

Title: Docker maven plugin ClientProtocolException (Windows 10 using Docker Toolbox)

Post Body:

I am trying to build a docker image using the docker-maven-plugin (provided by spotify: <https://github.com/spotify/docker-maven-plugin>) but things aren't really working out. At first I got this exception:

```
org.apache.http.conn.HttpHostConnectException: Connect to localhost:2375
```

I found out that I had to create an env. variable to fix this (<https://github.com/spotify/docker-maven-plugin/issues/135>):

```
set DOCKER_HOST=tcp://192.168.99.100:2376
```

So after setting that env. variable I keep getting this ClientProtocolException:

```
[ERROR] Failed to execute goal com.spotify:docker-maven-plugin:0.3.258:build (default-cli) on project docker_micro_maven: Exce
```

I have no idea how I could fix this, any input would be greatly appreciated.

Accepted Answer:

You have to configure virtualbox port forwarding with host listen on 2375 port :

```
> VBoxManage modifyvm 'default' --natpf1 'guestssh,tcp,,2375,,2376'
```

guestssh is the name of port forwarding, you can choose the name.

if VBoxManage isn't recognise, you can replace it by full path :

```
'pathVirtualBox\VBoxManage.exe'
```

And after if you have an issue like 'The server failed to respond with a valid HTTP response' copy the certs files from

```
'%USER%.docker\machines\certs'
```

to

```
'%USER%.docker'
```

cause docker-maven-plugin read this files in '%USER%.docker' (don't copy directory certs, just files).

I think we can override maven properties to replace port 2375 and certs path, but i don't still find it.

Highest Rated Answer:

fixed this by:

```
<plugin>                                <groupId>com.spotify</groupId>                                <artifactId>docker-maven-plugin</artifactId>
```

Important are these two tags:

```
<dockerHost>https://192.168.99.100:2376</dockerHost> <dockerCertPath>/Users/your_user/.docker/machine/machines/default</docker
```

I am using a dockerfile, which path you have to define with this tag:

```
<dockerDirectory>src/main/docker</dockerDirectory>
```

Now you can build your jar and generate docker image via:

```
mvn package docker:build
```