



WALCHAND COLLEGE OF ENGINEERING, SANGLI

WALCHAND LINUX USERS' GROUP



METAMORPHOSIS

LIMITED SEATS

Navigating the DevOps Wave

EXCITING PRIZES



Let's Dock It!

SESSION 1

DockerVerse

SESSION 2

15 & 16 FEB



REGISTER AT

Sailing to K8s

SESSION 3

Pod Power

SESSION 4

WARGAMES

POWERED BY
IT'S FOSS

CONTACT WITH US

₹349 | ₹299 each

Main & Mini CCF

94220 17108
82084 32514

Mr. Yash Patil
President
WLUG

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

Dr. R. R. Rathod
HoD Information
Technology

Dr. A. J. Umbarkar
Staff Advisor
WLUG

Dr. A. R. Surve
Staff Advisor
WLUG

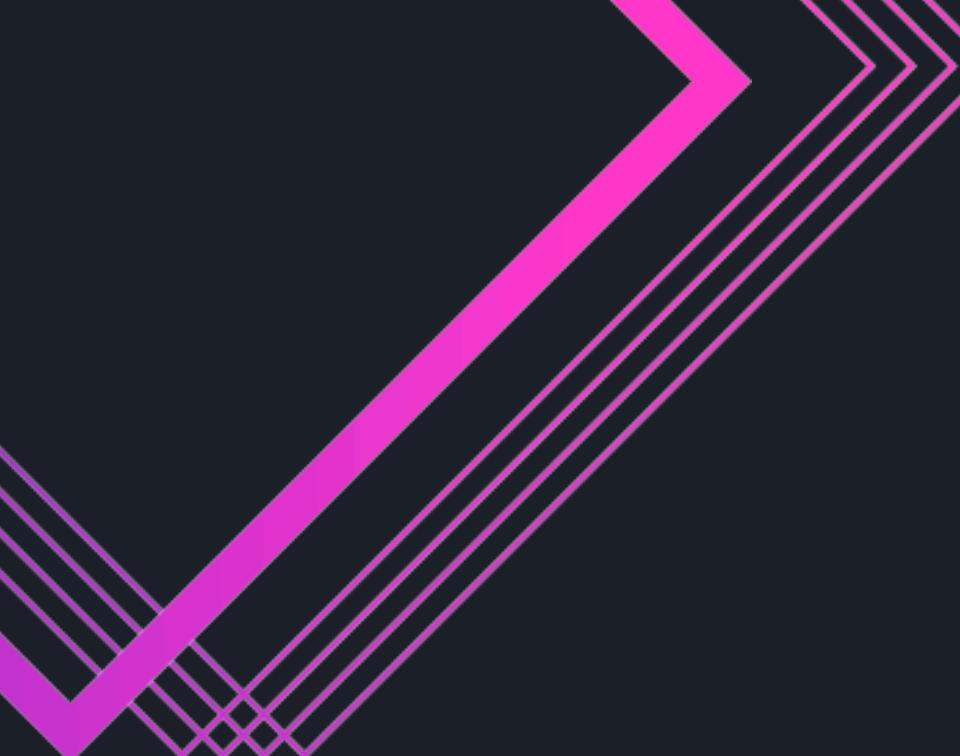
Dr. M. M. Khot
Associate Dean
WCE

Dr. U. A. Dabade
I/C Director
WCE

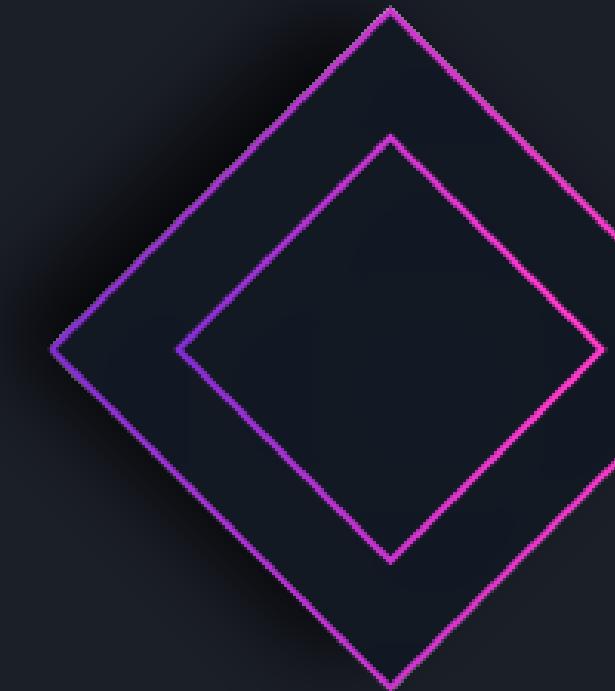
CONTENTS

- ❖ Docker Data Management
- ❖ Docker Volumes
- ❖ Basics of Networking
- ❖ Docker Network
- ❖ Docker Compose



