1. AWS, Assure, JCP, Docker, Kubernet

2.Automation - Software Delivery Process-Python, Rubby, Go

3.CI/CD-Github and Jenkins

4.Infrastructure as a Code-Ansible and Terraform

7.Monitoring-Grafana and Prometheus

1.Azure-Docker

2.Automation-Python

3.CI/CD-Github and Jenkins

4.Infrastructure as a code-Ansible

5.Monitoring-Grafana

Github have mainly comand -1.git config --global user.email "208r1a6645@gmail.com"

2.git config --global user.name "Prateek"

3.git config --list

Git- in vs code two command are there Clone and status

1.git clone "file name"

2.cd change directory

3.ls - it used to show how many file are there

4.git status

Add and Commit

-git commit-m "some message"

Push Command to push github to vs code

-git push origin main

Command to back to previous directory

-cd ..

-mkdir LocalRepo

-----------------

-git init

-git remote add origin https://github.com/Prateek2134/localRepo.git

-git remote -v

-git branch

-git branch -M(name)----> using for changing branch

-git push -u origin main

-----------------------------

Docker / Container / Kubernets

Ans. To create a package, which we want to use it and deploy on it in Local Machine or in the Data Center or in the CLoud. For example you want the package to contain the OS maybe something like Linux , Windows amd may be in the runtime Python and as well your application .

How to create such a package - Docker

to create a Docker Image , once u have a Docker Image, Docker runtime to run this specific image wherever u want to run it. Once you have a Docker Image in a running state, that we can call it has Container.

Kubernetes is popular Container.

----------------------------------------------------------------------------

Docker Image

Commands:

attach Attach local standard input, output, and error streams to a running container

commit Create a new image from a container's changes

cp Copy files/folders between a container and the local filesystem

create Create a new container

diff Inspect changes to files or directories on a container's filesystem

events Get real time events from the server

export Export a container's filesystem as a tar archive

history Show the history of an image

import Import the contents from a tarball to create a filesystem image

inspect Return low-level information on Docker objects

kill Kill one or more running containers

load Load an image from a tar archive or STDIN

logs Fetch the logs of a container

pause Pause all processes within one or more containers

port List port mappings or a specific mapping for the container

rename Rename a container

restart Restart one or more containers

rm Remove one or more containers

rmi Remove one or more images

save Save one or more images to a tar archive (streamed to STDOUT by default)

start Start one or more stopped containers

stats Display a live stream of container(s) resource usage statistics

stop Stop one or more running containers

tag Create a tag TARGET\_IMAGE that refers to SOURCE\_IMAGE

top Display the running processes of a container

unpause Unpause all processes within one or more containers

update Update configuration of one or more containers

wait Block until one or more containers stop, then print their exit codes

Global Options:

--config string Location of client config files (default

"C:\\Users\\raopr\\.docker")

-c, --context string Name of the context to use to connect to the

daemon (overrides DOCKER\_HOST env var and

default context set with "docker context use")

-D, --debug Enable debug mode

-H, --host list Daemon socket to connect to

-l, --log-level string Set the logging level ("debug", "info",

"warn", "error", "fatal") (default "info")

--tls Use TLS; implied by --tlsverify

--tlscacert string Trust certs signed only by this CA (default

"C:\\Users\\raopr\\.docker\\ca.pem")

--tlscert string Path to TLS certificate file (default

"C:\\Users\\raopr\\.docker\\cert.pem")

--tlskey string Path to TLS key file (default

"C:\\Users\\raopr\\.docker\\key.pem")

--tlsverify Use TLS and verify the remote

-v, --version Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

C:\Windows\System32>docker --version

C:\Windows\System32>docker run hello-world

------------------------------------------------------

-docker --version

-docker run hello-world

-docker images

- docker pull <image names>

-docker pull <image names>

-docker pull openjdk:20

-docker search <image names>

To view Container Command is:docker ps

To view all Container-docker ps -a

-docker run --name pythonContainer -d python

- docker run --name pythonContainer1 -it -d python

Execution-docker exec -it b3728bbb9682 python3

-----------------------------------------------------------

We Can take images from https://hub.docker.com/search?q=

---------------------------------------------------------

>>> class test:

... pass

...

>>> print(ob)

<\_\_main\_\_.test object at 0x7f98c3a8b770>

>>> ob=test()

>>> print(ob)

<\_\_main\_\_.test object at 0x7f98c3af12e0>

>>>

-----------

-it means Intractive mode to run continue

--------------------------

java run in docker command:

docker exec -it javaContainer jshell

System.out.print("12")

String str="q34"

str.length()

5+453

exit()

Exit()

clear

cls

exit()

/history

jshell> /exit

| Goodbye

-----------------------------

MySQl Command:

PS C:\Users\raopr\Desktop\Docker> docker run --name mysql -e MYSQL\_ROOT\_PASSWORD=root -d mysql

-docker inspect mysql

-------------------------------

-docker login <name>

-docker commit <name>

-docker push <name>

-docker network <name>

-docker copy <name>

-docker logs <name>

-docker volume <name>

-docker logout <name>

-------------------

Docker Image:

First create a folder as name of ubuntu-image to stores images in it take path in command line.

