DEVOPS FOUNDATIONS

CERTIFICATE

- Zainuddin Saiyed -

SESSION - 2

SHIFTKEY ACADEMY

VIRTUAL MACHINES?

What is Virtual Machines?

A **virtual machine**, or **VM**, is a digitized version of a physical computer. Virtual machines can run programs and operating systems, store data, connect to networks, and do other computing functions. However, a VM uses entirely virtual resources instead of physical components [1].

Key Characteristics of VM:

- **Strong isolation** between VMs (Each VM runs its own complete operating system)
- **Resource-intensive** (VMs require more system memory and storage)
- **Slower to start up** (as they take up a lot of space and can be slow to start)
- **Better security** (Strong isolation between VMs and the host system)

Read More: [1] https://cloud.google.com/learn/what-is-a-virtual-machine

CONTAINER CONCEPTS

SHIFTKEY ACADEMY

What are Containers?

A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another [2].

Key Characteristics of containers:

- Shared OS (Containers share the host computer's operating system)
- Lightweight (These are much smaller and faster than VMs)
- Efficient resource utilisation (Each container has just what the app needs to run)
- Quick start time
- Higher portability (Easily moved between different environments)

Read More: [2] https://www.docker.com/resources/what-container/

Containers vs VMs

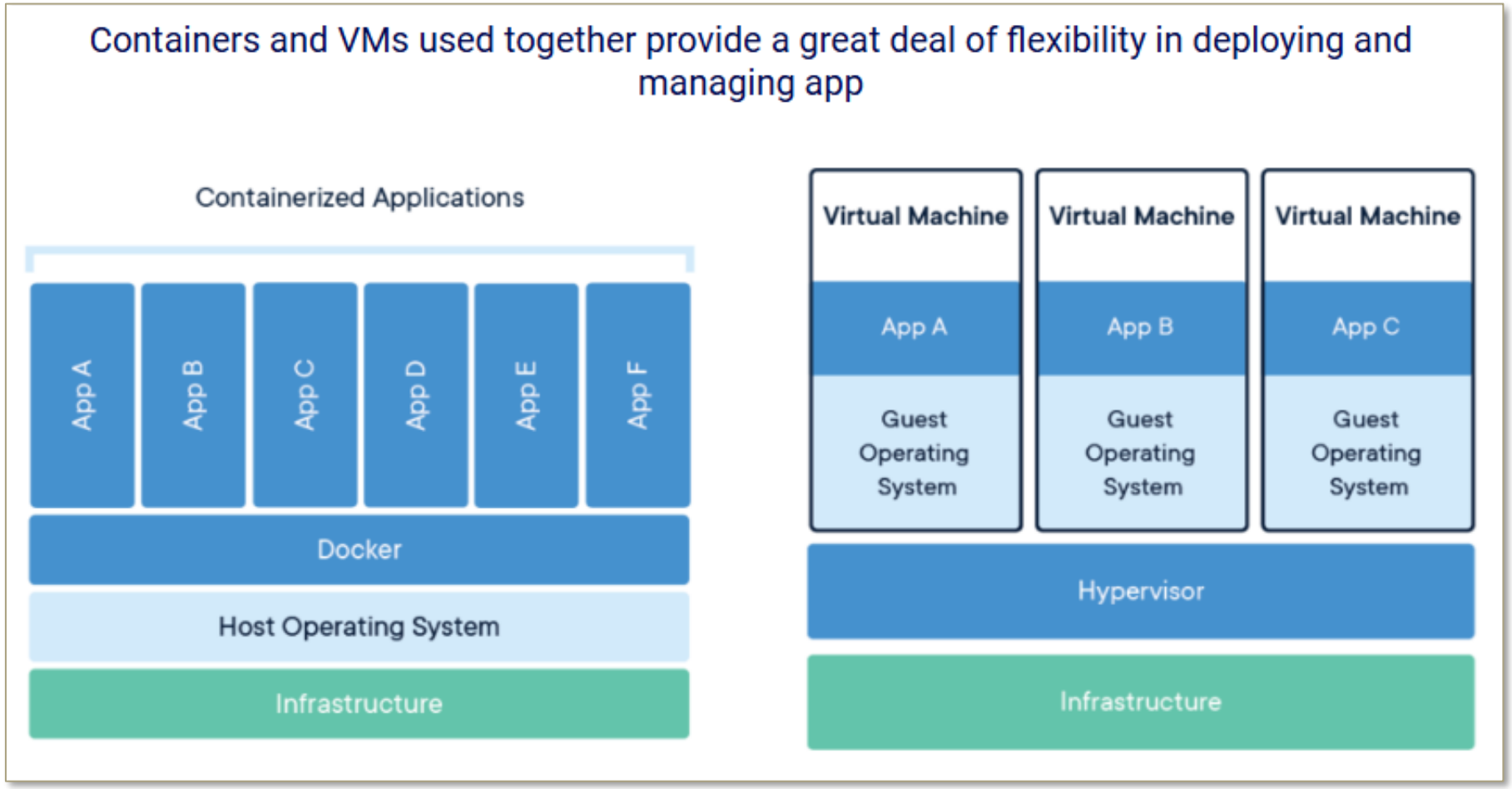


Image from: https://www.docker.com/resources/what-container/

SHIFTKEY ACADEMY

Containers vs VMs

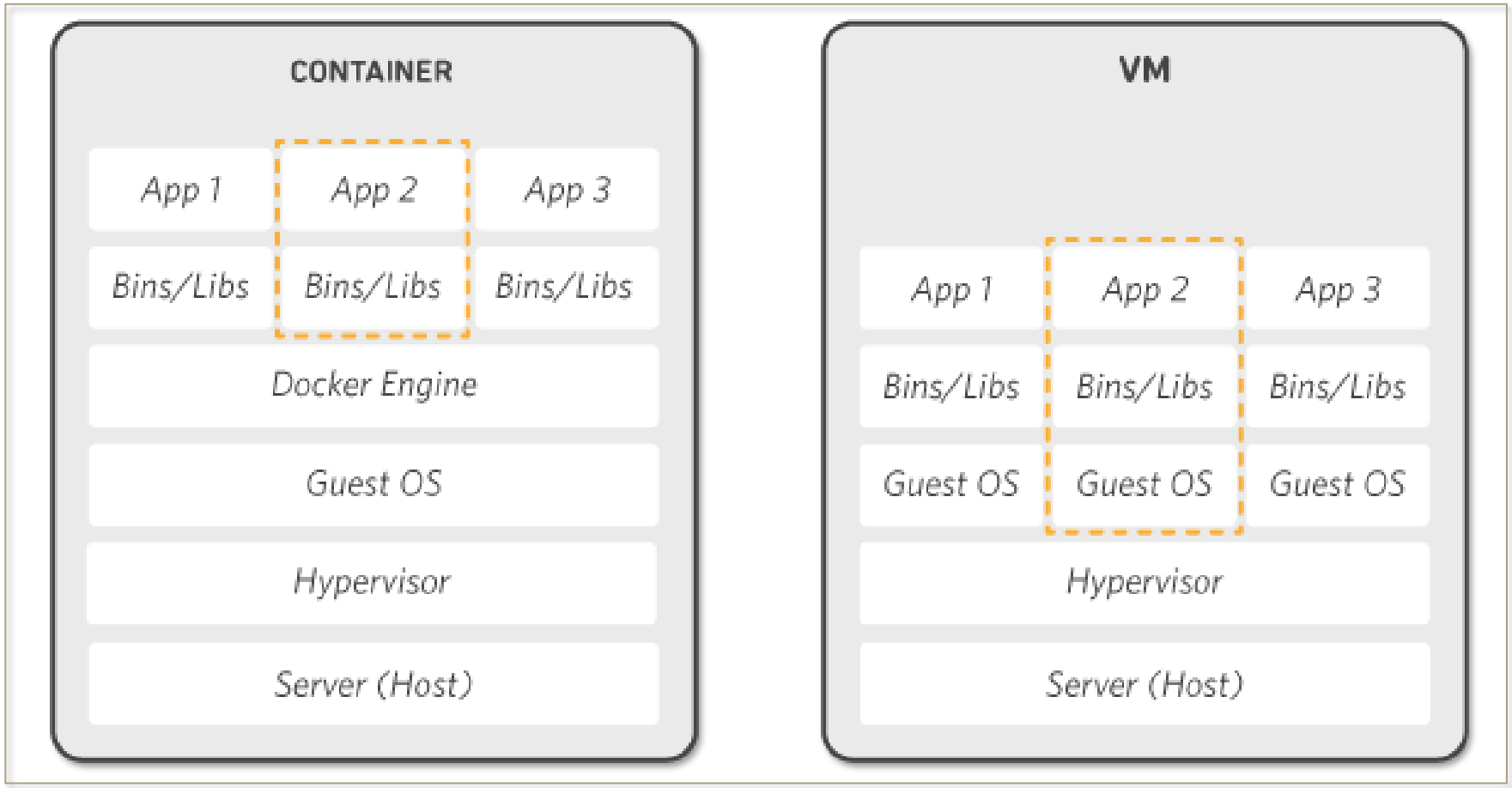


Image from: https://aws.amazon.com/docker/

What is the **aim** of containers?

