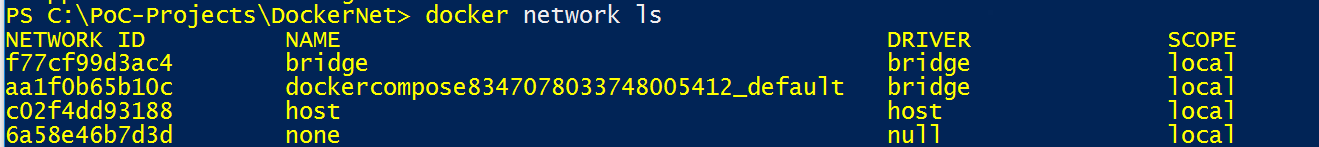
Container Discovery using Docker Network

## Setting up a Docker Network:

Docker itself by-default provides two kinds of network drivers – 1.) bridge and 2.) host

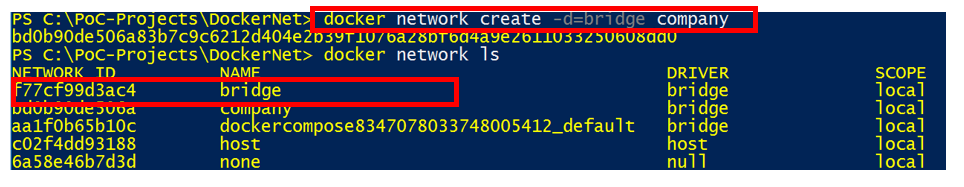
Use the following commands to see them all: *docker network ls*



To create a new network use the following command

*docker network create –d=bridge company*

In this example, *bridge* is used as the network driver



Keep in mind this is just a virtual network accessible inside only the host and not accessible from outside host.

## Attaching a Container to the Network:

To bind a container to use this new virtual network, use the following docker run command:



In this example, image *swarmgs/customer* is named as a container *customer-api* and will be using network of *company*. *docker ps* command produces following output:



## Container discovery:

Docker network are pre-loaded with features of container discovery.

In order to discover the container A by container B, both the containers need to be on same virtual network and container B needs to just use name (in the above example, customer-api) to discover the container.

See the example below,



In this example, balance-api container discovers the customer-api by its name by virtue of being in same network “company”. This way, we don’t need to expose the dependency api of customer-api on main host.

## Setting up the Docker network using Docker Compose:

Docker compose has a feature of setting up a default network with the name of compose file. For example, on running, *dockercompose.yml* file will create a virtual network as *dockercompose\_default.*

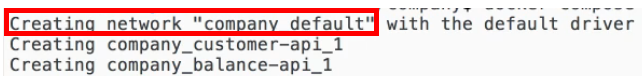
Rest of the things remain almost identical in compose file (company.yml) as we do manually using command line. Following is the compose file that would achieve same result as the command lines above:



Run the compose file in detached mode as following:



This outputs as following:



Note: in case wondering about command to stop a particular api say, customer-api using compose, following is the command:



## Load balancing:

Ideal state expected:

