Walchand College of Engineering, Sangli

Walchand Linux Users' Group

WLUG
COMMUNITY | KNOWLEDGE | SHARE

METAMORPHOSIS

An Insightful Workshop

9th & 10th March 2024

DOCKER

- Basics Of Containerization
- Dockerfile
- Networking & Compose

GOLANG

- Basics Of GoLang
- GoRoutines & Concurrency
- Setting up a GoLang Project

CONNECT WITH US

Main & Mini CCF

LIMITED SEATS

REGISTER AT

EXCITING PRIZES

Entry Fee: Rs 299/-

meta2k24.wcewlug.org

Mr. D. N. Gangji
President
Walchand Linux Users' Group

Dr. M. A. Shah
HoD Computer Science and Engineering

Dr. R. R. Rathod
HoD Information Technology

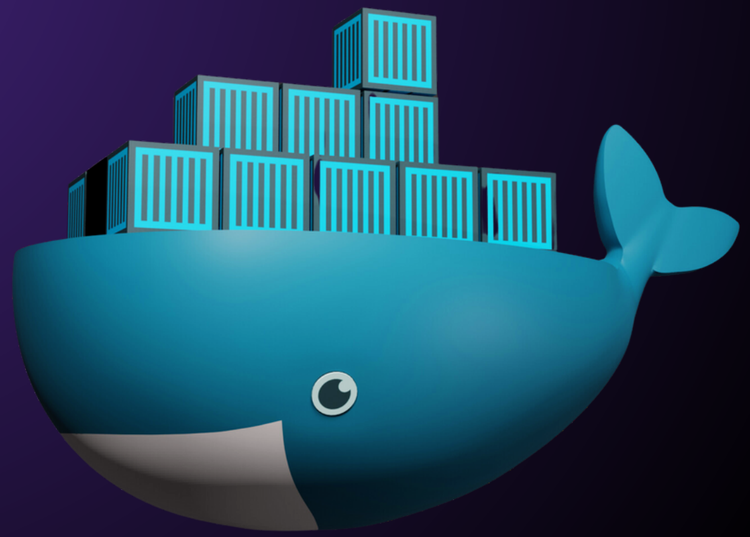
Dr. A. J. Umbarkar
Staff Advisor
Walchand Linux Users' Group

Dr. A. R. Surve
Assoc. Dean Student Activities and Staff Advisor

Dr. U. A. Dabade
I/C Director
Walchand College of Engineering

WLUG

COMMUNITY | KNOWLEDGE | SHARE



Docker

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Content of the session

Part-1

DevOps Introduction

Part-2

Introduction to Docker

Part-3

Docker and Containerization

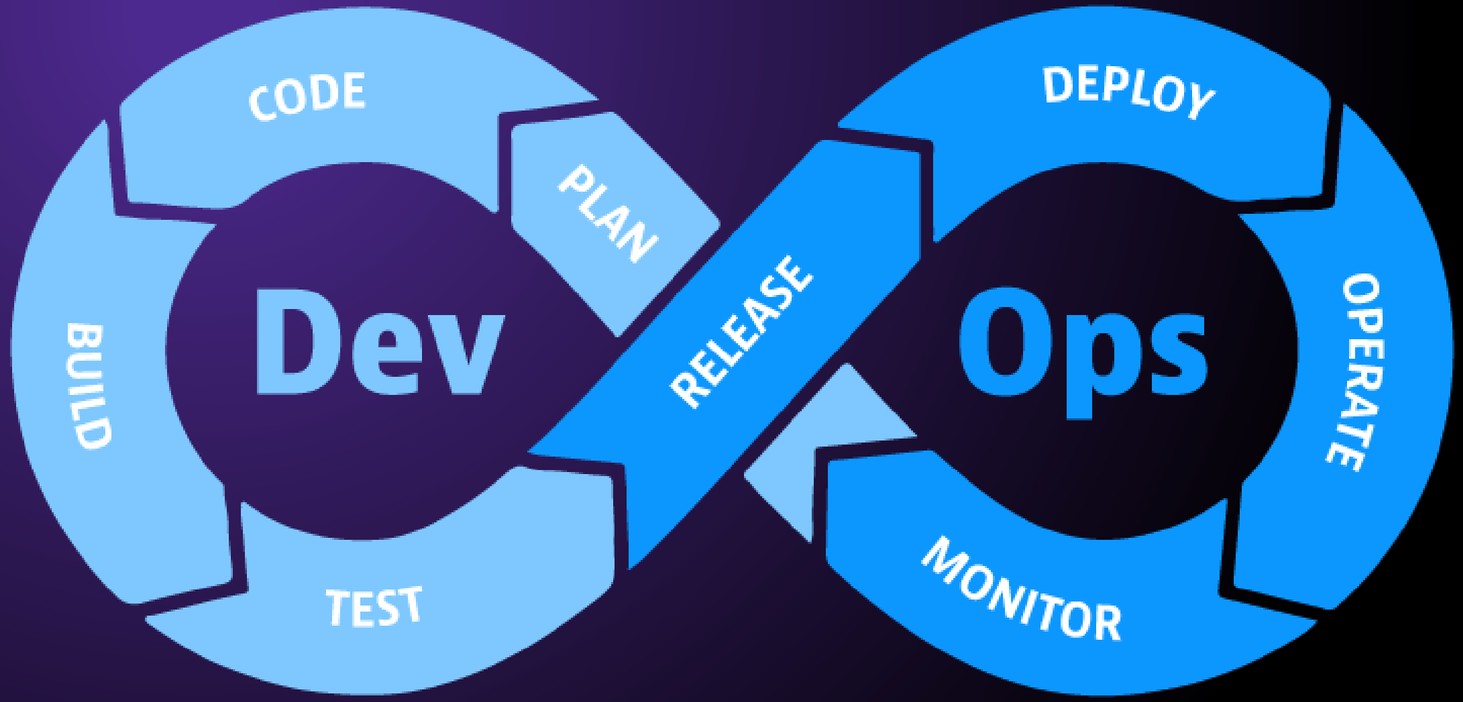
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Software Development Life Cycle

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

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DevOps



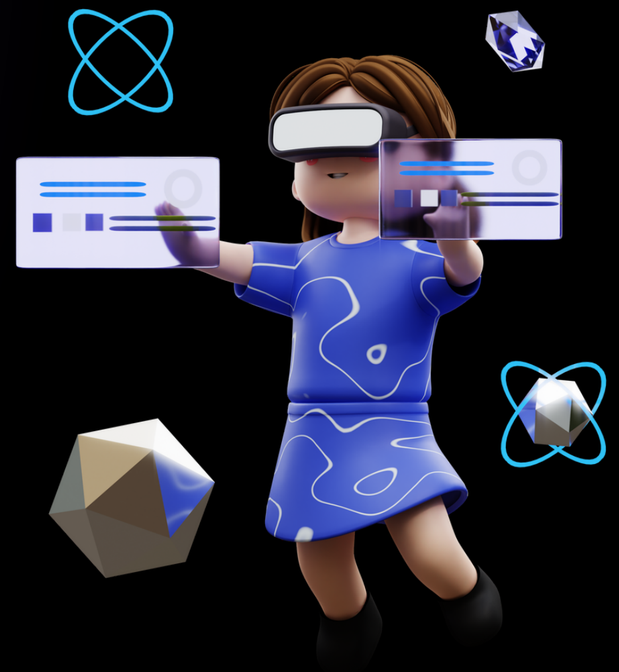
WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHAT IS VIRTUALIZATION?

WLUG
COMMUNITY | KNOWLEDGE | SHARE

VIRTUALIZATION

- Act of creating a virtual version of something
- It includes virtual computer hardware, virtual storage devices and virtual network resources



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Virtual Machine

- Runs as a process in an application window
- VMs can run multiple operating system environments on a single physical computer
- It saves physical space, time and management costs

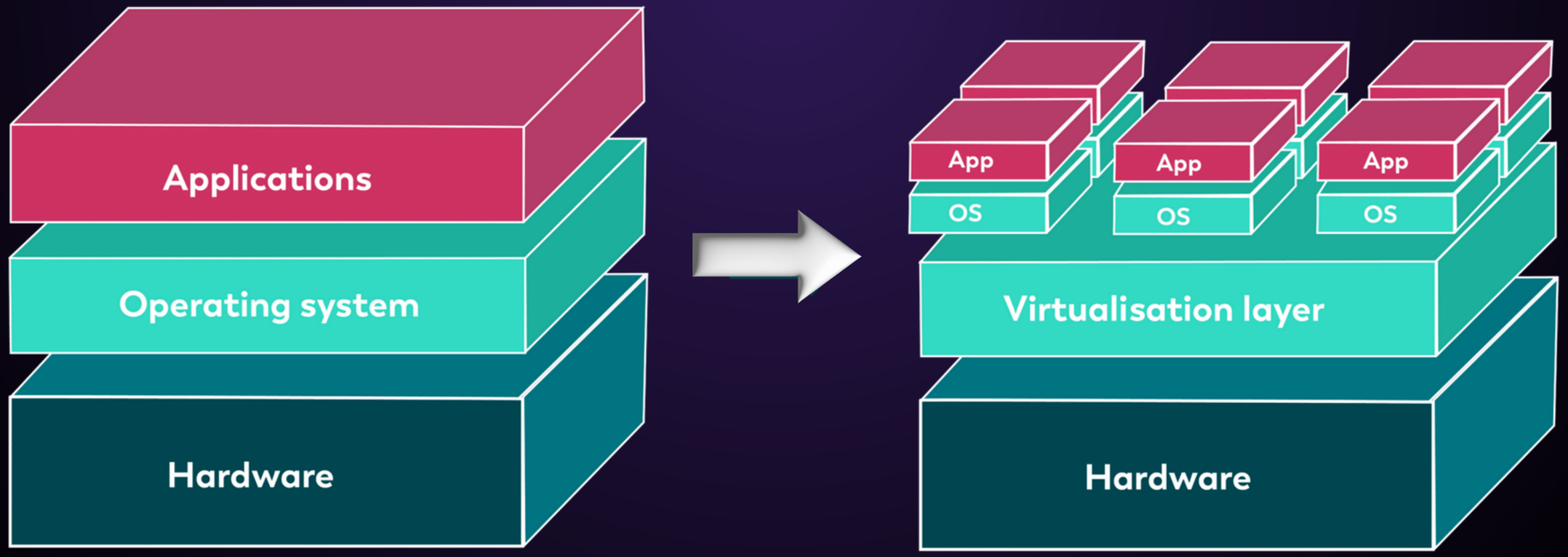
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Hypervisor

- 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 VMs by virtually sharing its resources

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Physical Server Vs Virtualization



TRADITIONAL ARCHITECTURE

VIRTUAL ARCHITECTURE

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Physical Server Vs Virtualization

• Resources are not shared between multiple users	• Resources are shared between multiple users
• Includes memory, hard-drive, processor, network connection and OS	• It is software based that emulates all the functions of a physical server

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Physical Server Vs Virtualization

• Hardware is used directly by an OS	• A hypervisor manages the virtualized resources
• It is used to run a single instance of an OS	• Runs an independent OS on top of the hypervisor

WLUG

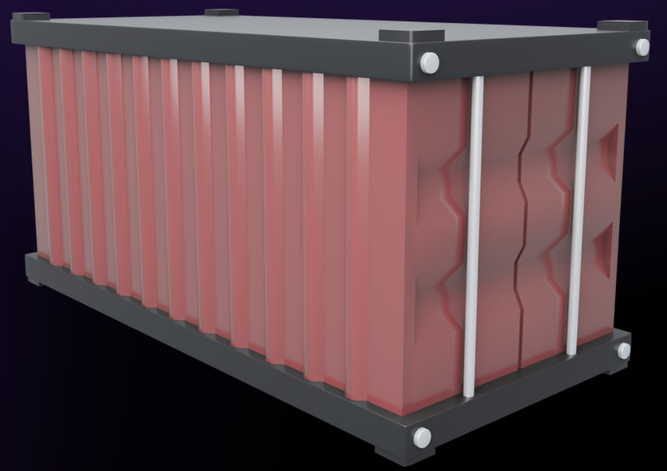
COMMUNITY | KNOWLEDGE | SHARE

WHAT IS CONTAINERIZATION?

