Welcome
Docker Certified Associate (DCA) Course

I am Md Nasir Uddin here as your DCA Course Instructure.

Md Nasir uddin
DevOps Engineer (Manager)
Anwar Enterprise System Ltd
2x Kubernetes, 2x AWS, 2x Linux, NSE4, CCNA

Docker History

Docker Inc:

Solomon Hykes created Docker in 2013 at dotCloud, a cloud hosting company.

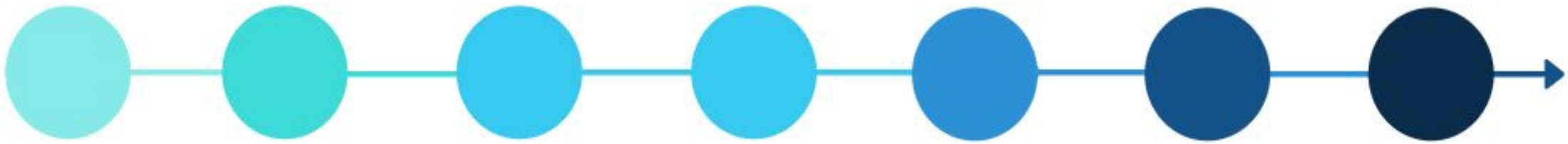
Initially using LXC as its default execution environment & Docker quickly gained popularity.

In March 2014, Docker open-sourced its technology, becoming a standout project on GitHub.

Docker is written in the Go programming language.

EVOLUTION OF CONTAINER TECHNOLOGY

Past, Present and Future



1970

Beginning of process isolation with the introduction of Unix v7 in 1970. It allowed for the segregation of file access for each process.

2004-2005

Solaris Containers was released in 2004 that combined system resource controls and boundary separation.

In 2005, Open Vz was released which provided operating system-level virtualization technology for Linux.

2006-2008

Google launched process Containers in 2006. It was designed for limiting, accounting, and isolating resource usage.

LXC (LinuX Containers, 2008) was the first and most complete implementation of Linux container manager.

2013-2014

Docker engine was first released in 2013.

Kubernetes was announced by Google in 2014.

2016-2017

Kubernetes was adopted by Cloud Native Computing Foundation (CNCF) in 2016

In 2017, all major players adopted/ supported Kubernetes

Security vulnerabilities revealed CVE-2016-5195

PRESENT

Kubernetes is now the gold standard. There has been an explosion of companies catering to the management of containers, CI/CD, and DevSecOps.

Growth of hybrid and multi-cloud environments

FUTURE

Hyper abstraction

Adoption of Serverless technologies

Focus on Security

Containers at the intelligent edge, and for mission-critical applications

Increase in complexity and skill shortage

Docker Architecture

What is Docker?

Docker is a platform for developing, packaging & running applications in containers, providing a consistent and portable environment across different systems

It uses operating-system-level virtualization to deliver software in packages called containers

What is Docker daemon?

Docker daemon runs on the host operating system. It is responsible for running containers to manage docker services.

Docker Architecture

Docker architecture

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.

Docker Client: The user interface for interacting with Docker through commands or an API.

