

Title: Consul and Spring Boot services in Docker - not deregistering

Post Body:

So we have Java microservices written with Spring-Boot, using Consul for service discovery and config management and running in Docker containers. All of it is working, but when a container dies or a service restarts the old service-id never goes away in Consul and the service forever after shows as 'Failing' in the Consul UI, even though the new container has registered and shows all Green.

We are not using heartbeat - but I cannot find much documentation on what the difference between heartbeat and healthcheck are for Consul.

Here's my bootstrap.yml

```
spring:
  application:
    name: my-service
  cloud:
    config:
      enabled: false
  consul:
    host: ${discovery.host}
```

There are other settings to enable heartbeat, but the docs say something about this putting more stress on the Consul cluster.

Has anyone managed to get Consul and Spring Boot/Docker services to actually de-register automatically? It actually doesn't cause any real problems, but it makes the Consul UI pretty useless to actually monitor for up/down services.

Accepted Answer: None

Highest Rated Answer:

Consul doesn't automatically deregister services.

See <https://groups.google.com/forum/#!topic/consul-tool/sIV5xfWRpEE> for the hint about the same question. According to that thread you need to either update the config or perform an [Agent API call](#). Since the agent is the source of truth, you shouldn't try to update via Catalog API. See [GitHub](#) for details. They also mention at the Google group that you don't necessarily have to deregister services if the node goes down gracefully, but that doesn't seem to be your use case.

Please have a look at [Consul not deregistering zombie services](#) for hints about automating the service de-registration using either the api or tools like [registrator](#).