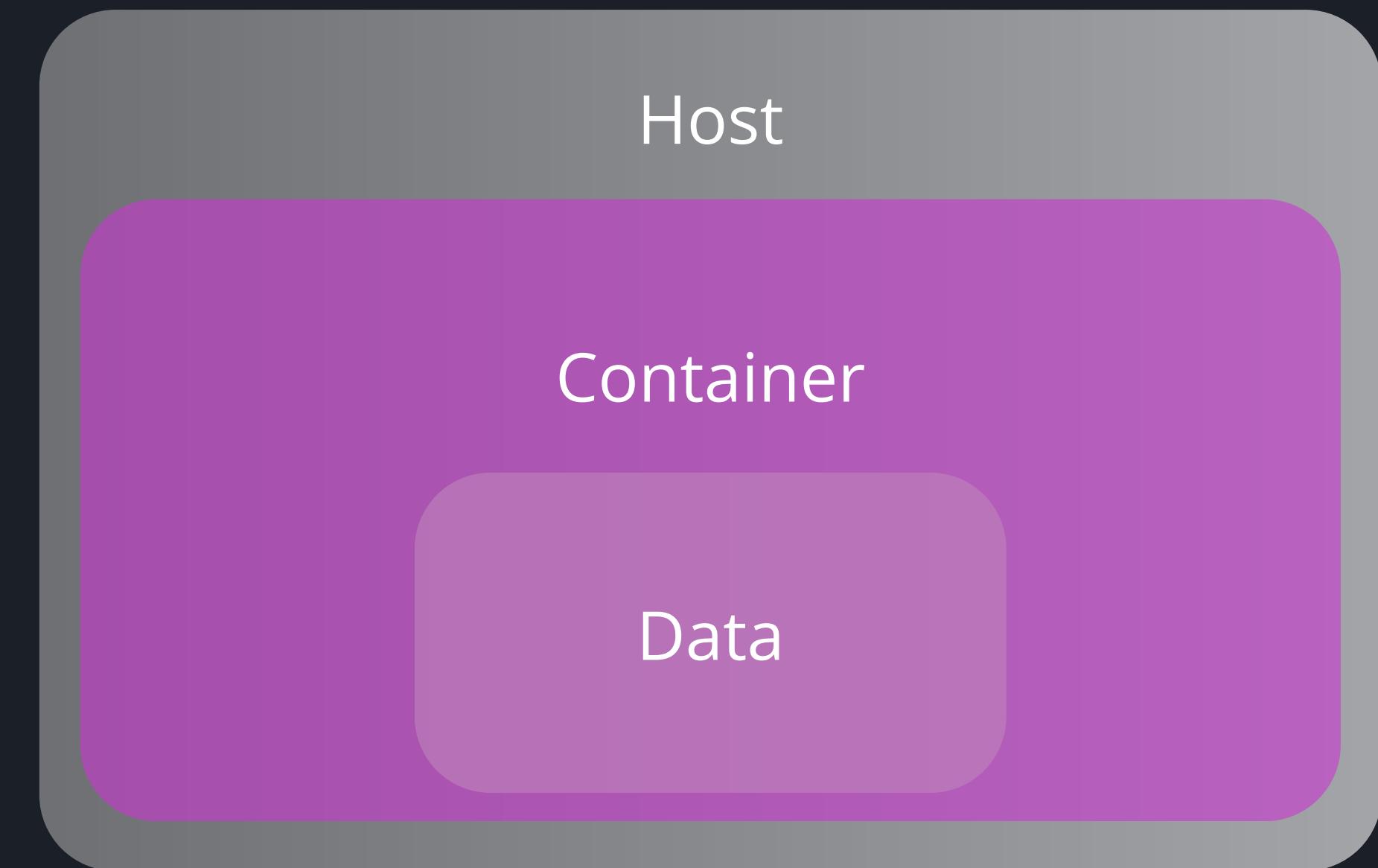


DOCKER DATA MANAGEMENT



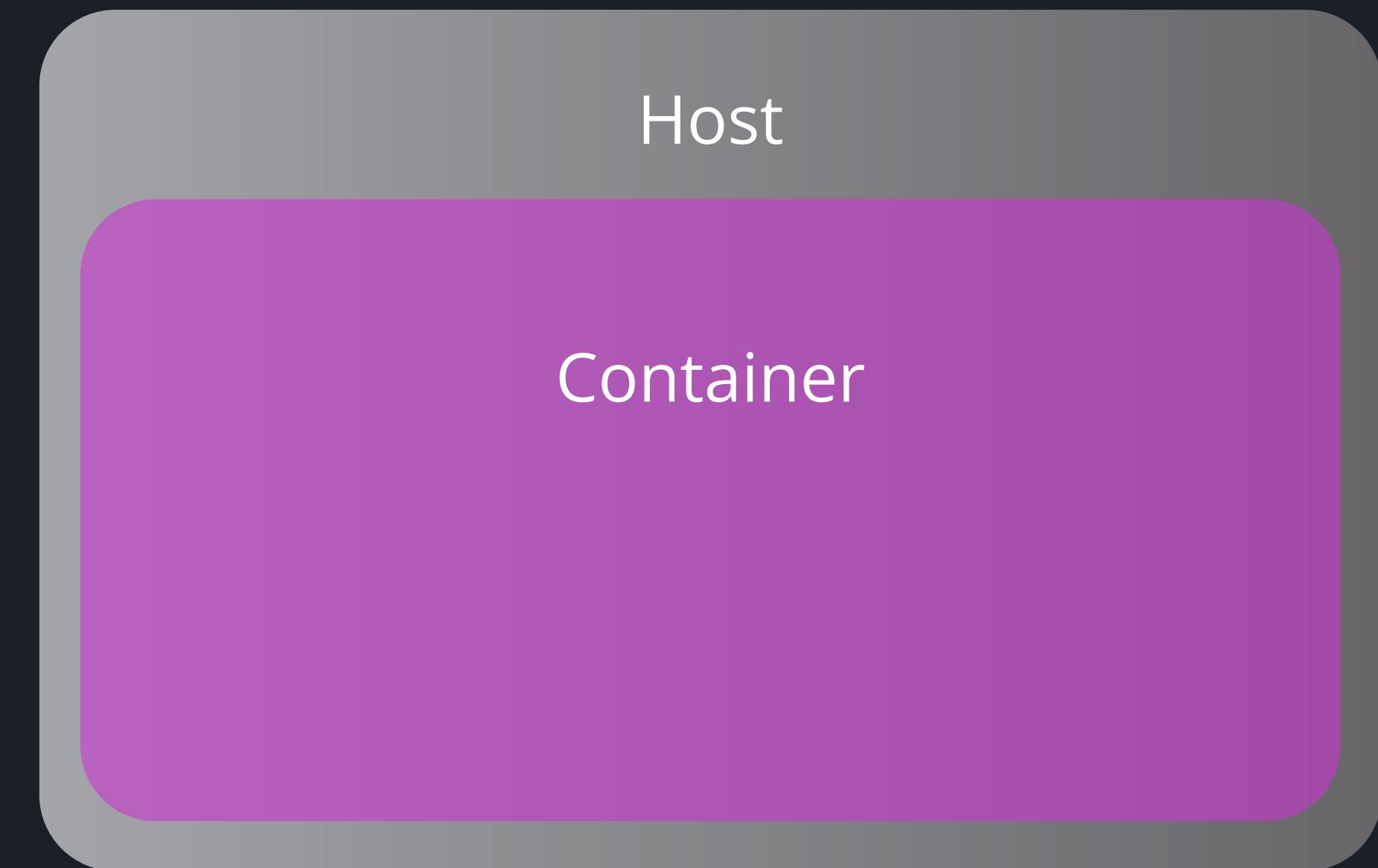
DOCKER DATA

When the container is created
and data is inserted

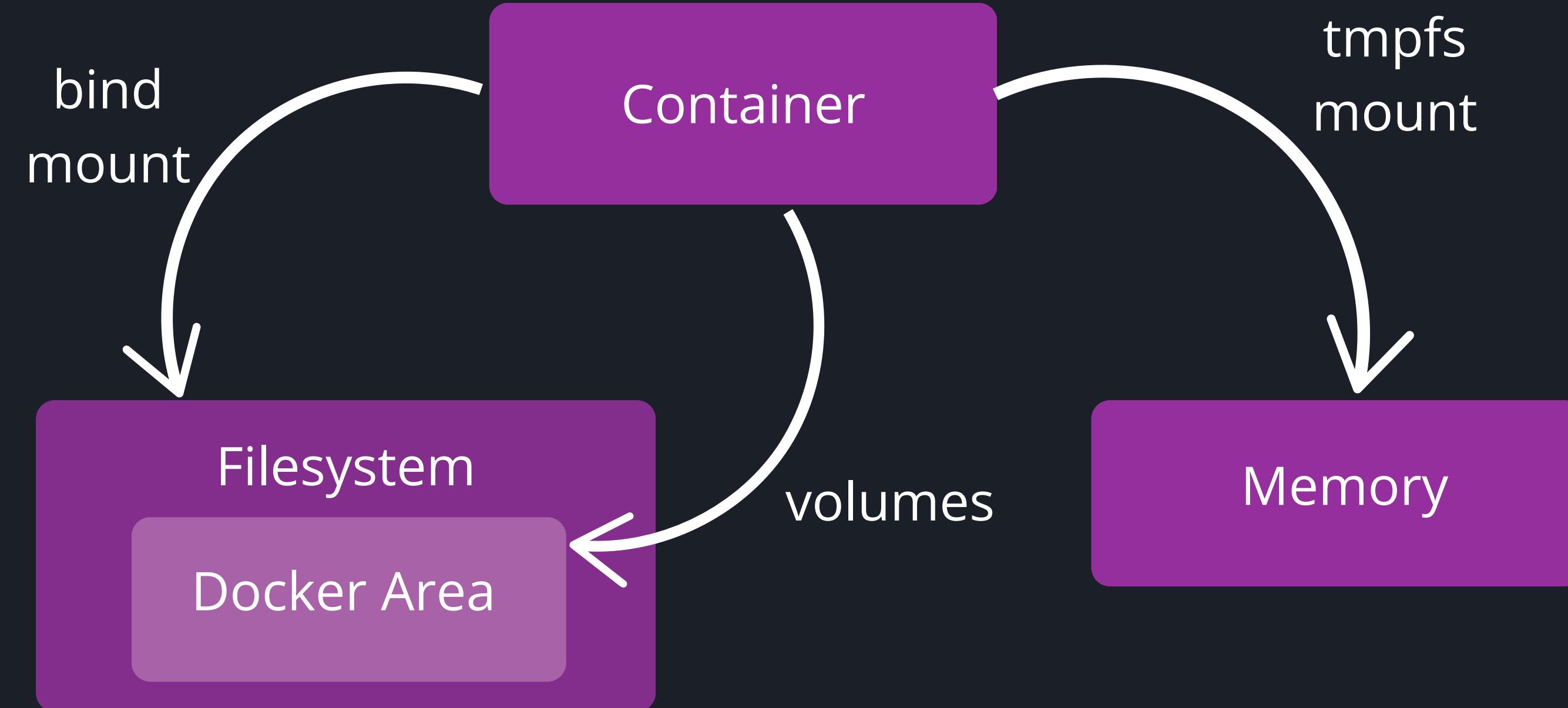


DOCKER DATA

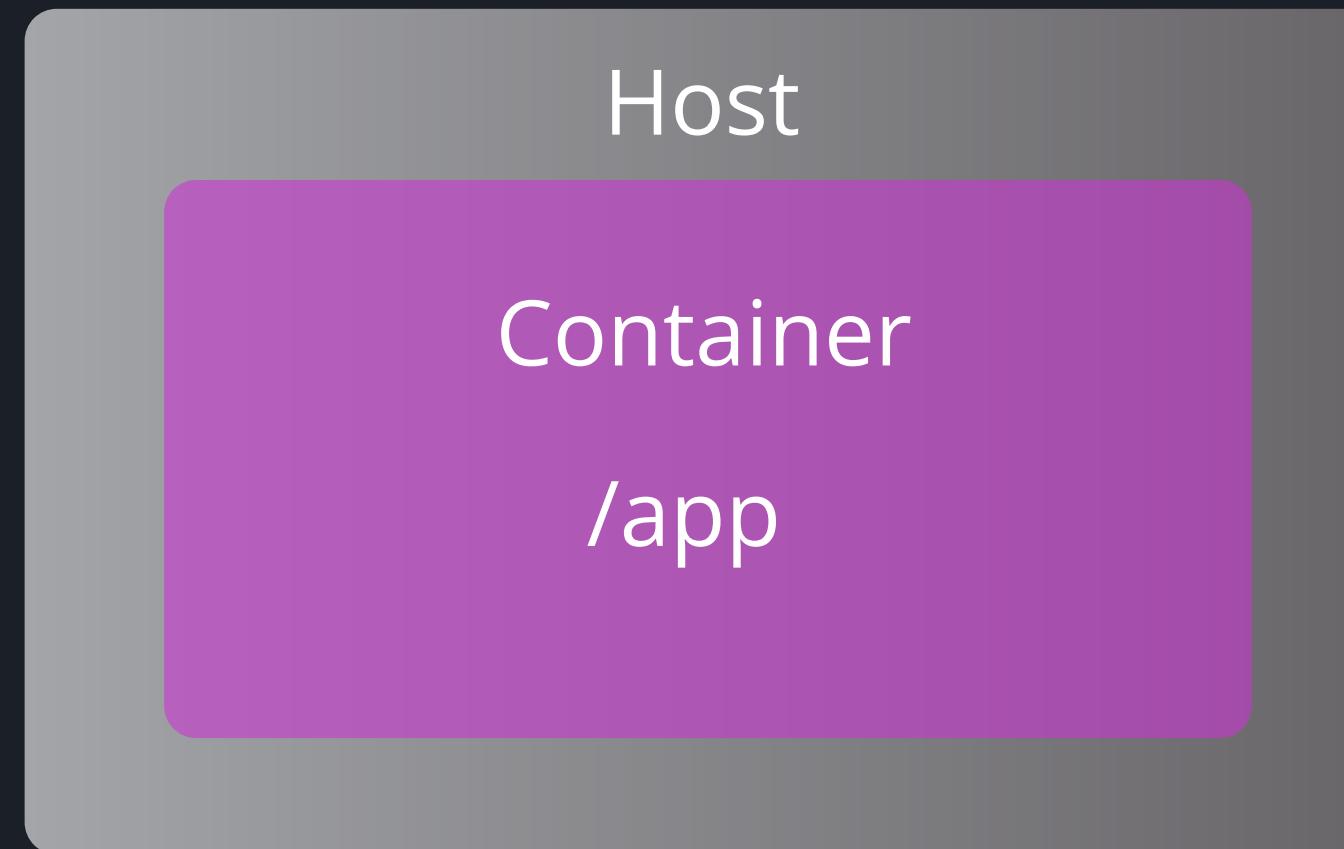
Deleting the container results
in data loss



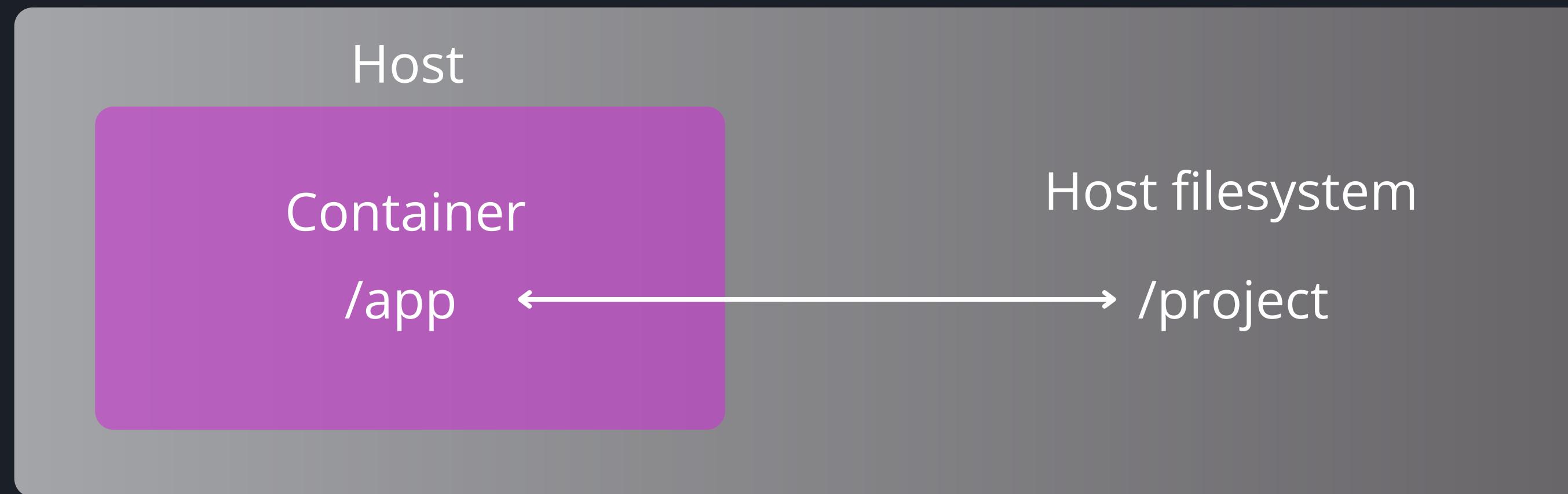
SOLUTIONS



BIND MOUNTS



BIND MOUNTS



BIND MOUNTS

- ❖ When you use a bind mount, a file or directory on the host machine is mounted from the host into a container
- ❖ Bind mounts are stored anywhere on the host system
- ❖ Rely on host file System
- ❖ Not managed by docker

