Title: How to reach docker containers by name instead of IP address?

Post Body:

Is there a way I can reach my docker containers using names instead of ip addresses?

I've heard of pipework and I've seen some dns and hostname type options for docker, but I still am unable to piece everything together.

Thank you for your time.

I'm not sure if this is helpful, but this is what I've done so far:

- installed docker container host using docker-machine and the vmwarevsphere driver
- started up all the services with docker-compose
- I can reach all of the services from any other machine on the network using IP and port

I've added a DNS alias entry to my private network DNS server and it matches the machine name that's used by docker-machine. But the machine always picks up a different IP address when it boots and connects to the network.

I'm just lost as to where to tackle this:

- network DNS server
- docker-machine hostname
- docker container hostname
- probably some combination of all of them

I'm probably looking for something similar to this question:

How can let docker use my network router to assign dhcp ip to containers easily instead of pipework?

Any general direction will be awesome...thanks again!

Accepted Answer:

Docker `1.10` has a built in DNS. If your containers are connected to the same *user defined* network (create a network `docker network create my-network` and run your container with `--net my-network`) they can reference each other using the container name. (Docs).

Cool!

One caveat if you are using Docker compose you know that it adds a prefix to your container names, i.e. `<project name>_<service name>-#`. This makes your container names somewhat more difficult to control, but it might be ok for your use case. You can override the docker compose naming functionality by manually setting the container name in your compose template, but then you wont be able to scale with compose.

Highest Rated Answer:

Create a new bridge network other than docker0, run your containers inside it and you can reference the containers inside that network by their names.

> Docker daemon runs an embedded DNS server to provide automatic service discovery for containers connected to user-defined networks. Name resolution requests from the containers are handled first by the embedded DNS server.

Try this:

```
docker network create <network name> docker run --net <network name> --name test busybox nc -l 0.0.0.0:7000 docker run --net <
```

First, we create a new network. Then, we run a busybox container named **test** listening on port 7000 (just to keep it running). Finally, we ping the test container by its name and it should work.