HACEP: Highly Available and Horizontally Scalable Complex Event Processing

Presenters:

Ugo Landini, Sanne Grinovero, Andrea Tarocchi, Andrea Leoncini

Abstract:

Recognizing important patterns in data, as they occur in real time, is critical. With complex event processing (CEP), organizations can recognize, understand, and react to business events faster by processing large volumes of inbound data. This includes data from social networks, remote sensor readings, online-gaming player events, credit card transactions, and location data from delivery vehicles. Critical CEP workloads require solutions that tolerate the failure of one or more processing nodes, while easily scaling to handle growing data volumes. They must process a high volume, velocity, and variety of inbound data, infer the context, and take appropriate action in real time while adhering to stringent availability and scalability service level agreements. In this lab, you'll implement a simple rule-based, event-based application on a cluster of distributed nodes. You'll see your events and rules scale up and down thanks to HACEP, which is the combination of Red Hat JBoss BRMS superpowered with Red Hat JBoss Data Grid.

Audience/Intro/Prerequisites:

This lab is geared towards Systems Administrators and Software Developers who might be interested in learning how to manage HACEP and write rules for it. HACEP is the combination of Red Hat JBoss BRMS and Red Hat JBoss Data Grid.

Attendees, during this session, will learn:

- What is Red Hat JBoss BRMS
- What is Red Hat JBoss Data Grid
- How HACEP combines BRMS and Data Grid
- HACEP Architecture
- How to manage an HACEP Cluster consisting of 4 EAP nodes
- How to write BRMS rules for HACEP
- HACEP internals

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

Document Conventions

Code and command line output will be placed into a block similar to the following:

```
This is a block! We can do all sorts of cool code and command line stuff here!

Look, more lines!

$ echo "Here's some command line output!"
Here's some command line output!
```

Topics of interest or vocabulary terms will either be referred to as URLs to their respective documentation/website, as a **bold** item, or in *italics*. The first encounter of the term should be a reference to its respective documentation.
Command line utilities, commands, or things otherwise found in code that are used throughout paragraphs will be written in a monospace font.

Notes are marked as **Note:** and any files that are displayed in their entirety are marked as **File Listing:** FILENAME

Prerequisites

No special background is needed to attend this session.
A basic knowledge of Rules Management Systems and Grid concepts will be helpful.

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

Overview

Oftentimes, when working with records of significant change of state in the application domain at a given point in time, known as *events*, it becomes necessary to identify and analyze patterns in such a way as to deduce higher-level impacts, time-based correlations, and relationships that can factor into or directly affect business processes, application code, decision-making, and the like. As a quick example, consider the gaming industry, where the actions performed by a player or set of players must be processed quickly and synchronously in order to best process their intent. The financial sector also lends itself to a simplified use case, wherein credit card transactions over a given period of time can be analyzed in order to assist in fraud detection and prevention. This process of detecting and selecting interesting events from within an event cloud, finding their relationships, and inferring new data is known as **Complex Event Processing**, or **CEP**.

When dealing with large amounts of events in a synchronous way, or even when considering time correlations amongst them, it's important to recognize high-availability as a requirement for an enterprise application related to CEP. Without consistency and redundancy, significant events can be overlooked or missed entirely, thus creating the possibility of exclusions or faulty assumptions concluded from processing an *event cloud*, or group of events.

Likewise, when considering the sheer volume of events possible within various application spaces, which still must be consumed in a consistent and reliable fashion, scalability becomes a factor. While early on, it's possible to simply increase hardware capacity for short-term gains, eventually, this model becomes unfeasible and thus, partitioning of capabilities and processing becomes a necessity.

HACEP

HACEP (pronounced *hä-sep*) is a scalable and highly-available architecture framework for enterprise- level usage of the complex event processing abilities in **Red Hat JBoss BRMS** (Business Rules Management System).

