

QNAP Network / Docker Network

[Preberi na naslovu](#)

Docker Network command

```
docker network create --help
```

```
Usage: docker network create [OPTIONS] NETWORK
```

Create a network

Options:

<code>--attachable</code>	Enable manual container attachment
<code>--aux-address map</code>	Auxiliary IPv4 or IPv6 addresses used by Network driver (default map[])
<code>--config-from string</code>	The network from which to copy the configuration
<code>--config-only</code>	Create a configuration only network
<code>-d, --driver string</code>	Driver to manage the Network (default "bridge")
<code>--gateway strings</code>	IPv4 or IPv6 Gateway for the master subnet
<code>--ingress</code>	Create swarm routing-mesh network
<code>--internal</code>	Restrict external access to the network
<code>--ip-range strings</code>	Allocate container ip from a sub-range
<code>--ipam-driver string</code>	IP Address Management Driver (default "default")
<code>--ipam-opt map</code>	Set IPAM driver specific options (default map[])
<code>--ipv6</code>	Enable IPv6 networking
<code>--label list</code>	Set metadata on a network
<code>-o, --opt map</code>	Set driver specific options (default map[])
<code>--scope string</code>	Control the networks scope
<code>--subnet strings</code>	Subnet in CIDR format that represents a network segment

Docker Command

Without UI, you can also use Docker command to create a network that belongs to `Qnet driver` and run a container with `--net` argument.

DHCP mode

Create a new network named `qnet-dhcp-eth0`

```
$ docker network create \  
-d qnet \  
--opt=iface=eth0 \  
--ipam-driver=qnet \  
--ipam-opt=iface=eth0 \  
qnet-dhcp-eth0  
  
# you get id  
9e3b67877569d6da0f5587c736c4981b3206c7f31bd22e7acdf1a347e41122c
```

Run container with Qnet DHCP mode

Run a container and connect it to `qnet-dhcp-eth0` network.

```
$ docker run --rm -it --net=qnet-dhcp-eth0 alpine ifconfig eth0  
eth0      Link encap:Ethernet  HWaddr 02:42:B8:3D:2B:07  
          inet addr:192.168.80.118  Bcast:0.0.0.0  Mask:255.255.254.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1 errors:0 dropped:1 overruns:0 frame:0  
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:60 (60.0 B)  TX bytes:0 (0.0 B)
```

Static mode

Create a new network named `qnet-static-eth0`.

```
docker network create
  -d qnet \
  --opt=iface=eth0 \
  --ipam-driver=qnet \
  --ipam-opt=iface=eth0 \
  --subnet=192.168.80.0/23 \
  --gateway=192.168.80.254 \
  qnet-static-eth0

# you get id
85fbe06a66d82ba8109d304e1b891598d7c21e9f6c9a99a34f586250c7d8b92d
```

Run container with Qnet Static mode

Run a container and connect it to `qnet-static-eth0` network.

```
$ docker run --rm -it --net=qnet-static-eth0 --ip=192.168.80.119 alpine ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:42:55:B1:84:92
          inet addr:192.168.80.119  Bcast:0.0.0.0  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:4 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:940 (940.0 B)  TX bytes:0 (0.0 B)
```

Docker Compose

There is another way to create a container with Qnet driver.

Write **docker-compose.yml**

Specify `qnet` driver for this network.

docker-compose.yml

```
version: '2'

services:
  qnet_dhcp:
```

```
image: alpine
command: ifconfig eth0
networks:
  - qnet-dhcp
```

```
qnet_static:
  image: alpine
  command: ifconfig eth0
  networks:
    qnet-static:
      ipv4_address: 192.168.80.119
```

```
networks:
  qnet-dhcp:
    driver: qnet
    driver_opts:
      iface: "eth0"
    ipam:
      driver: qnet
      options:
        iface: "eth0"
```

```
qnet-static:
  driver: qnet
  driver_opts:
    iface: "eth0"
  ipam:
    driver: qnet
    options:
      iface: "eth0"
  config:
    - subnet: 192.168.80.0/23
      gateway: 192.168.80.254
```

Revision #3

Created 2023-05-01 01:31:52 UTC by Dušan Fefer

Updated 2023-05-01 15:59:39 UTC by Dušan Fefer