Docker

=> It is

containernization platform

Tool

=> Containernization means code + environment or

(not just jar files,, lots of dependent softwares are required) code + dependencies packing

Where code is our app/project code and env/dependencies are like OS, JVM, web server s/w, DB, jar files /libraries and etc..

=> Docker is platform independent tool i.e

the code can be in any language and env.. can be there in any setup

=> Docker Tool makes our code easily portable and deployable across the multiple machines

Orchestration is managing the docker container

by enabling feature called autoscaling i.e increases or decreases docker containers creation as needed

For Orchestration, we need to use kubernates tool (k8s)

=>Jenkins is CI/CD tool i.e it automates the build and deployment process =>Docker is Containernization Tool i.e it packs both code and environment =>Kubernates(K8S) is Orchestration Tool i.e it enables auto scaling on docker containers (docker instance having

(UI

--user Interface)

Front End

Project Application stack (Typical new projects)

Angular/ReactJs with

html, css, java script, bootstrap (or)

(Business/Service logics) Including persistence logics)

(Data Store)

Back End

Data base

Java (spring boot Rest/MS/data jpa/...) NET python

Node JS

code+ env.. packing)

+

oracle, MySQL,sql server, PostgreSQL MongoDB. Cassendra

(Spring boot mvc with jsp files

is bit outdated front end)

Profiles / Environments in Project development to Release to Production

==========

==========

=> Setting up profile is all about keeping the env/setup ready that is required for the Project execution

different profiles/env.. we generally develop/ test the code at various levels to make sure

=> Using these setups of

code is executing is smoothly

Code Repository /VCS Software

Project Code

GIT Hub Repo

Profiles/ Env.. For Testing Project

Dev Profile /env

-> jdk 11

-> tomcat 10

(For software

Company)

& &

=>

For every Profil, we need keep one Linux VM Instance of AWS ready to develo, deploy and test project

=> Dev profile VM is or both development and unit testing

=> Test /UAT /Pre Prod profile VMs ar purely for Testing at various levels

=> Prod profile VM is pure for keeping the Project in Live code for endusers

--> Window OS

-> MySQL

-> .....

(SIT Profile /env)

Test Profile /env

-> jdk 11

-> tomcat 10

--> windows

-> mySQL

note:: The Tests that happens at client organization after releasing the Project is called UAT (User acceptance Test)

Prod Profile /env

(For software Company)

UAT Profile /env

Pre-Prod Profile /env

-> jdk 11

-> tomcat 10

(For client org)

-> jdk 11

-> wildfly

-> Linux OS

-> Oracle

-> jdk 11

-> wildfly

-> Linux OS

-> Oracle

(For Client Org)

(For Client Org)

Life with out Docker

=========

-> Linux OS

-> mySQL

Team

=> Dev Ops team is responsible to keep these Profiles /env.. related machines ready to develop/test the project in different env... => if Docker is not there , the dev Ops needs to keep all the profiles/envs.. very much ready manually which is complex process => Some times

software company machines setup of development and Test/SIT (System Integration Test) may not match with Client Org's machines setup (Cloud or outside Cloud) (we generally observe the following mismatches)

Company

(Client org)

=> software uses window machines/mac mac machines for development but they use Linux/ unbuntu machines have

=> Software company might developed the java code using java 11 setup, But the client org may provide java 8 setup

=> Software company might have used Oracle 11g, but the client organization may provide oracle 19c setup

and etc..

(all these indicates compitablity issues run the Project code in different profile/env.. machines)

=>if s/w company releases the software product to multiple client orginizations

then keeping the setup ready for all those client organizations becomes big headache for for the software company

of

To solve these problems take the support containerization using the Docker tool where the code and env will be packed

Life with Docker

========

====

=> Docker automates the env/profile setup process by using the concept called containerization which

performs code + env packing

Dev Container

Test Cotnainer

UAT Container

Proj code+

tomat,...

windows

env like

DB s/w,jdk,

Proj code

Proj code

env like

env like

DB s/w,jd,

DB s/w,jd

tomat,...

tomat,...

Linux

linux

Container Engine (Docker engine)

HOST OS (regular windows, linux and etc..)

