Day 1 :

26-09-2022

Phase 5 :

Testing and deployment

TestNG : unit testing framework

Selenium : Automation testing tool

Overview of AWS : S3 and EC2 instance

Docker

Intro CI and CD tools : Jenkin

Introduction to Kubernetes

Test NG Test NG is an open source testing framework base upon jUnit. Test NG mean next generation. It is a light weighted framework which help to do unit testing as well as automation testing with Selenium tool.

Test NG provided lot of pre defined annotation

Like

@Test

@BeforeMethod

@AfterMethod

@BeforeTest

@AfterTest

@BeforeSuite

@AfterSuite

In TestNG , TestNG class is like a junit test case

In Test NG we have to create the XML file to make the TestNG suite class.

Inside that xml file we provide class Test NG class details.

If we want to create the TestNG project we need take the help maven project and add the dependencies.

Day 2 :

27-09-2022

Testing suite using XML file

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<suite parallel=*"false"* name=*"Suite"*>

<test name=*"MyTest"*>

<classes>

<class name=*"com.service.HookTestingMethods"*>

<methods>

<exclude name=*"f1"*></exclude>

</methods>

</class>

<class name=*"com.service.SampleTest"*>

<methods>

<include name=*"f1"*></include>

</methods>

</class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Testing more than one test function using groups

Using group attribute we can add more than one test function part of same class or different class.

TestNG generate the report by default in the form of html.

Selenium with TestNG

Selenium is an open source web ui automation testing framework .

Automation testing framework

its suite of software consists of Selenium WebDriver, Selenium Grid and Selenium IDE.

We can do testing using Selenium using different language like Java, Python, JS, C# etc.

Selenium WebDriver support by all browser like Chrome, IE, Edge, fire fox etc.

jUnit and TestNG provided set of annotation which help to do the testing with set of assertion methods.

With help of jUnit and TestNG we can do UI Testing with Selenium.

JavaScript testing : Jasmine and Mocha tool is like a junit testing for JavaScript.

Jasmine(JS) 🡪 jUnit(Java)

Mocha(JS) 🡪 TestNG (Java)

Angular testing : jasmine with Karma (Test Runner )

With help of Selenium we can do UI testing doesn’t matter that application develop in any language.

Java developer, Selenium provided lot of classes with help of Jar which help do the testing for UI Application using Java code.

Using Selenium we can load the static as well as dynamic web page and we can read the DOM element (HTML contents) from that web page using different types of selector like id, class, tagName, className etc.

Then we have to depend upon jUnit and TestNG testing to check the tags connect ie actual and expectation.

Selenium Web Driver using Java

Selenium provided pre defined class ie WebElement which contains pre defined methods which help to read the dom elements.

By class which contains static method to access the dom elements.

Day 3 :

28-09-2022

**Selenium With TestNG**

We can do the Testing for Using selenium with help of Selenium IDE.

Using this IDE we can record the flow the application.

Day 4 :

29-09-2022

DevOps : Development and Operation Team

Git

Maven

Gradle

Docker

Kubernetes

Agile

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

Docker is known as Advanced Version of VM(Virtual Machine).

Virtual machine provide an environment which help create Virtual OS and Virtual OS we can run the n number of application base upon the requirements.

VM ware software

Base Machine is Window 10 with 16 RAM

If we are planning one VM ie Unix or Linux

4 GM

50 HD GM

But if I want to run 10 VM at time.

Virtualization : running one or more OS and inside OS we can run multiple application.

Virtual OS base upon base OS.

Using Docker we run the application with the of Docker container.

Using Docker we can create Containerization application.

Virtualization is an abstract version of a physical machine. While containerization is the abstraction version of an application.

Dockerfile : A Docker file is a blue print or set of instruction that defines how our image build.

.java

Docker Image : Docker images are the source code for our container or The file system and configuration details for our application which are used to create and run the container. We can create Docker image using Dockerfile

Jar or file

Docker Container : container is a run the instance or instance of images or run time environment. Running an instance of Docker image container turn the actual application or run.

Running program in JVM

This application or container running in Docker engine environment.

docker --version

docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

Or

Then login to Virtual Lab

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

docker images : This command is use to display all images available in current machine

docker pull imageName : This command is use to pull the image from Docker hub account

docker pull hello-world

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

docker hub : Docker hub is like a git hub which help push and pull the user defined or pre-defined images. It like a like remote repository which help to store more than one images.

docker pull busybox

docker run –it busybox

docker pull alpine

docker run –it alpine

Creating user defined image with the help of busbox

Dockerfile

FROM busybox

CMD [ "echo","Welcome my busybox image created akash" ]

docker build –t my-busybox . –f Dockerfile

docker run my-busybox

Creating user defined image to display the date using alpine

Creating image to run the Java program

First create the folder

Write Java program

public class Demo {

    public static void main(String args[]){

        System.out.println("Welcome to Simple Java Program using Docker");

    }

}

The create the Dockerfile

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

CMD ["java","Demo"]

Create the image image

docker build –t my-java-app . –f Dockerfile

docker run my-java-app

03-10-2022

**Maven goal**

mvn clean : it help to clean the project

mvn compile : it compile the projet

mvn package : it will create jar or war file which help to deploy the project.

