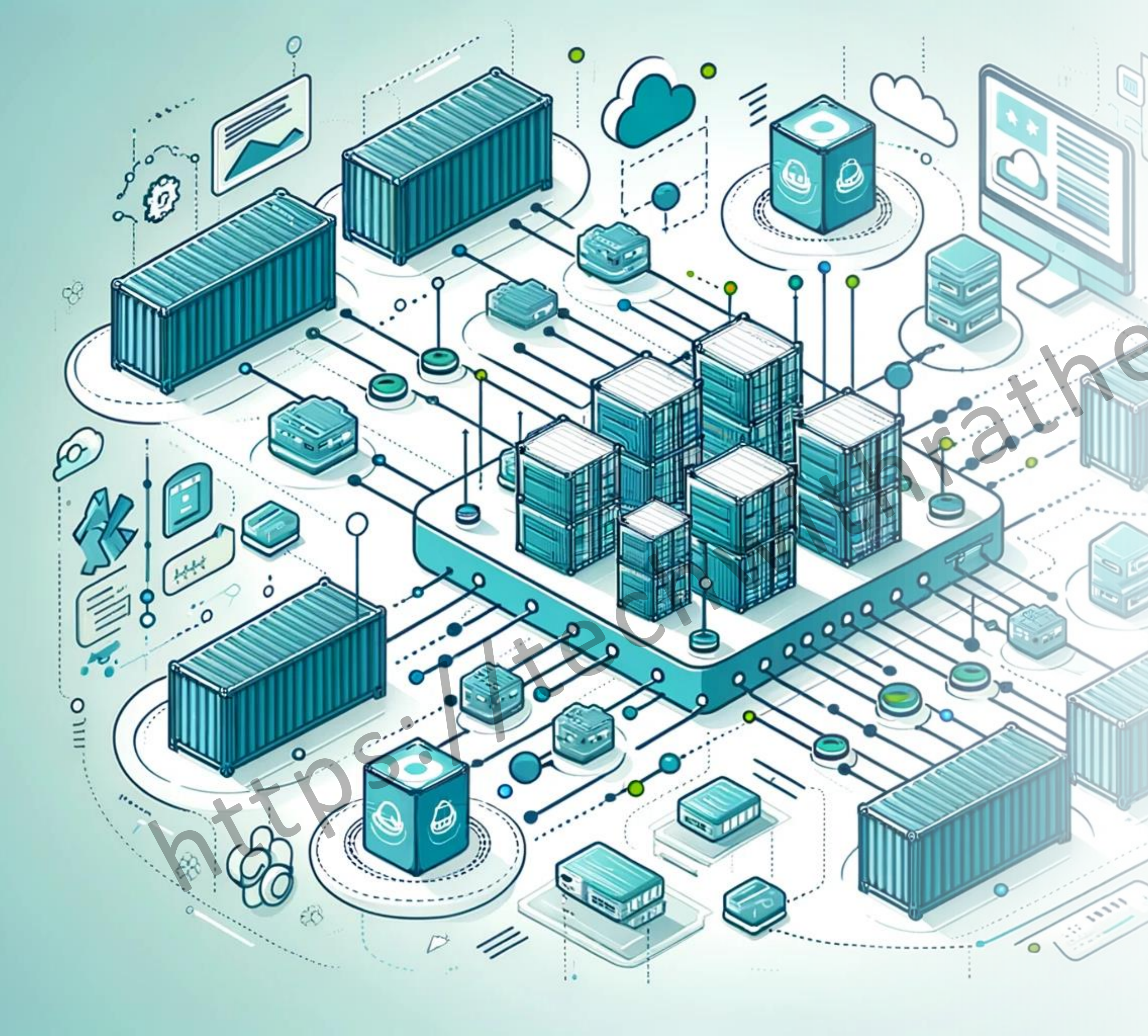
docker

Docker Networking



Introduction to Docker Networking

Docker Networking:

