Configure Oracle DS via web portal & manual steps

Hoe kan je een Oracle datasource toevoegen via de Widfly admin portal en een paar handelingen met de hand.

1. toevoegen sqldriver aan wildfly:

mkdir /opt/wildfly/modules/system/layers/base/com/oracle/ojdbc/main

container cp /location/oracle/driver/ojdbc7.jar e6e:/ opt/wildfly/modules/system/layers/base/com/oracle/ojdbc/main

Maak een module.xml aan in deze folder:

<!--?xml version="1.0" encoding="UTF-8"?-->

<module xmlns="urn:jboss:module:1.1" name="com.oracle.ojdbc">

  <resources>

    <resource-root path="ojdbc7.jar">

  </resource-root></resources>

  <dependencies>

    <module name="javax.api"></module>

    <module name="javax.transaction.api"></module>

  </dependencies>

</module>

Kopieer de driver zelf met docker cp.

Zet de driver definitie in de standalone.xml onder de datasource drivers.

<driver name="oracle" module="com.oracle.ojdbc">

        <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>

<xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>

    </driver>

Voeg nu de datasource toe via de web interface van Widfly.

<datasource jta="true" jndi-name="java:/EBX\_REPOSITORY" pool-name="EBX\_REPOSITORY" enabled="true" use-ccm="true">

<connection-url>jdbc:oracle:thin:@172.17.0.2:1521:XE</connection-url>

<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>

<driver>oracle</driver>

<security>

<user-name>ebx</user-name>

<password>oracle</password>

</security>

</datasource>

Dit heb ik geautomatiseerd in Dockerfile-minjust. Daar zit ook nog het aanpakken van de ebs home en kopiëren van de ebx.properties in.

Oracle 12cR1 Docker

Je kan er op 3 manieren aankomen:

1. docker store
2. Zelf bouwen: <https://github.com/oracle/docker-images>
3. Dockerhub: <https://hub.docker.com/r/sath89/oracle-12c/>

De eerste heb ik niet geprobeerd. Ik moest gegevens doorgeven en ik vroeg me af of ze me daarna gaan stalken. Dus maar niet gebruikt.

De tweede optie heb ik geprobeerd. Het probleem was dat alles voor Linux is geschreven en ik in de WSL (Windows Subsystem for Linux) op een bepaald moment vastliep.   
  
---> 7f3d5935bdee

Step 11/24 : COPY --chown=oracle:dba $INSTALL\_FILE\_1 $INSTALL\_RSP $INSTALL\_DB\_BINARIES\_FILE $INSTALL\_DIR/

COPY failed: stat /var/lib/docker/tmp/docker-builder597605230/linuxx64\_12201\_database.zip: no such file or directory

Ik heb hier geen oplossing voor gevonden dus ben gestopt.

3 lukt wel. Ik kreeg nog wel de foutmelding:

C:\>docker run -d -p 9090:8080 -p 1521:1521 sath89/oracle-12c

1c173cdab363d8a689ace8a7d0af9da03ab7532c69f69105dfc6c6870f17e6ee

docker: Error response from daemon: driver failed programming external connectivity on endpoint agitated\_einstein (7708d47e27ac52af5397e5f37db41a563f3f1eeec58daffebc7e97eb5206fed8): Error starting userland proxy: mkdir /port/tcp:0.0.0.0:9090:tcp:172.17.0.3:8080: input/output error.

Ik heb gewoon docker herstart en dat lijkt te werken. Er was wel een opmerking op internet dat dit mogelijk niet zou werken als je herstart ofzo. Moet maar zien.

In ieder geval heb ik een orcacle db draaien

Via <http://localhost:9090/em> (sys as sysdba/oracle) inloggen.   
  
Aanmaken van een orale user ebx/oracle met de volgende grants:

1. CREATE SESSION
2. UNLIMITED TABLESPACE
3. CONNECT (is al default)
4. ALTER SESSION
5. CREATE TABLE
6. CREATE SEQUENCE

Gebruikersnamen bij Oracle zijn niet case-sensitief. De uiteindelijke gebruikersnaam wordt EBX.

Wordt nu automatisch aangemaakt met plsql script.

# Add datasource (source documentation) Add data source for EJBCA to use.

Start JBoss and run the following commands in JBoss CLI. Note that "-jindi-name" is linked to the database.properties value (default value in this example).

$ wildfly-12.0.0.Final/bin/jboss-cli.sh --connect

data-source add --name=ejbcads --driver-name="mariadb-java-client.jar" --connection-url="jdbc:mysql://127.0.0.1:3306/ejbca" --jndi-name="java:/EjbcaDS" --use-ccm=true --driver-class="org.mariadb.jdbc.Driver" --user-name="ejbca" --password="ejbca" --validate-on-match=true --background-validation=false --prepared-statements-cache-size=50 --share-prepared-statements=true --min-pool-size=5 --max-pool-size=150 --pool-prefill=true --transaction-isolation=TRANSACTION\_READ\_COMMITTED --check-valid-connection-sql="select 1;"

:reload()

# Add a datasource to WildFly from the standalone.xml file

Behind the scenes the management console is updating the standalone.xml file. If you’re not familiar with WildFly, the standalone.xml is the configuration script for your server. It contains configuration for everything you want to run in your server.

Let’s manually do everything we did in the first segment of this video. So we need to add the JDBC driver first.

Navigate to the /modules/system/layers/base directory in your WildFly directory. Create subfolders /org/postgresql/main The subfolders need to match the package structure of your JDBC driver. So if you’re using an Oracle database, you’d want to create /com/oracle/main. Copy your JDBC driver into the directory you created.

Next you need to create a XML file in the directory called module.xml. The XML file looks like this:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <!--?xml version="1.0" encoding="UTF-8"?-->  <module xmlns="urn:jboss:module:1.1" name="org.postgres">    <resources>      <resource-root path="postgresql-9.4-1201.jdbc4.jar">    </resource-root></resources>    <dependencies>      <module name="javax.api"></module>      <module name="javax.transaction.api"></module>    </dependencies>  </module> |

Make sure the resource-root element path matches the exact name of your JDBC driver.

Next we need to edit the standalone.xml file in the /wildfly-9.0.0.CR2/standalone directory. Look for the drivers element. It should be under the datasources element. We need to add the driver here with the following XML snippet.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <drivers>      <driver name="postgresql" module="org.postgres">          <xa-datasource-class>org.postgresql.Driver</xa-datasource-class>      </driver>  </drivers> |

This adds the driver to WildFly. Now we need to configure the datasource. Just above in the datasources element, add this XML snippet.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <datasource jndi-name="java:jboss/datasources/PostgresDS" pool-name="PostgresDS" enabled="true" use-java-context="true">      <connection-url>jdbc:<postgresql://192.168.99.100:32770/demo1></connection-url>      <driver>postgresql</driver>      <security>          <user-name>postgres</user-name>          <password>password</password>      </security>  </datasource> |

Restart WildFly, and you should be good to go. You can verify your connections in the management console.