Guided Exercise: Configuring JGroups for a Clustered Web Application

In this lab assignment, you will configure the JGroups UDP and TCP stacks to enable clustering of web applications.

Resources	
Files:	/home/student/JB248/labs/jgroups-cluster1 /home/student/JB248/labs/jgroups-cluster2 /home/student/JB248/labs/config-jgroups
App URL:	http://172.25.250.254:8080/cluster http://172.25.250.254:8180/cluster
Resources	/home/student/JB248/labs/config-jgroups/cluster.war /home/student/JB248/labs/config-jgroups/new tcp-stack.cli

Results

You should be able to configure and test a two-node cluster of EAP instances using the default UDP stack, as well as a custom TCP stack.

before you start

Run the following command to verify that EAP was installed to /opt/jboss-eap-7.0 and that no EAP instances are running, to create the required files, and to download the cluster.war application.

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

1. You need to open multiple ports on the workstation virtual machine for this lab work. Briefly review the firewall configuration in the /home/student/JB248/labs/config-jgroups/jgroups-firewall-rules.sh file. Run the script as follows:

```
[student@workstation standalone]$ sudo sh \
        /home/student/JB248/labs/config-jgroups/jgroups-firewall-rules.sh
```

2. Verify that the ports on the firewall are open by running the following command:

```
[student@workstation standalone]$ firewall-cmd --list-all --zone=public public (default) interfaces:

  sources:
  services: dhcpv6-client http ldap mountd mysql nfs rpc-bind ssh 54300/tcp 57600/tcp
  55200/udp 54200/tcp 55300/udp 23364/udp 8080/tcp 8180/tcp
45700/udp 7600/tcp 57700/tcp 45688/udp 7700/tcp
```

```
...
```

3. Start the standalone EAP servers.

In this guided exercise, two separate instances are required running the standalone-full-ha.xml configuration. The standalone full-ha.xml configuration file defines an interface named private in the <interfaces> section, which is used for inter-cluster communication.

use

In a production environment, the public and private interfaces will be bound to two physically separate networks with different IP addresses. In this lab work, we are binding both interfaces to the IP of the workstation virtual machine (172.25.250.254).

3.1. The first standalone server will start with the following features:

- **Use the directory /home/student/JB248/labs/jgroups-cluster1 as base directory.**
- **Use the standalone-full-ha.xml configuration file.**
- **Define the IP of the public interface as 172.25.250.254.**
- **Define the private interface IP as 172.25.250.254.**
- **Set the node name to jgroups-cluster1.**

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 \ --server-
    config=standalone-full-ha.xml \
-Djboss.server.base.dir=/home/student/JB248/labs/jgroups-cluster1 \
-Djboss.bind.address=172.25.250.254 \
-Djboss.bind.address.private=172.25.250.254 \
-Djboss.node.name=jgroups-cluster1
```

Verify that the first instance started without errors by reviewing the /home/student/JB248/labs/jgroups-cluster1/log/server.log file, using the tail -f command.

use

In the standalone-full-ha.xml file, there is no console handler defined by default, hence the need to use the tail command to view the server logs.

3.2. The second standalone server will start with the following features:

- **Use the directory /home/student/JB248/labs/jgroups-cluster2 as base directory.**
- **Use the standalone-full-ha.xml configuration file.**
- **Define the IP of the public interface as 172.25.250.254.**
- **Define the private interface IP as 172.25.250.254.**
- **To avoid port conflicts, define the port offset with a value out of 100.**
- **Set the node name to jgroups-cluster2.**

To start the server, 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-full-ha.xml \
-Djboss.server.base.dir=/home/student/JB248/labs/jgroups-cluster2 \
-Djboss.bind.address=172.25.250.254 \
-Djboss.bind.address.private=172.25.250.254 \
-Djboss.socket.binding.port-offset=100 \
-Djboss.node.name=jgroups-cluster2
```

Verify that the second instance started without errors by reviewing the /home/student/JB248/labs/jgroups-cluster2/log/server.log file, using the tail -f command.