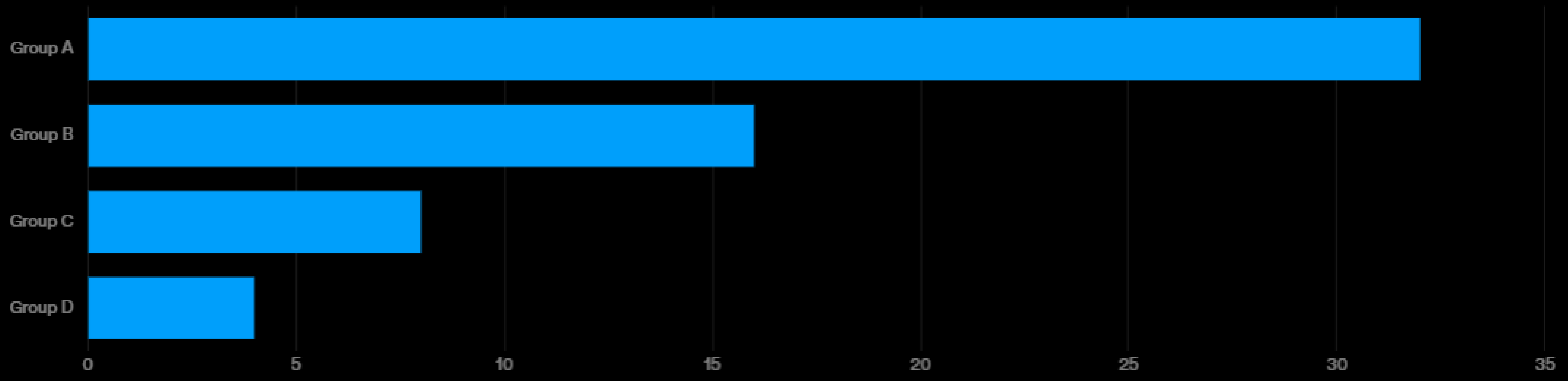
- Docker networking allows containers to communicate with each other and the outside world.

Key Points:

- Ensures isolation and security.
- Integral for container orchestration.

Network Drivers: Briefly mention Bridge, Host, Overlay, Macvlan, and None, setting the stage for detailed discussion in subsequent slides.

Container Networking Models



Bridge

- Used for single-host networking.

Overlay

- Used for multi-host networking.

Host

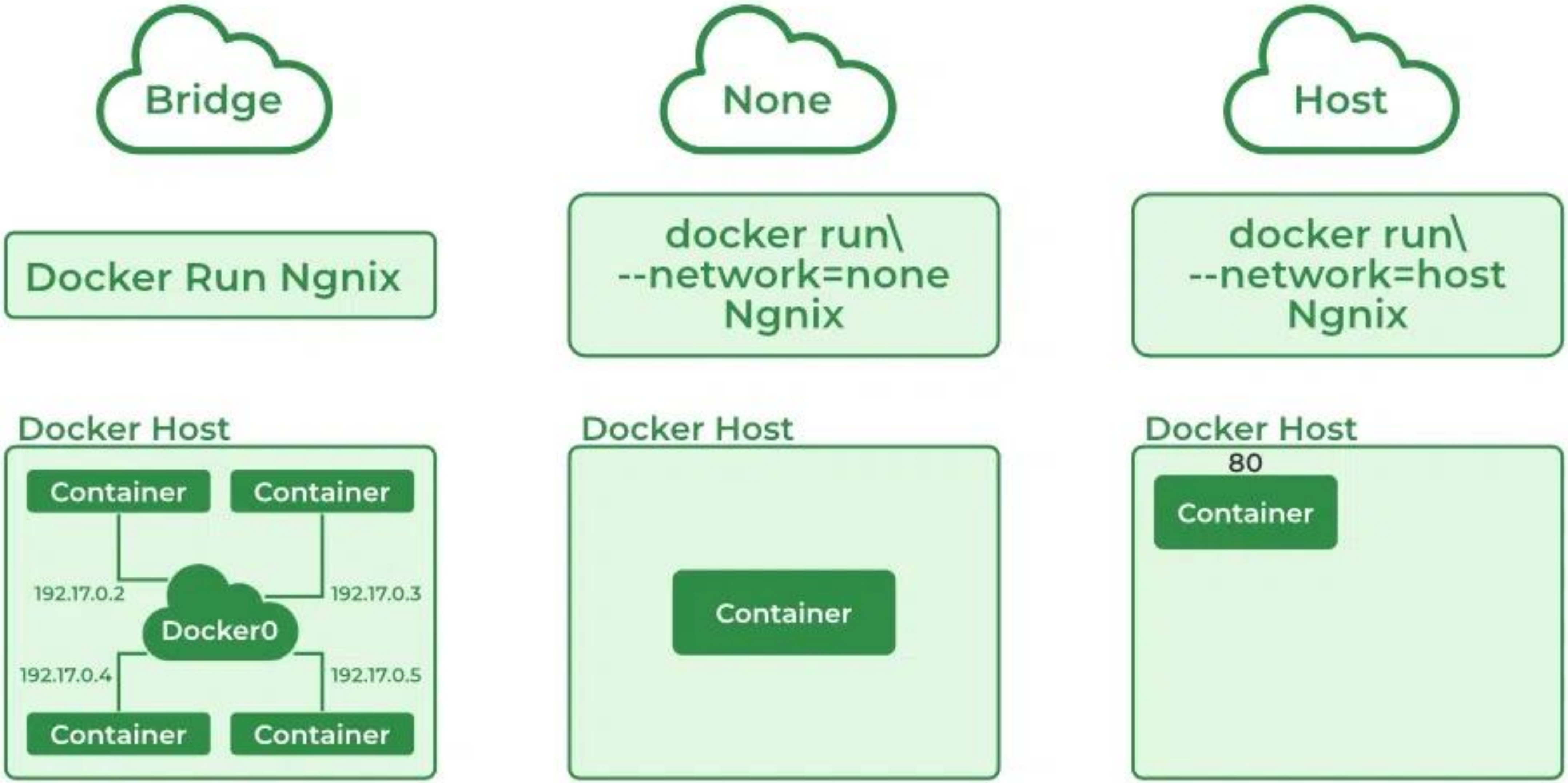
- Used for host networking.