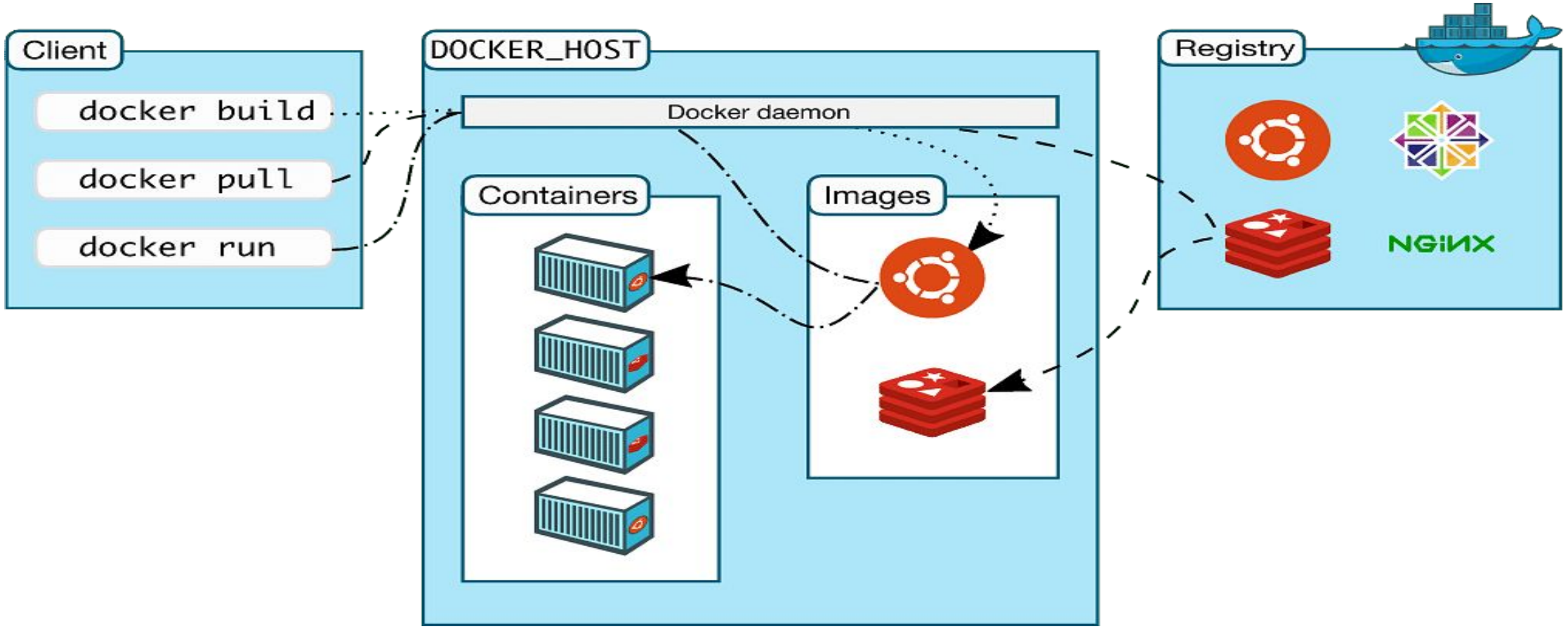
Docker Daemon (Host): A background process managing containers on the host machine.

Docker Registry: Repositories storing and distributing Docker images, like Docker Hub.

Docker Architecture

Docker architecture

Docker follows Client-Server architecture, which includes the three main components that are Docker Client, Docker Host, and Docker Registry.



