Title: Springboot client unable register with Eureka using Docker container id

Post Body:

I have several microservices running in Docker Data Center. I have the same Eureka configuration across the services/applications. But some of the applications are registering with their `eth0` IP address instead of the container ID.

I have tried setting the `preferIpAddress` as `false` but it is not enforcing all the time.

There is no pattern. The same service which registers with container ID during the previous deployment gets registered with IP the other time. I want my services to register always with its container id. Is there a way to enforce it or am I missing something?

Note: I have also cleared all the old docker images from the registry, deployment nodes and tried from the scratch as well.

Eureka Server Config:

`eureka:    instance:        hostname: discovery    client:        registerWithEureka: false    fetchRegistry: false    serviceUrl:`

Microservices client config (It is same across all the microservices)

`eureka:    client:        healthcheck:        enabled: true    serviceUrl:        defaultZone: http://discovery:8761/eureka/    inst`

Eureka Dashboard Snapshot:

Accepted Answer:

In docker the container-id will be set as the hostname of the container by default. Containers can talk with each other using container-id (or here hostname)

So this issue can be solved by preferring hostname instead of ip.

But only way of making sure that registration happens through hostname is by setting `eureka.instance.hostname` [reference](#)

In docker you can set the container-id at run-time from the hostname by using the entry point as shell script (for example start.sh) and the script should be something similar to

`#!/bin/sh export HOST_NAME=`hostname` java -Djava.security.egd=file:/dev/./urandom -Xmx1024m -jar /app.jar`

and make sure you add `eureka.instance.hostname=${HOST_NAME}` in your application.yml

or you can reuse the `HOSTNAME` variable which is set by default in Docker and configuration becomes `eureka.instance.hostname=${HOSTNAME}`

I have added the same info in [documentation](#)

Update: looks like this can also be fixed by using `endpoint_mode: dnsrr` in the compose file (have to confirm). [Refer this](#)

Highest Rated Answer:

Try setting `eureka.instance.hostname` in your client config.