Title: Docker (Spring Boot or Thorntail) and Keycloak

Post Body:

i have a Problem Running Spring Boot and Keycloak both in docker containers.

I started with Keycloak with mysql as db running in docker.

```
services:   mysql:     image: mysql:5.7     container_name: mysql     volumes:       - mysql_data:/var/lib/mysql     environme
```

Then i added my realm (SpringBootKeycloak), my client (testclient), and a user with role 'user'. After that i added spring-security to my Spring-boot-application. And edited my application.yml

```
spring:   main:     banner-mode: 'off'   application:     name: testclient     version: @project.version@   jpa:     hibernate
```

accoring to that i added my SecurityConfig:

```
  /**    * Secure appropriate endpoints    */   @Override   protected void configure(HttpSecurity http) throws Exception {
```

Running my SpringBoot-Application locally is working fine. I have to login with keycloak and get redirected to localhost:8090. But when i add my SpringBoot-Application to my docker-compose and start it in a container i get still to keycloak for login, but when i should redirect i get a 403.

```
  testclient:     image: testclient     container_name: testclient     environment:       JAVA_OPTS: '-agentlib:jdwp=transport
```

with following container log:

```
{'@timestamp':'2018-08-16T11:50:11.530+00:00','@version':'1','message':'failed to turn code into token','logger_name':'org.key
```

I can't figure out how to solve this...

EDIT 1: One more information: I'm running docker on Windows.

EDIT 2: A SOLUTION

My Working solution contains following:

1. Step, add keycloak as hosts

   To make things work, you’ll need to make sure to add the following to your hosts file (/etc/hosts on Mac/Linux, c:\Windows\System32\Drivers\etc\hosts on Windows).

   127.0.0.1 keycloak

   This is because you will access your application with a browser on your machine (which name is localhost, or 127.0.0.1), but inside Docker it will run in its own container, which name is keycloak.

2. Step

Inner Docker port and published port needs to be same:

```
services:   mysql:     image: mysql:5.7     container_name: mysql     volumes:       - mysql_data:/var/lib/mysql     environme
```

Step 3: keycloak definition in application.yml for Spring boot edited auth-server-url:

```
    keycloak:   realm: SpringBootKeycloak   auth-server-url: http://keycloak:8080/auth   <--- edited   resource: testclient
```

The ugly thing coming with this solution: You cant map your Docker Port onto another port to access from url. ports: - 8080:8080 i spend a lot of time testing other combinations, with the result that the access url port has to be the same as inner docker port (8080 in my case).

EDIT 4:

Same thing is working with Thorntail.

To change the port for Keycloak add...

```
environment:   JAVA_OPTS: '-Djboss.socket.binding.port-offset=10 -Xms64m -Xmx512m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=2
```

... for keycloak in docker-compose. -Djboss.socket.binding.port-offset=10 sets default port (8080) + offset (10) The rest are default values for keycloak. Don't forget to edit 'ports' and 'auth-server-url'

Accepted Answer:

I think your problem is `auth-server-url: http://localhost:8180/auth`. That `localhost` effectively has a different meaning when your app is running inside a docker container.

Inside the container it needs to be the name of the container i.e. `keycloak`. This is a bit awkward as when you connect to keycloak from your host machine you'd want to use `localhost` but the token issuer url needs to match to the url on which the token was requested (otherwise the token is rejected) so you end up having to put `keycloak` into your etc/hosts file.

You are in good company with this problem - I've encountered this working with Activiti. And you can find the JHipster project dealing with it in the same way - they say:

> To make things work, you'll need to make sure to add the following to your hosts file (`/etc/hosts` on Mac/Linux, `c:\Windows\System32\Drivers\etc\hosts` on Windows).
>
> ```
> 127.0.0.1 keycloak
> ```
>
> This is because you will access your application with a browser on your machine (which name is `localhost`, or `127.0.0.1`), but inside Docker it will run in its own container, which name is `keycloak`.

Highest Rated Answer:

Thank you for the question and the answer! I was struggling with this for quite sometime and getting connection refused error. The reason was because I was mapping 8180:8080, and as soon as I mapped **8080:8080** everything started working! Thanks again!