Phase 5

Day 1 :

11-04-2022

Total 10 classes (one two hours)

Testing and deployment

TestNG (next generation on testing ) like jUnit

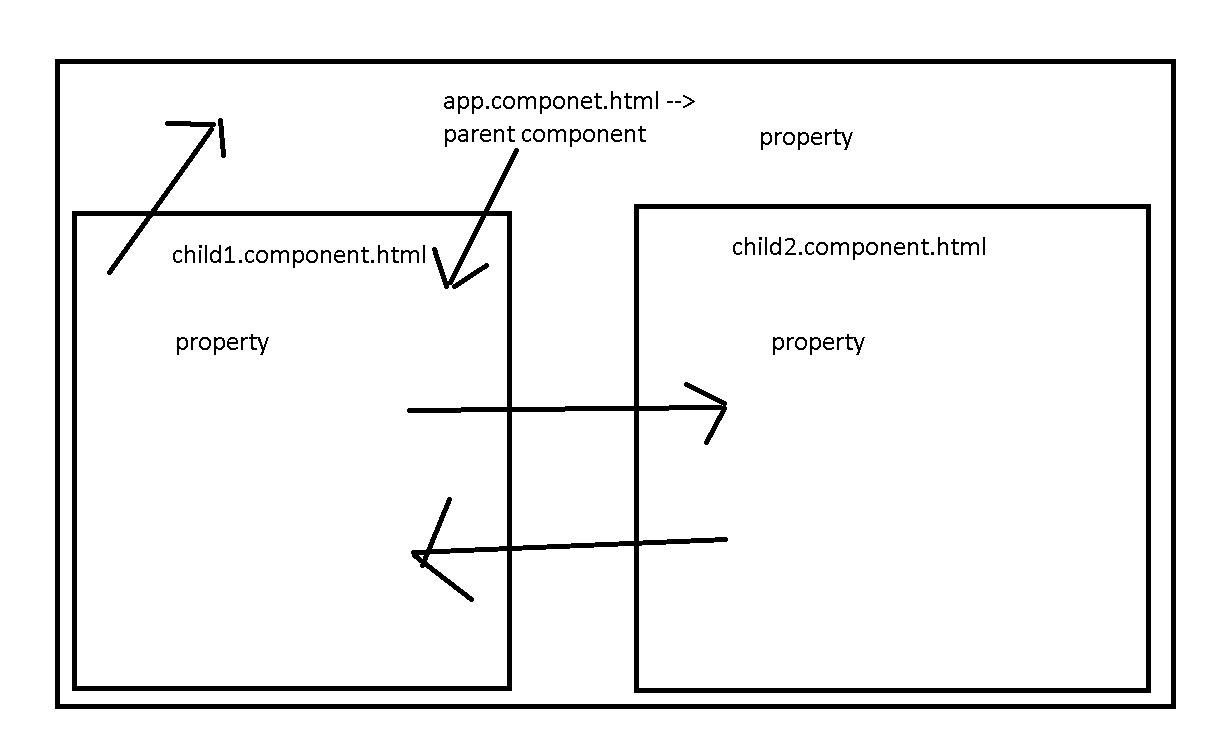
Selenium Automation tool to test the application

Docker

Kubernetes Overview

Cloud computing and AWS Overview EC2 and S3

Component communication



Ng new component-communication

Routing -🡪 no

Styling 🡪 css

Create two component using command as ng

ng g c child1

ng g c child2

ng g s shared

Share the data between component communication

1. Parent – child : using @Input decorator we can share the data from parent to child.
2. Child – Parent :
   1. Using @Output with EventEmitter API we can share the data between child to parent.
   2. Using @ViewChild we can share the data between child to parent.
3. Sibling :
   1. Using sessionStorage and localStorage : HTML5 and JavaScript features.
   2. Shared service class : using Angular features
   3. Using RxJS with Observable we can share the data.

Angular Routing

Ng new angular-routing

Routing -🡪 Yes

Styling 🡪 css

Angular routing is use to move from one component html page to another component html without loading whole page.

It is use to navigate from one component to another component base path provided in routing file.

Ng g c aboutus

Ng g c contactus

Ng g c feedback

Ng g c login

Ng g c home

Angular provided pre-defined ie <router-outlet></router-outlet>

This tag behave like a place holder. Which load the component html code base upon the path provided in routing file.

Day 2 :

12-04-2022

jUnit Testing 4.x and 5.x

TestNG

Unit testing : unit testing is use to test function functionality working property or not.