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

LAB 1: INSTALLATION

You will find all the needed material already pre-downloaded on your system. Part of your first task is to complete the installation and setup of Hacep. Estimated time (20 minutes)

A brief tour of the file system

Most of the software is installed under the `/opt` directory.
Check that your `/opt` directory is similar to:

```
[root@hacep ~]# ll /opt/
total 16
drwxr-xr-x.  7 root root  141 Apr 12 15:16 ./
dr-xr-xr-x. 23 root root 4096 Mar 23 11:16 ../
lrwxrwxrwx.  1 root root   27 Apr 12 14:59 amq ->
drwxr-xr-x.  6 root root   99 Mar 22 12:18 apache-maven-3.3.9/
lrwxrwxrwx.  1 root root    5 Apr 12 14:58 jboss ->
drwxr-xr-x. 12 root root 4096 Apr 12 15:17 jboss-eap-7.0_CONFIGURED/
drwxr-xr-x.  5 root root  147 Mar 22 11:54 repos/

[root@hacep ~]# ll /opt/repos/
total 0
drwxr-xr-x. 5 root root 147 Mar 22 11:54 ./
drwxr-xr-x. 7 root root 141 Apr 12 15:16 ../
drwxrwxr-x. 3 root root  75 Nov 21 02:16 jboss-brms-bpmsuite-6.4.0.GA-maven-repository/
drwxr-xr-x. 3 root root  96 Jul 12  2016 jboss-datagrid-7.0.0-maven-repository/
drwxrwxr-x. 3 root root  96 Apr 19  2016 jboss-eap-7.0.0.GA-maven-repository/
```

You will find:

- A preconfigured instance of Red Hat JBoss AMQ 6.3.0 (`jboss-a-mq-6.3.0.redhat-187_CONFIGURED`)
- A preconfigured instance of EAP 7 with HACEP (`jboss-eap-7.0_CONFIGURED`)

- A fresh Maven 3.3.9 installation

- A repos directory that contains the full maven repositories for Red Hat JBoss BRMS 6.4.0, Red Hat JBoss Data Grid 7.0.0 and Red Hat JBoss EAP 7.0.0
- 2 broken links (`/opt/jboss` and `/opt/amq`)

```
[root@hacep]# ll /opt
total 8
lrwxrwxrwx.  1 root root   27 Apr 12 14:59 amq ->
drwxr-xr-x.  6 root root   99 Mar 22 12:18 apache-maven-3.3.9
lrwxrwxrwx.  1 root root    5 Apr 12 14:58 jboss ->
drwxr-xr-x. 15 root root 4096 Apr 13 10:51 jboss-a-mq-6.3.0.redhat-187_CONFIGURED
drwxr-xr-x. 12 root root 4096 Apr 12 15:17 jboss-eap-7.0_CONFIGURED
drwxr-xr-x.  5 root root  147 Mar 22 11:54 repos
```

The home directory contains HACEP source code and a setenv.sh script that sets all the needed environment vars. You won't need to use this script unless you want to skip the HACEP installation and configuration.

```
[root@hacep ~]# ll ~
total 20
-rw-------.  1 root root 1492 Dec  1 19:14 anaconda-ks.cfg
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Desktop
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Documents
drwxr-xr-x.  2 root root 4096 Apr 12 15:29 downloads
drwxr-xr-x. 11 root root  272 Apr  5 09:53 hacep
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Music
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Pictures
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Public
-rw-r--r--.  1 root root  281 Apr 12 15:02 setenv.sh
-rw-r--r--.  1 root root 7698 Mar 30 15:03 settingsEAP7-merge.xml
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Templates
drwxr-xr-x.  2 root root    6 Apr  5 09:18 Videos
drwxr-xr-x.  5 root root   74 Apr  5 12:10 workspace
```

The setenv exports all important environment variables. These same env vars are set in the `.bashrc`.

```
[root@hacep ~]# cat setenv.sh
export JAVA_HOME=/usr/lib/jvm/java-1.8.0
export JAVA_OPTS=-Djava.security.egd=file:/dev/./urandom
export JBOSS_HOME=/opt/jboss
export AMQ_HOME=/opt/amq
export M2_HOME=/opt/apache-maven-3.3.9
export HACEP_REPO=/opt/repos
export PATH=$PATH:$M2_HOME/bin:$JBOSS_HOME/bin:$AMQ_HOME/bin
```

