NOVEMBER 2022

SCALER DOCKER MASTERCLASS

Akhil Sharma
Armur A.I

Shipping Software

Problem 1

Software works in one machine, but not in another.

Problem 2

Dependencies don't work well together - version issues.

Problem 3

Moving parts (Ubuntu, PHP, SQL) version issues.

03

NOVEMBER 2022

Collaborating With Developers

Problem 1

Sharing complete working projects with others is a pain.

Problem 2

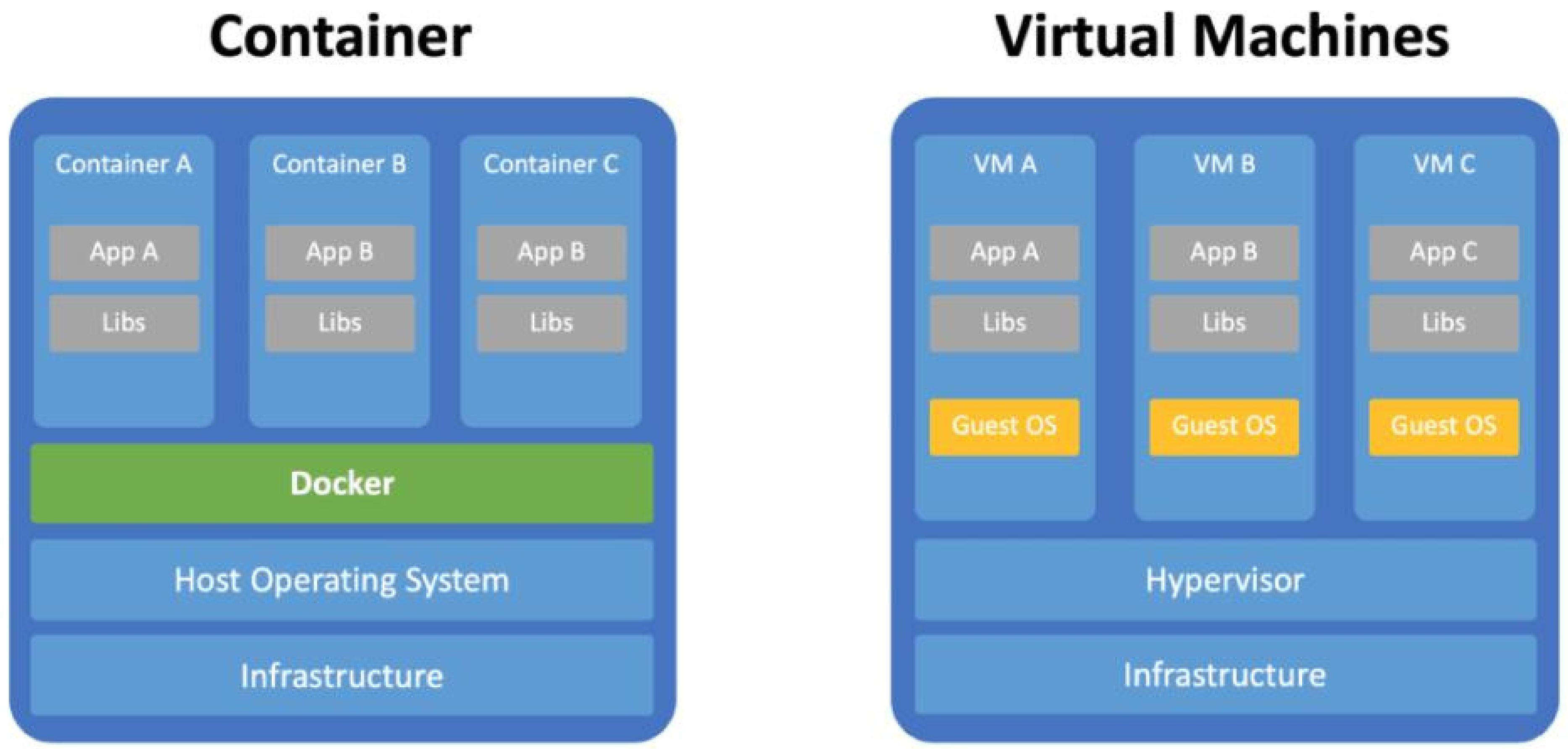
Re-install everything from scratch on new machines and servers.

Problem 3

OS issues - what works on Ubuntu doesn't on Mac or Windows

Solution

Portability
OS Independent
Dependency Packaging
No need to re-install everything, just Docker
Environment Isolation



Source: https://miro.medium.com/max/1024/0*zBEamnh9NKe7Rmdk.png

NOVEMBER 2022

Important Terms

Images

A template used to "build" a "container". Similar to a snapshot in a VM.

Containers

A run-time instance of a docker image

Docker File

Contains the commands required to assemble an image.
You can build images from this

Docker Hub

Place to host your images and share with the world.

Daemon

docker'd' listens to events and managers images, containers, networks, volumes.

Registry

Stateless, highly scalable server side app that stores and let s you distribute docker images.

Docker Features

Re-use

Base Docker Images

Shared Library

Single place for multiple images

Dependency Management

Containerize all dependencies and avoid version errors.

OS Independence

Can work on windows, mac and linux (but under the surface, mac uses linux hyprvisor and windows uses linux subsystem)

Versioning

Track versions, roll back to previous versions, easy.

Scalable

Can start off multiple containers to scale a service

Flexible

Can add or remove containers with ease, even if containers fail

Secure

Not VM level security, but basic security is present.

But really, Why Docker??

Cost Effective = save money
Granular Updates = save time
Speed of Deployment = save more time + happy clients
Smaller Devops Staff = save more money

NOVEMBER 2022

Docker Hub

Docker hub website

Global Repo

Get access to container images from developers across the world

Private Repo

Share private repos with your team (has authorization) for use on projects

Pull

Directly pull from docker hub, images are scanned and secure

Docker official images

Based by docker, not from 3rd party developers

Builds and Webhooks

github, gitlab integrations and triggers events after successful push to docker hub

Comparison

VMs	Docker
Takes minutes	Takes seconds
Separate instances required for deployment	Easy deployment, requires a single image
Entire OS needs to be loaded before starting, so less efficient.	Requires less memory
Own kernel, so higher security privileges	Containers share the host kernel, can be compromised.

Docker Lingo

Prune

Deletes all stopped containers, all unused and dangling images

Unused

Images that haven't been assigned or used in a container

Dangling

A new build of the image was created, old image now useless.

Sandbox

Apps are isolated with container sandboxes. Ideal for workloads that require application-level security and isolation.

Client

Client is an app responsible for sending the commands, daemon picks them up

Pull and Push

Downloading images and uploading new images

Volumes

Preferred mechanism for persisting data generated by Docker Containers.

Different from bind mounts, these are completely managed by docker.

Networks

Link multiple docker containers together

A single container can be a part of multiple networks

"Bridges" are private default networks created

Services

Run multiple containers, volumes and networks together at the same time.
Clubbing these is called a "service"

Compose and Swarm

Docker compose enables you to start and stop services

While docker compose does it on a single host, swarm is used for orchestration across multiple hosts.