

Day 1 :

26/08/2023

SDLC : Software Development Life cycle

Water fall model

Requirement gathering 6 months

Plan

Design

Coding / testing

Build the project development mode

Deploy the project production mode

Provide the service

Increment model

V model

Agile model

Sprint : time duration to develop small module 1 or 2 week. Login page, feedback page

Dashboard

DevOps : Development and Operation

Development people develop the application using any language like java, python, etc

Operation team interact with customer or end user client and they are responsible to maintain the application.

Admin

Developer

Tester

Admin

Database designer

Architecture etc

Devops tools

Git and git hub

Maven and Gradle (Java development)

CI and CD tool using Jenkin (Continuous Integration and Continuous deployment or delivery)

Selenium tool (testing)

Ansible tool (monitoring tool as well as configuration tool)

Docker container

Nagios tool

Kubernetes

Git and github

Sub version control which help to **record** the application flow.

Dev1	login page	
Dev2	java or python code	merge the code
Dev3	database	

Unix commands

ls : this command display all files and folder present in current directory

pwd : print working directory : it show current path of terminal or command prompt.

mkdir foldername: this command is use to create the folder

cd foldername : move inside a folder

cd .. : come outside a folder or move to parent directory of current folder

rmdir foldername : to remove folder

touch filename : to create empty file

vi filename : open the file in vi mode

once open hit i key to move inside a vi editor mode

write the contents

esc : to come out from editor mode to normal mode

:wq : write and q quite (save and exit)

cat filename : it is use to read contents from a file

cat means concatenate

Git : Git is version control system which help to track or record changes done in the application or project or app.

Git also known a distributed sub version control.

First create the folder

Then create the file and write the contents

git --version

git init it is use to make local folder as git repository

init command create .git folder insider that current directory

ls -a : it display all files and folder with hidden folder.

git status : this command is use to check the current status of your repository

git add filename : to add files or folder normal local folder to tagging area.

Or

git add . : this command is use to add all files and folder present in

Current directory.

git commit -m "message" : this command use to pass the task from tagging area to

local repository.

```
git config --global user.email "akash300383@gmail.com"
```

```
git config --global user.name "akash"
```

steps

1. Create folder with any name ie Demo
2. Then create the file with any name ie test and write the contents inside that file.
3. Then open the terminal inside that folder please use `pwd`
4. `git init`
5. `git status`
6. `git add .`
7. `git status`
8. `git commit -m "message"`
9. first time we need to set config details as email and name
10. `git config --global user.email "akash300383@gmail.com"`
11. `git config --global user.name "akash"`
12. Then please commit using command as `git commit -m "done"`
13. `git status`

`git branch` : branch is like a pointer which holds more than one commit details.

By default git provides default branch. Default branch name may be `master` or `main`.

If we want to check branch details present in local repository

`git branch`

command to create user defined branch

`git branch branchname` this command is used to create user defined branch

`git checkout branchname` this command is used to switch from one branch
to another branch.

Current branch is master or any other branch

`git merge branchname` this command adds all tasks in current branch

`git branch -D branchname` this command is use to remove the branch

Demo.java

int a; akash branch

int b; Vikash branch

Remote repository help us to share the code between two or more than one tabme.

Git hub

Git lab

Bitbucket

Aws

Azure

Private cloud etc

git hub : it is a type of remote repository provided by micro soft organization.

Day 2 :

27/08/2023

We want to connect local repository with remote repository

1. Token base authentication
 2. SSH Client
- More

To connect local repository with remote repository

`git remote add origin URL`

`git remote add origin https://github.com/Kaleakash/test_rep.git`

`git remote add origin https://token@github.com/Kaleakash/test_rep.git`

`git push -u origin main` (it is use to push the code)

how to resolve the conflict

1. first create Repo2 folder
2. then open terminal inside that folder
3. create sample file
4. add some data 1st, 2nd
5. using git init make folder as repository
6. git add .
7. git commit -m "done changes in master branch"
8. create the branch
9. git branch akash
10. git checkout akash
11. in akash branch we will add 3rd and 4th message.
12. Then git add .
13. Then git commit -m "in akash branch done some changes in sample file"
14. Create another branch with name as Vikash