VOLUMES

- ◆ Volumes are persistent data stores for containers
- ◆ Docker manages them and the data resides on the host filesystem, typically under
/var/lib/docker/volumes/vol_name/_data

DOCKER VOLUMES

Create

```
docker volume create <volume_name>
```

List

```
docker volume ls
```

Inspect

```
docker volume inspect <volume_name>
```

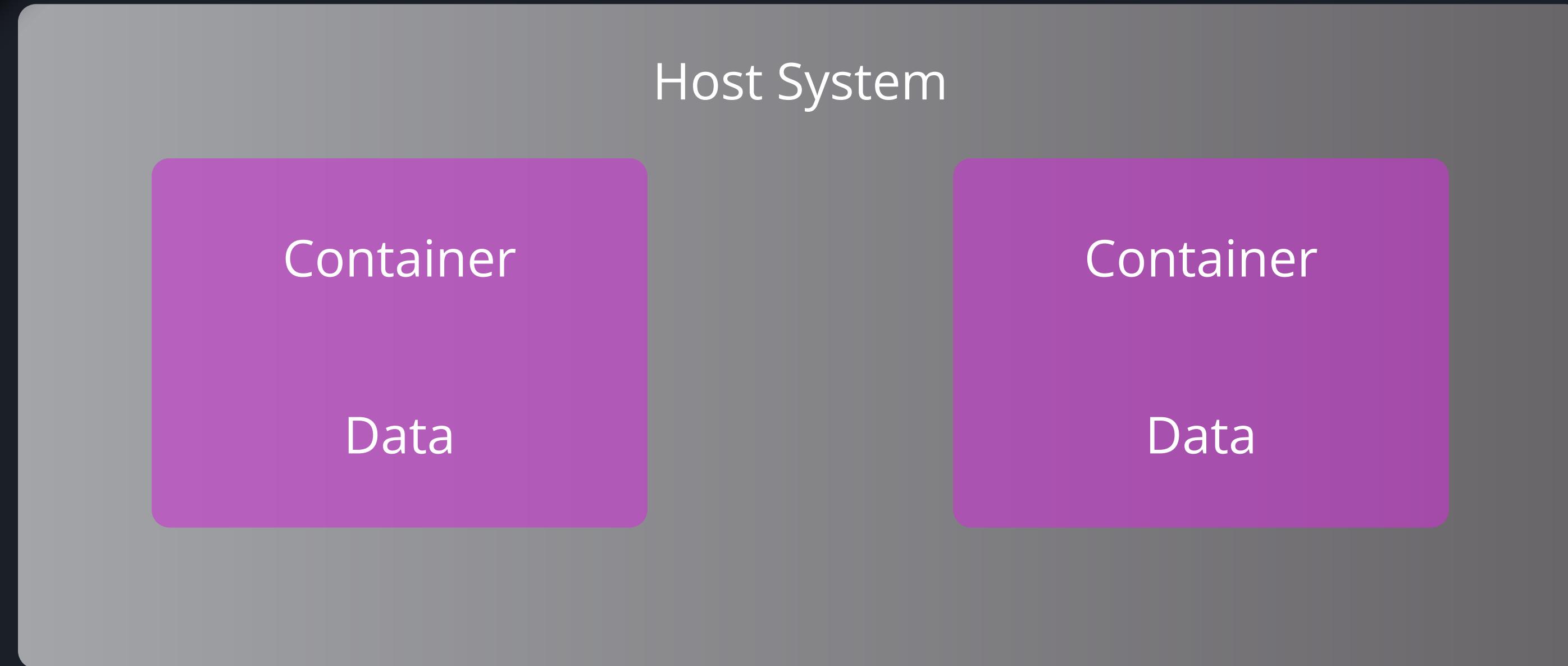
Remove

```
docker volume rm <volume_name>
```

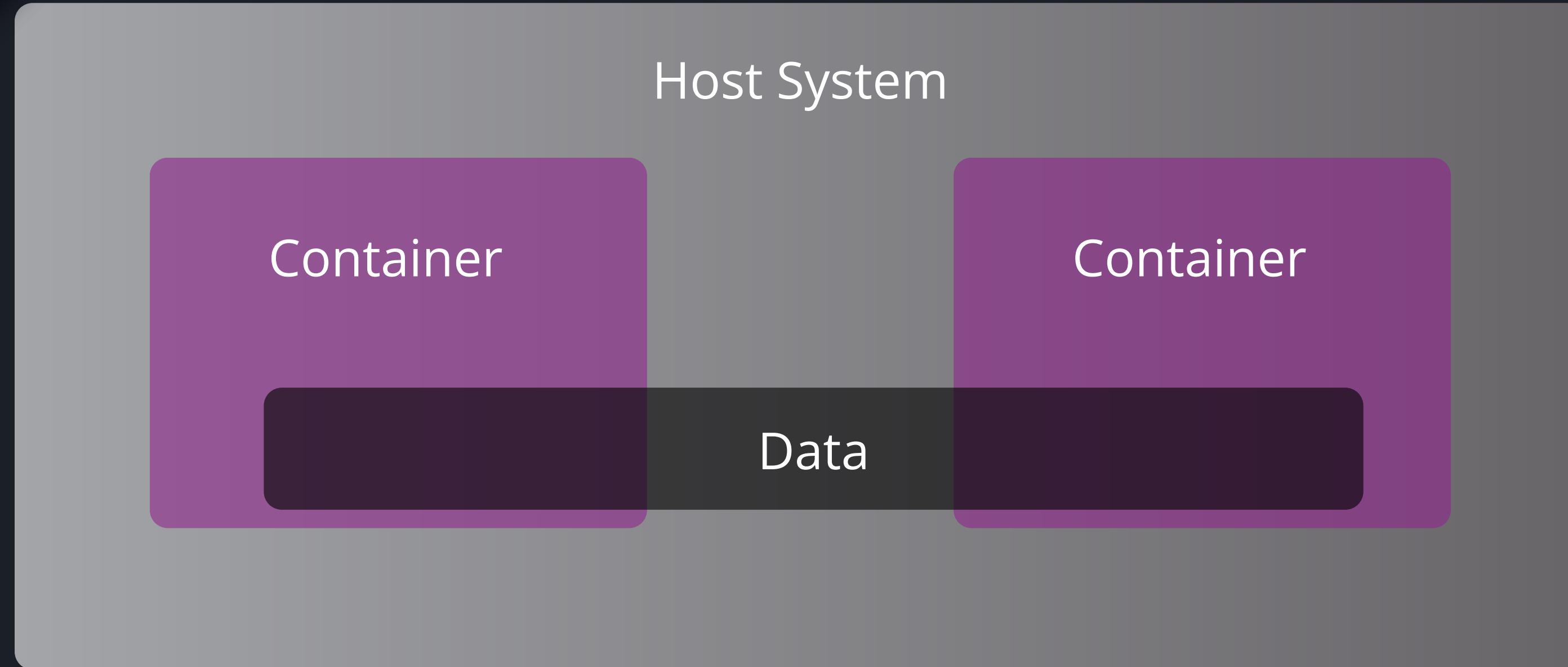
Prune

```
docker volume prune
```

SHARED VOLUMES



SHARED VOLUMES



SHARED VOLUMES

- ❖ Centralized data sharing between containers
- ❖ Ensures data consistency across multiple containers
- ❖ Changes made by one container are instantly visible to others