The hacep directory contains all the HACEP modules. You will need it to add new rules to the HACEP system.

```
[root@hacep ~]# ll ~/hacep
total 44
drwxr-xr-x. 2 root root    32 Mar 31 14:55 docs
drwxr-xr-x. 2 root root    50 Mar 31 14:55 example-maven-settings
drwxr-xr-x. 5 root root    97 Apr  5 09:53 hacep-commons-test
drwxr-xr-x. 5 root root    97 Apr  5 09:53 hacep-core
drwxr-xr-x. 5 root root    97 Apr  5 09:53 hacep-core-camel
drwxr-xr-x. 5 root root    97 Apr  5 09:53 hacep-core-model
drwxr-xr-x. 9 root root   214 Apr  5 09:53 hacep-examples
-rw-r--r--. 1 root root 11325 Mar 31 14:55 LICENSE
-rw-r--r--. 1 root root 22080 Mar 31 14:55 pom.xml
-rw-r--r--. 1 root root  6739 Mar 31 14:55 README.md
```

Task 1.1: Install EAP

Open another tab on your terminal, or another terminal window.

NOTE: if you want to skip EAP installation, you can just copy the preconfigured EAP and go directly to Task 1.3:

```
[root@hacep /]# cd /opt
[root@hacep opt]# cp -r jboss-eap-7.0_CONFIGURED jboss-eap-7.0
[root@hacep opt]# domain.sh
```

Open another tab on your terminal, or another terminal window.

Unzip Red Hat JBoss EAP 7.0.0 in the /opt directory, add an admin user and start the domain mode:

```
[root@hacep ~]# cd
[root@hacep ~]# unzip downloads/jboss-eap-7.0.0.zip -d /opt
```

```
[root@hacep ~]# add-user.sh -u admin -p admin
[root@hacep ~]# domain.sh
```

Now you should be able to access to the EAP management console at `http://localhost:9990/` to check that everything is ok

Task 1.2: Build HACEP

Open another tab on your terminal, or another terminal window.

To install HACEP, you must first build it from source. Launch the following commands from the hacep source directory to build everything (and also run the full suite of unit tests):

```
[root@hacep ~]# cd ~/hacep
[root@hacep ~]# mvn -s ../settingsEAP7-merge.xml -Psupported-GA-ee7 -o clean install
```

This is the expected output:

```
… lots of maven stuff which you ignore, though sometimes there are important things :)
[INFO] --- maven-install-plugin:2.4:install (default-install) @ hacep-eap-playground ---
[INFO]                                                                       Installing
/root/hacep/hacep-examples/hacep-eap-playground/target/hacep-eap-playground-1.0-SNAPSHOT.
war                                                                                  to
/root/.m2/repository/it/redhat/jdg/examples/hacep-eap-playground/1.0-SNAPSHOT/hacep-eap-p
layground-1.0-SNAPSHOT.war
[INFO]      Installing     /root/hacep/hacep-examples/hacep-eap-playground/pom.xml      to
/root/.m2/repository/it/redhat/jdg/examples/hacep-eap-playground/1.0-SNAPSHOT/hacep-eap-p
layground-1.0-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] HACEP Parent ...................................... SUCCESS [  0.308 s]
[INFO] HACEP Commons Test ................................ SUCCESS [  1.899 s]
[INFO] HACEP Core Model .................................. SUCCESS [  0.185 s]
[INFO] HACEP Core ........................................ SUCCESS [ 13.509 s]
[INFO] HACEP Core Camel .................................. SUCCESS [  3.344 s]
[INFO] HACEP Examples .................................... SUCCESS [  0.006 s]
[INFO] HACEP Example Model ............................... SUCCESS [  0.113 s]
[INFO] HACEP Example Rules ............................... SUCCESS [  2.762 s]
[INFO] HACEP Performance Client .......................... SUCCESS [  0.588 s]
```

```
[INFO] HACEP Base Playground .............................. SUCCESS [  0.418 s]
[INFO] HACEP JVM Playground ............................... SUCCESS [ 13.389 s]
[INFO] HACEP EAP Playground ............................... SUCCESS [  1.815 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 39.883 s
[INFO] Finished at: 2017-04-12T16:29:11+02:00
[INFO] Final Memory: 109M/1091M
[INFO] ------------------------------------------------------------------------
```

Check that every module has a success status.
You will now have a new HACEP war (`hacep-eap-playground-1.0-SNAPSHOT.war`) in the mvn directory. `/root/.m2/repository/it/redhat/jdg/examples/hacep-eap-playground/1.0-SNAPSHOT`.

