Guided Exercise: Configuring the Batch Subsystem

In this lab, you will configure the batch subsystem to support batch job execution.

Resources	
Files:	**/home/student/JB248/labs/batch** **/home/student/JB248/labs/config-batch**
App URL:	**N/A**
Resources	**/home/student/JB248/labs/config-batch/batch job.war**

Results

You must be able to configure the batch subsystem and run batch jobs.

Before you begin

Run the following command to verify that EAP has been installed to /opt/jboss-eap-7.0, that no EAP instances are running, and that you have completed the previous guided exercise, to create the required files, and to download the batch-job.war application.

```
[ student@workstation ~]$ lab config-batch setup
```

1. Start the standalone EAP server instance

In this guided exercise, you will use a standalone instance of EAP with the base directory at /home/student/JB248/labs/batch on the workstation virtual machine.

1.1. To start it, run the following commands from the workstation virtual machine:

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

1.2. Verify that the instance started without errors by looking at the console in which you started the instance.

2. Deploy the batch-job.war application.

The batch-job application defines a batch job named "myjob" with three steps:

1. Read – Simulates reading data from a source location.

2. Process: Simulates the processing of data read from the source location.

3. Write – Simulates writing data after processing to a destination location.

The application is available in the /home/student/JB248/labs/config batch/ folder. Extract the batch-job.war file using the command:

```
[student@workstation ~]$ unzip \ /home/
student/JB248/labs/config-batch/batch-job.war
```

The job is defined in the WEB-INF/classes/META-INF/batch-jobs/myjob.xml file inside the batch-job.war file. Each step is described as an XML tag named step in the XML file. The batchlet refers to a bean name defined using the CDI.

Job execution will be triggered using the EAP CLI.

2.1. Open a new terminal window and run the EAP CLI tool to connect to the standalone instance with the following commands:

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

2.2. Deploy the batch-job.war application.

```
[standalone@127.0.0.1:9990 /] deploy \ /home/
student/JB248/labs/config-batch/batch-job.war
```

23. In the console, check if the application was successfully deployed. You should see a message similar to the following:

```
12:16:27,321 INFO [org.jboss.as.server] (management-handler-thread - 5)
  WFLYSRV0010: Deployed "batch-job.war" (runtime-name : "batch-job.war")
```

3. Start the batch job.

By default, EAP 7 is configured with an in-memory work pool and uses a thread-pool with 10 threads. An in-memory job repository stores job metadata in RAM, and this information is lost when the server is shut down or restarted.

3.1. The batch-job application defines a job called "myjob". start this job using the EAP CLI:

```
[standalone@127.0.0.1:9990 /] /deployment=batch-job.war/subsystem=\ batch-jberet:start-
job(job-xml-name=myjob)
```

Repeat the above command several times, run the job 2-3 times, and watch the console output.

3.2. You should see the following output in the job execution console:

```
[stdout] (Batch Thread - 1) ReadBatchlet: Reading Data... [stdout] (Batch Thread
- 1) ProcessBatchlet: Processing Data... [stdout] (Batch Thread - 1) WriteBatchlet:
Writing Data...

[stdout] (Batch Thread - 2) ReadBatchlet: Reading Data... [stdout] (Batch Thread
- 2) ProcessBatchlet: Processing Data...
```

```
[stdout] (Batch Thread - 2) WriteBatchlet: Writing Data...
```

4. **View the statistics about the execution of the job.**

   Statistics about each executed job are stored in the default in-memory repository, and EAP manages the execution of each batch job using a set of ten threads. View job statistics using the EAP CLI.

