
Metamorphosis
2K23

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

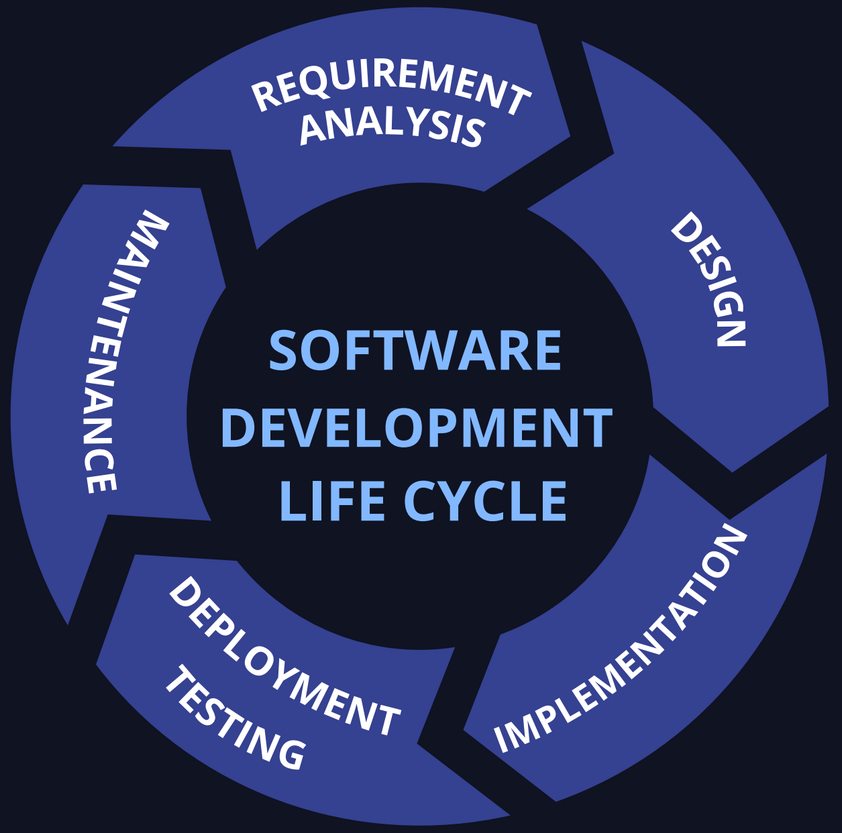
SOFTWARE DEVELOPMENT PROCESS

- What is the software development process?
- What is the need of software development process?
- Is it only related to coding ?

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

SOFTWARE DEVELOPMENT PROCESS CYCLE

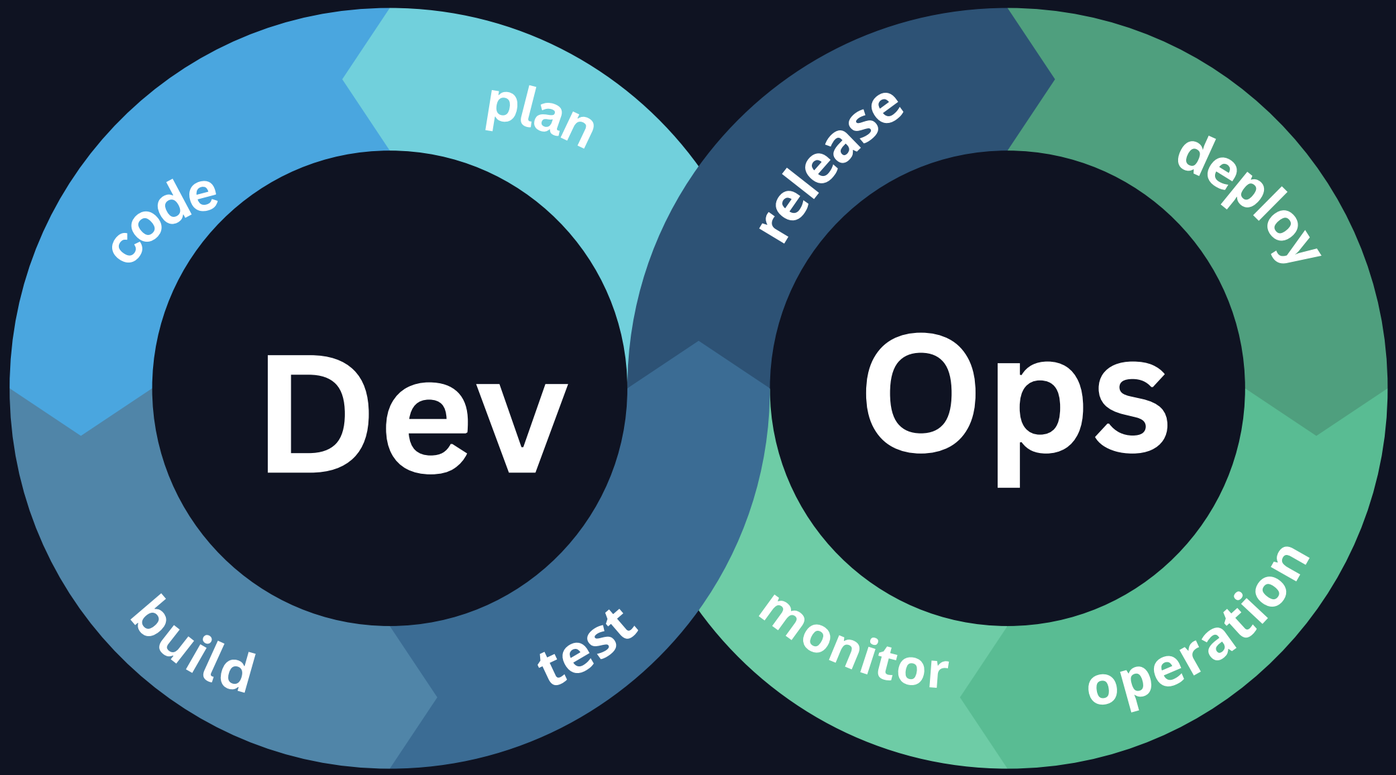


META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

EVOLUTION IN SOFTWARE DEVELOPEMENT

Development and Operations



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

SOFTWARE DEVELOPMENT PROCESS

Docker overcomes the issues in DevOps

TESTING

DEPLOYMENT

MAINTENANCE

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHAT IS VIRTUALIZATION?

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

VIRTUALIZATION

In computing, virtualization refers to the act of creating a virtual(rather than actual) version of something, this includes virtual computer hardware, virtual storage devices and virtual network resources.

