**Docker Engine installation on Amazon Linux:**

***sudo yum update –y***

***sudo amazon-linux-extras install docker***

***sudo yum install -y docker***

***docker --version***

***sudo systemctl enable docker***

***sudo systemctl start docker***

***Note: add the normal user to docker group to provide docker access to normal user***

***usermod –aG docker <username>***

**Docker Commands:**

1. **docker --version:** version of docker engine installed
2. **docker login:** login with username and pwd to the dockerhub/organizationRepo(ECR)
3. **docker search <imageName>** : search the images on hub
4. **docker ps** : shows the running containers only
5. **docker ps -a** : shows all the container
6. **docker images:** shows the list of local images
7. **docker pull <imagename>** : pulls the image into local
8. **docker create --name <name> <imageID> or <imagename>:<version>** : creates a docker container for the image
9. **docker start <containerID**> : starts the container
10. **docker stop <containerID>** : stops the container
11. **docker kills <containerID>** : kills the container
12. **docker run -d <imageID> or <imagename>:<version>**: (pull+create+start) container
13. **docker rm < containerID>** : removes the container
14. **docker rmi <imageID> or <imagename>:<version>** : removes the image
15. **docker images -a -q** : to get all the image Ids
16. **docker rmi ‘docker images -a -q’** : removes all the images
17. **docker run -d -p <hostport>:<containerPort> <imageId/imageName:version**> : configure Ports + run the container
18. **docker run -d -p --net host <imageId/imageName:version**> : configure with Host Netowrk + run the container
19. **docker run --name mongodb -d -p --network=<NetworkName> <imageId/imageName:version**> : configure with own custom network + run the container
20. **docker network create <networkName> :** creates a own custom network(bridge)
21. **docker run -d -p <hostport>:<containerPort> --name <containerName> <imageId/imageName:version>** : Name a container + Configure ports + run container
22. **docker exec -it <conatinerId/ContainerName> bash** : get inside a container
23. **docker inspect <containerId/ ContainerName>** : info of container(including the IP)
24. **docker logs <conatinername>**
25. **docker build -t <imageName>:version .** – generates an image from the dockerfile
26. **docker rename <oldContainerName> <newContainerName>**