15. `git checkout -b Vikash` (it will create the branch and switch to that branch)
16. in sample file in Vikash we will add the message as 5th and 6th.
17. Then `git add .`
18. `Git commit -m "done change in sample file by Vikash branch"`
19. Please move the master branch ie `git checkout master`.
20. Please verify current branch using command as `git branch`
21. Then in master branch merge the code from akash branch
22. `Git merge akash`
23. Using `cat sample` read the data from sample file
24. Out must be 1st, 2nd, 3rd, 4th
- 25.

Download or clone the repository

1. Create the folder with any name `ie devopstrainig` In VM
2. Then open the terminal
3. `git clone URL`
4. `git clone`
https://github.com/Kaleakash/devops_aug_2023_trainig_batch.git
5. use `ls` command to see downloaded folder
6. using `cd` command please move inside that folder.
7. `cd devops_aug_2023_trainig_batch`
8. `ls` command to see the more than one file.

if we do any changes in local repository
we need to add, commit and push

```
git add .  
git commit -m "done some changes in file"  
git push -u origin main
```

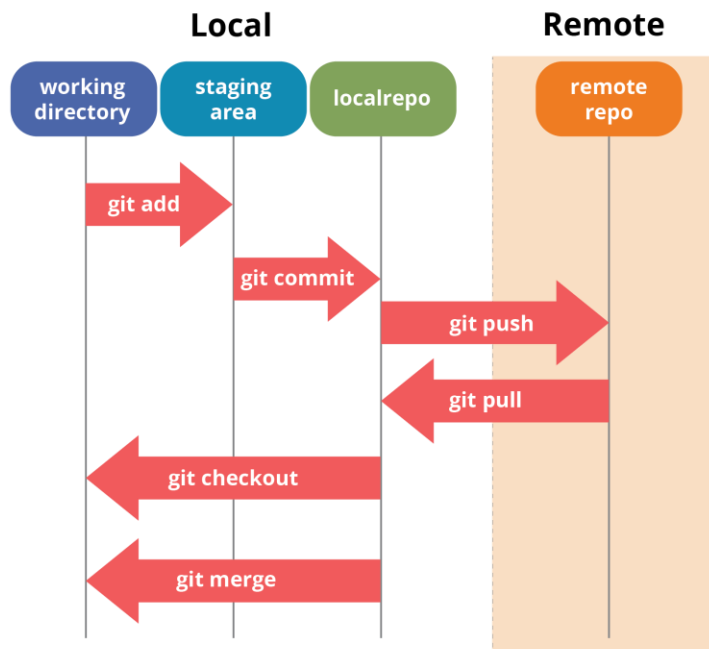
1. open the terminal inside a repository folder
2. make sure `.git` folder present using command verify `ls -a`
3. `git pull`

`git clone URL` : it help to download fresh repository in local machine

`git pull` : it will pull new updated from remote to local repository

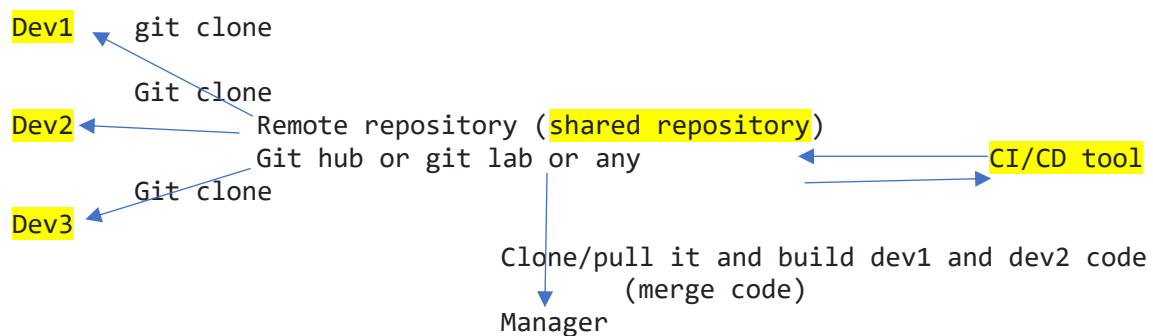
Git Vs Git hub

Git is Remote repository provided by micro soft.
Command line or GUI tool which
Help to interact with any remote repository



