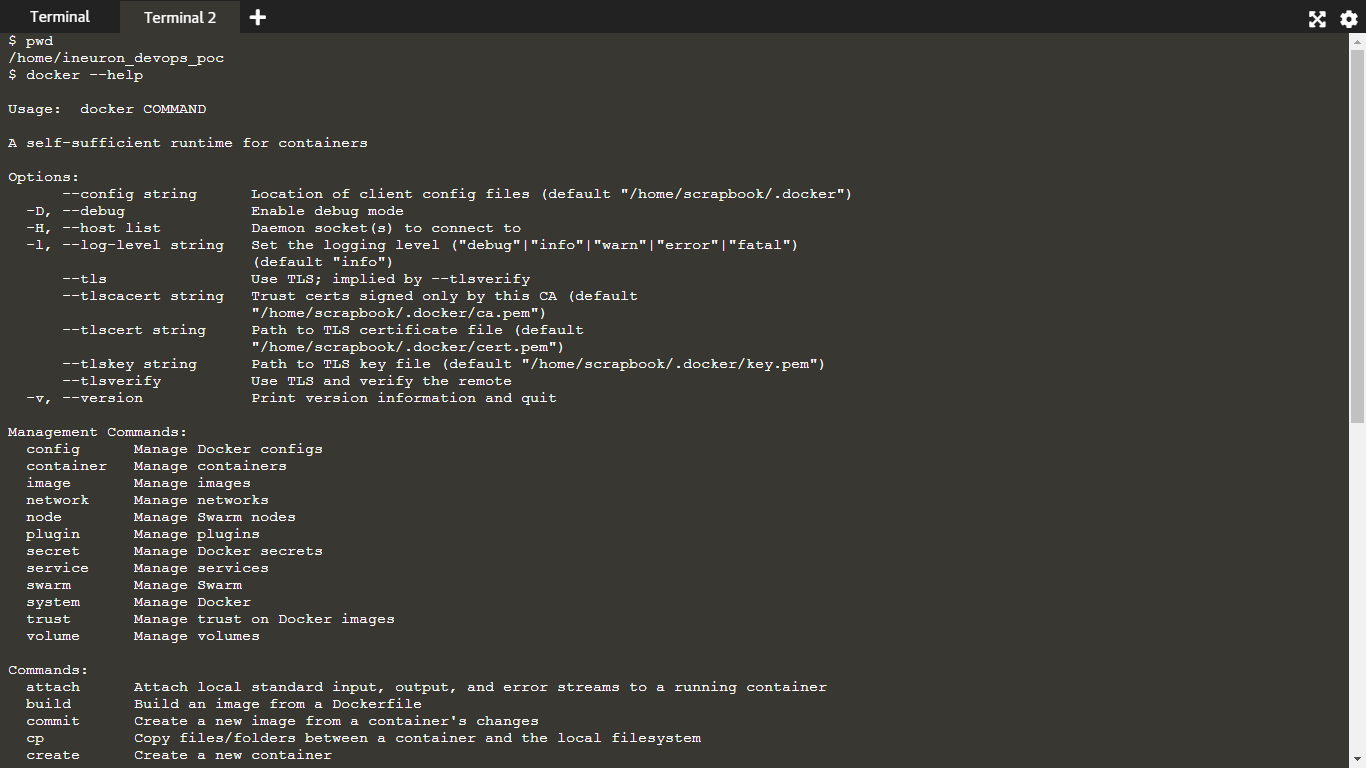
1. If you vaguely remember the command and you’d like to confirm it, how will you get help on that particular command?