META

VIRTUALIZATION

WLUG
COMMUNITY | KNOWLEDGE | SHARE

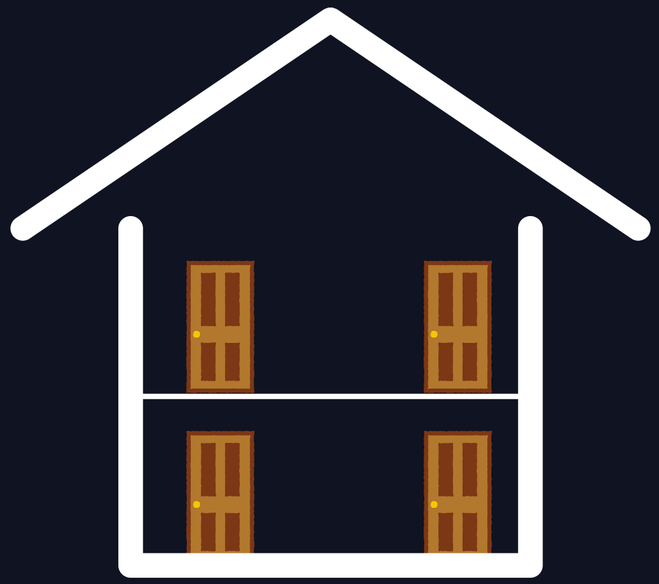
Let's consider an example

META

VIRTUALIZATION

WLUG
COMMUNITY | KNOWLEDGE | SHARE

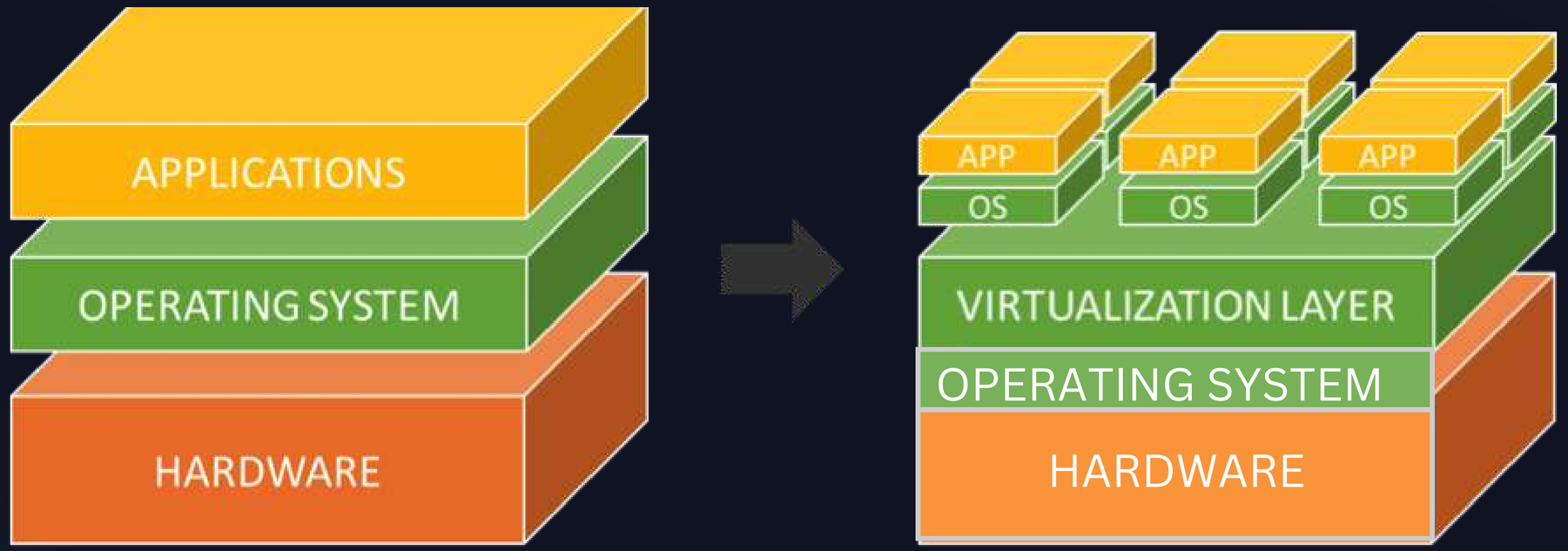
Let's consider an example



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Physical Server Vs Virtualization

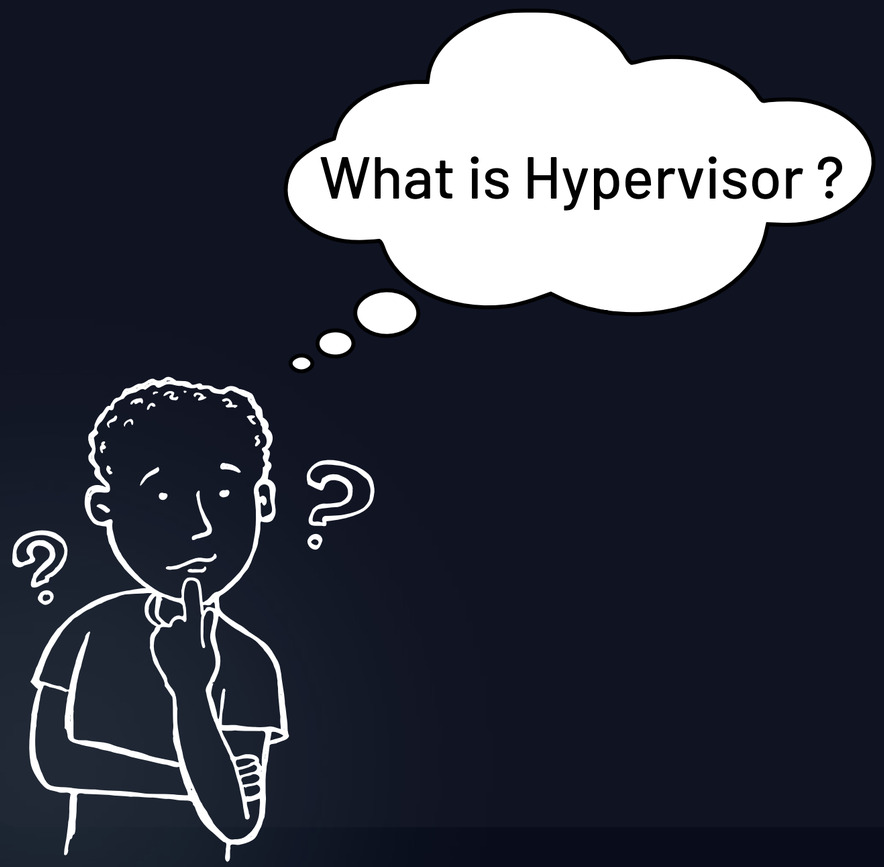


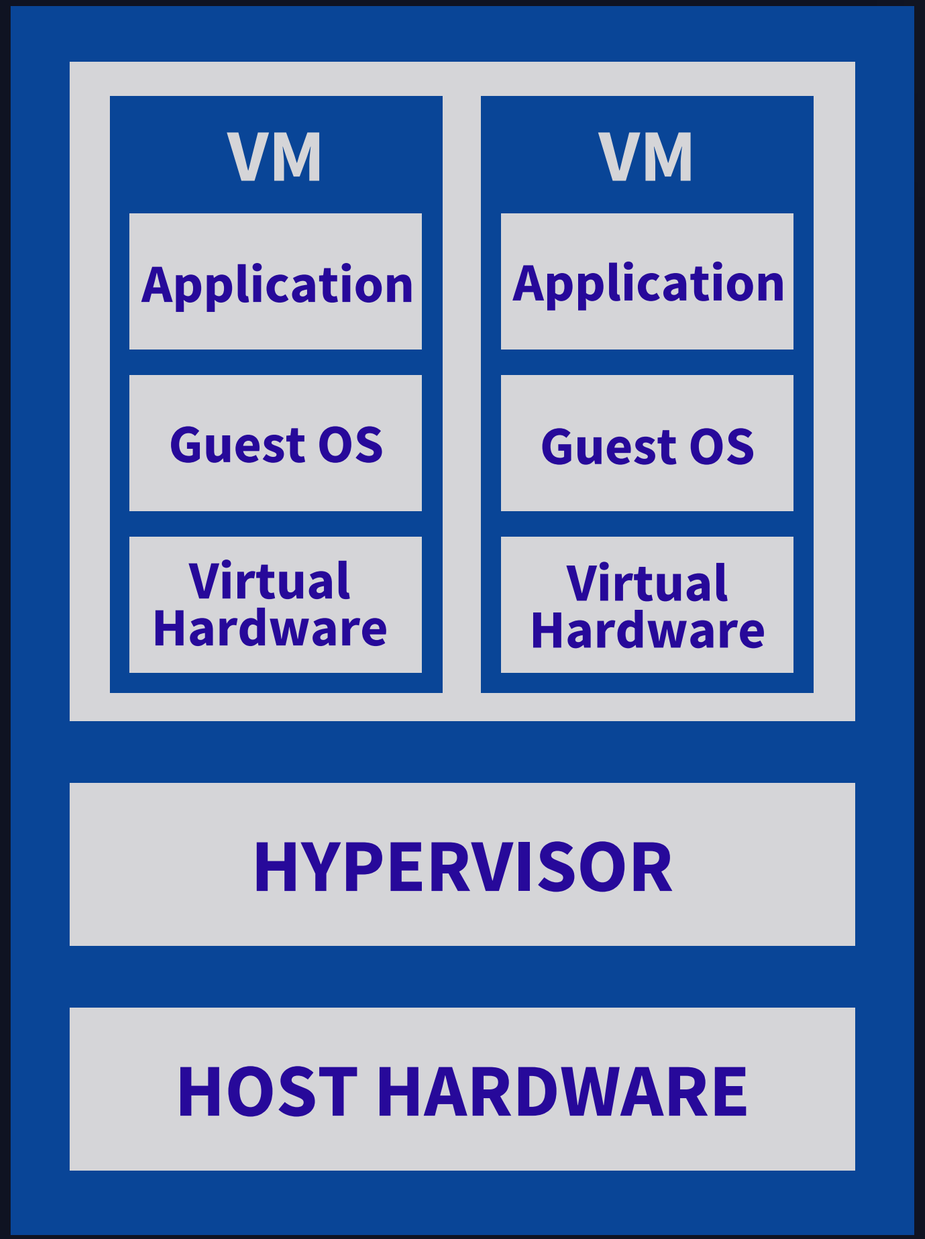
TRADITIONAL ARCHITECTURE

VIRTUAL ARCHITECTURE

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE





META

Hypervisor

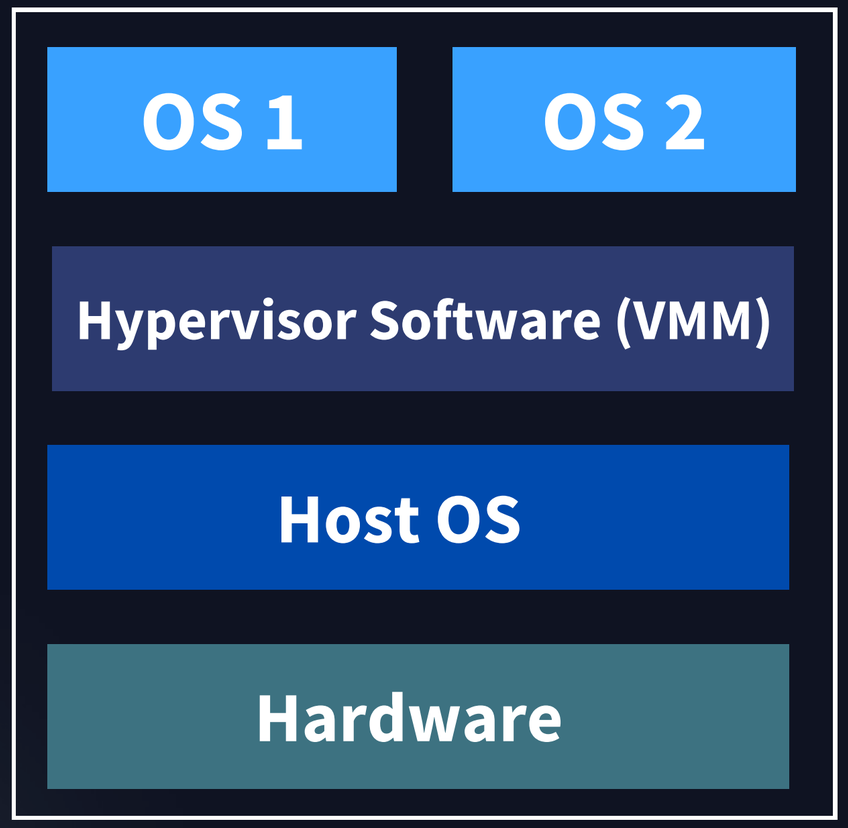
WLUG
COMMUNITY | KNOWLEDGE | SHARE

- A hypervisor is also called a virtual machine manager(VMM).
- It is a software/firmware that creates and runs virtual machines.
- It allows one host computer to support multiple guest VMs by virtually sharing its resources.