CI and CD tools

Continuous Integration and Continuous delivery or deployment



java	Git hub	Jenkin
python	Bit bucket	GoCD
angular		

Generally shared repository can be private or public.
While creating or after created that repository we will send the invitation
To all developer to join that repository.

Default branch can be master or main.

Dev1 need to create login page using html and css

Dev2 need to create java or python code

Dev3 need to create database tables.

If they do all their task in main or master branch after changes done they can add, commit and push the code to shared repository.

Don't do your task in main or master branch. While doing your task please create user defined branch and push that branch in remote repository.

In remote repository we will check user defined branch if code is correct we will merge that code into master or main branch.

Build phase :

You need to compile and run the application using that language.

Javac

Java java

Py python

Ng angular

Dev1 push the code in remote repository

In remote repository we need to verify the code and merge the code. And we need build the application.

After dev2 or dev3 code merge in master or main branch we can build successfully or it generate some error.

CI and CD tools.

Jenkin : it is a type of CI and CD tools. It is an open source ci/cd tool base upon java technologies. Plugin base ci and cd tool. GUI based tool.

Open the browser

<http://localhost:8080>

it will ask login details

username : admin

password : admin

in Jenkin we need to create the job. Every job responsible to build the project.

Day 3 :

02/09/2023

Open the terminal in VM.

git clone URL

next

git pull (but make sure terminal open inside that folder).

Web Service : Giving the service for web application when both the application running using different technologies

API : Application Programming interface.

Web Service

Java

python

XML/JSON

eXtensible markup language

JSON : JavaScript object notation

HDFC

XML/JSON

HSBC

Micro service :

Please refer the website the set the trigger time using crons

<https://crontab.guru/>

Day 4 :

03/09/2023

Open the terminal and start /stop jenkins service

`sudo service jenkins stop` unix

`sudo service jenkins start` start

user name : admin

password : admin

`java -jar jenkins.war`

<http://localhost:8080>

Jenkins provide few pre installed tools like Maven, Gradle, Git, ant etc. These tools help us to run java technologies.

Jenkins Pipeline a pipeline is a collection of event or job which interconnected with each other to perform a specific task.

Maven goal

Clean

Compile

Install jar or war unit test

Test

Package

```
pipeline {  
  agent any  
  stages {  
    stage('Hello') {  
      steps {  
        echo 'Hello World'  
        sh 'git --version'  
        sh 'v'  
      }  
    }  
  }  
}
```

Maven : Maven is open source build tool we use in java technologies to build the project.

Maven goal

Clean : clean complete project doesn't matter it contains one file or more than one file

