**Docker Cheat-Sheet**

* sudo usermod -aG docker $USER : to run command as a non-root user.
* docker version : shows the installed docker version
* service docker start : start docker service

***Basic info***

* docker ps : lists only running containers
* docker ps -a : shows all containers includes created, running, exited.
* docker images -a : lists all images created
* docker logs : shows the logs of container
* docker inspect : view detail information about a image or a container
* docker stats : view resource usage statistics for one or more containers
* docker port : list the port mappings for a container
* docker top : to view the processes inside a container

***Manage images and containers***

* docker build -t [tag name] [directory] : build a new docker image
* docker run [image] : starts a new container from an image
* docker run --name [container] [image] : assign a name of container
* docker run -d [image] : starts a container in backend
* docker run -p [Hostport:Containerport] [image] : map all ports to container
* docker run -it [image] : interact with the image through command line
* docker run -e [my Var=my prop] [image] : specify an environmental variable for a docker container
* docker exec -it [container-ID/name] [executable] : start a shell or entering inside a running container
* docker rename [old name] [newname] : rename the container
* docker save [image] > [archive file] or
* docker save -o [archive file] [image] : save an image to a tar archive
* docker commit [image] [image name] : save a running docker container as an image
* docker load -i [archive file] : load an image to a tar archive
* docker start [container] : start the exited container
* docker stop [container] : stop the running container
* docker kill [container] : forced shutdown of running container
* docker rm : delete the exited container
* docker rm -f : deleted running container forcefully & dangling containers
* docker rm $(docker ps -a -q) : removes all running container
* docker rmi [image] : removes a docker image
* docker rmi $(docker images -q) : removes all docker images
* docker system prune : removes all dangling containers, unused images and containers
* docker login : login cli session with registry like Docker hub using credentials
* docker remote -v : lists out remote docker host
* docker push [username/repository: tag] : upload images to registry
* docker tag [image] [username/repository: tag] : set the tag for images pushed to Docker hub
* docker pull [username/repository: tag] : download images from registry

***Docker-Compose, Volumes & Network***

* docker compose build : build containers running from a directory of your docker-compose.yml file
* docker compose up -d : start multiple containers at once
* docker compose down : stop all running containers and also remove them
* docker compose logs : shows the logs of running containers with docker compose
* docker compose -d --scale up [service]=[no. of times] : scale up the services or containers with limited times as prescribed in Yaml file
* docker service [service] --replicas [no. of times] [image] : autoscaling services with set number of times
* docker volume create [volume] : create a volume in the machine
* docker run -v [HostDir:TargetDir] -it [container] [image]: connect a container to a volume or mapping local directory with a docker container
* docker volume rm [volume] : remove unused volume from machine
* docker volume ls : lists all volumes
* docker volume inspect [volume] : inspects the volume

***Docker Swarm***

* docker swarm init : enable first node of docker system
* docker swarm join --token [token] --listen -addr [ip:port] : add a node to a swarm cluster
* docker swarm join-token : retrieve the join token
* docker node ls : list nodes in a cluster
* docker node rm [node-name] : remove a node from swarm cluster
* docker service ls : list services in docker swarm
* docker service ps [service] : list containers in a service
* docker service rm [service] : remove a service