   4.1. **To view the statistics, run the following command:**

```
[standalone@127.0.0.1:9990 /] /deployment=batch-job.war/\ subsystem=batch-
jberet:read-resource(recursive=true,include-runtime=true)
```

   4.2. **The result appears as follows:**

```
"myjob" =>
            { "instance-count" => 3, "running-
                  executions" => 0, "execution" =>
                  { "4" => { "batch-
                        status" =>
                              "COMPLETED", "create- time" =>
                              "2016-06-08T11:01:08.358+0530", "end-time" =>
                              "2016-06-08T11:01:08.363+0530", "exit-status" =>
                              "COMPLETED", "instance-id" => 4L,
                              "last-updated-time" =>
                              "2016-06-08T11:01:08.363+0530", "start-time" => "2016-06-08T11:01:08.359+
                              0530"
                        },
...
```

   4.3. **View the thread-pool statistics of the batch subsystem using the following command (simulate multiple batch runs by running the command to start the job from Step 3.1 multiple times in rapid succession):**

```
[standalone@127.0.0.1:9990 /] /subsystem=batch-jberet/\ thread-
pool=batch:read-resource(include-runtime=true,recursive=true)
```

   4.4. **The result appears as follows:**

```
{
      "outcome" => "success",
      "result" =>
            { "active-count" => 0,
            "completed-task-count" => 14L, "current-
            thread-count" => 10, "keepalive-time "
            => { "time" => 30L, "unit" =>
                  "SECONDS"

            },
            "largest-thread-count" => 10, "max-
            threads" => 10, "name" =>
            "batch", "queue-size"
            => 0, "rejected-count"
            => 0, " task-count" => 14L,
            "thread-factory" =>
            undefined
      }
}
```

The active-count attribute indicates the number of jobs that are currently running.

The completed-task-count attribute indicates the number of jobs that were executed successfully.

The current-thread-count attribute indicates the number of threads that are currently active.

5. Stop the EAP instance by pressing Ctrl+C in the terminal window you started in the instance.

   Close the EAP CLI sessions by typing quit or press Ctrl+C in the terminal window in which you started the CLI sessions.

   This concludes the guided exercise.

Lab Work: Configuring the Batch Subsystem

In this lab job, you will configure and run a batch job in a managed domain.

Resources	
Files	/opt/domain /tmp/bookstore-job.war
app url	http://172.25.250.10:8080/bookstore

Result

You should be able to configure and run a batch job on the servers in the managed domain. It will also persist and view job execution statistics against a MySQL database.

before you start

Use the following command to download the relevant lab files, create the required lab folders, verify that the bookstore application is deployed, and ensure that the managed domain is configured correctly:

```
[student@workstation ~]$ lab batching-lab-final setup
```

You can use the EAP 7 management console or the JBoss EAP CLI to achieve your goals, keeping in mind that the EAP CLI is the preferred option in production environments.

An EAP administrator has configured a managed domain with two host controllers running the servera and serverb virtual machines, respectively, and the domain controller on the workstation. The domain and host configuration files are stored in the / opt/domain folder on all three machines. It will start the managed domain and deploy the bookstore-job application, which defines a batch job that counts the number of pending orders in the bookstore database. You will start the batch job using the EAP CLI and you will need to configure the batch subsystem to store batch job execution details in a separate MySQL database named bkjobs.

1. **Start the domain controller in the workstation virtual machine. Because domain controller configuration files are kept in the /opt/domain folder on workstation, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also note that the host file for the domain controller is named host-master.xml and is located in the /opt/domain/ configuration folder. (Tip: Pass the --host-config=host-master.xml argument to domain.sh.)**

   Note that the /opt/domain directory is owned by the jboss user, so you must start the domain controller using sudo -u jboss /opt/jboss-eap-7.0/ bin/domain.sh ...

2. The two host controllers on servera and serverb connect to the host controller domain and get the latest configuration of the domain. Start the two host controllers on servera and serverb.

   2.1. Start the host controller on servera. Because the host controller configuration files are kept in the /opt/domain folder on servera, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also notice that the host file for the host controller is named host-slave.xml and is located in the /opt/domain/configuration folder. (Hint: Pass the --host config=host-slave.xml argument to domain.sh.)