Aim:

- Build
- Deploy
- Run/Execute
  any Application anywhere (without dependency on system config)

Helpful Resources:

- *https://aws.amazon.com/compare/the-difference-between-docker-vm/*
- *https://aws.amazon.com/docker/*
- (Lets see this:) *https://www.qa.com/resources/blog/docker-vs-virtual-machines-differences-you-should-know/*

DOCKER CONCEPTS

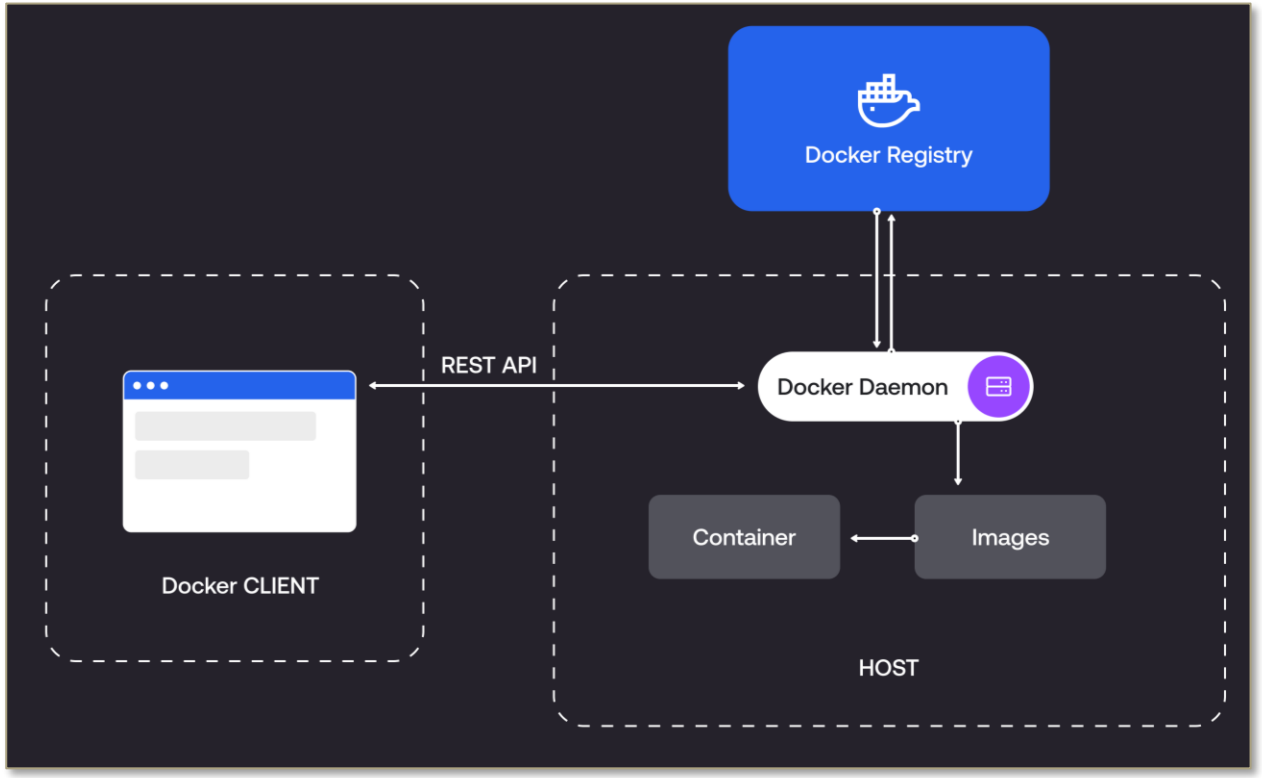
SHIFTKEY ACADEMY

What is Docker?

An open-source platform for developing, shipping, and running applications in containers. Docker helps deliver software quickly.

Docker Terms:

1. **Image**: Representation of our entire application.
2. **Container**: A standard unit in which the application service resides and executes.
3. **Engine:** Creates, and executes docker containers which can be deployed on physical or cloud infrastructure.
4. **Host**: Runs the Docker daemon
5. **Registry**: Stores Docker images (e.g., Docker Hub)