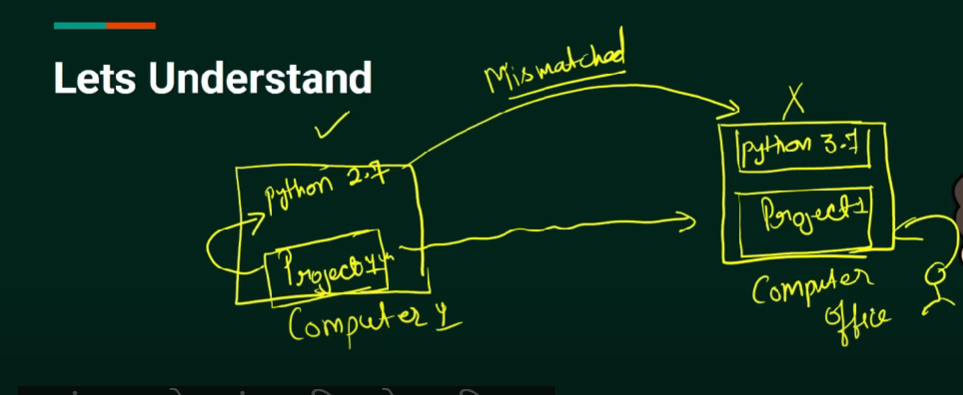
PS C:\Users\raopr\Desktop\ubuntu-image> cd .\ubuntu-image\

PS C:\Users\raopr\Desktop\ubuntu-image\ubuntu-image> code .

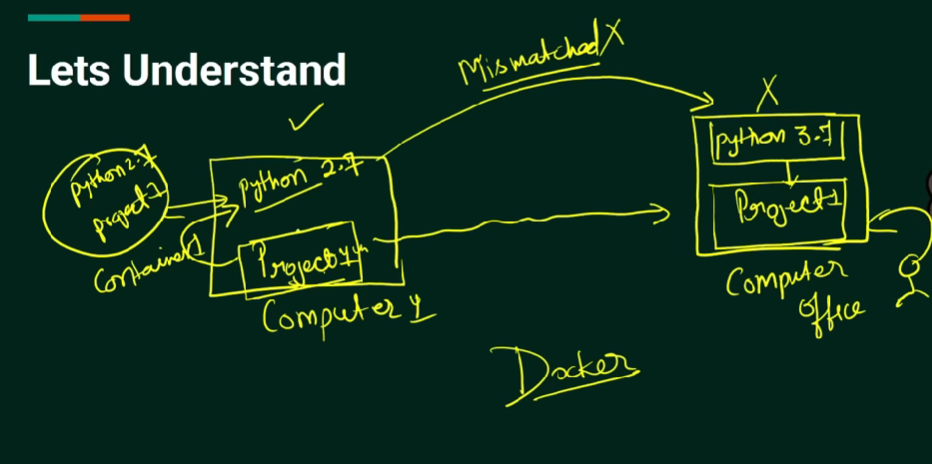
Go to VS Code

- docker build -t myubuntuimage .

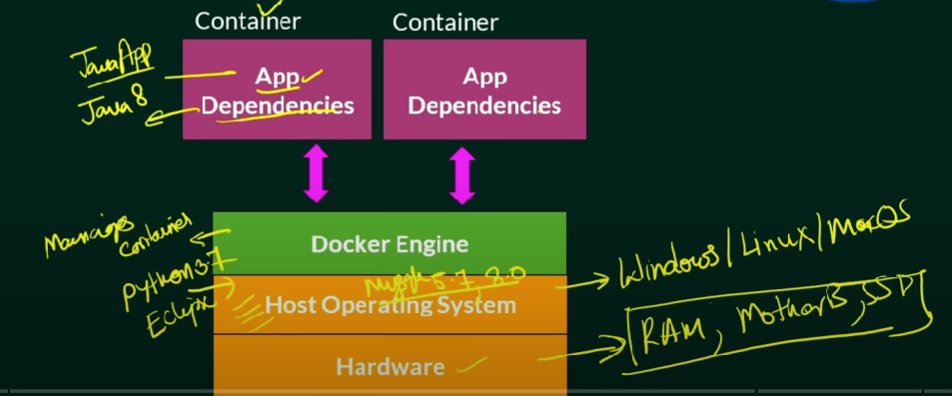
DOCKER:



SOLUTION:DOCKER



DOCKER ARCHIETECTURE:



HOW TO BUILD DOCKER WITH ASSURE

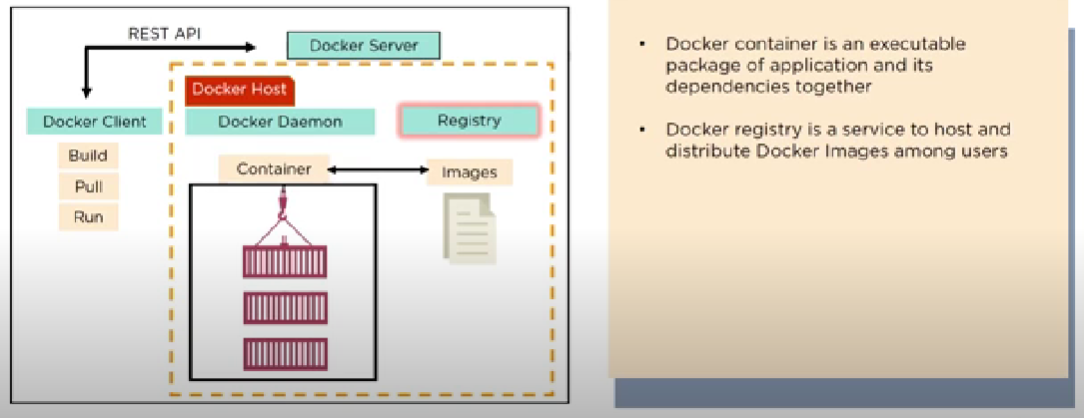
Build custom Docker images using [Azure DevOps Hosted Linux agent](https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/agents?view=vsts" \t "https://azuredevopslabs.com/labs/vstsextend/docker/_blank)

Push and store the Docker images in a private repository

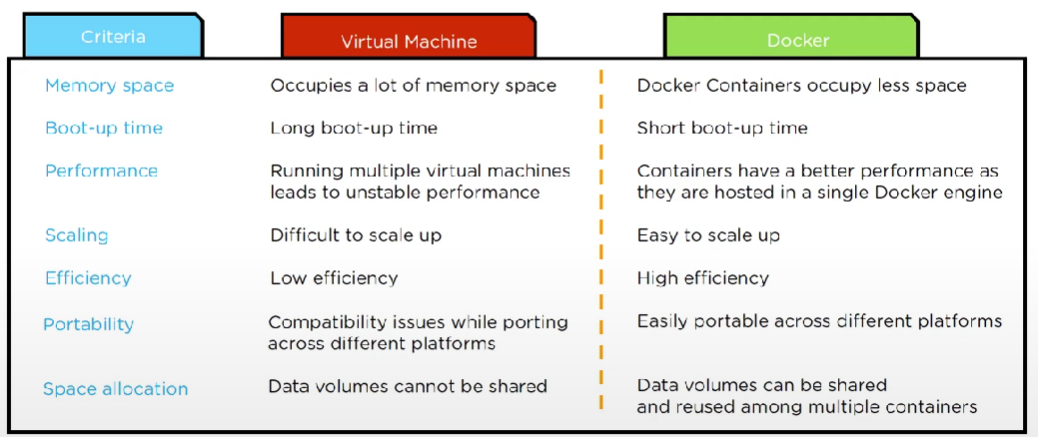
Deploy and run the images inside the Docker Containers

Interview Questions for Docker:

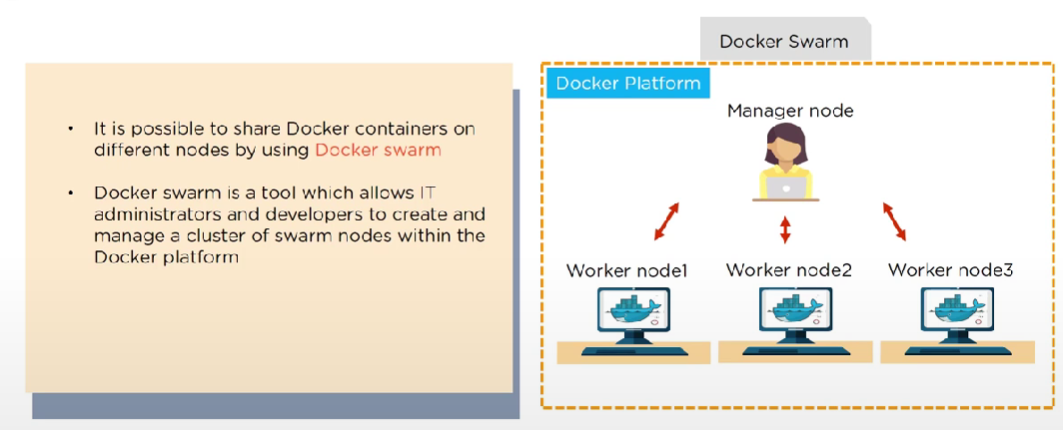
1. Archeitecture of Docker

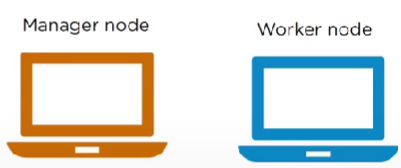


1. Advantages of Docker over Virtual Machine



1. How do we share Docker containers with different nodes.



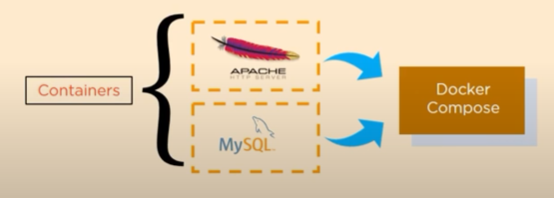


1. WHat are the commands to create a Docker swarm.

Docker swarm init --advertise-addr<MANAGER-IP>

1. How to run multiple containers using a single service.

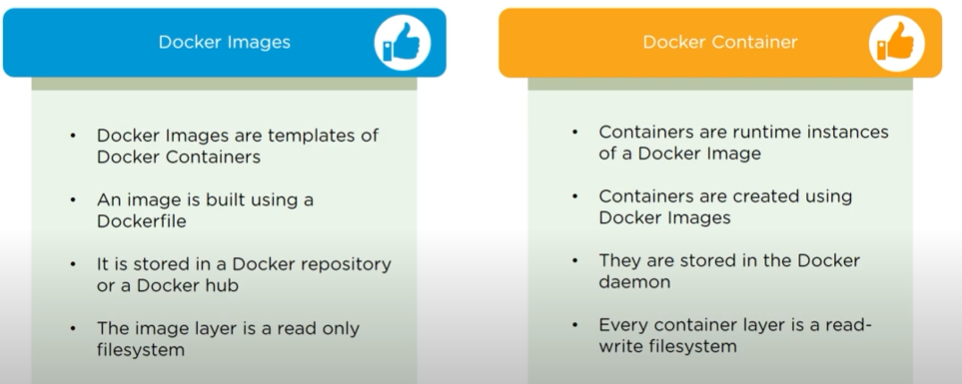
By using Docker Compose



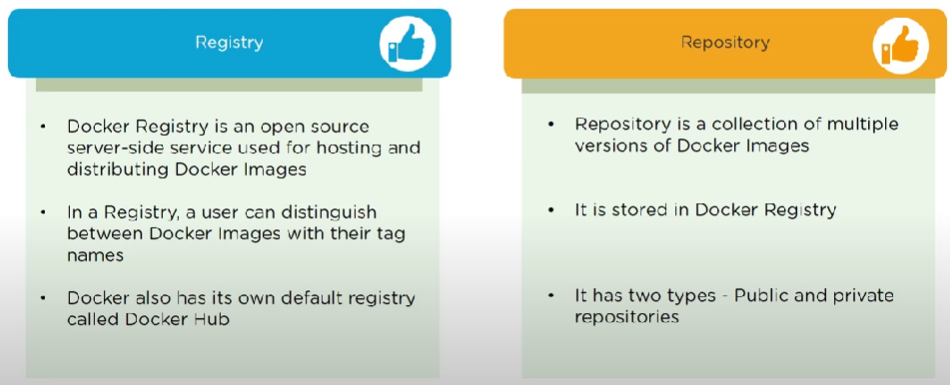
1. Dockerfile: It is used for creating Docker images using the build command.

DOCKER FIle-->DOCKER Image--->DOCKER Container-->DOCKER Hub-->Containers

1. Differences between Docker image and container.



8.Registry VS Repository



1. What Cloud platforms that support Docker

AWS, Azure and Google CLoud

ANSIBLE:

It is an open source software , uisng for configuration management and application deployment.

It use its own declarative lannguage to describe system configuration-YAML.

WHY:

.Bring huge time saving

.Agentless

.Playbooks are easy to read and edit

.It is open Source

Some Command:

1. sudo apt update
2. sudo apt install x11-apps
3. lsb\_release -a
4. sudo apt-get update
5. sudo apt-add-repository ppa:ansible/ansible
6. sudo apt-get install software-properties-common
7. sudo apt-get install ansible--- ansible installation

Linux eye command: xeyes &

Calculator command: xcalc

1. It is an automation
2. It is a push operation

Node1

Archietecture:

ssh

ssh

ssh

Node3

Node2

Ansible Master