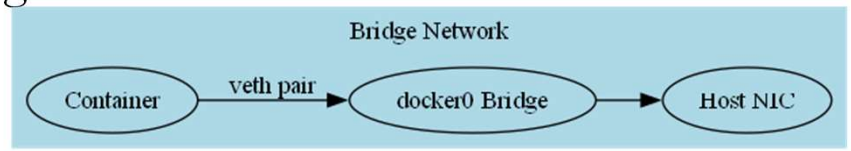
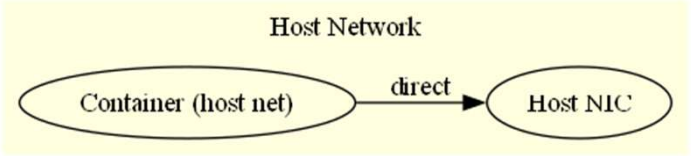
Docker Networking, Storage, and Compose

Docker Network Types

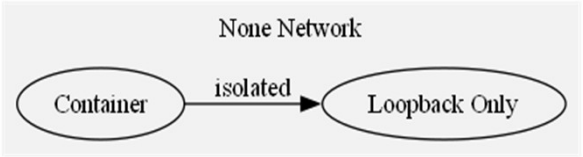
bridge: Isolated container communication on a single host



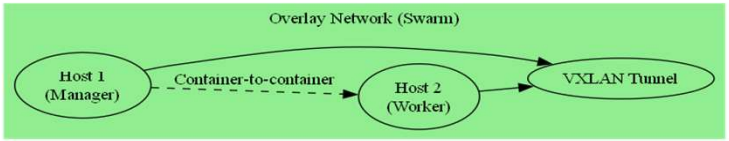
host: Shares host's network stack, good for performance



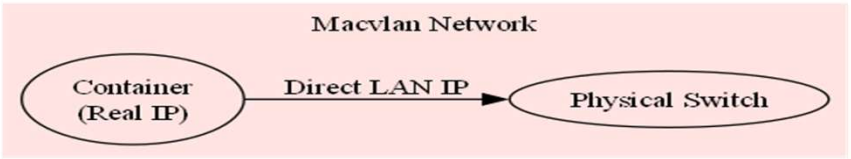
none: No network, max isolation



overlay: Multi-host networking using VXLAN (Swarm)



macvlan: Direct access to LAN, legacy app support



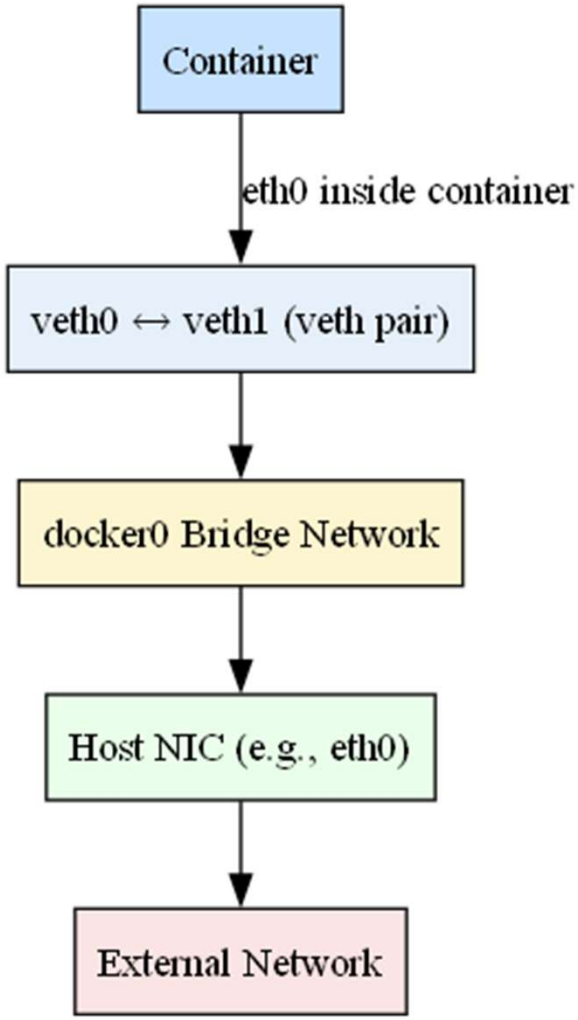
How Docker Networking Works Internally

Each container has a veth pair

One end in container, other on bridge (docker0)

iptables handles NAT and forwarding

Embedded DNS server (127.0.0.11) for name resolution



Docker Volumes

Managed by Docker, stored at /var/lib/docker/volumes/

Persistent across container lifecycle

Decoupled from container image

Can be shared across containers

Preferred method for production storage

Bind Mounts and tmpfs

Bind Mounts: Host-managed paths, flexible but less portable

tmpfs: In-memory, ideal for secrets, auto-cleared on exit

Docker Compose Overview

Tool to define and run multi-container apps

Uses docker-compose.yml for declarative configuration

Built-in networking and volume reuse

Useful for development, testing, and CI environments

Compose Lab: Flask + Redis

Flask web app using Redis to store page view count

docker-compose.yml defines web and redis services

Run with: docker-compose up --build

Access app at http://localhost:5000

Tear down with: docker-compose down -v