3.3. Observe the log file for jgroups-cluster1 after the jgroups-cluster2 instance starts successfully. You should see the following entries showing that there are now two members in the cluster:

```
2016-05-25 03:08:57,593 INFO
  [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (thread-1,ee,jgroups-
  cluster1) ISPN000094: Received new cluster view for channel server: [jgroups-cluster1|1] (2) [jgroups-
  cluster1, jgroups-cluster2]
2016-05-25 03:08:57,596 INFO
  [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (thread-1,ee,jgroups-
  cluster1) ISPN000094: Received new cluster view for channel web: [jgroups-cluster1|1] (2) [jgroups-
  cluster1, jgroups-cluster2]
 ...
```

use

You may see some WARN messages, such as the following, in the server logs:

```
2016-05-30 13:13:19,132 WARN
  [org.infinispan.topology.ClusterTopologyManagerImpl] (transport  thread--p13-t2)
ISPN000197: Error updating cluster member list: org.infinispan.util.concurrent.TimeoutException:
  Replication timeout for jgroups-cluster2
```

This is a recurring issue in the EAP 7 GA release (see https://issues.jboss.org/browse/JBEAP-794) and can be safely ignored. It will be fixed in a later release of EAP 7.

4. Test the clustering using the configured UDP stack so default in the standalone-full-ha.xml file. Use the test application cluster.war, which is available in the /home/student/JB248/labs/ config-jgroups folder.

5. Verify that the two EAP instances in the cluster are using the UDP protocol for communication with the tcpdump tool to monitor traffic on the workstation virtual machine. Communication takes place at the multicast address 230.0.0.4 and port 45688 by default.

Open a new terminal window and run the following command:

```
[student@workstation ~]$ sudo tcpdump -i eth0 udp port 45688 -vvv
```

You should see output similar to the following:

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes 13:15:31.727165 IP (tos 0x8,
ttl 2, id 37698, offset 0, flags [DF], proto UDP (17), length 133) workstation.lab.example.com.55200 > 230.0.0.4.45688:
  [bad udp
       cksum 0x8d9f ->
  0x7397!] UDP, length 105
13:15:32.532451 IP (tos 0x8, ttl 2, id 37699, offset 0, flags [DF], proto UDP (17), length 68)

  workstation.lab.example.com.55200 > 230.0.0.4.45688: [bad udp cksum 0x8d5e -> 0x91d3!] UDP, length 40
  13:15:32.533185 IP (tos 0x8, ttl
2, id 37700, offset 0, flags [DF], proto UDP (17), length 68) workstation.lab.example.com.55300 > 230.0.0.4.45688:
  [bad udp
       cksum 0x8d5e -> 0xdcf2!] UDP, length 40 13:15:32.644452 IP (tos 0x8, ttl 2, id 37701, offset 0, flags [DF],
  proto UDP (17), length 113)
workstation.lab.example.com.55200 > 230.0.0.4.45688: [bad udp cksum 0x8d8b -> 0xdd43!] UDP, length 85
```

6. 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 node.

6.1. 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=127.0.0.1:9990
```

6.2. Deploy the cluster application:

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

6.3. Connect to the second instance of the cluster:

```
[standalone@127.0.0.1:9990 /] connect 127.0.0.1:10090
```

Deploy the cluster application:

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

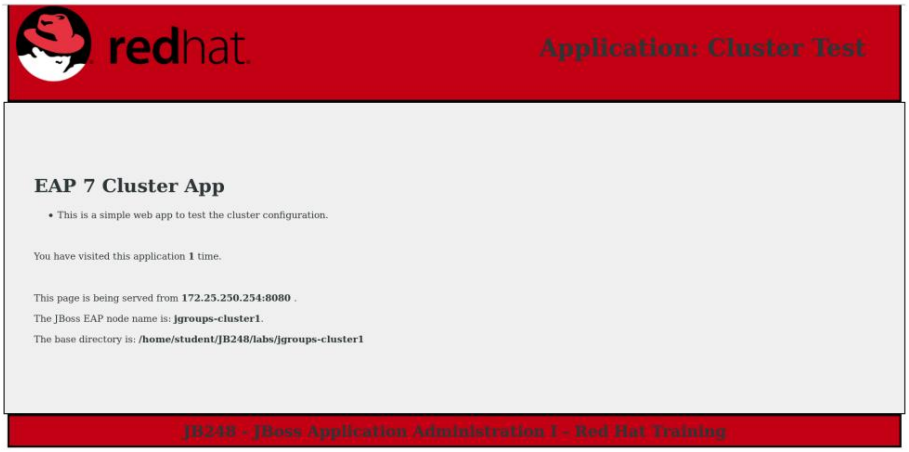
6.4. In the log files of both instances, verify that the application was successfully deployed. You should see a message similar to the following:

```
2016-05-25 03:41:24,749 INFO [org.jboss.as.server] (management-handler-thread - 5) WFLYSRV0010: Deployed
  "cluster.war" (runtime-name : "cluster.war")
```

