Docker 1.10 with Fan Networking in Ubuntu 16.04

I'm thrilled to introduce Docker 1.10.3, available on every Ubuntu architecture, for Ubuntu 16.04 LTS, and announce the General Availability of Ubuntu Fan Networking!

That's Ubuntu Docker binaries and Ubuntu Docker images for:

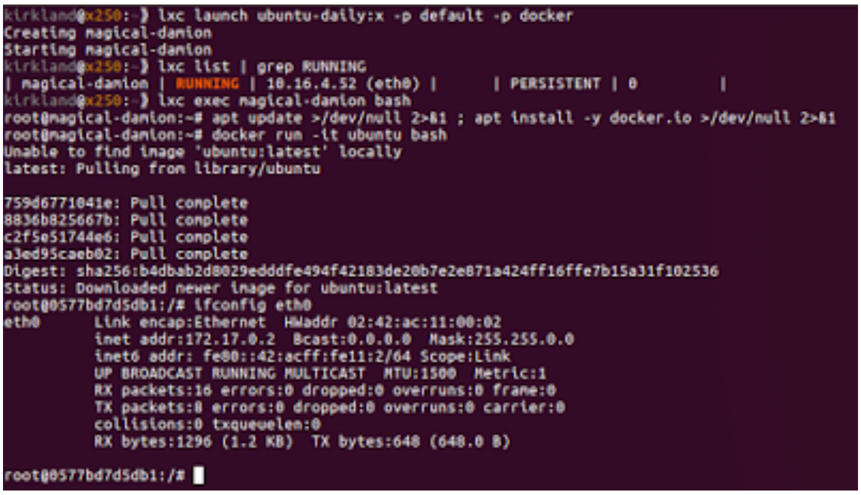
- *armhf* (rpi2, et al. IoT devices)
- *arm64* (Cavium, et al. servers)
- *i686* (does anyone seriously still run 32-bit intel servers?)
- *amd64* (most servers and clouds under the sun)
- *ppc64el* (OpenPower and IBM POWER8 machine learning super servers)
- *s390x* (IBM System Z LinuxOne super uptime mainframes)

That's Docker-Docker-Docker-Docker-Docker-Docker, from the smallest Raspberry Pi's to the biggest IBM mainframes in the world today! Never more than one '*sudo apt install docker.io*' command away.

Moreover, we now have Docker running inside of LXD! Containers all the way down. Application containers (e.g. Docker), inside of Machine containers (e.g. LXD), inside of Virtual Machines (e.g. KVM), inside of a public or private cloud (e.g. Azure, OpenStack), running on bare metal (take your pick).

Let's have a look at launching a Docker application container inside of a LXD machine container:

```
kirkland@x250:~⟫ lxc launch ubuntu-daily:x -p default -p docker
Creating magical-damion
Starting magical-damion
kirkland@x250:~⟫ lxc list | grep RUNNING
| magical-damion | RUNNING | 10.16.4.52 (eth0) |      | PERSISTENT | 0          |
kirkland@x250:~》 lxc exec magical-damion bash
root@magical-damion:~# apt update >/dev/null 2>&1 ; apt install -y docker.io >/dev/null
root@magical-damion:~# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
759d6771041e: Pull complete
8836b825667b: Pull complete
c2f5e51744e6: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:b4dbab2d8029edddfe494f42183de20b7e2e871a424ff16ffe7b15a31f102536
Status: Downloaded newer image for ubuntu:latest
root@0577bd7d5db1:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 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:1296 (1.2 KB)  TX bytes:648 (648.0 B)
```



```
kirkland@x250:~❯ lxc launch ubuntu-daily:x -p default -p docker
Creating magical-damion
Starting magical-damion
kirkland@x250:~❯ lxc list | grep RUNNING
| magical-damion | RUNNING | 10.16.4.52 (eth0) |      | PERSISTENT | 0         |
kirkland@x250:~❯ lxc exec magical-damion bash
root@magical-damion:~# apt update >/dev/null 2>&1 ; apt install -y docker.io >/dev/null 2>&1
root@magical-damion:~# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu

759d6771041e: Pull complete
8836b825667b: Pull complete
c2f5e51744e6: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:b4dbab2d8029edddfe494f42183de20b7e2e871a424ff16ffe7b15a31f102536
Status: Downloaded newer image for ubuntu:latest
root@0577bd7d5db1:/# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:ac:11:00:02
          inet addr:172.17.0.2  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::42:acff:fe11:2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16 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:1296 (1.2 KB)  TX bytes:648 (648.0 B)

root@0577bd7d5db1:/# █
```

Oh, and let's talk about networking… We're also pleased to announce the general availability of Ubuntu Fan networking — specially designed to connect all of your Docker containers spread across your network. Ubuntu's Fan networking feature is an easy way to make every Docker container on your local network easily addressable by every other Docker host and container on the same network. It's high performance, super simple, utterly deterministic, and we've tested it on every major public cloud as well as OpenStack and our private networks.

