marcelwiget

Bridge Docker Container with host interface

Bridge Docker Container with host interface

Docker automatically creates a bridge docker0 and assigns IP addresses for Containers to this network. What if one wants to simply connect containers to on of the compute hosts networks? This can be achieved by creating and assigning a custom bridge to the Docke engine: https://docs.docker.com/engine/userguide/networking/default_network/custom-docker0/.
Here are my notes on how to deploy this on Ubuntu 14.04:

Create host bridge br0

Install bridge-utils, find out which interface and record its MAC address to attach to the bridge.

```
 sudo apt-get update
sudo apt-get install bridge-utils
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 0c:c4:7a:30:89:e8
          inet addr:10.0.0.32  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ec4:7aff:fe30:89e8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:92253 errors:0 dropped:0 overruns:0 frame:0
          TX packets:42687 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:89831251 (89.8 MB)  TX bytes:20941057 (20.9 MB)
          Memory:df360000-df37ffff
```

Modify /etc/network/interfaces by removing the settings for eth0 and add the settings for the new bridge br0, assign the MAC address of eth0 to the bridge. While not strictly required, it makes the change transparent to external hosts and DHCP can still be used to assign the same IP address to the bridge br0 as was assigned to eth0:

```
 # The primary network interface
auto eth0
#iface eth0 inet dhcp
#
# br0 for Docker engine
auto br0
iface br0 inet dhcp
  bridge_ports eth0
  bridge_stp off
  bridge_hw 0c:c4:7a:30:89:e8

iface br0 inet6 dhcp
```

Activate the new bridge. While this worked for me without a reboot, I had to open a new ssh session to the host after dhcp completed:

```
 sudo ifdown eth0 && sudo ifup br0
Ignoring unknown interface eth0=eth0.

Waiting for br0 to get ready (MAXWAIT is 32 seconds).
Internet Systems Consortium DHCP Client 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/br0/0c:c4:7a:30:89:e8
Sending on   LPF/br0/0c:c4:7a:30:89:e8
Sending on   Socket/fallback
DHCPDISCOVER on br0 to 255.255.255.255 port 67 interval 3 (xid=0xd9b48431)
DHCPREQUEST of 10.0.0.32 on br0 to 255.255.255.255 port 67 (xid=0x3184b4d9)
DHCPOFFER of 10.0.0.32 from 10.0.0.1
DHCPACK of 10.0.0.32 from 10.0.0.1
bound to 10.0.0.32 -- renewal in 33116 seconds.
```

Log back in and check the new bridge:

```
 ifconfig br0
br0       Link encap:Ethernet  HWaddr 0c:c4:7a:30:89:e8
          inet addr:10.0.0.32  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::1/64 Scope:Link
          inet6 addr: fe80::ec4:7aff:fe30:89e8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:759 errors:0 dropped:0 overruns:0 frame:0
          TX packets:558 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:112071 (112.0 KB)  TX bytes:87466 (87.4 KB)
```

Configure docker-engine for br0

Decide what IP subnet range from the br0's subnet will be dedicated to Docker Containers. Its currently not possible to simply define an IP range, at least not with Docker version 1.10.3, so pick a range defined by a subnet mask. In my case I picked 10.0.0.128/26, which gives me the range of 10.0.0.128 to 10.0.0.191. Yes, that includes the subnet broadcast and network address, because its simply bridged to the actual IP sbnet of br0.

Add the following line to /etc/default/docker:

```
 DOCKER_OPTS="-b=br0 --ipv6 --fixed-cidr=10.0.0.128/26"
```

Stop and remove all running and stopped containers, otherwise Docker might refuse to start when an existing Container has an IP outside the assigned range.

Restart Docker and check that it is actually running:

```
 mwiget@sm:~$ sudo service docker restart
docker stop/waiting
docker start/running, process 20733
mwiget@sm:~$ sudo service docker status
docker start/running, process 20733
```

All set, though its advisable to reboot the host and see if it will survive a reboot. Be prepared to access the console if networking fails.

Launch a Container and check its IP address:

```
 docker run --name ubuntu1 -ti ubuntu:14.04
Unable to find image 'ubuntu:14.04' locally
14.04: Pulling from library/ubuntu
203137e8afd5: Pull complete
2ff1bbbe9310: Pull complete
933ae2486129: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:1c8b813b6b6656e9a654bdf29a7decfcc73b92a62b934adc4253b0dc2be9d0a2
Status: Downloaded newer image for ubuntu:14.04
root@9645410940ea:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:0a:00:00:80
          inet addr:10.0.0.128  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::242:a00:80/64 Scope:Link
          inet6 addr: fe80::42:aff:fe00:80/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:758 (758.0 B)  TX bytes:648 (648.0 B)

root@9645410940ea:/#
root@9645410940ea:/# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=255 time=2.96 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=255 time=1.25 ms
^C
--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.250/2.106/2.962/0.856 ms
root@9645410940ea:/#
```

Bingo!

March 20, 2016
Uncategorized

docker, ubuntu

Published by mwiget

View all posts by mwiget

3 thoughts on “Bridge Docker Container with host interface”

1. **Xem**

   says:
   April 8, 2017 at 7:43 pm
   Hey there,
   Container gets IP as host subnet as you mentioned. But container could not ping to other IPs except docker host 😦
   Any idea what wrong?

   Thank You!

   1. **mwiget**

      says:
      April 9, 2017 at 4:56 am
      Hmm, maybe some iptables either blocking or nat’ing the containers traffic? You can run tcpdump on br0 to see what might be going on, then also on eth0 (or whatever your host interface is).

2. **mwiget**

   says:
   May 8, 2017 at 9:30 am
   While using my own hint on a ubuntu 16.04 server with many ethernet ports (some unconfigured), I faced a 5 minutes bootup delay:

   “A start job is running for raise network interfaces (5 mins 3 sec)”

   Google’d it and found a workaround:
   https://ubuntuforums.org/showthread.php?t=2323253

   Disabled the service and also reduced the timeout (just in case ;-)):

   sudo systemctl disable systemd-networkd-wait-online.service

   Also set the timeout down to 10 seconds in
   sudo vi /etc/systemd/system/network-online.target.wants/networking.service
   TimeoutStartSec=10sec

   That did the trick for me.

UP ↑