Understanding Network Drivers

Explanation: Network drivers provide the underlying technology for container networking.

Types:

- **Bridge:** Default, for isolated networks on a single host.
- **Host:** Removes network isolation between container and host.
- **Overlay:** Connects multiple Docker daemons.
- **Macvlan:** Assigns a MAC address to containers.
- **None:** Disables networking.



Bridge

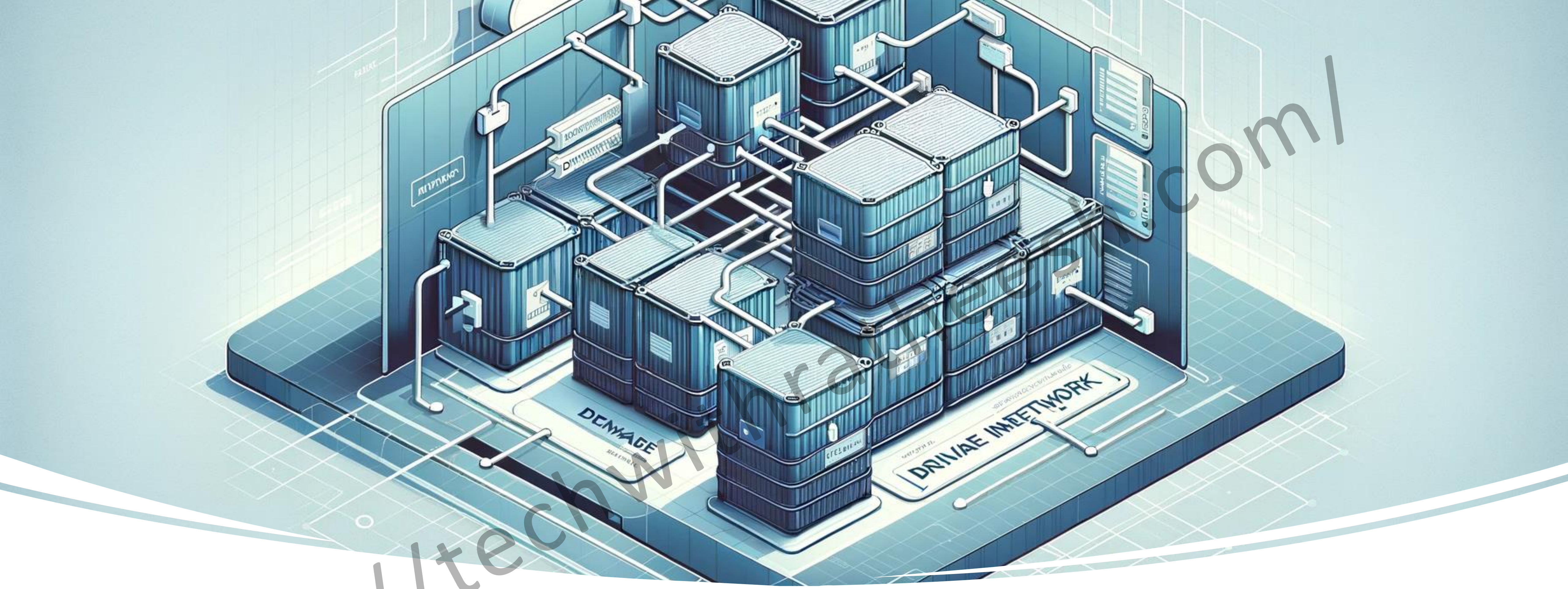
Docker Run Ngnix

None

```
docker run\
--network=none
Ngnix
```

