Vocational English II (Mesleki Yabancı Dil II) Week 6

Engineering Faculty

Computeer Engineering

Prepared by: Dr Ercan Ezin

QUIZ TIME!

INTRODUCTION

THIS WEEK WE WILL WORK ON

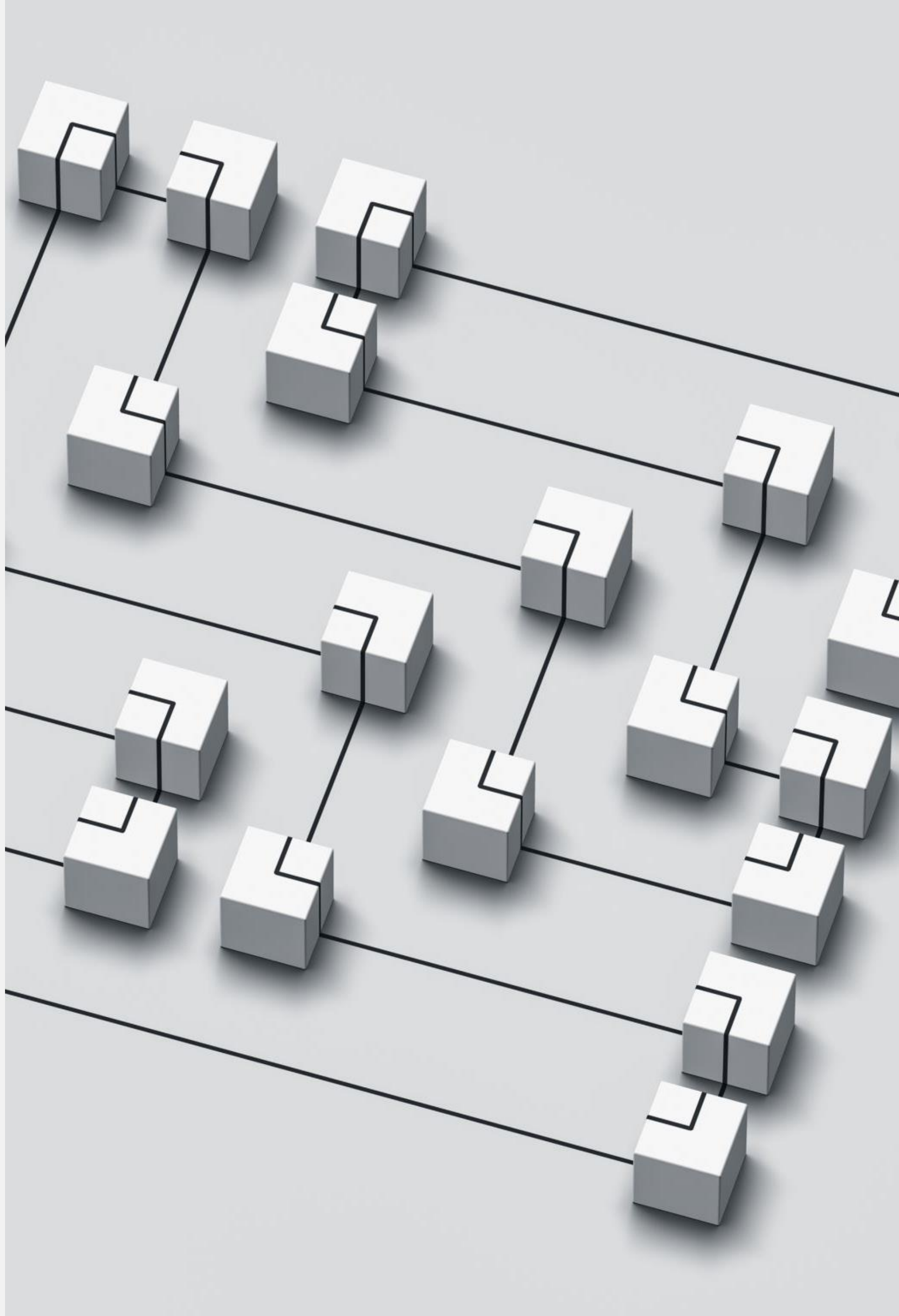
Operating Systems

BLOG: UNDERSTANDING DOCKER

https://www.techtarget.com/searchitoperations/definition/Docker

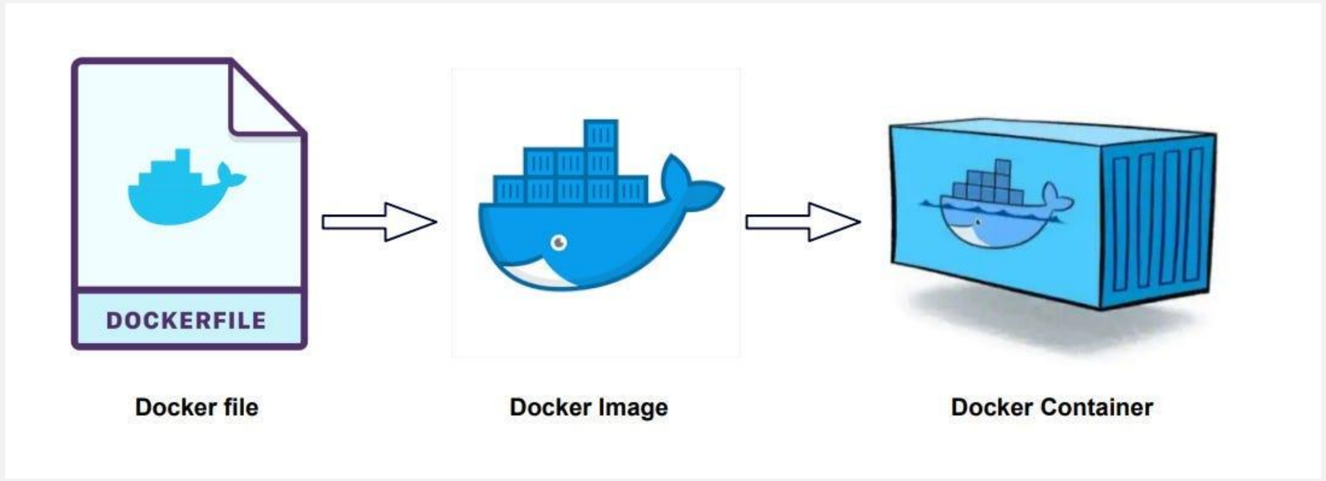
WHAT IS DOCKER?

- Docker is an open source software platform used to create, deploy and manage virtualized application containers on a common operating system (OS), with an ecosystem of allied tools. Docker gives software developers a faster and more efficient way to build and test containerized portions of an overall software application.