Compile

Install : install some dependencies

Test : test the project

Package : creating jar or war file

```
pipeline {
  agent any
  tools {
    // Install the Maven version configured as "M3" and add it to the path.
    maven "M3"
  }

  stages {
    stage('Build') {
      steps {
        // Get some code from a GitHub repository
        git 'https://github.com/jglick/simple-maven-project-with-tests.git'

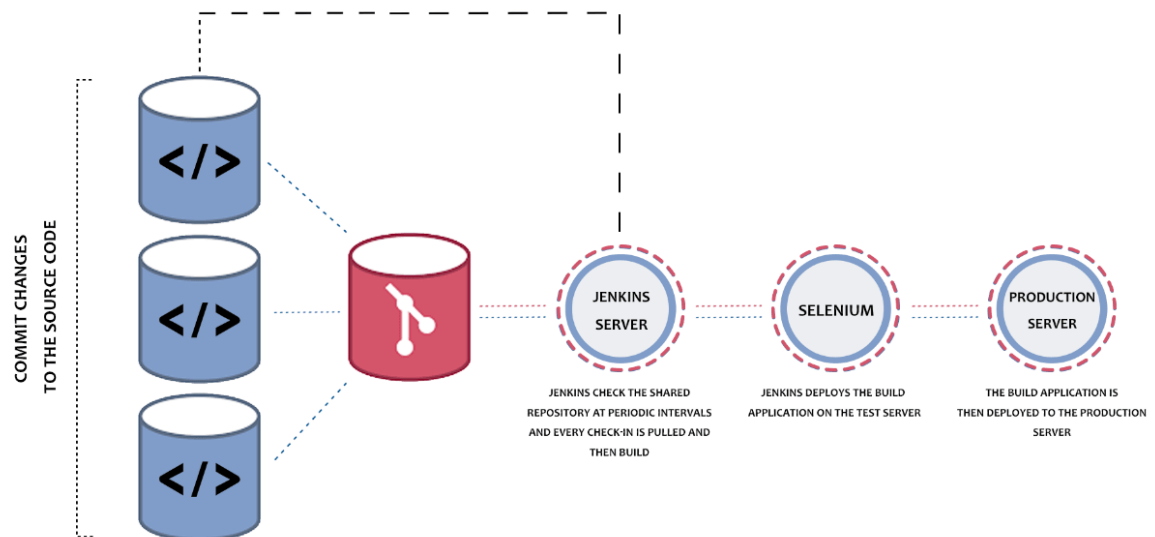
        // Run Maven on a Unix agent.
        sh "mvn -Dmaven.test.failure.ignore=true clean package"

        // To run Maven on a Windows agent, use
        //bat "mvn -Dmaven.test.failure.ignore=true clean package"
      }
    }
    post {
      // If Maven was able to run the tests, even if some of the test
      // failed, record the test results and archive the jar file.
      success {
        junit '**/target/surefire-reports/TEST-*.xml'
        archiveArtifacts 'target/*.jar'
      }
    }
  }
}
```

```

    }
}
}
}

```



If we want to run more than one command with help of normal job.

Day 5 :

Install python in VM

```
sudo apt-get install python3
```

```
sudo apt-get install python3-pytest
```

```
py ops.py
```

```
py *.py
```

or

python3 ops.py

https://github.com/Kaleakash/python_jenkins_file.git

git URL which contains jenkins pipeline script to build python program as well as run python program.

Post build

We can send notification through email

We can push this project in production environment.

We can push this project to testing environment ie Selenium

Build can be success or failure.

Docker

Few we to run any application or server or tools.

We need system software. OS ie window, linux, Unix or Mac etc.

Server name : tomcat, IIS, nginx , apache, web logic, jboss, WAS etc

Database server : mysql, oracle, db2 RDBMS

Mongo db, HBase, Neo4j no SQL etc

Tools : SAP, Info metica, IIB, ESB, Portal server etc.

VM : Virtual Machine etc.

VMWare .ios

VMWare help use to do Virtualization.

Guest OS ; it can be unix, linux, window XP etc

Limitation of VMWare or Virtualization

Base machine is Window 11 : with RAM 16 hard disk 1tb

If I want to ru Guest OS with the help of VMWare software

Window XP --→ boot up the window XP, we need provide RAM 4 gb and external hard disk 50gb. We need share the base machine resources. Etc.

We want to run 10 VM

Docker : Docker is an advanced OS Virtualization software platform which makes it easy to create, deploy and run the application in Docker container.

Container : run time environment or engine.

JRE : Java Run environment

Node JS : JavaScript run time environment.

Web Container : web container provide run time environment to run the application.

Database Container : it help to store the data in table format.