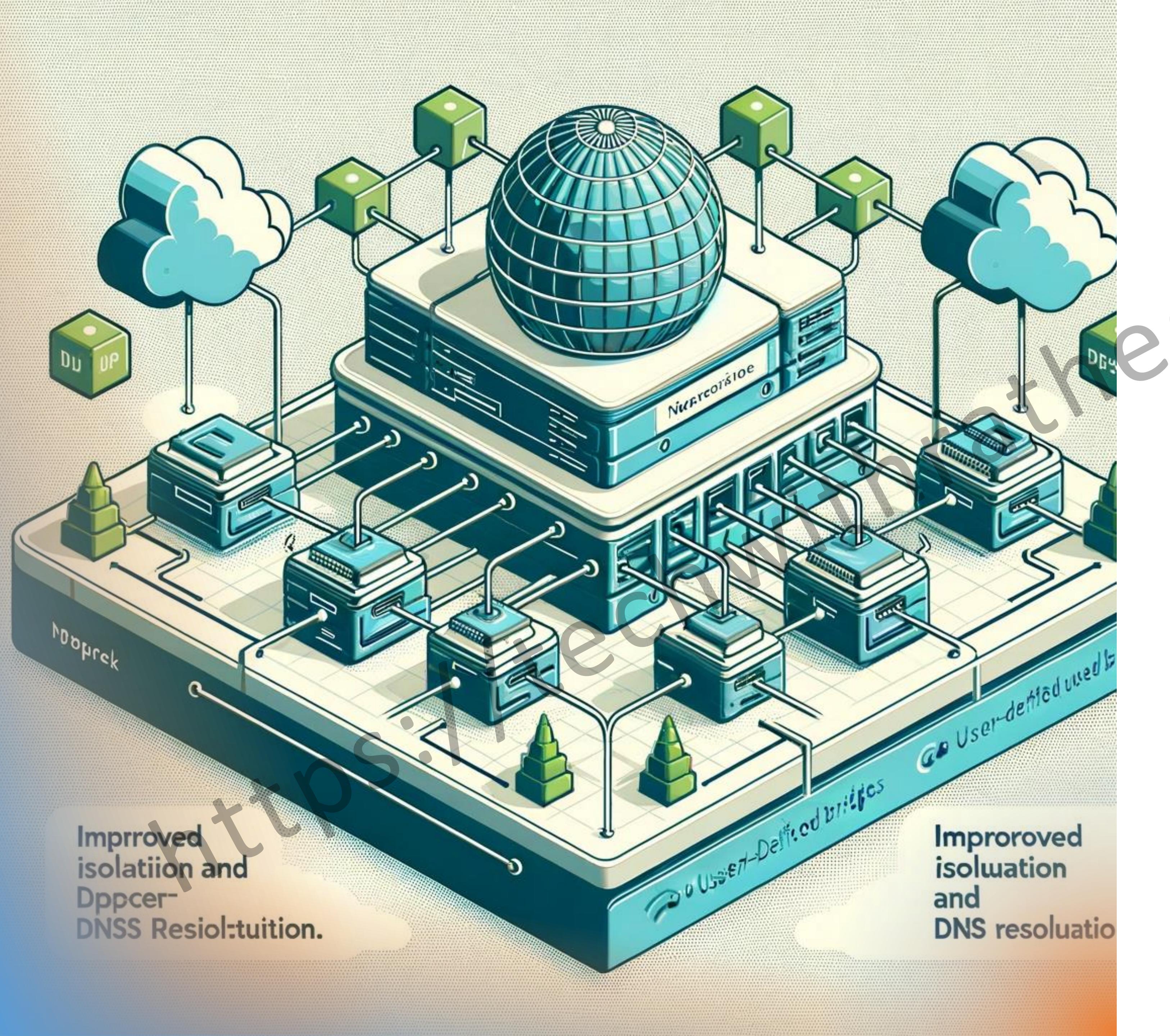
Host

```
docker run\
--network=host
Ngnix
```



The Default Bridge Network

- **Overview:** Automatically created network for containers.
- **Characteristics:**
- - Private internal network on the host.
- - Containers on different bridges cannot communicate directly.
- **Limitations:** Not ideal for complex networking needs.


DU UP
Nurcoribloe
DG
Noprck
User-defined used b
User-Defi:ed bridges
Imprroved isolatiion and Dppcer- DNSS Resiol:tuition.
Improroved isoluation and DNS resoluatio

