**DOCKER Mastery: with kubernetes**

**Commands:**

*docker version* # Provides version of the docker container

*docker info* # provides the brief level of information about the docker client which is installed

docker command -> OLD WAY

docker command sub-command [ARGS] -> New Way

**Docker Machine/Compose/Swarm**

Docker Machine is a single container/machine

Docker Compose is a multilevel of docker container

Docker Swarm is a multi-multi of orchestration of both Docker Machine and Docker Compose

**Image Vs Container**

Image is the application that we want to run

Container is the running instance of the image

Single image in a multiple container

docker hub is used as registry

*docker container run –publish port:port image\_name*

--publish is used to open port

**e.g. docker container run --publish 80:80 nginx**

**80:80 -> local to the machine and environment**

**8080:80 -> run 8080 in local**

**if two ports are on the same local it will fail**

*docker container run --publish 80:80 --detach nginx*

**--detach is used to run in background**

docker container run -p 80:80 -d --name test\_box nginx

**Logs from a Container**

*docker container logs container\_name*

**List out the Process running in a container**

*docker container top container\_name*

***List the running container***

*docker container ls # only running container*

*docker container ls -a # All container*

**Delete Container(s)**

*docker container rm -f container1 container2 container3*

***Container Vs Images***

Container are Just a process nothing else

*docker container inspect container\_name*  # Brief level of what is happening inside a container

# or Information about the container like Network

*docker container stats container\_name*  # Statistics about the container like CPU/MEM usage

**SSH into a Docker Container**

*docker container -it --name bleh container\_name*

*-it is for interactive terminal*

but the terminal will go off as soon as you come out of the terminal window, so restarting the terminal should be choice here

*docker container start -ai container\_name # To login to a Stopped Container*

*docker container exec -it container\_name # to execute something in docker container*

**Docker exec Vs run**

Docker Container has a number of option where the user can login to the container using SSH and then manipulate something on the whim of the User

but docker exec and docker run work in a different way of one another,

docker run creates a image of and helps you to login into the container

docker container exec will login to the existing image and work

exec can be used so that you can start working where you left off on the other hand docker container run will destroy the old environment and rebuid a new one.

*docker container port container\_name # tells us about the ports the container is using*

**Filter Just Like working with JSON**

*docker container inspect --format '{{ .NetworkSettings.IPAddress }}' webhost*

**Network in Docker**

*docker network ls*  # Lists out the networks in the system. such

*root@nurawat:~# docker network ls*

*NETWORK ID NAME DRIVER SCOPE*

*41ab53b95282 bridge bridge local*

*21d2a7f66c13 host host local*

*cabc207d1063 none null local*

*docker network create network\_name*  #this way a new network can be created which can be used to create network bridge beween two container

*docker network inspect network\_bridge\_name*

This will list out all the container/process which are using the particular network which are using the network `*network\_bridge\_name*` *inside it*

*docker network connect/disconnect <network\_bridge> <container1> <container2>*

the above commands attaches/detaches the ***network\_bridge***  to and from a particular container

Docker Networks: Default Gateway

1. Create your apps so frontend/backedn sit on the same docker network
2. their inter-communcation never lesaves host
3. all texternally exposed port closed by default
4. you must manually expose via -p, which is better default scurity
5. This gets even better later ith swarm and overlays