

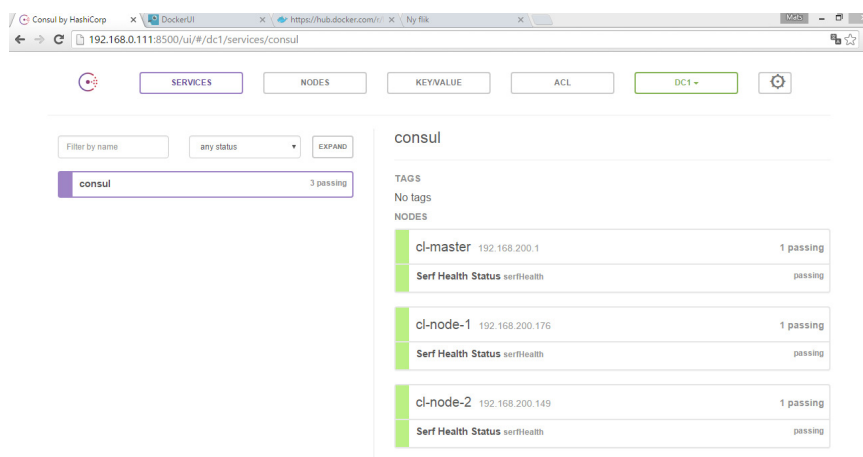


Working with multiple networks in Docker

In Docker we have now the ability to work with multi-host networking. This document describes how to do that.

Document is based on my cluster environment, but could easily transform to your environment.

Cluster environment are from Hypriot Cluster Lab running on Raspberry PI 2.



Configuring multiple host network

Does it exist any networks on my Docker host?

Step 1: Connect to your Docker host, with ssh or (in my case PUTTY).

Step 2: Run the following command: `docker -H name_for_your_docker_host:port_number_for_Docker network ls`, press Enter.

In my case: `docker -H cl-master:2378 network ls`, press Enter.

```
HyprIoTOS: root@cl-master in ~
$ docker -H cl-master:2378 network ls
NETWORK ID          NAME                DRIVER
7c7b3f73221e       cl-node-2/host     host
d5e3b39b3782       cl-master/none     null
d92a18c33c23       cl-node-1/host     host
58d7c3f8ebc0       cl-node-1/bridge   bridge
6ee30e7f1959       cl-node-2/none     null
188c2a42e840       cl-master/bridge   bridge
7e0e77c816ce       cl-master/host     host
e82bbb6f5d45       cl-node-1/none     null
77967c3de499       cl-node-2/bridge   bridge
HyprIoTOS: root@cl-master in ~
$
```

These are the default networks.

Create new overlay network named easec1-net.

Step 3: To create a new overlay network, use following command: `docker -H name_for_your_docker_host:port_number_for_Docker network create -d overlay name_for_your_network`, press Enter.

`docker -H cl-master:2378 network create -d overlay easec1-net`, press Enter.

```
HyprIoTOS: root@cl-master in ~
$ docker -H cl-master:2378 network create -d overlay easec1-net
45dbcabda19f0957cc7659e57b44db095446016471dcbbe3f9f8a09c540ed97e
HyprIoTOS: root@cl-master in ~
$ docker -H cl-master:2378 network ls
NETWORK ID          NAME                DRIVER
58d7c3f8ebc0       cl-node-1/bridge   bridge
d92a18c33c23       cl-node-1/host     host
7c7b3f73221e       cl-node-2/host     host
6ee30e7f1959       cl-node-2/none     null
d5e3b39b3782       cl-master/none     null
45dbcabda19f       easec1-net          overlay
e02bbb613d43       cl-node-1/none     null
77967c3de499       cl-node-2/bridge   bridge
188c2a42e840       cl-master/bridge   bridge
7e0e77c816ce       cl-master/host     host
HyprIoTOS: root@cl-master in ~
```

As you can see, we have created our own overlay network. Any container started with these network, could communicate with any other container started with the same network.

Start using the new overlay network

Best way to see how it works, is to start some new containers at different nodes.

Start containers on cl-node-1.

Step 1: Use the following command: `docker -H name_for_your_docker_host:port_number_for_Docker run -itd --name=webserver --net=name_for_your_network --env="constraint:node==name_for_your_other_node" hypriot/rpi-nano-httpd`, press Enter.

`docker -H cl-master:2378 run -itd --name=webserver --net=easecl-net --env="constraint:node==cl-node-1" hypriot/rpi-nano-httpd`, press Enter.

```
HyprIoTOS: root@cl-master in ~
$ docker -H cl-master:2378 run -itd --name=webserver --net=easecl-net --env="constraint:node==cl-node-1" hypriot/rpi-nano-httpd
e18b46aa89015352217faa3f74fbc9336a52ff947a32f5613ecb3c1d071de86e
HyprIoTOS: root@cl-master in ~
$
```

With `env="constraint:node==name_for_node"` we could tell Docker swarm master to create the container on these node, normally will it create on first possibly available node.

List the running containers that we have now.

Step 2: Run the command: `docker -H name_for_your_docker_host:port_number_for_Docker ps`, press Enter.

