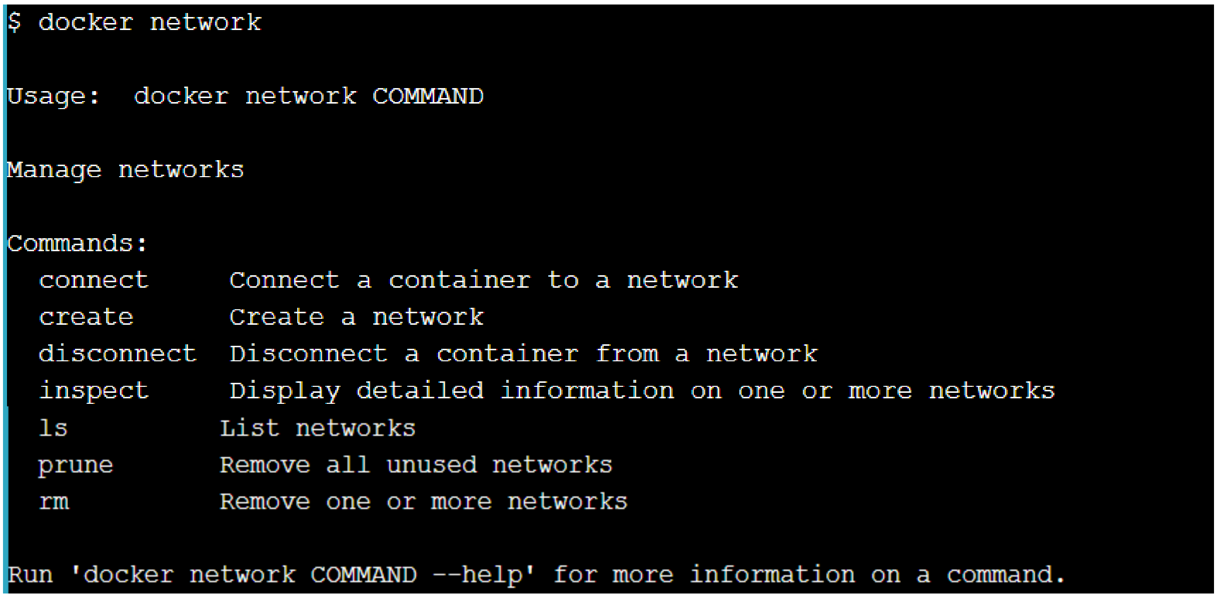
Docker Networking Hands-on Lab

Kinga Gumowska

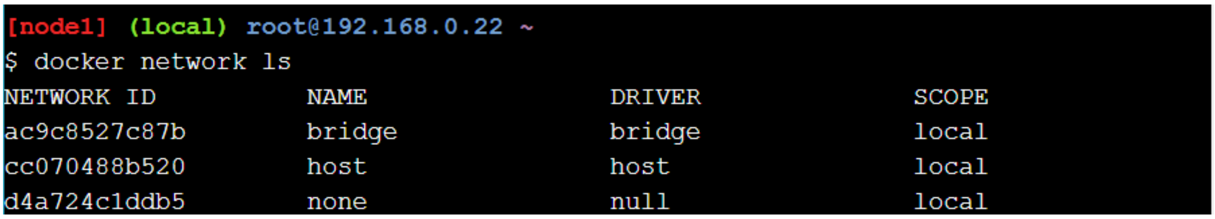
Section #1 - Networking Basics

Step 1: The Docker Network Command



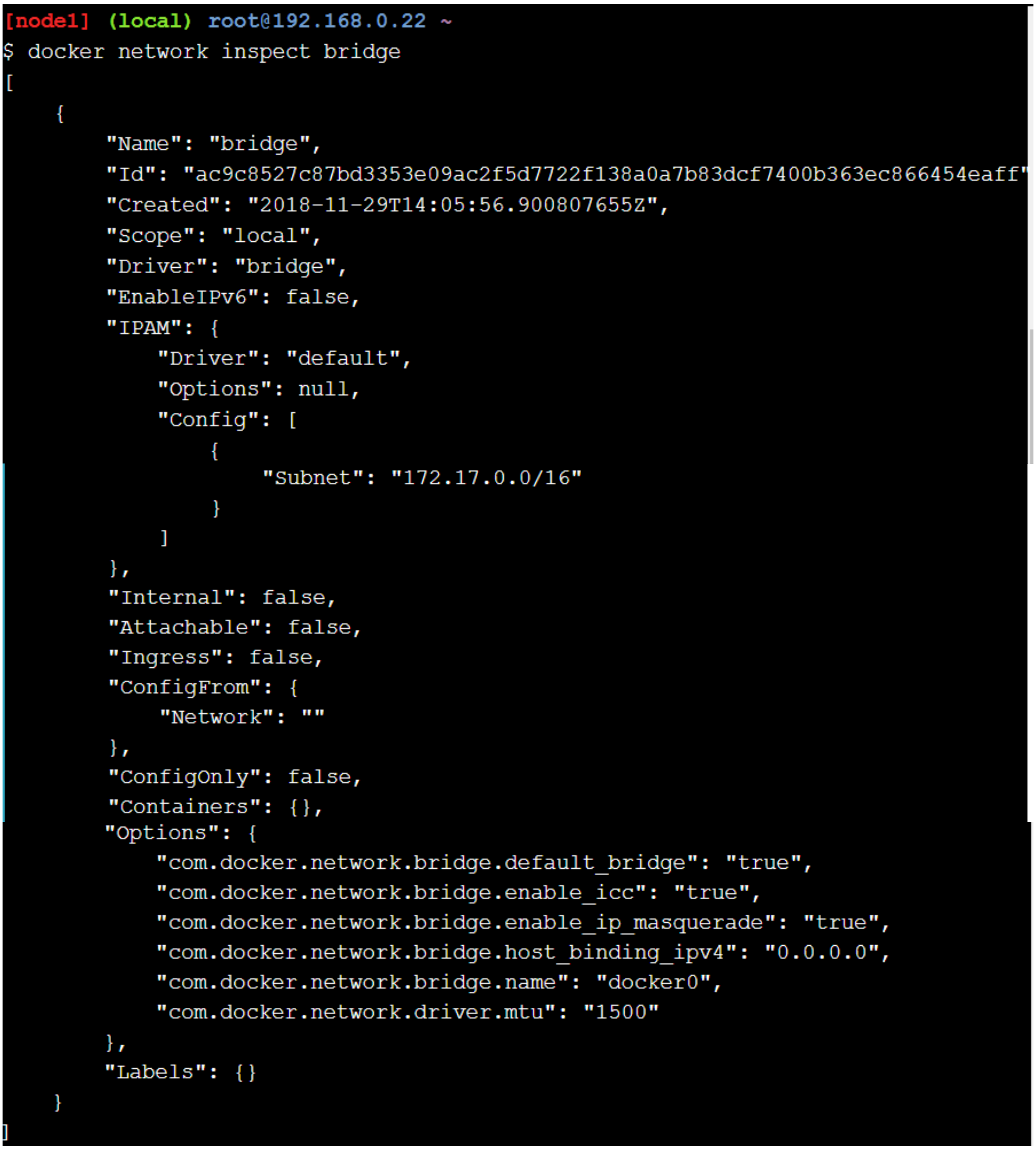
```
$ docker network

Usage:  docker network COMMAND

Manage networks

Commands:
  connect     Connect a container to a network
  create      Create a network
  disconnect  Disconnect a container from a network
  inspect     Display detailed information on one or more networks
  ls          List networks
  prune       Remove all unused networks
  rm          Remove one or more networks

Run 'docker network COMMAND --help' for more information on a command.
```

Step 2: List networks



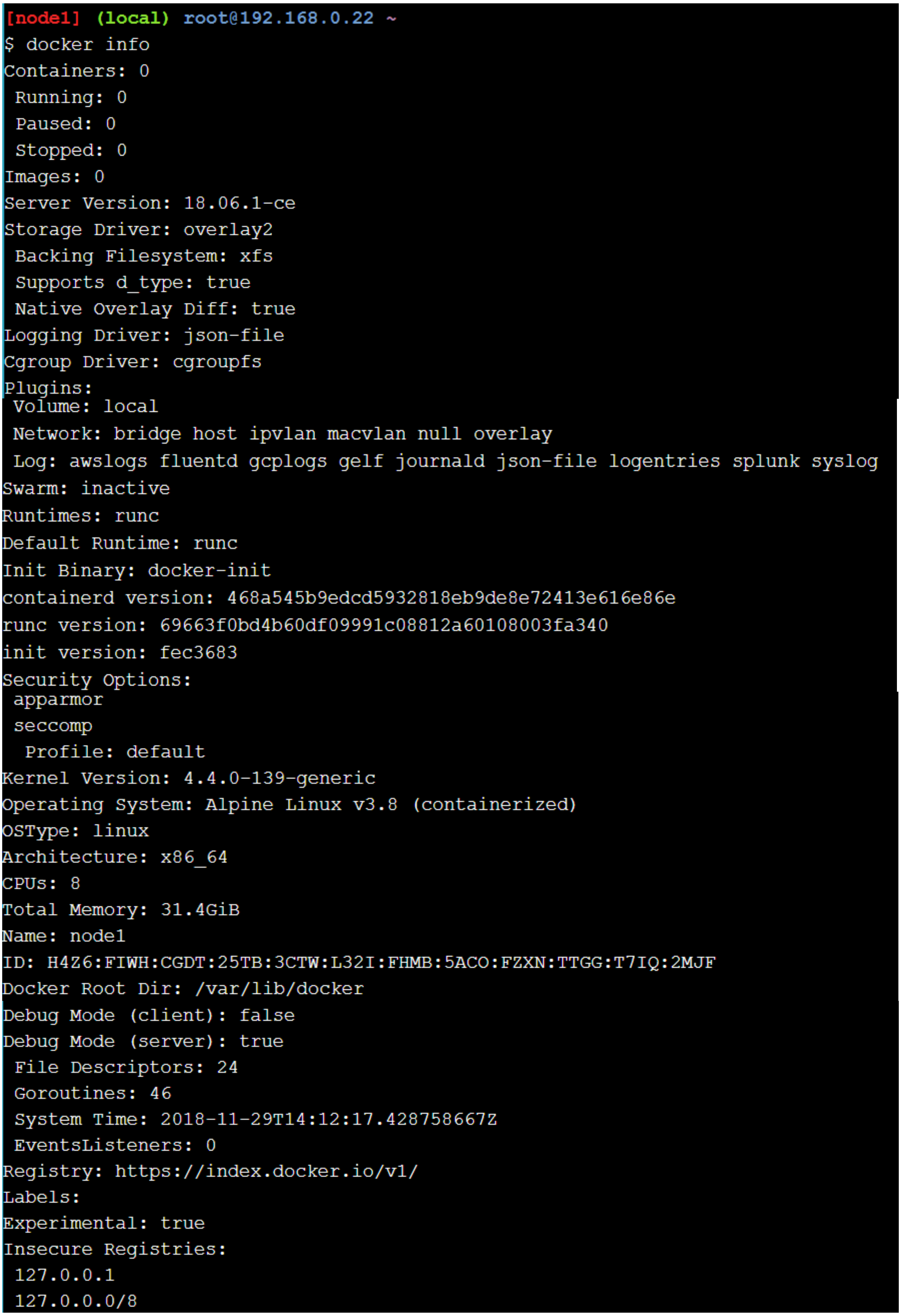
```
[node1] (local) root@192.168.0.22 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ac9c8527c87b        bridge              bridge              local
cc070488b520        host                host                local
d4a724c1ddb5        none                null                local
```

Step 3: Inspect a network



```
[node1] (local) root@192.168.0.22 ~
$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "ac9c8527c87bd3353e09ac2f5d7722f138a0a7b83dcf7400b363ec866454eaff"
        "Created": "2018-11-29T14:05:56.900807655Z",
        "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": {}
    }
]
```

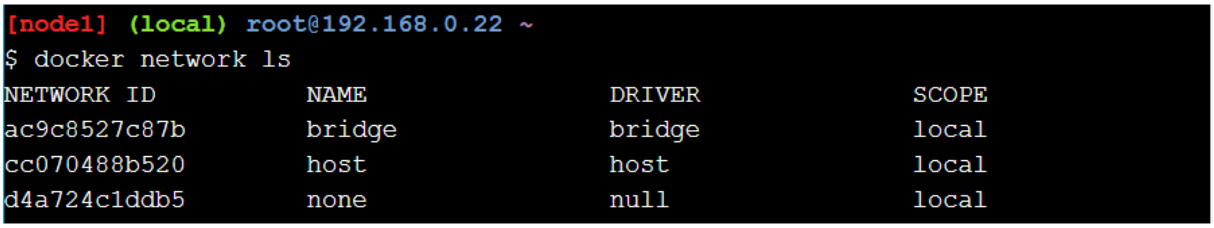
Step 4: List network driver plugins



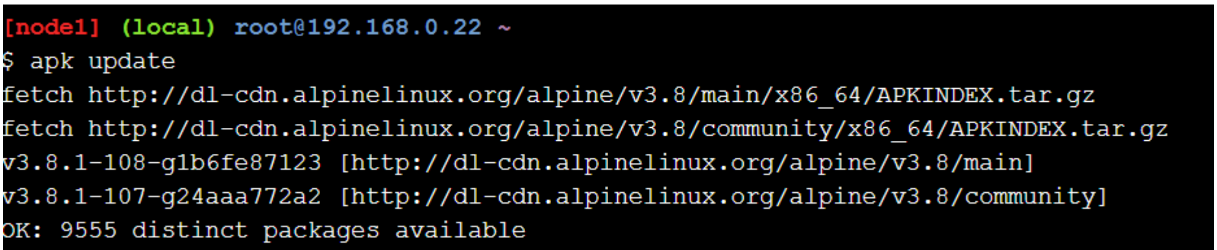
```
[node1] (local) root@192.168.0.22 ~
$ docker info
Containers: 0
 Running: 0
 Paused: 0
 Stopped: 0
Images: 0
Server Version: 18.06.1-ce
Storage Driver: overlay2
 Backing Filesystem: xfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: 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: 4.4.0-139-generic
Operating System: Alpine Linux v3.8 (containerized)
OSType: linux
Architecture: x86_64
CPUs: 8
Total Memory: 31.4GiB
Name: node1
ID: H4Z6:FIWH:CGDT:25TB:3CTW:L32I:FHMB:5ACO:FZXN:TTGG:T7IQ:2MJF
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): true
 File Descriptors: 24
 Goroutines: 46
 System Time: 2018-11-29T14:12:17.428758667Z
 EventsListeners: 0
Registry: https://index.docker.io/v1/
Labels:
Experimental: true
Insecure Registries:
 127.0.0.1
 127.0.0.0/8
```