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Container

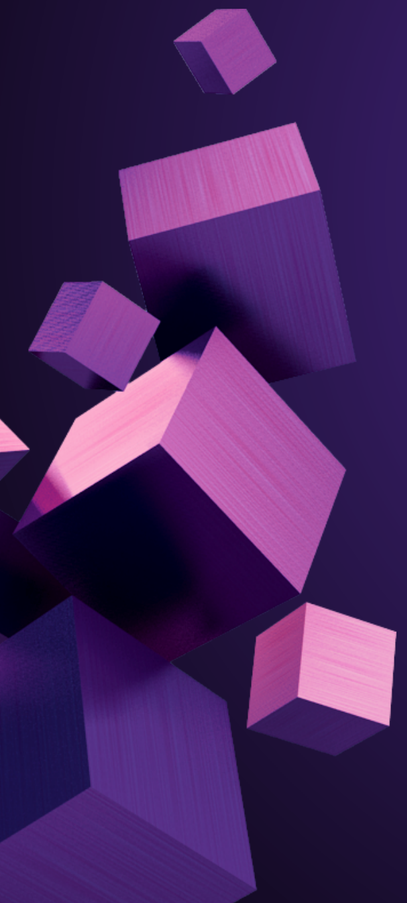
- A container is like a portable package for software
- It wraps up the code and everything it needs to run smoothly, making sure the application works consistently whether it's on one computer or another



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Containerization

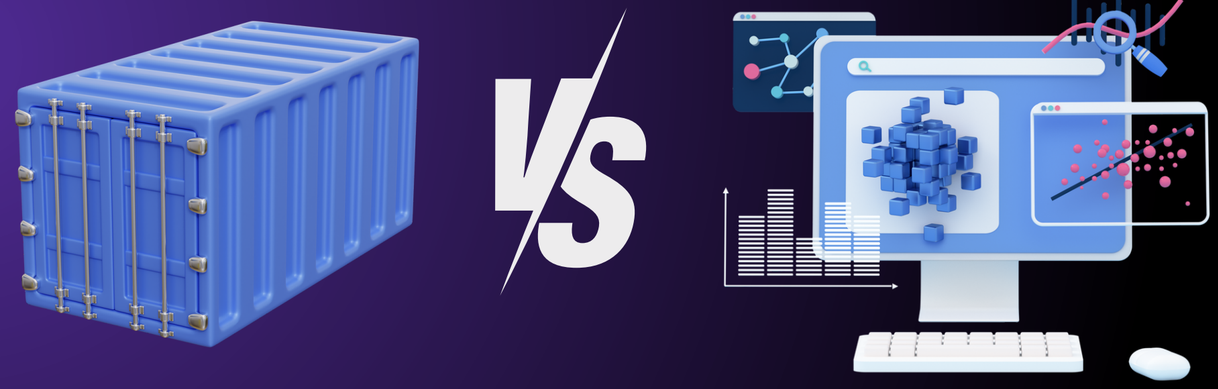
- Containerization is a process that bundles an application code with all the files and libraries it needs to run on any infrastructure
- It is Operating System level Virtualization
- Containers are said to share the host system’s kernel with other containers



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Containers vs VMs

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



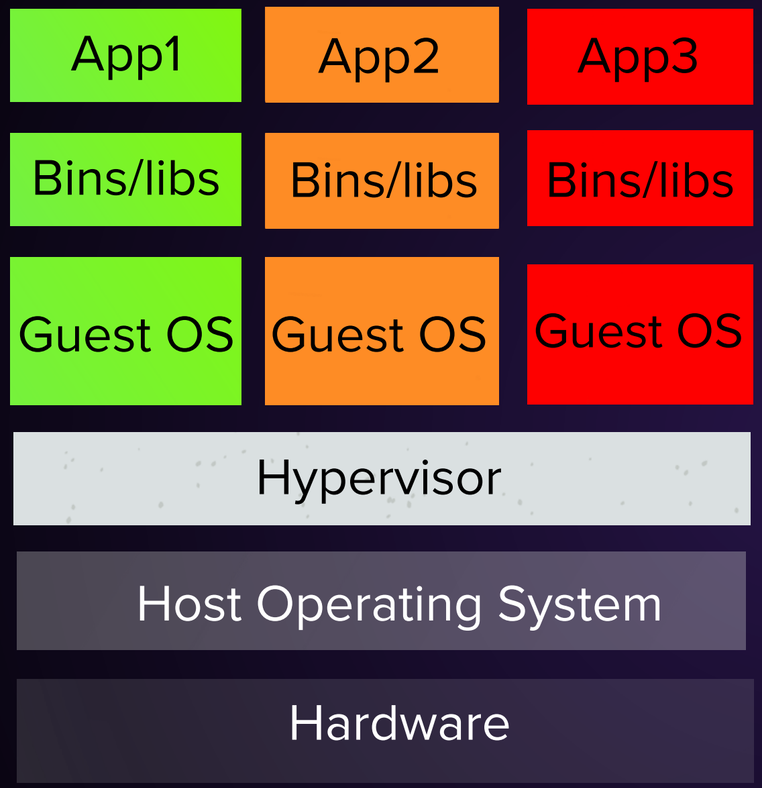
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Operating System

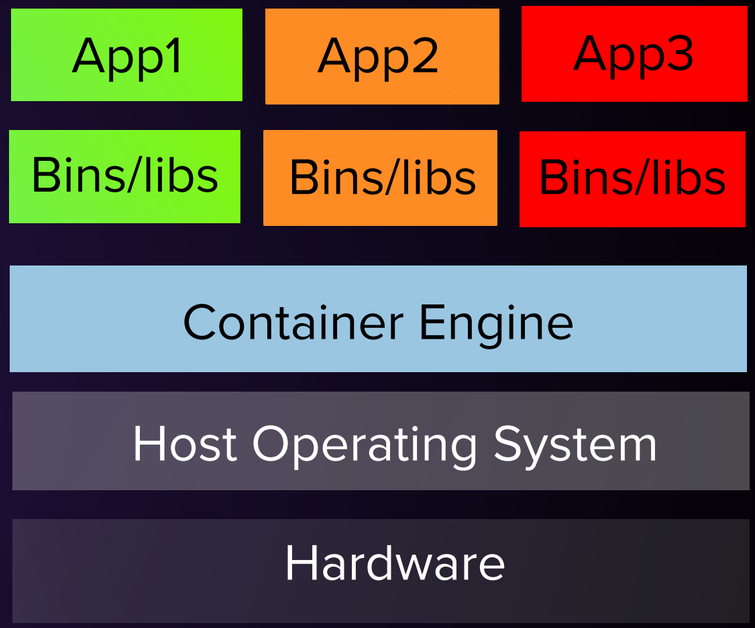
Containers	VMs
It has only essential files of the Operating System	It has a whole copy of the Operating System

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Architecture



Virtual Machine

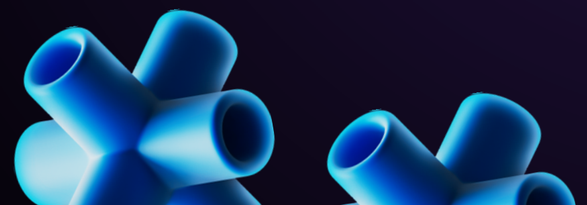


Container

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Isolation

Containers	VMs
Container isolation on the same server ensures complete separation between multiple containers.	It provides complete isolation between the guest OS and the host



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Speed & Portability

Containers	VMs
Containers are faster	VMs, being full copies of the host server, are resource-heavy, leading to slower performance.

WLUG
COMMUNITY | KNOWLEDGE | SHARE

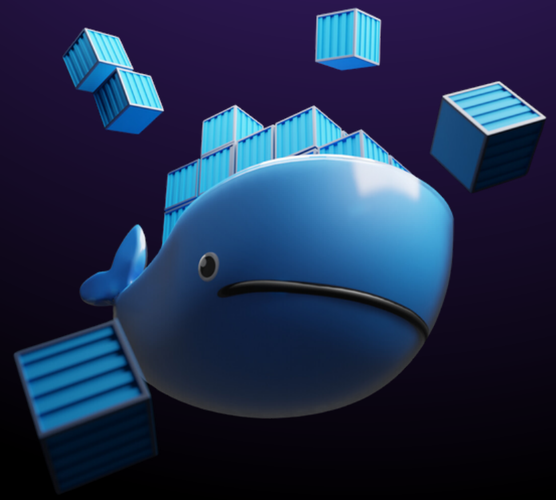
Resource & Efficiency

Containers	VMs
Containers use less memory and hence are more efficient	VMs use more resources and hence are less efficient

WLUG
COMMUNITY | KNOWLEDGE | SHARE

What is Docker?

- An open-source platform designed to make it easier for developing, shipping, and running applications
- A Platform for Consistent and Isolated Environment to run application
- Developed by Kamel Founadi, Solomon Hykes, and Sebastien Pahl in 2013
- It was written in Go programming language



WLUG
COMMUNITY | KNOWLEDGE | SHARE

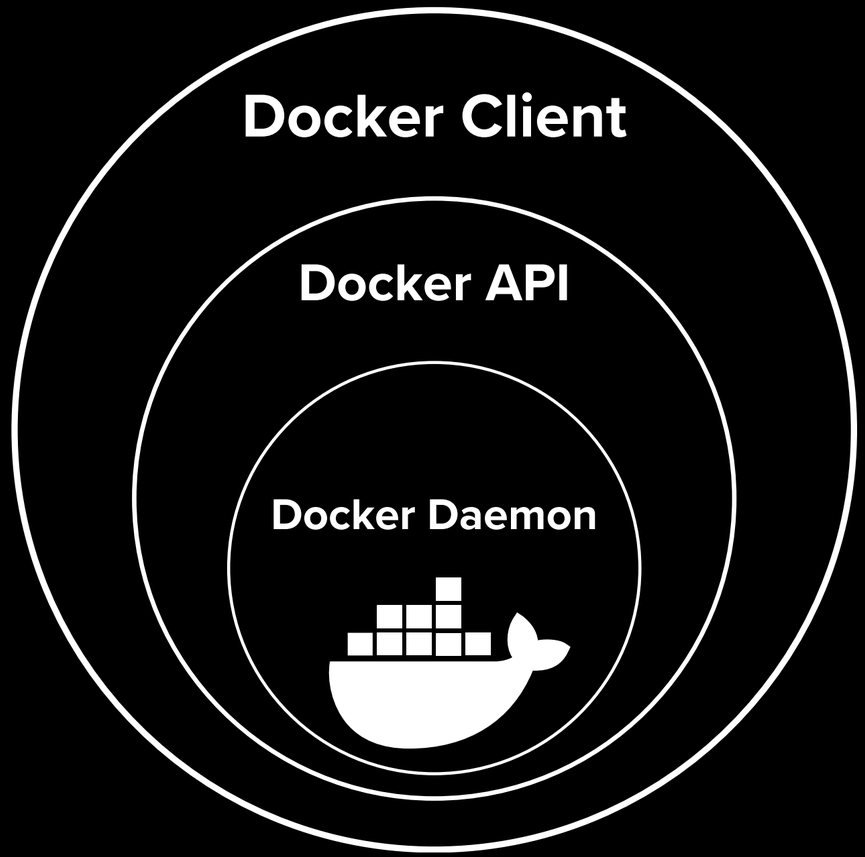
Why Docker ?