7. Test the cluster using the UDP stack.

By default, EAP is configured for clustered storage, using the UDP stack. In this mode, EAP nodes can automatically join and leave the cluster without manual configuration, and nodes discover each other using UDP multicast.

7.1. Navigate to http://172.25.250.254:8080/cluster to see the test application running on the jgroups-cluster1 instance. Refresh the page a few times and notice that the counter increases by one each time you refresh the page.



Navigate to http://172.25.250.254:8180/cluster to see the test application running on the jgroups-cluster2 instance. Notice that the counter value was not reset, but increased by one. Refresh the page a few times and notice that the counter increases by one each time you refresh the page.

7.2. Shut down the jgroups-cluster2 instance by pressing Ctrl+C in the command window. terminal on which you started the instance. Navigate to http://172.25.250.254:8080/ cluster to see the test application running on the jgroups-cluster1 instance.

Notice that the counter value has not been reset, but reflects the last value (plus one), as noted before shutting down jgroups-cluster2.

Refresh the page a few times and notice that the counter increases by one each time you refresh the page. The EAP clustering component uses the JGroups stack (UDP by default) to replicate the counter value to all nodes in the cluster.

7.3. Shut down the jgroups-cluster1 instance by pressing Ctrl+C in the terminal window in which you started the instance.

8. Test the cluster using the TCP stack.

In many data center networks, UDP multicast is disabled, and you must use TCP to cluster EAP nodes. In contrast to the UDP stack, where EAP nodes can be added to a cluster automatically, in the TCP stack configuration, the IP addresses of cluster nodes are manually managed by the EAP administrator.

8.1. Use the EAP CLI to define a new TCP stack configuration. Open the /home/student/JB248/ labs/config-jgroups/new-tcp-stack.cli file in a text editor and review the commands. The content of the file should look like this:

```
# Add a new TCP stack called "tcpping" batch

/subsystem="jgroups"/stack="tcpping":add()
```

```
/subsystem="jgroups"/stack="tcpping":add-protocol(type="TCPPING") /subsystem="jgroups"/
stack="tcpping"/transport="TRANSPORT":add(socket  binding="jgroups-tcp",type="TCP") run-
batch


# Customize the protocol settings for tcpping batch /

subsystem="jgroups"/stack="tcpping"/protocol="TCPPING"/
property="initial_hosts":add(value="node1[port1],node2[port2]") /subsystem="jgroups"/
stack="tcpping"/protocol="TCPPING"/ property="port_range":add(value="10") /
subsystem="jgroups"/stack="tcpping":add-
protocol(type="MERGE2") /subsystem="jgroups"/stack="tcpping":add-protocol(socket-
binding="jgroups-tcp  fd",type="FD_SOCK") /subsystem="jgroups"/stack="tcpping":add-protocol(type="FD") /
subsystem="jgroups"/
stack="tcpping":add-protocol(type="VERIFY_SUSPECT") /subsystem="jgroups"/
stack="tcpping":add-protocol(type="BARRIER") /subsystem="jgroups"/stack="tcpping":add-
protocol(type="pbcast.NAKACK") /subsystem="jgroups"/stack="tcpping":add-
protocol(type="UNICAST2") /subsystem="jgroups"/stack="tcpping":add-protocol(type="pbcast.STABLE") /
subsystem="jgroups"/stack="tcpping":add-protocol(type="pbcast.GMS") /subsystem="jgroups"/
stack="tcpping":add-protocol(type="UFC") /subsystem="jgroups"/stack="tcpping":add-
protocol(type="MFC") /subsystem="jgroups"/stack="tcpping":add-protocol(type="FRAG2") /
subsystem="jgroups"/stack="tcpping":add-protocol(type="RSVP") /subsystem=jgroups/
channel=ee:write-attribute(name=stack,value=tcpping) run-batch :reload
```

The initial_hosts property is a comma-separated list of server instances and the ports (in brackets) that will be part of the cluster.