HOW DOCKER WORKS

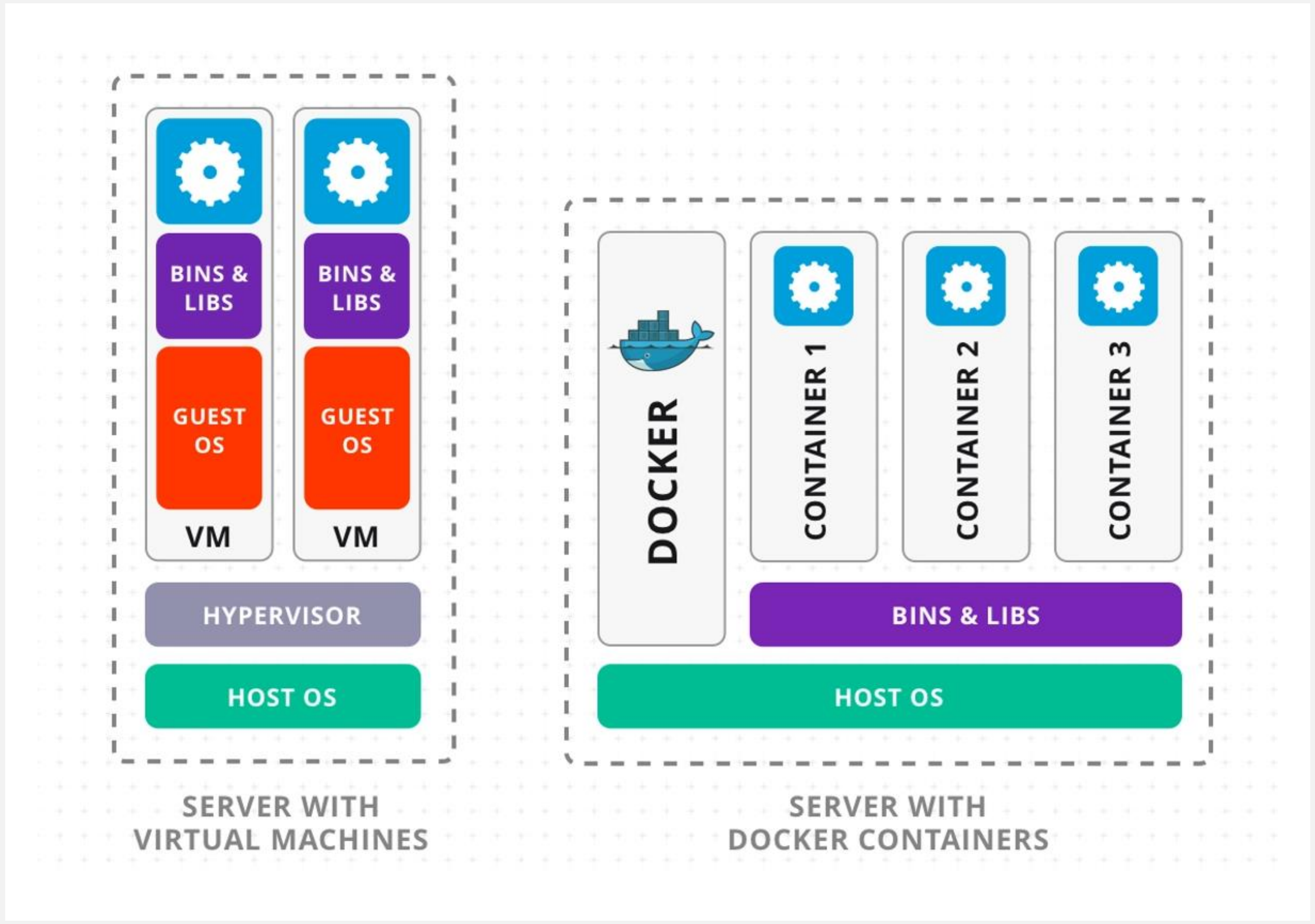
- Docker packages, provisions and runs containers. A container packages the application service or function with all of the libraries, configuration files, dependencies and other necessary parts and parameters to operate.
- Docker images contain all the dependencies needed to execute code inside a container.



Instructions and Commands

Snapshot of the computer program

Lightweight, stand-alone executable package

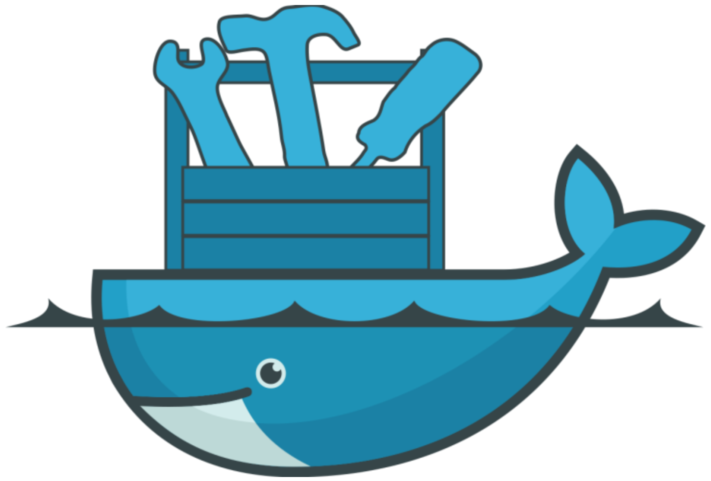


DOCKER VS. VIRTUAL MACHINES

- Docker uses resource isolation in the OS kernel to run multiple containers on the same OS.
- This is different than virtual machines (VMs), which encapsulate an entire OS with executable code on top of an abstracted layer of physical hardware resources.

KEY USE CASES FOR DOCKER

- Continuously deploying software
- Building a microservice-based architecture
- Migrating legacy applications to a containerized infrastructure
- Enabling hybrid cloud and multi-cloud applications