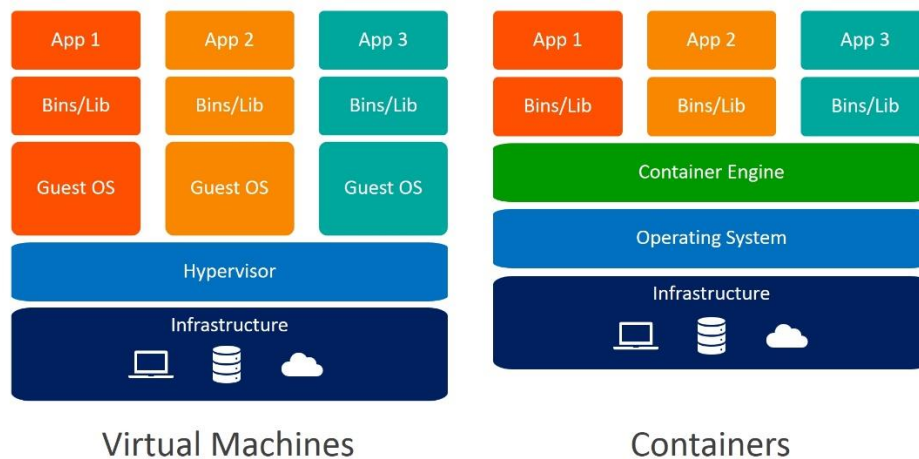
Docker Container : it is unit of deployment or software. Which contains everything to run the application. Ie code, runtime (software), tool and system libraries or database or server etc.

Docker is use to create Containerization application

Virtualization Vs Containerization

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

Containerization is the abstract version of an application or server or tool or etc.



But run Docker in base machine we require **docker engine**.

Mkdir

Open the Terminal and

Write the command as

docker --version

this command provide docker version

docker info

this command provide docker details

docker pull imageName

: this command is use to pull the image from
docker hub to local machine.

docker images

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

docker run imageName/imageld : this command is use to run the application
using docker image.

Docker pull the image by default from **Docker.hub**

Docker hub is an open source remote repository which contains lot of images

Which we can pull as well as push.

Docker hub provide private as well as public remote repository.

Docker hub is like a git hub.

In Git hub we can push any types of file or folder.

Docker hub contains docker images which is responsible to run the application

Using container.

Like other repository provided by AWS or Azure or Google cloud or private cloud etc.

Please docker hub account.

Dockerfile : A Docker file is a blue print or set of instruction that defines
How our images is build. Or Docker file use to create the image.

Docker image : Docker images contains everything to run the application.

Or

Doker image are the source code for your containers.

Using docker file we can create the image

Docker Container : instance of images or running process etc.

Once you run the image the running container become up and it will run that application which we

Mention in docker file which create image.

docker ps this command is use to display running container

docker ps -a this command is use to display all container
ie running as well as stopped mode.

docker run --name c-container hello-world this command is use
run the container with specific name or custom name

Day 6 :

10/09/2023

```

File Edit View Search Terminal Help
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker --version
Docker version 20.10.12, build e91ed57
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest   9c7a54a9a43c   4 months ago  13.3kB
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:dcba6daec718f547568c562956fa47e1b03673dd010fe6ee58ca806767031d1c
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest
akash300383gmai@ip-172-31-16-158:~/Desktop$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

```

Creating images

1. creating simple image to display welcome message.

Dockerfile

FROM busybox

CMD ["echo","Welcome docker! This image created by akash"]

docker build -t my-bosybox . -f Dockerfile

docker images

docker run bosybox

-t : tag

-f : file

2. image to run console base application

```
a=10
b=20
sum=a+b
print("sum is",sum)
```

Then create the Dockerfile

```
FROM python:3
ADD myfile.py /
RUN pip install pystrich
CMD ["python","./myfile.py"]
```

1st file pull python image with version 3

2nd add myfile in python image

3rd if required install some plugin base upon requirement.

4th open command prompt and run the python program

then create the image

```
docker build -t my-python . -f Dockerfile
```

docker image

```
docker run my-python
```

Web Application Image

Html, css, JavaScript, typescript, **angular or react** js or jquery etc.

Html etc.

Html is use to create the web page or web application.