8.2. Edit the initial_hosts property and replace the values with the addresses IP and ports of the two EAP instances as follows (remember that the jgroups-cluster2 instance is running with a port-offset value of 100):

```
..."initial_hosts":add(value="172.25.250.254[7600],172.25.250.254[7700]")
```

8.3. Start both instances of EAP again as described in steps 3.1 and 3.2.

8.4. Run the following command in a new terminal on the virtual machine workstation to create the new TCP stack on the jgroups-cluster1 instance:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./jboss-cli.sh --connect \
   --controller=localhost:9990 \
   --file=/home/student/JB248/labs/config-jgroups/new-tcp-stack.cli
The batch executed successfully
...
```

8.5. Similarly, run the following command in a new terminal on the workstation virtual machine to create the new TCP stack on the jgroups cluster2 instance:

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

```
[student@workstation bin]$ ./jboss-cli.sh --connect \
   --controller=localhost:10090 \
   --file=/home/student/JB248/labs/config-jgroups/new-tcp-stack.cli
The batch executed successfully
...
```

8.6. Verify that the new TCP stack named tcpping is defined in the /home/student/JB248/labs/jgroups-clusterX/configuration/standalone full-ha.xml file of both instances, where 'X' denotes the instance number:

```
<stack name="tcpping">
    <transport type="TCP" socket-binding="jgroups-tcp"/>
        <protocol type="TCPPING">
            <property name="initial_hosts">
                172.25.250.254[7600],172.25.250.254[7700] </property>
            <property
            name="port_range"> 10

            </property> </
        protocol>
        <protocol type="MERGE2"/>
                        ...
</stack>
```

Also check that the default stack has been set to the new tcpping definition:

```
<channels default="ee">
    <channel name="ee" stack="tcpping"/> </channels>
```

8.7. Repeat steps 7.1 and 7.2 to test the new TCP stack. Verify that the test application behaves similar to how it did when you configured the instances with the default UDP stack.

8.8. Observe the output of the tcpdump command as described in Step 5 and verify that you are NOT seeing UDP traffic on port 45688 because the nodes are using the TCP protocol for communication. Observe the TCP traffic on localhost port 7600 when both instances are running using the tcpdump command:

```
[student@workstation ~]$ sudo tcpdump -i lo tcp port 7600 -vvv
```

You should see output similar to the following:

```
tcpdump: listening on lo, link-type EN10MB (Ethernet), capture size 65535 bytes 13:42:27.894727 IP (tos 0x0,
ttl 64, id 56200, offset 0, flags [DF], proto TCP (6), length 4225) workstation.lab.example.com.7600 >

 workstation.lab.example.com.54083: Flags [P.], cksum 0x5ea4 (incorrect -> 0xe452), seq
 2980730036:2980734209, ack 3719238263, win 10, options [nop,nop,TS val 34228000 ecr 34227223], length
 4173 13:42:27.894782 IP (tos 0x0, ttl 64, id 27899, offset 0, flags [DF], proto TCP

 (6), length 52)
```

```
    workstation.lab.example.com.54083 > workstation.lab.example.com.7600: Flags
[.], cksum 0x4e57 (incorrect -> 0x1e8b), seq 1, ack 4173, win 32742, options [nop,nop,TS val 34228000
ecr 34228000], length 0
```

9. Stop the two EAP instances by pressing Ctrl+C in the terminal window in which you started the instances.

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

Exit the tail and tcpdump command sessions by pressing Ctrl+C in the terminal window in which you started the commands.

This concludes the guided exercise.