Guided Exercise: Configuring an XA Data Source

In this lab assignment, you will create and configure a MySQL XA data source, and implement a Java application to test the data source.

Resources	
Files:	/home/student/JB248/labs/standalone /tmp/
App URL:	http://localhost:8080/dstest
Resources	dstest.war

Results

You should be able to create and test an XA data source based on the MySQL driver installed in the previous exercise.

before you start

Before beginning the guided exercise, run the following command to verify that EAP was installed in /opt/jboss-eap-7.0, that no EAP instance is running, to verify that the driver was installed correctly, and to download the application dstest.war:

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

1. Start the standalone EAP instance.

Use the following command to start an EAP instance to access the management console:

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

Before proceeding, wait for the server to finish starting up.

2. Configure the XA JDBC driver. Open a new terminal window and run the following commands to connect to EAP via the CLI:

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

use

The administrator username is jbossadm and the password is JBoss@RedHat123.

In XA, the controller and data source class are different from EAP, and must be passed explicitly as a parameter from the command line. The expected parameters for the data source are:

- ***Name of the data source:*** **bookstorexa**
- ***Nombre de JNDI:*** **java:jboss/datasources/bookstorexa**
- ***Driver name:*** **mysql**
- ***XA data source class:***
  com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
- ***Username:*** **bookstore**
- ***Password:*** **redhat**
- **XA data source parameters:**
  - ÿ ***Server name:*** **localhost**
  - ÿ ***Database name:*** **bookstore**

To create a data source using these parameters, run the following command from the CLI session:

```
[standalone@localhost:9990] xa-data-source add --name=bookstorexa \ --jndi-
name=java:jboss/datasources/bookstorexa --driver-name=mysql \ --xa-datasource-
class=com.mysql.jdbc.jdbc2.optional.MysqlXADataSource \ --user-name=bookstore --
password=redhat \ --xa-datasource-
properties=[{"ServerName"=>"localhost"},\ {"DatabaseName"=>"bookstore"}]
```

3. Test the data source.

3.1. Deploy dstest.war. The dstest.war window.

```
[standalone@localhost:9990] deploy \ /tmp/
dstest.war
```

3.2. Use your browser to go to http://127.0.0.1:8080/dstest/ and access the dtest app.

3.3. • Enter java:jboss/datasources/bookstorexa for the JNDI name.

• Enter bookstore.CatalogItem for the name of the table.

Click List to test the data source.

3.4. Read the results page and verify that the data source search has been correct. You should see the contents of the CatalogItem table in the bookstore database.

4. Perform cleaning.

4.1. Undeploy the dstest.war application:

```
[standalone@localhost:9990] undeploy dstest.war
```

4.2. Exit the EAP CLI:

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

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

This concludes the guided exercise.

Lab Work: Configuring Data Sources

In this lab assignment, you will configure EAP to grant database access through a MySQL database to the bookstore application, instead of using an embedded database.

Resources	
Files	/opt/domain
app url	http://172.25.250.10:8080/bookstore http://172.25.250.11:8080/bookstore
Resources	bookstore.war mysql-connector-java.jar

Results

You must be able to install a MySQL JDBC module and create a data source in a managed domain.

before you start

Use the following command to verify that no EAP instance is running and that the managed domain is configured correctly, based on the previous lab work in the final chapter, and to download the bookstore.war file:

```
[student@workstation ~]$ lab datasources-lab setup
```

1. Add the MySQL module to each host.

In the previous lab work, the bookstore application was not using MySQL. To align with the final solution, the library is required to be backed by a MySQL database. In a managed domain, each host must add the module for the MySQL JDBC driver individually.

1.1. On workstation, servera, and serverb, copy the MySQL module located at /usr/share/java/mysql-connector-java.jar to the /opt/jboss eap-7.0/bin/ directory owned by user jboss. This is done to avoid permissions issues when adding the module as the jboss user.

1.2. Using the EAP CLI without connecting, create a new module on workstation for MySQL JDBC with dependencies javax.api and javax.transaction.api and named com.mysql. In the previous step, the driver was copied to /opt/jboss eap-7.0/bin.

1.3. On servera, use the EAP CLI offline and create a new module for MySQL JDBC with dependencies javax.api and javax.transaction.api, and named com.mysql.

1.4. Repeat the command in serverb to create a new module for MySQL JDBC.

1.5. Evaluate if the driver was successfully installed as a module on workstation, servera and serverb by checking if the directory was created with module.xml and mysql connector-java.jar.

2. Define the MySQL driver.
After installing the MySQL JDBC module, the driver must be defined for use by a data source. Use the following steps to define the MySQL driver.

2.1. Start the domain controller on workstation.

2.2. Start the host controller on servera.

23. Start the host controller on serverb.

2.4. Using the EAP CLI started above, connect to the domain controller at workstation and register the JDBC driver for the full-ha profile. Give the driver a name mysql.