META

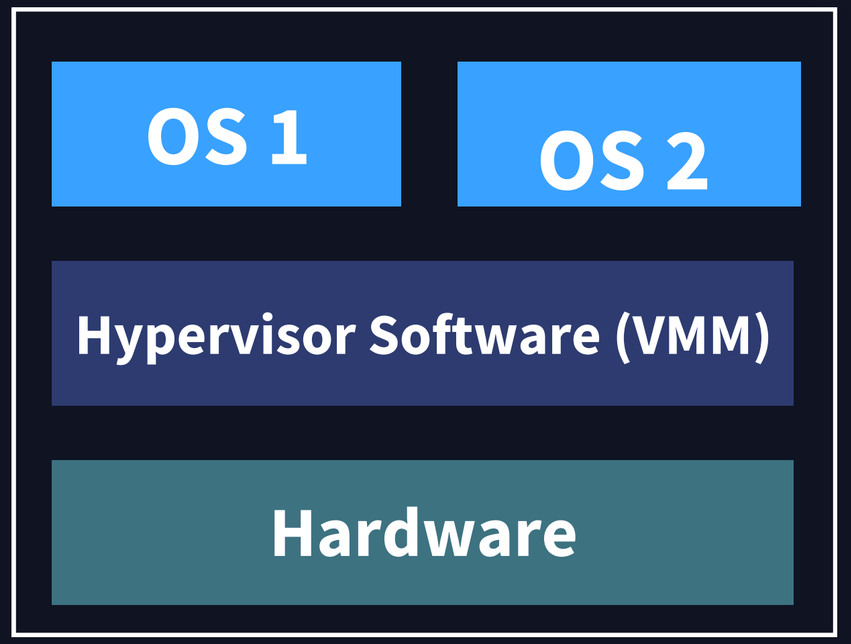
Types of Hypervisor

WLUG
COMMUNITY | KNOWLEDGE | SHARE



Hosted Architecture

Ex. Oracle VM VirtualBox



Bare-Metal Architecture

Ex. VMware ESXi.

META

Types of Hypervisor

WLUG
COMMUNITY | KNOWLEDGE | SHARE

I) Type-1/Bare metal hypervisor

- Runs directly on the host’s hardware to manage guest operating systems
- Very efficient because they have direct access to the physical hardware resources

II) Type-2/Hosted Hypervisor

- Software is installed on an operating system
- Hypervisor asks the operating system to make hardware calls

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Comparisons

Physical Server

- Resources are not shared between multiple users

Virtualization

- Resources are shared between multiple users

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Comparisons

Physical Server

- Includes memory, hard- drive, processor, network connection and OS

Virtualization

- It is software based that emulates all the functions of a physical server

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Comparisons

Physical Server

- Hardware is used directly by an OS

Virtualization

- A hypervisor manages the virtualized resources

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Comparisons

Physical Server

- It is used to run a single instance of an OS

Virtualization

- Runs an independent OS on top of the hypervisor

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Drawbacks of Virtualization

- It involves upfront costs.
- Working on shared hardware resources hosted on a third-party resource.
- To ensure the availability of software, security, resource can be a tedious task.
- As the complexity of tasks increases, substantially higher time is required.

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHAT IS CONTAINERIZATION?

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Container

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

META

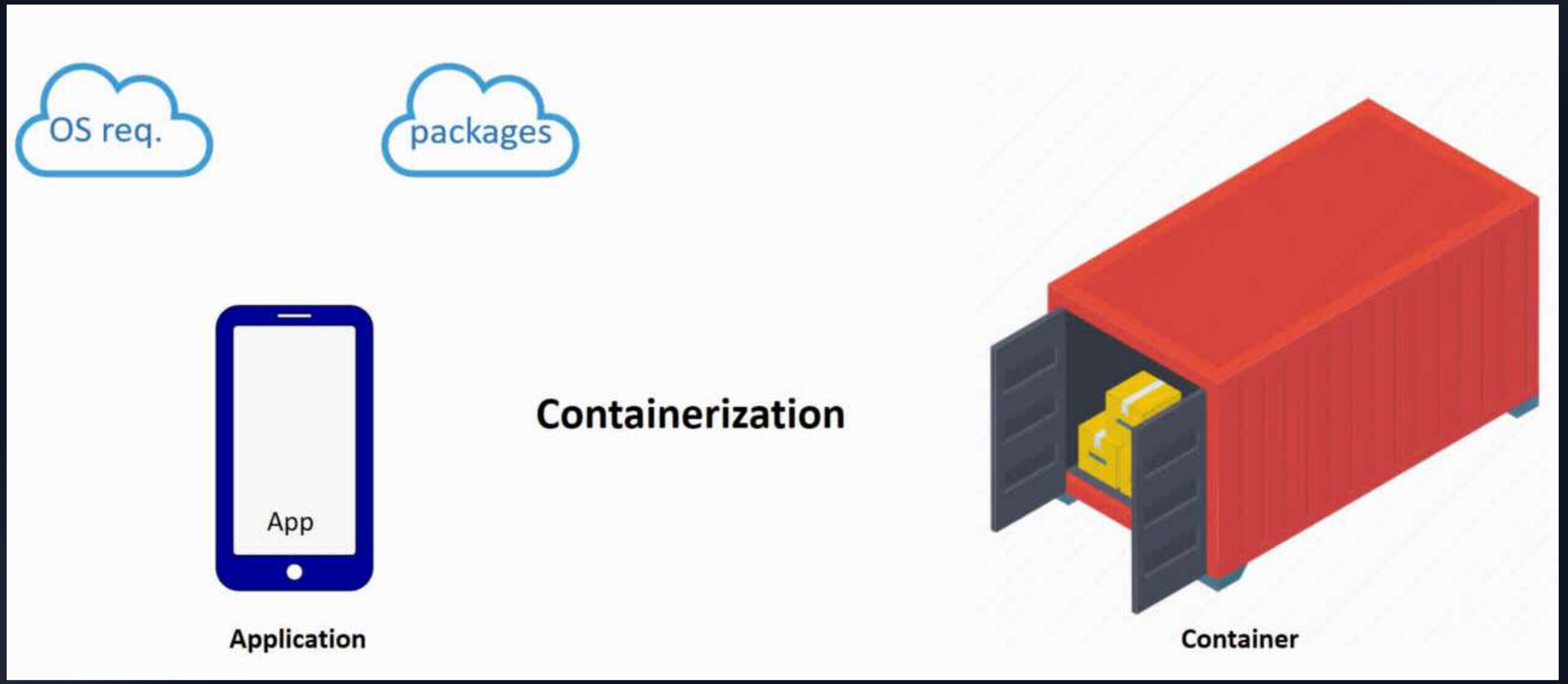
Containerization

- Containerization is a process that bundles an application code with all the files and libraries it needs to run on any infrastructure.
- It is Operating System level Virtualization.
- Containers are said to share the host system’s kernel with other containers.

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Containerization



META

Containers vs VMs

- Operating System
- Architecture
- Isolation
- Speed and Portability
- Resources and Efficiency

