

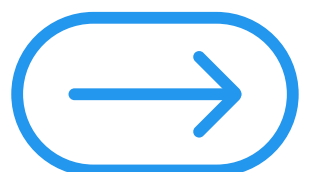
Day 32/50



# Docker Networking

**CONNECTING CONTAINERS THE RIGHT  
WAY**

*Let your containers communicate securely  
and effectively.*



# WHY DOCKER NETWORKING MATTERS

**Containers often need to:**

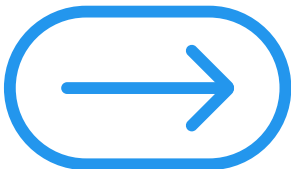
- **Talk to each other (e.g., app ↔ database)**
- **Expose services to the outside world**
- **Stay isolated from other containers**

**Docker networking handles all that — with zero manual IP work.**



# TYPES OF DOCKER NETWORKS

Type	Use case
Bridge	Default. Great for container-to-container
Host	Shares host's network stack (fast but risky)
None	No network (full isolation)
Overlay	Multi-host networking (Swarm/K8s)
Macvlan	Assign MAC address (advanced)



# HOW BRIDGE NETWORK WORKS

- Docker creates a default bridge network
- Containers get virtual IPs
- They can talk via container names (DNS resolution!)
- Perfect for most dev/test use cases



Example:

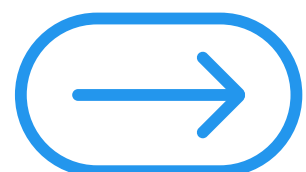


```
docker network create my-bridge
```

```
docker run -d --name db --network my-bridge mysql
```

```
docker run -d --name app --network my-bridge my-app
```

**NOW APP CAN CONNECT TO DB:3306**



# HOW BRIDGE NETWORK WORKS



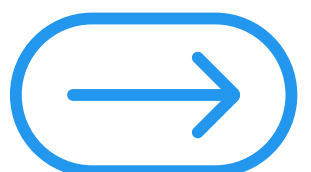
```
docker network ls
```

```
docker network inspect bridge
```

```
docker network create my-net
```

```
docker run --network my-net nginx
```

```
docker network connect my-net app
```



# EXPOSING CONTAINERS TO OUTSIDE



```
DOCKER RUN -D -P 8080:80 NGINX
```

- Maps host port 8080 → container port 80
- Accessible at <http://localhost:8080>

🛑 Be mindful of security — don't expose

sensitive containers without firewalls.



# BEST PRACTICES

- ✓ Use named networks for multi-container setups
- ✓ Use container names instead of hardcoded IPs
- ✓ Avoid host network unless you really need speed
- ✓ Secure public ports with firewalls & reverse

proxies

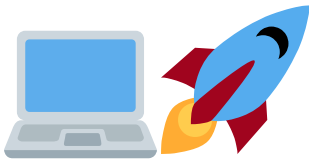
**Networking is how Docker containers collaborate.**

**If you want to build real-world apps, Docker  
networking is not optional.**



Join the ride as we break down  
DevOps, one day at a time!

**FOLLOW ME**

Let's build, break, and learn –  
together. 

**RAGHAVENDER CHARI**