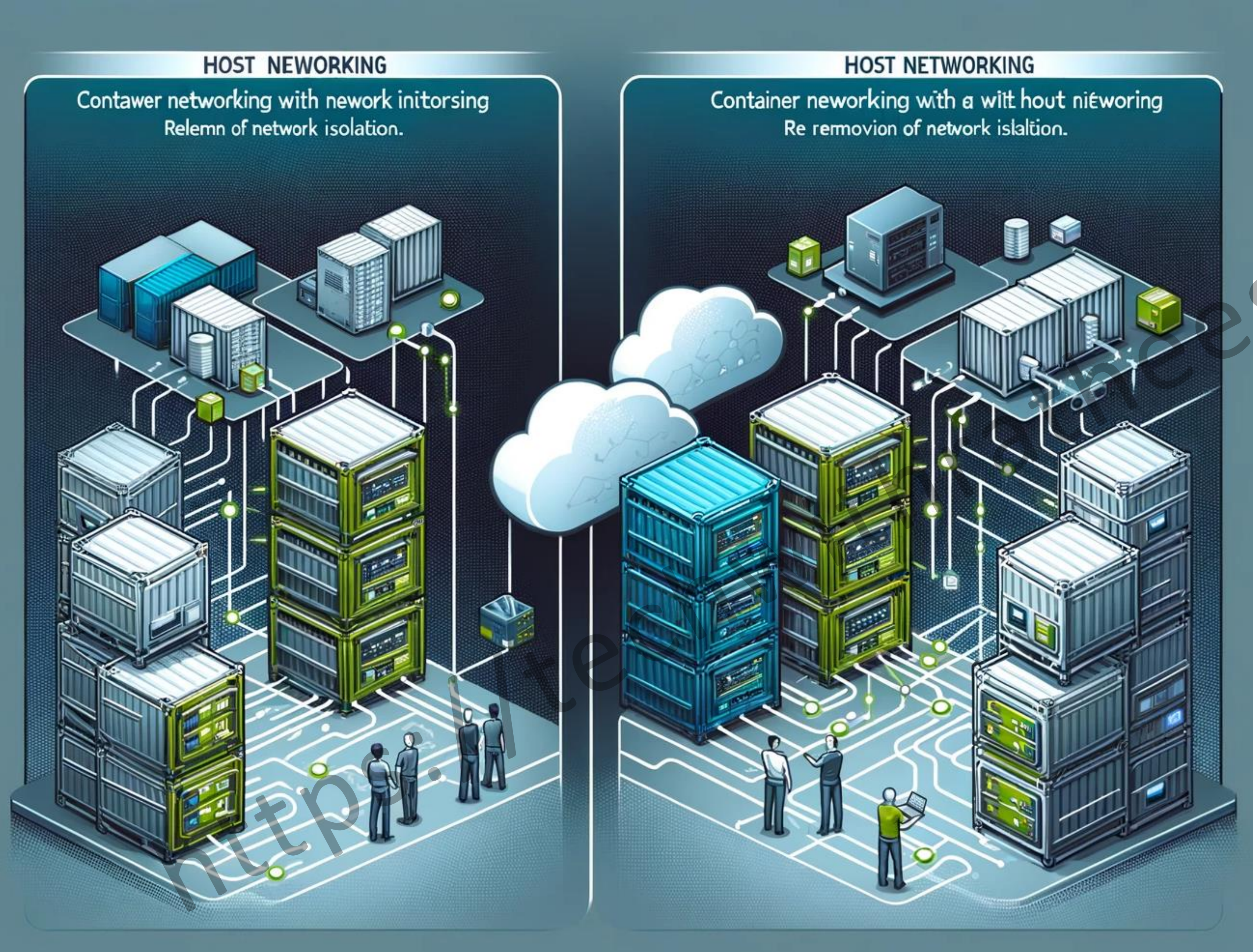
User-Defined Bridges

Benefits: more flexibility and better isolation than the default bridge.