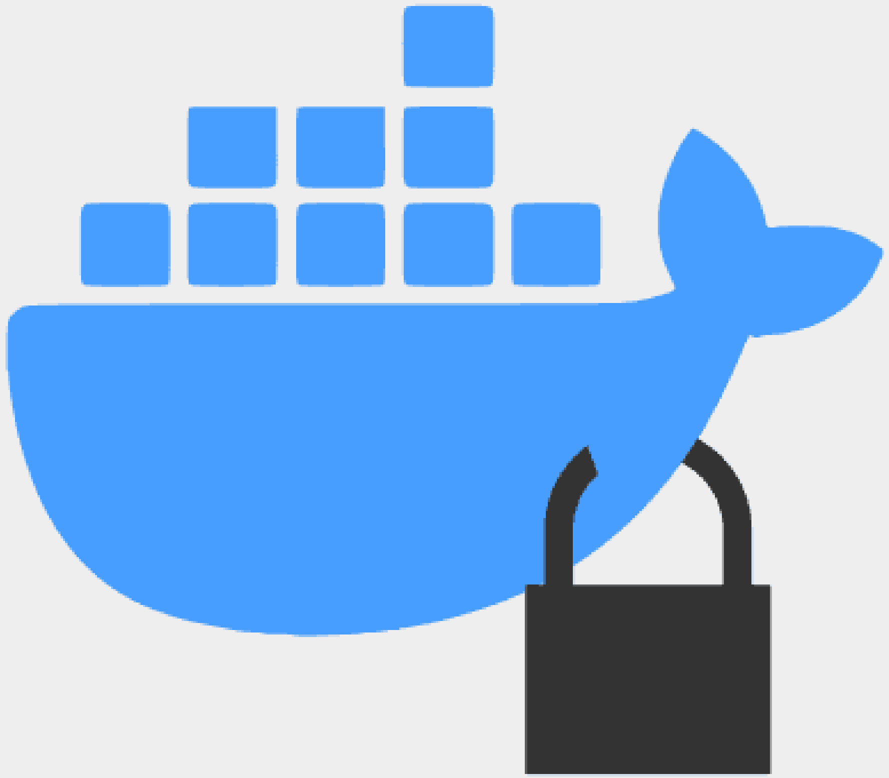
DOCKER ARCHITECTURE: COMPONENTS AND TOOLS

- Docker consists of various components and tools that help create, verify and manage containers.
- The Docker Engine is the underlying technology that handles the tasks and workflows involved in building container-based applications.
- Other components: Docker Hub, Trusted Registry, Docker Swarm, Universal Control Plane, Compose, Content Trust.

