Guided Exercise: Configuring Load Balancing

In this lab assignment, you will load balance HTTP requests across a predetermined set of servers using Undertow.

Resources	
Files:	/home/student/JB248/labs/standalone /home/student/JB248/labs/config-lb
App URL:	http://172.25.250.254:8080/cluster http://172.25.250.254:8180/cluster http://172.25.250.254:8280/cluster
Resources	/home/student/JB248/labs/config-lb/cluster.war

Results

You should be able to configure a load balancer that will be responsible for balancing requests among the available hosts in a cluster.

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 cluster.war application.

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

1. Start the standalone EAP servers.

In this guided exercise, three separate instances are required running the standalone-ha.xml configuration. The first will be the load balancer and the others will be part of a cluster.

1.1. The first standalone server will be used as a load balancer with the following features:

• Use the /home/student/JB248/labs/standalone folder as the base directory.

• Use the standalone-ha.xml configuration file.

• Define the public interface IP as 172.25.250.254.

To configure it, run the following commands from the workstation virtual machine:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh --server-config=standalone-ha.xml \ -Djboss.server.base.dir=/
home/student/JB248/labs/standalone-instance \  -Djboss.bind.address=172.25.250.254
```

use

The configuration file is not required to create a load balancer. It is used here to avoid problems with other lab work that uses the default.

1.2. The second independent server will be part of the cluster and must have the following features:

- **Use la carpeta /home/student/JB248/labs/config-lb/server1 como base directory.**
- **Use the standalone-ha.xml configuration file.**
- **Define the public interface IP as 172.25.250.254.**
- **To avoid port conflicts, use a value of 100 as the port offset. ports.**
- **Define the node name as server1.**

To configure it, run the following commands from the workstation virtual machine:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh --server-config=standalone-ha.xml \ -Djboss.server.base.dir=/
home/student/JB248/labs/config-lb/server1 \  -Djboss.bind.address=172.25.250.254
-Djboss.socket.binding.port-offset=100 \ -Djboss.node.name=server1
```

use

The node name is required to use the persistent session feature.

1.3. The third standalone server will be part of the cluster and must have the following features:

- **Use la carpeta /home/student/JB248/labs/config-lb/server2 como base directory.**
- **Use the standalone-ha.xml configuration file.**
- **Define the public interface IP as 172.25.250.254.**
- **To avoid port conflicts, use a value of 200 as the port offset. ports.**
- **Define the node name as server2.**

To configure it, run the following commands from the workstation virtual machine:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh --server-config=standalone-ha.xml \ -Djboss.server.base.dir=/
home/student/JB248/labs/config-lb/server2 \  -Djboss.bind.address=172.25.250.254
-Djboss.socket.binding.port-offset=200 \ -Djboss.node.name=server2
```

2. Deploy the cluster.war application.

The cluster application must be deployed on the two separate servers in the cluster. The application must be deployed twice, once per host.

2.1. Open a new terminal window and run the CLI tool to connect to the first instance of the cluster with the following commands:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./jboss-cli.sh -c --controller=127.0.0.1:10090
```

2.2. Deploy the cluster application:

```
[standalone@127.0.0.1:10090 /] deploy \ /home/
student/JB248/labs/config-lb/cluster.war
```

23. Connect to the second instance of the cluster:

```
[standalone@127.0.0.1:10090 /] connect 127.0.0.1:10190
```

Deploy the cluster application:

```
[standalone@127.0.0.1:10190 /] deploy \ /home/
student/JB248/labs/config-lb/cluster.war
```

3. Test the cluster.

Open a web browser and point to http://172.25.250.254:8180/cluster. You should see the cluster application. Notice that it has information about the instance that responded to the request as the base directory.

Please refresh the page a few times to increase the number of hits, then point the browser to http://172.25.250.254:8280/cluster. Notice that the total hits increased by one, even though you are running the application from a different server, which means that the application is clustered.

4. Configure the load balancer.

Although the cluster is configured, each server must be accessed individually. Set up a new load balancer using the first EAP instance (the one without port offset).

4.1. Return to the CLI tool and connect to the load balancer instance:

```
[standalone@127.0.0.1:10190 /] connect 127.0.0.1:9990
```

4.2. To configure a new load balancer, the Undertow subsystem must support a reverse proxy that will load balance the request between two servers. Create a reverse proxy named cluster-handler:

```
[standalone@127.0.0.1:9090 /] /subsystem=undertow/configuration=\ handler/reverse-
proxy=cluster-handler:add
```

4.3. Create a new socket binding to be used by the AJP protocol running on the server named server1 (172.25.250.254:8109). Name it remote-server1:

```
[standalone@127.0.0.1:9090 /] /socket-binding-group=standard-sockets\ /remote-
destination-outbound-socket-binding=remote-server1\ :add(host=172.25.250.254,
port=8109)
```

note

The standalone server named server1 was started with a port offset of 100, so the AJP port will be 8109.

4.4. Create in Undertow a reference to the socket binding, created earlier to server1, and bind it to the AJP schema; this way you should access the cluster context path:

```
[standalone@127.0.0.1:9090 /] /subsystem=undertow/configuration=handler\ /reverse-
proxy=cluster-handler/host=server1\ :add(outbound-
socket-binding=remote-server1, scheme=ajp, \ instance-id=server1, path=/
cluster)
```

Important

To enable persistent session, the instance-id attribute must have the same value specified by the jboss.node.name system property.

4.5. Create a new socket binding that will use the AJP protocol running on the server named server2 (172.25.250.254:8209). Name it remote server2:

```
[standalone@127.0.0.1:9090 /] /socket-binding-group=standard-sockets\ /remote-
destination-outbound-socket-binding=remote-server2\ :add(host=172.25.250.254,
port=8209)
```

4.6. Create in Undertow a reference to the socket binding, created to server2, and bind it to the AJP schema to access the cluster context path:

```
[standalone@127.0.0.1:9090 /] /subsystem=undertow/configuration=handler\ /reverse-
proxy=cluster-handler/host=server2\ :add(outbound-
 socket-binding=remote-server2, scheme=ajp,\ instance-id=server2, path=/
cluster)
```

4.7. Create a new reverse proxy location named /cluster, which refers to the cluster-handler:

```
[standalone@127.0.0.1:9090 /] /subsystem=undertow/server=default-server/\ host=default-
host\ /location=\/
cluster:add(handler=cluster-handler)
```

5. Test the load balancer.

5.1. Open a web browser and point to http://172.25.250.254:8080/cluster. You should see the cluster application. Refresh the browser a few times and notice that each request is processed by the same server. Determine which EAP is handling your current request by looking at The JBoss EAP node name is tag in the application.

5.2. Stop the host responding to requests by pressing Ctrl+C in the corresponding terminal window.

5.3. Return to the web browser and refresh the page. The load balancer forwards your request to the other server without losing the current value of visits.

6. Perform cleaning.

6.1. Exit the EAP CLI:

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

6.2. Stop the EAP instances by pressing Ctrl+C in the terminal window that you are running EAP.

This concludes the guided exercise.