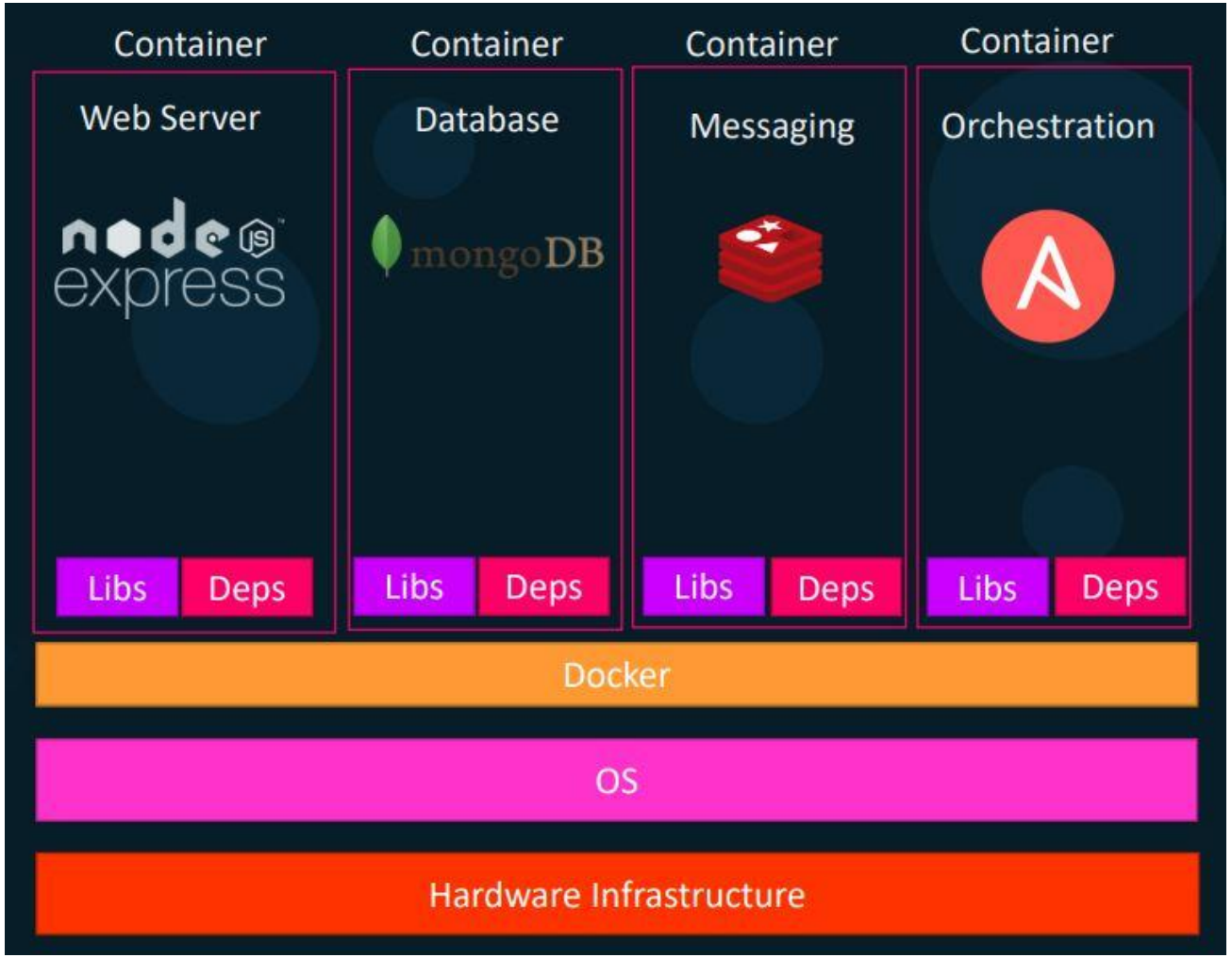
Why do you need docker & What can it do?

Dockerization offers several advantages over traditional approaches such as using virtual machines (VMs) or bare-metal servers.

1. Containerize Applications
2. Run each service with its own dependencies in separate containers
3. Consistent & Isolated Environment
4. Rapid Application Deployment
5. Ensures Scalability & Flexibility
6. DevOps and Continuous Integration/Continuous Deployment (CI/CD)
7. Better Portability
8. Cost-Effective
9. In-Built Version Control System
10. Resource Utilization:
11. Security