- Solution to “But Works On My Machine Problem...”
- Rapid Application Developement
- Portability
- Efficient Utilization

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker Engine

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



WLUG
COMMUNITY | KNOWLEDGE | SHARE

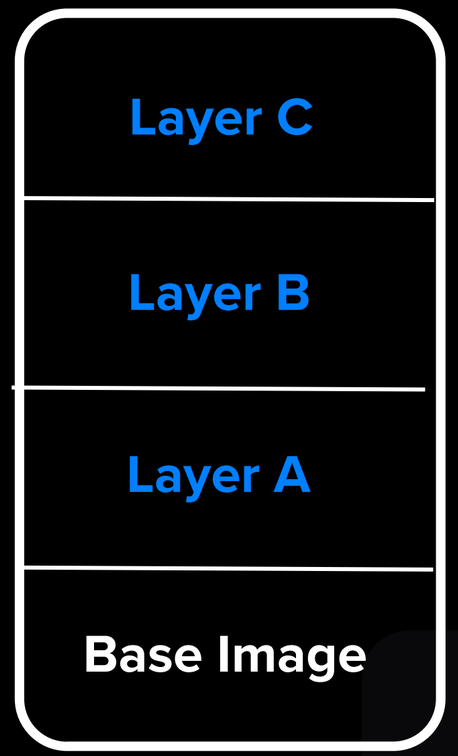
Docker Components

- DOCKER DAEMON
- DOCKER HOST
- DOCKER IMAGE
- DOCKER CLIENT
- DOCKER CONTAINER
- DOCKER REGISTRY

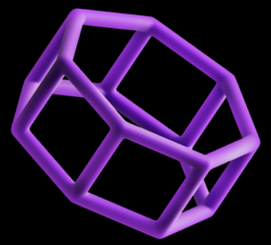
WLUG
COMMUNITY | KNOWLEDGE | SHARE

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

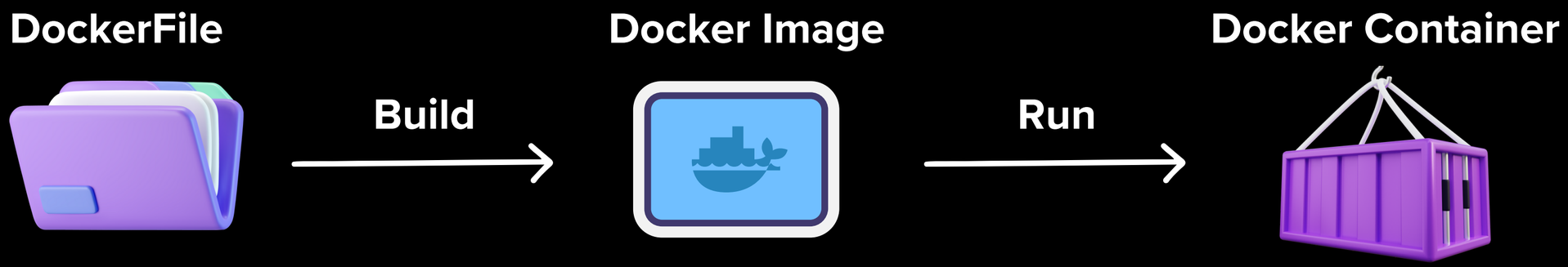


WLUG
COMMUNITY | KNOWLEDGE | SHARE



Docker Container

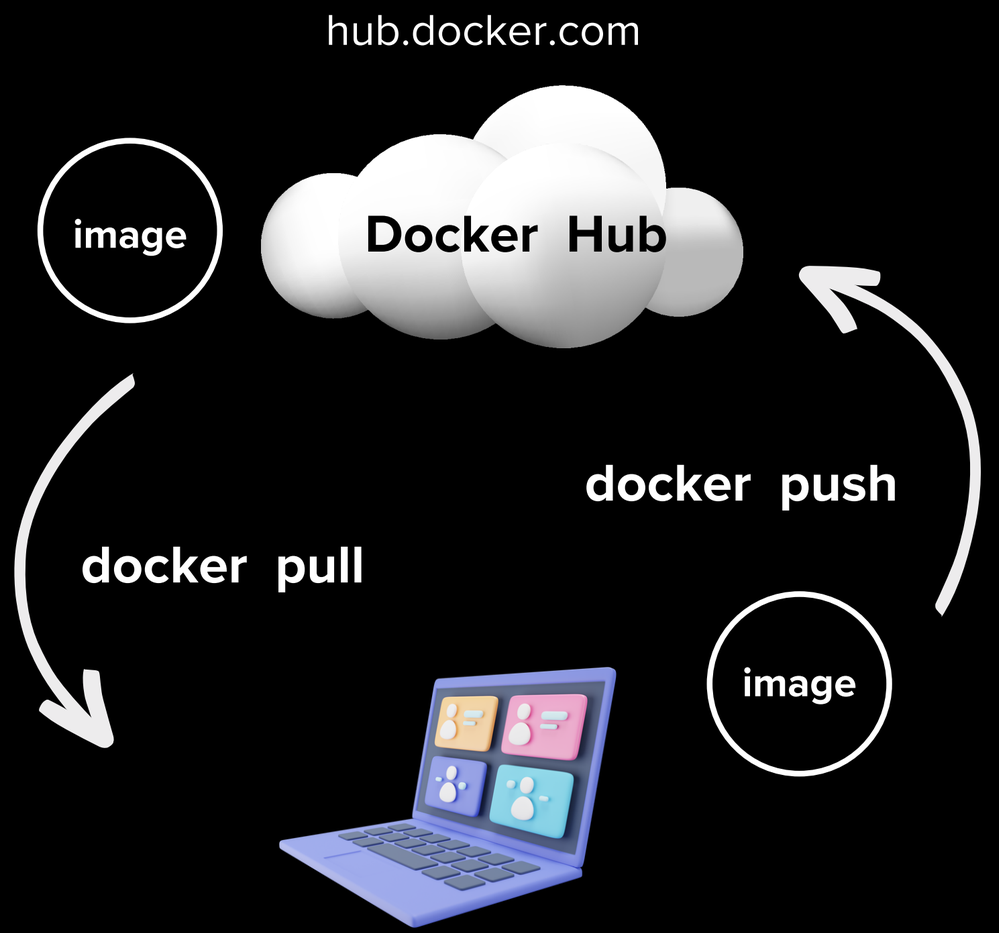
- Running instance of an image
- Run Independantly as Isolated Process
- Mutable and follows layer file system
- Holds entire resources to run an application



WLUG
COMMUNITY | KNOWLEDGE | SHARE

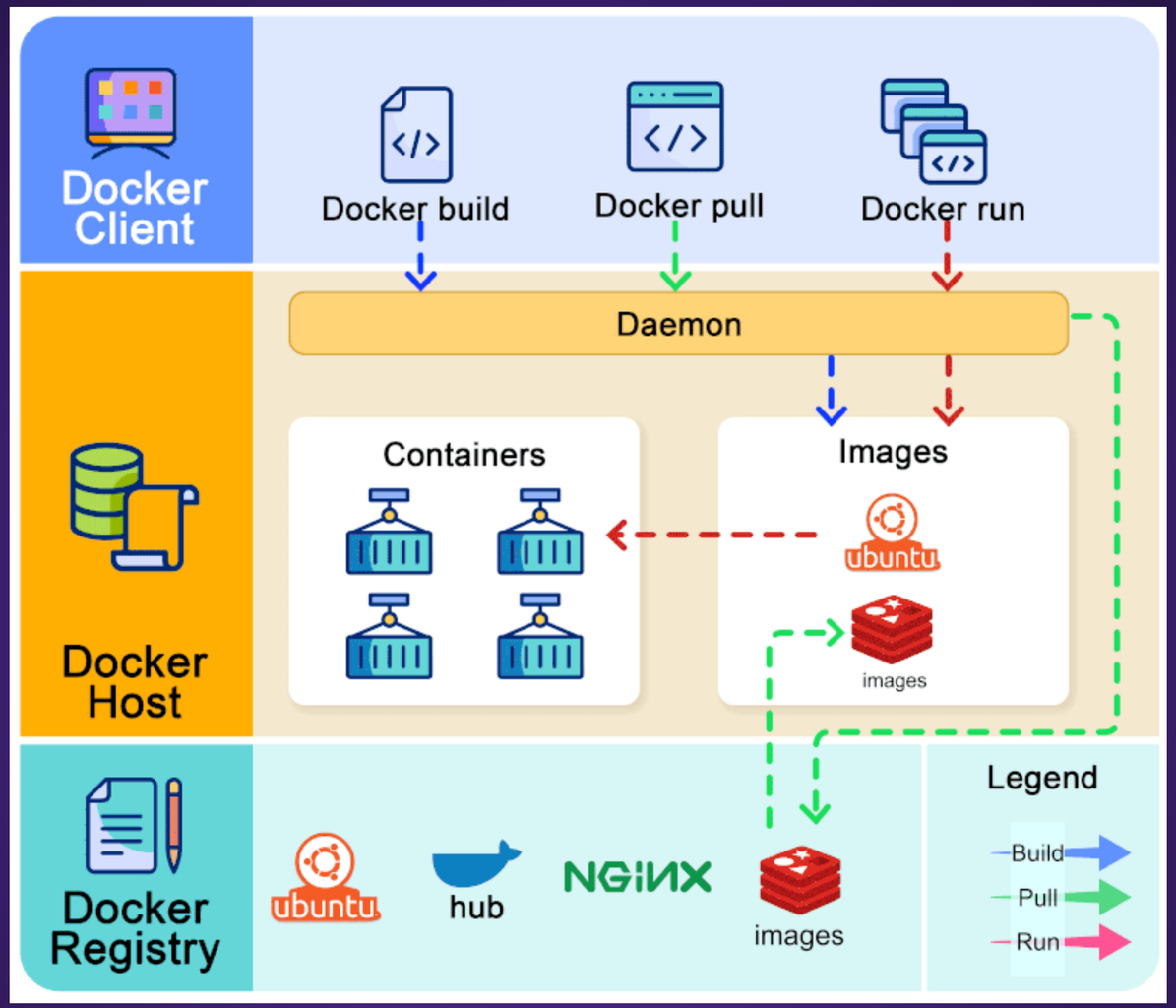
Docker Registry

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



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker Architecture

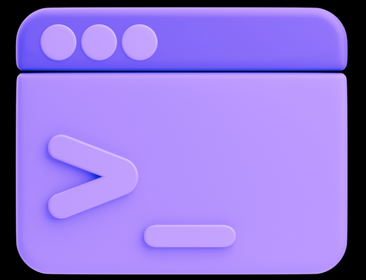


WLUG
COMMUNITY | KNOWLEDGE | SHARE

Linux Commands

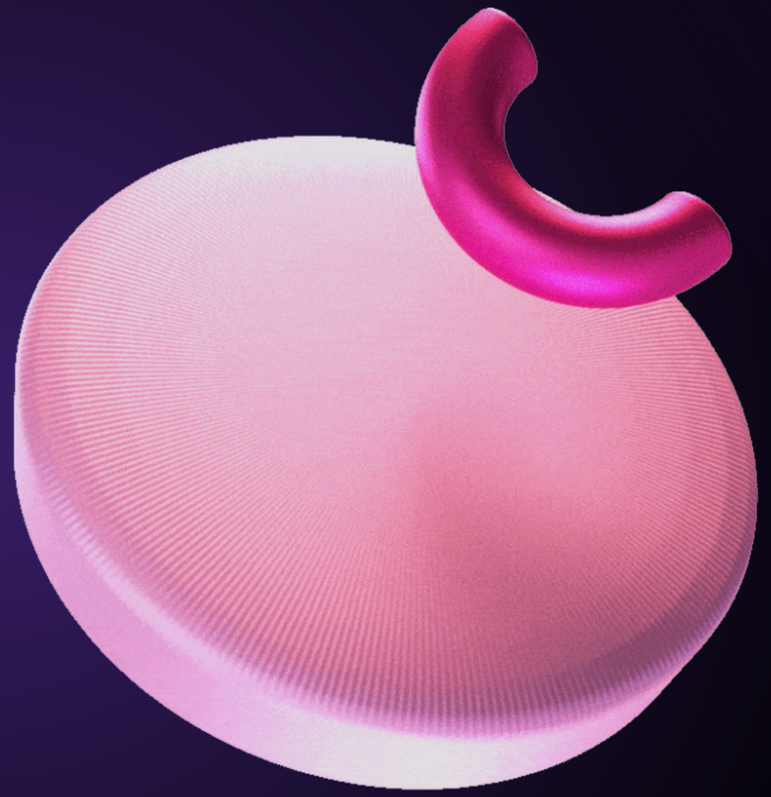
Basic Linux Commands

pwd	Prints the current working directory
ls	Lists the files and directories in the current working directory
cd	Changes the current working directory
cat	Displays the content of a file
mkdir	Creates a new directory



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker Installation



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Hands On !

Linux

Images

Containers

Dockerfile

Hello Docker !

