Setup

- After creating a new VM, connect to it through powershell through ssh.
- INSTALL DOCKER :
  - `curl -fsSL https://get.docker.com -o get-docker.sh`
  - `sh get-docker.sh`
- Now type `docker info` and check whether are you able to see both client and Server versions ??
  - if NO :
    - we need to add our username to docker group, we can add by using below command.
    - `sudo usermod -aG docker Krishna`
    - after doing this exit from server and re-connect to it for effective results. `exit`
    - after re-connecting to it type `docker info`, now you will be able to see both Client and Server versions.

View Docker images and Containers

Images:

- To view Docker images in local repository
  - `docker image ls`
- To get list of image id's
  - `docker images -q`
- To delete all images in single shot
  - `docker rmi $(docker images -q) -f`

Container:

- To view Docker continer in local repo
  - `docker ps -a`
- To list of container id's
  - `docker ps -q`
- To delete all containers in single shot
  - `docker rm $(docker ps -q) -f`

download images

- To pull images from online repos that is Docker hub
  - `docker image pull _image-name_`
- If you want to download specific version of image you need to mention tag[:*tag-name*]
  - `docker image pull nginx:perl`

Create and Run container

Create and run container directly in single command

- `docker run -d -P nginx` [name and port will be assigned by docker]
- `docker run -P httpd` [image will be created and will be on exited state, you can manually run the container]

Create Image and then create container with image

- Create Image/pull image
    - `docker image pull nginx`
- Create Container with ngnix
    - `docker container create --name abcd -P nginx`
- Now start container through id/name
    - `docker container start abcd`
- Create and Run container at a time and by assigning specific port
    - `docker container run --name nginx1 -d -p 5000:80 nginx`
    - To get inside of this container to use bash
        - `docker exec -it nginx1 bash`

Run JAVA Application in container

- we have image in docker hub with pre installed java , i choosed `amazoncorretto:17`
    - Pull image
        - `docker image pull amazoncorretto:17`
    - Create container and run it : *create and then run* : *create*: * `docker container create --name java2 -p 5003:8080 -it amazoncorretto:17` *Run*: * `docker container start java2` *Create and run in single shot*: * `docker container run --name java2 -d -p 5003:8080 -it amazoncorretto:17`
    - *Get inside of container to use bash*:
        - `docker exec -it java2 bash`
    - *Download the java Application*:
        - ` curl -O *link*
    - *Run the JAVA application*:
        - `java -jar _filename_`