Overview

Skill Level: Any Skill Level

OverviewUsing the default docker0 bridge and the port mapping works for most of the scenarios, but not all the scenarios, for example, you want to put all the docker containers in a flat network to provide full-access between the containers on different docker hosts. There are several ways to configure the docker multi-host networking, this […]

Ingredients

Basic docker knowledge

Step-by-step

Overview

Using the default docker0 bridge and the port mapping works for most of the scenarios, but not all the scenarios, for example, you want to put all the docker containers in a flat network to provide full-access between the containers on different docker hosts. There are several ways to configure the docker multi-host networking, this post will cover one of these ways: using Linux bridge to directly bridge the docker containers to the external network.

Create the linux bridge

```
brctl addbr br0
brctl addif br0 enp0s1
brctl setfd br0 0
ifconfig br0 10.0.189.109 netmask 255.255.0.0
```

The bridge configuration done by the commands above is not persistent through reboots, to make it be persistent, modify the network interface configuration files, this is an example on Ubuntu:

```
 root@docker:~# cat /etc/network/interfaces.d/br0
    auto br0
    iface br0 inet static
      address 10.0.189.109
      netmask 255.255.0.0
      gateway 10.0.0.57
      bridge_ports enp0s1
      bridge_fd 0
      bridge_hello 2
      bridge_maxage 12
      bridge_stp off
root@docker:~# service networking restart
```

When the bridge is created successfully, the brctl show command will show something like:

```
root@docker:~# brctl show br0
bridge name bridge id STP enabled interfaces
br0 8000.42570a00bd6d no enp0s1
root@docker:~#
```

Let docker to use the bridge

Option 1:

Create a new docker network to use this Linux bridge and explicilty specify –net with docker run command.

```
docker network create --driver=bridge --ip-range=10.0.190.0/24 --subnet=10.0.0.0/16 --aux-address='ip1=10.0.190.1' --aux-address='ip2=10.0.190.2' --aux-address='ip3=10.0.190.3' -o "com.docker.network.bridge.name=br0" br0

docker run --net=br-admin -it liguangcheng/ubuntu-16.04-ppc64el
```

Option 2:

Have docker to use the Linux bridge as default network.

Update /etc/default/docker with the following line:

```
DOCKER_OPTS="--bridge=br0 --fixed-cidr=10.0.190.0/24 --default-gateway=10.0.0.57"

service docker restart

docker run -it liguangcheng/ubuntu-16.04-ppc64el
```

Verify if the docker containers are connected to the bridge correctly

If the docker containers are connected to the bridge correctly, brctl show <bridge_name> will show new veth ports.

```
root@docker:~# brctl show br0
bridge name bridge id STP enabled interfaces
br0 8000.42570a00bd6d no enp0s1
                         veth335eaf4
root@docker:~#
```

docker inspect <containername> will show the right network information

```
 root@docker:~# docker inspect docker2

......

"Networks": {
   "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "48f33644c3903b6c1ef73c88e1a459aa1a3af61af17d8444c8cee66cede863fc",
      "EndpointID": "f3522847e621a83de3cce29592ddc5334ebc2b3d5d8516f9fe3a71ecc480d316",
      "Gateway": "10.0.0.57",
      "IPAddress": "10.0.190.1",
```

```
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:0a:00:be:01"
    }
......
root@docker:~#
```

Of course, login the docker container and verify if the network connection is working.

```
 root@docker:~# docker exec docker2 ping -c 1 www.ibm.com
PING e2874.x.akamaiedge.net (23.35.36.128) 56(84) bytes of data.
64 bytes from a23-35-36-128.deploy.static.akamaitechnologies.com (23.35.36.128): icmp_seq=1 ttl=40 time=230 ms

--- e2874.x.akamaiedge.net ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 230.755/230.755/230.755/0.000 ms
root@docker:~#
```


Tags docker, networking

by LIGUANGCHENG

2 comments on"Bridge the docker containers to external network"

1. 

   TomsFilatovs February 19, 2017
   Something I ran into following this guide was that Docker automatically adds iptables rules to isolate its networks and so, by making the machine’s physical network connection one of those networks, anything on the original docker bridge network was cut off from the physical one, which created problems as I was using the original network for testing.
   The solution was creating a modified systemd docker.service configuration file with ‘–iptables=false’ appended to the ‘ExecStart=…’ line and adding a rule needed for NAT of the original bridge network to the system iptables configuration by hand.

   Log in to Reply

- 

  TomsFilatovs February 19, 2017
  But I mean that problem arises -if like me- you want to be able to assign containers IPs in the whole network available to the physical host and so specify the same –ip-range and –subnet which I now see creates more problems than it solves.

  Log in to Reply