=>

Docker Containernization

Front

End comps

Backend Comps

DB s/w

HOST

OS will change based on setup of the profile

docker

=>Docker logo is ship carrying comps

indicating docker makes the codeEnyeasily

shippable code +Env.. (Docker containers which hold code +env.. are easily shippable or portable across the multiple machines)

Libraries

Operating

Systems

Virtual Machines

Servers

When we move Docker container to any Machine that machine just need to have Docker software or engine So that docker managed code can be executed in the software setup of Docker container itself

=> Once

any machine'

we keep Project code + env setup in the Docker container, we can make that docker container working in having any OS for any developer or end user

=> Docker passes the statement to Programmers/Developers

u just keep Application code ready becoz the Docker it self arranges all s/ws, OS and etc.. that are required to execute the code in different profiles /environments Docker Definitation

==========

Docker is a tool or platform for packaging, deploying and running the Applications with portability

Advantages with docker

==========

and

a) Docker enables the developers/devops team to separate our application/project the intrastructure (hardware setup) so that we can deliver software project quickly

+ env..

Docker

b) Docker packages the software/project and its env/dependencies into units called containers that have every thing

that the project needs to run including libraries(jar files), tools, code and Run time setup

c) By using the docker style packaging, deployment, execution and shipping which is quick process which reduce the time delay beween developing the code in Dev and running the code prod Profile

and

profile

in

Docker Container

Project code

+

Environment like

OS, DB, JVM, server

Libraries and etc..

Docker Architecture (High Level)

======

(dockerFile)

Build

Docker File

Run

of

(inputs)

Docker Image (Planning packaging) (Building plan as Miniature)

Docker Container (code + env packing (Instance of packaging) (real building)

=>We release docker containers to different machines to execute the code of Docker Container in the env or setup that is there in the Docker it self (After all docker Conntainer maintains both Code and env..)

(Like Building plan on paper)

Main Comps of Docker Architecture

=>Docker file (Set of instructions to build the docker Image)

=> Docker Image (Package which contains the planning of Source code + env packing)

=> Docker Registry (The registry where docker images are available : eg:Docker HUB, Aws ECR and etc..)

=> Docker Container (An Instance of Docker image using which we can run our application code

in certain env.. /profile that is specified in the docker image)

=> Docker Client (An implicit comp in the docker tool or software who is actually resposible to execute the Docker commands)

Keeping Docker desktop software ready in windows OS

==========

https://docs.docker.com/desktop/install/windows-install/

step1) gather system requirements for isntalling Docker desktop on windows

WSL 2 backend, x86_64 Hyper-V backend, x86_64 WSL 2 backend, Arm (Beta)

WSL version 1.1.3.0 or later.

•

Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.

WSL 2 backend, x86_64 Hyper-V backend, x86_64

WSL 2 backend, Arm (Beta)

• Windows 10 64-bit:

•

•

We recommend Home or Pro 22H2 (build 19045) or higher, or Enterprise or Education 22H2 (build 19045) or higher.

Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.

Windows 10 64-bit:

•

Minimum required is Home or Pro 21H2 (build 19044) or higher, or Enterprise or Education 21H2 (build 19044) or higher.

•

We recommend Home or Pro 22H2 (build 19045) or higher, or Enterprise or Education 22H2 (build 19045) or higher.

•

Turn on the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation

•

.

Minimum required is Home or Pro 21H2 (build 19044) or higher, or Enterprise or Education 21H2 (build 19044) or higher.

•

The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:

•

Turn on Hyper-V and Containers Windows features.

The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:

•

64-bit processor with Second Level Address Translation (SLAT) [

•

4GB system RAM

Enable hardware virtualization in BIOS. For more information, see Virtualization.

•

64 bit processor with Second Level Address Translation (SLAT) [

•

4GB system RAM

•

Turn on BIOS-level hardware virtualization support in the BIOS settings. For more information, see Virtualization.

wsl2:: windows sub system for linux

step2) make sure that wsl2(windows sub system for Linux) and Hyper-V and installed in the computer and it is restarted

=> In windows search box, search for turn on windows features on or ff ---> select Hyper-V, windows sub system for Linux checkboxes--> ok ---> installation process -----> restart the computer

step3) confirm the wsl2 installation

```
C:\Users\Nataraz>
--status
C:\Users\Nataraz>wsl
Default Distribution: docker-desktop
Default Version: 2
C:\Users\Nataraz> wsl --version
WSL version: 2.2.4.0
Kernel version: 5.15.153.1-2
WSLg version: 1.0.61
```

