**DOCKERS**

**DOCKERS**

Dockers is a build and deployment tool the main idea is we can put together all the required dependencies installation commands together the code so that it installs the dependencies first to run the application. We can automate this by putting installs the dependencies first to run the application. We can automate this by putting inside a .yml file and specify the actions.

**CONTAINERS**

Containers follows the same concept of classes in java we can put many images inside it, these containers can contact with other and also contact with external applications.

**IMAGES**

Images follows object concept of java we can add many images inside the containers, we can download images from the Dockerhub.official images has official tags in the hub.

**REPOSITORY**

Repository is a collection of different docker images with the same name, that have different tags, each tag usually represents a different version of the image.

**TYPES OF DEFAULT NETWORK IN DOCKERS**

1) None network

2) Bridge network

3) Overlay network

**None Network**

It provides the maximum level of network protection. It not a good choice, if network or internet connection is required. Suites well there where the container require the maximum security and network access is not necessary.

**Bridge Network**

In a bridge network, containers have access tom two network interfaces.

* A loopback interface
* A private interface

All containers in the same bridge network can communicate with each other.

**Overlay Network**

It supports multi-host networking out-of-the-box.

It requires some pre-existing conditions before it can be created.

* Running Dockers engine in Swarm mode
* A key-value store such as consul.

**BASIC COMMANDS**

**docker ps**  - We can can get running containers using this

Commands

**docker images**  - See all the images

**docker run image\_name** - Run the images

**docker run -d image\_name** - Run the images in background

**docker run pd image\_name:tag** - We can run the specific version of the image

using the command

**docker run -d --name Image\_name my\_name** - We can give a name for the Images

**docker pull image\_name** - We can pull image from the hub using this Command

**docker start -d container\_name** - Starts the container

**docker stop** - Stops the running container

**docker logs container\_name** - See the logs of the particular container

**docker push image\_name** - Pushes the image of the container to the docker Hub

**docker swarm init** - It initialize a swarm. The docker engine targeted by this Command becomes a manager in the newly created single-Node swarm

**docker swarm join** – Join a swarm as a swarm node

**docker swarm leave** – Leave the swarm