   Note that the /opt/domain directory is owned by the jboss user, so start the host controller using sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...

   2.2. Start the host controller on serverb. Because the host controller configuration files are kept in the /opt/domain folder on serverb, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also notice that the host file for the host controller is named host-slave.xml and is located in the /opt/domain/configuration folder. (Hint: Pass the --host config=host-slave.xml argument to domain.sh.)

   Note that the /opt/domain directory is owned by the jboss user, so start the host controller using sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...

   23. Verify that both host controllers connect to the domain controller and form a managed domain. Look at the console window in which you started the domain controller and verify that both servera and serverb are registered as slaves to the domain controller.

3. Run the EAP CLI and connect to the domain controller to configure the servers in the managed domain.

4. Verify that the bookstore application has been deployed to the group of Group1 servers.

   4.1. Stop the servers in the Group2 server group, as they are not being used in this lab work.

   4.2. Verify that bookstore.war has been deployed to the server group Group1.

   4.3. If the bookstore app has not been deployed, deploy bookstore.war in the Group1 server group. It is available in /tmp/bookstore.war on the workstation virtual machine.

   4.4. Look at the console window on servera and servera and verify that both server instances, servera.1 and serverb.1 have been successfully deployed to the bookstore application.

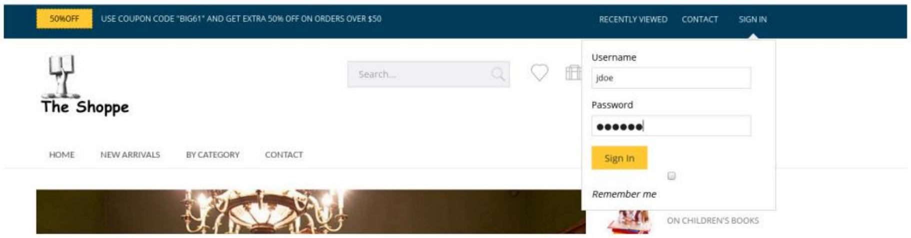
5. Deploy the bookstore-job application.

5.1. Deploy bookstore-job.war to the Group1 server group. This available in /tmp/bookstore-job.war on the workstation virtual machine.

5.2. Look at the console window on the servera and serverb virtual machines where you started the host controllers and verify that both server instances, servera.1 and serverb.1, have been successfully deployed to the bookstore-job application.

6. Place orders using the bookstore app.

6.1. Navigate to http://172.25.250.10:8080/bookstore to access the bookstore application. Click the SUBSCRIBE link in the upper right corner and log in with username jdoe and password redhat.



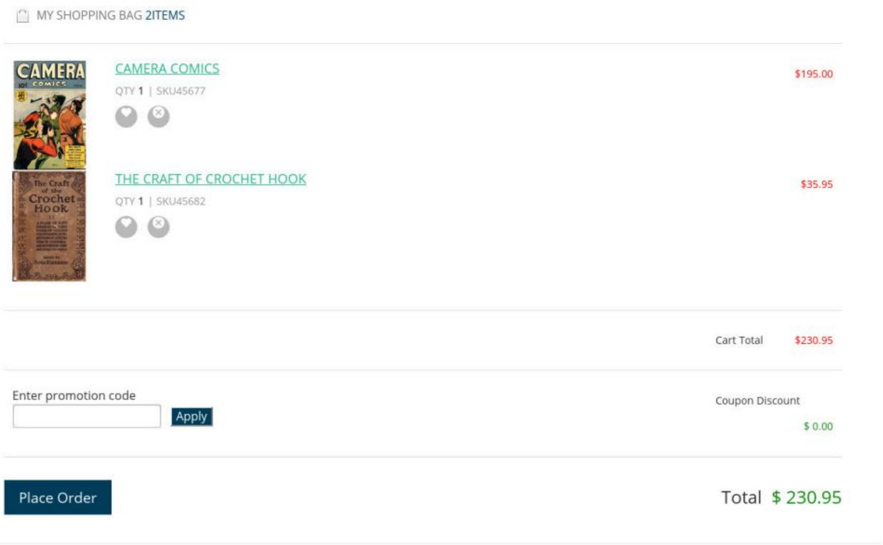
Username
jdoe

Password
●●●●●●

Bookstore login

6.2. Browse the different categories of books available in the store and add some books to the shopping cart by clicking BUY on the particular book description page.