Section #2 - Bridge Networking

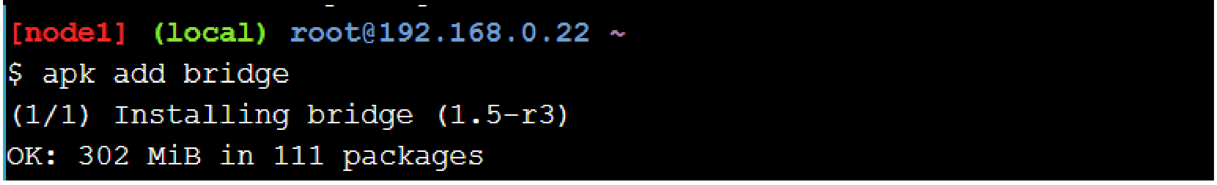
Step 1: The Basics



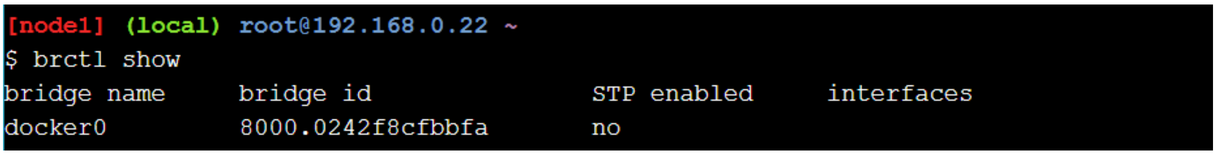
```
[node1] (local) root@192.168.0.22 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ac9c8527c87b        bridge              bridge              local
cc070488b520        host                host                local
d4a724c1ddb5        none                null                local
```



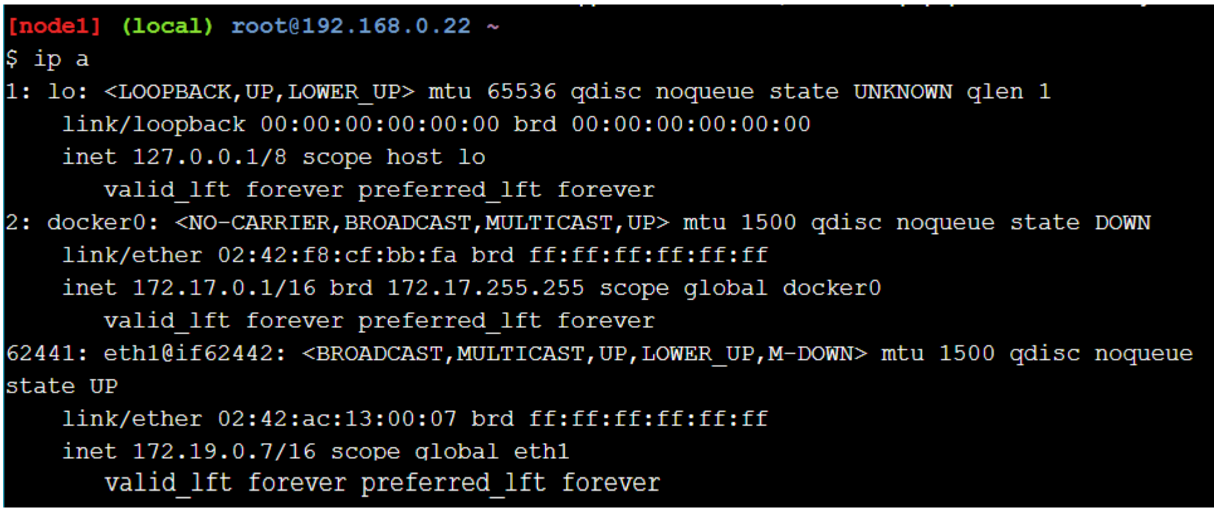
```
[node1] (local) root@192.168.0.22 ~
$ apk update
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.8/community/x86_64/APKINDEX.tar.gz
v3.8.1-108-g1b6fe87123 [http://dl-cdn.alpinelinux.org/alpine/v3.8/main]
v3.8.1-107-g24aaa772a2 [http://dl-cdn.alpinelinux.org/alpine/v3.8/community]
OK: 9555 distinct packages available
```



```
[node1] (local) root@192.168.0.22 ~
$ apk add bridge
(1/1) Installing bridge (1.5-r3)
OK: 302 MiB in 111 packages
```

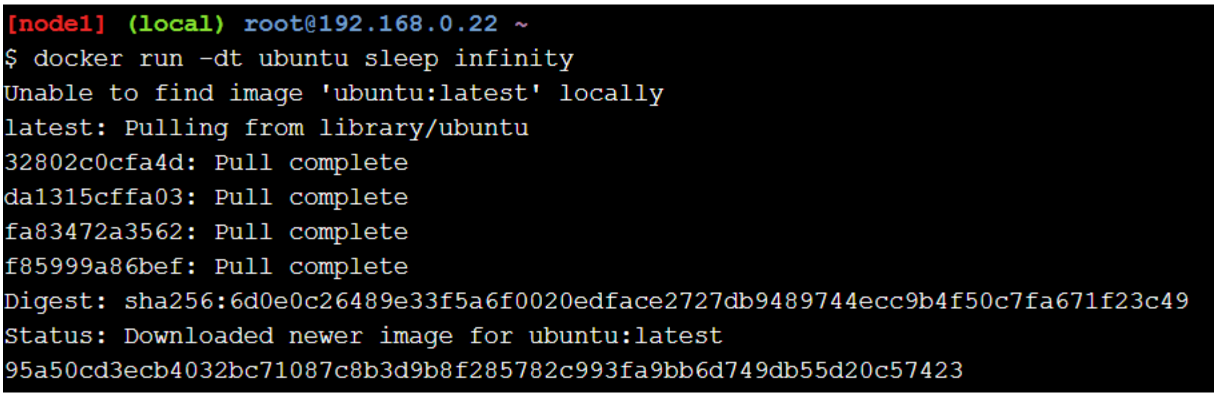


```
[node1] (local) root@192.168.0.22 ~
$ brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.0242f8cfbbfa       no
```

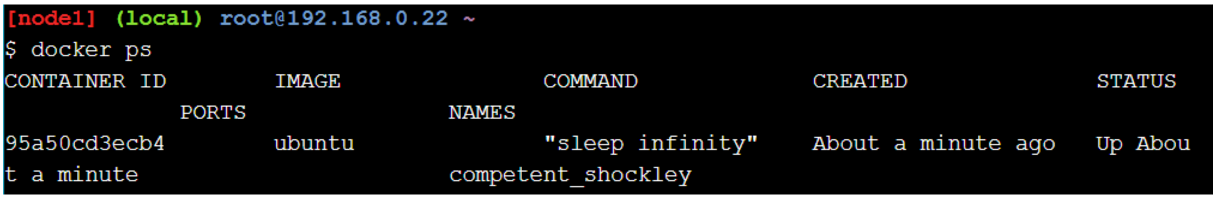


```
[node1] (local) root@192.168.0.22 ~
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN
    link/ether 02:42:f8:cf:bb:fa 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
62441: eth1@if62442: <BROADCAST,MULTICAST,UP,LOWER_UP,M-DOWN> mtu 1500 qdisc noqueue
state UP
    link/ether 02:42:ac:13:00:07 brd ff:ff:ff:ff:ff:ff
    inet 172.19.0.7/16 scope global eth1
       valid_lft forever preferred_lft forever
```

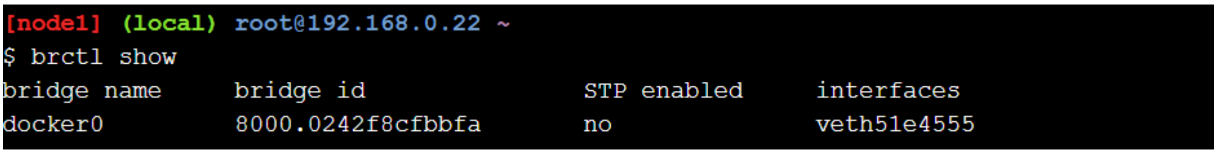
Step 2: Connect a container



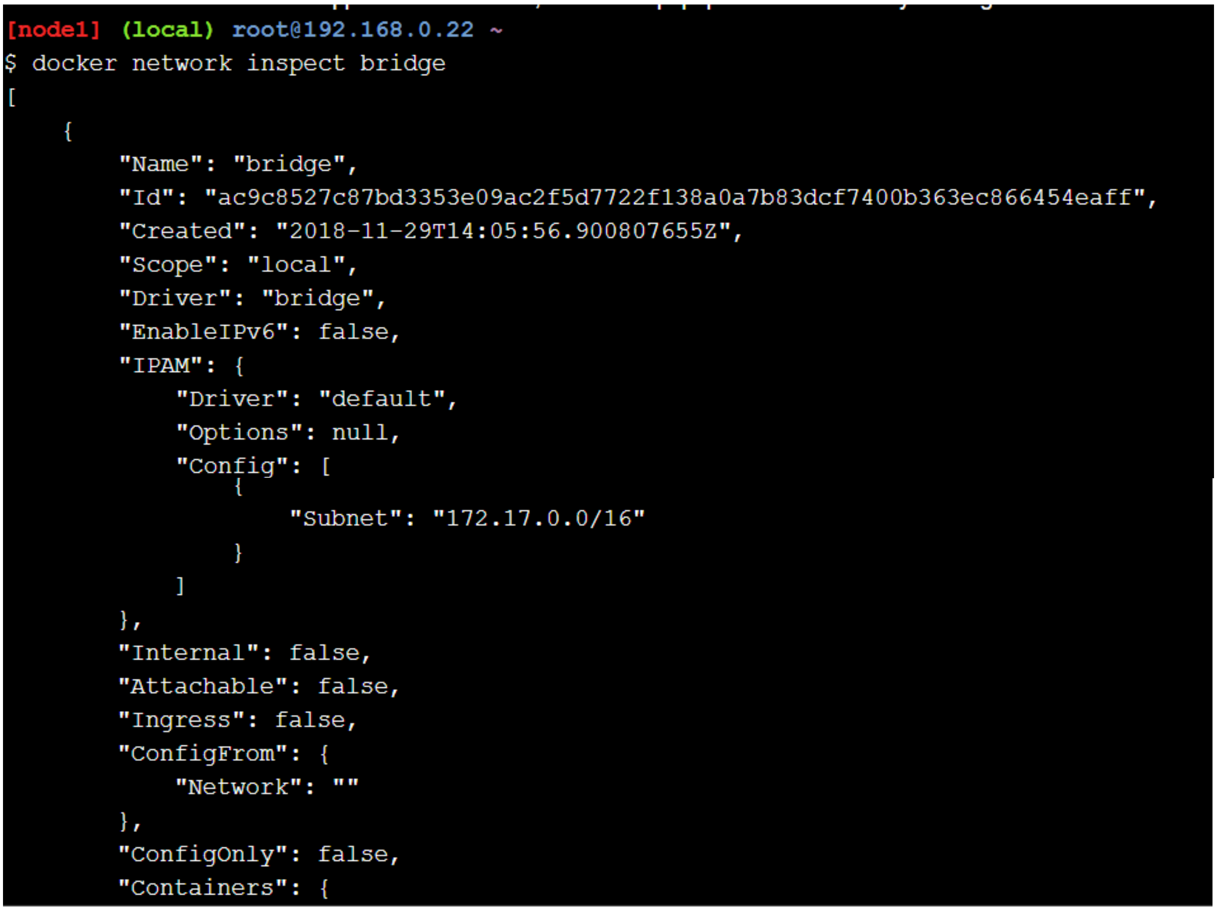
```
[node1] (local) root@192.168.0.22 ~
$ docker run -dt ubuntu sleep infinity
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
32802c0cfa4d: Pull complete
da1315cffa03: Pull complete
fa83472a3562: Pull complete
f85999a86bef: Pull complete
Digest: sha256:6d0e0c26489e33f5a6f0020edface2727db9489744ecc9b4f50c7fa671f23c49
Status: Downloaded newer image for ubuntu:latest
95a50cd3ecb4032bc71087c8b3d9b8f285782c993fa9bb6d749db55d20c57423
```



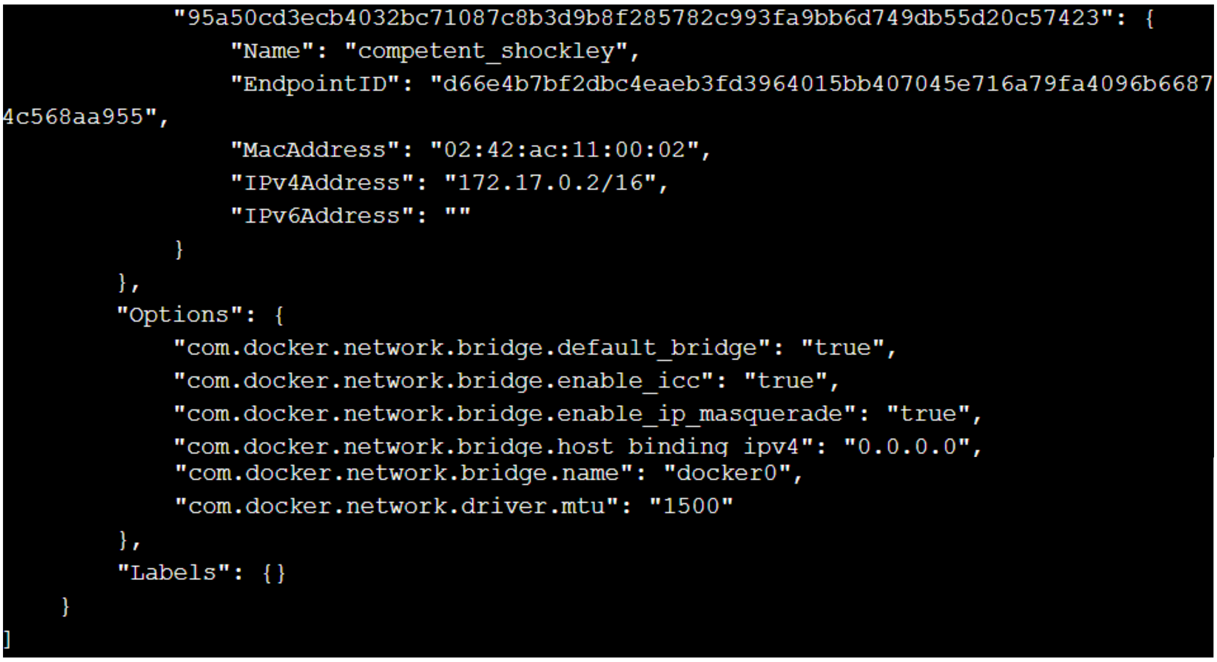
```
[node1] (local) root@192.168.0.22 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED              STATUS
             PORTS               NAMES
95a50cd3ecb4        ubuntu              "sleep infinity"    About a minute ago   Up Abou
t a minute                       competent_shockley
```



```
[node1] (local) root@192.168.0.22 ~
$ brctl show
bridge name     bridge id               STP enabled     interfaces
docker0         8000.0242f8cfbbfa       no              veth51e4555
```

