

DRs

↓ Docker:

(developing, shipping & running application)

✓ platform

Open

Sol<sup>n</sup> Docker

→ Separate application from infrastructure (to deliver the quickly)

⇒ Dependency hell in software development

- Complexes of interdependencies among the components.
- Causing conflict
- Version issues & inconsistent behaviour

Consequence:- delayed services

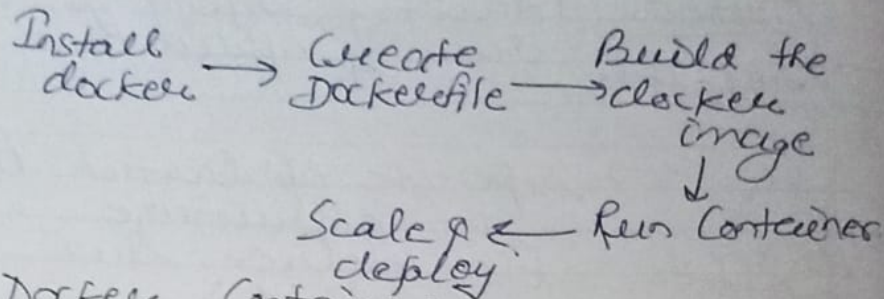
Sol<sup>n</sup>:- (Docker):- Encapsulating applications & their dependencies into self-contained units

⇒ How Docker address the problem

- Consistent environment
- Version Control
- Reproducible builds
- Scalability & flexibility
- faster development workflow



⇒ Implementing docker in your development workflow:-



⇒ Docker Containers:-

- light weight
- standalone
- executable package
- secure (isolation capabilities)

⇒ Docker desktop:-

is a complete end to end application for docker environment on desktop machine for developers.

⇒ Docker engine:- Server side installation, will be preferred for servers, cloud machines & it's only compatible with linux.

→ Docker CE (free)

→ Docker EE (paid)

→ EE = support + Enterprise support

⇒ Docker backbone

• Build → Ship → Run



②

development

build the  
image  
↓  
Container

package  
isolation  
(libraries)

and application  
on.

developers.

side

refered

machines &  
with

l)

- f  
cube  
uppost

Run

Note:- hypervisors allow multiple VMs to run on a single machine.

Note:- Inside a Container all the necessary executable, binary codes, libraries & configuration files.

→ Containers do not contain O.S images. this makes them more lightweight & portable with significantly less overhead.

⇒ Docker is all about speed.

→ develop faster

→ Build faster

→ Test faster

→ Deploy faster

→ Update faster

→ Recover faster

very large  
gigabytes

lots of  
system  
memory

↓  
VMs

→ more  
efficient

Container

piece of space  
allows you to  
install other  
stuff inside of it

• shared OS  
kernel.  
• lightweight  
• standalone

more  
space  
are

⇒ Internal processing of docker:

Look for the image in image  
cache (local memory)



↓  
look in remote docker repository  
(default hub.docker.com)  
(we can create our own rep.)

↓  
download latest version  
of image

↓  
Create new Container  
based on the image

↓  
give it a Virtual IP  
on private net  
inside docker engine

↓  
Open port 80 on host  
machine & route to  
port 80 inside Container

↓  
start Container

\*\*  
types of network driver

- ↳ bridge (default)
- ↳ host
- ↳ none
- ↳ Overlay
- ↳ IPvlan
- ↳ macvlan

→ Docker image? Read only template.

→ A dockerfile is a text file that  
contains instructions for building  
image

- ↳ base image
- ↳ configure setting.



⑦

→ Docker Compose: It is a utility used to define & run multi-container Docker applications.

\* It uses a YAML file (docker-compose.yml) to specify services, networks, & volumes for an application, allowing developers to define & manage complex application environments with ease.

⇒ Docker Swarm: to create & manage a cluster of Docker host called nodes to deploy & scale containerized application across multiple machines.

Note: Docker Swarm is a simpler, built-in orchestration tool provided by Docker, while Kubernetes is a more feature-rich, standalone container orchestration platform.

Kubernetes offers advanced capabilities for scaling, self-healing & managing containerized application at scale, making it suitable for complex production-grade environments.



⇒ Docker Container can store data  
Using docker volumes or bind  
mounts

→ Volumes are managed by docker  
& persist data even if the  
Container is stopped or  
removed.

→ Bind mount link a directory  
on the host machine to a  
directory in the container,  
allowing for direct access of  
host files

⇒ Docker hub

- Cloud based repository
- Storing & sharing docker
- Centralized image location
- find distribute & collaborate
- Image versioning
- Automated builds.

Docker

- 6 Page notes. ] 1 hr
- 7 S.S.
- 1 Pdf Command. ] 1 hr