6.3. After you've added a few books to your shopping cart, click Place Order on the Shopping Cart page.



MY SHOPPING BAG 2ITEMS

CAMERA COMICS $195.00
QTY 1 | SKU45677

THE CRAFT OF CROCHET HOOK $35.95
QTY 1 | SKU45682

Cart Total $230.95

Enter promotion code

Coupon Discount $ 0.00

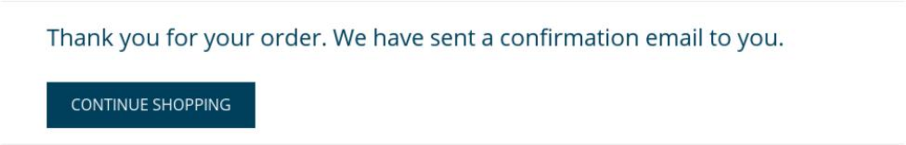
Total $ 230.95

bookstore shopping cart

6.4. On the Billing Information page, enter your address information in the form and select the Ship to same address check box in the Shipping Information column. To continue, click NEXT.

6.5. On the Payment Information page, select Payment Type, enter a Credit Card Number, Expiration Date, and Cardholder Name; then click REVIEW to continue.

6.6. On the Review page, review the information you entered and click SUBMIT to place the order. If your order was accepted, you should see the following message:



Thank you for your order. We have sent a confirmation email to you.

Successful order message in bookstore

7. Configure the batch subsystem.

7.1. Create a pool of threads for the batch jobs called bk-jobs-pool with a max-threads value of 20 and a keepalive-time value of 20 seconds.

7.2. Create a new JDBC job repository to store job execution statistics. The configure script created a data source named bkjob-ds, which points to a MySQL database named bkjobs. Create a new jdbc-job-repository named bookstore-job repository and reference the bkjob-ds data source.

7.3. Connect to the bkjobs database using the MySQL client and verify that the tables for storing job execution data have been created with the CLI command above (for the MySQL client, use username=bookstore and password= redhat).

7.4. Since you haven't run any jobs, there should be no data in these tables. (Use SQL select commands to check.)

7.5. By default, the batch subsystem uses an in-memory job repository and references a set of threads named batch. Change the default job repository to the newly defined bookstore-job repository and the default thread pool to bk-jobs-pool.

7.6. To change the default thread set and job repository, the host controllers will need to be reloaded. Reload hosts servera and serverb.

8. Start the batch job defined in the bookstore-job application.

8.1. Start the library batch job using the EAP CLI. You must specify the host and server on which you want the batch job to run. Run the batch job on servera.1. The name of the job defined in the bookstore-job application is myjob.

8.2. Observe the servera console window. You should see the following output, indicating that there is a pending order in the bookstore database:

```
INFO [com.redhat.training.batch.job.ReportBatchelet] (Batch Thread - 2)
  Processing job...
 ...
```

```
INFO [com.redhat.training.batch.job.ReportBatchelet] (Batch Thread - 2) There
  are 1 Orders in the DB
```

8.3. Repeat Steps 6.2-6.6, place some more commands in the library, run the batch job again and verify that the correct order count is displayed as a result of the batch job.

9. See the statistics about the work of the bookstore.

9.1. Use the EAP CLI to view statistics about batch jobs.

9.2. Connect to the bkjobs database using the MySQL client, as described in Step 7.3, and verify that the results of the select queries from Step 7.4 show statistics for the jobs you ran. Run a few more jobs and verify that the output of the SELECT queries matches the data returned in the EAP CLI commands.

