Day 1 : 04-02-2022

Phase 4 : Testing and Deployment

1. Testing : Plain javascript testing, angular testing and node js testing.
2. Grunt : JavaScript task runner
3. Docker : Docker, Docker compose, Docker Swarm and Overview of Kubernetes.
4. CI and CD tool using Jenkin
5. Overview of AWS : S3 and EC2

Debug the JavaScript Application

We can develop the application using JavaScript in MEAN stack

1. Client Side JavaScript and Server Side JavaScript.

Client Side JavaScript : HTML, CSS, Bootstrap, JavaScript, jQuery, Angular / React JS /Vue JS

We write set of code to do the task.

100

4 line code

We keep break point using console.log(“1”);

10 line code console.log(“2”);

15 line code console.log(“3 ”+res);

Testing : Testing is use to find the defects or error or bugs in the application.

If we run the program which develop in any language indirectly we are doing testing for that application.

Operation.js

function add(a,b){

var sum = a+b;

return 0;

}

Testing are divided into two types

1. Black box testing

Input Process Output

Post Man

SOAP UI

Selenium tool

1. White box testing

Input Process Output

Manual testing

Automation testing

Unit testing : unit testing is a type of software testing where individual component or code or function of application tested individually.

Unit testing is a type of white box testing.

Using unit testing we can check function functionality working or not.

jUnit

nUnit

Jasmine : Jasmine is a type of open source framework which help to do the unit testing for Client side as well as server side JavaScript code.

Jasmine is DOM less Simple JavaScript testing framework.

Plain Client Side JavaScript testing -🡪 Jasmine Framework

1. With browser plugin
2. With node js

Old version Jasmine we were depends upon karma plugin get the result.

But new Version of Jasmine provided runner features to display the result on browser or console.

Angular Framework : Angular framework internally provide all configuration details for Jasmine.

Angular jasmine framework to do testing for angular component, service, module, pipe etc.

Angular use Karma a test runner. Which help to provide the testing result with respective browser.

React JS use JEST testing framework to do the unit testing.

Backend node js or express js testing

Jasmine Testing framework to testing node js application

Mocha is a light weighted testing framework we do testing for node js application

Mocha with Chai (test library framework).

Jasmine, JEST, Mocha

Suite : JavaScript testing framework provided pre defined function ie describe() which is use to add more than one test function ie test spec. Suite is like a container which hold more than spec with the help of it() functions.

describe(“message”,callback)

Spec : Spec provided it() function which help to test the JavaScript function functionality.

it(“message”,callback)

assert : testing framework provided lot of pre defined function which help to match actual and expected output. Which come in the form of assertXXX

expect(true).toBe(true)

expect(true).not.toBe(true)

expect(a).toEqual(bar)

expect(message).toMatch(/bar/)

expect(message).toMatch('bar')

expect(a.foo).toBeDefined()

expect(a.foo).toBeUndefined()

expect(a.foo).toBeNull()

expect(a.foo).toBeTruthy()

expect(a.foo).toBeFalsy()

expect(message).toContain('hello')

expect(pi).toBeGreaterThan(3)

expect(pi).toBeLessThan(4)

expect(pi).toBeCloseTo(3.1415, 0.1)

expect(func).toThrow()

describe(“Operation Testing ”,()=> { : suite

it(“addition testing ”,()=> { :spec

})

it(“subtraction testing ”,()=> {

})

})

Day 2 : 05-02-2022

Angular Testing :

Angular internally use jasmine testing framework to test angular component and services.

It provided karma as test runner to get the output on browser as well as console.

Angular with jasmine and karma

React JS with JEST testing framework

Angular use jasmine testing framework which provide describe(), it() and more than expect() function.

Angular use karma test runner to get testing result.

Angular use utilities class to test angular component and service.

Those utilities classes part of @angular/core/testing

TestBed is a class part of @angular/core/testing which provided set of function to do the testing for angular component and service.

ng test

Node JS Testing

Node JS testing we can do using jasmine or mocha

Create the folder back-end testing – using node js

Using npm init (create package.json file)

npm install jasmine

npm install jasmine-node

or

npm install jasmine -g

npm install jasmine -D

npm install jasmine-node -D

after installation now we need to create jasmine init

this command is use to create spec directories inside this director we need to keep

all our testing related files.

Now create src folder and inside this folder we need to create more than one node js file with

Node js code.

Operation.js

function add(a,b){

    let sum = a+b;

    return sum;

}

module.exports = {add};

OperationSpec.js

let obj = require("../src/operation");

describe("Operation testing",()=> {

    it("Addition testing ",()=> {

        let result = obj.add(100,200);

        expect(300).toEqual(result);

    })

})

Express JS Testing using Jasmine. Jasmine provided describe, it and expect to test the JavaScript.