VS

WLUG
COMMUNITY | KNOWLEDGE | SHARE

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Operating System

Containers

- It has only essential files of the Operating System

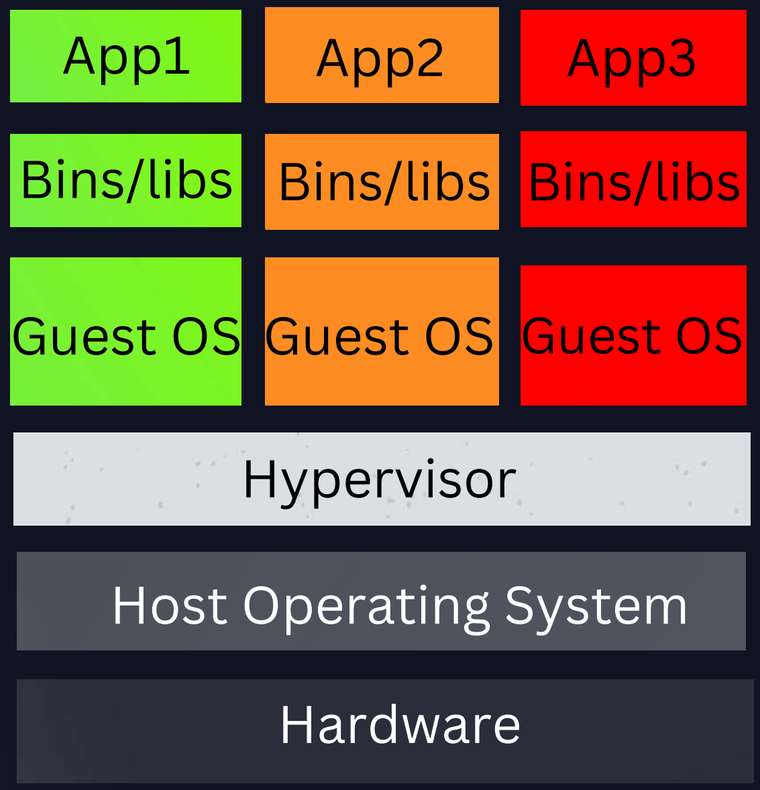
VMs

- It has a whole copy of the Operating System

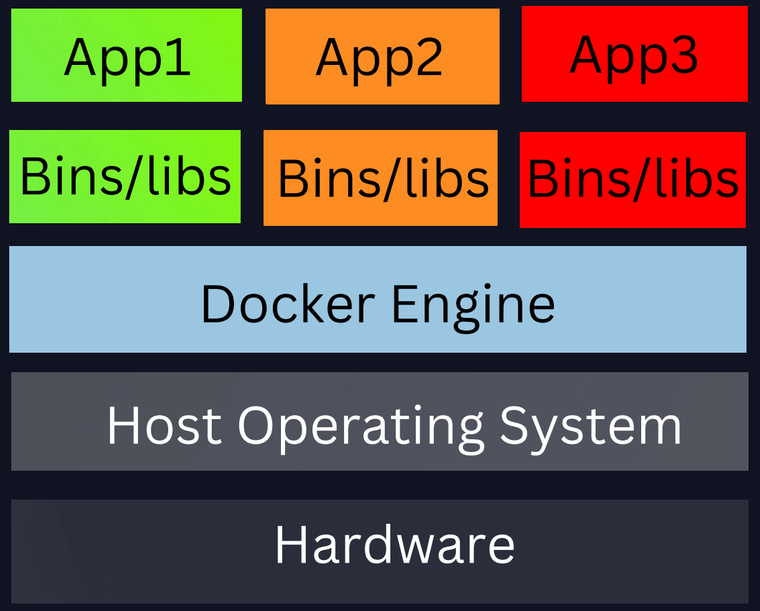
META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Architecture



Virtual Machine



Container

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Isolation

Containers

- Isolation can be provided to multiple containers on the same server, ensuring they are completely isolated from each other

VMs

- It provides complete isolation between the guest OS and the host

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Speed and Portability

Containers

- Containers are faster

VMs

- VMs being a whole copy of the host server on its operating system, VMs are resource-heavy hence slower

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Resource and Efficiency

Containers

- Containers use less memory and hence are more efficient

VMs

- VMs use more resources and hence are less efficient

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

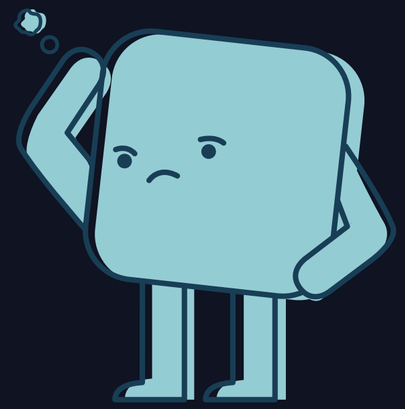
What is Docker?

- Docker is an open source platform designed to make it easier for developing, shipping, and running applications.
- Docker was developed by Kamel Founadi, Solomon Hykes, and Sebastien Pahl in 2013.
- It was written in Go programming language.

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHAT IS THE NEED OF DOCKER?



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Why Docker?

Save money

Reuse
Container

Fast

Lightweight and
more granular
update

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER INSTALLATION

META

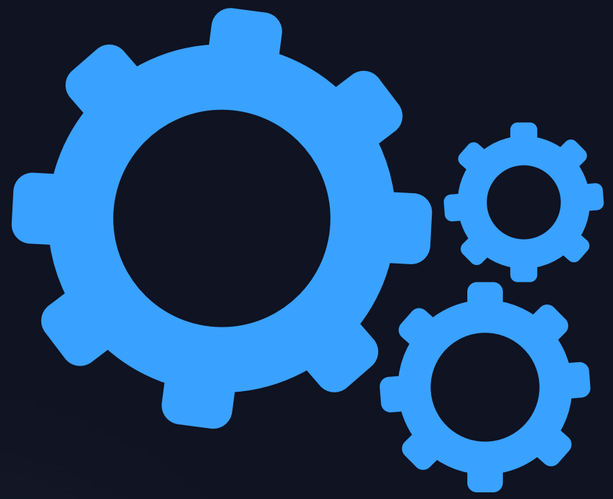
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Basic Commands

- **sudo** - allows regular users to run programs with the security privileges of the superuser or root user
- **ls** - Lists all files and directories in the present working directory
- **cd** - To change to a particular directory
- **mkdir** <directoryname> - Creates a new directory in the present working directory or at the specified path
- **touch** <filename> - Creates a new file
- **cat** <filename> - Shows the content of the given filename