We write the set of code inside method or function.

Junit in Java

Nunit in asp.net

Angular framework is use Jasmine

React JS JEST

jUnit 5 version support Java 1.8 features.

jUnit 5 is a combination of multiple modules

jUnit 5 = Junit platform + Junit Jupiter + Junit vintage

Mock the object or create fake object or proxy object.

Controller layer depends upon the service layer

Service layer depends upon the dao layer

Dao layer depends upon the resource layer

Resource layer depends upon on the resources ie database.

Controller layer create the mock object of service layer

Service layer create the mock objet of dao layer

To do mocking concept

jMockito is a open source API or framework. We can use with jUnit to achieve mocking concept.

Day 3 :

13-04-2022

TestNG : Test NG is testing open source framework inspired by jUnit and nUnit. It provided some new functionality that makes more power full and easier to use.

Test NG : Next Generation

Next generation on testing.

Test NG mainly use to do integration testing.

Features

1. It generate HTML reports
2. It provided lot of different types of annotation. So we can run more than one test case in group and priority.
3. Parallel testing case achieve using TestNG
4. Generate the logs. (We can store the execution of a program).

@Test it is same like jUnit 4.x or 5.x

@BeforeMethod it is equal to @BeforeEach

@AfterMethod it is equal to @AfterEach

@BeforeClass it is equal to @BeforeAll

@AfterClass it is equal to @AfterAll

Test suite : it is use to run more than one test case class. every test case class contains more than one test function.

In jUnit testing we run test suite using class but in TestNG we run using xml file.

CarTest

carLoan

carColor

carPrice

BikeTest

bikeLoan

bikeColor

bikePrice

vehiclesuite.xml

file responsible to run two test cases class with 6 test method.

TestNG group

TestNG priority

TestNG Report

Selenium Automation tool

Selenium is one of most widely used open source Web UI (User interface) automation testing tool.

It support by all browser, all platform as well as different programming language.

Selenium supported by Java, C#, Python, Php, Perl etc. To do the testing we require Web driver base upon the browser. Using selenium you are doing the test for the UI develop any language like Java, Angular, Php, Asp.net.

We have to download the web driver based upon the browser like chrome, firefox, IE etc.

Selenium can be used to automate the functional testing and it can be integrate with other maven, gradle, docker , jenkin etc.

Selenium IDE : all browser provided plugin for Selenium IDE which help to records the all operation which we done in any web page.

It automatically create the test cases which help to re-run this different values.

Install node module using npm command as

Npm install –g selenium-side-runner

Then set the path for WebDriver in environment variable.

Then open the command prompt run the project

selenium-sider-runner MyApp.side

please install docker

after install successfully please verify

docker --version

docker images

it will display few column names.

May be you will get some error as WSL update please go through google or youtube and try to debug the error.

18-04-2022

Docker is a container. Which help to create containerization.

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

VM Software

To run the Virtual OS we have to provide RAM and External Hard disk space.

My base machine contains 16 GM ram I have to provide at least 4 GM RAM to run the VM OS.

If I want to run 10 VM at time.

Docker container is light weighted container it will package up an application and deploy it as on with the help of in build libraries and other dependencies.

Virtualization and Containerization

Virtualization is an abstract version of physical machine or device.

Containerization is the abstract version of an application. We are running the application in our machine without installing any necessary software to run the application But we are running in engine. Engine is responsible to provide the run time environment.

Docker Container : This is running process or instance of images. When the container start the application become up.

Docker Image : Docker image contains everything you ned to run our application or It is a file system and configuration of our application which are used to create the container or Docker images is source code for our container.

Docker File : A Docker file is a blue print of our application which help to create the image or set of instruction which help create the image.

Docker registry Docker registry is use to publish more than one images.

There are two type of registry

Public

Private

Docker organization provided Docker hub it is a like a git hub which hep to publish our registry.

Open the command or terminal of Virtual Lab

docker --version

or

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

docker images

or

sudo docker images

command to pull the image

docker pull imagename

docker pull hello-world : hello-world is a pre-defined image

or

sudo docker pull hello-world

command to run the image

docker run imageName/imageId

docker pull busybox

or

sudo docker pull busybox

if images is OS image then we have to run the image using command as

docker run –it busybox

docker pull debian

docker pull alpine

busybox, debian and alpine are os images

if we want to create own image. First we have to take the help of OS images ie busybox or alpine

on that image we can install necessary software which help to run the application.