`docker -H cl-master:2378 ps`, press Enter.

```
HyprIoTOS: root@cl-master in ~
$ docker -H cl-master:2378 ps
CONTAINER ID        IMAGE                               COMMAND
e18b46aa8901        hypriot/rpi-nano-httpd            "/httpd 80"
efb1f1b18ba4        hypriot/rpi-dockerui              "/dockerui -e http://"
843b9d0cd896        hypriot/rpi-consul:0.6.0          "/consul agent -serve"
8cf5f1061861        hypriot/rpi-swarm                 "/swarm join --advert"
e8335c8407a8        hypriot/rpi-consul:0.6.0          "/consul agent -serve"
d3a2950190d9        hypriot/rpi-swarm                 "/swarm join --advert"
8edf0689b89b        hypriot/rpi-swarm                 "/swarm manage consul"
ge_1
b10a913359b9        hypriot/rpi-consul:0.6.0          "/consul agent -serve"
0664561796ce        hypriot/rpi-swarm                 "/swarm join --advert"
HyprIoTOS: root@cl-master in ~
$
```

To see it work, we need the final piece, we start a new container that have a web client that talks with the container with the webserver.

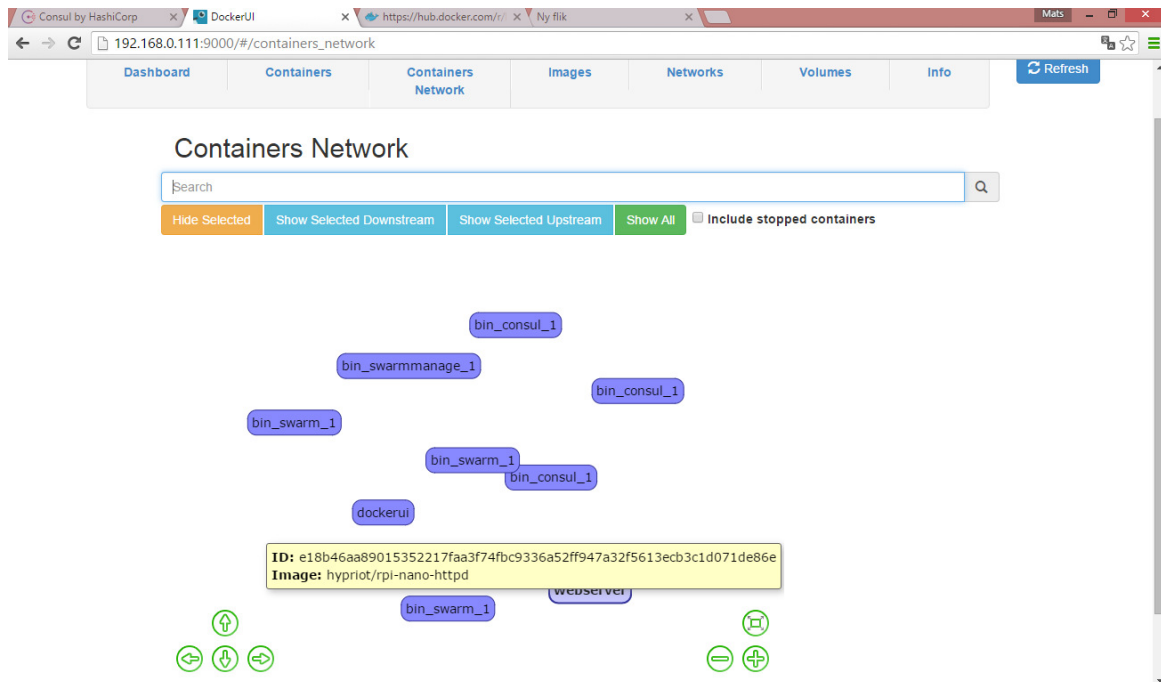
Step 3: Run the command: `docker -H name_for_your_docker_host:port_number_for_Docker run -it --rm --net=name_for_your_network --env="constraint:node==name_for_your_other_node" hypriot/armhf-busybox wget -O- http://webserver/index.html`, press Enter.

```
docker -H cl-master:2378 run -it --rm --net=easec1-net --env="constraint:node==cl-node-2" hyprriot/armhf-busybox wget -O- http://webserver/index.html, press Enter.
```

```
HyprriotOS: root@cl-master in ~
$ docker -H cl-master:2378 run -it --rm --net=easec1-net --env="constraint:node==cl-node-2" hyprriot/armhf-busybox wget -O- http://webserver/index.html
Connecting to webserver (10.0.0.2:80)
<html>
<head><title>Pi armed with Docker by Hyprriot</title>
  <body style="width: 100%; background-color: black;">
    <div id="main" style="margin: 100px auto 0 auto; width: 800px;">
      
    </div>
  </body>
</html>
-
100% |*****| 304 0:00:00 ETA
HyprriotOS: root@cl-master in ~
$
```

These steps explains how easy you could set up overlay networks in Docker.

If you use DockerUI, you could see your network, as shown in picture below:



I have hoover over the webserver.

Thanks for the team behind Hyprriot, visit there website at <http://blog.hyprriot.com/> .