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Linux

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Container

WLUG

COMMUNITY | KNOWLEDGE | SHARE

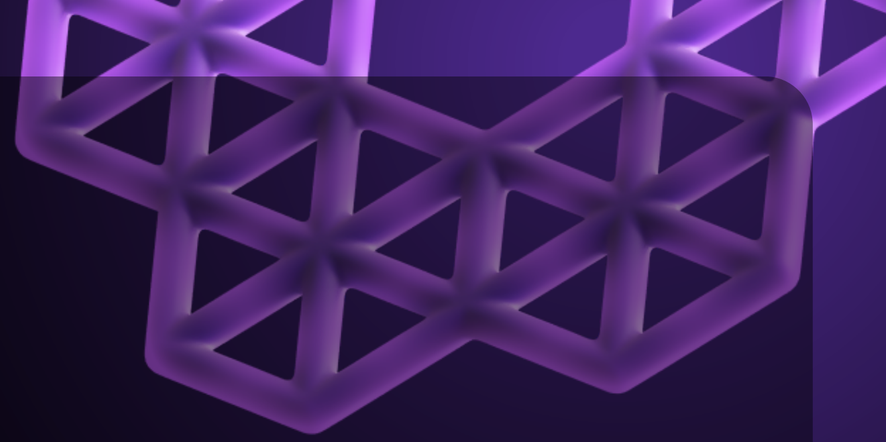
Virtual Machine

WLUG

COMMUNITY | KNOWLEDGE | SHARE

Dockerfile

WLUG
COMMUNITY | KNOWLEDGE | SHARE



Images

Basic Image Commands

docker images	Lists all the locally available Docker images
docker pull	Downloads a Docker image from a registry
docker rmi	Removes one or more Docker images
docker tag	Creates a tag TARGET_IMAGE that refers to SOURCE_IMAGE

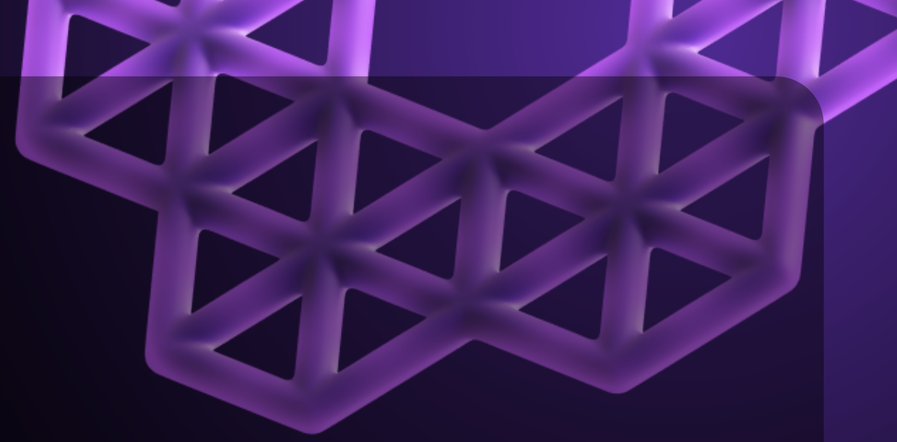
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Containers

Basic Container Commands

docker run	Create and start a container from an image
docker ps	List running containers
docker stop	Stop a running container
docker exec	Run a command in a running container
docker rm	Remove one or more containers

WLUG
COMMUNITY | KNOWLEDGE | SHARE



Dockerfile

Dockerfile Parameters

FROM	Specifies the base image for the new image
WORKDIR	Sets the working directory for subsequent instructions
COPY	Copies files or directories from the host machine to the container
EXPOSE	Informs Docker that the container will listen on the specified network ports at runtime
CMD	Provides a default command to execute when the container starts

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Containers Harmony

Content of the session

Part-1

Docker Volumes

Part-2

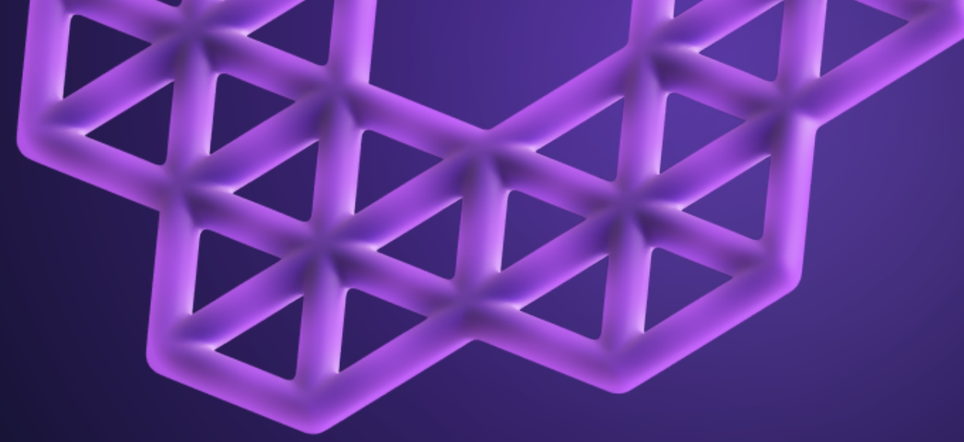
Docker Networking

Part-3

Docker Compose

WLUG

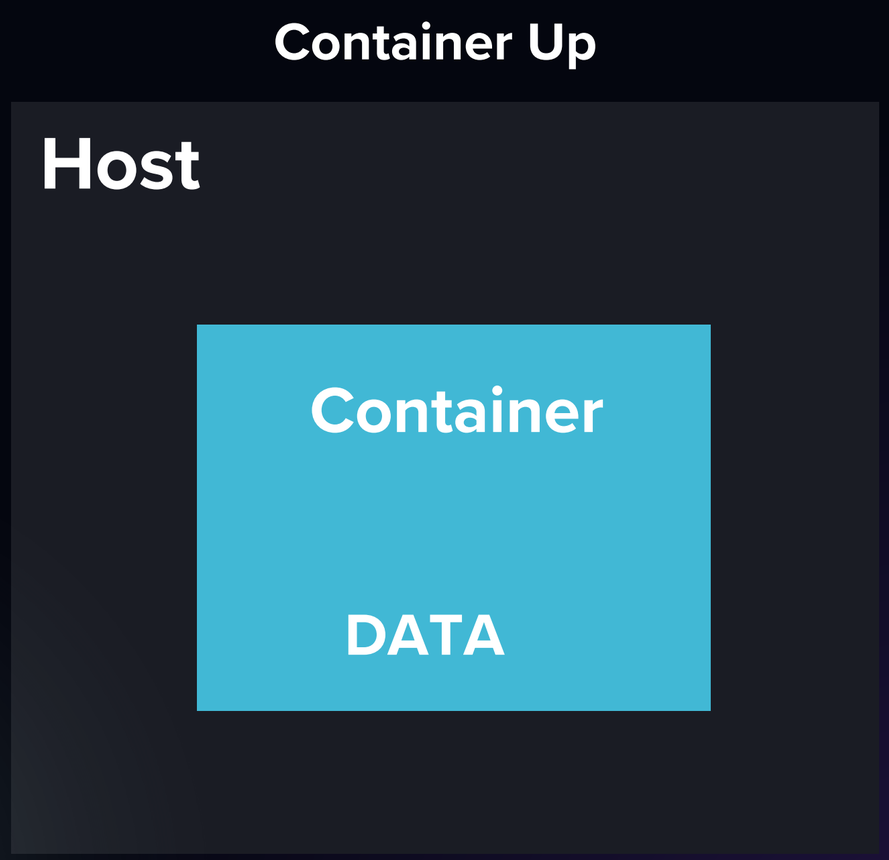
COMMUNITY | KNOWLEDGE | SHARE



MANAGING DATA ON CONTAINERS

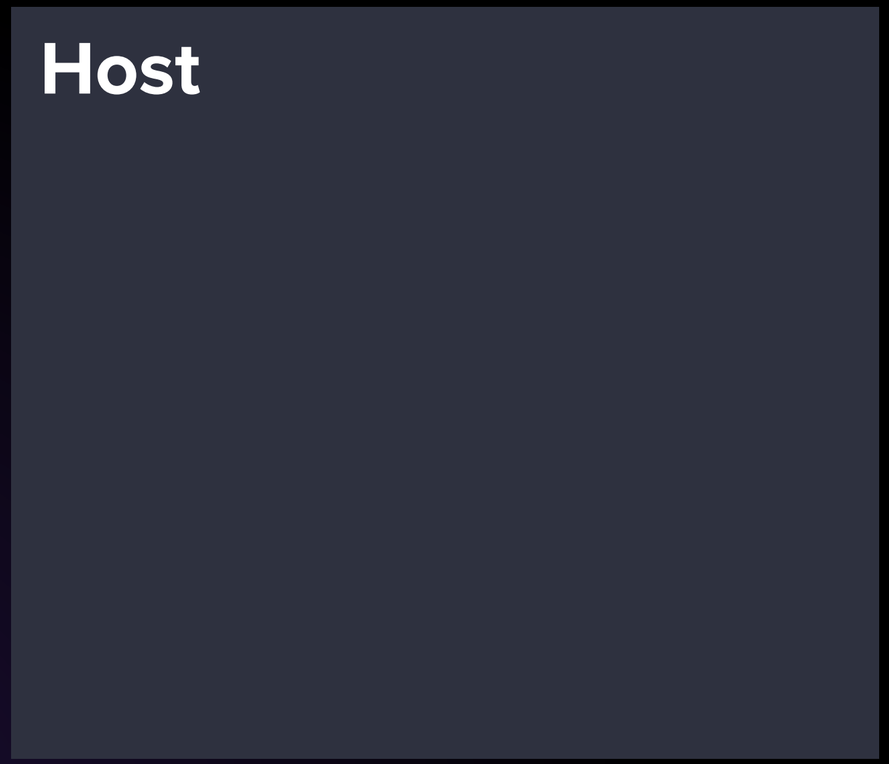
WLUG
COMMUNITY | KNOWLEDGE | SHARE

MANAGING DATA ON CONTAINERS



WLUG
COMMUNITY | KNOWLEDGE | SHARE

MANAGING DATA ON CONTAINERS



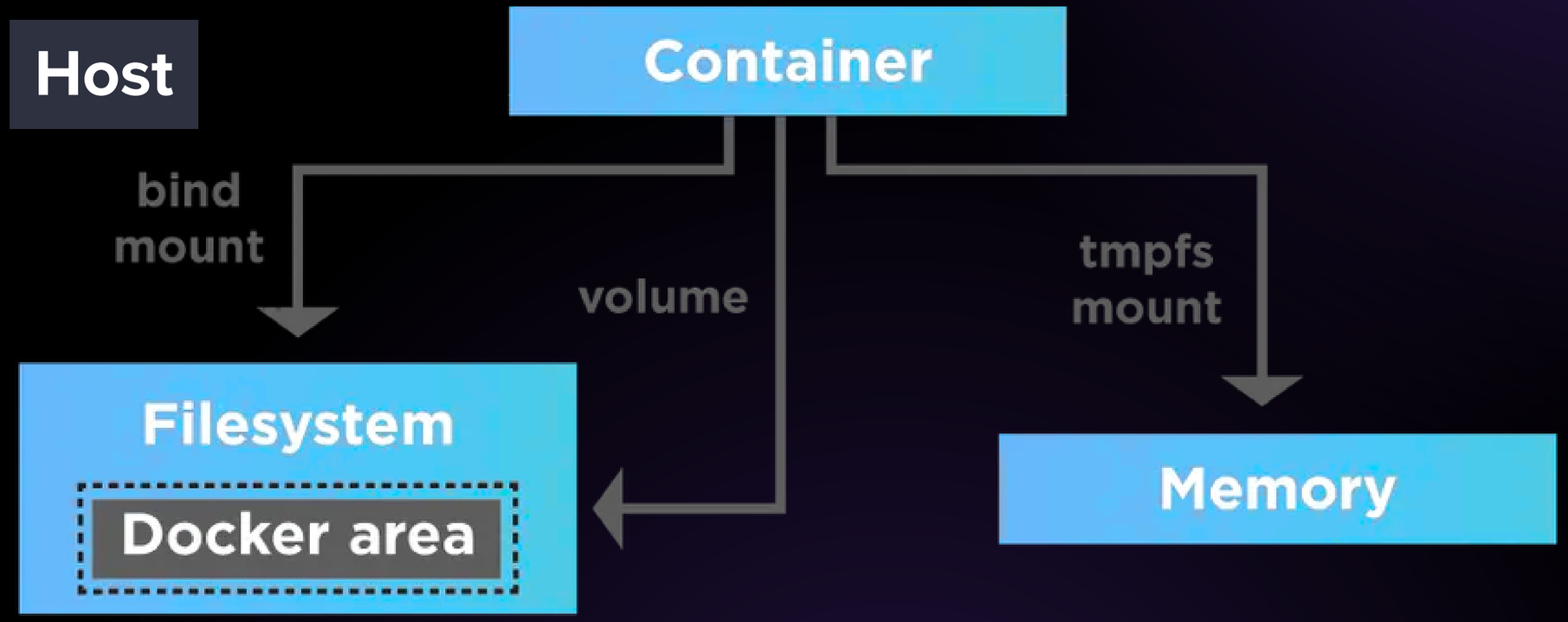
Container Down

After removing the container,
the Data is gone !!