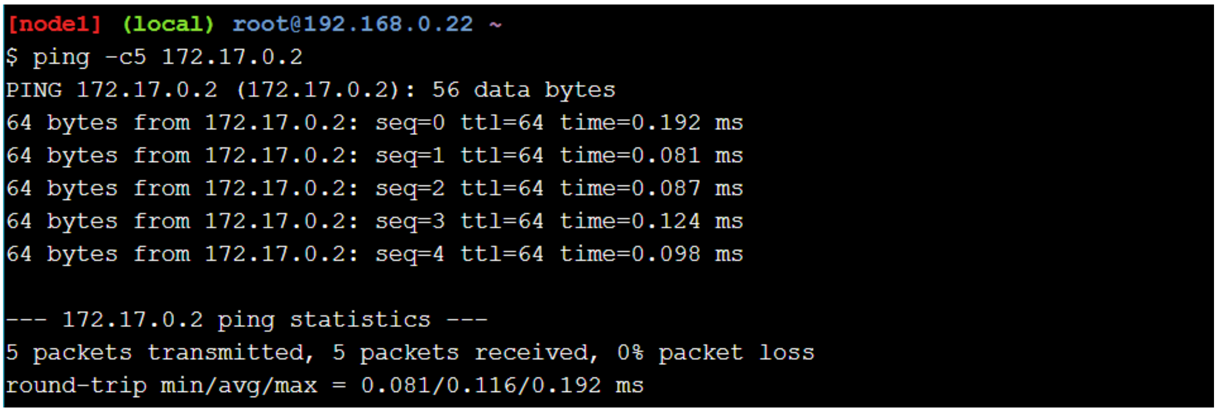


```
[node1] (local) root@192.168.0.22 ~
$ docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "ac9c8527c87bd3353e09ac2f5d7722f138a0a7b83dcf7400b363ec866454eaff",
        "Created": "2018-11-29T14:05:56.900807655Z",
        "Scope": "local",
        "Driver": "bridge",
        "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": {
```

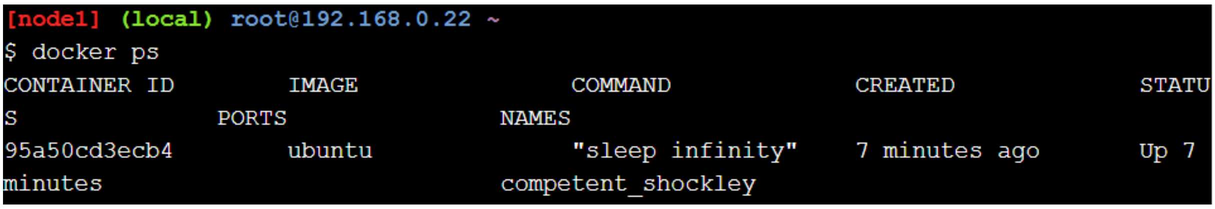


```
            "95a50cd3ecb4032bc71087c8b3d9b8f285782c993fa9bb6d749db55d20c57423": {
                "Name": "competent_shockley",
                "EndpointID": "d66e4b7bf2dbc4eaeb3fd3964015bb407045e716a79fa4096b6687
4c568aa955",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "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": {}
    }
]
```

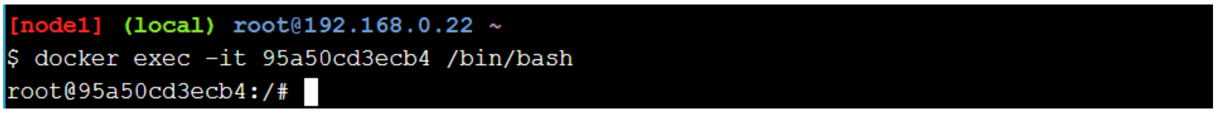
Step 3: Test network connectivity



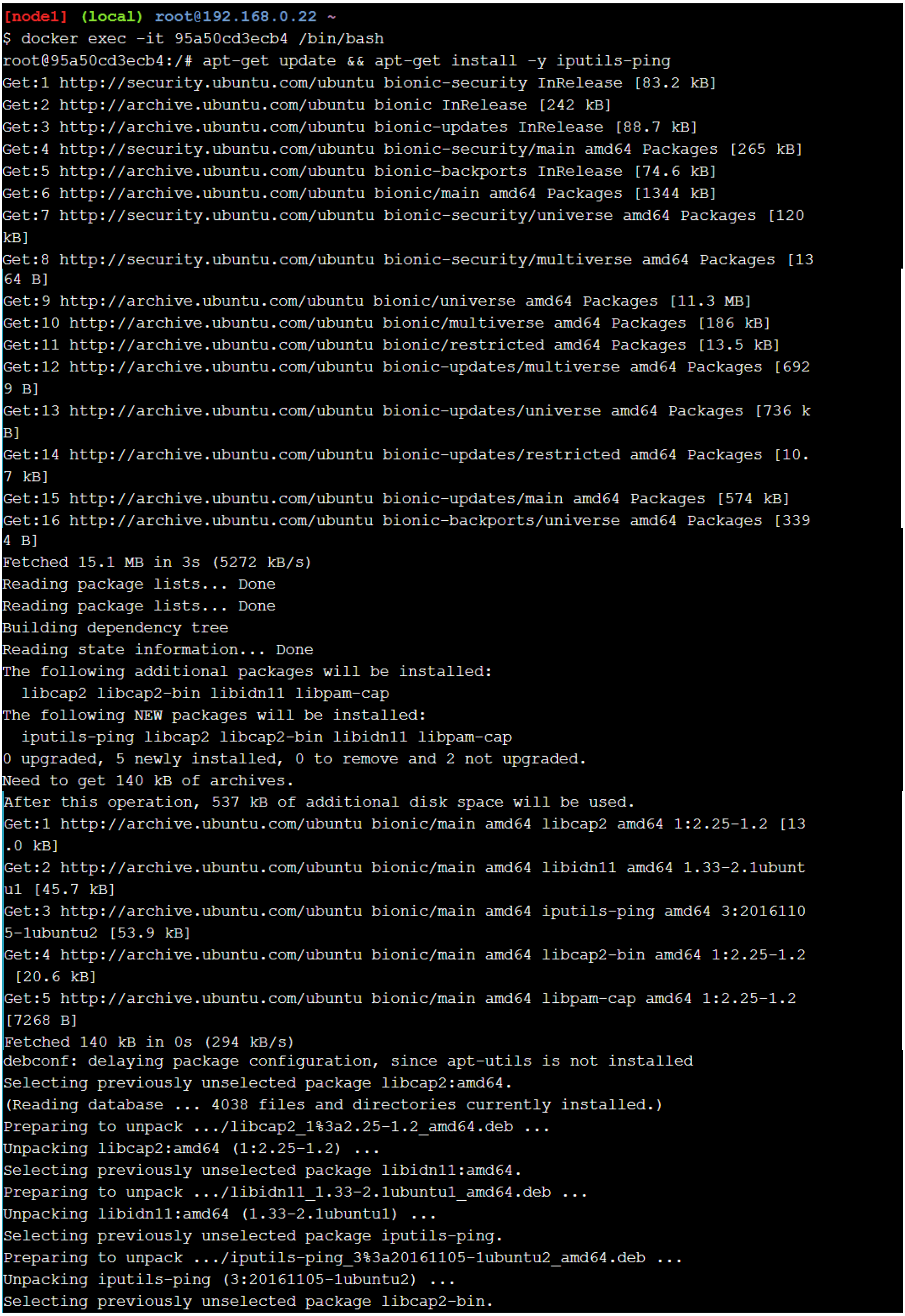
```
[node1] (local) root@192.168.0.22 ~
$ ping -c5 172.17.0.2
PING 172.17.0.2 (172.17.0.2): 56 data bytes
64 bytes from 172.17.0.2: seq=0 ttl=64 time=0.192 ms
64 bytes from 172.17.0.2: seq=1 ttl=64 time=0.081 ms
64 bytes from 172.17.0.2: seq=2 ttl=64 time=0.087 ms
64 bytes from 172.17.0.2: seq=3 ttl=64 time=0.124 ms
64 bytes from 172.17.0.2: seq=4 ttl=64 time=0.098 ms

--- 172.17.0.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.081/0.116/0.192 ms
```



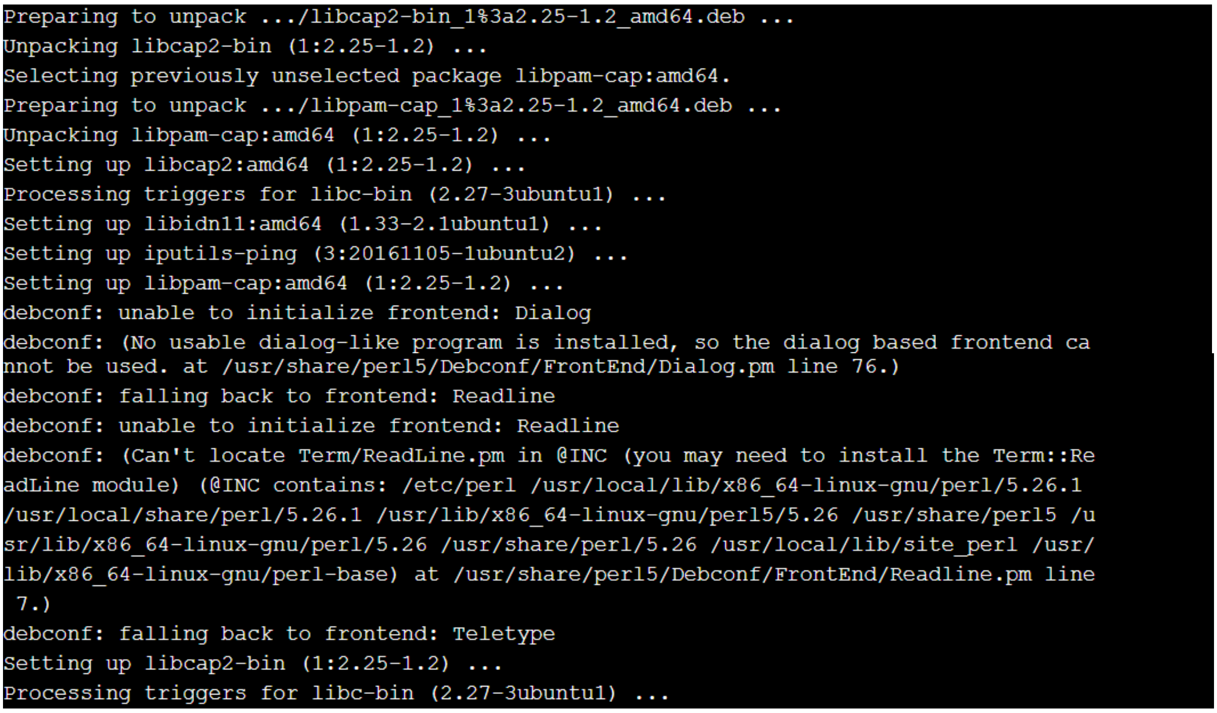
```
[node1] (local) root@192.168.0.22 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATU
S               PORTS               NAMES
95a50cd3ecb4        ubuntu              "sleep infinity"    7 minutes ago       Up 7
minutes                             competent_shockley
```



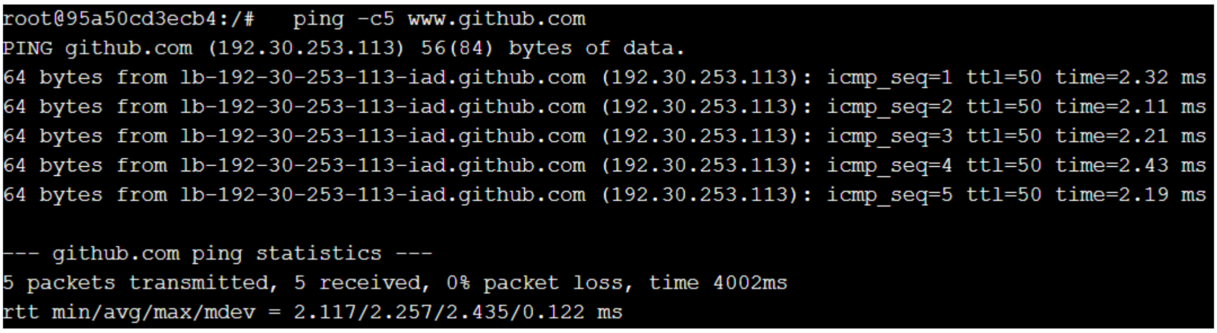
```
[node1] (local) root@192.168.0.22 ~
$ docker exec -it 95a50cd3ecb4 /bin/bash
root@95a50cd3ecb4:/#
```



