Title: Unable to connect to Kafka run in container from Spring Boot app run outside container
Post Body:

I'm running kafka locally via:

docker-compose.yml

```
  zookeeper:    image: 'bitnami/zookeeper:latest'    ports:    - 2181:2181    environment:    - ALLOW_ANONYMOUS_LOGIN
```

My Spring Boot application is run with:

application.yml:

```
spring:  application:    name: testkafka  kafka:    bootstrap-servers: localhost:9092  server:    port: 8080
```

When I run it and it tries to send to a topic on kafka, I get:

```
org.springframework.kafka.KafkaException: Reply timed out    at org.springframework.kafka.requestreply.ReplyingKafkaTemplate.
```

If I run the spring boot from inside a docker container (using one docker compose file), then it *does* work:

Running Both in the Same Compose:

```
version: '3.0' services:    service1:    build: ./Service    ports:    - 8080:8080    - 5005:5005    links:    - zo
```

How can I get the kafka container to allow connections from outside itself/docker?

EDIT: Tried these changes:

```
kafka:    image: 'bitnami/kafka:latest'    ports:    - 9092:9092    environment:    - KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181
```

and:

```
spring:  application:    name: testkafka  kafka:    bootstrap-servers: kafka:9092  server:    port: 8080
```

This still times out

Accepted Answer:

> If I run the spring boot from inside a docker container (using one docker compose file), then it does work

It shouldn't work, actually. Kafka isn't running as part of the application, so this section is not pointing at the Kafka container.

```
kafka:    bootstrap-servers: localhost:9092
```

It would need to be `kafka:9092` within the Docker network.

> and ... `bootstrap-servers: kafka:9092` ... still times out

That's because you still have `KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://localhost:9092`

Both inside and outside of Docker network, by specifying `KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://localhost:9092`, you're saying that your clients receive the Kafka bootstrap connection as `localhost:9092`, which would work outside of the Docker network only because you have made both a port forwarding and your container is running on localhost, however, inside the Docker network, as mentioned, localhost would mean *that* application container, not the broker.

The solution would be to create two port mappings via that property, as discussed in length by [this blog post](#)

Plus, [Confluent provides a fully configured Docker Compose](#) with appropriate mappings that'll work inside and out of Docker

```
    ports:    - "9092:9092"    - "29092:29092"    environment:    KAFKA_BROKER_ID: 1    KAFKA_ZOOKEEPER_CONNECT:
```

For applications within Docker network, use `kafka:9092`, for applications outside, use **localhost:29092**

EDIT The bitnami config variables have changed. Refer this section of the README

https://github.com/bitnami/bitnami-docker-kafka/blob/master/README.md#accessing-kafka-with-internal-and-external-clients

Highest Rated Answer: None