WE LOST
THIS PAGE :(

WLUG
COMMUNITY | KNOWLEDGE | SHARE

SOLUTIONS



WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHY VOLUMES OVER BIND MOUNT

Bind Mounts :

- It has been available in Docker since its earliest days for data persisting
- It maps a directory or file on the host to a directory or file within the container
- You must explicitly create a path to the file or folder to place the storage

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WHY VOLUMES OVER BIND MOUNT

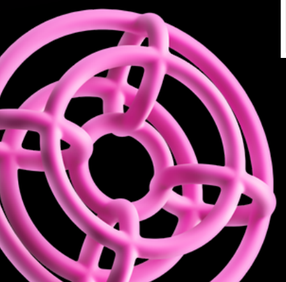
Docker Volumes :

- They are managed by Docker independently of directory structure and host OS
- Volumes can be more safely shared among multiple containers
- Volume drivers enable storing volumes remotely, encrypting contents, or adding additional functionality

WLUG
COMMUNITY | KNOWLEDGE | SHARE

VOLUMES

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



WLUG
COMMUNITY | KNOWLEDGE | SHARE

HOW TO PERSIST DATA AMONG CONTAINERS

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Docker Network

- Default Network
- Create Network
- Connect
- Networking between 2 networks

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Network

- Network : Collection of devices that are connected & can communicate with each other

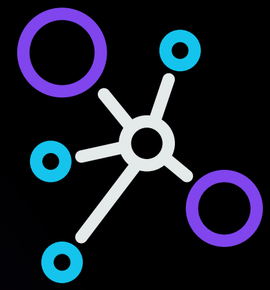
- IP address : A unique numberical label assigned to each device on network

- Gateway : Entry & Exit point for data traffic between different networks

- Subnet : Way to divide larger network into smaller, more manageable parts

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Network example



A

B

C

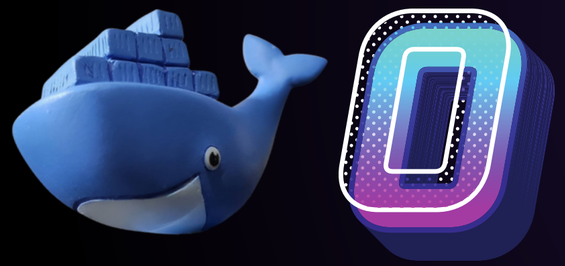
D

WLUG
COMMUNITY | KNOWLEDGE | SHARE

WLUG
COMMUNITY | KNOWLEDGE | SHARE

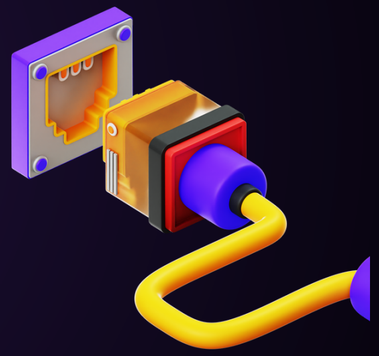
Bridge Network

- Known by many names :
  1. docker 0
  2. Virtual bridge interface
  3. The default bridge



- Communication between Containers on same host

- Connects to virtual ethernet interfaces on host



- Can be a hardware device or a software device

WLUG
COMMUNITY | KNOWLEDGE | SHARE

How docker connects to internet

Ping: To test reachability of host device on internet

Traceroute: Command for displaying possible route from sender’s ip to reciver’s ip

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Network Commands

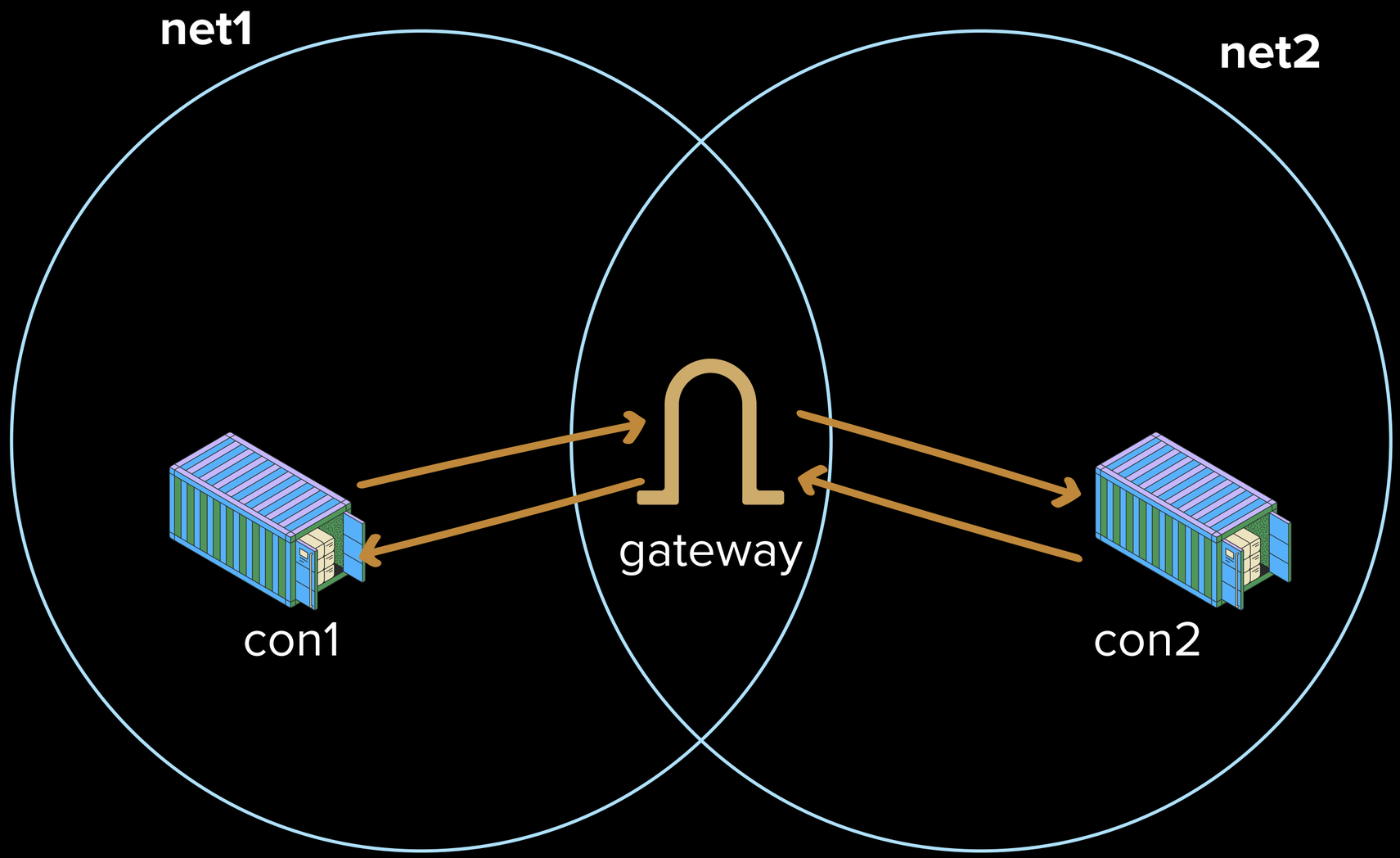
$ docker network create {network_name}	To create a network
$ docker network inspect {network_name}	To inspect the network
$ docker network connect {network_name} {container_name}	To connect container to a network
$ docker network disconnect {network_name} {container_name}	To disconnect container from a network

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Hands On!!

WLUG
COMMUNITY | KNOWLEDGE | SHARE

Our Goal

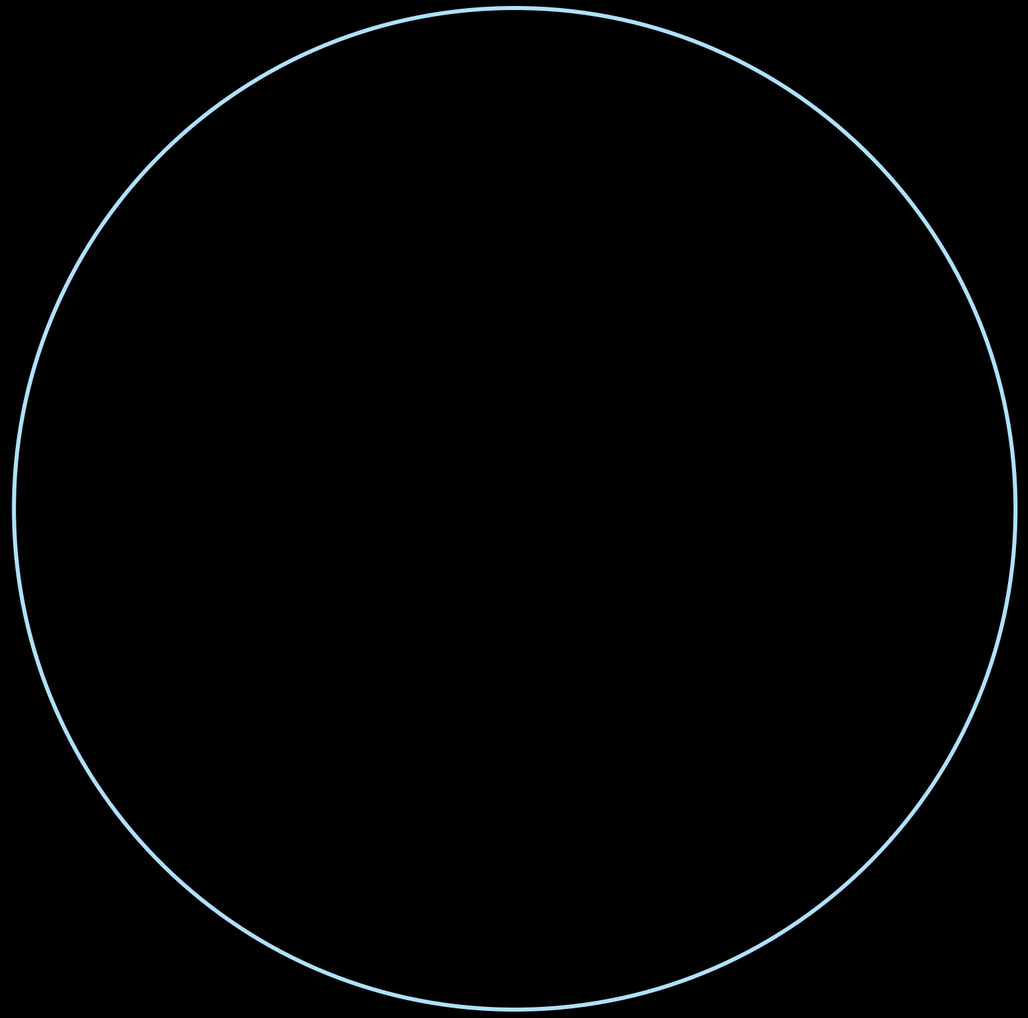


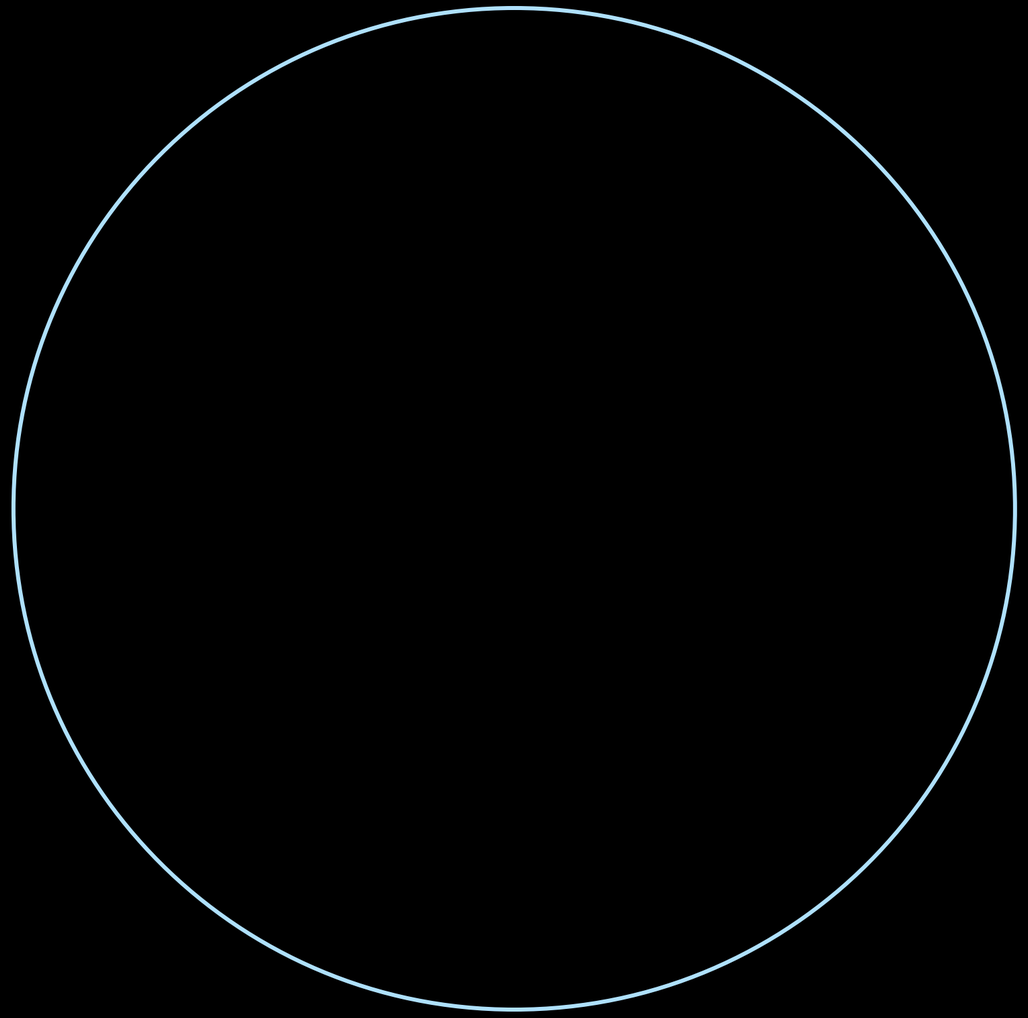
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Setting up the network

