Workshop: Define a Destination

Workshop

Follow along with the instructor as you define a couple of new destinations.

- ☐ 1. Define a New Queue
  - ☐ 1.1. Start by changing directories in the CLI to the default hornetq-server level of the full profile:

```
cd /profile=full/subsystem=messaging/hornetq-server=default
```

Insight

The **messaging** subsystem comes preconfigured with one provider named **default**, which is the HornetQ messaging provider.

  - ☐ 1.2. Add a new queue named **devQueue**:

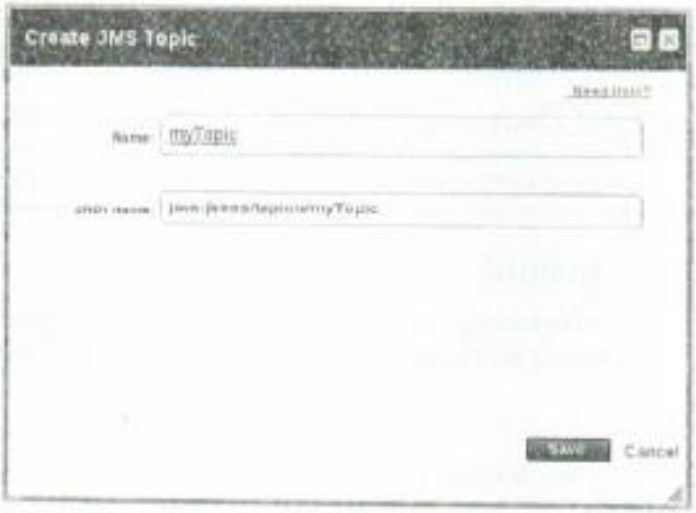
```
./jms-queue=devQueue:add(entries=["java:jboss/jms/devQueue"])
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => undefined
}
```

  - ☐ 1.3. Verify the queue is defined properly by invoking **read-resource** on it:

```
./jms-queue=devQueue:read-resource
{
    "outcome" => "success",
    "result" => {
        "durable" => true,
        "entries" => ["java:jboss/jms/devQueue"],
        "selector" => undefined
    }
}
```

- ☐ 2. Define a Topic
  - ☐ 2.1. Go to the **Profiles** page of the Management Console.
  - ☐ 2.2. Switch to the **full** profile.
  - ☐ 2.3. Click on the **Destinations** page under the **Messaging** section.
  - ☐ 2.4. Click the **View** link next to the **default JMS Messaging Provider**. The **Queues/Topics** page should be displayed.

- ☐ 2.5. The **Queues/Topics** page has two tabs: **Queues** and **Topics**. Verify your **devQueue** appears in the list of queues on the **Queues** tab.
- ☐ 2.6. Click on the **Topics** tab.
- ☐ 2.7. Click the **Add** button to add a new topic.
- ☐ 2.8. Enter **myTopic** for the **Name** and **java:jboss/topic/myTopic** for the **JNDI Name**.



- ☐ 2.9. Click the **Save** button. Verify your new topic now appears in the list of **Topics**.
- ☐ 2.10. You can also verify the new topic by entering the following command in the CLI:

```
./jms-topic=myTopic:read-resource
{
    "outcome" => "success",
    "result" => {"entries" => ["java:jboss/topics/myTopic"]}
}
```

Lab 08_01: Defining Topics & Queues

Performance Checklist

Lab Overview:

In this exercise, you will define a JMS topic and queue.

Lab Resources/Configuration:

Lab Files Location:	*LABS/Lab08_01*
Application URL:	`http://192.168.0.xx:8680/JMSTest/run`

Success Criteria: The JMSTest application will successfully send 5 messages to your topic and queue.

Outcome: A new JMS topic and queue defined in your **`full-ha`** profile.

Lab Outline:

1. View the Connectors
2. View the Acceptors
3. Define a Topic
4. Define a Queue
5. Deploy a JMS Application
6. Test the Topic and Queue
7. View the Message Journal Files

☐ 1. View the Connectors

☐ 1.1. Enter the following CLI command:

```
/profile=full-ha/subsystem=messaging:read-resource(recursive=true)
```

☐ 1.2. What is the value of the **`cluster-password`** attribute?

_____

☐ 1.3. What is the file size of the HornetQ persistence journal? _____

☐ 1.4. Notice the **`journal-type`** is **`ASYNCIO`**. What is the other possible value for **`journal-type`**? _____

☐ 1.5. How many remote connectors are configured?

_____

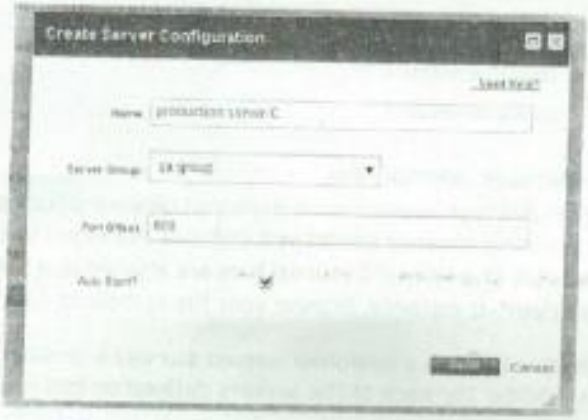
☐ 1.6. If a client uses the connection factory whose JNDI name is **`"java:/ConnectionFactory"`**, what type of connector would be retrieved?

