Docker Networking Lab

Maciej Cyranowicz
172289

1. Networking Basics

a. List networks

```
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
d60f3e7e39f0        bridge              bridge              local
fb2c7ca007c4        host                host                local
4fa0995f67d2        none                null                local
```

b. Inspect a network

```
$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "d60f3e7e39f0dc02002ed04b311a2e6eead1ba05deca111d8a9dabfa7cbaa0d7",
        "Created": "2018-12-09T18:38:53.0198232322Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

c. List network driver plugins

```
$ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 18.06.1-ce
Storage Driver: overlay2
 Backing Filesystem: xfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 468a545b9edcd5932818eb9de8e72413e616e86e
runc version: 69663f0bd4b60df09991c08812a60108003fa340
init version: fec3683
Security Options:
 apparmor
 seccomp
  Profile: default
Kernel Version: 4.4.0-96-generic
Operating System: Alpine Linux v3.8 (containerized)
OSType: linux
Architecture: x86_64
CPUs: 8
Total Memory: 31.4GiB
Name: node1
ID: EYRA:YHIV:V2Z3:ZC7A:VDQ4:4XGT:F3ZU:HP5A:57PH:NLHP:P4TS:AOUU
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 23
 Goroutines: 45
 System Time: 2018-12-09T18:41:34.477755332Z
 EventsListeners: 0
Registry: https://index.docker.io/v1/
Labels:
Experimental: true
Insecure Registries:
 127.0.0.1
 127.0.0.0/8
Live Restore Enabled: false

WARNING: No swap limit support
WARNING: bridge-nf-call-iptables is disabled
WARNING: bridge-nf-call-ip6tables is disabled
```

2. Bridge networking

a. The basics

```
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
d60f3e7e39f0        bridge              bridge              local
fb2c7ca007c4        host                host                local
4fa0995f67d2        none                null                local
[node1] (local) root@192.168.0.48 ~
$ apk update
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/community/x86_64/APKINDEX.tar.gz
v3.8.1-142-g179c136670 [http://dl-cdn.alpinelinux.org/alpine/v3.8/main]
v3.8.1-142-g179c136670 [http://dl-cdn.alpinelinux.org/alpine/v3.8/community]
OK: 9558 distinct packages available
[node1] (local) root@192.168.0.48 ~
$ apk add bridge
(1/1) Installing bridge (1.5-r3)
OK: 302 MiB in 111 packages
[node1] (local) root@192.168.0.48 ~
$ brctl show
bridge name         bridge id               STP enabled     interfaces
docker0             8000.0242c7ec657d       no
```

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:c7:ec:65:7d brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
14180: eth0@if14181: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether ba:d3:39:ca:60:50 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.48/23 scope global eth0
       valid_lft forever preferred_lft forever
14184: eth1@if14185: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue state UP
    link/ether 02:42:ac:12:00:0a brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.10/16 scope global eth1
       valid_lft forever preferred_lft forever
```

b. Connect a container

```
$ docker run -dt ubuntu sleep infinity
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
32802c0cfa4d: Pull complete
da1315cffa03: Pull complete
fa83472a3562: Pull complete
f85999a86bef: Pull complete
Digest: sha256:6d0e0c26489e33f5a6f0020edface2727db9489744ecc9b4f50c7fa671f23c49
Status: Downloaded newer image for ubuntu:latest
87dc9eedb85b0af7c6bb7695c535f7f6cca9e33081034b63a7e81feedf63a1f4
```

```
$ docker ps
CONTAINER ID        IMAGE          COMMAND             CREATED            STATUS           PORTS           NAMES
87dc9eedb85b        ubuntu         "sleep infinity"    22 seconds ago     Up 21 seconds                    fervent_lamport
```

```
$ brctl show
bridge name     bridge id           STP enabled     interfaces
docker0         8000.0242c7ec657d   no              veth5f59d66
```

```
$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "d60f3e7e39f0dc02002ed04b311a2e6eead1ba05deca111d8a9dabfa7cbaa0d7",
        "Created": "2018-12-09T18:38:53.0198232322Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "87dc9eedb85b0af7c6bb7695c535f7f6cca9e33081034b63a7e81feedf63a1f4": {
                "Name": "fervent_lamport",
                "EndpointID": "db0d85dd8b4ce4aa1d8b2e4c9bb3ad99e9768ff40fd53a98320df08b4f9f6db7",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

c. Test network connectivity

```
$ docker ps
CONTAINER ID    IMAGE       COMMAND             CREATED         STATUS          PORTS       NAMES
87dc9eedb85b    ubuntu      "sleep infinity"    2 minutes ago   Up 2 minutes                fervent_lamport
```

