Title: Using the host ip in docker-compose

Post Body:

I want to create a **docker-compose** file that is able to run on different servers.

For that I have to be able to specify the host-ip or hostname of the server (where all the containers are running) in several places in the **docker-compose.yml**.

E.g. for a consul container where I want to define how the server can be found by fellow consul containers.

```
consul:   image: progrium/consul   command: -server -advertise 192.168.1.125 -bootstrap
```

I don't want to hardcode 192.168.1.125 obviously.

I could use **env_file:** to specify the hostname or ip and adopt it on every server, so I have that information in one place and use that in docker-compose.yml. But this can only be used to specifiy environment variables and not for the advertise parameter.

Is there a better solution?

Accepted Answer: None

Highest Rated Answer:

docker-compose allows to use environment variables from the environment running the compose command.

See documentation at https://docs.docker.com/compose/compose-file/#variable-substitution

Assuming you can create a wrapper script, like @balver suggested, you can set an environment variable called `EXTERNAL_IP` that will include the value of `$(docker-machine ip)`.

Example:

```
#!/bin/sh export EXTERNAL_IP=$(docker-machine ip) exec docker-compose $@
```

and

```
# docker-compose.yml version: '2' services:   consul:     image: consul     environment:       - 'EXTERNAL_IP=${EXTERNAL_IP}'
```

Unfortunately if you are using random port assignment, there is no way to add `EXTERNAL_PORT`, so the ports must be linked statically.

PS: Something very similar is enabled by default in HashiCorp Nomad, also includes mapped ports. Doc: https://www.nomadproject.io/docs/jobspec/interpreted.html#interpreted_env_vars