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

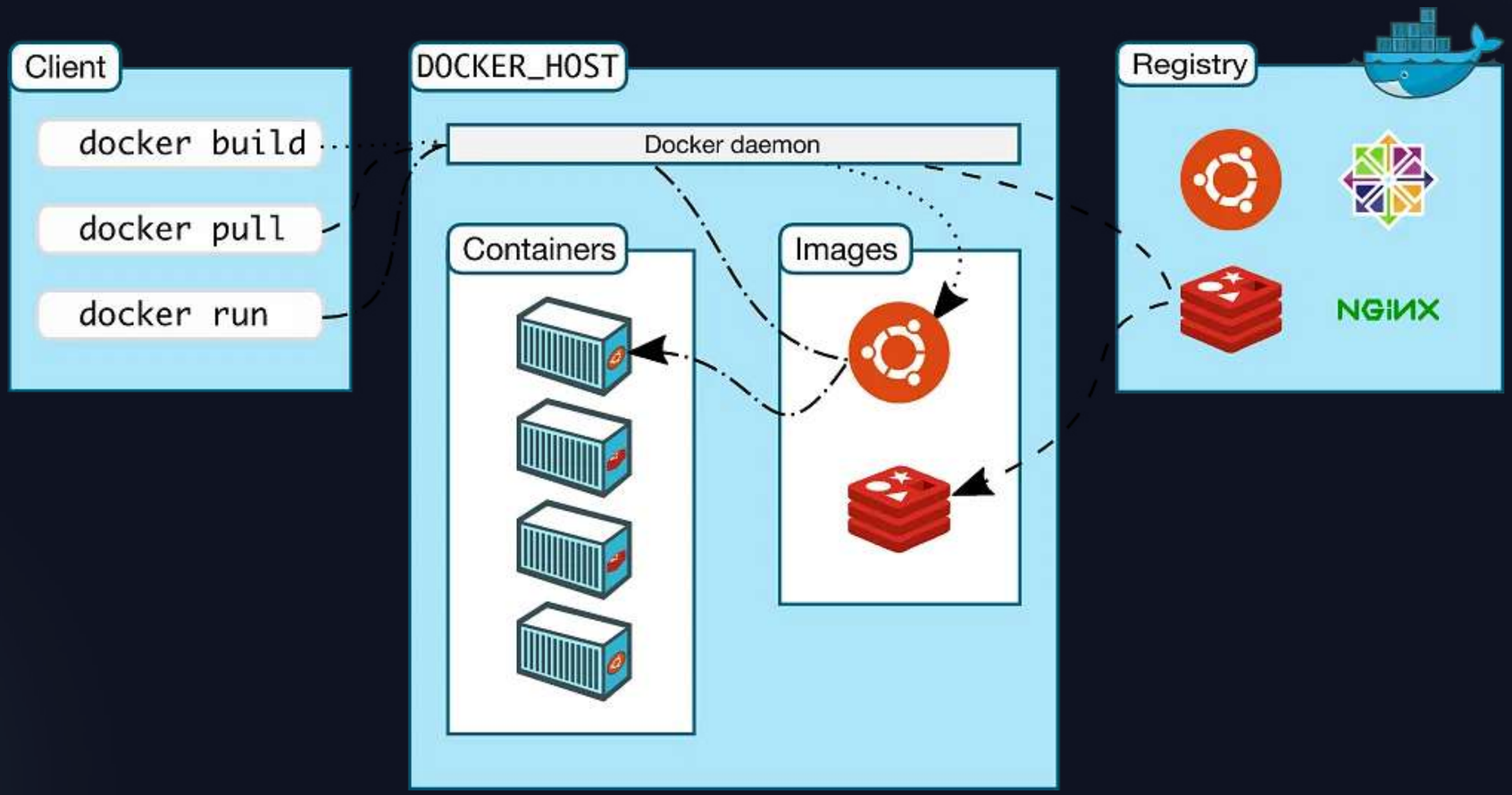


DOCKER ARCHITECTURE

META

DOCKER ARCHITECTURE

WLUG
COMMUNITY | KNOWLEDGE | SHARE



META

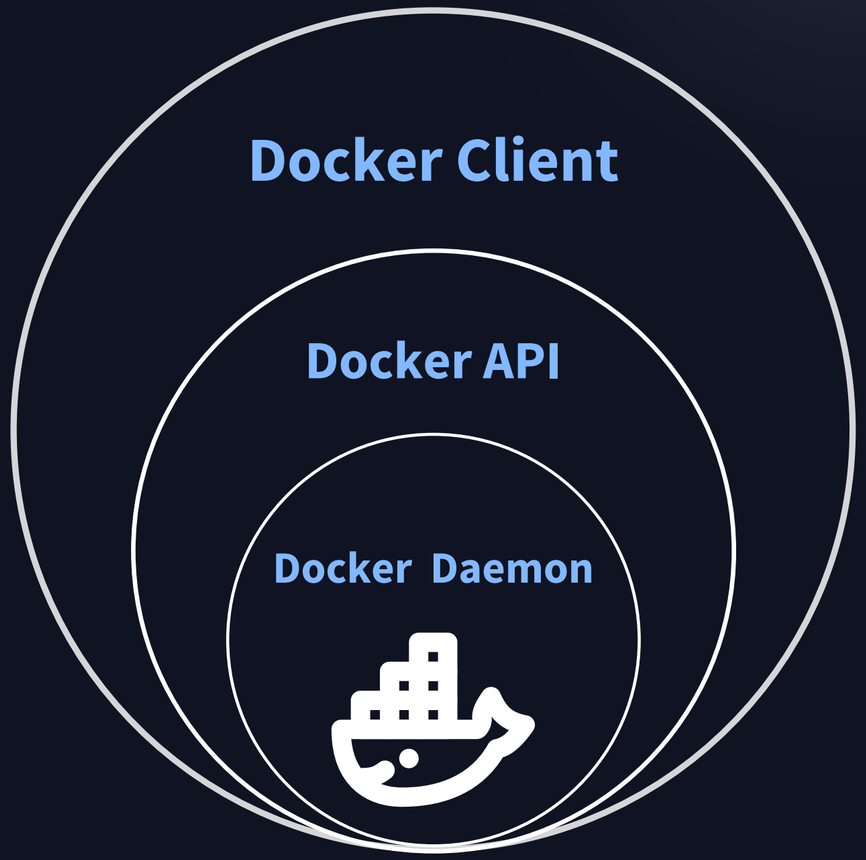
WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER COMPONENTS

META

DOCKER ENGINE

- Manages docker services
- Communicates with host OS
- Provides environment for containerization

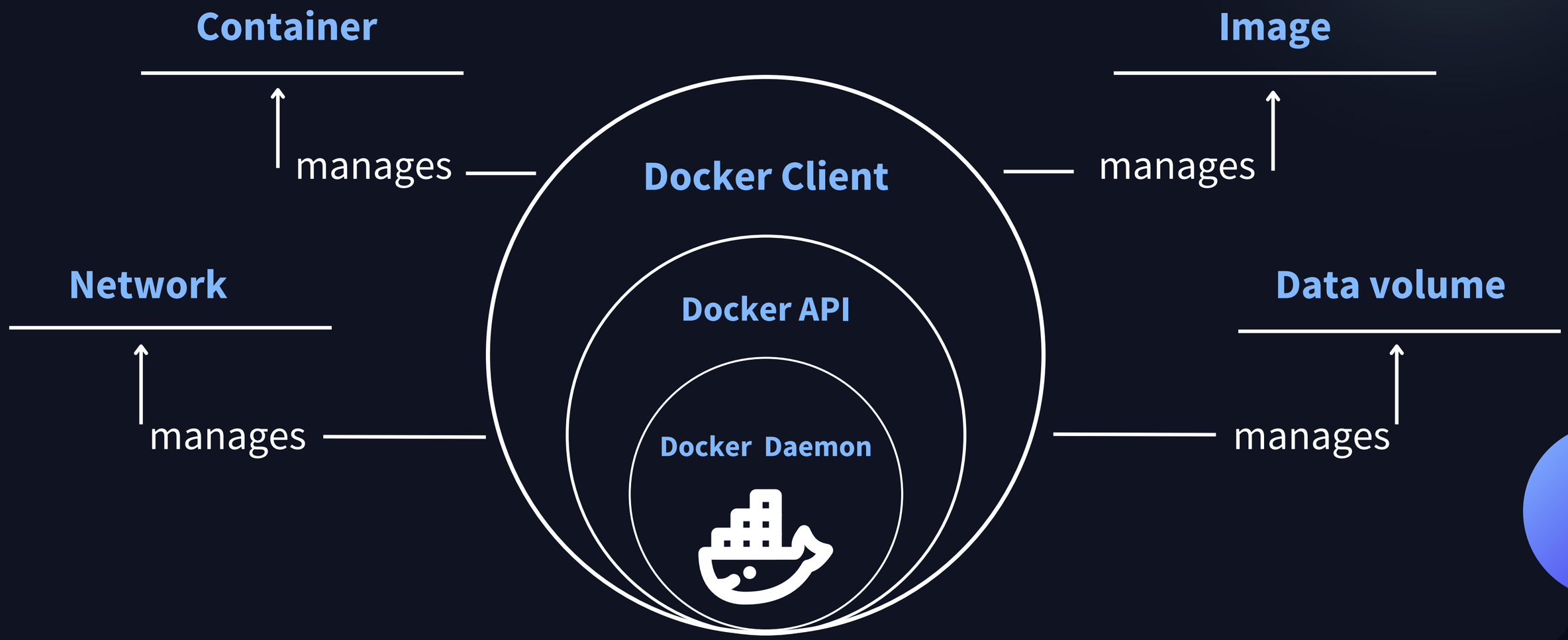


WLUG
COMMUNITY | KNOWLEDGE | SHARE

META

DOCKER HOST

WLUG
COMMUNITY | KNOWLEDGE | SHARE

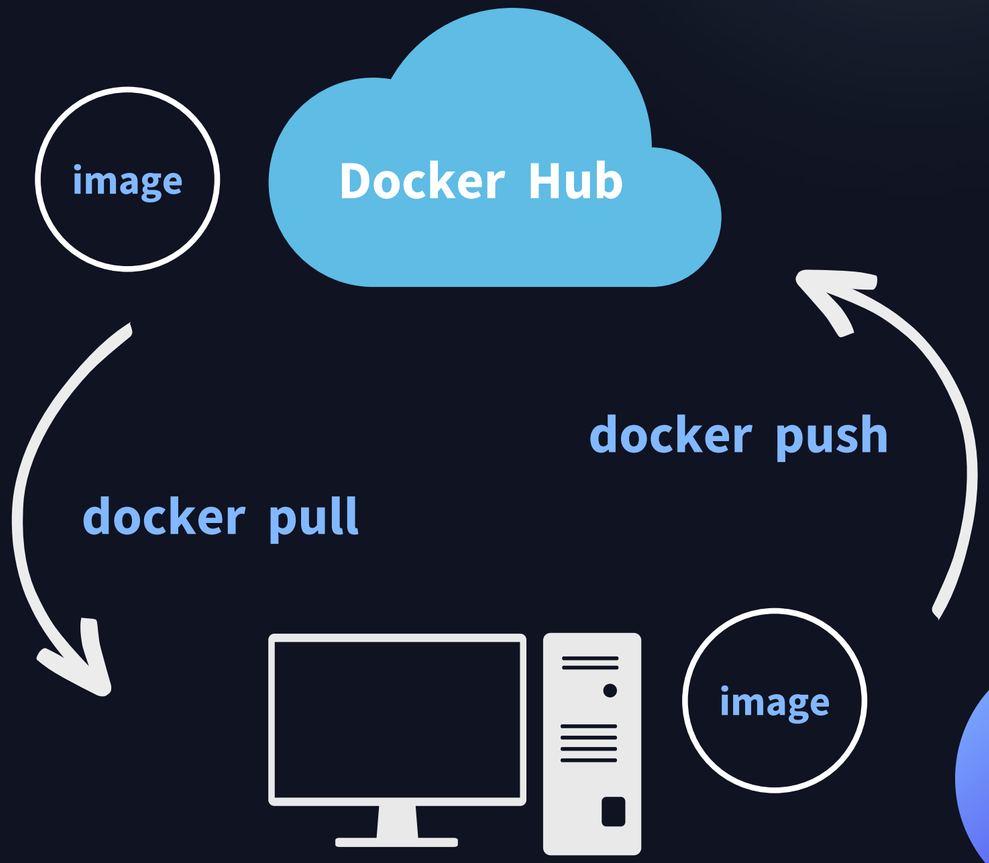


META

DOCKER REGISTRY

hub.docker.com

- A cloud-based registry service
- Share and access official docker images
- Two types of services provided:
  - Public Registry- Docker Hub
  - Private Registry