04-10-2022

Create the spring boot project and create new rest api

Then using mvn command create the jar or war.

Then create Dockerfile

**FROM** openjdk:11

**COPY** target/spring-boot-with-docker-0.0.1-SNAPSHOT.jar .

**CMD** ["java","-jar","spring-boot-with-docker-0.0.1-SNAPSHOT.jar"]

docker build –t my-spring-app . –f Dockerfile

if image is responsible to run the server

docker run –d –p 9090:9090 my-spring-app/imageId

docker ps : This command is use to check the running container

or

docker container ls : this command is use to check running container

before pushing the image to docker hub account we have to create tag for that image. Tag is just like a unique identity like version number

docker tag imageName dockerHubAccountId/imageName:version

after tag created then you can push your image to docker hub account

docker push akashkale/my-spring-boot123:1.0

docker pull akashkale/my-spring-boot123:1.0

docker images

docker run –d –p 9393:9090 akashkale/my-spring-boot123:1.0

docker ps

or

docker container ls

<http://localhost:9393>

EC2 : instance

We will EC2 instance using terminal

That terminal is known as Virtual Server Machine terminal.

We have install the Java software

Then we have to install git

So we have to create the project in our machine and push to git

And pull the project in EC2 instance using git clone

Or

We create the Jar and upload that jar file in S3

So if we run the Java project or angular project that project we can access using live IP Address.

To open the IP address we have to cu

app.component.html

<h2>Welcome to Angular project created by Akash Kale</h2>

ng serve –o (development mode)

<http://localhost:4200>

now we have to build the project

ng build This command is use to build the project

after build inside an angular project folder. dist folder and project folder contains all build files.

Nginx : it is an open source server which help to develop the front technologies.

Now create the Dockerfile

FROM nginx

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

Then build the project

docker build -t my-angular123 . -f Dockerfile

nginx default port number is 80.

docker run –d –p 80:80 my-angular123

<http://localhost:80>

docker run –d –p 81:80 my-angular123

<http://localhost:81>

first create the tag with number or alphabets

**docker tag my-angular123 akashkale/my-angular123:1.0**

**docker push akashkale/my-angular123:1.0**

AWS : S3 : Amazon Simple Storage Service : it is like google drive which help to share the data of any type without a aws account as well as outside base upon the security apply for that bucket.

Bucket is like a container which provide unique name which help to add more one folder or files.

05-10-2022

We create the spring boot project and that project we will push to git repository.

Please open the terminal inside spring boot project

git init

git add .

git commit –m “spring boot with aws”

git push –u origin master

CI and CD: Continuous Integration and Continuous Delivery or deployment

Dev1

Login page

Dev2 GitHub

application page Remote

Dev3

Feedback page

Build the project : compile, run program, test project and create jar/war/ear file using maven goal or gradle task

Integration phase :we are combining more than one developer code and build the project.

CI and CD

Jenkin : It is a type of open source CI and CD tool also known as automation tool or server which is responsible to interact with remote repository and pull and project and build the project. Jenkin develop using Java technologies. It is plugin base CI and CD tool.



1. Jenkin exe
2. We can download war file and using tomat we can run the Jenkin
3. Using docker we can run the jenkin

docker images : it display all images

docker ps : it display all containers

or

docker contains ls

docker start containerId/containerName

docker stop containerId/containerName

docker rm containerId : it remove stopped container

docker rm dockerId –f it remove running container forcefully.

docker rmi imageId/imageName

docker rmi imageId/imageName –f

docker ps –a : it display all container it may be running or stopped

docker system prune –a it delete all stopped container, network, images

run the Jenkin image

Below command it will pull and run the Jenkin application on port number 8080

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

or

virtual lab or EC2 instance

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

After pull successfully it will start the Jenkin then open the browser and write

<http://localhost:8080>

it will ask the password. Please check the password in console and terminal.

After password paste in textfield it will ask to install the suggested plugin. Please install it.

Jenkin Pipe line : Jenkin pipe line allow us to define a complete list of jobs or event or task which interconnected to each others.

Verify version

Build the project

Test the application

Deploy the application

Docker compose

Docker Swarm

Kubernetes

Docker swarm and Kubernetes are known as container management tools. Which responsible to manage more than one container. Using these tools we can scale up, down, availability, scheduling, failure of one of the container.

Every container is responsible to execute any application like Java, spring boot, angular etc.