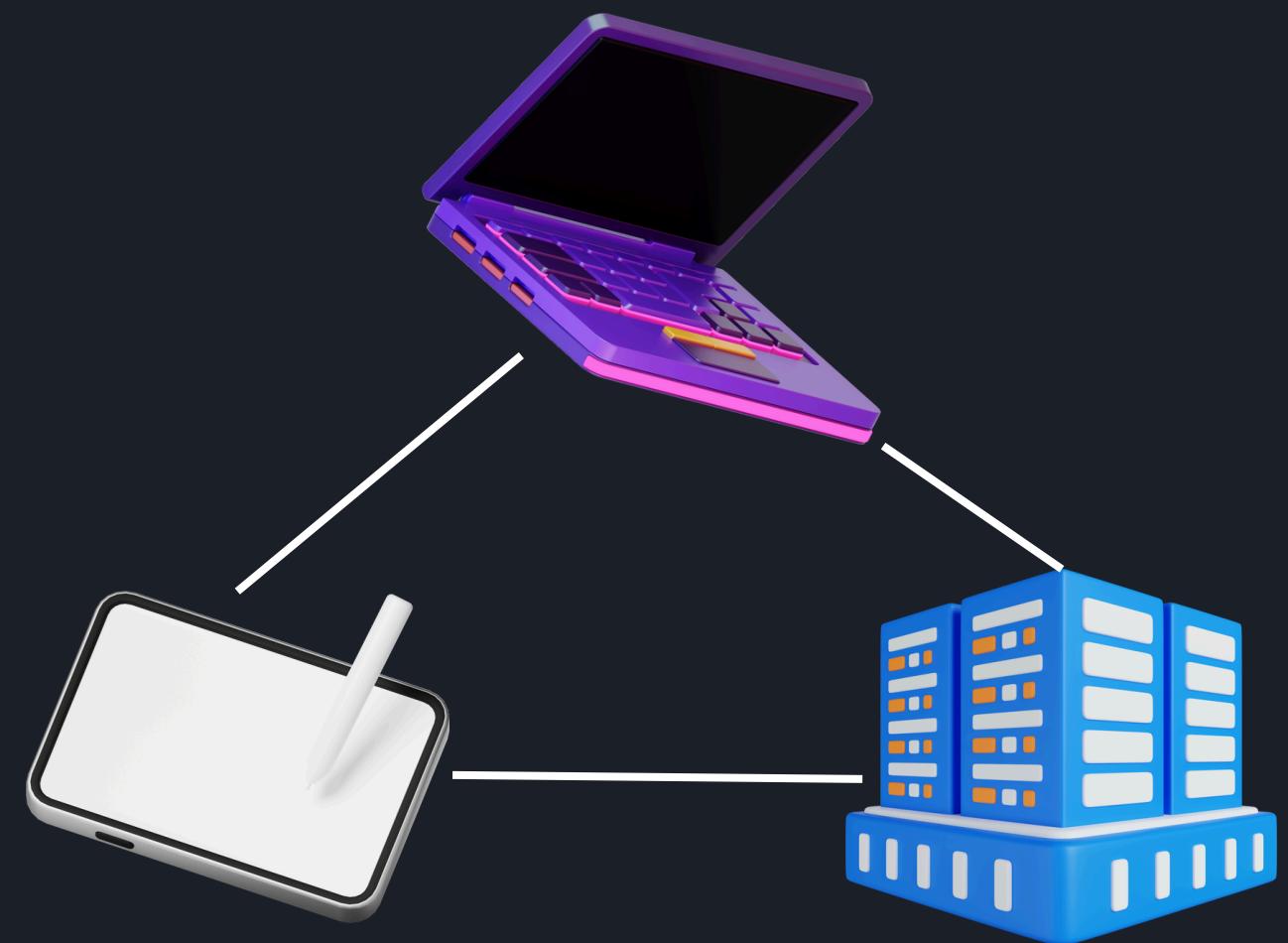


NETWORKING



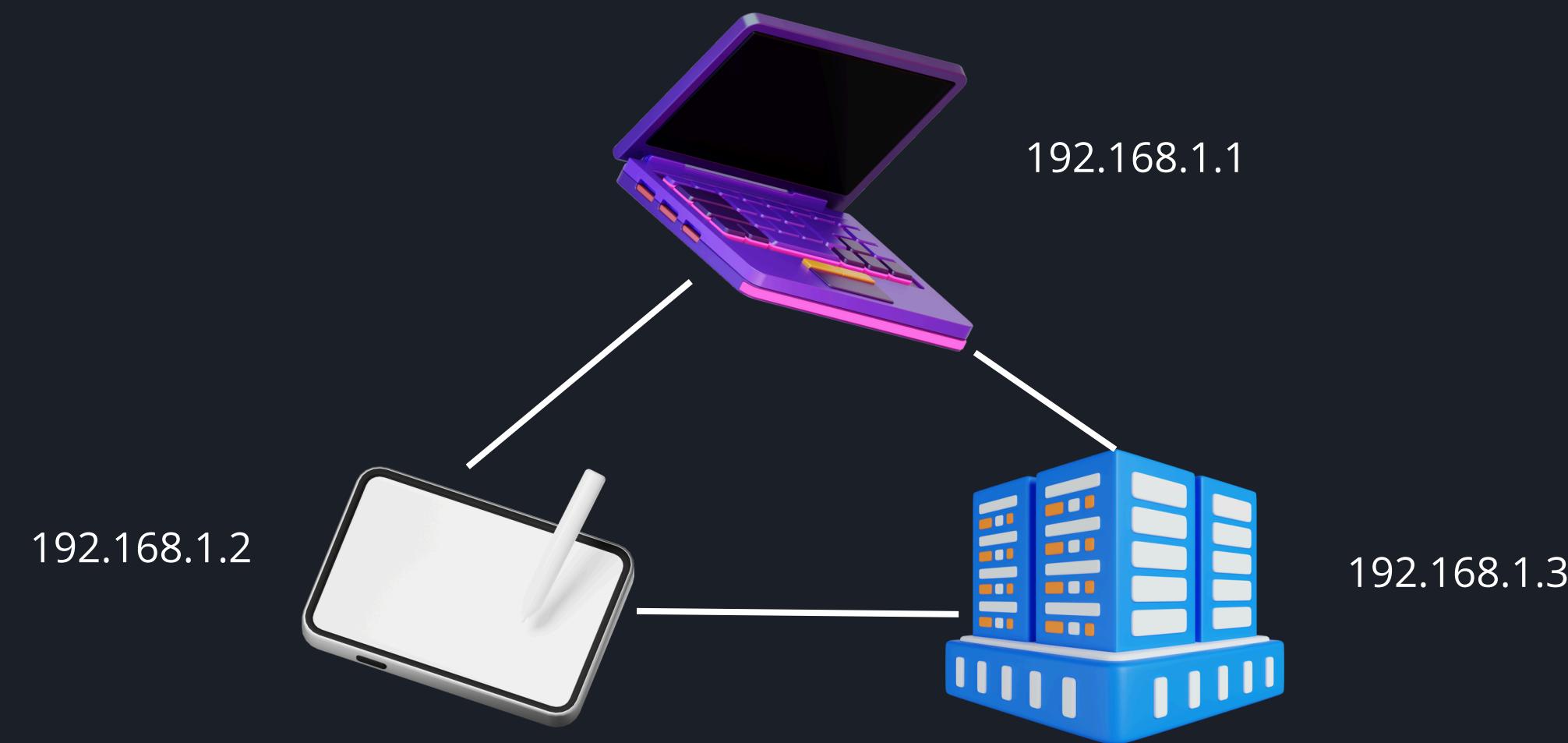
NETWORK

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



IP ADDRESS

A unique numerical label assigned to each device on network



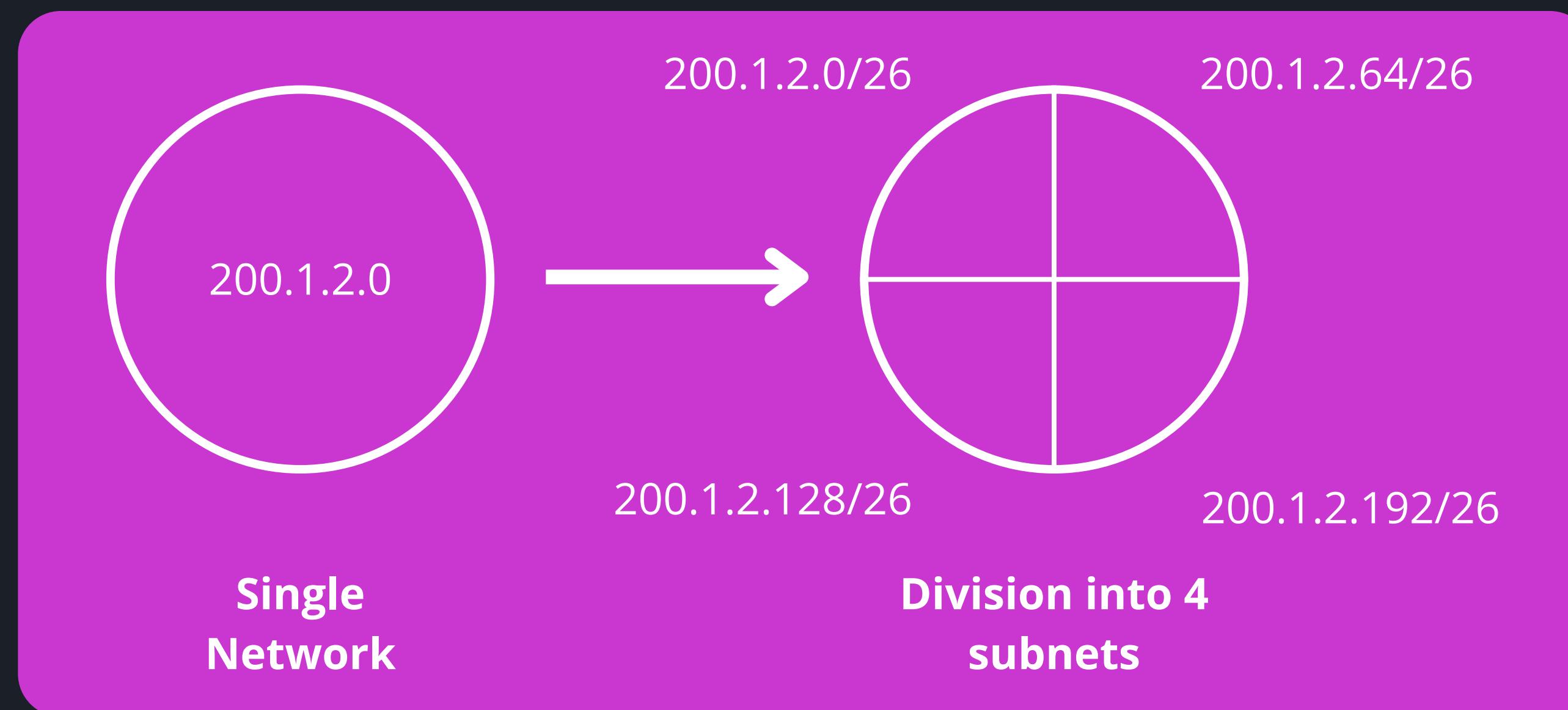
GATEWAY

Entry & exit point for data traffic between different networks



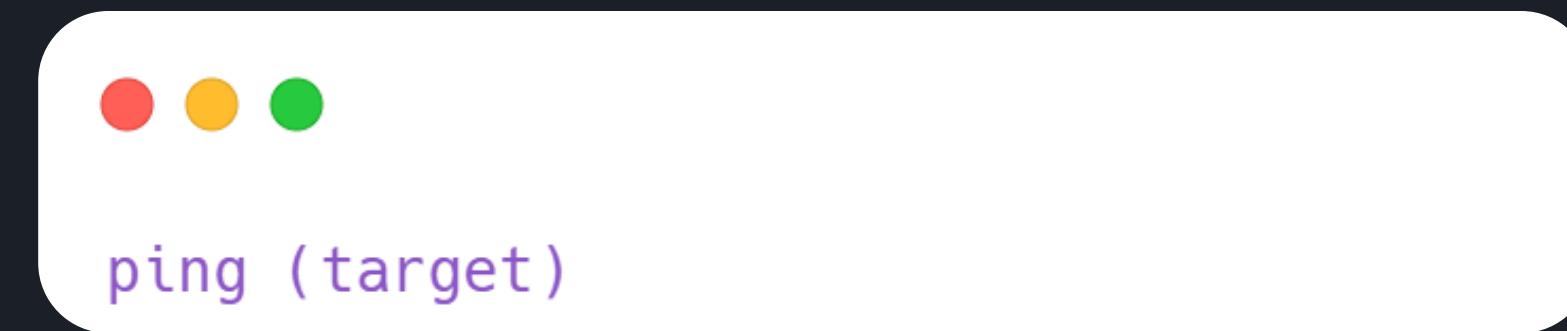
SUBNET

Way to divide larger network into smaller, more manageable parts

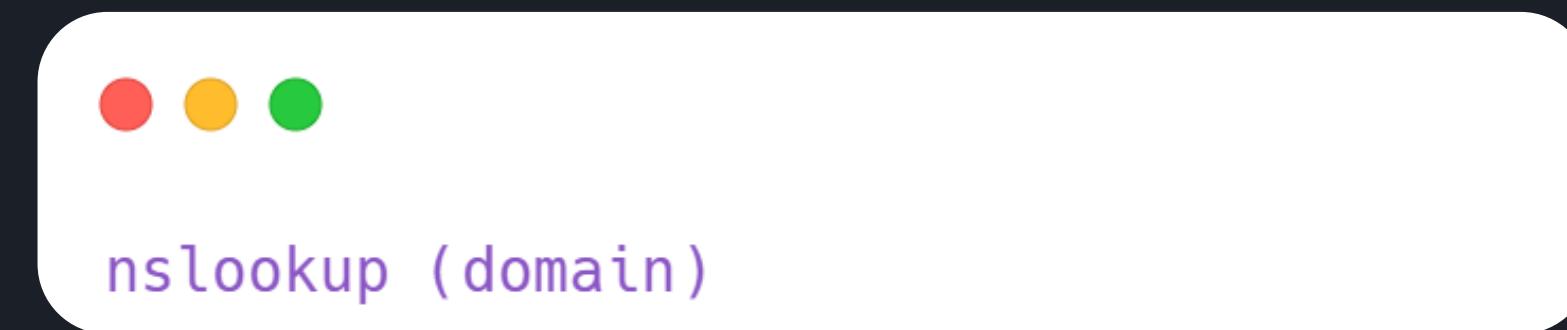


COMMANDS

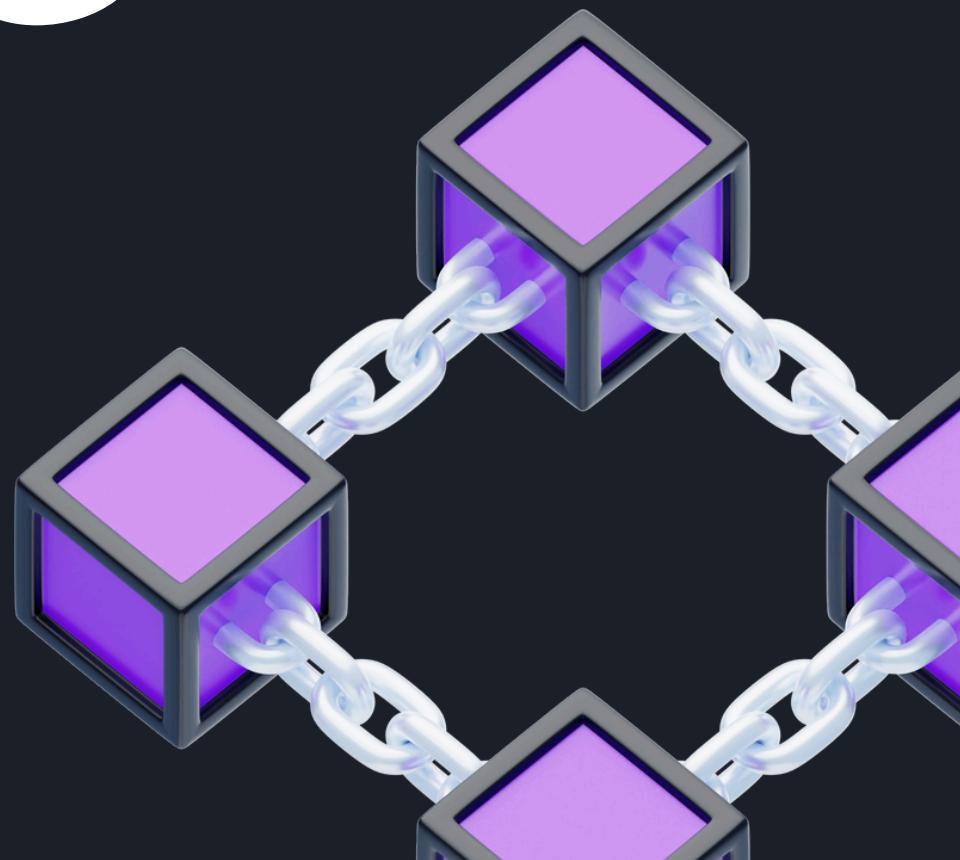
- ❖ Ping - checks connectivity between devices

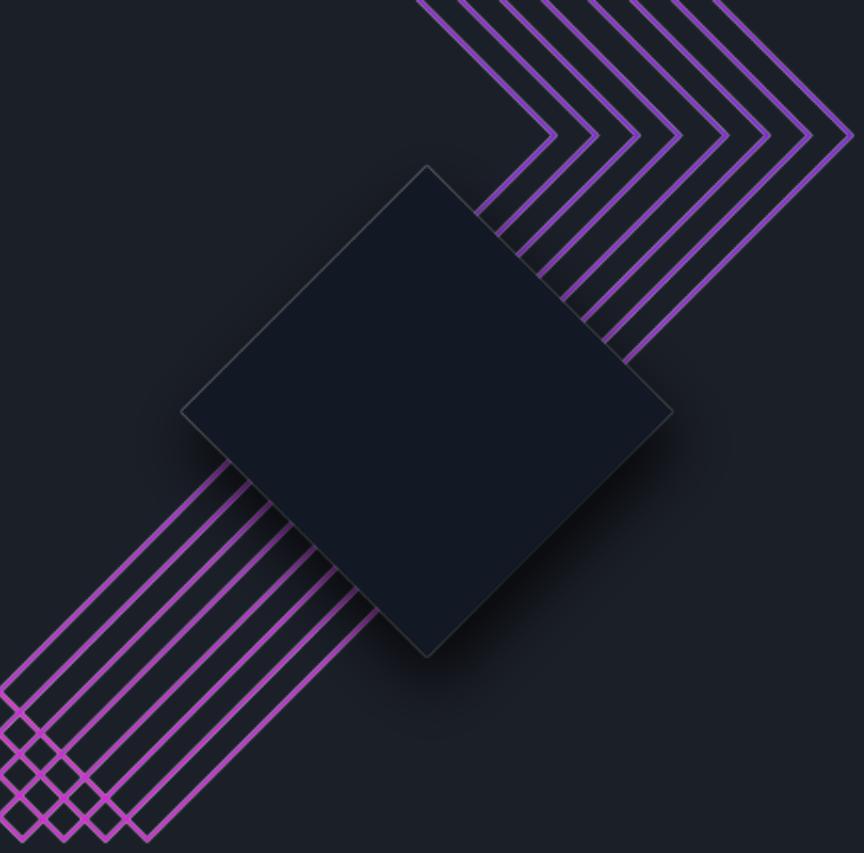


- ❖ nslookup - domain to IP resolution