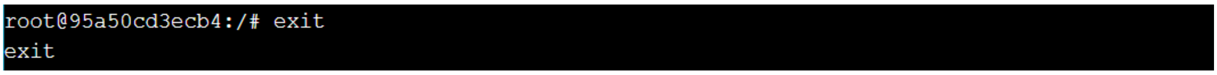
```
[node1] (local) root@192.168.0.22 ~
$ docker exec -it 95a50cd3ecb4 /bin/bash
root@95a50cd3ecb4:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [265 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [120
kB]
Get:8 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [13
64 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [692
9 B]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [736 k
B]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.
7 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [574 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [339
4 B]
Fetched 15.1 MB in 3s (5272 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 2 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 1.33-2.1ubunt
u1 [45.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 iputils-ping amd64 3:2016110
5-1ubuntu2 [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 (294 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.
```



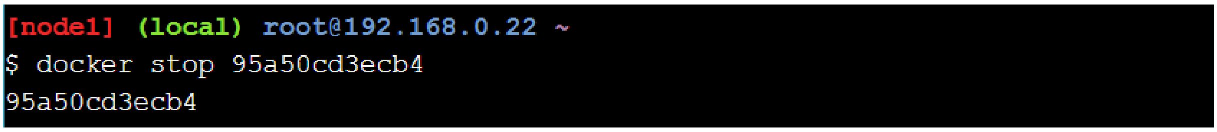
```
Preparing to unpack .../libcap2-bin_1%3a2.25-1.2_amd64.deb ...
Unpacking libcap2-bin (1:2.25-1.2) ...
Selecting previously unselected package libpam-cap:amd64.
Preparing to unpack .../libpam-cap_1%3a2.25-1.2_amd64.deb ...
Unpacking libpam-cap:amd64 (1:2.25-1.2) ...
Setting up libcap2:amd64 (1:2.25-1.2) ...
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 ca
nnot 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::Re
adLine 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-linux-gnu/perl5/5.26 /usr/share/perl5 /u
sr/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@95a50cd3ecb4:/#   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=2.32 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=2 ttl=50 time=2.11 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=3 ttl=50 time=2.21 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=4 ttl=50 time=2.43 ms
64 bytes from lb-192-30-253-113-iad.github.com (192.30.253.113): icmp_seq=5 ttl=50 time=2.19 ms

--- github.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4002ms
rtt min/avg/max/mdev = 2.117/2.257/2.435/0.122 ms
```

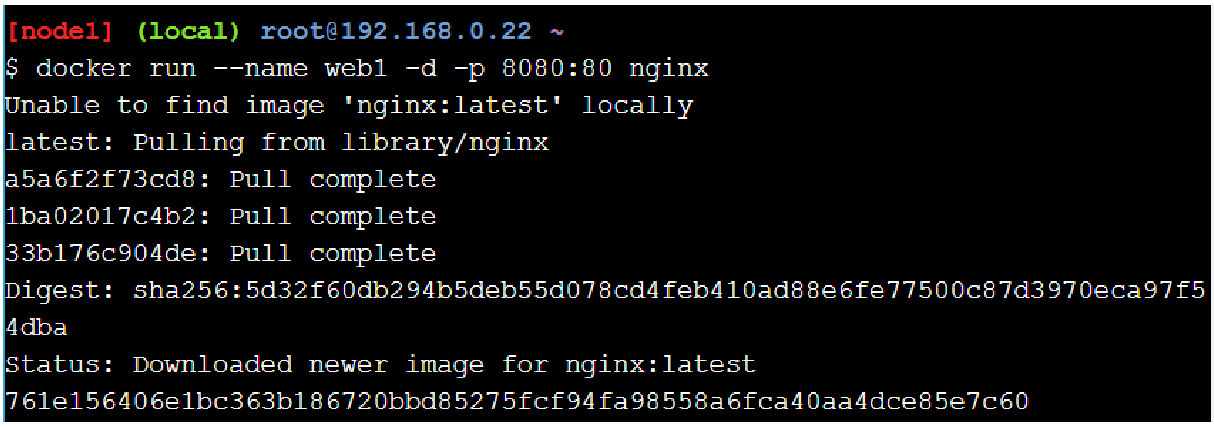


```
root@95a50cd3ecb4:/# exit
exit
```

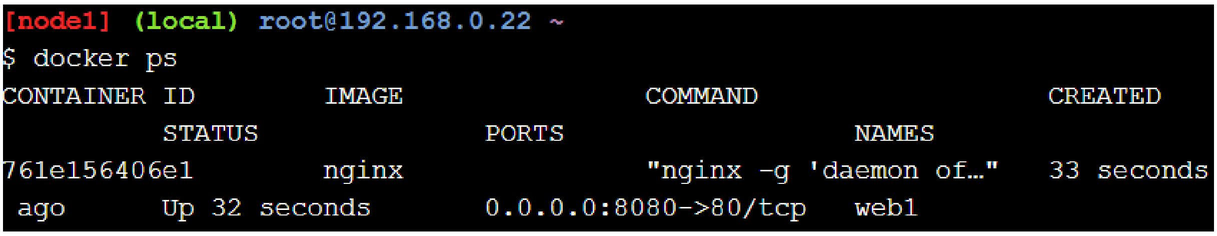


```
[node1] (local) root@192.168.0.22 ~
$ docker stop 95a50cd3ecb4
95a50cd3ecb4
```

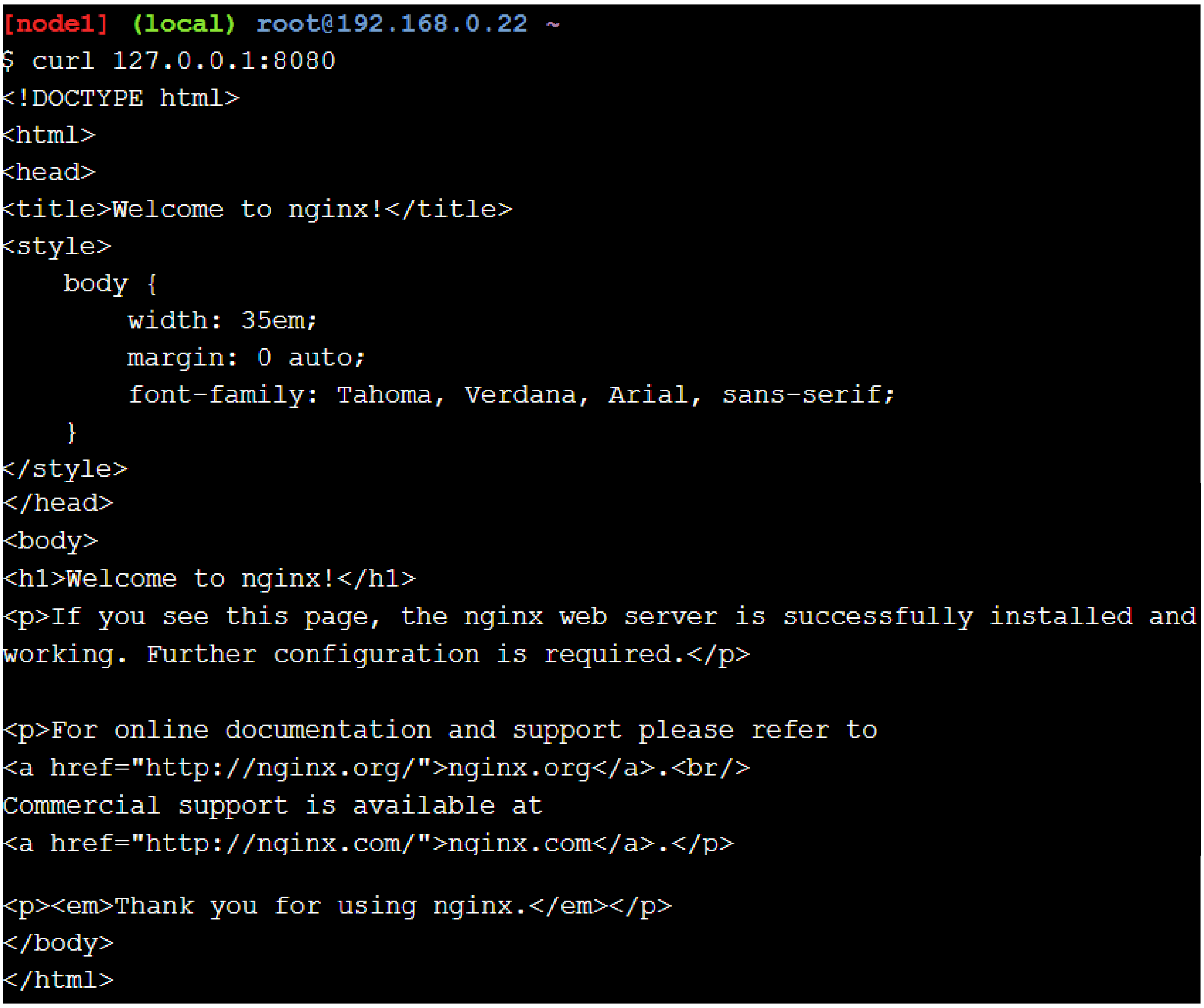
Step 4: Configure NAT for external connectivity



```
[node1] (local) root@192.168.0.22 ~
$ 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: sha256:5d32f60db294b5deb55d078cd4feb410ad88e6fe77500c87d3970eca97f5
4dba
Status: Downloaded newer image for nginx:latest
761e156406e1bc363b186720bbd85275fcf94fa98558a6fca40aa4dce85e7c60
```



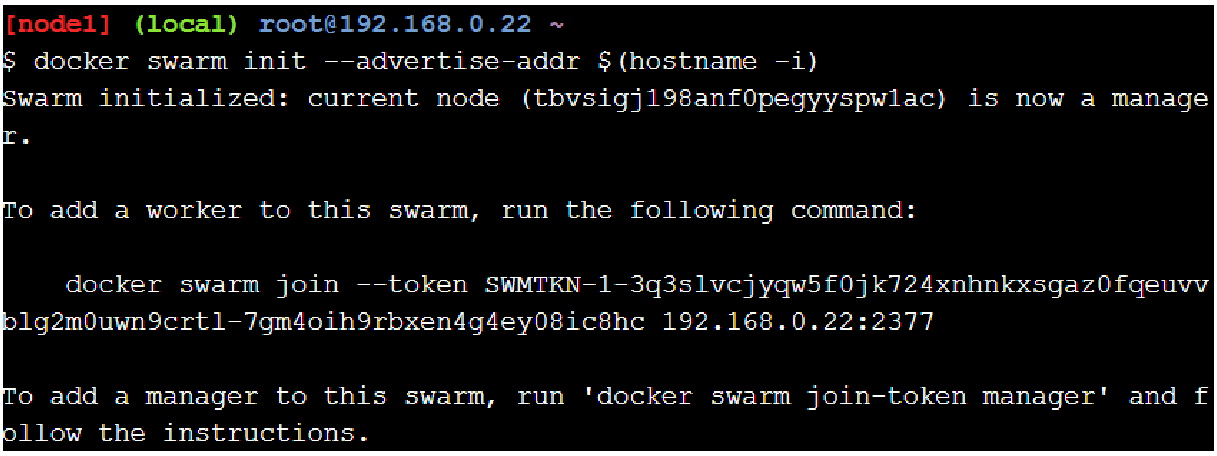
```
[node1] (local) root@192.168.0.22 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
          STATUS              PORTS                  NAMES
761e156406e1        nginx               "nginx -g 'daemon of…"   33 seconds
 ago       Up 32 seconds       0.0.0.0:8080->80/tcp   web1
```



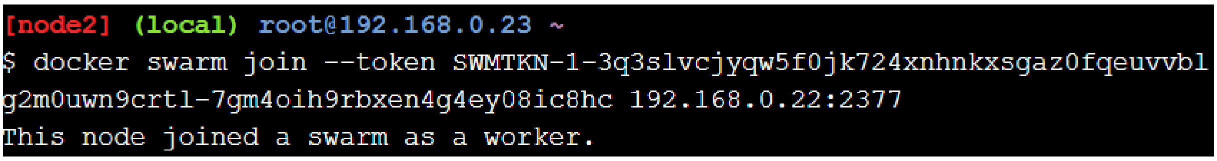
```
[node1] (local) root@192.168.0.22 ~
$ curl 127.0.0.1:8080
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<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>
```

Section #3 - Overlay Networking

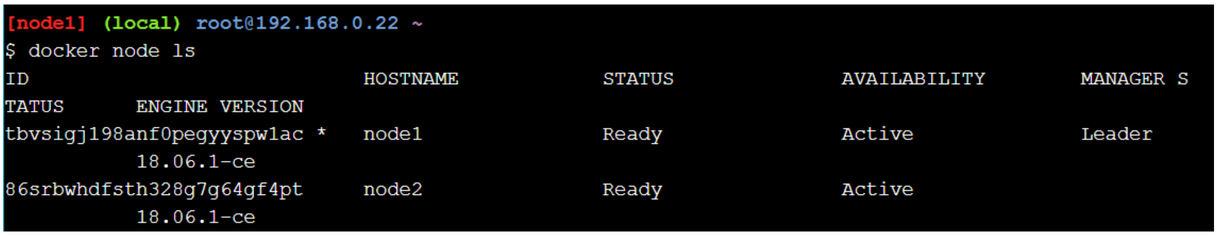
Step 1: The Basics



```
[node1] (local) root@192.168.0.22 ~
$ docker swarm init --advertise-addr $(hostname -i)
Swarm initialized: current node (tbvsigj198anf0pegyyspw1ac) is now a manage
r.

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

    docker swarm join --token SWMTKN-1-3q3slvcjyqw5f0jk724xnhnkxsgaz0fqeuvv
blg2m0uwn9crtl-7gm4oih9rbxen4g4ey08ic8hc 192.168.0.22:2377

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

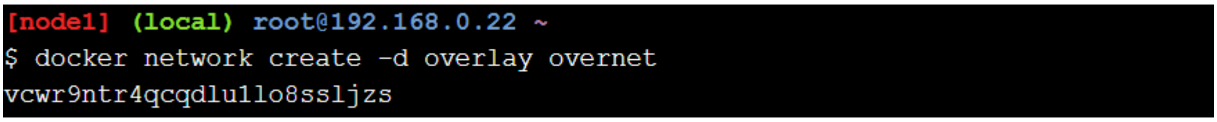