Docker images folder

Inside that folder create

My Busy Box folder

If we want to create the custom image.

We have to create the file with name Dockerfile without extension.

Now create the docker file

Dockerfile

FROM busybox:latest

CMD ["date"]

The command to build the image

docker build -t my-busybox . -f Dockerfile

another images to get simple message

Dockerfile

FROM alpine:latest

CMD ["echo","Welcome to Simple Docker Example"]

Docker build –t my-alpine . –f Dockerfile

docker run my-alpine

We will create the image to run the simple Java Program

Create Java program

Demo.java

class Demo {

    public static void main(String args[]){

      System.out.println("Welcome to Simple Java program running to docker");

    }

}

**Then create the image file**

Dockerfile

FROM openjdk:8

COPY Demo.java .

RUN javac Demo.java

CMD ["java","Demo"]

docker build -t my-java . -f Dockerfile

docker run my-java

creating image to run the jsp or Servlet page

To run the jsp and servlet we require server ie tomcat.

After created web application we have to create the war file for servlet and jsp.

And that war file we have deploy on server like tomcat.

Index.jsp

<html>

<head>

</head>

<body>

    <%

        String name="Akash";

        out.println("Welcome to Simple JSP Page running through Docker.."+name);

    %>

</body>

</html>

Then using command create the war file

jar java archive .java and .class

war web archive .java(servlet), .html, .css, .js, .xml and .jsp

ear enterprise archive .java(servlet), .html, .css, .js, .xml and .jsp (with ejb program)

jar -cvf my-jsp.war .

Dockerfile

FROM tomcat:latest

ADD my-jsp.war /usr/local/tomcat/webapps/

CMD ["catalina.sh", "run"]

docker build -t my-jsp . -f Dockerfile

to run the image if image is responsible to run the web application

we have to write the command as

tomcat default port number is 8080

docker run –d –p 8080:8080 my-jsp

-d : detached mode (background)

-p : expose the port number

Left side port number can be same or different

Right side port number actual server port number

To verify the container is running or not we have to use the command as

docker ps (process status) : it will display only container.