Usage: Creating with Docker Network Create.

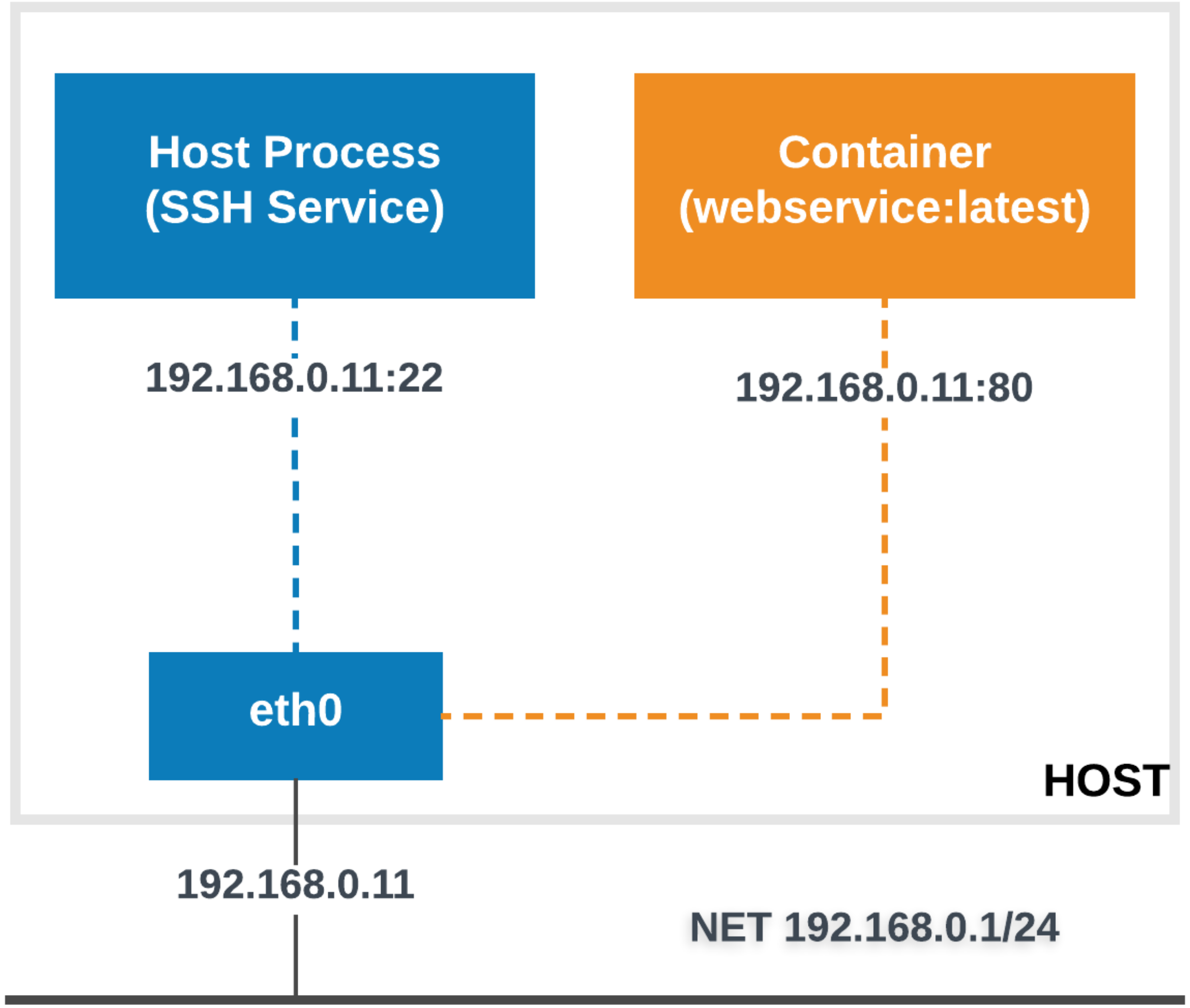
- Connecting containers to the network.