```
[node2] (local) root@192.168.0.23 ~
$ docker swarm join --token SWMTKN-1-3q3slvcjyqw5f0jk724xnhnkxsgaz0fqeuvvbl
g2m0uwn9crtl-7gm4oih9rbxen4g4ey08ic8hc 192.168.0.22:2377
This node joined a swarm as a worker.
```

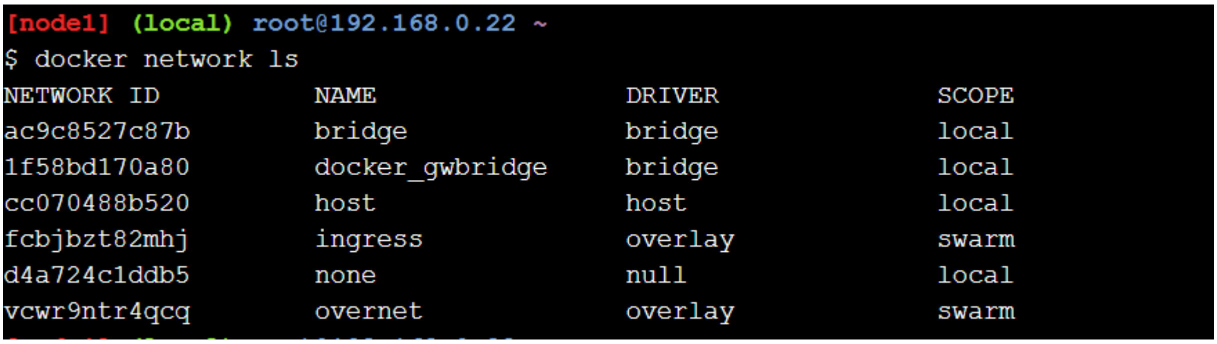


```
[node1] (local) root@192.168.0.22 ~
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILITY        MANAGER S
TATUS      ENGINE VERSION
tbvsigj198anf0pegyyspw1ac *   node1               Ready               Active              Leader
           18.06.1-ce
86srbwhdfsth328g7g64gf4pt     node2               Ready               Active
           18.06.1-ce
```

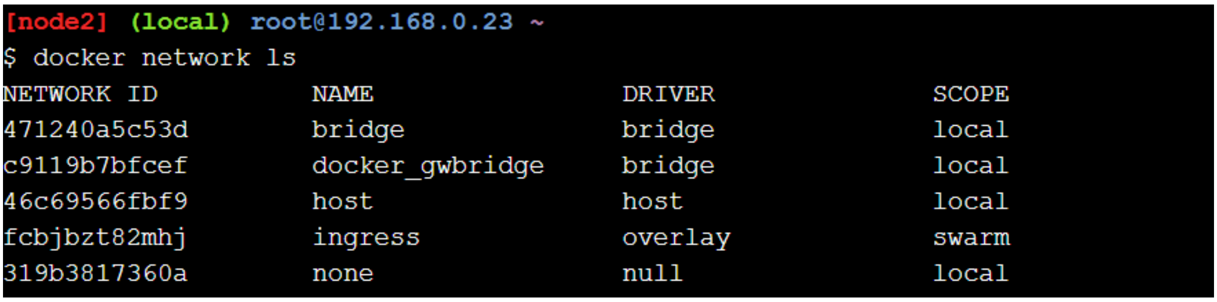
Step 2: Create an overlay network



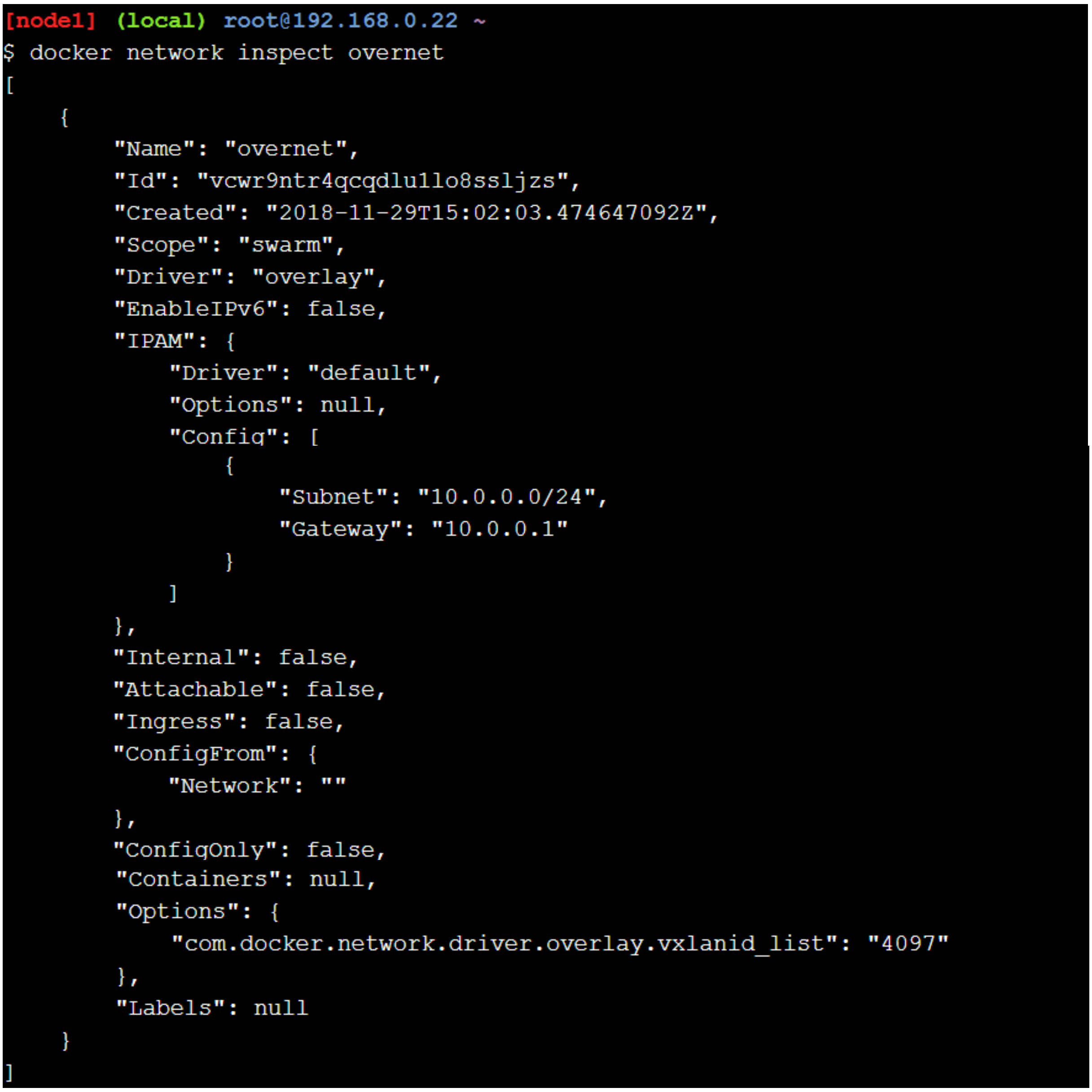
```
[node1] (local) root@192.168.0.22 ~
$ docker network create -d overlay overnet
vcwr9ntr4qcqdlu1lo8ssljzs
```



```
[node1] (local) root@192.168.0.22 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
ac9c8527c87b        bridge              bridge              local
1f58bd170a80        docker_gwbridge     bridge              local
cc070488b520        host                host                local
fcbjbzt82mhj        ingress             overlay             swarm
d4a724c1ddb5        none                null                local
vcwr9ntr4qcq        overnet             overlay             swarm
```

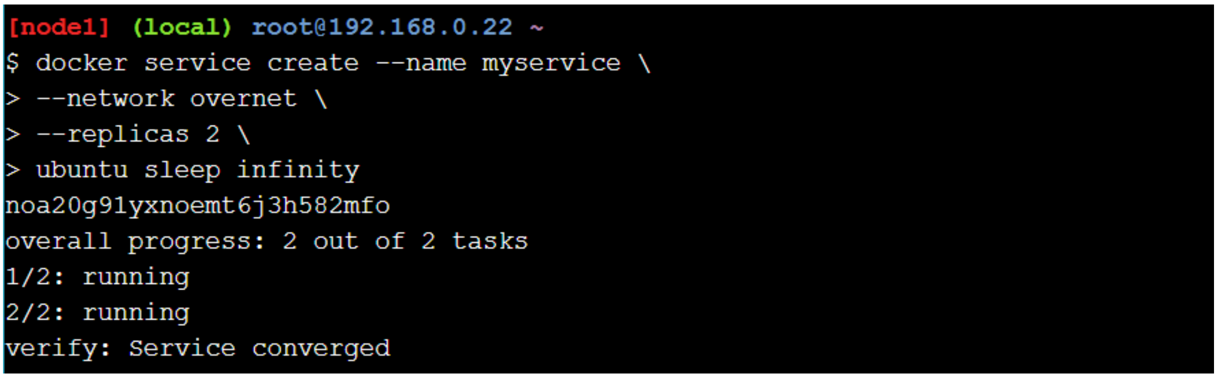


```
[node2] (local) root@192.168.0.23 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
471240a5c53d        bridge              bridge              local
c9119b7bfcef        docker_gwbridge     bridge              local
46c69566fbf9        host                host                local
fcbjbzt82mhj        ingress             overlay             swarm
319b3817360a        none                null                local
```

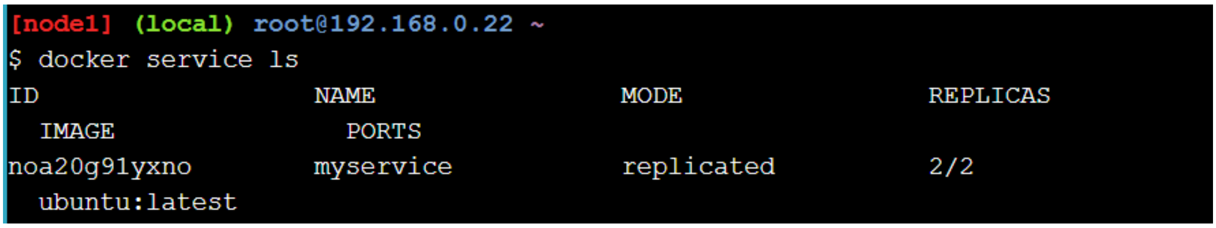


```
[node1] (local) root@192.168.0.22 ~
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "vcwr9ntr4qcqdlu1lo8ssljzs",
        "Created": "2018-11-29T15:02:03.474647092Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "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
    }
]
```

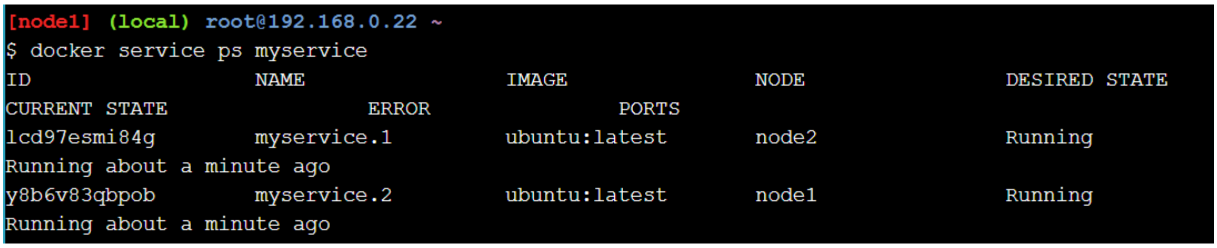
Step 3: Create a service



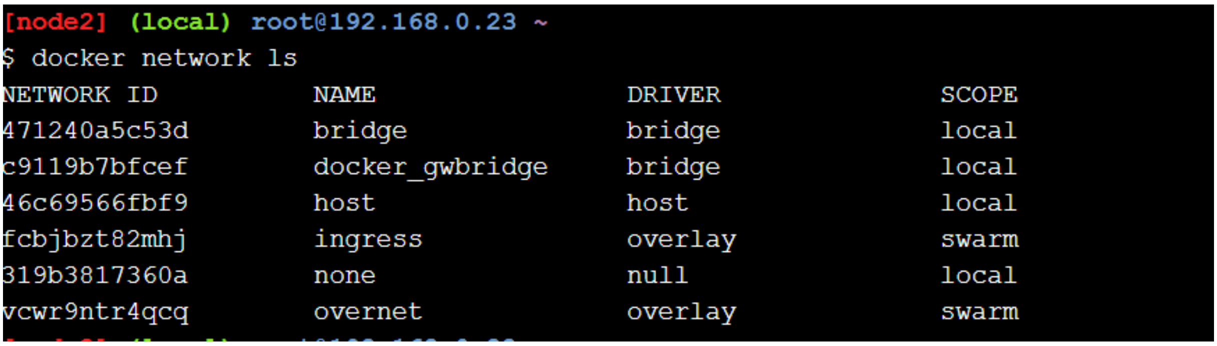
```
[node1] (local) root@192.168.0.22 ~
$ docker service create --name myservice \
> --network overnet \
> --replicas 2 \
> ubuntu sleep infinity
noa20g91yxnoemt6j3h582mfo
overall progress: 2 out of 2 tasks
1/2: running
2/2: running
verify: Service converged
```



