Introduction to Docker | PACKT Books

In this article by **Rajdeep Dua**, the author of the book Learning Docker Networking, we will look at an introduction of Docker networking and its components.

(For more resources related to this topic, see here.*)*

Docker is a lightweight containerization technology that has gathered enormous interest in recent years. It neatly bundles various Linux kernel features and services, such as namespaces, cgroups, SELinux, and AppArmor profiles, over union filesystems such as AUFS and BTRFS in order to make modular images. These images provide a highly configurable virtualized environment for applications and follow a **write once, run anywhere** workflow. Applications can be as simple as running a process to a highly scalable and distributed one. Therefore, there is a need for powerful networking elements that can support various complex use cases.

Each Docker container has its own network stack, and this is due to the Linux kernel's net namespace, where a new net namespace for each container is instantiated and cannot be seen from outside the container or from other containers.

Docker networking is powered by the following network components and services.

Linux bridges

These are L2/MAC learning switches built into the kernel and are to be used for forwarding.

Open vSwitch

This is an advanced bridge that is programmable and supports tunneling.

NAT

Network address translators are immediate entities that translate IP addresses and ports (SNAT, DNAT, and so on).

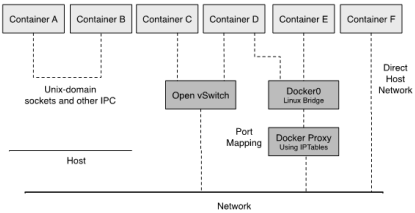
IPtables

This is a policy engine in the kernel used for managing packet forwarding, firewall, and NAT features.

AppArmor/SELinux

Firewall policies for each application can be defined with these.

Various networking components can be used to work with Docker, providing new ways to access and use Docker-based services. As a result, we see a lot of libraries that follow a different approach to networking. Some of the prominent ones are Docker Compose, Weave, Kubernetes, Pipework, and Libnetwork. The following figure depicts the root ideas of Docker networking:



Docker networking modes

Docker networking is at a very nascent stage, and there are many interesting contributions from the developer community, such as Pipework, Weave, Clocker, and Kubernetes. Each of them reflects a different aspect of Docker networking. We will learn about them in later chapters. Docker, Inc. has also established a new project, where networking will be standardized. It is called **libnetwork**.

Libnetwork implements the **Container Network Model** (**CNM**), which formalizes the steps required to provide networking for containers while providing an abstraction that can be used to support multiple network drivers. The CNM is built on three main components—sandbox, endpoint, and network.

Sandbox

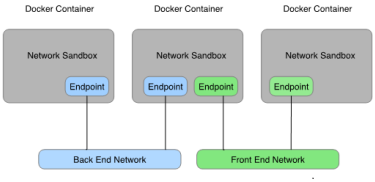
A sandbox contains the configuration of a container's network stack. This includes management of the container's interfaces, routing table, and DNS settings. An implementation of a sandbox could be a Linux network namespace, a FreeBSD jail, or other similar concept. A sandbox may contain many endpoints from multiple networks.

Endpoint

An endpoint connects a sandbox to a network. An implementation of an endpoint could be a veth pair, an Open vSwitch internal port, or something similar. An endpoint can belong to only one network but may only belong to one Sandbox.

Network

A network is a group of endpoints that are able to communicate with each other directly. An implementation of a network could be a Linux bridge, a VLAN, and so on. Networks consist of many endpoints, as shown in the following diagram:



The Docker CNM model

The CNM provides the following contract between networks and containers:

- All containers on the same network can communicate freely with each other
- Multiple networks are the way to segment traffic between containers and should be supported by all drivers
- Multiple endpoints per container are the way to join a container to multiple networks
- An endpoint is added to a network sandbox to provide it with network connectivity

In this article, we learned about the essential components of Docker networking, which have evolved from coupling simple Docker abstractions and powerful network components such as Linux bridges and Open vSwitch. We also talked about the next generation of Docker networking, which is called libnetwork.