net1 : 20.0.0.0/24

net2 : 20.0.1.0/24

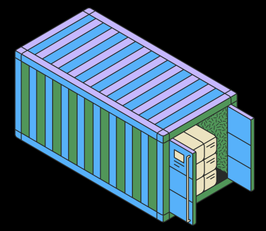




WLUG
COMMUNITY | KNOWLEDGE | SHARE

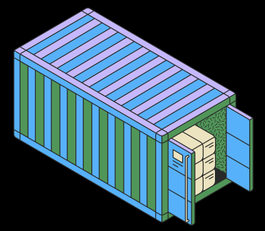
Adding con to net

net1 : 20.0.0.0/24



con1
20.0.0.2

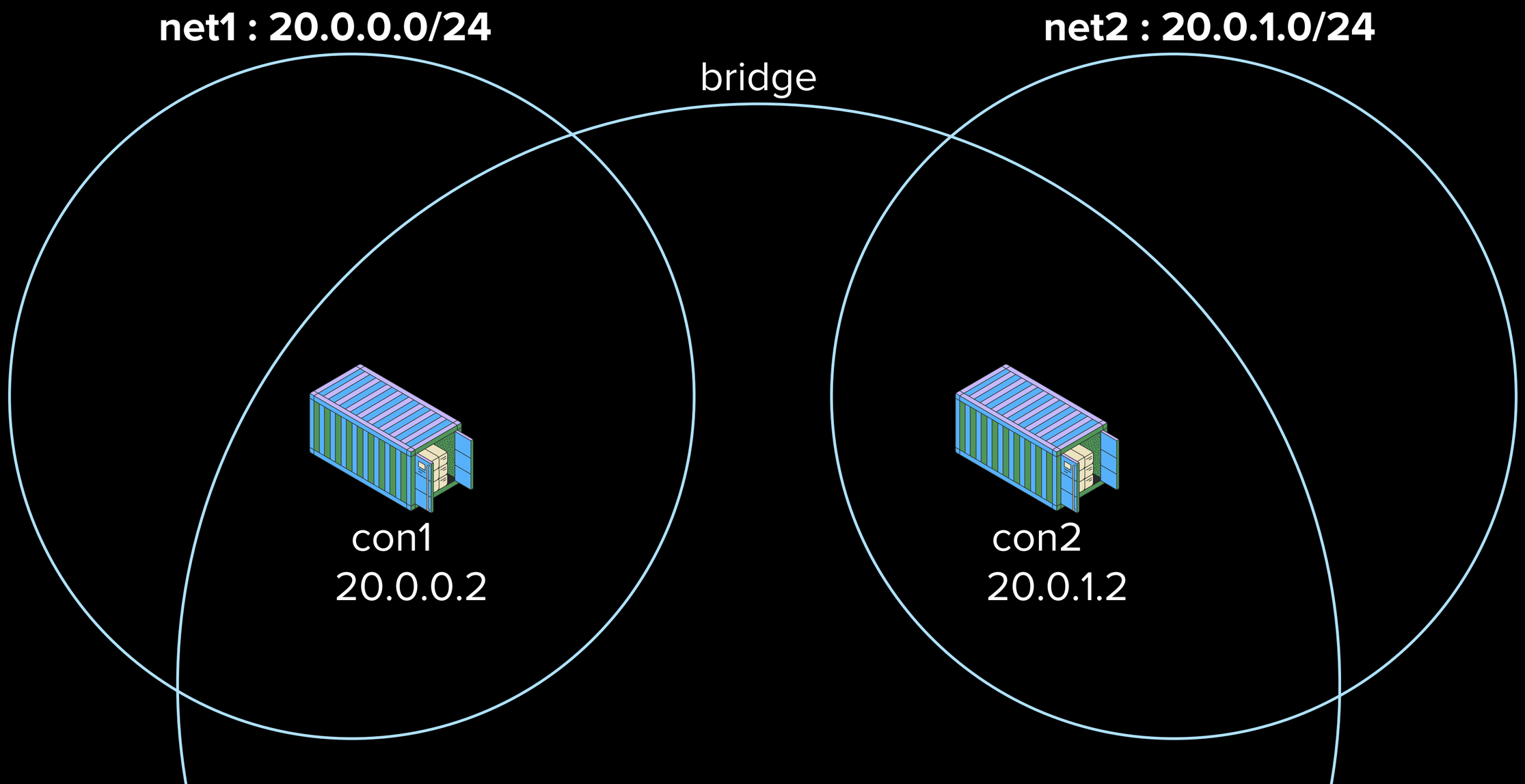
net2 : 20.0.1.0/24



con2
20.0.1.2

WLUG
COMMUNITY | KNOWLEDGE | SHARE

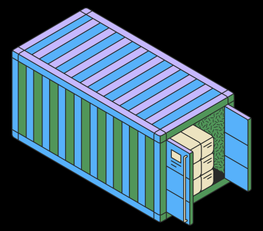
Removing default net



WLUG
COMMUNITY | KNOWLEDGE | SHARE