```
[node1] (local) root@192.168.0.22 ~
$ docker service ls
ID                  NAME                MODE                REPLICAS
  IMAGE               PORTS
noa20g91yxno        myservice           replicated          2/2
  ubuntu:latest
```

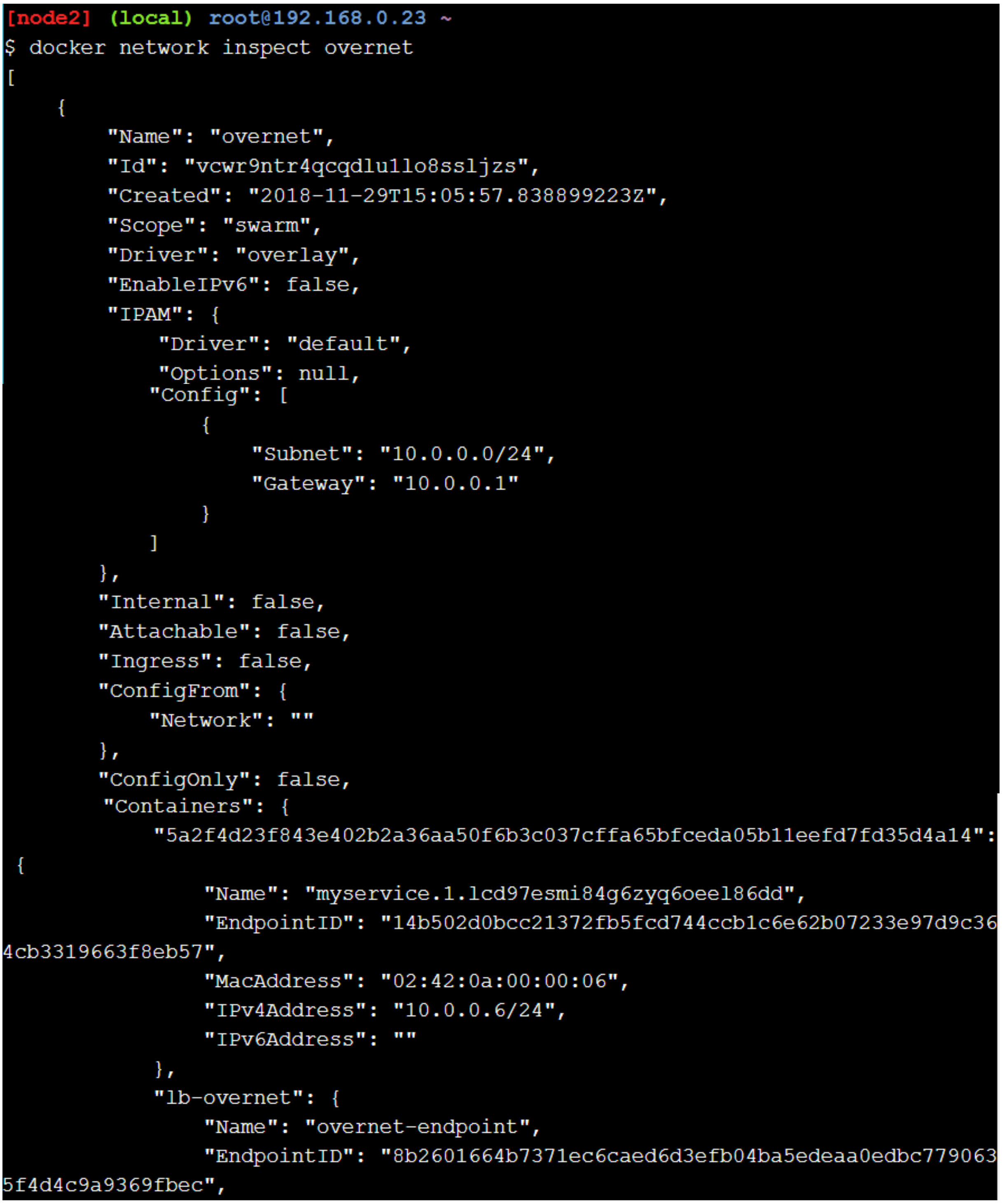


```
[node1] (local) root@192.168.0.22 ~
$ docker service ps myservice
ID                  NAME                IMAGE               NODE                DESIRED STATE
CURRENT STATE           ERROR               PORTS
1cd97esmi84g        myservice.1         ubuntu:latest       node2               Running
Running about a minute ago
y8b6v83qbpob        myservice.2         ubuntu:latest       node1               Running
Running about a minute ago
```

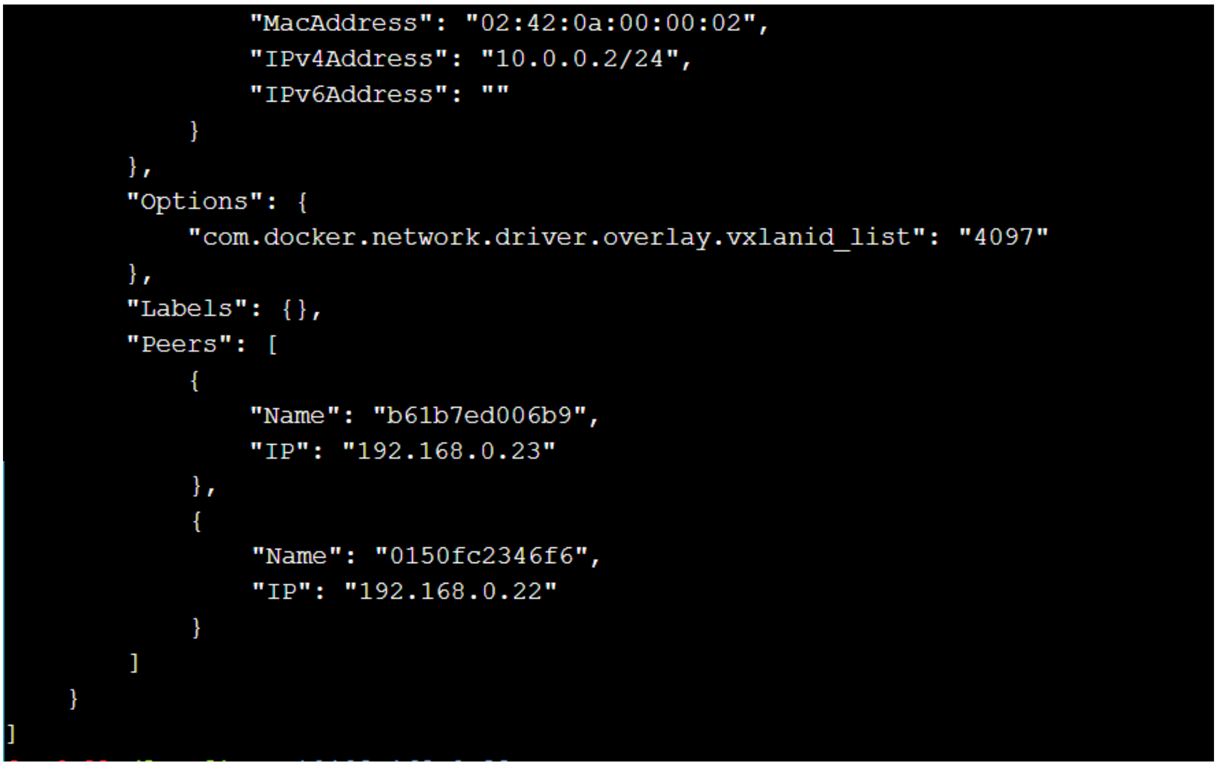


```
[node2] (local) root@192.168.0.23 ~
$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
471240a5c53d        bridge              bridge              local
c9119b7bfcef        docker_gwbridge     bridge              local
46c69566fbf9        host                host                local
fcbjbzt82mhj        ingress             overlay             swarm
319b3817360a        none                null                local
vcwr9ntr4qcq        overnet             overlay             swarm
```

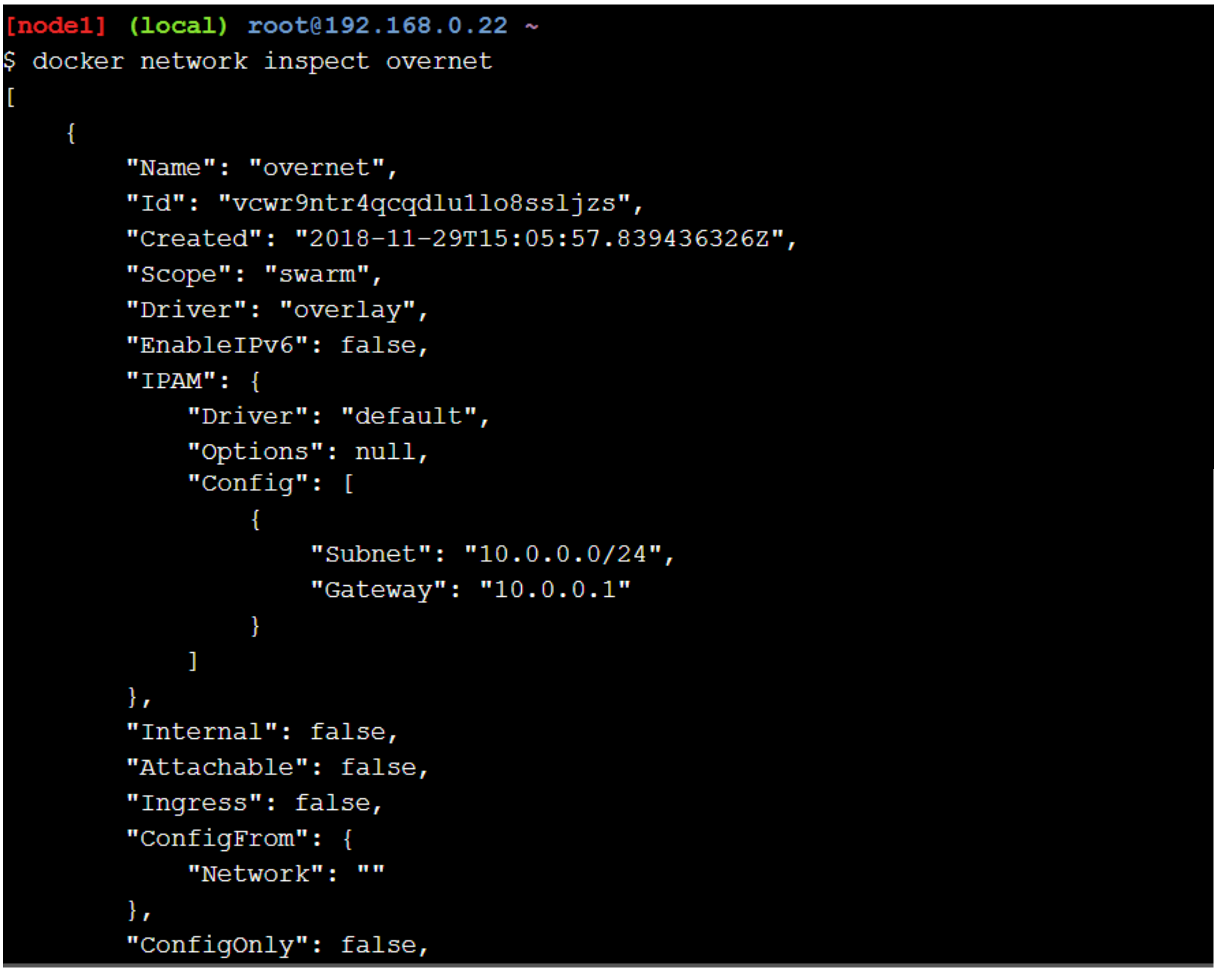
Step 4: Test the network



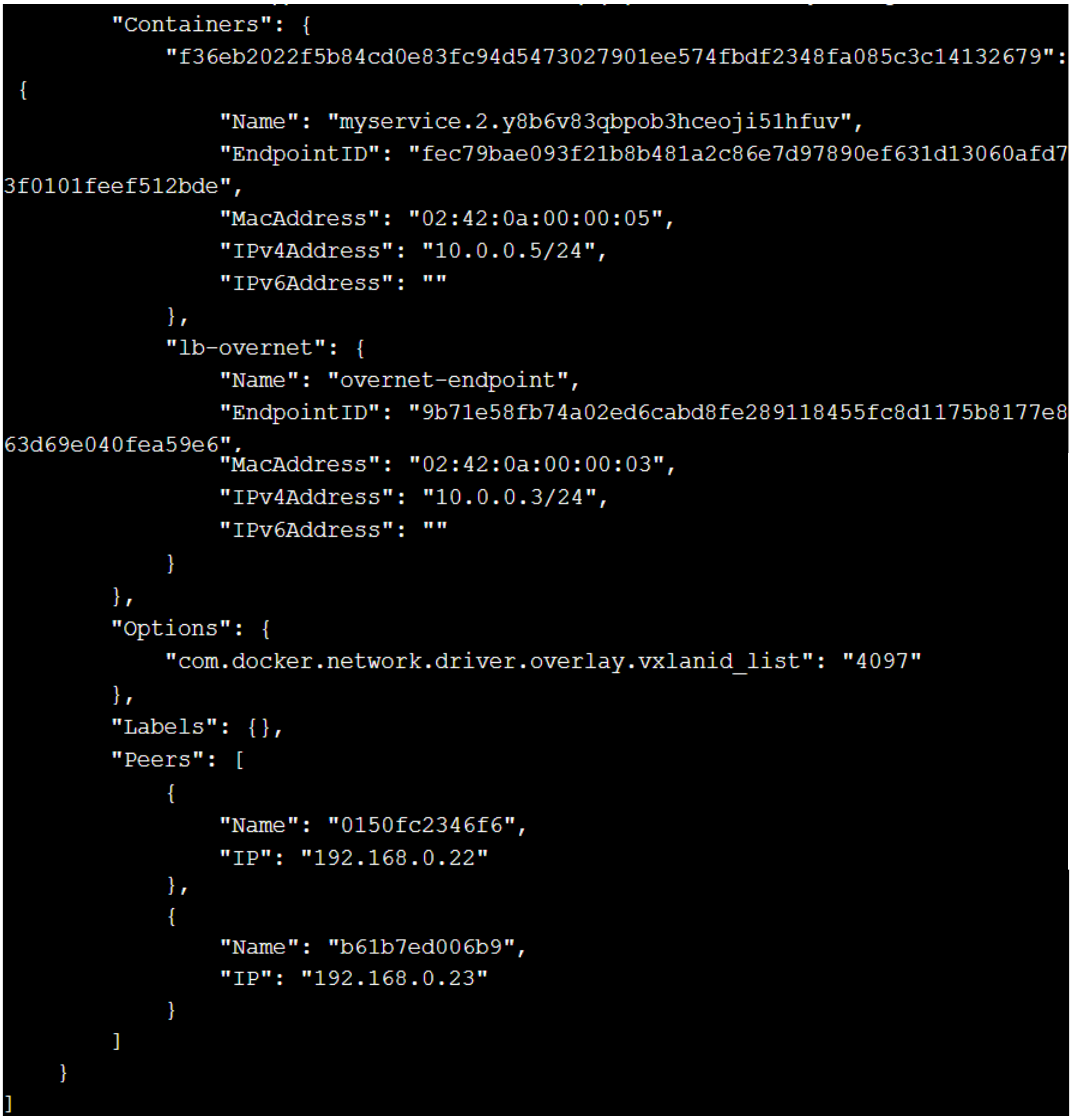
```
[node2] (local) root@192.168.0.23 ~
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "vcwr9ntr4qcqdlu1lo8ssljzs",
        "Created": "2018-11-29T15:05:57.838899223Z",
        "Scope": "swarm",
        "Driver": "overlay",
        "EnableIPv6": false,
        "IPAM": {
            "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": {
            "5a2f4d23f843e402b2a36aa50f6b3c037cffa65bfceda05b11eefd7fd35d4a14":
 {
                "Name": "myservice.1.lcd97esmi84g6zyq6oeel86dd",
                "EndpointID": "14b502d0bcc21372fb5fcd744ccb1c6e62b07233e97d9c36
4cb3319663f8eb57",
                "MacAddress": "02:42:0a:00:00:06",
                "IPv4Address": "10.0.0.6/24",
                "IPv6Address": ""
            },
            "lb-overnet": {
                "Name": "overnet-endpoint",
                "EndpointID": "8b2601664b7371ec6caed6d3efb04ba5edeaa0edbc779063
5f4d4c9a9369fbec",
```



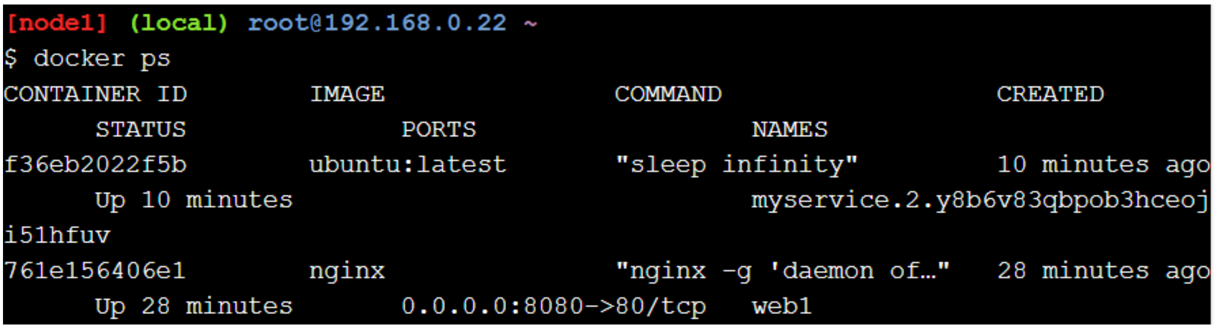
```
                "MacAddress": "02:42:0a:00:00:02",
                "IPv4Address": "10.0.0.2/24",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.driver.overlay.vxlanid_list": "4097"
        },
        "Labels": {},
        "Peers": [
            {
                "Name": "b61b7ed006b9",
                "IP": "192.168.0.23"
            },
            {
                "Name": "0150fc2346f6",
                "IP": "192.168.0.22"
            }
        ]
    }
]
```