DOCKER NETWORKING





TYPES OF DOCKER NETWORKS

01

Default
Bridge

02

User-defined
Bridge

03

Host

04

None



BRIDGE NETWORK

- ❖ It creates connection between:
 1. containers connected to the same network
 2. container and the host

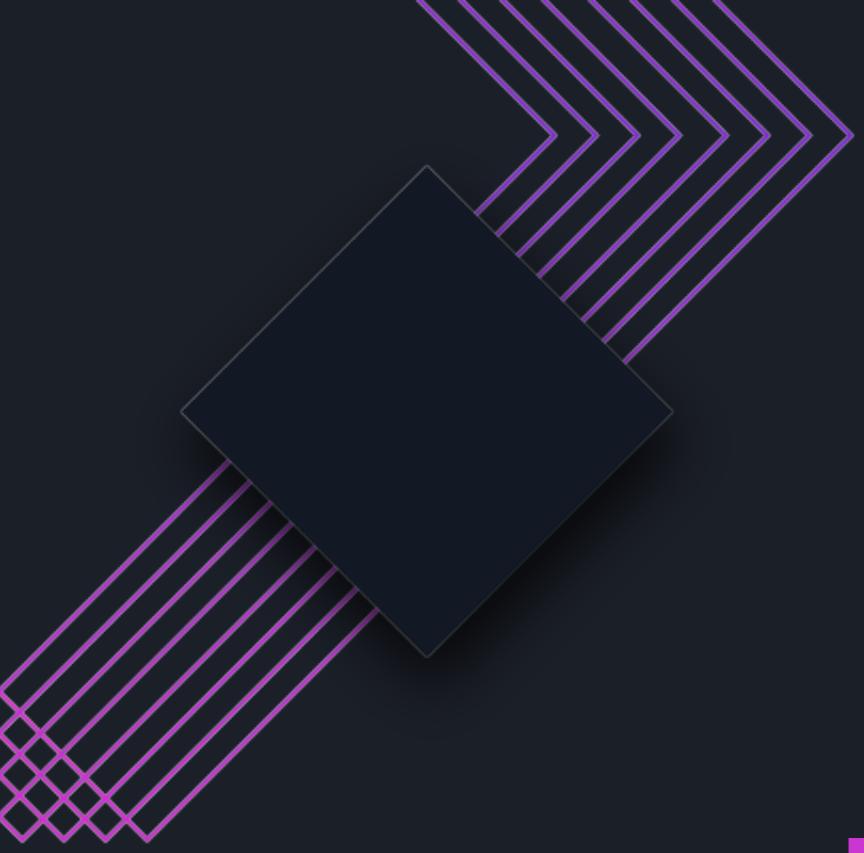
TYPES OF BRIDGE NETWORK

Default

- ❖ Docker0
- ❖ No in-built DNS
- ❖ No isolation

User-defined

- ❖ Any name
- ❖ DNS provided
- ❖ Containers are isolated



NETWORKING COMMANDS



- ◆ View all the available networks:

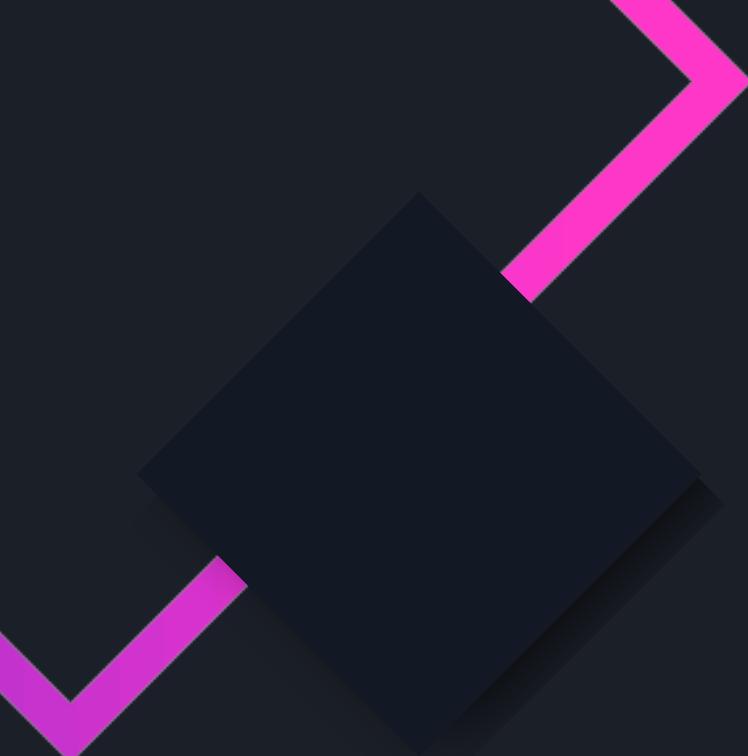


```
docker network ls
```

- ◆ Create a user-defined network:



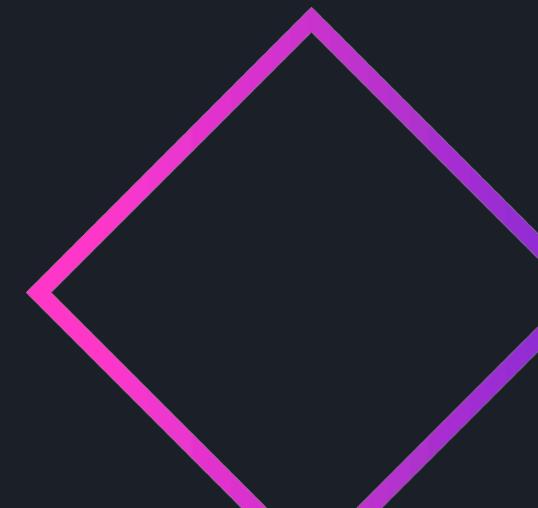
```
docker network create network_name
```