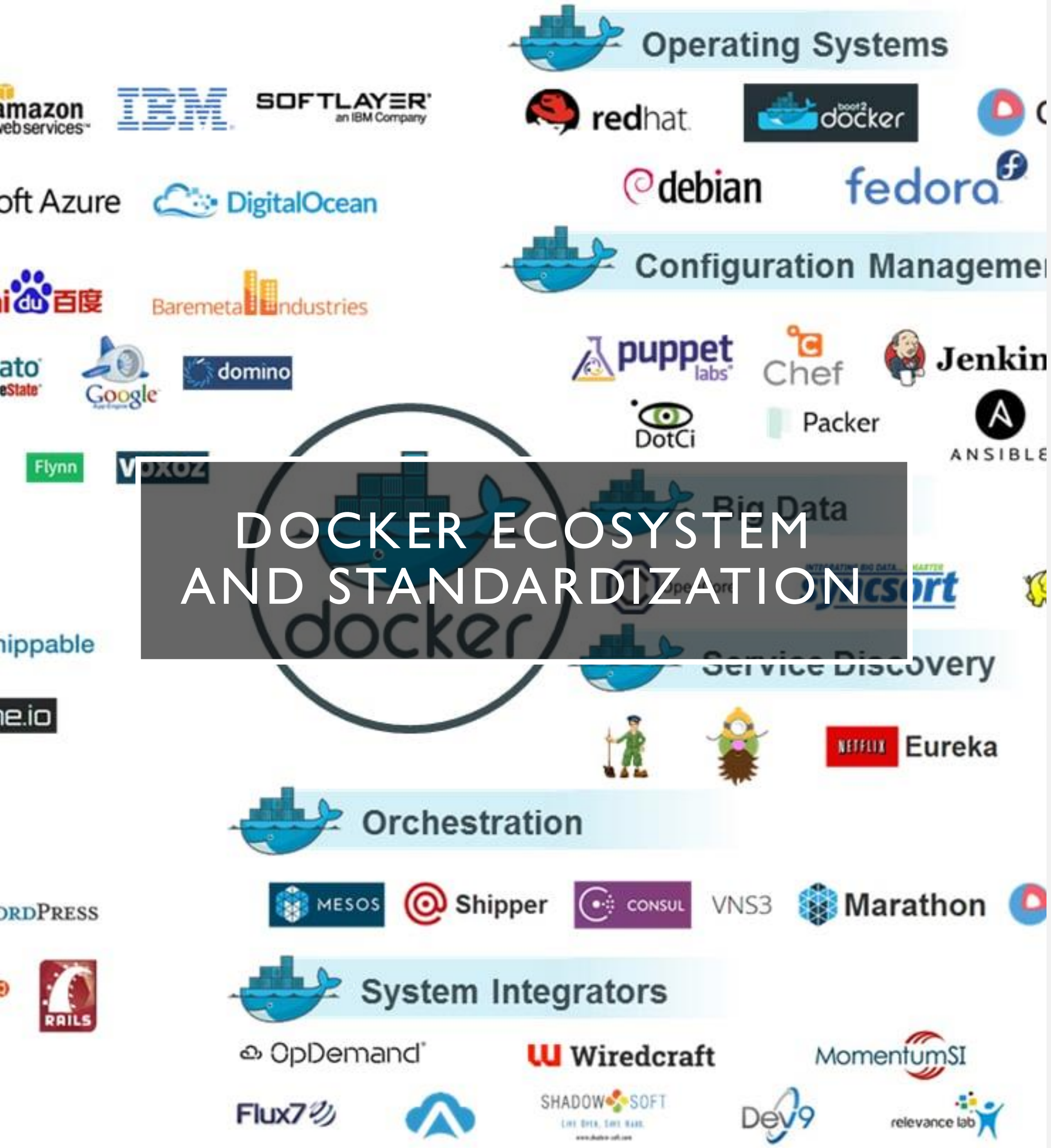
DOCKER ENTERPRISE VERSIONS

- Docker Enterprise 3.0, released in 2019, added blue-green container cluster upgrades and the ability to build multiservice container-based applications run from any environment.
- Docker Desktop Enterprise, Docker Kubernetes Service, and Docker Enterprise as a Service were also introduced.

DOCKER SECURITY