Now copy the HACEP war in the `/opt/jboss` directory

```
[root@hacep ~]# cp
~/.m2/repository/it/redhat/jdg/examples/hacep-eap-playground/1.0-SNAPSHOT/hacep-eap-playg
round-1.0-SNAPSHOT.war /opt/jboss
```

Task 1.3: Install AMQ

Open another tab on your terminal, or another terminal window

NOTE: if you want to skip AMQ installation, you can just make a link to the preconfigured AMQ:

```
[root@hacep /]# cd /opt
[root@hacep opt]# cp -r jboss-a-mq-6.3.0.redhat-187_CONFIGURED
jboss-a-mq-6.3.0.redhat-187
[root@hacep opt]# amq
```

If you skipped EAP installation and go directly to Task 1.6

Unzip Red Hat JBoss AMQ 6.3.0 in the /opt directory:

```
[root@hacep ~]# unzip downloads/jboss-a-mq-6.3.0.redhat-187.zip -d /opt
```

open the `/opt/amq/etc/users.properties` file and uncomment the following line to enable the admin user (i.e. remove the leading '#' character):

```
admin=admin,admin,manager,viewer,Operator, Maintainer, Deployer, Auditor, Administrator,
SuperUser
```

To configure the JCA connection between EAP and A-MQ, you need to retrieve the JBoss A-MQ Resource Adapter (RAR). So you have to unzip the RAR file and save it for later:

```
[root@hacep ~] unzip -o /opt/amq/extras/apache-activemq-5.11.0.redhat-630187-bin.zip -d
/tmp
[root@hacep ~]cp
/tmp/apache-activemq-5.11.0.redhat-630187/lib/optional/activemq-rar-5.11.0.redhat-630187.
rar /opt/jboss/activemq-rar.rar
```

Now you can start AMQ:

```
[root@hacep ~] amq
```

And verify that it's up and running opening firefox at the url `localhost:8181`, using the `admin` user with the `admin` password

Task 1.4: Configure EAP

Open another tab on your terminal, or another terminal window.
Now you have to start the JBoss EAP Command Line.

```
[root@hacep ~]# jboss-cli.sh
You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for
the list of supported commands.
[disconnected /] connect
[domain@localhost:9990 /]
```

First, remove all default servers and server groups from the EAP installation. We'll start several machines and we need to save every bit of memory we can:

```
/server-group=main-server-group:stop-servers
/server-group=other-server-group:stop-servers
/host=master/server-config=server-one:remove
```

```
/host=master/server-config=server-two:remove
/host=master/server-config=server-three:remove
/server-group=main-server-group:remove
/server-group=other-server-group:remove
```

Now you have to create a new server profile based on the full profile. You can do this by cloning it:

```
[domain@localhost:9990 /] /profile=full:clone(to-profile=hacep-full)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => undefined
}
```

Now, configure a new server-group which uses this profile

```
[domain@localhost:9990 /] /server-group=hacep:add(socket-binding-group=full-sockets,
profile=hacep-full)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => undefined
}
```

Add 4 servers to the server-group. This is done with the following CLI commands. Note that we are starting with a port-offset of 400 on the first server, 500 on the second, 600 on the third and 700 on the fourth.

```
[domain@localhost:9990 /] /host=master/server-config=hacep-1:add(group=hacep,
socket-binding-port-offset=400,auto-start=false)

[domain@localhost:9990 /]/host=master/server-config=hacep-2:add(group=hacep,
socket-binding-port-offset=500,auto-start=false)

[domain@localhost:9990 /] /host=master/server-config=hacep-3:add(group=hacep,
socket-binding-port-offset=600,auto-start=false)

[domain@localhost:9990 /] /host=master/server-config=hacep-4:add(group=hacep,
socket-binding-port-offset=700,auto-start=false)
```

Deploy the A-MQ Resource Adapter (the RAR file we copied to `/opt/jboss`) to the hacep server group:

```
[domain@localhost:9990 /] deploy /opt/jboss/activemq-rar.rar --server-groups=hacep
```