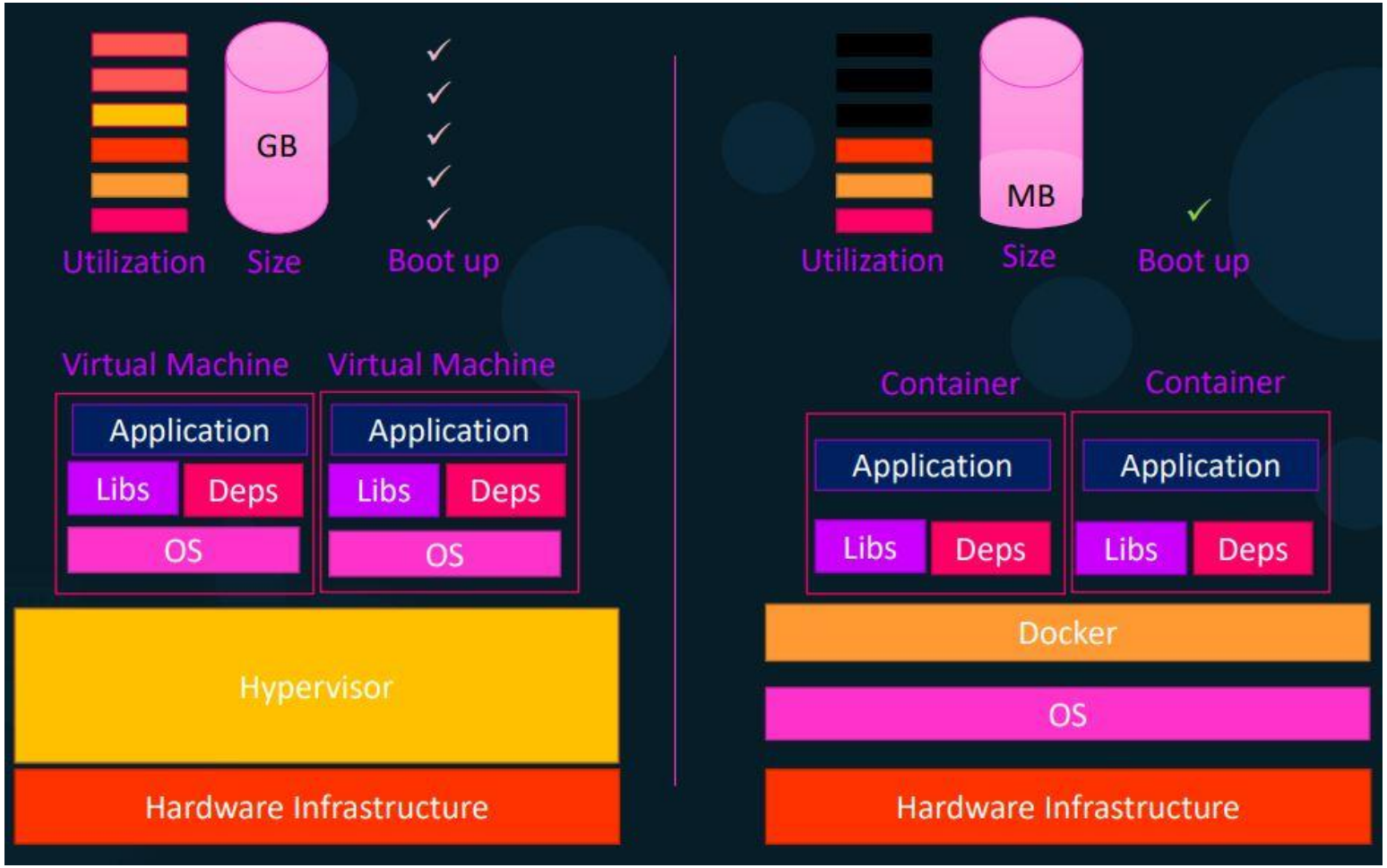
What are containers?

A container is a unit of software that packages code and its dependencies so the application runs quickly and reliably across computing environments.



Containers vs Virtual Machines

Virtual machines is that virtual machines virtualize an entire machine down to the hardware layers and containers only virtualize software layers above the operating system level.



How is it done?

docker run nginx

What is a Docker Registry?

The registry allows Docker users to pull images locally, as well as push new images to the registry (given adequate access permissions when applicable).

DockerHub

DockerHub is a hosted registry solution by Docker Inc. Besides public and private repositories, it also provides automated builds, organization accounts, and integration with source control solutions like Github and Bitbucket .

Other Public Registries
Amazon Elastic Container Registry (ECR)
Google Container Registry (GCR)
Azure Container Registry (ACR)

Overview of Docker editions!