```
[node1] (local) root@192.168.0.22 ~
$ docker network inspect overnet
[
    {
        "Name": "overnet",
        "Id": "vcwr9ntr4qcqdlu1lo8ssljzs",
        "Created": "2018-11-29T15:05:57.839436326Z",
        "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,
```

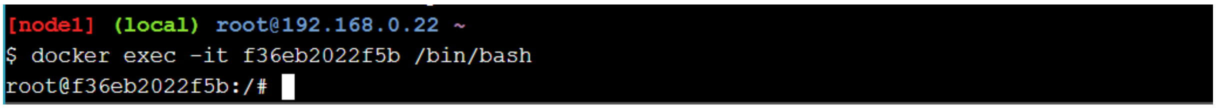


```
        "Containers": {
            "f36eb2022f5b84cd0e83fc94d5473027901ee574fbdf2348fa085c3c14132679":
 {
                "Name": "myservice.2.y8b6v83qbpob3hceoji51hfuv",
                "EndpointID": "fec79bae093f21b8b481a2c86e7d97890ef631d13060afd7
3f0101feef512bde",
                "MacAddress": "02:42:0a:00:00:05",
                "IPv4Address": "10.0.0.5/24",
                "IPv6Address": ""
            },
            "lb-overnet": {
                "Name": "overnet-endpoint",
                "EndpointID": "9b71e58fb74a02ed6cabd8fe289118455fc8d1175b8177e8
63d69e040fea59e6",
                "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": "0150fc2346f6",
                "IP": "192.168.0.22"
            },
            {
                "Name": "b61b7ed006b9",
                "IP": "192.168.0.23"
            }
        ]
    }
]
```

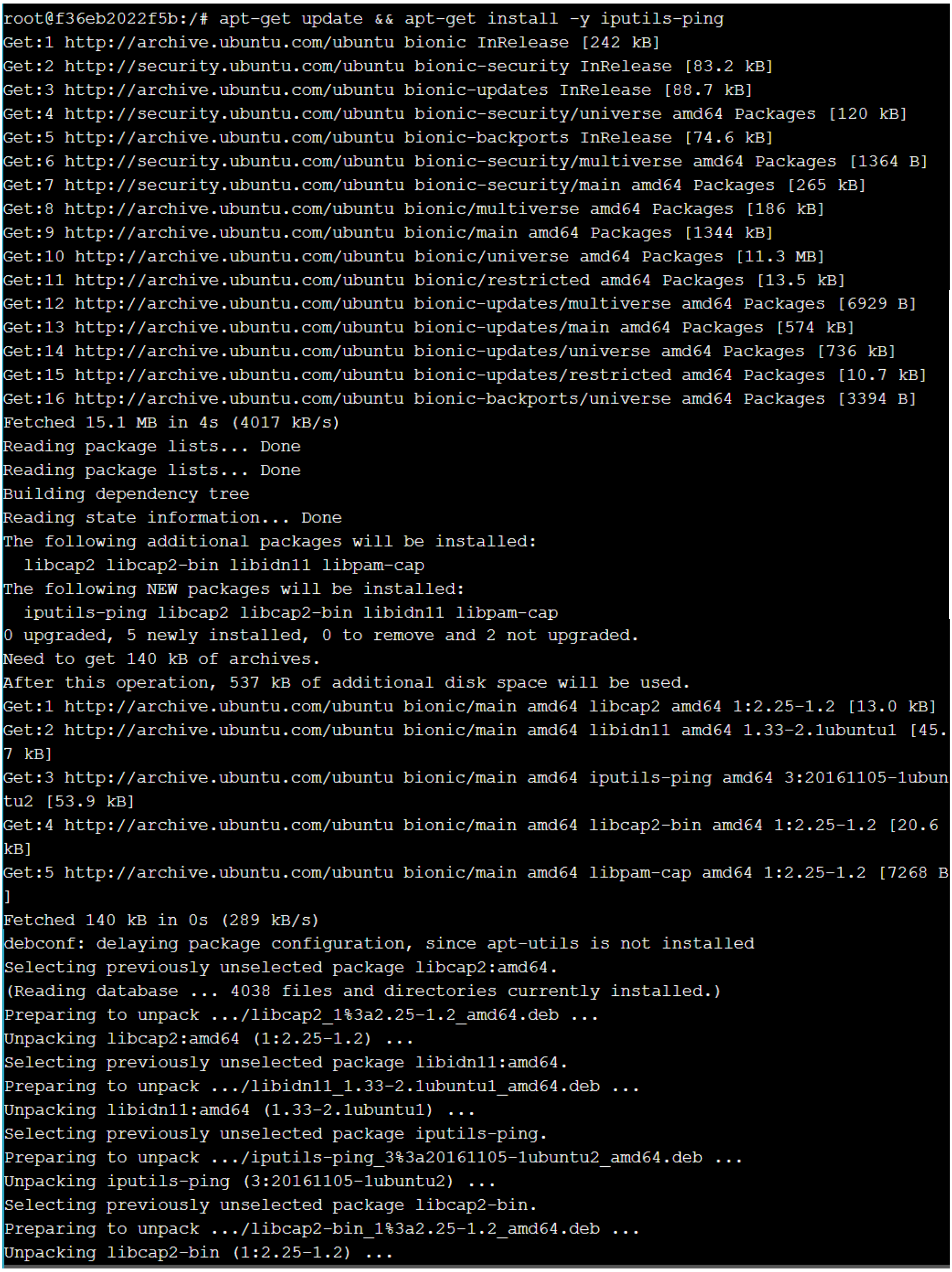


```
[node1] (local) root@192.168.0.22 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
      STATUS              PORTS                  NAMES
f36eb2022f5b        ubuntu:latest       "sleep infinity"         10 minutes ago
      Up 10 minutes                              myservice.2.y8b6v83qbpob3hceoj
i51hfuv
761e156406e1        nginx               "nginx -g 'daemon of…"   28 minutes ago
      Up 28 minutes       0.0.0.0:8080->80/tcp   web1
```

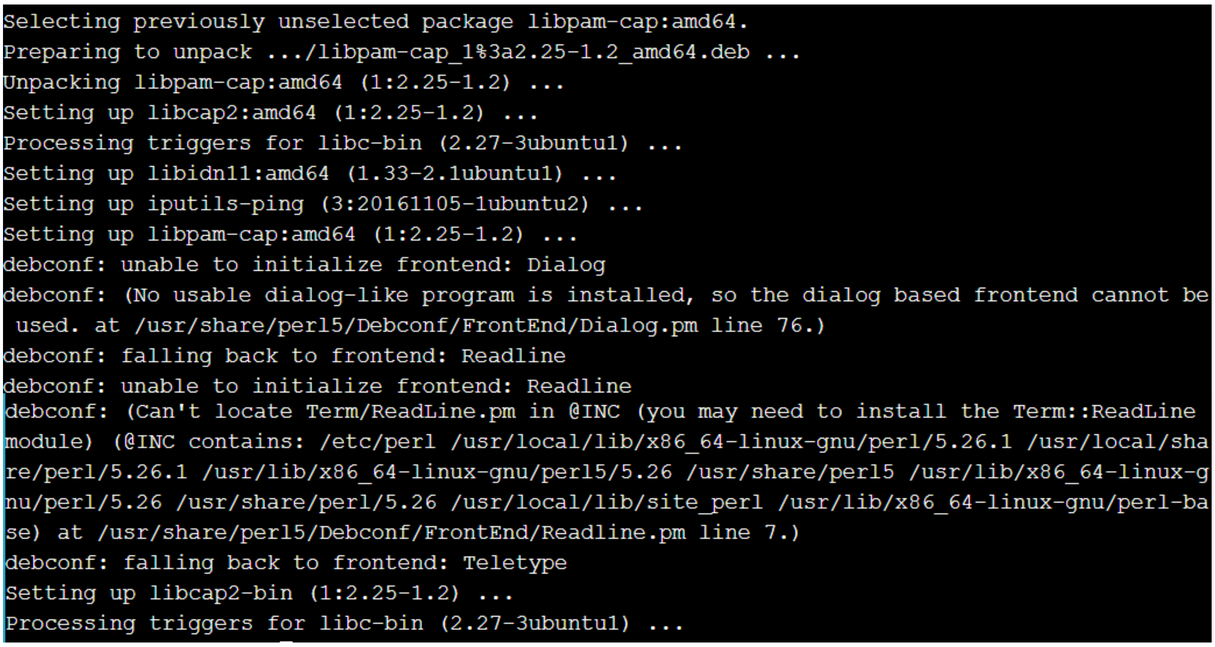
Step 5: Test service discovery



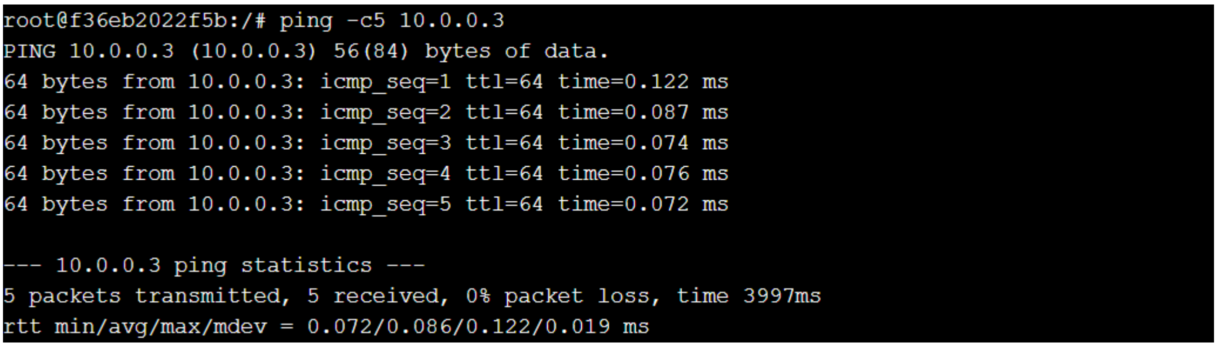
```
[node1] (local) root@192.168.0.22 ~
$ docker exec -it f36eb2022f5b /bin/bash
root@f36eb2022f5b:/#
```



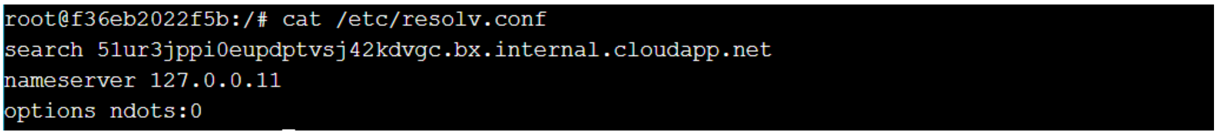
```
root@f36eb2022f5b:/# apt-get update && apt-get install -y iputils-ping
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [83.2 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [120 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [1364 B]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [265 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [6929 B]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [574 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [736 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [10.7 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [3394 B]
Fetched 15.1 MB in 4s (4017 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 2 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 3:20161105-1ubun
tu2 [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 (289 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) ...
```



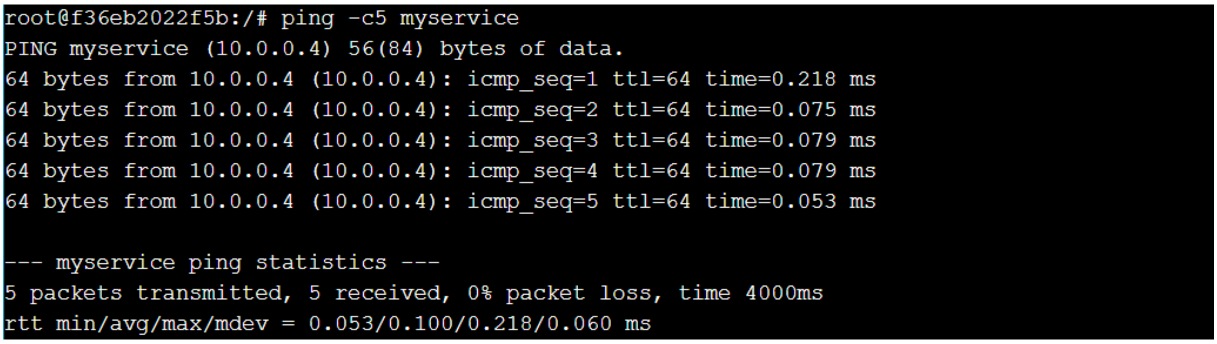
```
Selecting previously unselected package libpam-cap:amd64.
Preparing to unpack .../libpam-cap_1%3a2.25-1.2_amd64.deb ...
Unpacking libpam-cap:amd64 (1:2.25-1.2) ...
Setting up libcap2:amd64 (1:2.25-1.2) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Setting up libidn11:amd64 (1.33-2.1ubuntu1) ...
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/sha
re/perl/5.26.1 /usr/lib/x86_64-linux-gnu/perl5/5.26 /usr/share/perl5 /usr/lib/x86_64-linux-g
nu/perl/5.26 /usr/share/perl/5.26 /usr/local/lib/site_perl /usr/lib/x86_64-linux-gnu/perl-ba
se) 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) ...
```



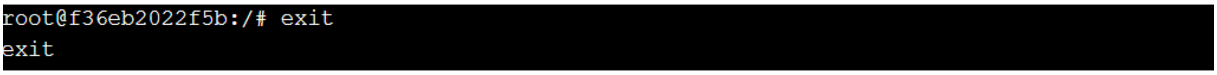
```
root@f36eb2022f5b:/# ping -c5 10.0.0.3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.122 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.074 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.076 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.072 ms