But Jasmine doesn’t provide any function call all http method like get, post, put and delete.

Node js provided third party module ie supertest which help call http methods.

Create express-js-testing folder

Npm init (package.json file)

npm install jasmine –D

npm install jasmine-node –D

npm install supertest –D

npm install express

then using jasmine init (create spec folder)

then create src folder.

inside this folder create app.js file with rest api.

Please install docker for window or mac or linux

docker --version

docker images

Day 3 : 11-02-2023

Docker : Docker is an open source platform for developing, shipping and running the application.

Or

Docker is an advanced OS Virtualization software platform that makes it easier to create, deploy the run the application in Docker container.

Container : It encapsulate environment, which run on top of very shallow level of abstraction, providing a virtual machine to run the application.

To run the application software or any application develop in any language we need the system software ie OS.

Some application or tool or server or app are os dependent.

VM ware software came in picture.

Oracle Virtual box

Limitation of VM ware software

Base machine : 16 GB RAM

Hard disk 500gm

Virtual OS : 4 GM RAM

50 GM

If want to run n number of Guest OS.

10 Guest OS we want to run

2GB RAM

In Virtualization we are going the share the resource from base to guest os.

Using Docker we are creating Containerization application.

Virtualization Vs Containerization

Virtualization is an abstract version of physical machine or OS. While containerization is an abstract version of an application.

Docker container is responsible to provide OS which help to run the application in the form abstract.

Docker engine provide the features to run the application.

Container : it is a run the time environment.

Node JS : to run the Server side javaScript program we need node js software.

Web Container

EJB Container

JMS Container

Application Container

Docker container

docker --version

or

sudo docker --version This command is use to check the version of docker

docker images this command is to display all images present in local machine

docker pull hello-world this command is use to pull the image

if that image present in our local machine it pull from docker engine else it will pull from remote repository ie Docker hub

docker run imageName/imageId;

docker run hello-world : this command is use to run the image

Docker Container : Running instance of Docker images container turn the actual application.

Docker Images : The file system and configuration of our application which is use to create the container. Docker images contains all configuration details which help to run the program.

Or

It is a template that holds set of instruction which create the running container.

Dockerfile : A Dockerfile is a type of simple file which contains set of instruction that defined how our image build. It is a series of steps that you have to defined what will help to create the image.

If we want to do the addition of two number using function in JavaScript or node js we need to create

Add.js

Set of syntax to write inside a Add.js to do the addition of two numbers.

function add(a,b){

var sum =a+b;

return sum;

}

Write a program to create rest api using express js module

let express = require(“express”);

let app = express();

app.listern(3000,()=>console.log(“server running ”));

Docker provided one source remote repository ie Docker hub. Docker hub is like a git hub which help to pull or push (publish) custom images. Docker hub provide us pre defined as well as user defined images. Which we can pull it and run it. Using pre defined images we can create our own images or custom image we can publish it or push it so other teams they can pull it and run it in their local machine with help of Docker engine.

In git hub we can push any type of data. But Docker hub hold only images.

Please create your own Docker hub with your personal emailid

Busybox

Debiean

Alpine

If we want to run any application using docker images we need to create Dockerfile. Inside docker file we need to pull os images and write configuration details to run our application.

Creating custom image to display message.

Create a folder as DockerImages

Dockerfile

FROM busybox

CMD ["echo","Welcome to Docker Custom Image created by Akash"]

To create the custom images

Docker build –t imageName . –f Dockerfile

docker build –t my-busy-box . –f Dockerfile

creating images to run the node js program

First the node js program

app.js

let a=10;

let b=20;

let sum = a+b;

console.log("sum of two number is "+sum);

function sayHello(name){

    return "Welcome to Node js user as "+name;

}

console.log(sayHello("Ajay"));

Dockerfile

FROM node:14

COPY app.js .

CMD ["node","app.js"]

docker build -t my-node . -f Dockerfile

docker run my-node

Creating image to run the express js application.

Create the folder

npm install express js pakage.json file created with required module created inside node\_module folder.

now create app.js file and create more than one rest api.

Day 4 :

12-02-2022

If images is responsible to run web application

my-node

docker run –d –p 3000:3000 my-node/imageId

docker run –d –p 3001:3000 my-node/imageId

docker run –d –p 3002:3000 my-node/imageId

To check running container in our machine

docker ps this command is use to display running container

Or

docker container ps

docker stop containerId/containerName : it is use to stop the container

docker start containerId/containerName : it is use to start the container

docker rm containerId/contaierName: it is use to remove the container

if we get the error then please stop and then remove or else

docker rm containerId/containerName –f

command to remove the image

docker rmi imageName; if images is connected with container the we can’t remove.

Creating the image for html, css and javascript

Nginx : it is open source server which help to deploy web application.