10. Perform cleaning and grading.

10.1.Press Ctrl+C in the terminal window in which you started the drivers host and domain controller to stop the managed domain. (Alternatively, you can shut down the domain controller using the JBoss EAP CLI command / host=master:shutdown()).

10.2.Press Ctrl+C to exit the EAP CLI if you used the CLI in the lab. (Alternatively, you can leave the CLI by typing exit.)

10.3 Exit the MySQL client session by typing quit or \q.

10.4.Run the following command from workstation to grade the assignment:

```
[student@workstation ~]$ lab batching-lab-final grade
```

This concludes the lab work.

Solution

In this lab job, you will configure and run a batch job in a managed domain.

Resources	
Files	/opt/domain /tmp/bookstore-job.war
app url	http://172.25.250.10:8080/bookstore

Result

You should be able to configure and run a batch job on the servers in the managed domain. It will also persist and view job execution statistics against a MySQL database.

before you start

Use the following command to download the relevant lab files, create the required lab folders, verify that the bookstore application is deployed, and ensure that the managed domain is configured correctly:

```
[student@workstation ~]$ lab batching-lab-final setup
```

You can use the EAP 7 management console or the JBoss EAP CLI to achieve your goals, keeping in mind that the EAP CLI is the preferred option in production environments.

An EAP administrator has configured a managed domain with two host controllers running the servera and serverb virtual machines, respectively, and the domain controller on the workstation. The domain and host configuration files are stored in the / opt/domain folder on all three machines. It will start the managed domain and deploy the bookstore-job application, which defines a batch job that counts the number of pending orders in the bookstore database. You will start the batch job using the EAP CLI and you will need to configure the batch subsystem to store batch job execution details in a separate MySQL database named bkjobs.

1. **Start the domain controller in the workstation virtual machine. Because domain controller configuration files are kept in the /opt/domain folder on workstation, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also note that the host file for the domain controller is named host-master.xml and is located in the /opt/domain/ configuration folder. (Tip: Pass the --host-config=host-master.xml argument to domain.sh.)**

   Note that the /opt/domain directory is owned by the jboss user, so you must start the domain controller using sudo -u jboss /opt/jboss-eap-7.0/ bin/domain.sh ...

   ```
   [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. The two host controllers on servera and serverb connect to the host controller domain and get the latest configuration of the domain. Start the two host controllers on servera and serverb.

2.1. Start the host controller on servera. Because the host controller configuration files are kept in the /opt/domain folder on servera, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also notice that the host file for the host controller is named host-slave.xml and is located in the /opt/domain/configuration folder. (Hint: Pass the --host config=host-slave.xml argument to domain.sh.)

Note that the /opt/domain directory is owned by the jboss user, so start the host controller using sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...

Open a new terminal window on the server virtual machine and run the following command:

```
[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
```

2.2. Start the host controller on serverb. Because the host controller configuration files are kept in the /opt/domain folder on serverb, use /opt/domain as the value of the jboss.domain.base.dir argument that you pass to the domain.sh startup script. Also notice that the host file for the host controller is named host-slave.xml and is located in the /opt/domain/configuration folder. (Hint: Pass the --host config=host-slave.xml argument to domain.sh.)

Note that the /opt/domain directory is owned by the jboss user, so start the host controller using sudo -u jboss /opt/jboss eap-7.0/bin/domain.sh ...

Open a new terminal window on the serverb virtual machine and run the following command:

```
[student@serverb ~]$ 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. Verify that both host controllers connect to the domain controller and form a managed domain. Look at the console window in which you started the domain controller and verify that both servera and serverb are registered as slaves to the domain controller.