But if two container want to communicate to each other.

7-10-2022

Docker is use to create the container. Container is responsible to execute the any particular application ie java application, spring boot, angular application.

While creating big or enterprise application we require multiple container

Like Spring boot container want to communicate with MySQL container

Like Spring boot with Angular application

Docker compose

Docker swarm

Kubernetes

These three modules help us to run multi container application which can communicate with each others.

Docker compose is the toolkit provided by Docker which help to build, ship and run multi container application.

Docker compose use yaml or yml file (yet another markup language). To read all container configuration details. Ie docker-compose.yml or docker-compose.yaml

Please create spring boot project and create one rest api

Spring boot running on port number 9090

Please create angular project

ng new product-frontend

cd product-frontend

ng g c product : create the component

ng g class product : create the model class

ng g s product : create the service class

CORS : Cross Origin Resource Sharing :

Front end technologies : Angular running on port number 4200

Backend technologies : Spring boot running on port number 9090

Two server or domain are going to communicate to each others.

I Docker compose file we can write more than one image details. With the help of docker compose we can run more than one container using docker-compose up, down, build etc.

Contain management tool

Docker swarm

Docker swarm is a group of either physical machine or virtual machine (node) that running Docker application and have been configured with help of cluster.

Once our group of machine or nodes have been clustered together. To control both machine Docker engine is taken care by Docker swarm

Docker swarm is a container orchestration tool. This tools is responsible to manager more than one container running in different node. Like scale up, down, container failure

Docker swarm provided different types of nodes.

1. Master Node : This node is responsible to assign the task the worker node.
2. Worker node : This node is responsible to to the task.
3. Leader node : The leader node is responsible to maintain the cluster environment.

10/10/2022

sudo docker swarm init

sudo docker swarm leave

sudo docker stack deploy –c docker-compose.yml mySer

sudo docker service ls

We have create the EC2 instance

Then in EC2 instance we have to install Jenkin

Then using EC2 instance IP Address with Jenkin port number ie 8080 we have to open the Jenkin Dashboard.

Using Global configuration we have to add Java, Git, Maven etc.

We have to create Jenkin Job to run Docker swarm

We have to install Docker plugin in Jenkin Environment.

Then we have to run Docker swarm command in Jenkin pipe line.

Open EC2 instance

First install java 11

sudo amazon-linux-extras install java-openjdk11

Now, to download the latest Jenkins package

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo

\*\*iv. \*\*To enable the installation of the package, import the key file from Jenkins-CI:

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key

Install Jenkins on the EC2 instance

sudo yum install jenkins

To start the Jenkins service

sudo service jenkins start

To check the status

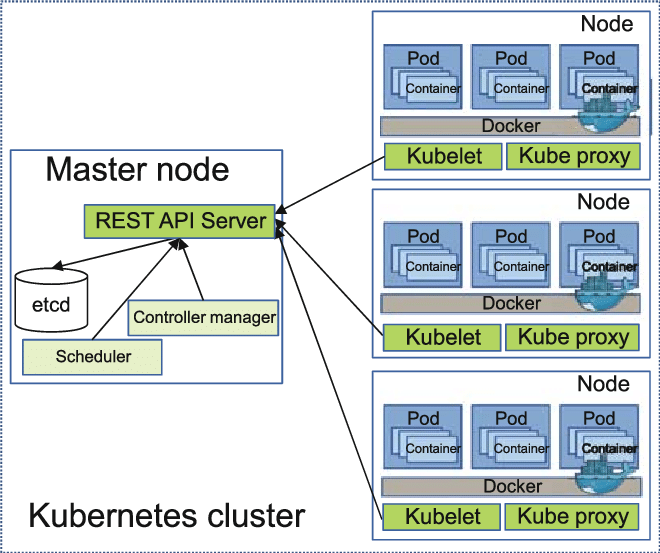
sudo systemctl status jenkins

This one is get the password

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Kubernetes : Kubernetes is open source container management tools. It also known as K8s. It is part of google and acquired by Cloud Native Computing. Kubernetes develop using go lang.

Kubernetes VS Docker Swarm



Pods :In One node we can create more than one pods and each pods contains more than one container and each container responsible to execute the specific application.

Using pods we can deploy we can deploy more than one application with help of container.

Life cycle of pods is taken care by container.

If any pods go down or fail, a replacement of that pods get created by controller.

Service : A collection of pods are bundled together in a service.

Service layer is responsible to expose our pods to one node to another nod using DNS and Port number.

Kubectl : Kubernetes command line tool, this tools is use to interact with Kubernetes cluster environment.

Minikube : Minikube provide a great features to run Kubernetes in single node cluster environment

So first install kubectl

Then install minikube and set the path

kubectl version

minikube version

Then after start the minikube

minikube start

minikube dashboards

<https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/>

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