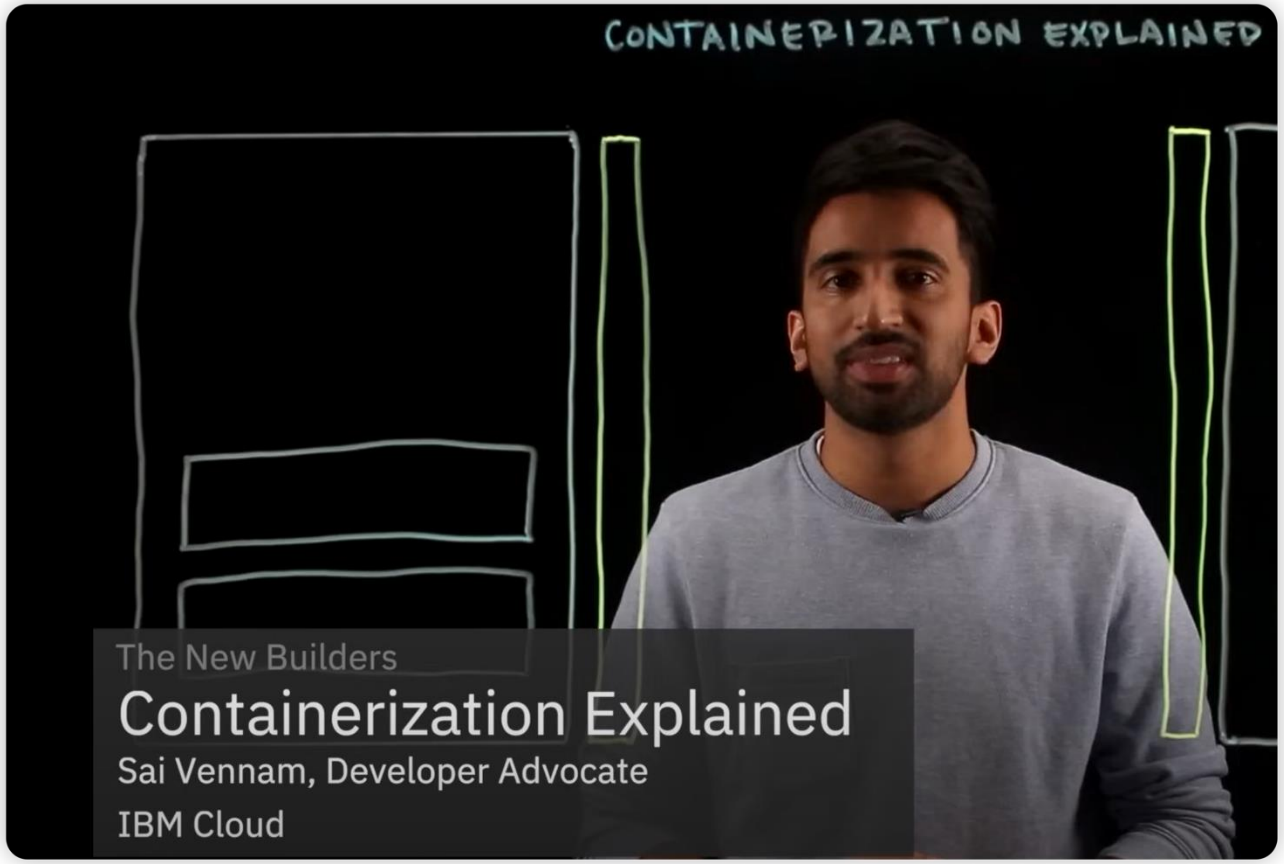
- A historically persistent issue with containers -- and Docker, by extension -- is security.
- Vulnerabilities can involve access and authorization, container images and network traffic among containers.
- Docker has regularly added security enhancements such as image scanning, secure node introduction, cryptographic node identity, and secure secret distribution.



