Networking overview

Container networking refers to the ability for containers to connect to and communicate with each other, or to non-Docker workloads.

A container has no information about what kind of network it's attached to, or whether their peers are also Docker workloads or not. A container only sees a network interface with an IP address, a gateway, a routing table, DNS services, and other networking details. That is, unless the container uses the none network driver.

Network drivers overview

Docker's networking subsystem is pluggable, using drivers. Several drivers exist by default, and provide core networking functionality:

- bridge: The default network driver. If you don't specify a driver, this is the type of network you are creating. Bridge networks are commonly used when your application runs in a container that needs to communicate with other containers on the same host. See Bridge network driver.

- host: Remove network isolation between the container and the Docker host, and use the host's networking directly. See Host network driver.

- overlay: Overlay networks connect multiple Docker daemons together and enable Swarm services and containers to communicate across nodes. This strategy removes the need to do OS-level routing. See Overlay network driver.

- ipvlan: IPvlan networks give users total control over both IPv4 and IPv6 addressing. The VLAN driver builds on top of that in giving operators complete control of layer 2 VLAN tagging and even IPvlan L3 routing for users interested in underlay network integration. See IPvlan network driver.

- macvlan: Macvlan networks allow you to assign a MAC address to a container, making it appear as a physical device on your network. The Docker daemon routes traffic to containers by their MAC addresses. Using the macvlan driver is sometimes the best choice when dealing with legacy applications that expect to be directly connected to the physical network, rather than routed through the Docker host's network stack. See Macvlan network driver.

- none: Completely isolate a container from the host and other containers. none is not available for Swarm services. See None network driver.

- Network plugins: You can install and use third-party network plugins with Docker.