3. Create the Bookstore MySQL data source.
Using the EAP CLI already running on the workstation, create the bookstore MySQL data source for the full-ha profile with the following properties: • Name: bookstore

• Nombre de JNDI: java:jboss/datasources/bookstore

• Driver name: mysql

• Connection URL: jdbc:mysql://172.25.250.254:3306/bookstore

• Username: bookstore

• Password: redhat

4. Test the data source.
Use the administration console to verify that the connection to the database is valid, making sure that the driver was installed correctly and that the bookstore data source was created with the correct values.

5. Deploy the bookstore application.
The bookstore application implemented in the previous labs was not configured to support a MySQL database. This version of bookstore.war has been updated to support MySQL. Use the EAP CLI to deploy /tmp/ bookstore.war to a group of servers Group1.

6. Check the bookstore app.
After you deploy the application to the Group1 server pool, visit the application on each of the servers in the server group.

7. Perform cleaning and grading.
7.1. Press Ctrl+C to stop the EAP CLI, domain, and host controllers.

7.2. Run the following workstation command to grade the assignment:

```
[student@workstation bin]$ lab datasources-lab grade
```

This concludes the lab work.

Solution

In this lab assignment, you will configure EAP to grant database access through a MySQL database to the bookstore application, instead of using an embedded database.

Resources	
Files	/opt/domain
app url	http://172.25.250.10:8080/bookstore http://172.25.250.11:8080/bookstore
Resources	bookstore.war mysql-connector-java.jar

Results

You should be able to install a MySQL JDBC module and create a data source in a managed domain.

Before you begin

Use the following command to verify that no EAP instance is running and that the managed domain is configured correctly, based on the previous lab work in the final chapter, and to download the bookstore.war file:

```
[student@workstation ~]$ lab datasources-lab setup
```

1. Add the MySQL module to each host.

In the previous lab work, the bookstore application was not using MySQL. To align with the final solution, the library is required to be backed by a MySQL database. In a managed domain, each host must add the module for the MySQL JDBC driver individually.

1.1. On workstation, servera, and serverb, copy the MySQL module located at /usr/share/java/mysql-connector-java.jar to the /opt/jboss eap-7.0/bin/ directory owned by user jboss. This is done to avoid permissions issues when adding the module as the jboss user.

```
[student@workstation ~]$ sudo cp /usr/share/java/mysql-connector-java.jar \ /opt/jboss-eap-7.0/bin
```

```
[student@servera ~]$ sudo cp /usr/share/java/mysql-connector-java.jar \ /opt/jboss-eap-7.0/bin
```

```
[student@serverb ~]$ sudo cp /usr/share/java/mysql-connector-java.jar \ /opt/jboss-eap-7.0/bin
```

1.2. Using the EAP CLI without connecting, create a new module on workstation for MySQL JDBC with dependencies javax.api and javax.transaction.api and named com.mysql. In the previous step, the driver was copied to /opt/jboss eap-7.0/bin.

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ sudo -u jboss ./jboss-cli.sh [disconnected /]
module add --name=com.mysql \ --resources=/opt/jboss-
eap-7.0/bin/mysql-connector-java.jar \ --dependencies=javax.api,javax.transaction.api
```

1.3. On servera, use the EAP CLI offline and create a new module for MySQL JDBC with dependencies javax.api and javax.transaction.api, and named com.mysql.

```
[student@servera ~]$ cd /opt/jboss-eap-7.0/bin [student@servera
bin]$ sudo -u jboss ./jboss-cli.sh [disconnected /] module add --
name=com.mysql \ --resources=/opt/jboss-eap-7.0/bin/mysql-
connector-java.jar \ --dependencies=javax.api,javax.transaction.api
```

1.4. Repeat the command in serverb to create a new module for MySQL JDBC.

```
[student@serverb ~]$ cd /opt/jboss-eap-7.0/bin [student@serverb
bin]$ sudo -u jboss ./jboss-cli.sh [disconnected /] module add --
name=com.mysql \ --resources=/opt/jboss-eap-7.0/bin/mysql-
connector-java.jar \ --dependencies=javax.api,javax.transaction.api
```

1.5. Evaluate if the driver was successfully installed as a module on workstation, servera and serverb by checking if the directory was created with module.xml and mysql connector-java.jar.

Use 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
```

2. Define the MySQL driver.

After installing the MySQL JDBC module, the driver must be defined for use by a data source. Use the following steps to define the MySQL driver.