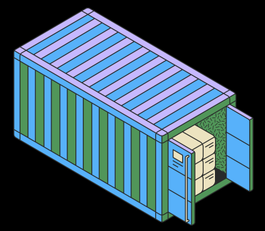
Now truly isolated

net1 : 20.0.0.0/24



con1
20.0.0.2

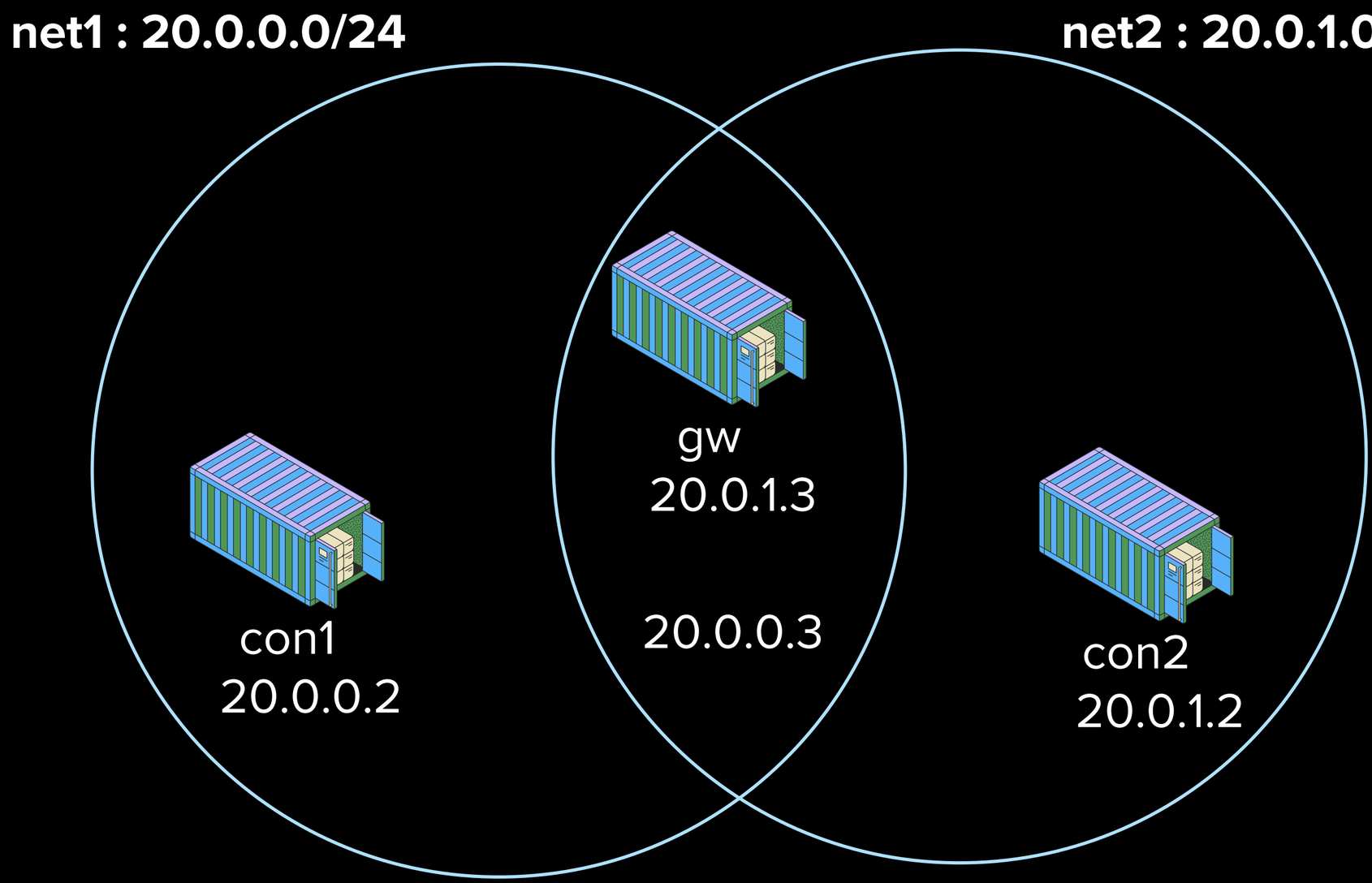
net2 : 20.0.1.0/24



con2
20.0.1.2

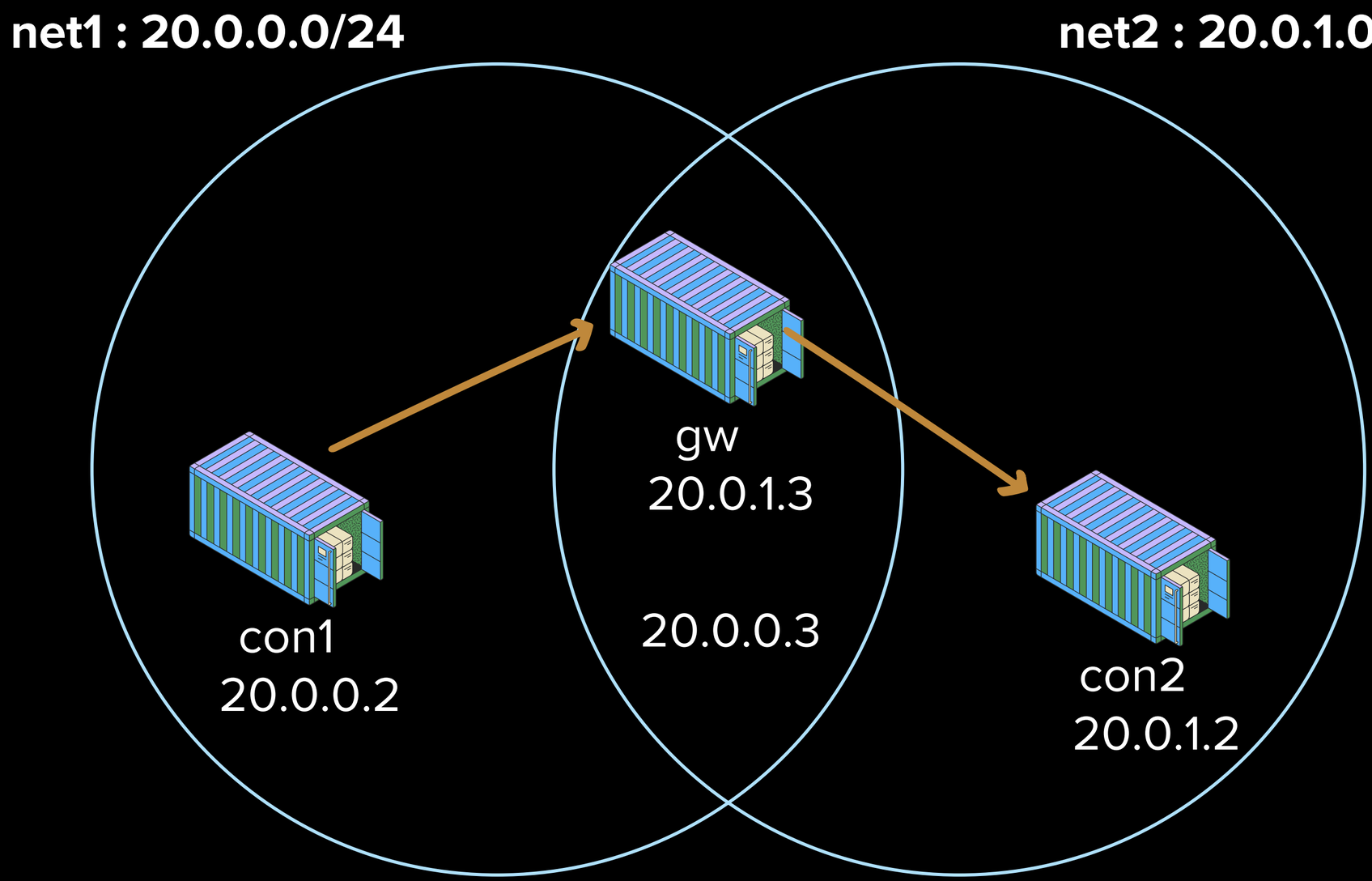
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Gateway



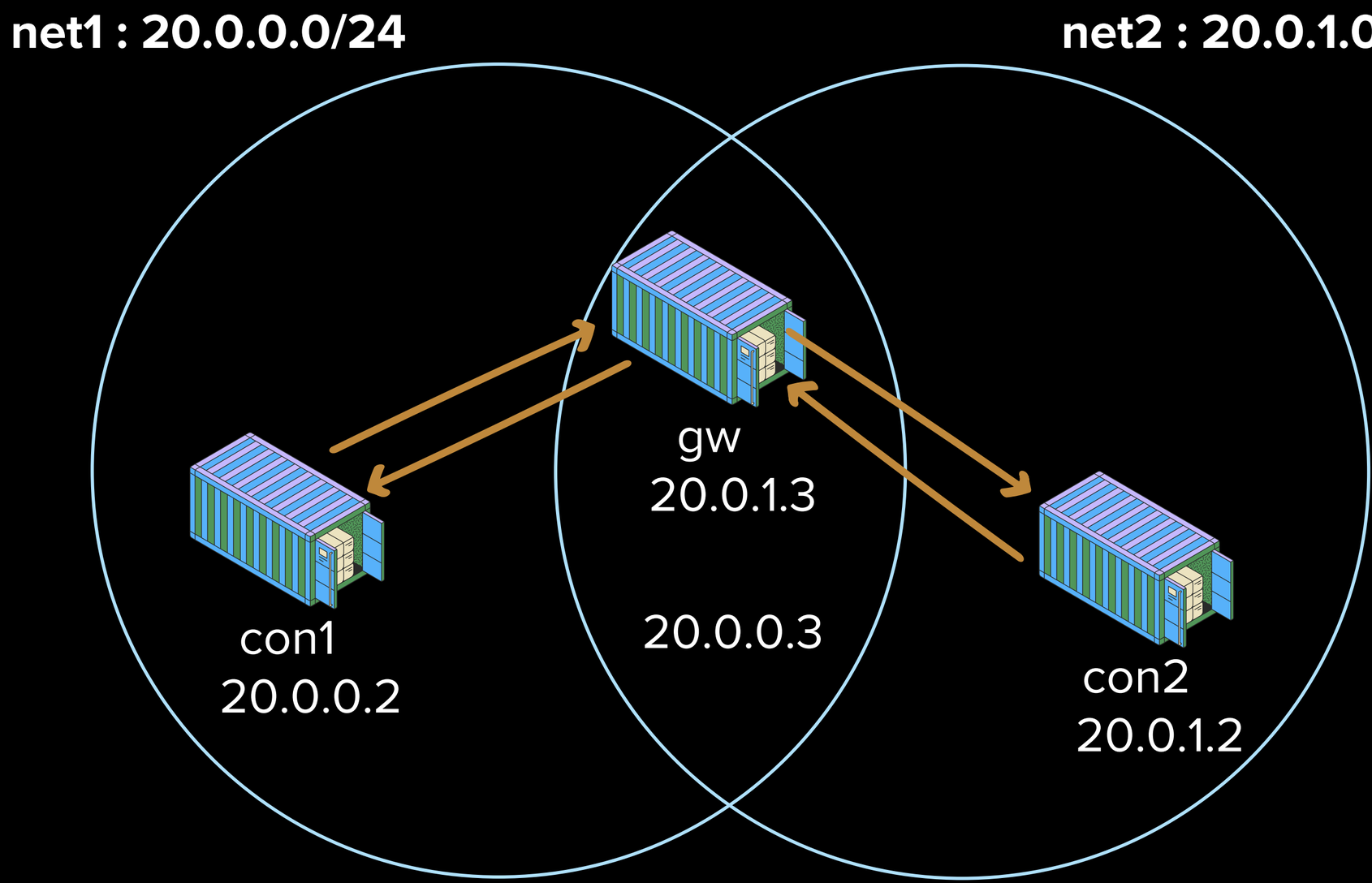
WLUG
COMMUNITY | KNOWLEDGE | SHARE

Routing

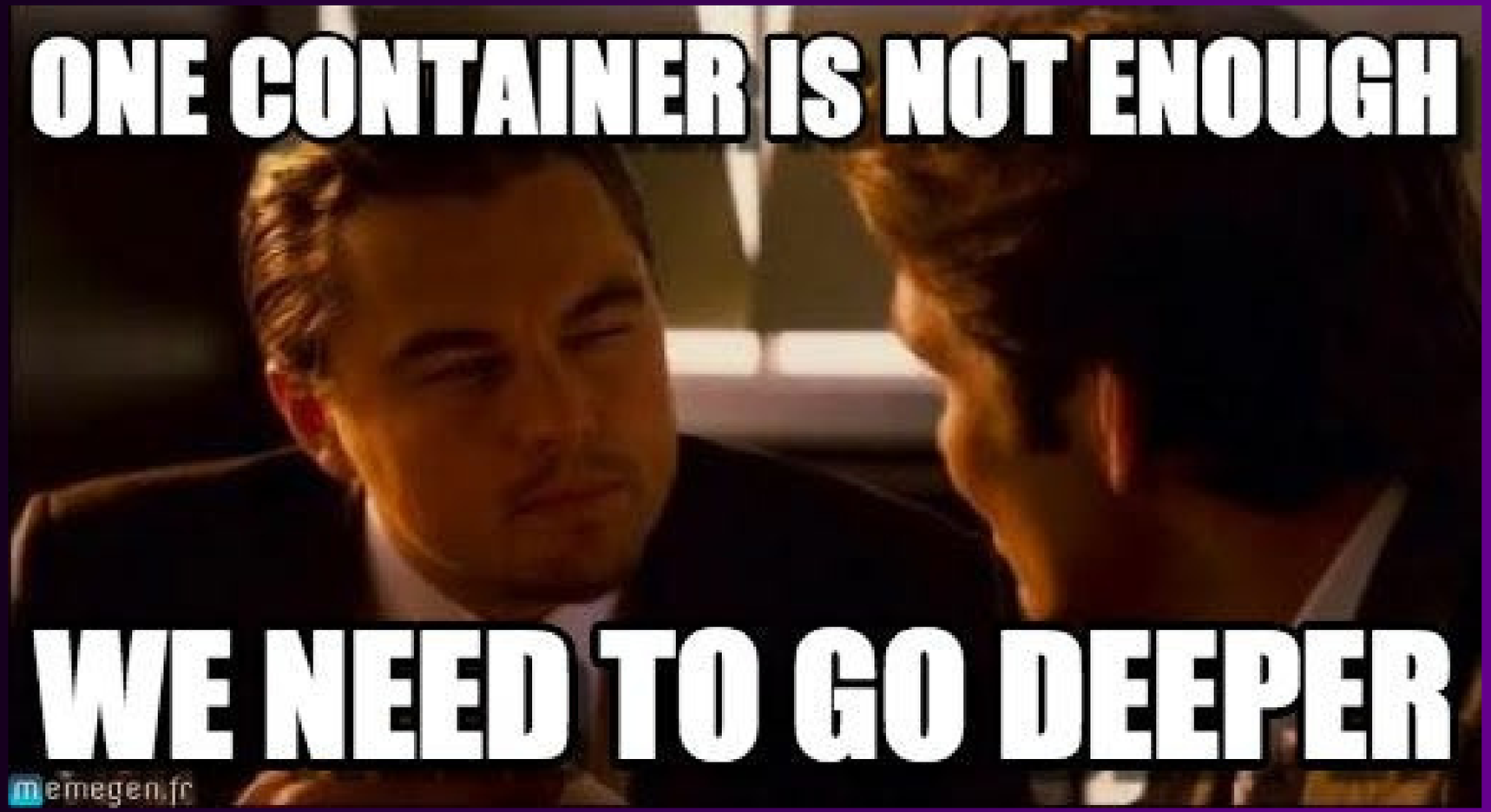


WLUG
COMMUNITY | KNOWLEDGE | SHARE

Goal Reached !!