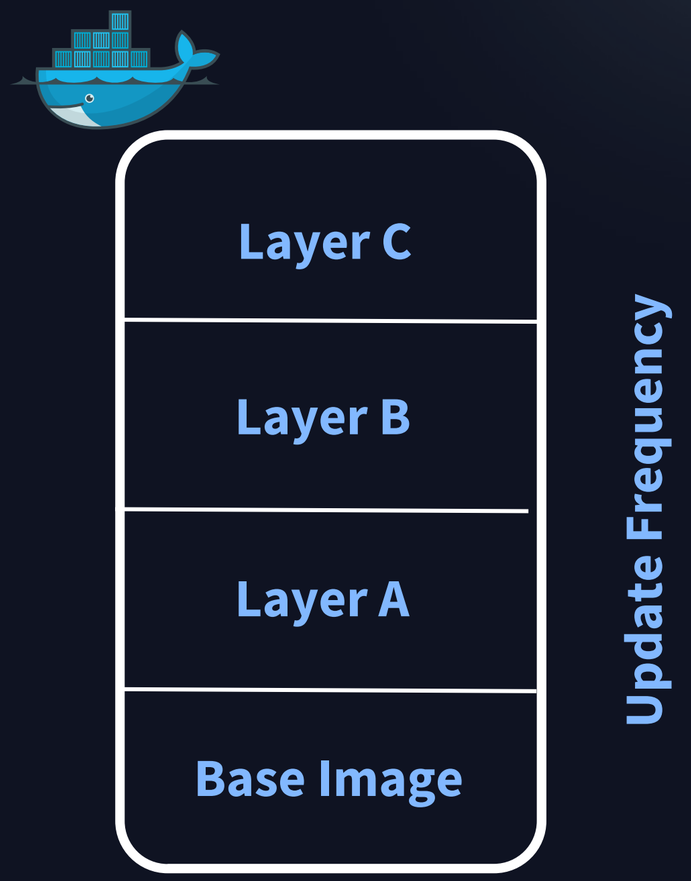


META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER IMAGES

- Read Only Template used to create container
- Holds all the dependencies to run a application
- Three ways to access an image-
  - Docker Hub
  - Existing Container
  - Build from Dockerfile



Layer by update frequency

META

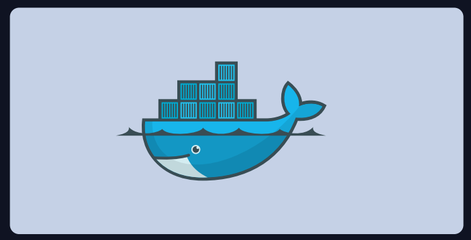
WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER CONTAINER

- Running instance of an image
- Mutable and follows layer file system
- Holds entire resources to run an application

DOCKER CONTAINER

Docker Image



Run

Docker Container

META

Dockerfile

- Text Document without any extension
- Contains set of instructions to build an image
- Used for automation of docker images

WLUG
COMMUNITY | KNOWLEDGE | SHARE

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Flow For Building Docker Container

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Hands On

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER FILE

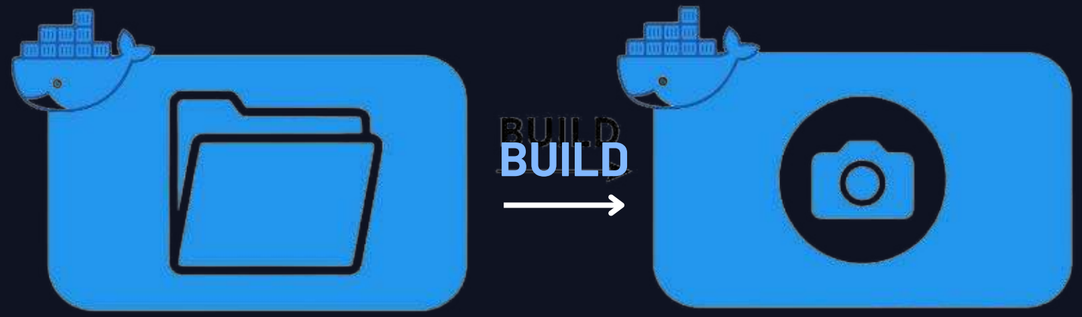
META

DOCKER FILE

Text Document without any extension

Instructions for docker to build Image

```
docker build
```



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

Dockerfile

home > prathamesh > Documents > wlug > Dockerfile

```
1  FROM python:3.8
2
3  MAINTAINER WLUG <www.wcewlug.org>
4
5  WORKDIR /app
6
7  COPY ./wlug.py ./
8
9  CMD ["python", "wlug.py"]
```

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

Format

INSTRUCTION arguments

e.g.

FROM ubuntu

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

Comment	# this is a comment-line

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

FROM	FROM <image>[:<tag>]
WORKDIR	WORKDIR /path/to/workdir

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

Environment variables	ENV variable_name=value
Accesing declared variables	$variable_name ${variable_name}

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

COPY & ADD	COPY <source> <destination> ADD <source> <destination>
RUN	RUN <command> RUN ["executable", "param1", "param2"]

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

CMD	CMD ["executable","param1","param2"] CMD ["param1","param2"] CMD command param1 param2

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MANAGING DATA ON CONTAINERS

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MANAGING DATA ON CONTAINERS

Host

Container

var/lib/mysql/data

Virtual File System

META

MANAGING DATA ON CONTAINERS

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Host

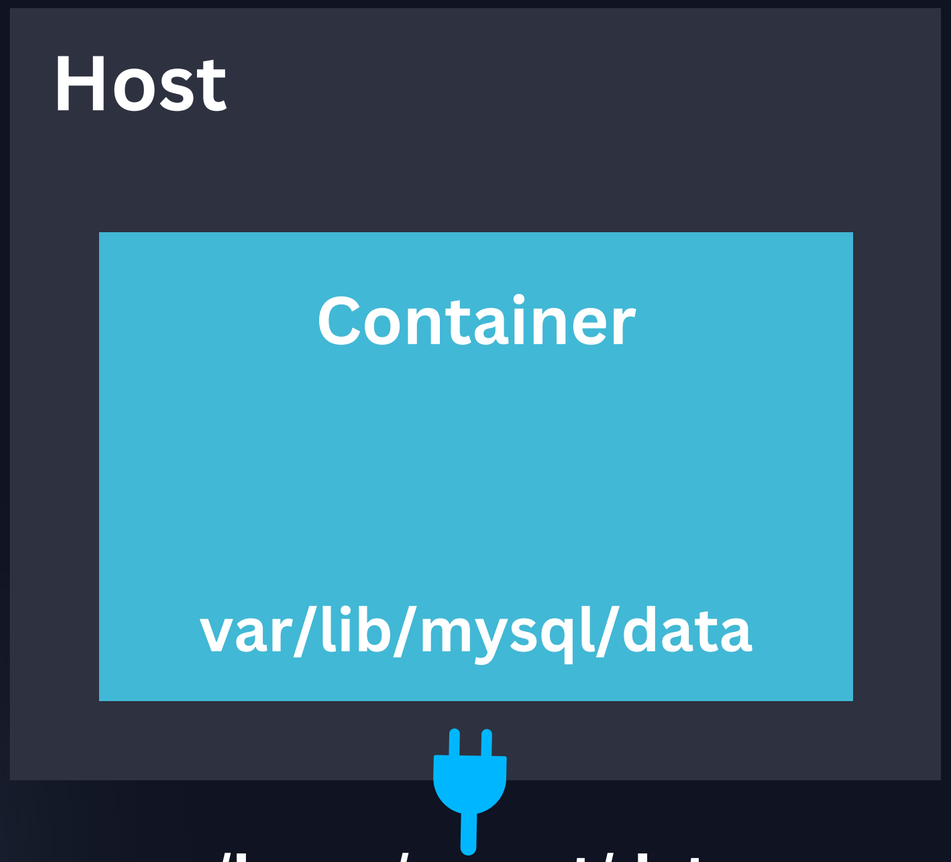
$ docker stop CONTAINER

After removing the container, the data is gone!!!