DOCKER ECOSYSTEM AND STANDARDIZATION

- Docker also played a leading role in an initiative to more formally standardize container packaging and distribution named the **Open Container Initiative**.
- More than 40 container industry providers are members of the Open Container Initiative, including AWS, Intel and Red Hat.

LISTENING ACTIVITY


CONTAINERIZATION EXPLAINED
The New Builders
Containerization Explained
Sai Vennam, Developer Advocate
IBM Cloud

https://www.youtube.com/watch?v=0qotVMX-J5s

QUIZ TIME!

10 Questions

WORDS OF THE WEEK

1. **Container** – A lightweight, standalone executable package that includes everything needed to run a piece of software: code, runtime, system tools, libraries, and settings.
2. **Operating System (OS)** – The system software that manages hardware and software resources and provides services for computer programs.
3. **Virtual Machine (VM)** – A software emulation of a physical computer that runs an operating system and applications in an isolated environment.
4. **Microservice** – An architectural style where applications are structured as small, independent services that communicate over APIs.
5. **Container Orchestration** – The automated arrangement, coordination, and management of multiple software containers, often across clusters of machines.
6. **Docker Engine** – The core component of Docker that creates and manages containers using a server-side daemon and a client-side CLI.
7. **Docker Swarm** – Docker's native tool for clustering and scheduling containers across multiple Docker hosts, treating them as a single virtual system.
8. **Kubernetes** – An open-source platform for automating deployment, scaling, and operations of application containers across clusters of hosts.
9. **Command-Line Interface (CLI)** – A text-based interface that allows users to interact with software by typing commands.
10. **Daemon** – A background process that handles requests for services such as managing containers, images, and networking.
11. **Dependencies** – The external libraries, packages, or resources a program needs in order to execute properly.
12. **Kernel** – The core component of an operating system that manages system resources and communication between hardware and software.
13. **Images** – Immutable snapshots used to create containers, containing the application code and its environment (dependencies, settings, etc.).
14. **Volumes** – Docker-managed directories used to store persistent data outside of containers, allowing data to survive container restarts.
15. **Hybrid Cloud** – An IT architecture that connects on-premises infrastructure with public cloud services, allowing data and apps to move between environments.
16. **Multi-Cloud** – A strategy where multiple cloud services from different vendors are used to improve redundancy, performance, or flexibility.
17. **Registry** – A storage and distribution system for named Docker images, including versioning and metadata.
18. **Security** – Measures and practices for protecting containers and systems from unauthorized access, data breaches, and vulnerabilities.
19. **Node** – A single machine, physical or virtual, in a container orchestration cluster that runs containers and services.
20. **Runtime** – The period during which a program is running, or the environment in which a program executes.

PS: Keep a journal where you note these words with their meanings and usages in a sentence.

EOF*

*End of Fun/File