Guided Exercise: Configuring JDBC Drivers

In this lab, you will install a MySQL JDBC driver as a module, and activate it in the server configuration.

Resources	
Files:	/usr/share/java/mysql-connector-java.jar /home/ student/ JB248/labs/standalone
App URL:	N/D

Results

You should be able to install a MySQL JDBC driver as a module within the EAP CLI.

before you start

Before beginning the guided exercise, run the following command to verify that EAP was installed at /opt/jboss-eap-7.0 and that no EAP instances are running, and to set up a new base directory for the EAP at / home/student/JB248/labs/standalone:

```
[student@workstation ~]$ lab datasource-driver setup
```

1. Start the EAP CLI.

Before enabling the MySQL JDBC driver, you must create a module for the driver. You can do this in EAP 7 using the CLI, as follows:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ sudo -u jboss ./jboss-cli.sh
```

2. Create the module.

The MySQL JDBC JAR file is located in the /usr/share/java directory of the workstation virtual machine. It is a driver provided by Red Hat's yum repository and was installed during the virtual machine installation by running yum -y install mysql-connector-java as root.

By installing it as a module, it becomes available as a driver in any EAP instance, based on this installation, to connect to MySQL databases.

Use the following command within the EAP CLI to create the module by pointing to the JDBC JAR file, listing the JAR dependencies and the MySQL driver vendor ID as the name:

```
[disconnected /] module add --name=com.mysql \ --resources=/
usr/share/java/mysql-connector-java.jar \ --
dependencies=javax.api,javax.transaction.api
```

Evaluate if the driver was successfully installed as a module by checking if a directory with module.xml and mysql-connector-java.jar was created, using the following command in the terminal window:

```
[student@workstation ~]$ ls /opt/jboss-eap-7.0/modules/com/mysql/main module.xml mysql-
connector-java.jar
```

Also, evaluate whether the module.xml file was generated correctly by comparing it to the following list:

```
<?xml version="1.0" ?>

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

    <resources>
        <resource-root path="mysql-connector-java.jar"/>
    </resources>

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

Open it from JBOSS_HOME/modules/com/mysql/main with your preferred text editor.

3. Define the MySQL driver.

To define the MySQL driver, the EAP CLI must connect to a running instance of EAP.

3.1. Open a new terminal window and start a standalone instance with /home/student/JB248/labs/standalone as the base directory:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh \
-Djboss.server.base.dir=/home/student/JB248/labs/standalone/
```

The server should start without issue with output similar to the following:

```
17:03:30,497 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: JBoss EAP 7.0.0.GA (WildFly Core
  2.1.2.Final-redhat-1) started in 3313ms -Started 261 of 509 services (332 services are lazy, passive or on-
  demand)
```

3.2. Return to the previous terminal that is running the EAP CLI in mode disconnected. Use the following command to connect to the currently running EAP instance:

```
[disconnected /] connect localhost:9990
```

If prompted, provide the username jbossadm and the password JBoss@RedHat123. The request should change to:

```
[standalone@localhost:9990 /]
```

3.3. Use the following command to define the MySQL driver by specifying the EAP module installed in the previous step:

```
[standalone@localhost:9990 /] /
subsystem=datasources\ /
jdbc-driver=mysql:add(driver-name=mysql,driver-module-name=com.mysql)
```

Notice that driver-module-name from com.mysql matches the name given to the module in the previous step. The following output confirms that the driver was installed successfully:

```
{"outcome" => "success"}
```

4. Check the controller.

Use the following command to inspect the new MySQL JDBC driver:

```
[standalone@localhost:9990 /] /subsystem=datasources/jdbc-driver=mysql:read-resource
```

The result should appear as follows:

```
{
    "outcome" => "success",
    "result" => {
        "deployment-name" => undefined, "driver-
        class-name" => undefined, "driver-datasource-
        class-name" => undefined, "driver-major-version" =>
        undefined, "driver-minor-version" => undefined,
        "driver-module-name" => "com.mysql", "driver-
        name" => "mysql", "driver-xa-datasource-class-
        name" => undefined, "jdbc-
        compliant" => undefined, "module-slot" => undefined, "profile"
        => undefined, "xa-datasource-class" =>
        undefined


    }
}
```

5. Perform cleaning.

5.1. Exit the EAP CLI:

```
[standalone@localhost:9990 /] exit
```

5.2. Stop the EAP instance by pressing Ctrl+C in the terminal window that is running EAP.

This concludes the guided exercise.