3. Run the EAP CLI and connect to the domain controller to configure the servers in the managed domain.

In a new terminal window on workstation, start the EAP CLI and connect to the domain controller as the jboss user:

```
[student@workstation ~]$ sudo -u jboss /opt/jboss-eap-7.0/bin/jboss-cli.sh \ --connect --
controller=172.25.250.254:9990
```

4. Verify that the bookstore application has been deployed to the group of Group1 servers.

4.1. Stop the servers in the Group2 server group, as they are not being used in this lab work.

```
[domain@172.25.250.254:9990 /] /server-group=Group2\ :stop-
servers(blocking=true)
```

4.2. Verify that bookstore.war has been deployed to the server group Group1.

```
[domain@172.25.250.254:9990 /] deployment-info --server-group=Group1
```

```
NAME                    RUNTIME-NAME STATE
bookstore.war bookstore.war enabled cluster.war
cluster.war enabled
```

4.3. If the bookstore app has not been deployed, deploy bookstore.war in the Group1 server group. It is available in /tmp/bookstore.war on the workstation virtual machine.

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

4.4. Look at the console window on servera and servera and verify that both server instances, servera.1 and serverb.1 have been successfully deployed to the bookstore application.

5. Deploy the bookstore-job application.

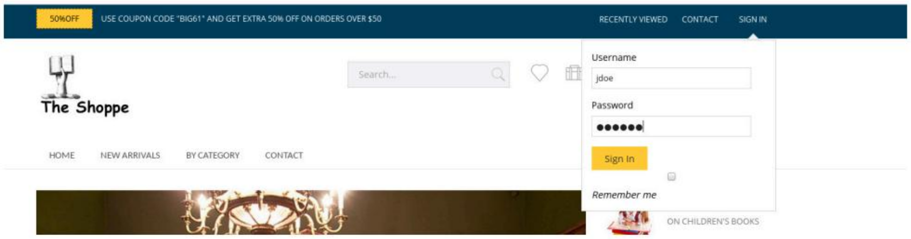
5.1. Deploy bookstore-job.war to the Group1 server group. This available in /tmp/bookstore-job.war on the workstation virtual machine.

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

5.2. Look at the console window on the servera and serverb virtual machines where you started the host controllers and verify that both server instances, servera.1 and serverb.1, have been successfully deployed to the bookstore-job application.

6. Place orders using the bookstore app.

6.1. Navigate to http://172.25.250.10:8080/bookstore to access the bookstore application. Click the SUBSCRIBE link in the upper right corner and log in with username jdoe and password redhat.



Username

jdoe

Password

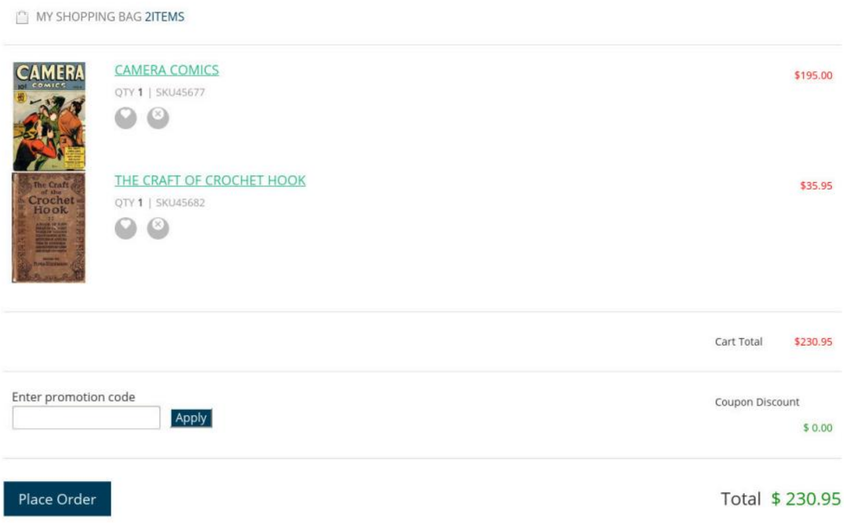
●●●●●●

□ Remember me

Bookstore login

6.2. Browse the different categories of books available in the store and add some books to the shopping cart by clicking BUY on the particular book description page.