docker ps -a

we can start and stop container using container id

docker stop containerid

docker start cotnainerid

we can delete the container using container id

docker rm containerid if we get the error can’t delete

before that stop the container and delete the container

docker rm containerId -f

to delete the image we have to use the command as

docker rmi imageId

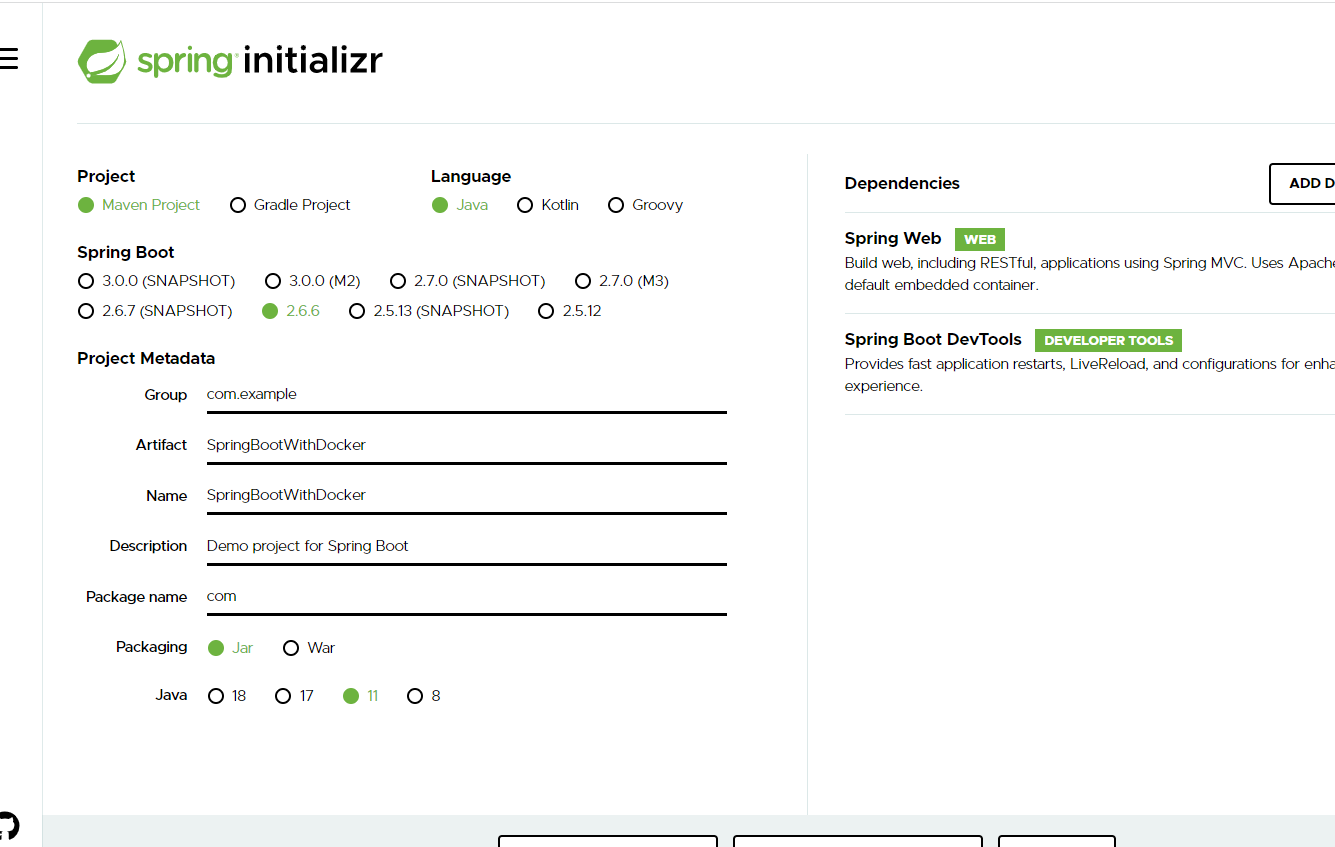
To run all images and container

docker system prune –a

Creating image to run the Spring boot application

Create the Spring boot application spring initializer

With simple Rest Message.



In this project you can create controller, service, dao etc.

After developing the application we have to create the jar file for this project.

using mvn package : it will create the jar inside a target folder.

Then create the Dockerfile

Dockerfile

FROM openjdk:11

COPY ./target/SpringBootWithDocker-0.0.1-SNAPSHOT.jar .

CMD ["java","-jar","SpringBootWithDocker-0.0.1-SNAPSHOT.jar"]

docker build -t my-spring-boot . -f Dockerfile

docker run -d -p 8181:8080 my-spring-boot

10 min to run

Then take break 20

Then angular project

ng new angular-docker-app

in html page

Welcome to Angular with Docker created by Akash Kale

After develop the application depending upon the requirement we have to build the project.

ng build

after build the project it will create the dist folder.

Dist folder contains all angular build project details.

Generally we deploy front end project in one of the open source nginx server.

Then create the to run the angular project on nginx server ie it is a open source server which help to deploy the front end technologies.

Dockerfile

FROM nginx

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

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

nginx default port number is 80

docker run -d -p 80:80 my-ng-angular

right side is actual port number of nginx server

lift side is expose port number it can be same or different.

Then hit on browser

<http://localhost:80>

every web application default port number 80 consider.

Before publish this image we have create tag for that image. Tag is just like a identity or version.

Docker tag imageName dockerhubaccountId/imageName:tag

Tag can be version or latest

docker tag my-ng-angular akashkale/my-ng-angular:1.0

docker push dockerhubaccount/imageName:version

docker push akashkale/my-ng-angular:1.0

20-04-2022

docker pull akashkale/my-ng-angular

docker run –d –p 81:80 akashkale/my-ng-angular:1.0

Container :

Every container is responsible to execute specific application.

To run the container we have to create the image with the of Dockerfile. Dockerfile hold the configuration details about what application which we want to execute.

Some time we want to make communication between one container to another container.

Docker compose

docker-compose.yml :

inside this file we are giving the details about more than one container how to run.

Example Spring boot running on one container

My SQL database running on another container

Angular framework running on another container.

If we want to run all these three container we can configure in docker-compose.yml file.

Please create the file with name as

docker-compose.yml

version: "3.9"

services:

  web:

    image: akashkale/my-ng-angular:1.0

    ports:

      - "85:80"

Run this application using command as

docker-compose up –d

docker-compose.yml

version: "3.9"

services:

  web:

    image: akashkale/my-ng-angular:1.0

    ports:

      - "85:80"

  springweb:

    image: my-spring-boot

    ports:

      - "8181:8080"

Docker compose is good if we want to run only few container.