Networking Features: enhanced DNS resolution, allowing container name-based communication.


HOST NEWORKING
Contawer networking with nework initorsing
Relemn of network isolation.
HOST NETWORKING
Container neworking with a wit hout niεworing
Re remmovion of network islaltion.

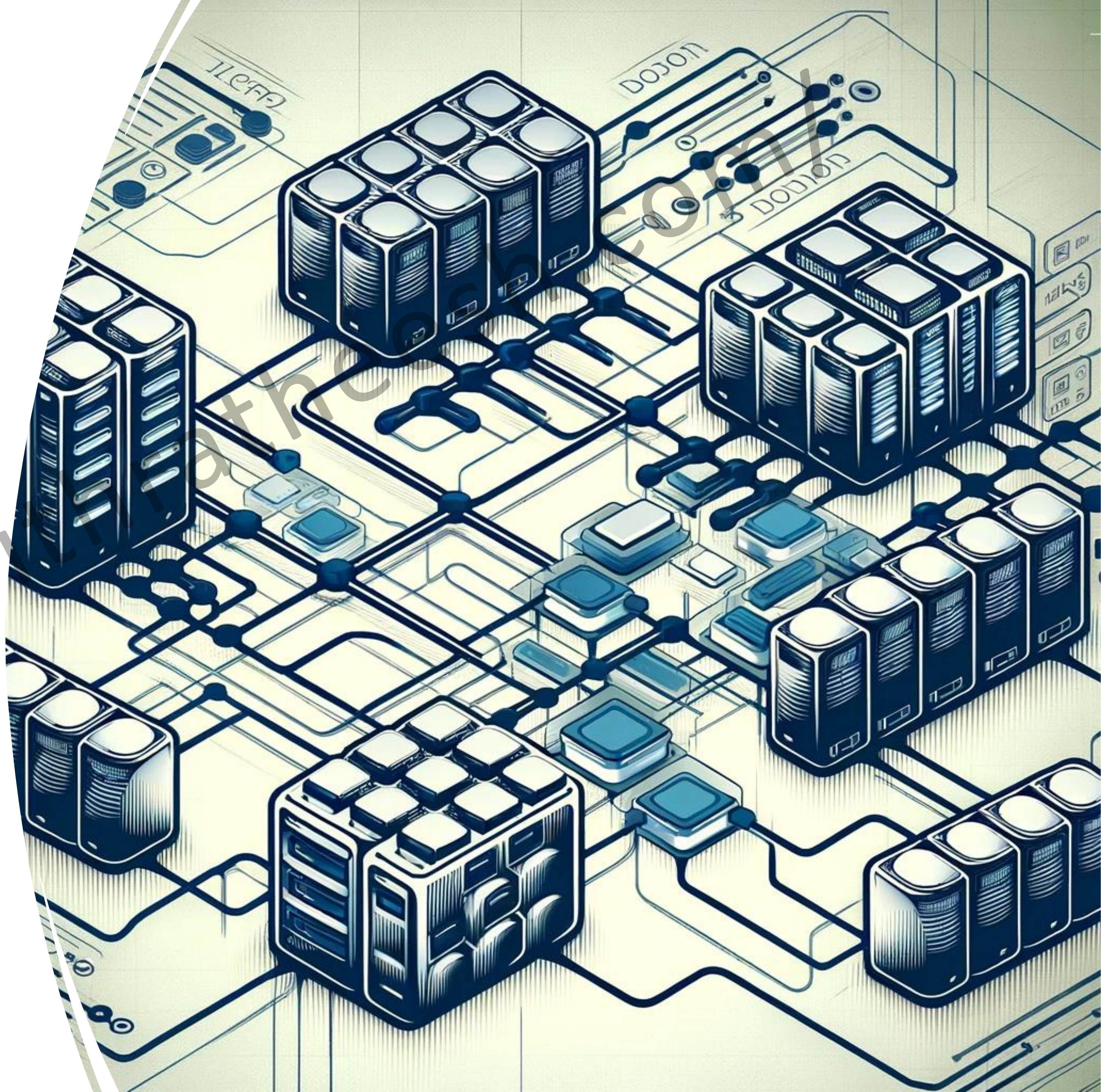
Host Networking

- **Concept:** Removes network isolation between container and Docker host.
- **Implementation:** Using --network=host in the Docker run command.
- **Use-Cases:** Performance-critical situations, like high-speed networking.


Host Process
(SSH Service)
Container
(webservice:latest)
192.168.0.11:22
192.168.0.11:80
eth0
HOST
192.168.0.11
NET 192.168.0.1/24

Overlay Networks

- **Purpose:** Enables network communication across multiple Docker hosts.
- **Core Components:**
- Overlay driver.
- Network control plane (managing endpoints).
- **Creating Overlays:** Steps to create and manage in a Docker Swarm environment.

