Course 4

**CB Full Stack - Integration and Deployment**

**Docker**

Creating image for angular application

ng new angular-docker

routing -🡪 no

styling 🡪 css

after project created we need to build the project

ng build

inside a project it will create dist folder which contains another sub folder as a project name and inside that folder contains all build file.

Nginx : it is a type of open source server which is also known as engine-X. which help to deploy View technology can be html, css, js, angular or react js.

Dockerfile

FROM nginx

COPY /dist/angular-docker/ /usr/share/nginx/html

Now you can create the image

docker build -t my-angular-app . -f Dockerfile

nginx server run on port number 80.

How to publish the image

1. We need to connect local machine to Docker hub account.

Through terminal or command prompt write the command

docker login

1. Before publish the image we need to create the tag for that image. Tag is is just like identity for that image.

docker tag imageName dockerHubAccountId/imageName:tag

docker tag my-angular-app akashkale/my-angular-app:v1

1. After tag created you can publish the image in docker hub account

docker push dockerhubacount/imageName:tag

docker push akashkale/my-angular-app:v1

docker run -d -p 81:80 akashkale/my-angular-app:v1

then open the browser

<http://localhost:81>

Spring boot Database MySQL

We created jar file mysql image

Then we create image which run on

That image internally run on different OS.

Base OS Image.

Network to communicate two container

Base OS -🡪 Window

Angular Image

We created build file

Then created image

Then run application

Docker – compose docker compose is a tool kit which provided set of command which help to start, stop, re-start more than one container which work independently or interact with each other to share the data.

Docker compose use yaml or yml (Yeat Another Markup language).

Inside this file we need to write all instruction to create image, run container and those container are communicating with each others.

docker-compose.yml