But in big application if we want to maintain more than like 100 container which are interacting with each other to develop big application

Container orchestration frameworks.

Container orchestration solves the problem by automating the scheduling, deployment, scalability, load balancing, availability, and networking of containers.

Kubernetes

Docker swarm

OpenSwift

Difference between

Docker swarm Vs Docker Compose Vs Kubernetes

Kubernetes :

It is a open source orchestration tool use to manager containerized application.

Docker is use to create containerized application.

It is use to create the container.

Kubernetes is use to maintain more than one container.

Kubernetes originally design by google.

Orchestration tool is responsible to maintain the life of the container.

Kubernetes develop using Go language.

Kubernetes basic component

1. Pods
2. Selectors and Label
3. Controller
4. Services

Pods : pods is collection of more than one container and every container is responsible to execute specific application using image. And those container running in same machine or node or host or different node or machine or host.

Pods is responsible to run more than container. They maintain the life of the container. They connect more than one container if they are running on different machine using port number and ip address. Those machine must be in one network ie cluster environment.

Kubernetes provided some command ie kubelet using these command we can connect more than one container running on same machine or different machine.

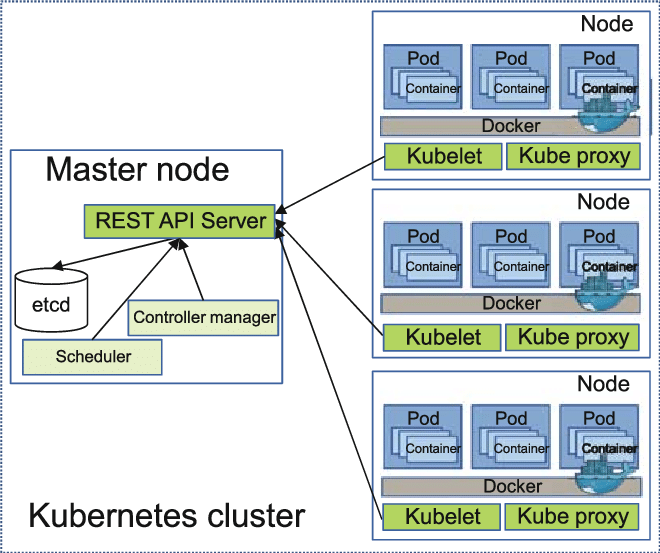
Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

Selector and label : Kubernetes attached key-value pairs for various object like services, pods and node or host or machines.

Using these labels we can search the resource very easily.

Controller : controller is responsible to maintain the life the pods. If any pods go down it will automatically created backup pods to those container.

Services : collection of pods are combined together in one service.



To run simple kubernetes application in base machine

kubectl : Kubectl is command line tool. Which help to run the commands against Kubernetes cluster.

It allow us to perform every possible Kubernetes operation through command prompt.

Kubectl is a client for the Kubernetes API.

<https://medium.com/@ggauravsigra/install-kubectl-on-windows-af77da2e6fff>

* [https://dl.k8s.io/release/**v1.23.3**/bin/windows/amd64/kubectl.exe](https://dl.k8s.io/release/v1.23.3/bin/windows/amd64/kubectl.exe)

minikube : minikube is a tool, using this tool we can run Kubernetes locally in personal laptop ie Window, Linux or Mac etc. Minikube provide it will provide single node cluster environments.

<https://minikube.sigs.k8s.io/docs/start/>

After installation kubectl and minikube successfully please verify the both the commands.

kubectl version

minikube version

First start he minikube

mnikube start

kubectl get pods

to create the pods we have to create container

container will create using image

image is use to run the application.

First create the project it can be angular or spring boot

If angular project : do the changes in app.component.html file and

Build the project

After build angular project successfully it will create dist folder.

Now we have to create the Docker file and run this project in nginx.

Dockerfile

FROM nginx

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

docker build –t angular-k8s . –f Dockerfile

Now we have to push this image to Docker hub.

Before pushing this image we have to create the tag for that image.

docker tag angular-k8s akashkale/angular-k8s:1.1

now you can push this image to Docker hub

docker push akashkale/angular-k8s:1.1

now we can deploy this image in minikube to maintain by Kubernetes

to deploy the image in Kubernetes environment

kubectl create deployment my-first-k8s --image=akashkale/angular-k8s:1.1

we have the verify container in minikube dashboards.

Please run this command.

minikube dashboard

If spring boot project : write controller and using mvn command create the jar file.