```
$ docker exec -i -t 87dc9eedb85b /bin/bash
root@87dc9eedb85b:/# echo test
test
root@87dc9eedb85b:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [130 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [277 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1367 B]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.7 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [753 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6931 B]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [589 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3655 B]
Fetched 15.1 MB in 2s (6101 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcap2-bin libidn11 libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2 libcap2-bin libidn11 libpam-cap
0 upgraded, 5 newly installed, 0 to remove and 4 not upgraded.
Need to get 140 kB of archives.
After this operation, 537 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcap2 amd64 1:2.25-1.2 [13.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libidn11 amd64 1.33-2.1ubuntu1 [45.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 iputils-ping amd64 3:20161105-1ubuntu2 [53.9 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcap2-bin amd64 1:2.25-1.2 [20.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpam-cap amd64 1:2.25-1.2 [7268 B]
Fetched 140 kB in 0s (297 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libcap2:amd64.
(Reading database ... 4038 files and directories currently installed.)
```

```
Preparing to unpack .../libcap2_1%3a2.25-1.2_amd64.deb ...
Unpacking libcap2:amd64 (1:2.25-1.2) ...
Selecting previously unselected package libidn11:amd64.
Preparing to unpack .../libidn11_1.33-2.1ubuntu1_amd64.deb ...
Unpacking libidn11:amd64 (1.33-2.1ubuntu1) ...
Selecting previously unselected package iputils-ping.
Preparing to unpack .../iputils-ping_3%3a20161105-1ubuntu2_amd64.deb ...
Unpacking iputils-ping (3:20161105-1ubuntu2) ...
Selecting previously unselected package libcap2-bin.
Preparing to unpack .../libcap2-bin_1%3a2.25-1.2_amd64.deb ...
Unpacking libcap2-bin (1:2.25-1.2) ...
Selecting previously unselected package libpam-cap:amd64.
Preparing to unpack .../libpam-cap_1%3a2.25-1.2_amd64.deb ...
Unpacking libpam-cap:amd64 (1:2.25-1.2) ...
Setting up libcap2:amd64 (1:2.25-1.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Setting up libidn11:amd64 (1.33-2.1ubuntu1) ...
Setting up iputils-ping (3:20161105-1ubuntu2) ...
Setting up libpam-cap:amd64 (1:2.25-1.2) ...
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (Can't locate Term/ReadLine.pm in @INC (you may need to install the Term::ReadLine module) (@INC contains: /etc/perl /usr/local/lib/x86_64-linux-gnu/perl/5.26.1 /usr/local/share/perl/5.26.1 /usr/lib/x86_64-li
nux-gnu/perl5/5.26 /usr/share/perl5 /usr/lib/x86_64-linux-gnu/perl/5.26 /usr/share/perl/5.26 /usr/local/lib/site_perl /usr/lib/x86_64-linux-gnu/perl-base) at /usr/share/perl5/Debconf/FrontEnd/Readline.pm line 7.)
debconf: falling back to frontend: Teletype
Setting up libcap2-bin (1:2.25-1.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
```

```
root@87dc9eedb85b:/# ping -c5 www.github.com
PING github.com (192.30.253.113) 56(84) bytes of data.
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=1 ttl=50 time=1.81 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=2 ttl=50 time=1.82 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=3 ttl=50 time=1.82 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=4 ttl=50 time=1.85 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=5 ttl=50 time=1.91 ms

--- github.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 1.813/1.845/1.913/0.036 ms
```

```
$ docker stop fervent_lamport
fervent_lamport
```

d. Configure NAT for external connectivity

```
$ docker run --name web1 -d -p 8080:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a5a6f2f73cd8: Pull complete
1ba02017c4b2: Pull complete
33b176c904de: Pull complete
Digest: sha256:5d32f60db294b5deb55d078cd4feb410ad88e6fe77500c87d3970eca97f54dba
Status: Downloaded newer image for nginx:latest
c86fd25e286b3b0b84c58dc3fa032b6aae955c7197cecc77240f9586e82603d4
```

```
$ docker ps
CONTAINER ID      IMAGE         COMMAND                CREATED         STATUS          PORTS                    NAMES
c86fd25e286b      nginx         "nginx -g 'daemon of…"  20 seconds ago  Up 20 seconds   0.0.0.0:8080->80/tcp     web1
```

```
$ curl 127.0.0.1:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

3. Overlay networking

a. The basics

```
$ docker swarm init --advertise-addr $(hostname -i)
Swarm initialized: current node (x8hdgv1nxoqoiwsm663th5ci0) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-5bfcynanfqdx23n05lnql4adycvqc28kifsu8qu5saskmr0kce-60li8cl0x73uq5wrryppajzwd 192.168.0.48:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