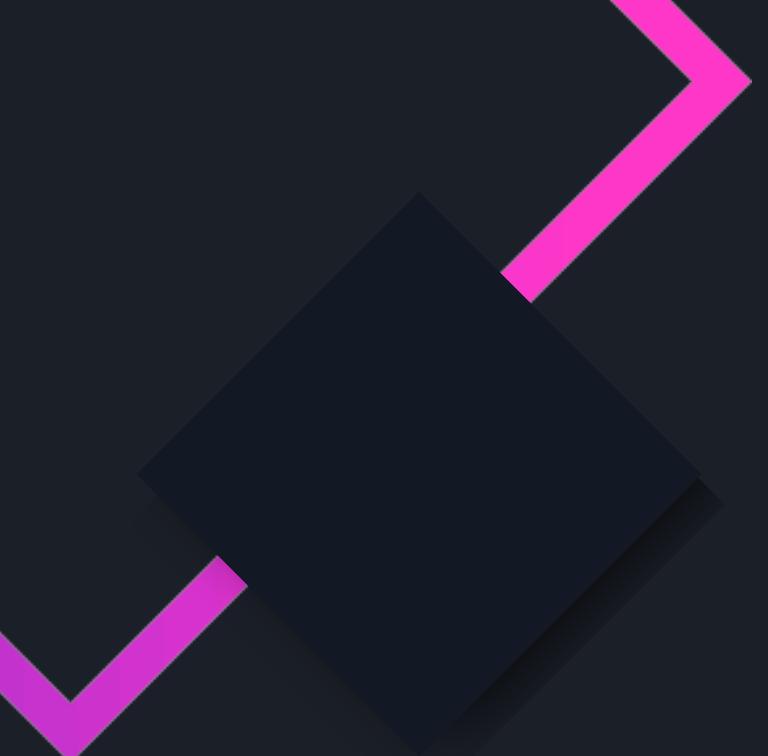
 Inspect a network:

```
docker network inspect network_name
```

 Connect a container to a network:

```
docker network connect network_name container_name
```





- ❖ Disconnect a container from a network:

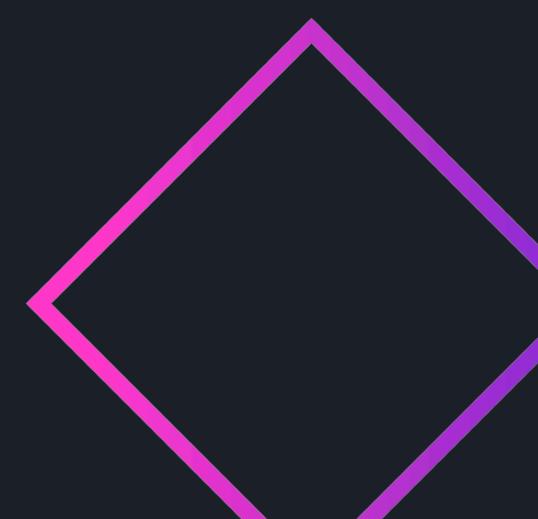


```
docker network disconnect network_name container_name
```