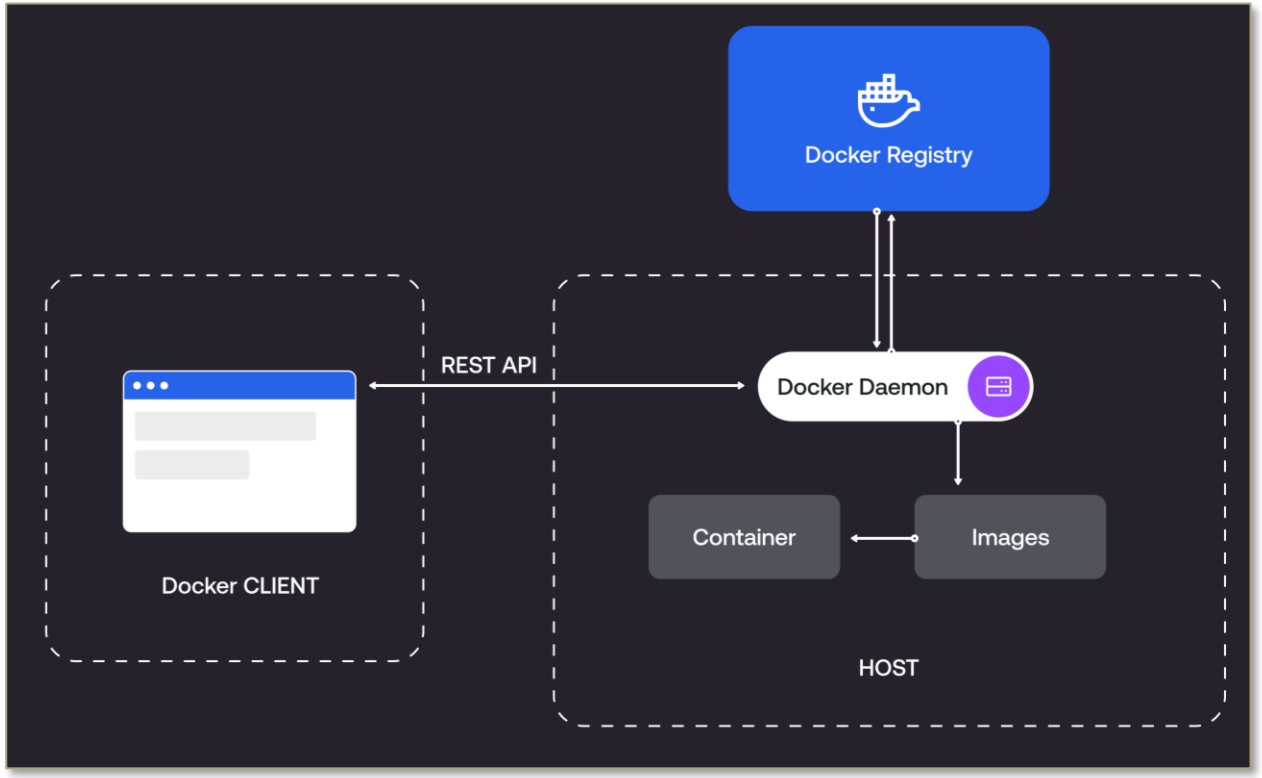


Read More: https://www.blacksmith.sh/blog/docker-architecture-the-components-and-processes-part-1

What is Docker?

Docker Terms:

1. **Client**: Command-line interface for interacting with Docker.
2. **Image**: Read-only templates used to create containers.
3. **Containers**: Runnable instances of images.
4. **Dockerfile**: Text file with instructions to build a Docker image
5. **Docker Hub**: Public registry for sharing and finding Docker images



Read More: https://www.blacksmith.sh/blog/docker-architecture-the-components-and-processes-part-1

SHIFTKEY ACADEMY

Basic Docker Commands

- **docker pull**: Download an image from a registry
- **docker run**: Create and start a container
- **docker ps**: List running containers
- **docker stop**: Stop a running container
- **docker build**: Build an image from a Dockerfile

Docker command line cheat sheet: https://docs.docker.com/get-started/docker_cheatsheet.pdf

HANDS ON

SHIFTKEY ACADEMY

MICROSERVICES ARCHITECTURE

SHIFTKEY ACADEMY

What are Microservices?

A microservices architecture is a type of application architecture where the application is developed as a collection of services. It provides the framework to develop, deploy, and maintain microservices architecture diagrams and services independently [3].

In short: Small, autonomous services that work together in contrast with monolithic architecture.

Characteristics of Microservices:

- **Independence** (Each service focuses on a specific business capability)
- **Scalability** (Services can be developed, deployed, and scaled independently)
- **Faster ship time** (Enables faster development and easier maintenance)
- **Diverse niche technology adoption** (Facilitates use of different technologies for different services)
- **Decentralized data management**

Read More: [3] https://cloud.google.com/learn/what-is-microservices-architecture

ANY QUESTIONS?

THANK YOU!

SHIFTKEY ACADEMY