Container networking

Estimated reading time: 3 minutes

The type of network a container uses, whether it is a bridge (https://docs.docker.com/config/containers/bridges/), an overlay (https://docs.docker.com/config/containers/overlay/), a macvlan network (https://docs.docker.com/config/containers/macvlan/), or a custom network plugin, is transparent from within the container. From the container's point of view, it has a network interface with an IP address, a gateway, a routing table, DNS services, and other networking details (assuming the container is not using the `none` network driver). This topic is about networking concerns from the point of view of the container.

Published ports

By default, when you create a container, it does not publish any of its ports to the outside world. To make a port available to services outside of Docker, or to Docker containers which are not connected to the container's network, use the `--publish` or `-p` flag. This creates a firewall rule which maps a container port to a port on the Docker host. Here are some examples.

Flag value	Description
`-p 8080:80`	Map TCP port 80 in the container to port 8080 on the Docker host.
`-p 192.168.1.100:8080:80`	Map TCP port 80 in the container to port 8080 on the Docker host for connections to host IP 192.168.1.100.
`-p 8080:80/udp`	Map UDP port 80 in the container to port 8080 on the Docker host.
`-p 8080:80/tcp -p 8080:80/udp`	Map TCP port 80 in the container to TCP port 8080 on the Docker host, and map UDP port 80 in the container to UDP port 8080 on the Docker host.

IP address and hostname

By default, the container is assigned an IP address for every Docker network it connects to. The IP address is assigned from the pool assigned to the network, so the Docker daemon effectively acts as a DHCP server for each container. Each network also has a default subnet mask and gateway.

When the container starts, it can only be connected to a single network, using `--network` . However, you can connect a running container to multiple networks using `docker network connect` . When you start a container using the `--network` flag, you can specify the IP address assigned to the container on that network using the `--ip` or `--ip6` flags.

When you connect an existing container to a different network using `docker network connect` , you can use the `--ip` or `--ip6` flags on that command to specify the container's IP address on the additional network.

In the same way, a container's hostname defaults to be the container's ID in Docker. You can override the hostname using `--hostname` . When connecting to an existing network using `docker network connect` , you can use the `--alias` flag to specify an additional network alias for the container on that network.

DNS services

By default, a container inherits the DNS settings of the Docker daemon, including the `/etc/hosts` and `/etc/resolv.conf` .You can override these settings on a per-container basis.

Flag	Description
`--dns`	The IP address of a DNS server. To specify multiple DNS servers, use multiple `--dns` flags. If the container cannot reach any of the IP addresses you specify, Google's public DNS server `8.8.8.8` is added, so that your container can resolve internet domains.
`--dns-search`	A DNS search domain to search non-fully-qualified hostnames. To specify multiple DNS search prefixes, use multiple `--dns-search` flags.
`--dns-opt`	A key-value pair representing a DNS option and its value. See your operating system's documentation for `resolv.conf` for valid options.

Flag	Description
`--hostname`	The hostname a container uses for itself. Defaults to the container's ID if not specified.

Proxy server

If your container needs to use a proxy server, see Use a proxy server (https://docs.docker.com/network/proxy/).

networking (https://docs.docker.com/glossary/?term=networking), container (https://docs.docker.com/glossary/?term=container), standalone (https://docs.docker.com/glossary/?term=standalone)