- ❖ Remove a network:



```
docker network rm network_name
```

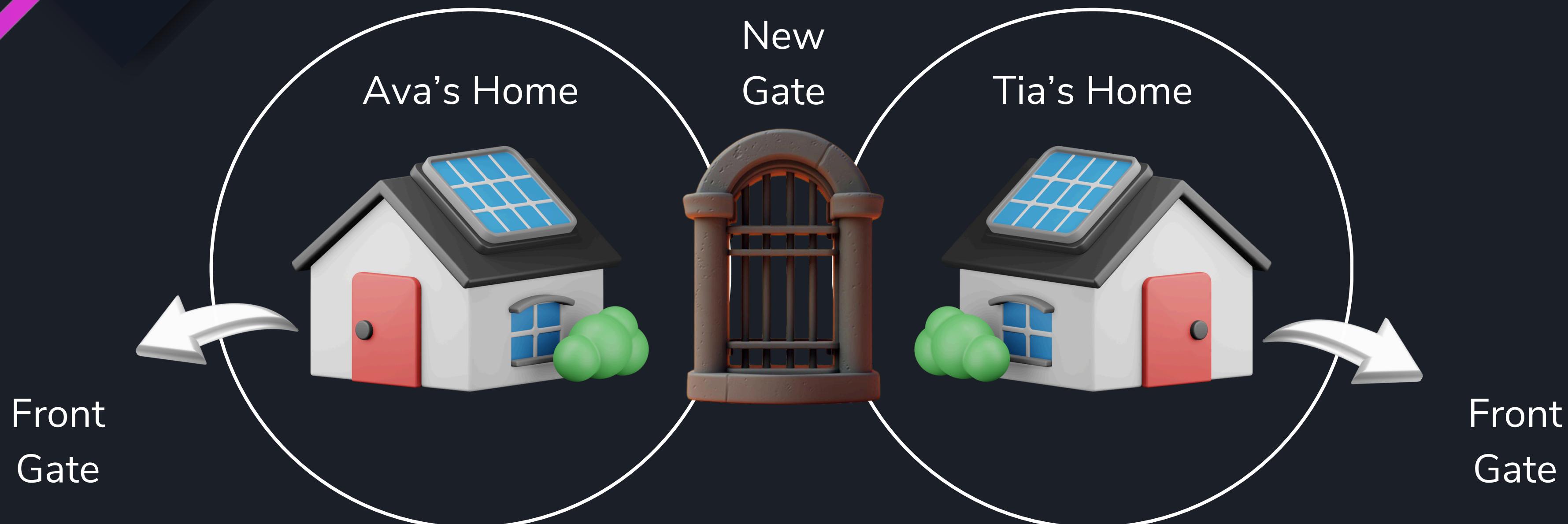


◆ Delete all unused networks:



```
docker network prune
```





net1

cont1



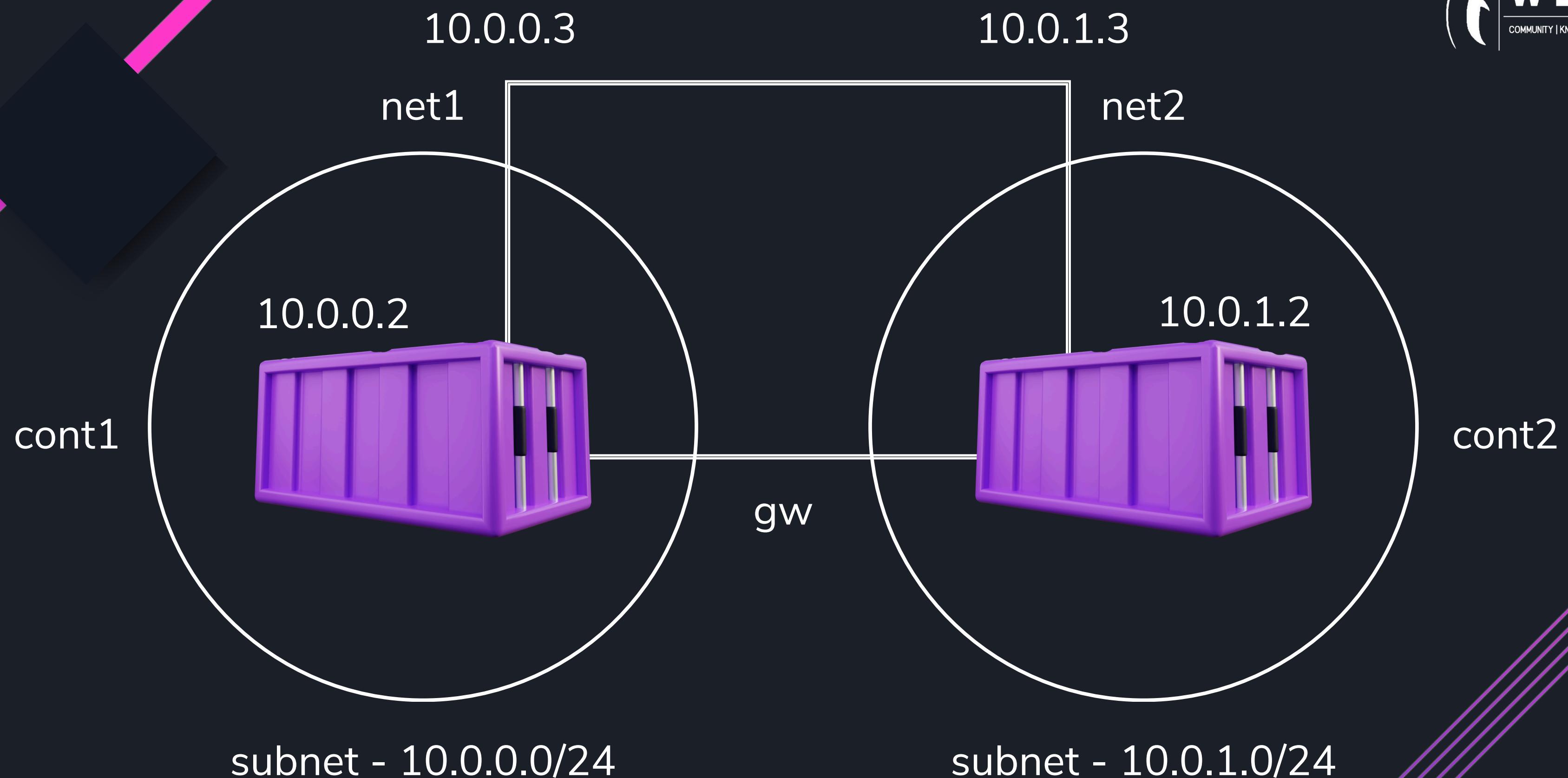
subnet - 10.0.0.0/24

net2

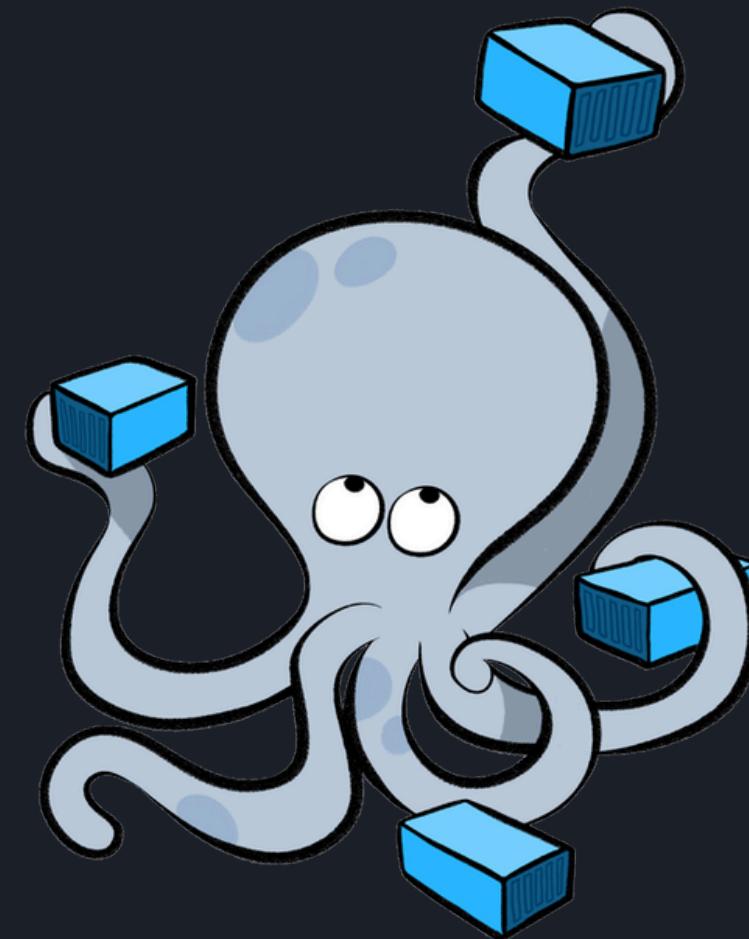
cont2



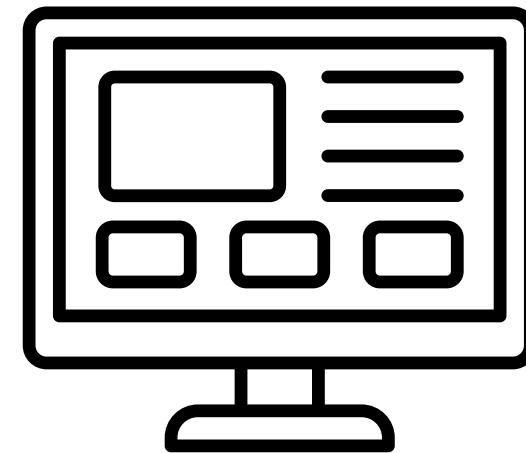
subnet - 10.0.1.0/24



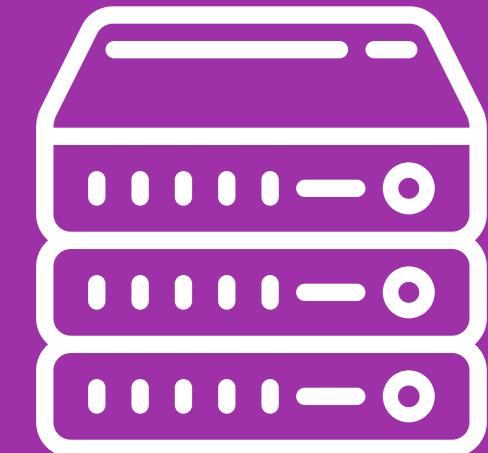
DOCKER COMPOSE



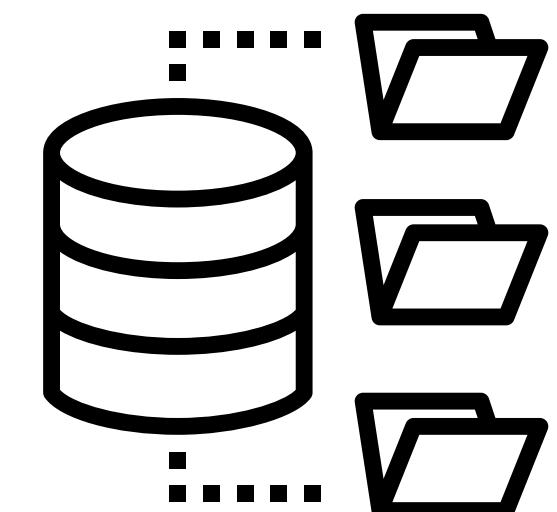
COMPONENTS OF AN APPLICATION



User Interface



Server

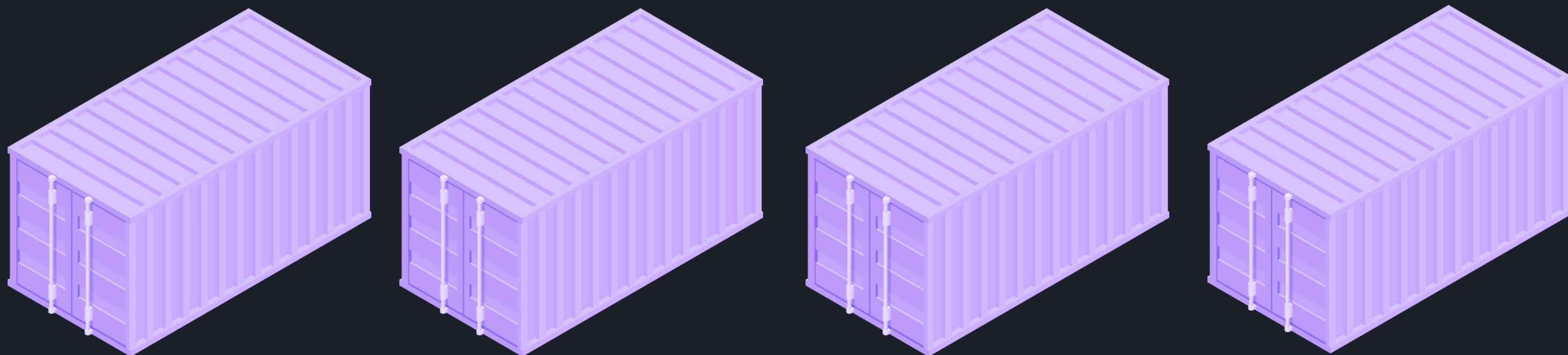


Database



Network

MULTI-CONTAINER APPLICATIONS





Docker Compose

WHY DOCKER-COMPOSE?

- ❖ Multi-container management
- ❖ Environment consistency
- ❖ Simplified networking
- ❖ Ease of deployment



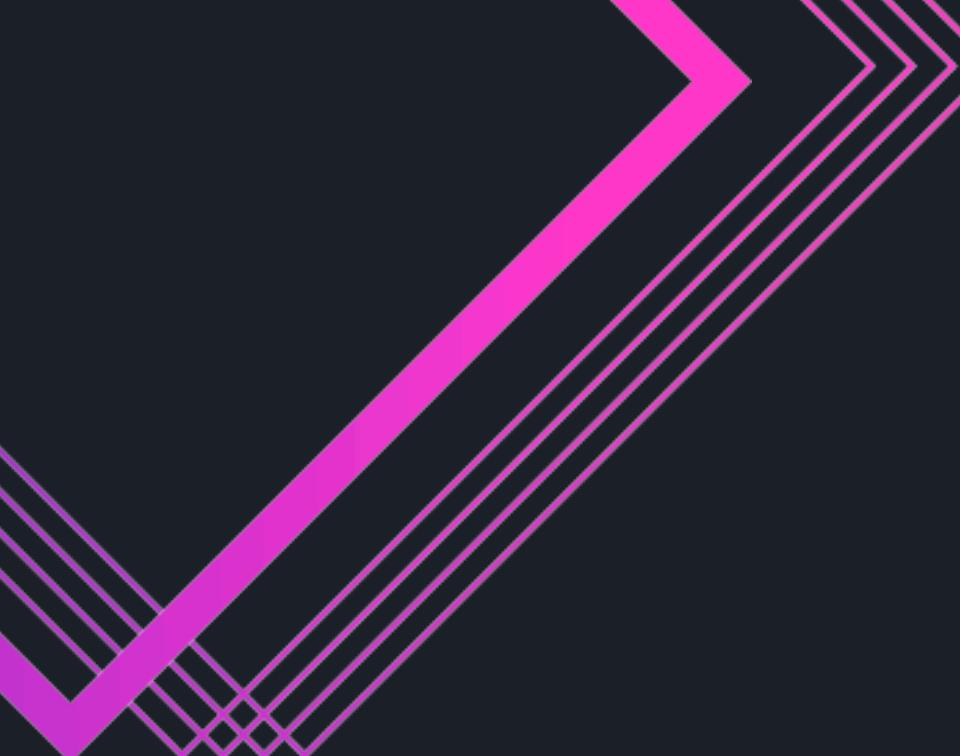
INSTALLATION



```
sudo apt update
```



```
sudo apt install docker-compose-plugin
```



HOW DOES IT WORK?



```
● ● ●

version: '3.9'

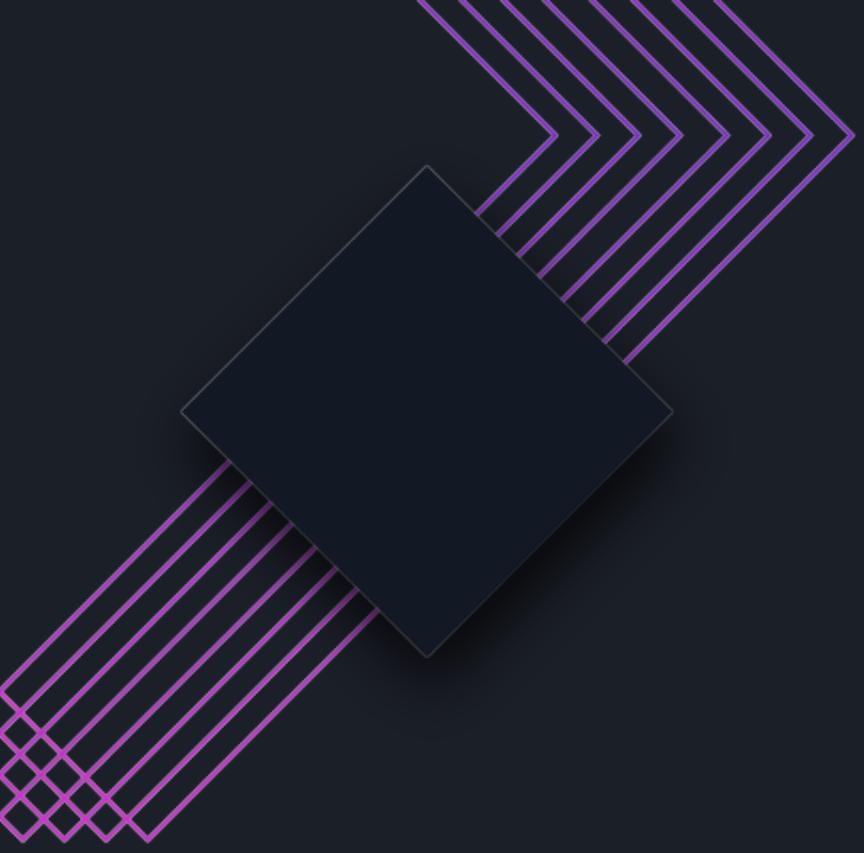
services:

  service1:
    image: image_name
    build:
      context: ./path_to_context
      dockerfile: Dockerfile
    ports:
      - "host_port:container_port"
    environment:
      ENV_VAR1: value1
      ENV_VAR2: value2
    volumes:
      - ./host_path:/container_path
    networks:
      - network_name
    depends_on:
      - service2

  volumes:
    volume_name:
      driver: local

  networks:
    network_name:
      driver: bridge
```

