**Java FSD Phase 5 : Testing and Deployment**

Day 1 26-10-2022

Testing using TestNG

Selenium

Docker

CI and CD tool

AWS Overview S3, EBS and EC2

Kubernetes Overview

Junit testing 5 in Phase 3

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

Read a,b

Compute sum = a+b

Write b

MVC Model View Controller

Service layer

Dao layer

Resource layer

We can test any layer code working or not without main method or function is known as testing. We are checking actual and expected output working or not.

Testing mainly divided into two types

Black box testing

Input --🡪 Process ---🡪 Output

White box testing

Input --🡪 Process --------🡪 Output

Unit Testing : Unit testing is a software development process in which the smallest testable part of an application or program are tested which written in function or method or class or modules. Unit testing is a type of white box testing.

class Operation {

public int add(int a, int b){

int sum = a+b;

return sum;

}

}

In Java we can do unit testing using two testing framework

1. jUnit
2. TestNG

TestNG is a testing framework inspired by Junit and Nunit testing framework. TestNG means next generation for testing. TestNG is light weighted framework which provided more functionality than jUnit.

jUnit is use to do the unit testing.

TestNG is use to do the unit testing as well as integration testing.

TestNG is an open source testing framework. It is like a unit testing but more powerful than UnitTesting.

It support parallel testing

It generate testing report by default in html format.

TestNG and jUnit is third party framework. To do testing using these framework we need to add jar files. We will add those jar file using maven tool.

In Testing using Junit and TestNG

Test case : it is a normal java class which contains more than one test function with @Test annotation which is responsible to test function functionality.

Test suite : test suite is a collection of more than one test case class. In JUnitTest we create Test suite class but in TestNG we create test suite using xml file.

Testing hook methods

TestNG provided lot of pre-defined annotation which help to do the task before testing as well as after testing method, test method, suite etc.

Test priority

By default TestNG follow methods names as ascending order to execute the test methods.

Selenium :

Selenium is an open source automation Web UI (User interface) tool or suite.

Selenium is support by all language like Java, C#, Python, JS etc. It is browser independent and Platform independent.

Using Selenium we can test any UI component develop in any language like Java, Spring boot, Asp.net, Php, Angular or React JS.

Selenium is use to do the integration testing.

We can configure selenium with any Devop tool like Jenkin, Maven, Gradle etc.

Selenium provided web driver with respective browser ie Chrome, IE, Edge or firebox which help to load the web page develop in any language.

Selenium provided API (classes and interfaces) which help to read the DOM or tags contents using different types of selector. Then we have to take the help of jUnit or TestNG framework to check excepted and actual output.

Selenium with TestNG to do the UI testing using Java technologies.

Day 2 27-10-2022

Selenium with TestNG

1. Login page testing
2. Google page testing

Docker

Container :

Virtualization create a virtual version of resources such as database, tool, software or application.

Virtualization let us divide a system into series of separate section each one act as distinct individual system.

VMWare software

Base Machine Window 11

RAM 16GB

Hard disk 1tb

Guest OS -🡪 Linux

RAM 4 gm

Hard disk 10GB

We want to run n number of G OS

Docker is an Advanced OS Virtualization software platform that makes easy to create, deploy and run the application in Docker container.

Using Virtual software we can create the virtualization

Using Docker container we can create containerization application.

Virtualization is an abstract version of physical machine or abstract version of OS.

Containerization is an abstract version of an application or tool or software.

Docker file :Docker file contains set of instruction which help to create the Docker images. This file contains all configuration details which help to run the application. Using this file we have to create the images.

Java Program is like as Docker file

Docker image : using Docker file we will create the images. It is a template that hold set of instruction which help to run the container.

Like a jar or war file

Container : it is known as run time environment or engine responsible to run the application.

Docker container : it is a running process or engine of an image or Running the instance of Docker image container turn the actual application or run the application.

Sample.c

Demo.java

Sample.py

To do docker concept we need Docker

1. We can install docker in our own machine
2. We have to use LMS portal labs etc.

In Virtual Lab open the terminal

docker --version this command is use to check the docker version

or

sudo docker --version

docker images this command is use to the images present in local

or virtual machine.

or

sudo docker images

docker pull imageName this command is use to pull the image

or

sudo docker pull imageName

docker pull hello-world

docker run imageName/imageId This command is use to run the image

hello-world is one of the pre-defined available in Docker hub account.

Docker hub is a remote repository which help to push and pull the images. It is like a github.

docker pull busybox

docker run –it busybox : it iterative mode. This image contains os. So using it we can open busybox os terminal.

If we are planning to create our own image then we have to take the help of OS images like

busybox

alpine

debian

OS Image

To create our own method we have to create the **Dockerfile**

To create the image using Docker file

Open the command prompt in the location where Dockerfile present

docker build –t imageName . –f Dockerfile

Now we will create the image to run the Java program

First create the Java Program

public class Demo {

    public static void main(String args[]){

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

    }

}

Then

Create the Dockerfile

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

CMD ["java","Demo"]

Build the image using Dockerfile

docker build –t my-java . –f Dockerfile

docker run my-java