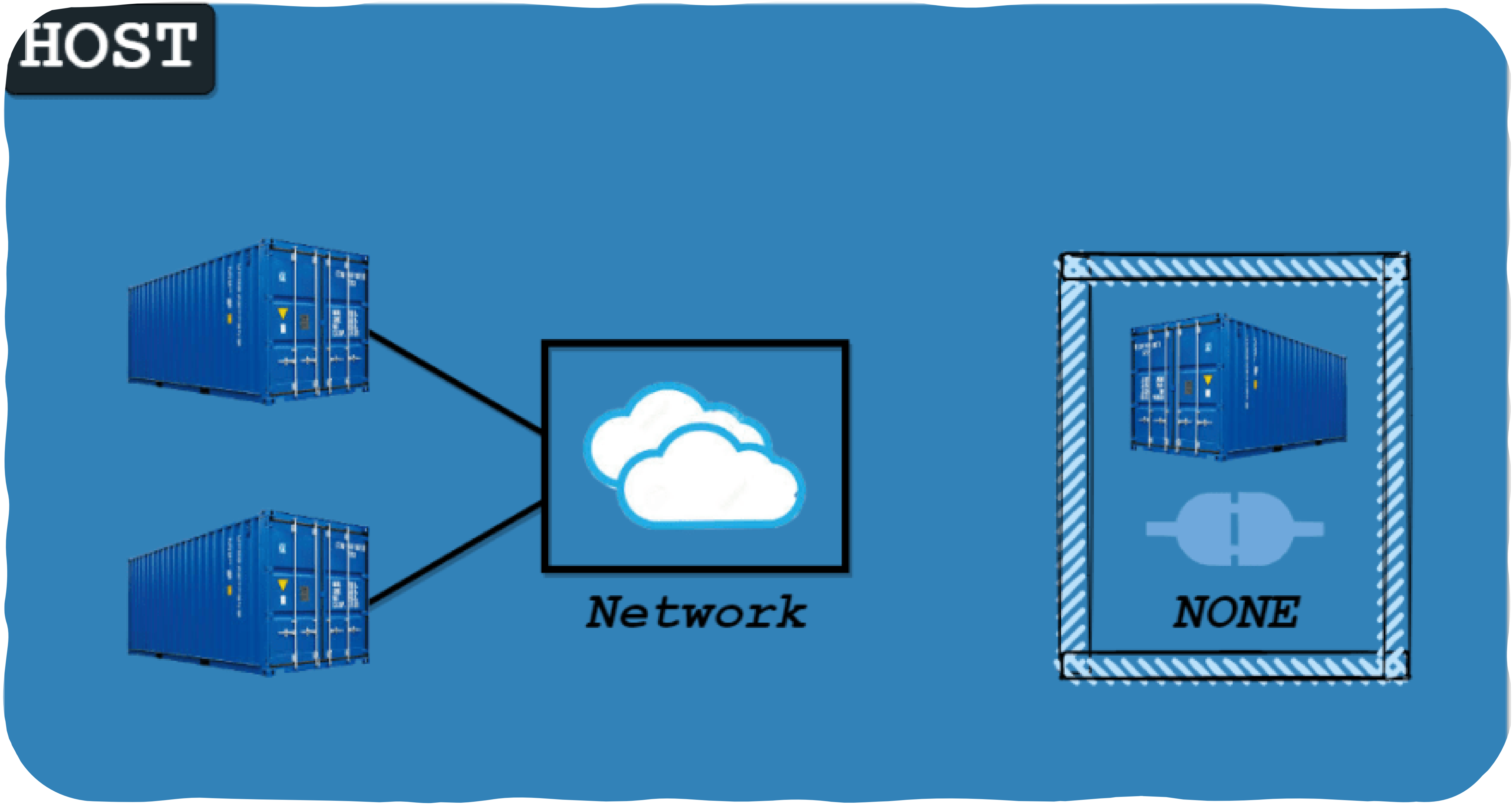


Macvlan Networks

- **Functionality:** Gives a container its own MAC and IP address.
- **Configuration:** Step-by-step guide on setting up a Macvlan network.
- **Scenarios:** Useful when migrating VMs to containers.

None Network

- **Role**: Completely disables networking for a container.
- **When to Use:** Security-intensive applications, testing environments.
- **Security Implications:** Offers the highest level of network isolation.

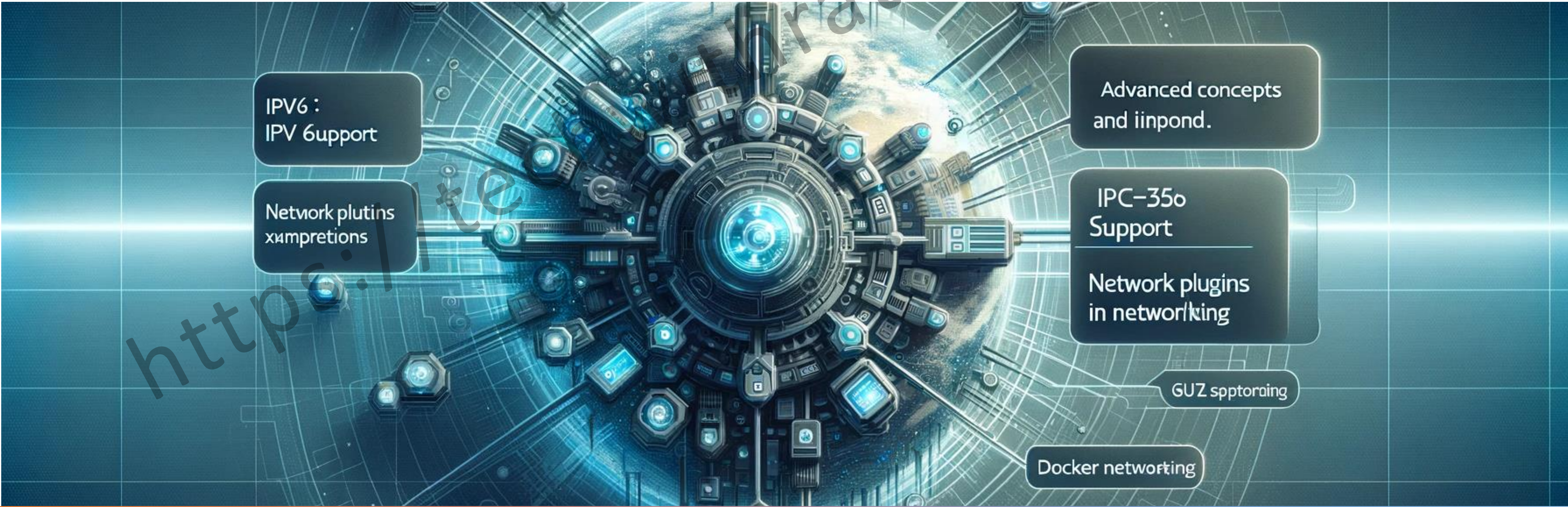

HOST
Network
NONE

Network Inspection and Troubleshooting

- **Tools:** Commands like docker network inspect.
- **Troubleshooting Tips:** Identifying common networking issues and their solutions.
- **Best Practices:** Recommendations for efficient network management.

Advanced Networking Concepts

- **IPv6 Support:** How Docker Handles IPv6 Networking.
- **Network Plugins:** Extending Docker networking capabilities.
- **Future Trends**: Evolving networking trends in Docker and containerization.



Contact

☎ +91 9446330906

✉ RATHEESHKUMAR.2008@GMAIL.COM

🌐 www.techwithratheesh.com

in linkedin.com/in/ratheesh-kumar-08722619

Thank You..!