Set some needed system properties:

```
[domain@localhost:9990 /]
/server-group=hacep/system-property=org.apache.activemq.SERIALIZABLE_PACKAGES:add(value="*")

[domain@localhost:9990 /]
/server-group=hacep/system-property=kie.maven.settings.custom:add(value="/root/settingsEAP7-merge
.xml")

[domain@localhost:9990 /] /server-group=hacep/jvm=default:add(heap-size=2G,max-heap-size=2G)

[domain@localhost:9990 /]
/server-group=hacep/jvm=default/:write-attribute(name=environment-variables.HACEP_REPO,value=/opt
/repos)
```

Configure the JCA A-MQ connector:

```
[domain@localhost:9990 /]
/profile=hacep-full/subsystem=resource-adapters/resource-adapter=activemq-rar.rar:add(archive="ac
tivemq-rar.rar", transaction-support=XATransaction)

[domain@localhost:9990 /]
/profile=hacep-full/subsystem=resource-adapters/resource-adapter=activemq-rar.rar/config-properti
es=UserName:add(value="admin")

[domain@localhost:9990 /]
/profile=hacep-full/subsystem=resource-adapters/resource-adapter=activemq-rar.rar/config-properti
es=Password:add(value="admin")

[domain@localhost:9990 /]
/profile=hacep-full/subsystem=resource-adapters/resource-adapter=activemq-rar.rar/config-properti
es=ServerUrl:add(value="tcp://localhost:61616?jms.rmIdFromConnectionId=true")

[domain@localhost:9990 /]
/profile=hacep-full/subsystem=resource-adapters/resource-adapter=activemq-rar.rar/connection-defi
nitions=ConnectionFactory:add(class-name="org.apache.activemq.ra.ActiveMQManagedConnectionFactory
", jndi-name="java:/HACEPConnectionFactory", enabled=true, min-pool-size=1, max-pool-size=20,
pool-prefill=false, same-rm-override=false, recovery-username="admin", recovery-password="admin",
recovery-plugin-class-name="org.jboss.jca.core.recovery.ConfigurableRecoveryPlugin",
recovery-plugin-properties={"EnableIsValid" => "false","IsValidOverride" => "true"})
```

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

Task 1.5: Deploy the HACEP playground application and start the servers

And finally deploy the HACEP EAP Playground application:

```
[domain@localhost:9990 /] deploy /opt/jboss/hacep-eap-playground-1.0-SNAPSHOT.war
--server-groups=hacep
```

Task 1.6: Start the servers

Open another tab on your terminal, or another terminal window.
Now you have to start the JBoss EAP Command Line.

```
[root@hacep ~]# jboss-cli.sh
You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for
the list of supported commands.
[disconnected /] connect
[domain@localhost:9990 /]
```

Now you can start the servers:

```
[domain@localhost:9990 /] /server-group=hacep:start-servers
```

In the terminal where you started domain.sh, watch the logs and wait for the four servers to be correctly started. The Last line should be something like:

```
[Server:hacep-4] 11:23:32,593 INFO
[org.apache.camel.component.jms.DefaultJmsMessageListenerContainer] (Camel (camel-2)
thread #0 - JmsConsumer[facts]) Successfully refreshed JMS Connection
```

Now open Firefox and use a new tab to show the HACEP EAP Playground GUI, which is at `http://localhost:8480/hacep-playground`

The UI should show just 4 black dots, each dot representing a different HACEP node in the 4 nodes cluster

Task 1.7: Feed the system with events and look at the playground demo GUI

Open another tab on your terminal, or another terminal window.

```
[root@hacep ~]# cd ~/hacep/hacep-examples/hacep-perf-client/
[root@hacep hacep-perf-client]# mvn -s ../../../settingsEAP7-merge.xml -Psupported-GA-ee7
-o exec:java -Dduration=480 -Dconcurrent.players=20 -Ddelay.range=5 -Devent.interval=1
-Dtest.preload=true -Dtest.messages=10 -Dbroker.host="localhost:61616"
-Dbroker.authentication=true -Dbroker.usr=admin -Dbroker.pwd=admin
-Dorg.apache.activemq.SERIALIZABLE_PACKAGES="*"
```

In the HACEP GUI, sessions (balls in UI) will be shown, connected to the black dots representing the server nodes. The larger balls represent live sessions running in their primary owner, the smaller balls represent the backups of these sessions (matching colors) on their backup nodes, providing high availability.

