

Stack Overflow is a community of 8.2 million programmers, just like you, helping each other. Join them; it only takes a minute:

How to connect with JMX from host to Docker container in Docker machine?

[Ask Question](#)

3

When I have running Docker container directly at my host, it is possible to connect to it without any problems.

My host has network 192.168.1.0/24 and IP address of the host is 192.168.1.20. My Docker container has IP address 172.17.0.2. When I connect to 172.17.0.2:1099 from jconsole it works.

When I put this service into Docker machine, it is not possible to connect to it.

My Docker machine has IP 192.168.99.100 and container in it has IP address 172.17.0.2 but when I use jconsole to connect to 192.168.99.100:1099 it does not work.

To repeat it:

192.168.1.20 --- 172.17.0.2:1099 works

192.168.1.20 --- (192.168.99.100 --- 172.17.0.2:1099) and connecting to 192.168.99.100:1099 from my host does not work.

It is worth to say that I can access services containerized in Docker machine via external IP address of the Docker machine, e.g. this will work:

192.168.99.100 --- (192.168.99.100:8080 --- 172.17.0.2:8080)

But when I use JMX it just does not work.

It is Tomcat service. I have this in scripts which starts Tomcat instance:

```
CATALINA_OPTS="-Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=n \
-Dcom.sun.management.jmxremote.port=1099 \
-Dcom.sun.management.jmxremote.rmi.port=1099 \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.ssl=false \
-Djava.rmi.server.hostname=IP address of Docker container
```

[tomcat](#) [networking](#) [docker](#) [jmx](#) [docker-machine](#)

[share](#) [improve this question](#)

asked Feb 17 '16 at 19:55



[stewenson](#)
547 9 23

It is started by docker-java library ... it is not so important, but when I do "docker-machine ssh" and in it "docker ps -a" I can see that my container has this port bindings / exposings: ad6be9184855 company/tomcat:8.0.30 0.0.0.0:1099->1099/tcp, 0.0.0.0:8000->8000/tcp, 0.0.0.0:8009->8009/tcp, 0.0.0.0:8080->8080/tcp – [stewenson](#) Feb 17 '16 at 20:20

[add a comment](#)

1 Answer

[active](#)

[oldest](#)

[votes](#)

3

I think the problem is probably the value of the `java.rmi.server.hostname` property. This needs to be the hostname or IP address that should be used by the JMX client to connect to your JVM. That is in the first case where you connect to your container directly using `172.17.0.2:1099`, this setting needs to be set to `172.17.0.2`. In the latter case where you access the container through the docker machine on `192.168.99.100:1099`, the setting needs to be set to `192.168.99.100`.

During my research for a very similar question (which was deleted in the meantime) I stumbled across a blog entry (which was deleted in the meantime as well). Although It's rather old it gave me an idea how the JMX connectivity works:

1. The JMX registry listens on port `<com.sun.management.jmxremote.port>` of the container
2. If you connect to the registry with JConsole, the registry provides the JMX service URL to the client.
3. This URL is used by the client to obtain the JMX objects

The service URL looks like this `service:jmx:rmi:///jndi/rmi://<java.rmi.server.hostname>:<com.sun.management.jmxremote.rmi.port>/jmxrmi`. That is in your case `service:jmx:rmi:///jndi/rmi://172.17.0.2:1099/jmxrmi`. As this address is only reachable from within the docker machine, connecting from remote is not possible. In my question I cover the same problem in regards to the RMI port...

There doesn't seem to be an out-of-the-box solution to this problem. However one can provide both JMX port and the external hostname (or IP) on startup of the container as environment variables, as suggested [here](#). These could then be used in the JMX config:

```
docker run -p 1099:1099 \
-e "JMX_HOST=192.168.99.100" \
```

```
-e "JMX_PORT=1099" \  
company/tomcat:8.0.30
```

and

```
CATALINA_OPTS="... \  
-Dcom.sun.management.jmxremote=true \  
-Dcom.sun.management.jmxremote.port=$JMX_PORT \  
-Dcom.sun.management.jmxremote.rmi.port=$JMX_PORT \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.ssl=false \  
-Djava.rmi.server.hostname=$JMX_HOST"
```

Not very nice, but it should work...