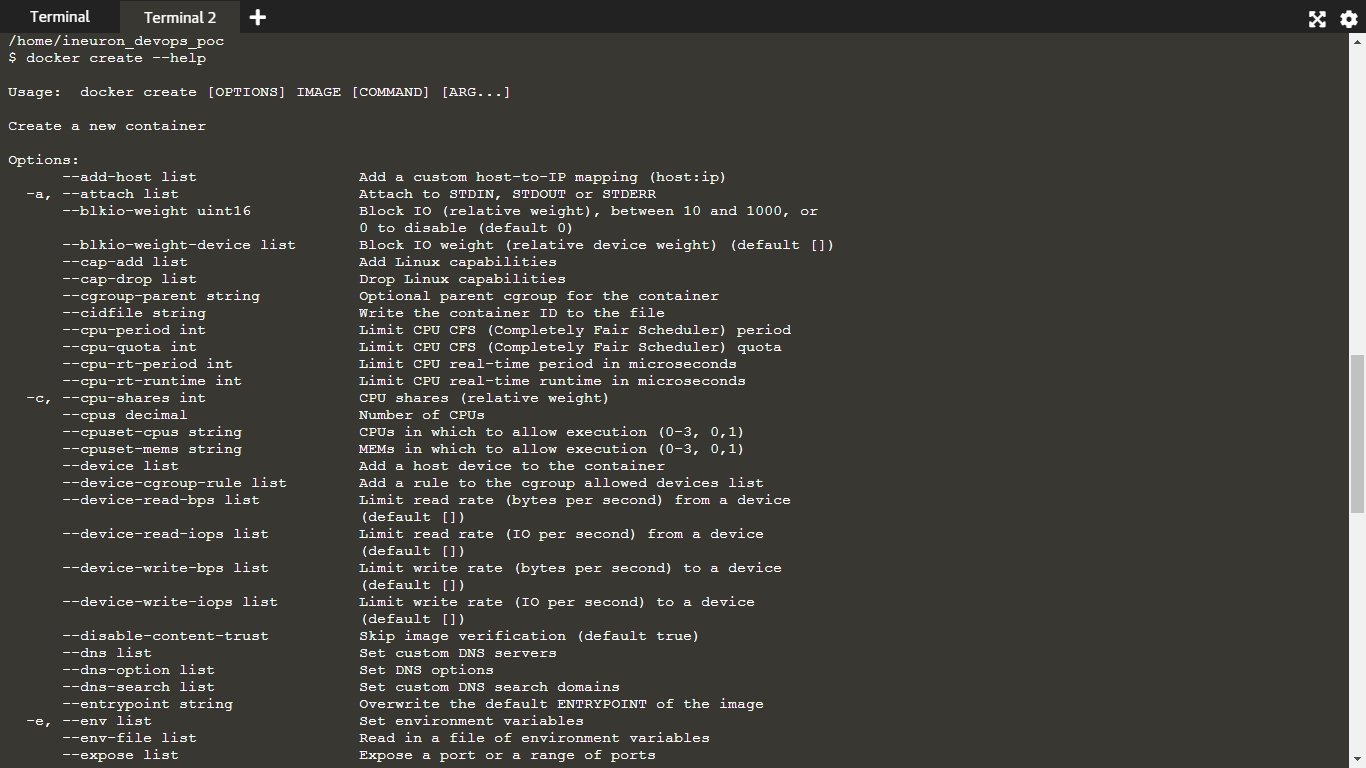
The below command is very useful as it gives you help on how to use a command,

**docker --help**

To get help on any particular command then use

**docker command --help** eg. **docker create --help**





1. How to log in to the docker repository?

We can use comamnd - **docker login**

**docker login localhost:8080**

1. How do you create a docker container from an image?

* There are 2 ways: only create and start later, or create and start at same time
* Command to only create a container  **docker create image\_name**
* you can check the conatiners which are in running mode with command **docker ps**
* since above created docker is not running we can check all docker conatiners state by using command **docker ps -a**
* to start you can give command **docker start container\_name**
* Will create & run at a same **time in interactive mode docker run -it image\_commands**
* **eg. docker run ubuntu sleep 10** -> this will create a container from ubuntu and run the command sleep for 10sec.. this is in **attached** mode
* **eg. docker run -it -d ubuntu sleep 10** -> this will be in **detached** mode
* let say you want to map the cotainer to external you can do so with

**docker run -p app\_name**

**docker run -p 8000:5000 app\_name**

* eg. now you want to mount the disk or keep data external for eg. you have mysql as app inside container that will keep all data in containers path only like /var/lib/mysql so you want to keep them in external outside of continainer or host then can do that

**syntax:**

* **docker run -v <path of external host> <path of/inside container> app\_name/image\_name**
* **docker run -v /home/ubuntu/data /var/lib/mysql mysql**