

Title: Spring Cloud Netflix and Docker Compose - cannot register services with eureka

Post Body:

I'm trying to use docker-compose to run 2 simple services locally (Ubuntu): a eureka server, and config server (which is also a eureka client). Both of these have simple dockerfiles that run `java -jar`, expose their ports, and individually work fine. I also tried to add `eureka.client.service-url.defaultZone=http://company-service-discovery:8761/eureka` to see if it would register itself, and it worked.

My config server cannot successfully register to the eureka server, and I've googled it and nothing I've seen helped me solve this.

According to docker-compose documentation at <https://docs.docker.com/compose/networking/> :

By default Compose sets up a single network for your app. Each container for a service joins the default network and is both reachable by other containers on that network, and discoverable by them at a hostname identical to the container name.

With the following example web should be able to use `postgres://db:5432` to communicate with the database.

```
version: '3' services: web: build: . ports: - '8000:8000' db: image: postgres ports: - '8001:5432'
```

I have used the same method to configure my services, but my config server gets connection refused when trying to register:

### docker-compose.yml

```
version: '3.3' services: company-service-discovery: build: company-service-discovery/ ports: - '8761:8761'
```

### config server bootstrap.yml

```
server: port: 8888 management: security: enabled: false spring: application: name: company-config-server
```

### eureka server bootstrap.yml

```
spring: application: name: company-service-discovery server: port: 8761 management: security: enabled: false
```

### exception

```
2017-07-26 14:25:05.738 WARN 1 --- [nfoReplicator-0] c.n.d.s.t.d.RetryableEurekaHttpClient : Request execution failed with
```

### question

Is there anything wrong with my configuration ? How can I make it work ?

Let me know if there's any info missing, I'll gladly give any info I can.

Accepted Answer: None

Highest Rated Answer:

Add the `defaultZone` configuration to the Eureka server's properties too (and change the `service-url` to `serviceUrl` in your **config server bootstrap.yml**).

### eureka server bootstrap.yml

```
spring: application: name: company-service-discovery eureka: client: serviceUrl: defaultZone: http://company-service-discovery:8761/eureka
```