First create the image for web application and inside the folder create the file with

Below code

```
<html>

  <head>


  </head>

  <body>

    <h1>Welcome to my web page created by Akash Kale!</h1>

  </body>

</html>
```

save the file with name as index.html

and run or open this file in chrome browser.

<http://www.google.com> : production env or live server

<http://localhost:8080> run on local machine with local server
dev env

or

<http://127.0.0.1:8080> default ip for every machine.

To run the web application we need server

Tomcat

Apache

IIS

Nginx

Etc

We use nginx open source server to create image for web application.

Server always run on port number :

Tomcat	8080
--------	------

Nginx	80
-------	----

MySQL Database	3306
----------------	------

Jenkins	8080
---------	------

Etc

If image contains web application with run server then we need to use the command as

```
docker run -d -p 80:80 imageName/imageld
```

right side port actual port number 80

publish port number can be same or different 80

-d : background or detached mode

-p : publish

index.html

```
<html>
```

```
  <head>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Welcome to my web page created by Akash Kale!</h1>
```

```
  </body>
```

```
</html>
```

save the file with name as index.html

and run or open this file in chrome browser.

Dockerfile

FROM nginx:1.6

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

docker build -t my-web . -f Dockerfile

docker run -d -p 80:80 my-web

using **docker ps**

if run please open browser and type as <http://localhost:80>

`docker stop containerId/containerName` stop container

`docker start containerId/containerName` start container

`docker rm containerId/containerName` remove container but first stop then remove

`docker rm containerId/containerName -f` without stop we can remove

`docker rmi imageName/imageId` if image is not link with any container or running container we can remove if we get error please first remove that container and then remove image

`docker rmi imageName/imageId -f`

now we will publish our local image into remote repository ie `docker hub`

`docker login`

it will ask your docker hub account
id and password

`my-web`

`docker tag imageName dockerHubAccountId/imageName:version`

version is like a tag

`docker tag my-web akashkale/my-web:1.0`

after created tag now you can push the image

`docker push dockerHubAccount/imageName:tag`

`docker push akashkale/my-web:1.0`

Application

Docker image

To run html page

Docker image

run my spring or python program,

REST API

frontend technology

backend technology

http://178.0.0.:80

html, css, js, typescript

java (spring boot)

store the data in file system
or database ie mysql or

angular or react or vue js

asp.net

oracle

python
from database.

we want store and retrieve the data
mysql database

node with express js

etc

Front end

backend

database

Angular

java

mysql

React

php

mysql

Network

network

Public

private

Public -> frontend and backend container

Private -> backend and database container

One image is responsible to run one application or modules (micro service)

Front end

backend

database

Image

image

image

Container1

Container2

Container3

http:

TCP

docker compose and Docker swarm and Kubernetes

docker compose is a toolkit which helps to run more than one container with help of **yaml or yml** etc.

Docker compose, Docker Swarm and Kubernetes are responsible to run more than one container.

Those containers execute independently as well as they can communicate with each other based upon their requirements.

Day 7 :

16/09/2023

Docker compose is a tool kit which is responsible to run more than one container using configuration file **.yml or .yaml**

docker-compose build it builds custom images

docker images

docker-compose up pull pre-defined images if required
and run all images part of that docker compose file

docker compose up --build -d

it will build it and run in background i.e. detached mode.

please open another terminal

docker ps

please verify all three containers running or not.

docker network ls it is used to verify all networks

version: '3.3'

services:

my-first-container:

image: nginx

ports:

- 80:80

my-another-container:

image: akashkale/my-web:1.0

ports:

- 81:80

docker-compose build

docker-compose up

or

docker-compose up --build -d

detached mode

docker-compose down

Docker Swarm or Docker Kubernetes

Docker compose is use to run more than one container and all container must be running in same node / same machine ie desktop or cloud machine.

Node word refer to physical machine or cloud machine or device.

Atul if we use different machine then we need to use different docker compose.

All machine are connected using ip address.