META

MANAGING DATA ON CONTAINERS

WLUG
COMMUNITY | KNOWLEDGE | SHARE



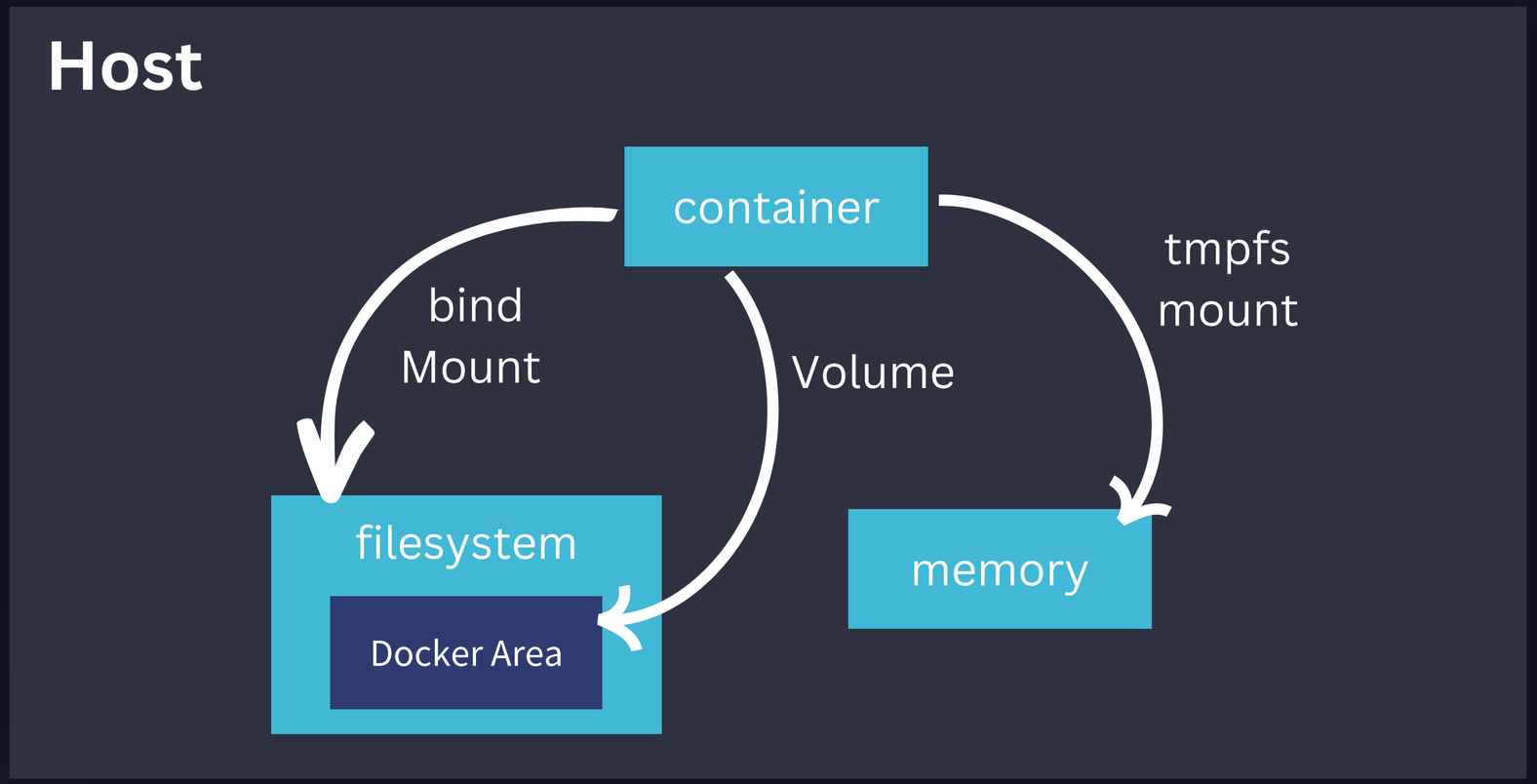
Virtual File System

Host File System (physical)

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MANAGING DATA ON CONTAINERS



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER VOLUMES

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER VOLUMES

Manage volumes

Create	$ docker volume create <volume-name>
List	$ docker volume ls
Inspect	$ docker volume inspect <volume-name>
Remove	$ docker volume rm <volume-name>

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER VOLUMES

PERSISTENCE

Host-Container
docker run -it -v <directory-name-on-host>: <directory-name-on-container> <image>

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER VOLUMES

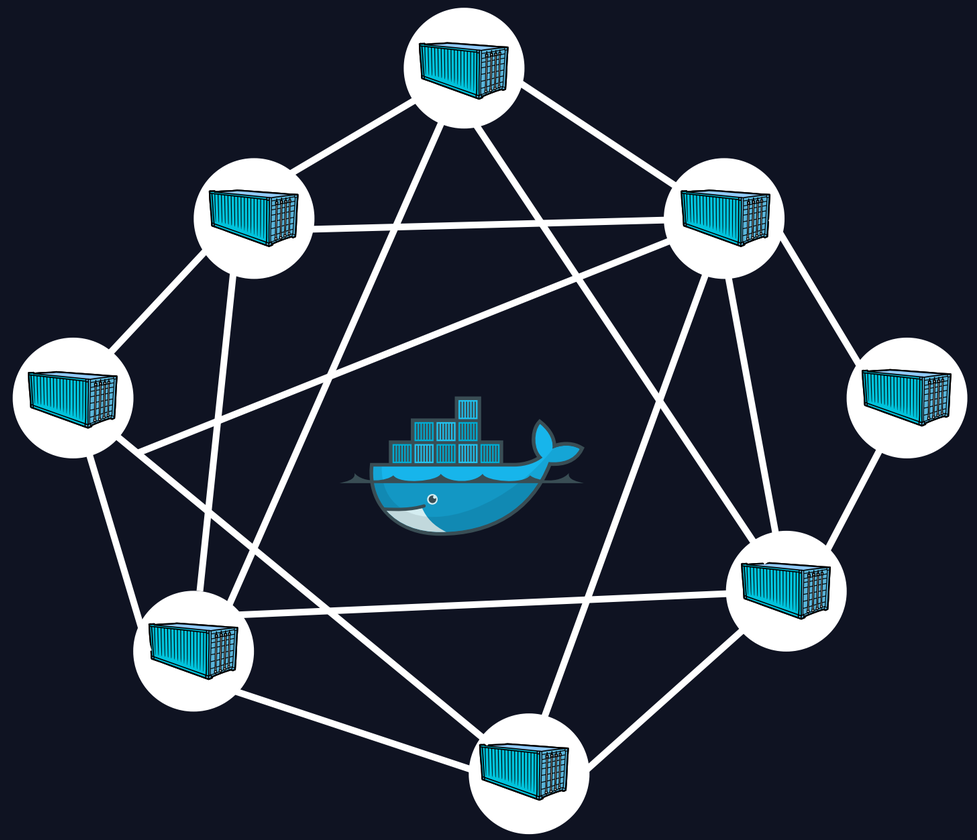
EPHEMERAL

Container 1	docker run --rm -it -v /<directory-name-on-container> --name <name-of-container> <image>
Container 2	docker run --rm -it --volumes-from <name-of-first-container> --name <new-name-of-container> <image>

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER NETWORKING



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

CONTAINER AND HOST PORT

- Multiple containers can run on your host machine
- Your laptop has only certain ports available