```
$ docker swarm join --token SWMTKN-1-5bfcynanfqdx23n05lnql4adycvqc28kifsu8qu5saskmr0kce-60li8cl0x73uq5wrryppajzwd 192.168.0.48:2377
This node joined a swarm as a worker.
```

```
$ docker node ls
ID                             HOSTNAME     STATUS      AVAILABILITY    MANAGER STATUS    ENGINE VERSION
x8hdgv1nxoqoiwsm663th5ci0 *    node1        Ready       Active          Leader            18.06.1-ce
6xqfzxcget1bqub9da9u9hzpg      node2        Ready       Active                            18.06.1-ce
```

b. Create an overlay network

```
$ docker network create -d overlay overnet
pyj7lyknuwtm4fksopgdhhbq4
```

```
[node1] (local) root@192.168.0.48 ~
$ docker network ls
NETWORK ID        NAME                DRIVER            SCOPE
d60f3e7e39f0      bridge              bridge            local
75af45c88750      docker_gwbridge     bridge            local
fb2c7ca007c4      host                host              local
deev51yi2os5      ingress             overlay           swarm
4fa0995f67d2      none                null              local
pyj7lyknuwtm      overnet             overlay           swarm
```

```
[node2] (local) root@192.168.0.47 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
df9441ce5107        bridge              bridge              local
11c5adede19e        docker_gwbridge     bridge              local
f2863f2a1831        host                host                local
deev51yi2os5        ingress             overlay             swarm
fb0140b41d11        none                null                local
```

```
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "pyj7lyknuwtm4fksopgdhhbq4",
        "Created": "2018-12-09T19:26:57.537419469Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.0.0/24",
                    "Gateway": "10.0.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": null,
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": null
    }
]
```

c. Create a service

```
$ docker service create --name myservice \
> --network overnet \
> --replicas 2 \
> ubuntu sleep infinity
q1uoep7vxqkzn3zykq2eyrbk0
overall progress: 2 out of 2 tasks
1/2: running   [==================================================>]
2/2: running   [==================================================>]
verify: Service converged
```

```
$ docker service ls
ID                  NAME                MODE                REPLICAS            IMAGE               PORTS
q1uoep7vxqkz        myservice           replicated          2/2                 ubuntu:latest
```

```
$ docker service ps myservice
ID                  NAME                IMAGE               NODE                DESIRED STATE       CURRENT STATE              ERROR               PORTS
b6kif6ge8x0g        myservice.1         ubuntu:latest       node1               Running             Running about a minute ago
puyw3eudt7wu        myservice.2         ubuntu:latest       node2               Running             Running about a minute ago
```

```
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
d60f3e7e39f0        bridge              bridge              local
75af45c88750        docker_gwbridge     bridge              local
fb2c7ca007c4        host                host                local
deev51yi2os5        ingress             overlay             swarm
4fa0995f67d2        none                null                local
pyj7lyknuwtm        overnet             overlay             swarm
```

d. Test the network

```
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "pyj7lyknuwtm4fksopgdhhbq4",
        "Created": "2018-12-09T19:30:09.328135557Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "10.0.0.0/24",
                    "Gateway": "10.0.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "9fb05a85f60f6dd2587630f49e71dd2dfa08fb6cbfad836b99f2ee99d01e2ce5": {
                "Name": "myservice.1.b6kif6ge8x0g9eul1wd5u33bf",
                "EndpointID": "90057354efb2526a67ee9c81cfe7b0dddc0d465286065a02a340df1193236ef9",
                "MacAddress": "02:42:0a:00:00:05",
                "IPv4Address": "10.0.0.5/24",
                "IPv6Address": ""
            },
            "lb-overnet": {
                "Name": "overnet-endpoint",
                "EndpointID": "83fafe3e499bd16ae7b40678045f9934cd8260dcd1bbee0d3759bf02c37772c5",
                "MacAddress": "02:42:0a:00:00:03",
                "IPv4Address": "10.0.0.3/24",
```

```
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": {},
        "Peers": [
            {
                "Name": "498bcbbf67b1",
                "IP": "192.168.0.48"
            },
            {
                "Name": "ffa8ab891c7d",
                "IP": "192.168.0.47"
            }
        ]
    }
]
```