_____

- ☐ 1.7. What JNDI name would a client lookup for accessing the **`RemoteConnectionFactory`**?

  _____

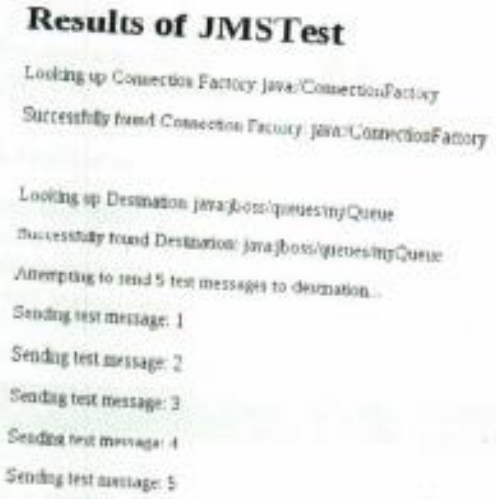
- ☐ 2. View the Acceptors
  - ☐ 2.1. In the Management Console, go the **Profiles** page and switch to the **`full-ha`** profile.
  - ☐ 2.2. Click on the **Connections** link in the **Messaging** subsystem, then click the **View** link of the **`default`** JMS messaging provider.
  - ☐ 2.3. You should now be viewing the **Acceptors** page. How many remote acceptors are configured? _____
  - ☐ 2.4. Change the **Type:** dropdown from **Type: Remote** to **Type: In-VM**. How many **In-VM** acceptors are defined? _____
  - ☐ 2.5. Click on the **Connector** page. Verify there is a connector configured that matches each of the remote and invm acceptors.
- ☐ 3. Define a Topic
  - ☐ 3.1. Using either the CLI or the Management Console, define a new topic in the **`full-ha`** profile named **`simpleTopic`** with a JNDI name of **`java:jboss/topics/simpleTopic`**.
- ☐ 4. Define a Queue
  - ☐ 4.1. Using either the CLI or the Management Console, define a new, non-durable queue in the **`full-ha`** profile named **`myQueue`** with a JNDI name of **`java:jboss/queues/myQueue`**.
- ☐ 5. Deploy a JMS Application
  - ☐ 5.1. You are going to deploy an application that tests your destinations, but you do not have a Server defined yet that uses the **`full-ha`** profile. Go to the **Server** page of the Management Console and select **`host3`**. Add a new Server named **`production-server-C`** that is a member of the **`qa-group`** Server Group. Use a port offset of **`600`** and configure it to auto start.



- ☐ 5.2. Go the **Runtime** page of the Management Console and start **production-server-C**.
- ☐ 5.3. Deploy the application **LABS/Lab08_01/JMSTest.war** onto your **qa-group**.

☐ 6. Test the Topic and Queue

- ☐ 6.1. Point your browser to `http://192.168.0.xx:8680/JMSTest/run`.
- ☐ 6.2. Enter **java:/ConnectionFactory** for the JNDI name of the Connection Factory.
- ☐ 6.3. Enter **java:jboss/topics/simpleTopic** for the JNDI name of the Destination.
- ☐ 6.4. Enter 5 for the number of test messages to send.
- ☐ 6.5. Click the **Submit** button.
- ☐ 6.6. You should see the "Results of JMSTest" page. Read through the results of the test and verify that your JNDI lookups succeeded, and also verify the test messages were delivered successfully.
- ☐ 6.7. Similarly, enter **java:jboss/queues/myQueue** for the JNDI name and verify your queue is also defined and working.



☐ 7. View the Message Journal Files

- ☐ 7.1. The JMSTest application is deployed on your **production-server-C**, which is running on your **machine3** instance managed by the Host Controller named **host3**. The HornetQ journal files are located in a subfolder of the **production-server-C** instance. Browse your file system to your **machine3/domain** folder.
- ☐ 7.2. Notice there is a subfolder named **servers**. Inside the **servers** folder is a subfolder for each of the servers defined on this host.
- ☐ 7.3. Look inside the **production-server-C/data** folder. Notice there are three folders that store messaging information:

1. **messagingbindings**
2. **messagingjournal**
3. **messaginglargemessages**

☐ 7.4. Look inside the **messagingjournal** folder. The **hornetq-data-#.hq** files are the journal files used by HornetQ of this server. There are currently two of these files, because your value of **journal-min-files** is 2.

Insight

The journal files are binary files, so you can't view their contents in a human-readable format. The **server.lock** files are used by HornetQ to synchronize reads and writes from the journal files.