To demonstrate the HA characteristics of the platform you can stop and start nodes of the cluster. The HACEP Playground GUI will show how the JDG cluster rebalances and performs its state-transfer, transferring and starting the Drools/BRMS CEP sessions on correct nodes. You can start and stop nodes using the following commands as a guideline:

```
[domain@localhost:9990 /] /host=master/server-config=hacep-4:stop
```

Wait some seconds and observe the GUI, the three remaining nodes should rebalance their sessions.
You can restart the node with the following command:

```
[domain@localhost:9990 /] /host=master/server-config=hacep-4:start
```

You can now play with the demo starting and stopping other nodes.

DO NOT stop the first node (`hacep-1`), because that's the one you are connected to with the webapp. If you do that by mistake, connect to another node (the offset is +100).
http://localhost:8480/hacep-playground
http://localhost:8580/hacep-playground
http://localhost:8680/hacep-playground
http://localhost:8780/hacep-playground

Task 1.8: Close the browser, stop the client and the 4 servers

When you are done, close the browser (because the D3 application is pretty memory hungry), stop the JMS client and stop the nodes too (from the CLI).

```
[domain@localhost:9990 /] /server-group=hacep:stop-servers
```

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

LAB 2: MODIFY THE RULES

Estimated time (10 minutes)

You will now do a simple modification of the EAP playground rule. To modify it, you will do a simple change to the DRL code and you'll recompile and redeploy the whole jar.

That is not the best way of doing that, and in another lab you'll do it in a much more effective way.

Task 2.1: Modify the DRL

Open JBoss Developer Studio (or vi):

```
[root@hacep ~]# /usr/local/devstudio/devstudio-wayland
```

Open the DRL file reward-point-level-demo.drl which is in /hacep/hacep-examples/hacep-rules/src/main/resources/rules/

If you prefer CLI, you can just use vi:

```
[root@hacep ~]# vi
/hacep/hacep-examples/hacep-rules/src/main/resources/rules/reward-point-level-demo.drl
```

Modify the rule as following, changing from 10 to 5 the number of times.

```
...
channels["playerPointsLevel"].send(new PlayerPointLevel(
            $playerId,
            $numberOfTimes.intValue() % 5,
            $numberOfTimes.intValue() / 5)
        );
end
```

This change means that now the user will still gain a point for each gameplay but it will increase its "player level" every 5 points instead of 10.

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

IMPORTANT: after modifying remember to save (`Ctrl+s`).

Task 2.2: Redeploy the app

To re-deploy the app, recompile it with Maven and copy again the war in the /opt/jboss directory:

```
[root@hacep ~]# cd ~/hacep
[root@hacep ~]# mvn -s ../settingsEAP7-merge.xml -Psupported-GA-ee7 -o -DskipTests clean
install
[root@hacep ~]# cp
~/.m2/repository/it/redhat/jdg/examples/hacep-eap-playground/1.0-SNAPSHOT/hacep-eap-playg
round-1.0-SNAPSHOT.war /opt/jboss

(if asked overwrite it)
```

Reconnect to the domain with the jboss CLI:

```
[root@hacep ~]# jboss-cli.sh
You are disconnected at the moment. Type 'connect' to connect to the server or 'help' for
the list of supported commands.
[disconnected /] connect
[domain@localhost:9990 /]
```

Redeploy the HACEP EAP Playground application and restart the servers:

```
[domain@localhost:9990 /] undeploy hacep-eap-playground-1.0-SNAPSHOT.war --server-groups=hacep
[domain@localhost:9990 /] deploy /opt/jboss/hacep-eap-playground-1.0-SNAPSHOT.war
--server-groups=hacep
[domain@localhost:9990 /] /server-group=hacep:start-servers
```

In the terminal where you started domain.sh, watch the logs and wait for the four servers to be correctly started. The Last line should be something like:

```
[Server:hacep-4] 11:23:32,593 INFO
[org.apache.camel.component.jms.DefaultJmsMessageListenerContainer] (Camel (camel-2)
thread #0 - JmsConsumer[facts]) Successfully refreshed JMS Connection
```