WLUG
COMMUNITY | KNOWLEDGE | SHARE



WLUG
COMMUNITY | KNOWLEDGE | SHARE

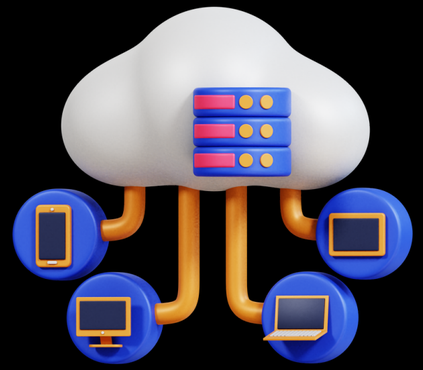
Docker Compose

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS

Frontend/UI

Database



Server

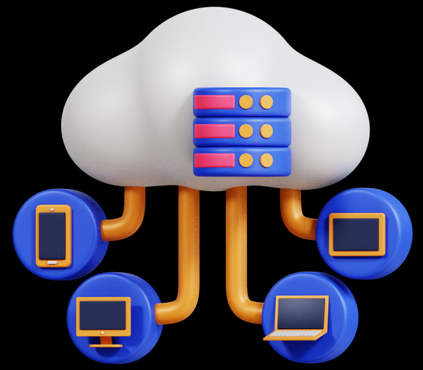
Network

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS

Frontend/UI

Database



Server

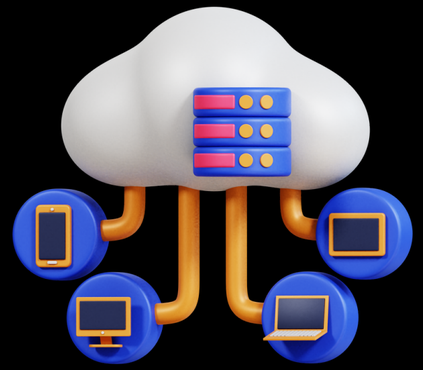
Network

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS

Frontend/UI

Database



Server

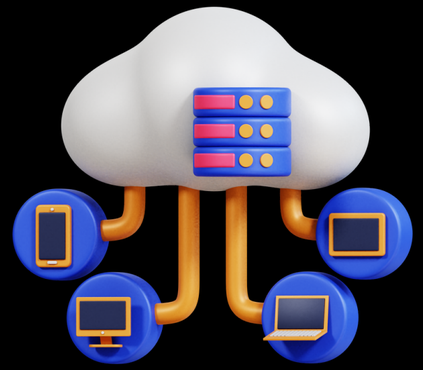
Network

WLUG
COMMUNITY | KNOWLEDGE | SHARE

MULTI-CONTAINER APPLICATIONS

Frontend/UI

Database

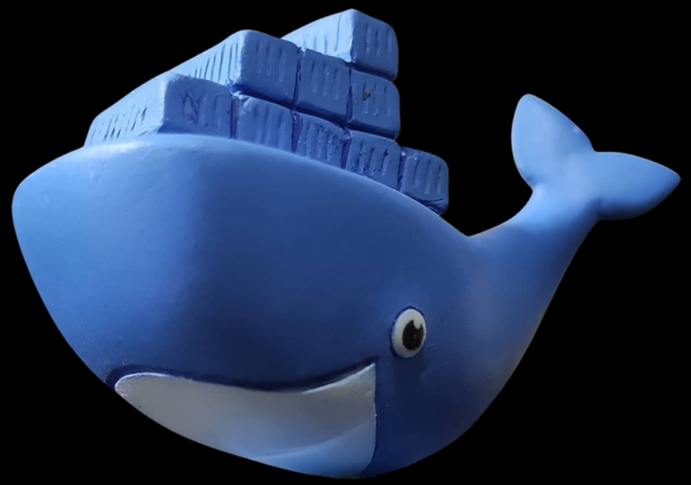


Server

Network

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER-COMPOSE

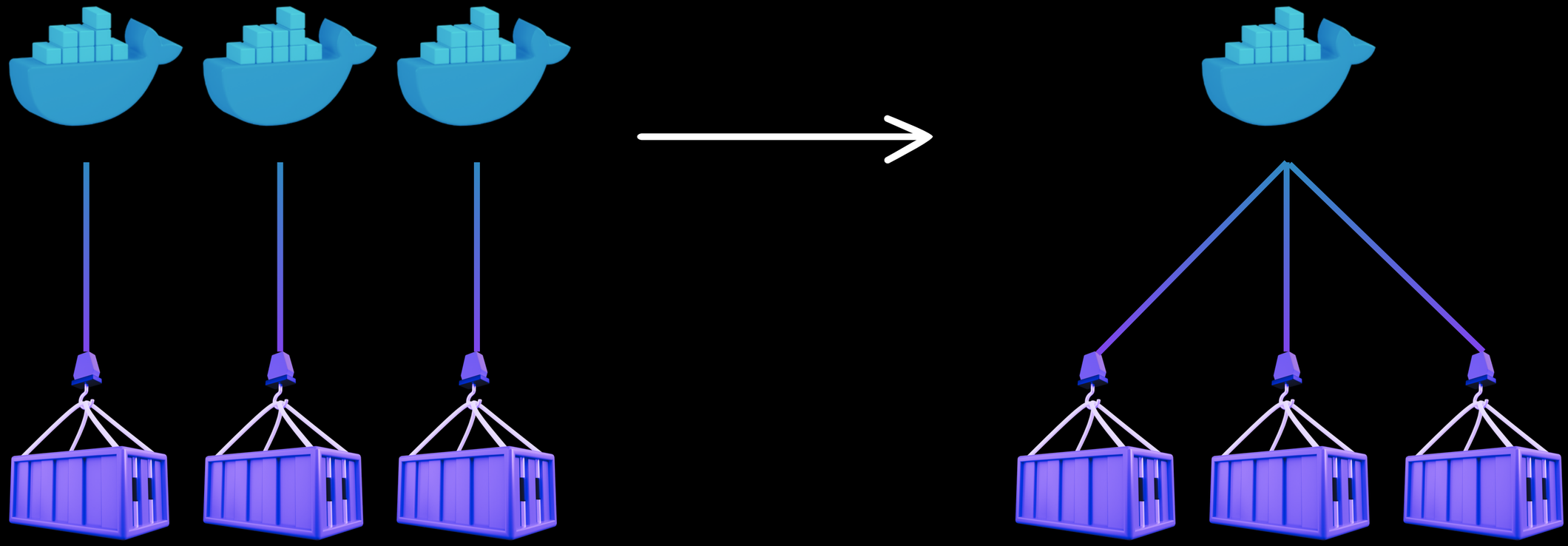


- A tool for defining and running multi-container Docker applications
- It allows us to define a set of services that make up our application and run them all together in a single command

WLUG
COMMUNITY | KNOWLEDGE | SHARE

DOCKER

DOCKER COMPOSE

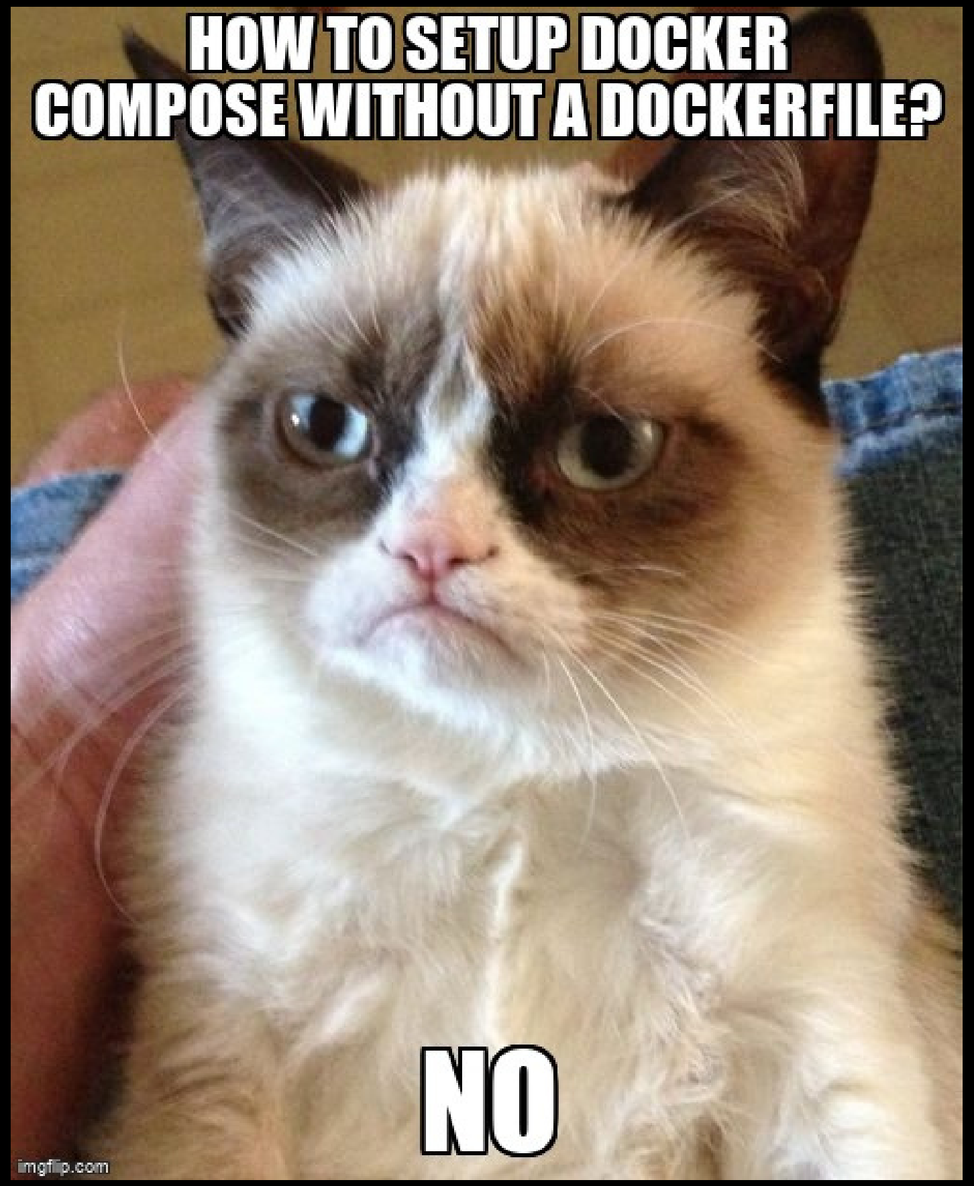


WLUG
COMMUNITY | KNOWLEDGE | SHARE

INSTALLATION

- sudo apt-get install docker-compose-plugin

WLUG
COMMUNITY | KNOWLEDGE | SHARE



WLUG
COMMUNITY | KNOWLEDGE | SHARE

How does this works?

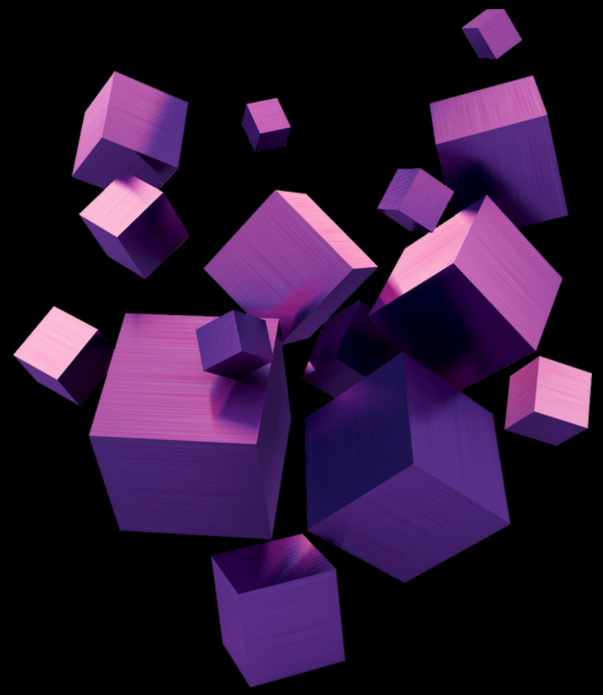
```yaml
version: '3.7'

services:

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

    service2:
        image: image_name
        volume:
           - /path/to/host:/path/to/container
        networks:
           - networkName
        command: command
        depends_on: service1

    service3:
      ...
```

Create docker-compose.yaml file

- Version
- Services
- Image
- Volume
- Networks

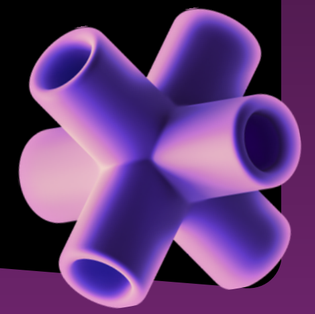
WLUG
COMMUNITY | KNOWLEDGE | SHARE

How does this works?



Run docker-compose.yaml

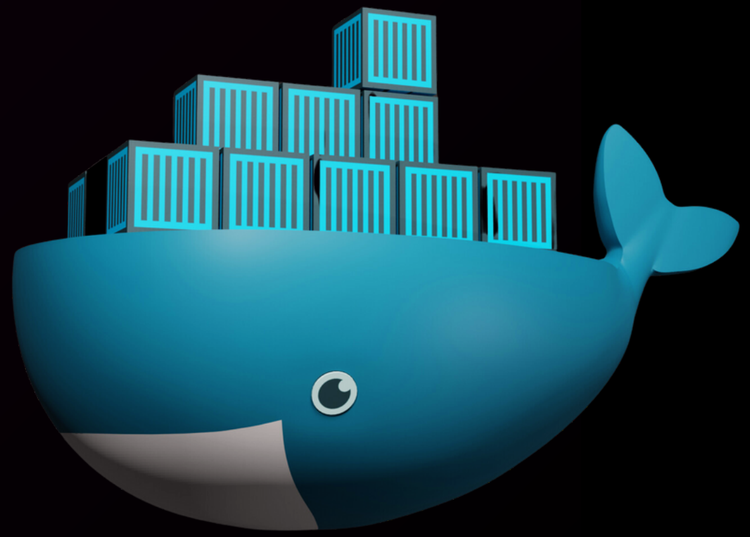
- Start Compose
  $ sudo docker compose up
- Stop Compose
  $ sudo docker compose down



WLUG
COMMUNITY | KNOWLEDGE | SHARE

Why Docker-Compose?

- Ability to configure, start and handle multiple containers at the same time
- Easy and fast management
- Multi-tasking
- Startup dependencies between containers



WLUG

COMMUNITY | KNOWLEDGE | SHARE

Thank You!