META

TYPES OF NETWORKING

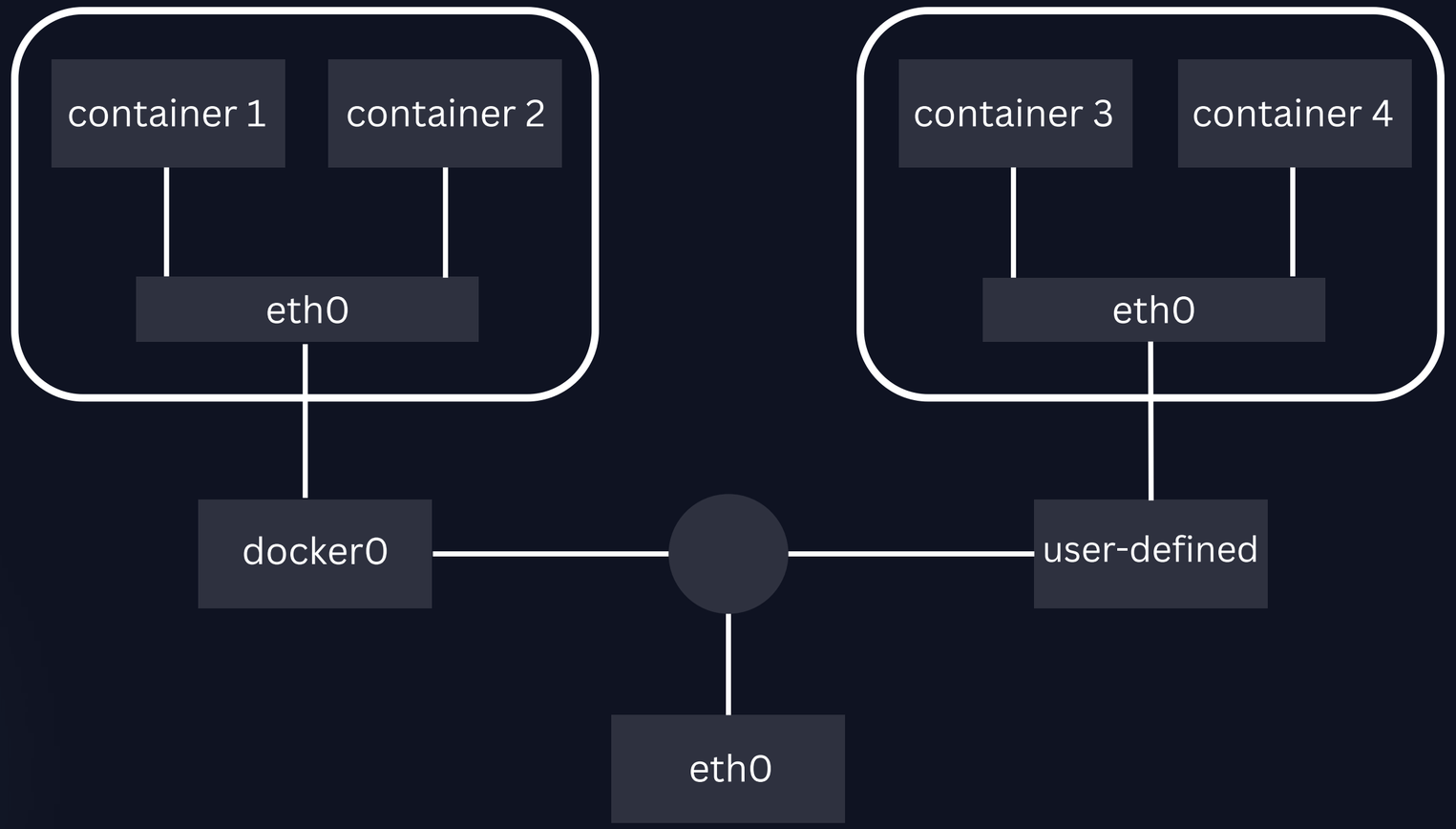
WLUG
COMMUNITY | KNOWLEDGE | SHARE

- Bridge Network
- Overlay Network
- Host Network
- Macvlan Network

META

DEFAULT BRIDGE AND USER-DEFINED BRIDGE NETWORK

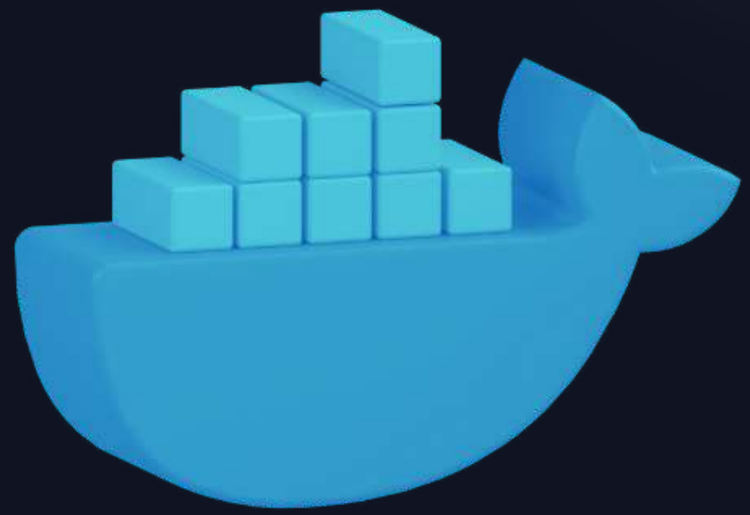
WLUG
COMMUNITY | KNOWLEDGE | SHARE



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS AND DOCKER-COMPOSE



META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS

Database

Network

Server

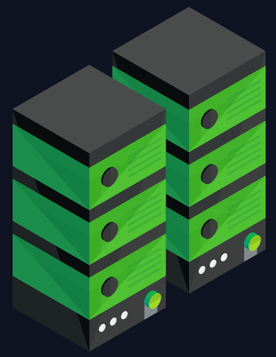
User Interface

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Database

Depends On



Server

Depends On

User Interface

META

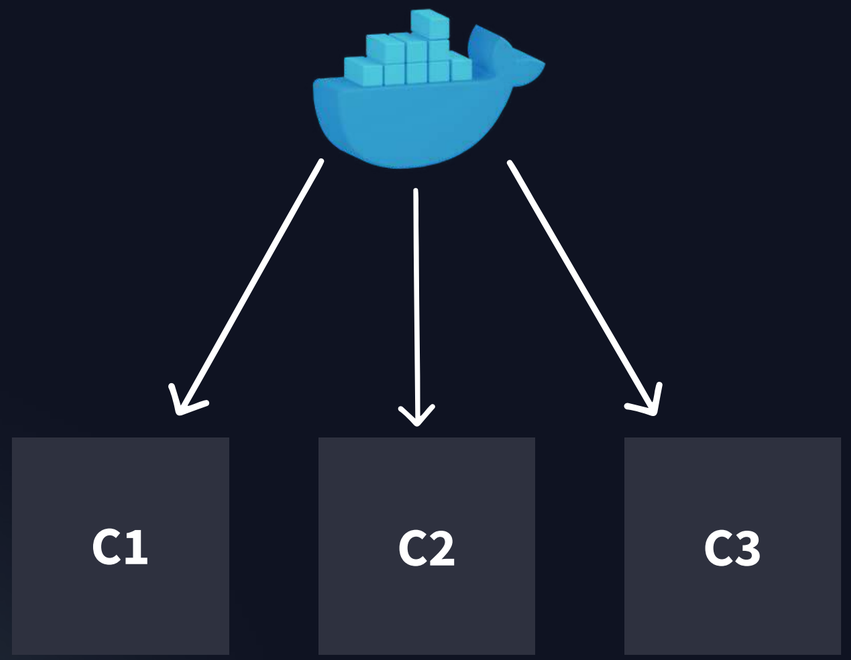
WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER-COMPOSE

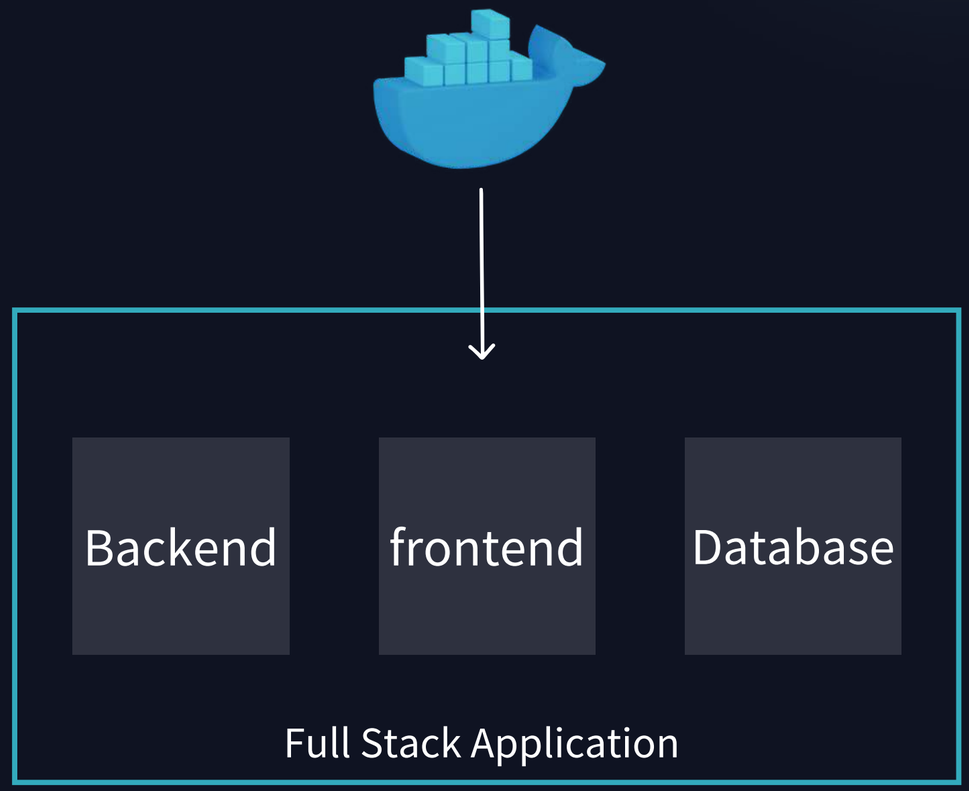
META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER



DOCKER-COMPOSE



META

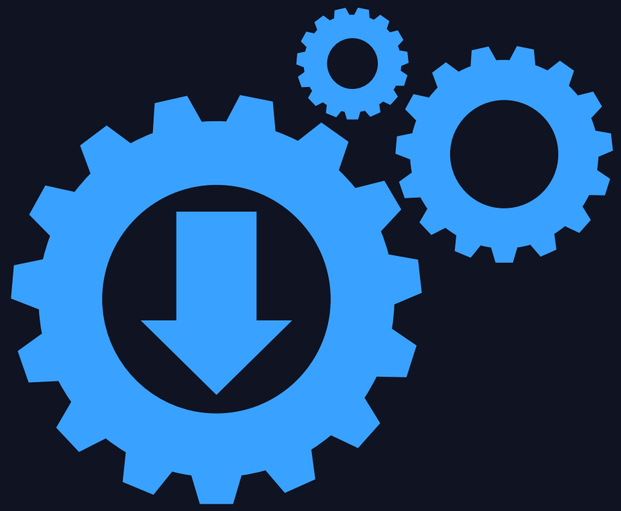
WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHY DOCKER-COMPOSE?

- Defining and sharing multi-container applications
- Handling multiple containers is difficult
- Easy and fast management
- Multitasking

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

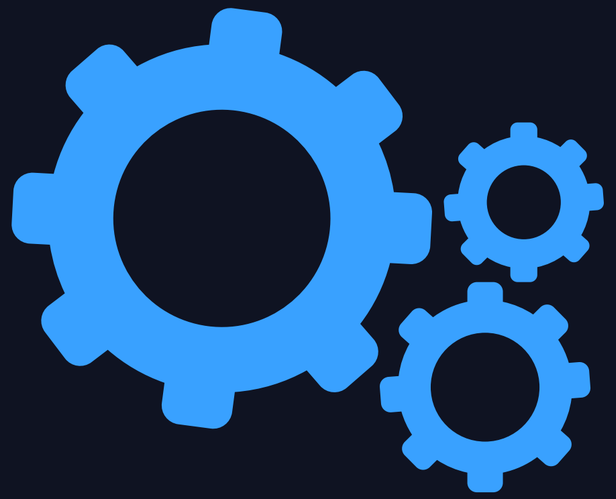


INSTALLATION

- sudo apt-get update
- sudo apt-get install docker-compose-plugin

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE



HOW DOES IT WORK?

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

```yaml
version: '3.7'

services:

  service1:
    image: image_name
    volume:
      - /path/to/host:/path/to/containe
    networks:
      - networkName
    command: command

  service2:
    image: image_name
    volume:
      - /path/to/host:/path/to/containe
    networks:
      - networkName
    command: command

  service3:
    ...
```

STEP-1
Create docker-compose.yaml

- version
- services
- volumes & networks

META

WLUG
COMMUNITY | KNOWLEDGE | SHARE

STEP-2

Run `docker-compose.yaml`

- **Start Compose**

sudo docker compose up

- **Stop Compose**

sudo docker compose down

WLUG
COMMUNITY | KNOWLEDGE | SHARE

THANK YOU