6.3. After you've added a few books to your shopping cart, click Place Order on the Shopping Cart page.



MY SHOPPING BAG 2ITEMS

CAMERA COMICS $195.00

QTY 1 | SKU45677

THE CRAFT OF CROCHET HOOK $35.95

QTY 1 | SKU45682

Cart Total $230.95

Enter promotion code

Coupon Discount $ 0.00

Total $ 230.95

bookstore shopping cart

6.4. On the Billing Information page, enter your address information in the form and select the Ship to same address check box in the Shipping Information column. To continue, click NEXT.

6.5. On the Payment Information page, select Payment Type, enter a Credit Card Number, Expiration Date, and Cardholder Name; then click REVIEW to continue.

6.6. On the Review page, review the information you entered and click SUBMIT to place the order. If your order was accepted, you should see the following message:

Thank you for your order. We have sent a confirmation email to you.

Successful order message in bookstore

7. Configure the batch subsystem.

7.1. Create a pool of threads for the batch jobs called bk-jobs-pool with a max-threads value of 20 and a keepalive-time value of 20 seconds.

```
[domain@172.25.250.254:9990 /] /profile=full-ha/subsystem=\ batch-jberet/
thread-pool=bk-jobs-pool:\ add(max-
threads=20,keepalive-time={unit= SECONDS,time=20})
```

7.2. Create a new JDBC job repository to store job execution statistics. The configure script created a data source named bkjob-ds, which points to a MySQL database named bkjobs. Create a new jdbc-job-repository named bookstore-job repository and reference the bkjob-ds data source.

```
[domain@172.25.250.254:9990 /] /profile=full-ha/subsystem=\ batch-jberet/jdbc-
job-repository=\ bookstore-jobs-
repository:add(data-source="bkjob-ds")
```

7.3. Connect to the bkjobs database using the MySQL client and verify that the tables for storing job execution data have been created with the CLI command above (for the MySQL client, use username=bookstore and password= redhat).

```
[student@workstation tmp]$ mysql -u bookstore -p
Enter password: ******

MariaDB [(none)]> use bkjobs;
Database changed

MariaDB [(bkjobs)]> show tables;

+--------------------+
 | tables_in_bkjobs                    |
+--------------------+
 | JOB_EXECUTION |
 | JOB_INSTANCE |
 | PARTITION_EXECUTION |
 | STEP_EXECUTION |
+--------------------+
4 rows in set (0.00 sec)
```

7.4. Since you haven't run any jobs, there should be no data in these tables. (Use SQL select commands to check.)

```
MariaDB [(bkjobs)]> select * from JOB_EXECUTION; Empty set
(0.00 sec)

MariaDB [(bkjobs)]> select * from JOB_INSTANCE; Empty set
(0.00 sec)
```

7.5. By default, the batch subsystem uses an in-memory job repository and references a set of threads named batch. Change the default job repository to the newly defined bookstore-job repository and the default thread pool to bk-jobs-pool.

```
 [domain@172.25.250.254:9990 /] /profile=full-ha/subsystem=\ batch-
jberet:write-attribute\ (name=default-
 job-repository,value=bookstore-jobs-repository)

 [domain@172.25.250.254:9990 /] /profile=full-ha/subsystem=\ batch-
jberet:write-attribute\ (name=default-
 thread-pool,value=bk-jobs-pool)
```

7.6. To change the default thread set and job repository, the host controllers will need to be reloaded. Reload hosts servera and serverb.

```
 [domain@172.25.250.254:9990 /] reload --host=server
 [domain@172.25.250.254:9990 /] reload --host=serverb
```

