Docker Toolbox IP address : 192.168.99.100

**Docker Command Format:**

docker <command> <sub-command> <options>

**Image vs Container:**

An Image the application we want to run. It is the Binary, Library and Source Code that all make up our application.

A Container is an instance of that image running as a Process.

We can have many containers running off the same image.

Docker’s default image “registry” is called Docker Hub (hub.docker.com)

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

**docker container run –-publish 80:80 –-detach nginx (**run in background**)**

**docker container run –-publish 80:80 –-detach –-name pankaj nginx (**give container name by ourself**)**

1. Download image ‘nginx’ from Docker Hub
2. Started a new container from that image
3. Opened port 80 on the host IP
4. Routes that traffic to the container IP, port 80. (You will get a “bind” error if the left number [host port] is being used by anything else, even another container. You can use any port you want on the left, like 8080:80 or 8888:80, then use localhost:8888 while testing)

**docker container ls**

Show all running container

**docker container ls -a**

Show all container

**docker container stop <container ID>**

Stop running container

**Run vs Start**

Run always start a new Container

Start always start an existing Stopped Container

**docker container logs <container-name>**

To see logs of container

**docker container top < container-name>**

Display running processes of a Container

**docker container –-help**

See all command

**docker container rm <container-id> <container-id> <container-id> <container-id> <container-id>**

Remove docker Container normally. This command can delete only stopped container, it can’t delete running container.

**docker container rm <container-id> <container-id> <container-id> <container-id> <container-id>**

Remove docker Container Forcefully.

**What happens in “docker container run”**

1. Looks for that image locally in image cache, doesn’t find anything
2. Then looks in remote image repository (default to Docker Hub)
3. Download the latest version (nginx: latest by default)
4. Creates new container based on that image and prepare to start
5. Gives it a virtual IP on a private network inside docker engine
6. Opens up port on host and forwards to port in container
7. Starts container by using the CMD in the image Dockerfile

**Container vs VM**

1. Containers aren’t Mini-VM’s
2. They are just process
3. Limited to what resources the can access
4. Exit when process Stops

**ps aux**

Show all running process

**ps aux | grep ‘pan’**

Show all running process and filter also

**docker container run –d –p 3306:3306 –name db –e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes mysql**

Run MYSQL container on port 3306 and setup the Environment path also