Docker Certified Associate

Container Network Model

The Container Network Model

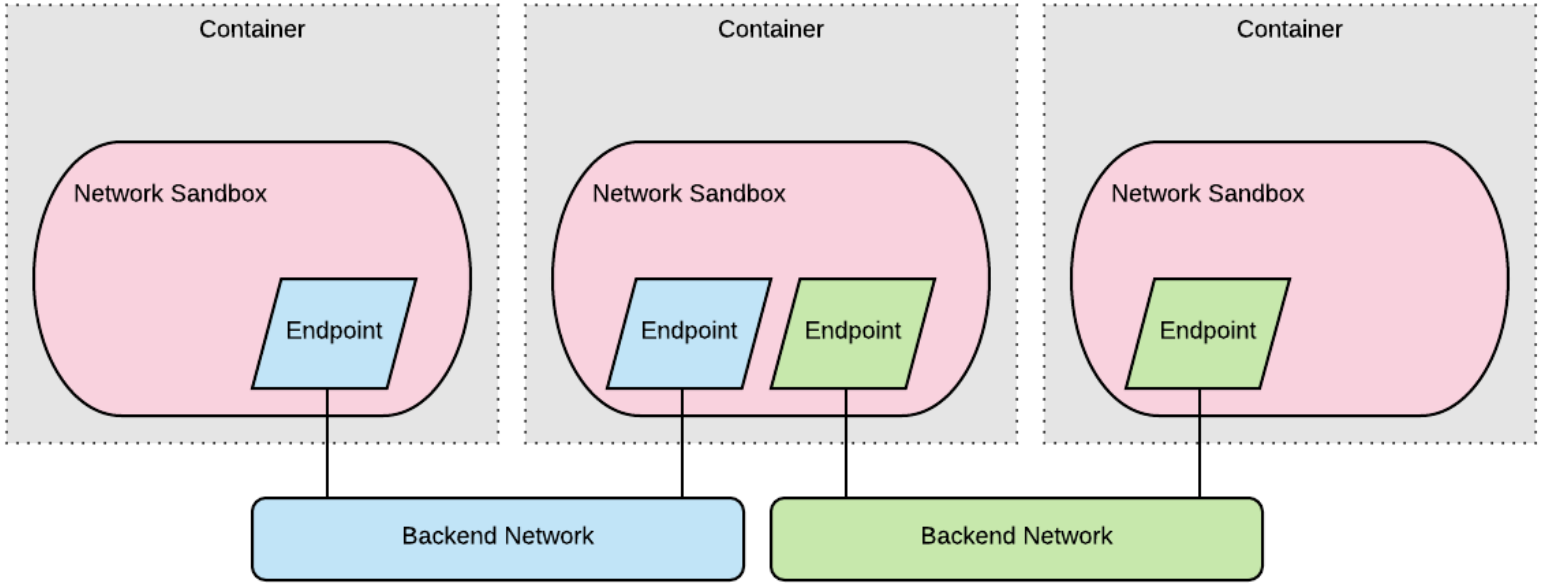
Implementation that formalizes how networking for containers is provided while allowing abstractions that are used to support multiple network drivers. It is built on three main components:

Components

- Sandbox – encompasses the network stack configuration, including management of interfaces, routing and DNS of 1 to N endpoints on 1 to N networks.
- Endpoint – interfaces, switches, ports, etc and belong to only one network at a time
- Network – a collection of endpoints that can communicate directly (bridges, VLANS, etc) and can consist of 1 to N endpoints

The Container Network Model

Basic Container Network Model – Overview



The Container Network Model

The CNM and Docker Daemon interface at multiple points in a container's lifecycle depending on its implementation as a single container, single host or a multi-replica service communicating in an overlay across a Docker Swarm cluster.

Objects inside the network model include:

- Network Controller
- Driver
- Network
- Endpoint
- Sandbox

Each of those having (potentially) options and labels, and interacting with each other as specified.

The Container Network Model

Docker Networking and IPAM (Internet Protocol Address Management)

Managing addresses across multiple hosts on separate physical networks while providing routing to the underlying swarm networks externally is ‘the IPAM problem’ for Docker (and any other container cluster management system).

Depending on the network driver chose, IPAM is handled at different layers in the stack. On a single host, IPAM is not as challenging and routing is generally handled manually or through port exposure and each network is specific to the host system.

Network drivers enable IPAM through DHCP drivers or plugin drivers so that complex implementations support what would normally be overlapping addresses.