8. Start the batch job defined in the bookstore-job application.

8.1. Start the library batch job using the EAP CLI. You must specify the host and server on which you want the batch job to run. Run the batch job on servera.1. The name of the job defined in the bookstore-job application is myjob.

```
 [domain@172.25.250.254:9990 /] /host=servera/server=servera.1\ /
deployment=bookstore-job.war/subsystem=\ batch-
jberet:start-job(job-xml-name=myjob) {

        "outcome" => "success",
        "result" => 3L
}
```

8.2. Observe the servera console window. You should see the following output, indicating that there is a pending order in the bookstore database:

```
INFO [com.redhat.training.batch.job.ReportBatchelet] (Batch Thread - 2)
  Processing job...
 ...
INFO [com.redhat.training.batch.job.ReportBatchelet] (Batch Thread - 2) There
  are 1 Orders in the DB
```

8.3. Repeat Steps 6.2-6.6, place some more commands in the library, run the batch job again and verify that the correct order count is displayed as a result of the batch job.

9. See the statistics about the work of the bookstore.

9.1. Use the EAP CLI to view statistics about batch jobs.

```
 [domain@172.25.250.254:9990 /] /host=servera/server=servera.1\ /
deployment=bookstore-job.war/subsystem=\ batch-
jberet/job=myjob:\ read-
resource(include-runtime= true,recursive=true) {

        "outcome" => "success", "result"
        => { "instance-
                count" => 3, "running-
                executions" => 0, "execution" =>
                { "3" => { "batch-
                        status" =>
                                "COMPLETED", "create-time" =>
                                "2016-06-09T05:36:07.399-0400", "end-time" =>
                                "2016-06-09T05:36:09.419-0400", "exit -status" =>
                                "COMPLETED", "instance-id" => 3L,
                                "last-updated-time" =>
                                "2016-06-09T05:36:09.419-0400", "start-time" => "2016
                                -06-09T05:36:07.400-0400"
 ...
        }
}
```

9.2. Connect to the bkjobs database using the MySQL client, as described in Step 7.3, and verify that the results of the select queries from Step 7.4 show statistics for the jobs you ran. Run a few more jobs and verify that the output of the SELECT queries matches the data returned in the EAP CLI commands.

10. Perform cleaning and grading.

10.1.Press Ctrl+C in the terminal window in which you started the drivers host and domain controller to stop the managed domain. (Alternatively, you can shut down the domain controller using the JBoss EAP CLI command / host=master:shutdown()).

10.2.Press Ctrl+C to exit the EAP CLI if you used the CLI in the lab. (Alternatively, you can leave the CLI by typing exit.)

10.3 Exit the MySQL client session by typing quit or \q.

10.4.Run the following command from workstation to grade the assignment:

```
[student@workstation ~]$ lab batching-lab-final grade
```

This concludes the lab work.

Summary

In this chapter, you learned the following:

- EAP 7 has introduced a batch subsystem and API that can be used to execute batch jobs that do not require interactive input.

- A batch job is defined in an XML file in the application's JAR or WAR file, in the META-INF/batch-jobs folder. A batch job is made up of a number of discrete steps.

- The EAP batch subsystem configuration consists of working repositories and wire sets.

- A thread pool consists of a pool of server threads that can be reused to run multiple batch jobs.

- A job repository stores the execution statistics of batch jobs. There are two main types of batch repositories:

  ÿ in-memory—Stores details of job executions in memory (RAM) and details are lost when the server is rebooted or shut down.

  ÿ JDBC—Stores job execution details persistently in a database. The details are available even if the server is rebooted or shut down.

- EAP comes pre-configured with an in-memory working repository and a set of 10 threads.

- The EAP administration console can be used to configure job repositories and thread pools in the batch subsystem.

- The EAP CLI can be used to configure job repositories and job sets. threads in the batch subsystem, as well as to start, stop, restart, and view statistics for batch jobs.

- In an EAP managed domain, you must inform the server on which to run the work when you start it using the EAP CLI.