2.1. Start the domain controller on workstation.

```
[student@workstation ~]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \ -Djboss.domain.base.dir=/
opt/domain/ --host-config=host-master.xml
```

2.2. Start the host controller on servera.

```
[student@servera ~]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \ -Djboss.domain.base.dir=/
opt/domain/ \
-Djboss.domain.master.address=172.25.250.254 \ --host-
config=host-slave.xml
```

23. Start the host controller on serverb.

```
[student@serverb bin]$ sudo -u jboss /opt/jboss-eap-7.0/bin/domain.sh \
-Djboss.domain.base.dir=/opt/domain/ \
-Djboss.domain.master.address=172.25.250.254 \ --host-
config=host-slave.xml
```

2.4. Using the EAP CLI started above, connect to the domain controller at workstation and register the JDBC driver for the full-ha profile. Give the driver a name mysql.

```
[disconnected /] connect 172.25.250.254:9990
[domain@172.25.250.254:9990 /] /profile=full-ha/subsystem=datasources\ /jdbc-
driver=mysql:add(driver-name=mysql,driver-module-name=com.mysql)
```

The following output confirms that the driver was installed successfully for each of the four servers:

```
{
    "outcome" => "success", "result"
    => undefined, "server-groups"
    => {
        "Group1" => {"host" => {
            "serve" => {"serve.1" => {"response" => {
                "outcome" => "success", "result"
                => undefined
            }}},
            "serverb" => {"serverb.1" => {"response" => {
                "outcome" => "success", "result"
                => undefined
            }}}
        }},
        "Group2" => {"host" => {
            "serve" => {"serve.2" => {"response" => {
                "outcome" => "success", "result"
                => undefined
            }}},
            "serverb" => {"serverb.2" => {"response" => {
                "outcome" => "success", "result"
                => undefined
            }}}
        }}
    }
}
```

3. Create the Bookstore MySQL data source.

Using the EAP CLI already running on the workstation, create the bookstore MySQL data source for the full-ha profile with the following properties: • Name: bookstore

• Nombre de JNDI: java:jboss/datasources/bookstore

• Driver name: mysql

• Connection URL: jdbc:mysql://172.25.250.254:3306/bookstore

• Username: bookstore

• **Password: redhat**

```
[domain@127.25.250.254:9990 /] data-source add \ --profile=full-
ha --name=bookstore --driver-name=mysql \ --jndi-name=java:jboss/
datasources/bookstore \ --connection-url=jdbc:mysql://
172.25.250.254:3306/bookstore \ --user-name=bookstore --password=redhat
```

4. Test the data source.

Use the administration console to verify that the connection to the database is valid, making sure that the driver was installed correctly and that the bookstore data source was created with the correct values.

4.1. Access the management console by visiting http://172.25.250.254:9990. Use the username jbossadm and the password JBoss@RedHat123.

4.2. At the top of the page, click Settings, then Profiles, and select the full-ha profile.

4.3. Click Subsystems and Data Sources to access the data source subsystem.

4.4. Select Non-XA under data sources, and then select the data source bookstore. Click View next to the data source.

4.5. On the overview page for the bookstore data source, click Connection, and then click Test Connection. A popup window should confirm that the connection is valid. If not, review the steps above, verify that the connection matches the given properties, and that the driver module has been installed correctly.

5. Deploy the bookstore application.

The bookstore application implemented in the previous labs was not configured to support a MySQL database. This version of bookstore.war has been updated to support MySQL. Use the EAP CLI to deploy /tmp/ bookstore.war to a group of servers Group1.

```
[domain@172.25.250.254:9990 /] deploy \ /tmp/
bookstore.war --server-groups=Group1
```

6. Check the bookstore app.

After you deploy the application to the Group1 server pool, visit the application on each of the servers in the server group.

The bookstore application can be accessed on servera.1 at http://172.25.250.10:8080/bookstore and serverb.1 at http://172.25.250.11:8080/bookstore.

7. Perform cleaning and grading.

7.1. Press Ctrl+C to stop the EAP CLI, domain, and host controllers.

7.2. Run the following workstation command to grade the assignment:

```
[student@workstation bin]$ lab datasources-lab grade
```

This concludes the lab work.

Summary

In this chapter, you learned the following:

- **Creating a data source first requires the installation of the JDBC driver.**
  relevant.

- **The administration console provides vendor-specific templates for the database.**
  data to facilitate the creation of the data source.

- **A connection pool avoids manually creating a connection every time one is needed,**
  opting instead to open many connections initially and get one from the "pool" as needed.

- **A JDBC driver can be installed using the EAP CLI module add command.**

- **One of the JDBC drivers can be used for multiple XA and non-XA data sources**
  relevant.

- **Connection pooling can be configured by adjusting the <min-pool size>, <max-pool-**
  size>, and <prefill> values.

- **Connections can be validated as they are removed from the pool (Validate-on**
  match) or with a certain time interval (background-validation).

- **XA data sources offer the ability to roll back partially failed multi-resource transactions.**