MSRDC version: 1.2.5326

Direct3D version: 1.611.1-81528511

DXCore version: 10.0.26091.1-240325-1447.ge-release Windows version: 10.0.22631.3880

note:: wsl2 provides Linux setup in windows OS where we can execute Linux and Docker commands

note:: Hyper-V is windows supplied driver s/w to make Docker software compitable with Windows OS

step3) download

o-

docker desktop for windows and install it

docs.docker.com/desktop/release-notes/

docker.docs Guides Manuals Reference

rview

Docker

ker Desktop

^

verview

stall

gn in

<plore Docker Desktop

ardened Docker Desktop

ev Environments (Beta)

ontainerd image store

Manuals Docker Desktop / Release notes

Docker Desktop release notes

This page contains information about the new features, improvements, kno Docker Desktop releases. For frequently asked questions about Docker Des

Docker Desktop versions older than 6 months from the latest release are nc

Take a look at the Docker Public Roadmap to see what's coming next.

asm workloads (Beta)

/nchronized file shares

4.33.1

2024-07-31

SL

PU support

in

step5) create an account docker.com

step6) Launch docker desktop tool

Download Docker Desktop

Windows (checksum) | Windows ARM Beta (checksum) |

Docker Desktop Installer.exe

(use this file for installation)

(restart the computer)

before launching desktop tool

step) execute the following commands on the command prompt to confirm the docker desktop installation

```
C:\Users\Nataraz>docker --version
Docker version 27.1.1, build 6312585
C:\Users\Nataraz>docker info
Client:
Version:
27.1.1
Context: desktop-linux
Debug Mode: false
Plugins:
buildx: Docker Buildx (Docker Inc.)
Version: v0.16.1-desktop.1
Path: C:\Program Files\Docker\cli-plugins\docker-buildx.exe
compose: Docker Compose (Docker Inc.)
Version: v2.29.1-desktop.1
Path:
C:\Program Files\Docker\cli-plugins\docker-compose.exe
debug: Get a shell into any image or container (Docker Inc.) Version: 0.0.34
Path: C:\Program Files\Docker\cli-plugins\docker-debug.exe
desktop: Docker Desktop commands (Alpha) (Docker Inc.)
Version:
Path:
v0.0.14
C:\Program Files\Docker\cli-plugins\docker-desktop.exe dev: Docker Dev Environments
(Docker Inc.)
Version: v0.1.2
Path: C:\Program Files\Docker\cli-plugins\docker-dev.exe extension: Manages Docker
extensions (Docker Inc.) Version: v0.2.25
C:\Users\Nataraz>docker search jdk
NAME
java
vespaengine/vespa-pipeline
chainguard/jdk
```

eclipse/centos_jdk8

chainguard/jdk-lts

eclipse/ubuntu_jdk8

dockette/jdk8 eclipse/alpine_jdk8

DESCRIPTION

STARS

OFFICIAL

DEPRECATED; use "openjdk" (or other JDK impl... Docker image with supported versions of Mave... Minimalist Wolfi-based Java JDK image using CentOS, JDK8, Maven 3, git, curl, nmap, mc, Minimalist Wolfi-based Java JDK image using Ubuntu, JDK8, Maven 3, git, curl, nmap, mc, My Oracle Java 8 Dockerfile Based on Alpine 3.3. JDK 1.8, Maven 3.3.9, T... 1

2008

[OK]

[OK]

[OK]

****

5

…

[OK]

****

17

…

5

C:\Users\Nataraz>docker

search openjdk

NAME

DESCRIPTION

openjdk eclipse-temurin

adoptopenjdk

sapmachine

ibm-semeru-runtimes

cimg/openjdk

Pre-release / non-production builds of OpenJ... Official Images for OpenJDK binaries built b... 573 DEPRECATED; use https://hub.docker.com/_/ecl... Official SapMachine Docker Image, SAP's buil... IBM Semeru Runtimes Official Images for Open... 46 The CircleCI OpenJDK (Java) Docker Convenien...

STARS 3938

376

OFFICIAL [OK] [OK] [OK]

51

[OK]

[OK]

8

we can search for the ready made docker images either from command prompt the commands as show above

or we can do the same search in the GUI secreens of Docker desktop

open docker setup ---> go to images section --->search images to run ---> search image like open jdk, MySQL,oracle,tomcat and etc...