--- 10.0.0.3 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.072/0.086/0.122/0.019 ms
```



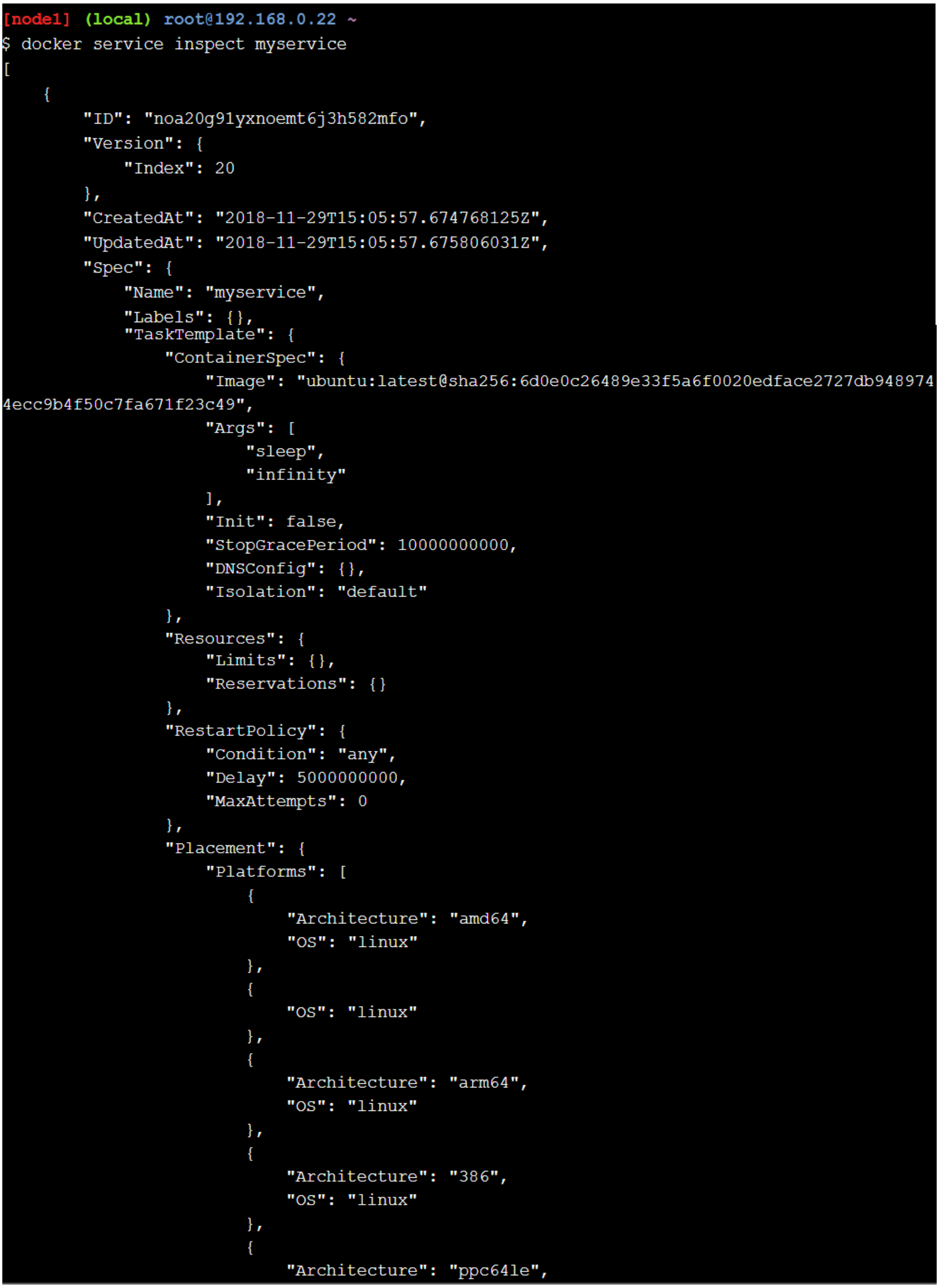
```
root@f36eb2022f5b:/# cat /etc/resolv.conf
search 51ur3jppi0eupdptvsj42kdvgc.bx.internal.cloudapp.net
nameserver 127.0.0.11
options ndots:0
```



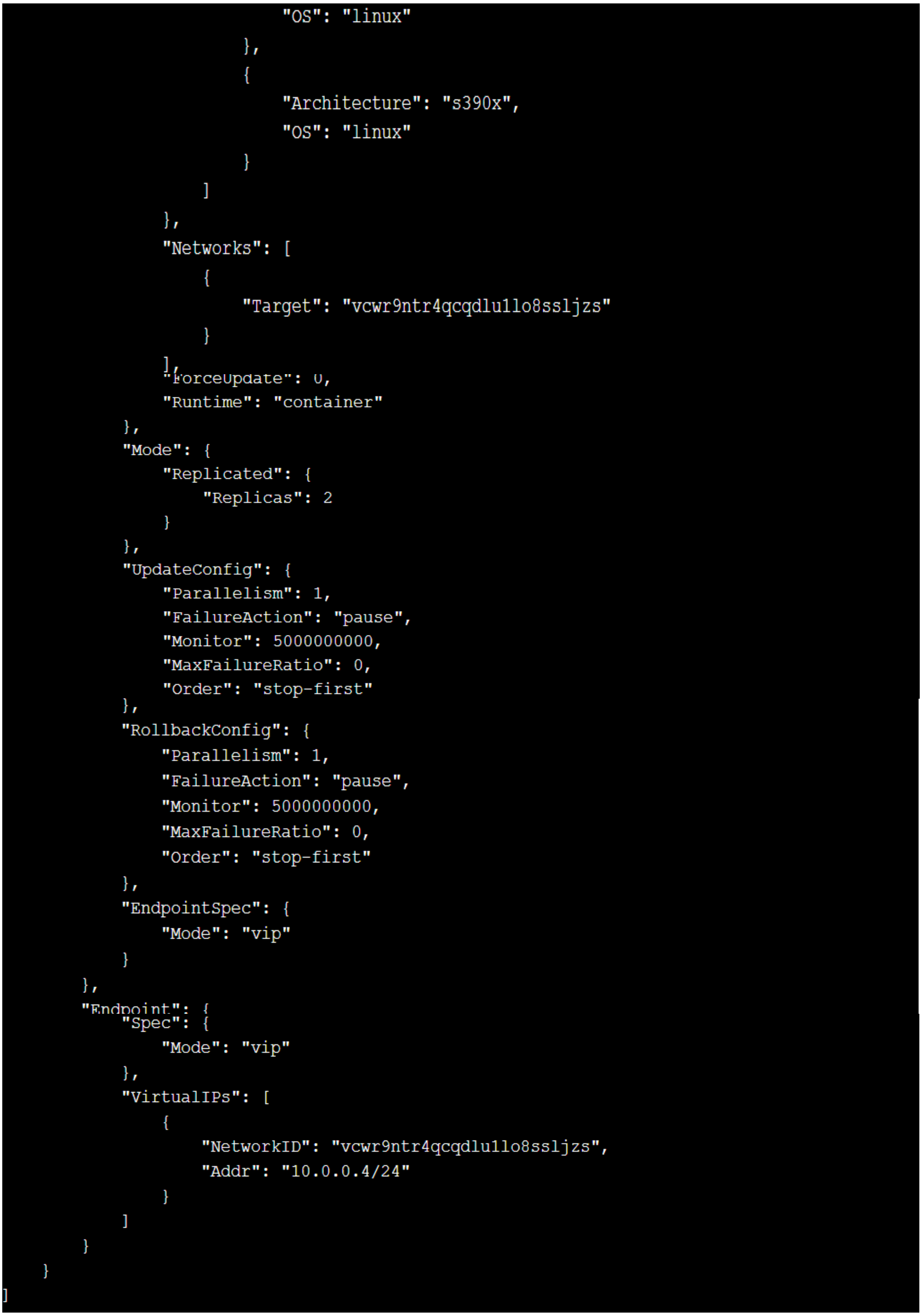
```
root@f36eb2022f5b:/# ping -c5 myservice
PING myservice (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=1 ttl=64 time=0.218 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=2 ttl=64 time=0.075 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=3 ttl=64 time=0.079 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=4 ttl=64 time=0.079 ms
64 bytes from 10.0.0.4 (10.0.0.4): icmp_seq=5 ttl=64 time=0.053 ms

--- myservice ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4000ms
rtt min/avg/max/mdev = 0.053/0.100/0.218/0.060 ms
```



```
root@f36eb2022f5b:/# exit
exit
```

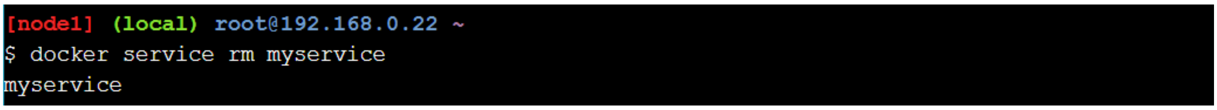


```
[node1] (local) root@192.168.0.22 ~
$ docker service inspect myservice
[
    {
        "ID": "noa20g91yxnoemt6j3h582mfo",
        "Version": {
            "Index": 20
        },
        "CreatedAt": "2018-11-29T15:05:57.674768125Z",
        "UpdatedAt": "2018-11-29T15:05:57.675806031Z",
        "Spec": {
            "Name": "myservice",
            "Labels": {},
            "TaskTemplate": {
                "ContainerSpec": {
                    "Image": "ubuntu:latest@sha256:6d0e0c26489e33f5a6f0020edface2727db948974
4ecc9b4f50c7fa671f23c49",
                    "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"
                        },
                        {
                            "Architecture": "arm64",
                            "OS": "linux"
                        },
                        {
                            "Architecture": "386",
                            "OS": "linux"
                        },
                        {
                            "Architecture": "ppc64le",
```

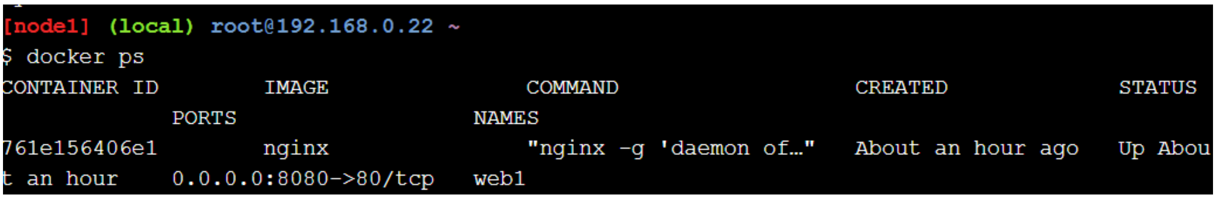


```
                            "OS": "linux"
                        },
                        {
                            "Architecture": "s390x",
                            "OS": "linux"
                        }
                    ]
                },
                "Networks": [
                    {
                        "Target": "vcwr9ntr4qcqdlu1lo8ssljzs"
                    }
                ],
                "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": "vcwr9ntr4qcqdlu1lo8ssljzs",
                    "Addr": "10.0.0.4/24"
                }
            ]
        }
    }
]
```

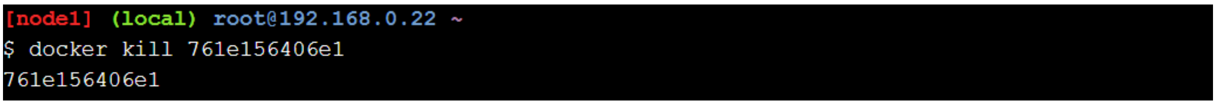
Cleaning Up



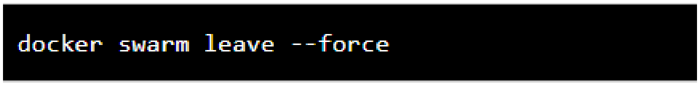
```
[node1] (local) root@192.168.0.22 ~
$ docker service rm myservice
myservice
```



```
[node1] (local) root@192.168.0.22 ~
$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
             PORTS                  NAMES
761e156406e1        nginx               "nginx -g 'daemon of…"   About an hour ago   Up Abou
t an hour     0.0.0.0:8080->80/tcp   web1
```



```
[node1] (local) root@192.168.0.22 ~
$ docker kill 761e156406e1
761e156406e1
```



```
docker swarm leave --force
```