Simply installing Ubuntu's Docker package will also install the [ubuntu-fan](#) package, which provides an interactive setup script, [*fanatic*](#), should you choose to join the Fan. Simply run '*sudo fanatic*' and answer the questions. You can trivially revert your Fan networking setup easily with '*sudo fanatic deconfigure*'.

```
kirkland@x250:~$ sudo fanatic
Welcome to the fanatic fan networking wizard.  This will help you set
up an example fan network and optionally configure docker and/or LXD to
use this network.  See fanatic(1) for more details.
Configure fan underlay (hit return to accept, or specify alternative) [10.0.0.0/16]:
Configure fan overlay (hit return to accept, or specify alternative) [250.0.0.0/8]:
Create LXD networking for underlay:10.0.0.0/16 overlay:250.0.0.0/8 [Yn]: n
Create docker networking for underlay:10.0.0.0/16 overlay:250.0.0.0/8 [Yn]: Y
Test docker networking for underlay:10.0.0.45/16 overlay:250.0.0.0/8
(NOTE: potentially triggers large image downloads) [Yn]: Y
local docker test: creating test container ...
34710d2c9a856f4cd7d8aa10011d4d2b3d893d1c3551a870bdb9258b8f583246
test master: ping test (250.0.45.0) ...
test slave: ping test (250.0.45.1) ...
test master: ping test ... PASS
test master: short data test (250.0.45.1 -> 250.0.45.0) ...
test slave: ping test ... PASS
test slave: short data test (250.0.45.0 -> 250.0.45.1) ...
test master: short data ... PASS
test slave: short data ... PASS
test slave: long data test (250.0.45.0 -> 250.0.45.1) ...
test master: long data test (250.0.45.1 -> 250.0.45.0) ...
test master: long data ... PASS
test slave: long data ... PASS
local docker test: destroying test container ...
fanatic-test
fanatic-test
local docker test: test complete PASS (master=0 slave=0)
This host IP address: 10.0.0.45
```

I've run '*sudo fanatic*' here on a couple of machines on my network — *x250* (10.0.0.45) and *masterbr* (10.0.0.8), and now I'm going to launch a Docker container on each of those two machines, obtain each IP address on the Fan (250.x.y.z), install *iperf*, and test the connectivity and bandwidth between each of them (on my gigabit home network). You'll see that we'll get 900mbps+ of throughput:

```
kirkland@x250:~⟫ sudo docker run -it ubuntu bash
root@c22cf0d8e1f7:/#  apt update >/dev/null 2>&1 ; apt install -y iperf >/dev/null 2>&1
root@c22cf0d8e1f7:/#  ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:fa:00:2d:00
          inet addr:250.0.45.0  Bcast:0.0.0.0  Mask:255.0.0.0
          inet6 addr: fe80::42:faff:fe00:2d00/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:6423 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4120 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22065202 (22.0 MB)  TX bytes:227225 (227.2 KB)
```

root@c22cf0d8e1f7:/# iperf -c 250.0.8.0
multicast ttl failed: Invalid argument

Client connecting to 250.0.8.0, TCP port 5001
TCP window size: 45.0 KByte (default)

---

[3] local 250.0.45.0 port 54274 connected with 250.0.8.0 port 5001
[ID] Interval Transfer Bandwidth
[3] 0.0-10.0 sec 1.05 GBytes 902 Mbits/sec

And the second machine:

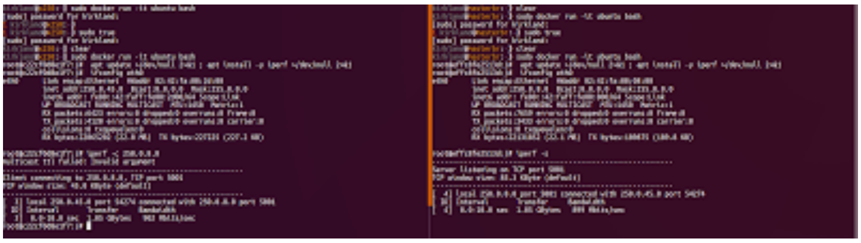
```
kirkland@masterbr:~⟫ sudo docker run -it ubuntu bash
root@effc8fe2513d:/#  apt update >/dev/null 2>&1 ; apt install -y iperf >/dev/null 2>&1
root@effc8fe2513d:/#  ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:fa:00:08:00
          inet addr:250.0.8.0  Bcast:0.0.0.0  Mask:255.0.0.0
          inet6 addr: fe80::42:faff:fe00:800/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:7659 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3433 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:22131852 (22.1 MB)  TX bytes:189875 (189.8 KB)
```

root@effc8fe2513d:/# iperf -s

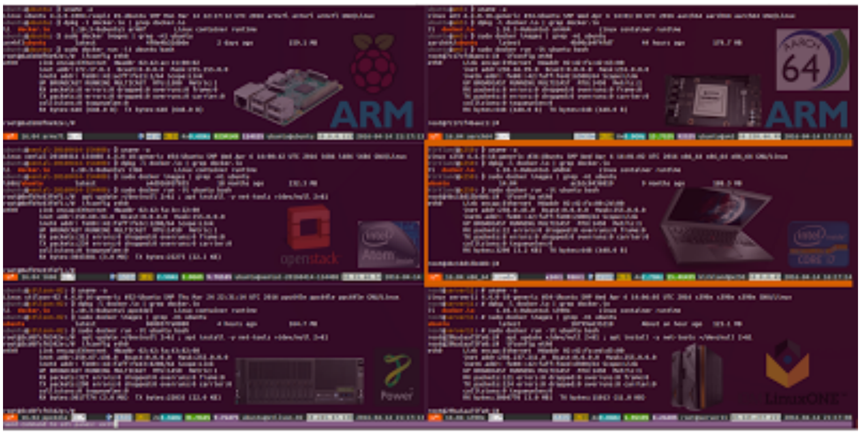
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)

---

[4] local 250.0.8.0 port 5001 connected with 250.0.45.0 port 54274
[ID] Interval Transfer Bandwidth
[4] 0.0-10.0 sec 1.05 GBytes 899 Mbits/sec



Finally, let's have another long hard look at the image from the top of this post. Download it in full resolution to study very carefully what's happening here, because it's pretty [*redacted*] amazing!



Here, we have a Byobu session, split into 6 panes (*Shift-F2* 5x Times, *Shift-F8* 6x times). In each pane, we have an SSH session to Ubuntu 16.04 LTS servers spread across 6 different architectures — *armhf*, *arm64*, *i686*, *amd64*, *ppc64el*, and *s390x*. I used the *Shift-F9* key to simultaneously run the same commands in each and every window. Here are the commands I ran:

```
clear
lxc launch ubuntu-daily:x -p default -p docker
lxc list | grep RUNNING
uname -a
dpkg -l docker.io | grep docker.io
sudo docker images | grep -m1 ubuntu
sudo docker run -it ubuntu bash
 apt update >/dev/null 2>&1 ; apt install -y net-tools >/dev/null 2>&1
 ifconfig eth0
 exit
```

That's right. We just launched Ubuntu LXD containers, as well as Docker containers against every one of Ubuntu's available architectures. How's that for Ubuntu everywhere!?!

Ubuntu 16.04 LTS will be one hell of a release!

Dustin Kirkland is part of Canonical's Ubuntu Product and Strategy team, working for Mark Shuttleworth, and leading the technical strategy, road map, and life cycle of the Ubuntu Cloud and IoT commercial offerings. Formerly the CTO of Gazzang, a venture funded start-up acquired by Cloudera, Dustin designed and implemented an innovative key management system for the cloud, called zTrustee, and delivered comprehensive security for cloud and big data platforms with eCryptfs and other encryption technologies. Dustin is an active Core Developer of the Ubuntu Linux distribution, maintainer of 20+ open source projects, and the creator of Byobu, DivItUp.com, and LinuxSearch.org. A Fightin' Texas Aggie Class of 2001 graduate, Dustin lives in Austin, Texas, with his wife Kim, daughters, and his Australian Shepherds, Aggie and Tiger. Dustin is also an avid home brewer.