Docker Engine - Community

Docker Engine - Enterprise

Docker Enterprise

Docker Engine Community: Ideal for individual developers and small teams looking to get started with Docker and experimenting with container-based apps.

Docker Engine Enterprise: Designed for enterprise development of a container runtime with security and an enterprise grade SLA in mind.

Docker Enterprise: Designed for enterprise development and IT teams who build, ship, and run business critical applications in production at scale.

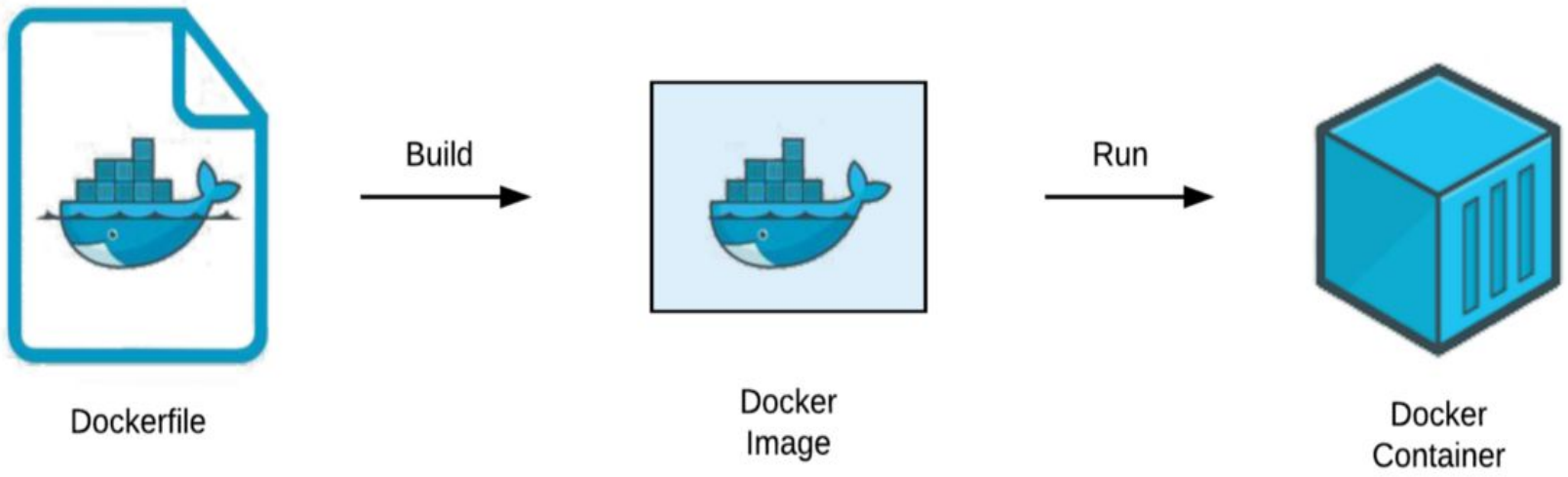
Ref: https://docs.docker.com.xy2401.com/install/overview/

Dockerfile, Image & Container

Dockerfile: Contains a set of Docker instructions that provisions your operating system the way you like, and installs/configure all your software.

Image: Compiled Dockerfile. Saves you time from rebuilding the Dockerfile every time you need to run a container. And it's a way to hide your provision code.

Containers: Containers offer a lightweight and efficient way to package and run applications, it's provide a consistent and isolated environment.



Install Docker Engine on Ubuntu

Prerequisites, OS requirements

To install Docker Engine, you need the 64-bit version of one of these Ubuntu versions:

- ✔ Ubuntu Jammy 22.04 (LTS)
- ✔ Ubuntu Impish 21.10
- ✔ Ubuntu Focal 20.04 (LTS)
- ✔ Ubuntu Bionic 18.04 (LTS)

Install Docker Engine from Here

Let’s try some basic command Here