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script>

        function displayname() {

            document.getElementById("result").innerHTML="Ravi"

        }

    </script>

    <style>

        h1{color:red;background-color: darkgray;}

    </style>

</head>

<body onload="displayname()">

    <h1>Welcome to Simple web page using Docker <span id="result"></span></h1>

</body>

</html>

Dockerfile

FROM nginx

COPY index.html /usr/share/nginx/html

docker build –t my-web-page . –f Dockerfile

nginx default port number 80.

docker run –d –p 80:80 my-web-page

docker run –d –p 81:80 my-web-page

creating image to run the angular application

first you create angular project

ng new angular-with-docker

then write the code base upon your requirement.

Then we need to build the project

ng build this command is use to create build file.

After build successfully it will create one folder with name as dist inside that project foldername

And that folder contains all build file which we need to give admin to deploy in actual server in production environment

Server can be tomcat, apache, IIS, nginx server.

Then create the Dockerfile

FROM nginx

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

Then create the images

**docker build -t my-angular-feb-2023 . -f Dockerfile**

now run the image nginx default port number 80

docker run -d -p 82:80 my-angular-feb-2023

then verify angular project with port number 82

http://localhost:82

if we want to push this image in docker hub registry

first we need to connect local machine terminal to docker hub account

docker login : it will ask docker hub account id and password.

Now we need to provide tag for our image before push in docker hub

docker tag imageName dockerHubAccountId/imageName:tag

tag can be number or version or any unique id

after provided tag for that image then we can publish this image in your docker hub account

docker push dockerhubaccount/imagename:version

docker push akashkale/my-angular-feb-2023:1.0

all people try

docker pull akashkale/my-angular-feb-2023:1.0

docker run –d –p 84:80 akashkale/my-angular-feb-2023:1.0

mongo db images

Each container is responsible to execute specific application develop in any language.

If we want to run more than one container may those container can interact with each other using tcp or http protocol when we need to use containerization tool like

Docker compose

Docker swarm

Kuberneties

Container can be n number of containers.

Docker swarm and Kuberneties are known as container management tool.

Management tool means it will management the life of the container like scale up, scale down, if any container do down provide backup for another container with same configuration etc.

These tools help us to deploy all container at time, build, up, down, start , stop etc.

Open

What is mean build tool.

Build tool is responsible to compile program, run program, creating executable file like jar, war file, exe file, creating documentation, testing the application, creating documentation.

CI and CD is known as

Inside CI and CD tools we can build the application with our without Docker.

Docker is use to run the application using container.

Kuberneties is use to run more than one container.

Docker compose : Docker compose is tool kit which help to run more than one container. Those container are running independently or they can be depends upon each others. To provide all container details we need to use yml file. This file contains all container configuration details.

Docker image for mongo db

docker pull mongo:5 pull mongo 5 version

docker run --name mymongo -d -p 27017:27017 mongo:5 running mongo images on port number 27017 with name as mymongo

docker exec -it mymongo bash : this command is use to open the mongo terminal

it open os terminal which contains mongo database.

then you write mongo command

show dbs;

then create database

create collection

view document from collection

if want to run Angular container : ngix port number 80

localhost:3000

frontend and backend connecting using browser with REST API. HTTP protocol.

Express js container : express container 3000

Express js container depends upon mongo db container connecting using TCP protocol.

Mongo db container : 27017 port number

Product CRUD Operation

If want to run more than one container those all container details we need to write the docker-compose.yml file and that file we need run through docker-compose command. Using this command we can build more than one container at time, we can up we down all container. Rather than running each container related commands.

CI and CD tools (Continuous Integration and Continuous Delivery or Deployment).

Dev1 Github (Organization Repository)

Shared repository (Shared Project Structure)-🡪CI/CD

Dev2

Dev3

CD/CD--🡪 Testing Server (Selenium tool) ----🡪Production Environment (live)

Devops : Git, Maven, Gradle, Docker, CI and CD, Kubernetes etc.

Dev1 responsible to create login page in angular

Dev2 responsible to create the application page in angular

Dev3 responsible to create the express js project

Dev1 or Dev2 or Dev3 once finish the ask allocated from him. They will push the code in shared repository.

That person need to build the project.

Build compile program, run the program, created executable file like exe, jar or war file, dist folder etc.

Using that build file we run the program.

We need to integrate more then one developer code develop in their and push to remote repository. We need to merge and re-build it again and again after every developer push the code in remote repository.



Jenkin is an open source CI and CD tool. Jenkin is base upon java and plugin based CI and CD tool which help build the application.

We can do Jenkin

1. Using software with respective OS.
2. Using Jenkin.war file with tomcat or any other java base server.
3. Using Docker

docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk11

it pull Jenkin image and run on port number 8080.

We will get the password in command prompt

<http://localhost:8080>

it will ask the password then paste password from command prompt and install suggested plugin.