Front end Machine	backend machine	database machine
Container	container	container

If number of client increase to access the application may be front end or backend or database.

We need up scale up upon on demand.

Kubernetes : Kubernetes is container management tool or K8S. It is also known as **orchestration tool**

Orchestration tool is responsible to deploying more than one container, scheduling, scaling and load balancing, configuration etc Etc.

Kubernetes is responsible to maintain more than one container those container can be run in same machine or difference machine ie node.

Docker Swarm Vs Kubernetes

Docker Swarm is part of Docker

1. No auto scaling
2. Does auto load balancing
3. Easy to develop the application
4. No GUI

Kubernetes is part of google

1. Auto scaling (up and down on demand)
2. We can do manually auto load balancing
3. Complicate to do configuration.
4. We can use GUI base upon tools.

Node : Node refer to machine or device or physical machine or cloud machine.

Cluster : it is a collection of host or combination of node (server or client). That helps you to aggregate their availability of resources. Like RAM, CPU, Disk, pool etc.

Public and private id address

<http://198.78.56.45:80> public outside a cluster.

<http://198.1.2.56> private

<http://198.1.2.57>

<http://198.1.2.58>

<http://198.1.2.59>

<http://198.78.56.45:80>

Namespace : it is a logical cluster or environment. Namespace is like package. It is widely used method which is scoping or dividing a cluster.

Java application → **java-app**

Python application → **python-app**

Angular application → `angular -app`

Pods : Kubernetes is responsible to run more than one container. In Kubernetes container can't communicate with each other directly. All container must be wrap in a functional unit and that unit is known as pods.

Each pods are responsible to run one container or more than one container.

Node contains more than one pods. Each pods can contains more than one container and each container responsible to run one application.

Project

We can create one cluster or more than one cluster

Inside each cluster we can add one or more than one node.(machine).

Each node contains one or more than one pods. Those pods are part of same namespace or different namespace.

Each pods contains more than one container. And each container responsible to run the application. That application can be java, python, php or node js.

Kubeadm : tools provided by Kubernetes which help to develop Kubernetes application

Unix or Linux non window. Kubeadm support cluster features.

Minikube in your local machine.

Minikube ie open source tools which provide single cluster environment for Kubernetes

To deploy the application.

It is GUI base.

kubeadm

kind

Docker desktop

Service

Day 8 :

17/09/2023

Please Pull repository in your local machine or VM if Git present or download

<https://github.com/Kaleakash/docker-compose-repository.git>

then open the terminal

```
docker-compose --version
```

```
docker-compose up --build -d
```

docker images

docker ps

after running on container

open the browser

<http://localhost:81>

then application open

store the data

if you want to stop

```
docker-compose down
```

```
docker run -it alpine
```

```
apk add openjdk11
```

```
apk add git
git clone https://github.com/Kaleakash/jenkinjava.git
cd jenkinjava
javac Demo.java
java Demo
```

Dockerfile

```
docker run -it alpine size
apk add openjdk11 size
apk add git size
git clone https://github.com/Kaleakash/jenkinjava.git
cd jenkinjava
javac Demo.java
java Demo
```

Dockerfile

```
FROM openjdk:8
COPY Demo.java .
RUN javac Demo.java
CMD ["java","Demo"]
```

Git --> Git Hub -> Jenkin ->

In Jenkin We can run docker image (Jenkin Pipe Line)

VSCode editor

<https://code.visualstudio.com/download>

Please create Jenkin pipeline job

And provide Git URL of my project

<https://github.com/Kaleakash/docker-compose-repository.git>

`docker-compose up --build -d`

but make sure docker running as well as docker-compose running.

Or

Create index.html

Create the image for index.html

Please create docker compose file to create image and run the container.

Push this code in git hub

Then create jenkins job or pipeline with trigger and run docker-compose file in jenkins environment

Then in Jenkins we need to configure authentication details for Docker hub.

AWS we need to create three instances

All required software we need to install.

Instance type must be medium 2 CPU

Master Node

Worker1 node

Worker2 node