```
$ docker exec -i -t 9fb05a85f60f /bin/bash
root@9fb05a85f60f:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [130 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [277 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:9 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1367 B]
Get:10 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.7 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6931 B]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [589 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [753 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3655 B]
Fetched 15.1 MB in 3s (6038 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcap2 libcap2-bin libidn11 libpam-cap
The following NEW packages will be installed:
  iputils-ping libcap2 libcap2-bin libidn11 libpam-cap
0 upgraded, 5 newly installed, 0 to remove and 4 not upgraded.
Need to get 140 kB of archives.
After this operation, 537 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcap2 amd64 1:2.25-1.2 [13.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 libidn11 amd64 1.33-2.1ubuntu1 [45.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 iputils-ping amd64 3:20161105-1ubuntu2 [53.9 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 libcap2-bin amd64 1:2.25-1.2 [20.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpam-cap amd64 1:2.25-1.2 [7268 B]
Fetched 140 kB in 0s (311 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libcap2:amd64.
(Reading database ... 4038 files and directories currently installed.)
```

```
root@9fb05a85f60f:/# ping -c5 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.106 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.067 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.049 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.060 ms

--- 10.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3996ms
rtt min/avg/max/mdev = 0.049/0.069/0.106/0.021 ms
```

e. Test service discovery

```
root@9fb05a85f60f:/# cat /etc/resolv.conf
search 51ur3jppi0eupdptvsj42kdvgc.bx.internal.cloudapp.net
nameserver 127.0.0.11
options ndots:0
```

```
root@9fb05a85f60f:/# ping -c5 myservice
PING myservice (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=1 ttl=64 time=0.118 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=2 ttl=64 time=0.078 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=4 ttl=64 time=0.070 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=5 ttl=64 time=0.077 ms

--- myservice ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 0.070/0.088/0.118/0.021 ms
```

```
$ docker service inspect myservice
[
    {
        "ID": "q1uoep7vxqkzn3zykq2eyrbk0",
        "Version": {
            "Index": 21
        },
        "CreatedAt": "2018-12-09T19:30:09.165080219Z",
        "UpdatedAt": "2018-12-09T19:30:09.166714511Z",
        "Spec": {
            "Name": "myservice",
            "Labels": {},
            "TaskTemplate": {
                "ContainerSpec": {
                    "Image": "ubuntu:latest@sha256:6d0e0c26489e33f5a6f0020edface2727db9489744ecc9b4f50c7fa671f23c49",
                    "Args": [
                        "sleep",
                        "infinity"
                    ],
                    "Init": false,
                    "StopGracePeriod": 10000000000,
                    "DNSConfig": {},
                    "Isolation": "default"
                },
                "Resources": {
                    "Limits": {},
                    "Reservations": {}
                },
                "RestartPolicy": {
                    "Condition": "any",
                    "Delay": 5000000000,
                    "MaxAttempts": 0
                },
                "Placement": {
                    "Platforms": [
                        {
                            "Architecture": "amd64",
                            "OS": "linux"
                        },
                        {
                            "OS": "linux"
                        },
```

```json
                    {
                        "Architecture": "arm64",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "386",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "ppc64le",
                        "OS": "linux"
                    },
                    {
                        "Architecture": "s390x",
                        "OS": "linux"
                    }
                ]
            },
            "Networks": [
                {
                    "Target": "pyj7lyknuwtm4fksopgdhhbq4"
                }
            ],
            "ForceUpdate": 0,
            "Runtime": "container"
        },
        "Mode": {
            "Replicated": {
                "Replicas": 2
            }
        },
        "UpdateConfig": {
            "Parallelism": 1,
            "FailureAction": "pause",
            "Monitor": 5000000000,
            "MaxFailureRatio": 0,
            "Order": "stop-first"
        },
        "RollbackConfig": {
            "Parallelism": 1,
            "FailureAction": "pause",
            "Monitor": 5000000000,
            "MaxFailureRatio": 0,
            "Order": "stop-first"
        },
        "EndpointSpec": {
            "Mode": "vip"
        }
    },
    "Endpoint": {
        "Spec": {
            "Mode": "vip"
        },
        "VirtualIPs": [
            {
                "NetworkID": "pyj7lyknuwtm4fksopgdhhbq4",
                "Addr": "10.0.0.4/24"
            }
        ]
    }
}
]
```

4. Clean up

```
$ docker service rm myservice
myservice
[node1] (local) root@192.168.0.48 ~
$ docker ps
CONTAINER ID        IMAGE            COMMAND                  CREATED          STATUS          PORTS                    NAMES
9fb05a85f60f        ubuntu:latest    "sleep infinity"         13 minutes ago   Up 13 minutes                            myservice.1.b6kif6ge8x0g9eul1wd5u33bf
c86fd25e286b        nginx            "nginx -g 'daemon of…"   29 minutes ago   Up 29 minutes   0.0.0.0:8080->80/tcp     web1
[node1] (local) root@192.168.0.48 ~
$ docker swarm leave --force
Node left the swarm.
```