YAML File



COMPONENTS OF YML FILE

01

Version

02

Services

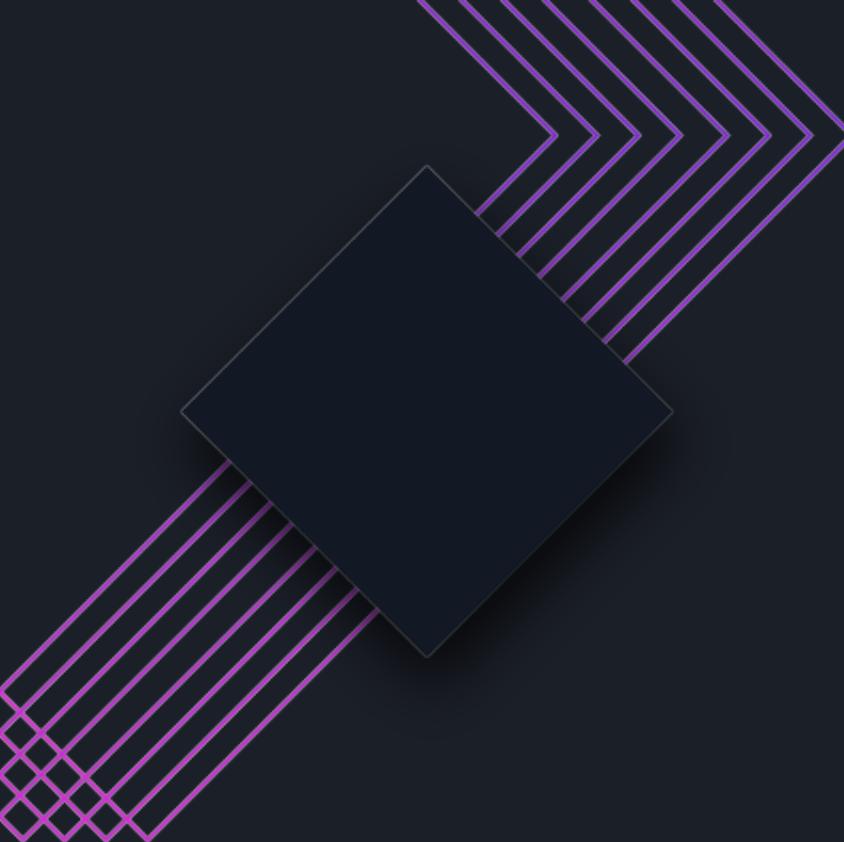
03

Volumes

04

Networks





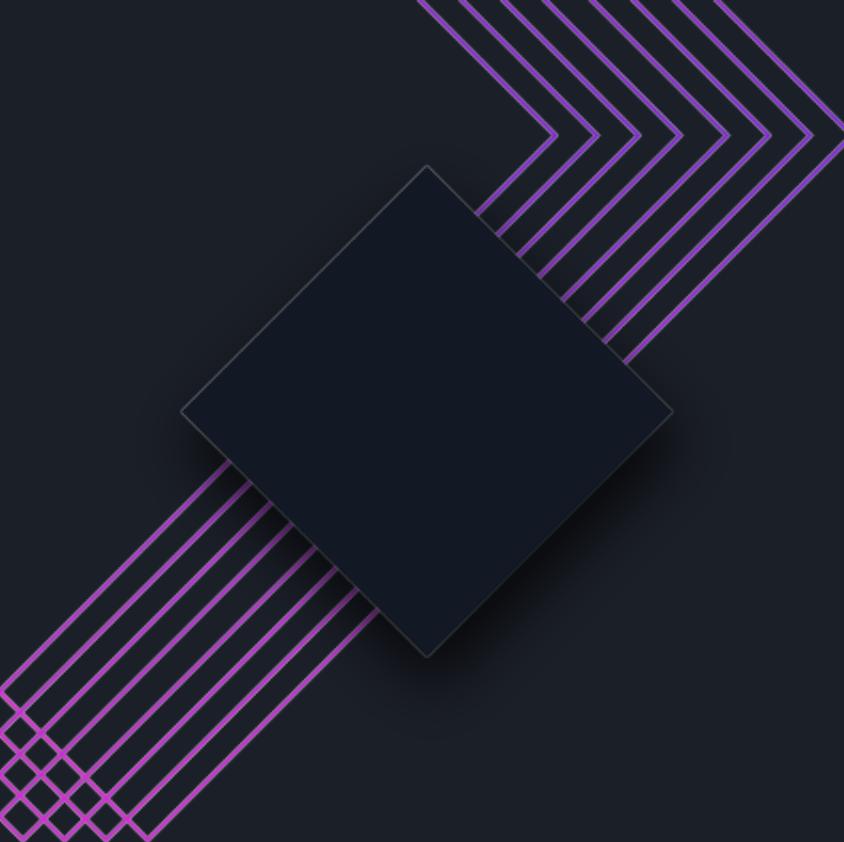
VERSION



01

- ❖ Specifies the syntax version of the Docker Compose file
- ❖ Ensure compatibility with Docker features





SERVICE

02

Defines the containers to be run

image

ports

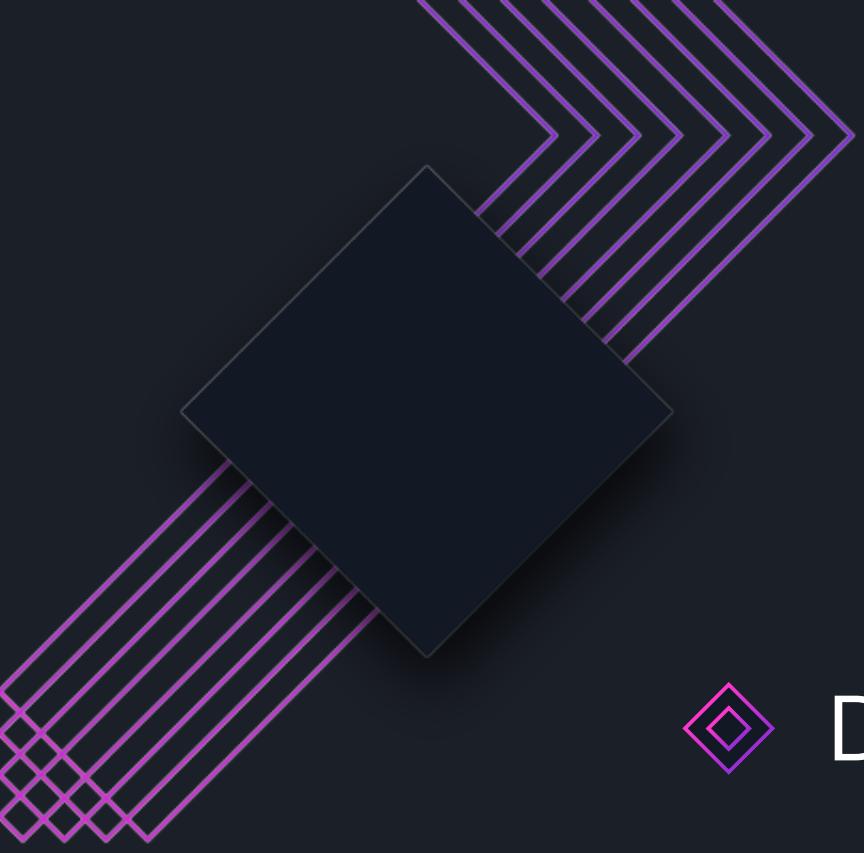
environment

volumes

networks

depends_on





VOLUMES

03

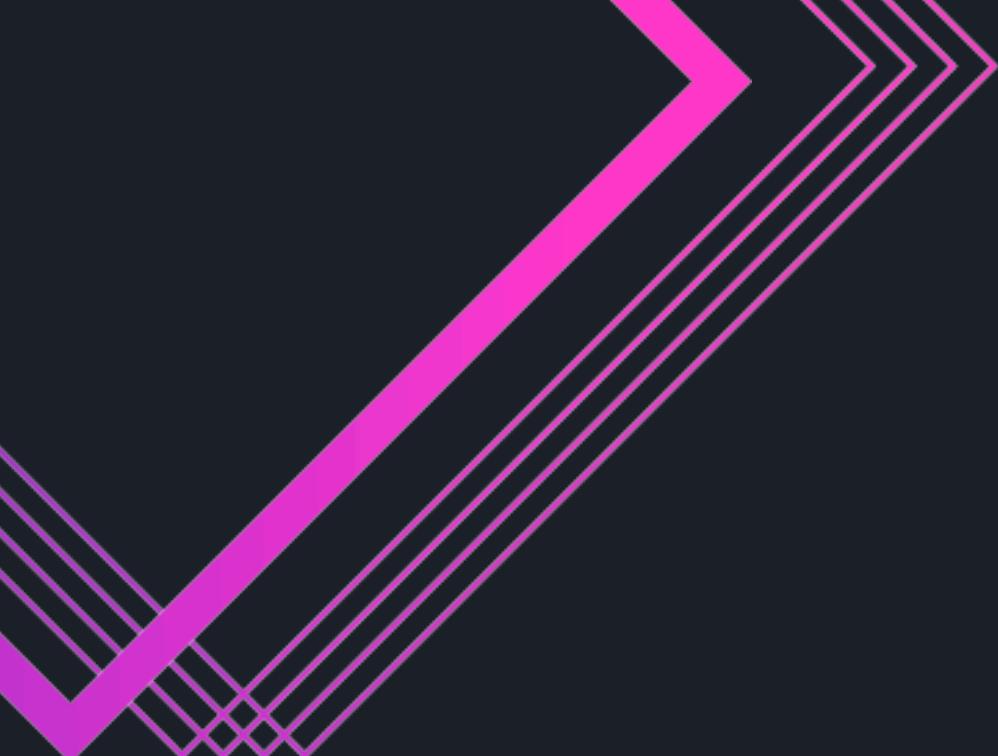
- ❖ Define shared storage between:
 - ❖ Host and containers
 - ❖ Among containers

NETWORKS

04

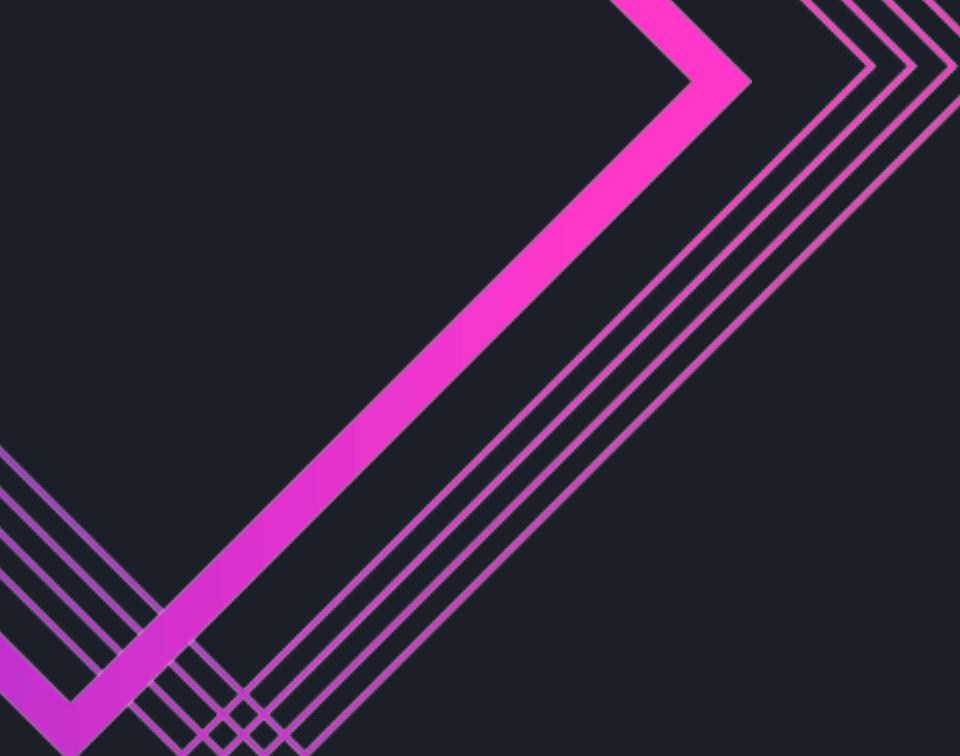
- ❖ Define custom networks for communication





**LET'S
COMPOSE!**





THANK YOU!