Open Firefox and use a new tab to show the HACEP EAP Playground GUI, which is at `http://localhost:8480/hacep-playground.`

Restart the JMS client from its terminal:

```
[root@hacep hacep-perf-client]# mvn -s ../../../settingsEAP7-merge.xml -Psupported-GA-ee7
-o exec:java -Dduration=480 -Dconcurrent.players=20 -Ddelay.range=5 -Devent.interval=1
-Dtest.preload=true -Dtest.messages=10 -Dbroker.host="localhost:61616"
-Dbroker.authentication=true -Dbroker.usr=admin -Dbroker.pwd=admin
-Dorg.apache.activemq.SERIALIZABLE_PACKAGES="*"
```

The rule should now fire every 5 points, so the animation should go faster.

Task 2.3: Close the browser and stop the JMS client

When you are done, close again the browser, and stop the JMS client.

Do NOT stop HACEP, we will update the live system in the next lab

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

LAB 3 (OPTIONAL): UPDATE THE RULES, THE RIGHT WAY

Estimated time (10 minutes)

You will now reverse the simple modification you did to the rule in Lab 2, this time using the rules hot deploy (instead of redeploying everything).

To do that, you will update the DRL, update the rule GAV, publish the rule to the Maven repo, instruct HACEP to perform the rules upgrade and let HACEP do the rest.

HACEP will:

- Stop the flow of messages
- Download the latest version of the rule
- Recompile the rule in real time
- Update the serialized sessions in the cache, if any
- Restart the flow of messages

Task 3.1: Modify DRL back

Open JBoss Developer Studio (or `vi)`:

```
[root@hacep ~]# /usr/local/devstudio/devstudio-wayland
```

Open the DRL file `reward-point-level-demo.drl` which is in `/hacep/hacep-examples/hacep-rules/src/main/resources/rules/`

If you prefer CLI, you can just use `vi`:

```
[root@hacep ~]# vi
~/hacep/hacep-examples/hacep-rules/src/main/resources/rules/reward-point-level-demo.drl
```

Modify the rule rolling back the number of times from 5 to 10.

```
...
channels["playerPointsLevel"].send(new PlayerPointLevel(
            $playerId,
            $numberOfTimes.intValue() % 10,
            $numberOfTimes.intValue() / 10)
       );
end
```

IMPORTANT: after modifying remember to save (`Ctrl+s`).

This change reverts back the rule from what was done in lab 2.

Task 3.2: Change the hacep-rules release version

Open the hacep-rules pom file `pom.xml` which is in `/hacep/hacep-examples/hacep-rules/`

Change the project version from 1.0 to 2.0:

```
...
<artifactId>hacep-rules</artifactId>
<version>2.0</version>
<name>HACEP Example Rules</name>
<packaging>jar</packaging>
```

IMPORTANT: be sure to modify the right `<version>` tag which is the one just after `<artifactId>hacep-rules</artifactId>`.

IMPORTANT: after modifying remember to save (`Ctrl+s`).

Task 3.4: Recompile the hacep-rules module

To recompile only the hacep-rules with Maven, open a terminal:

```
[root@hacep ~]# cd ~/hacep/hacep-examples/hacep-rules/
[root@hacep ~]# mvn -s ../../../settingsEAP7-merge.xml -Psupported-GA-ee7 -o clean install
```

This should put version 2.0 of hacep-rules in your local .m2 directory, you can verify with:

```
[root@hacep ~]# ll ~/.m2/repository/it/redhat/jdg/examples/hacep-rules/
total 4
drwxr-xr-x. 2 root root 166 Apr 15 11:47 1.0
drwxr-xr-x. 2 root root  88 Apr 15 11:24 2.0
-rw-r--r--. 1 root root 376 Apr 15 11:47 maven-metadata-local.xml
```

Task 3.5: Trigger the new rules hot deploy

HACEP gets cluster commands from a special queue, aptly named **commands**. This queue is used to manage the whole HACEP cluster, which could be much bigger than just four nodes: so it's pretty useful to be able to send just one single command to operate on the whole cluster.

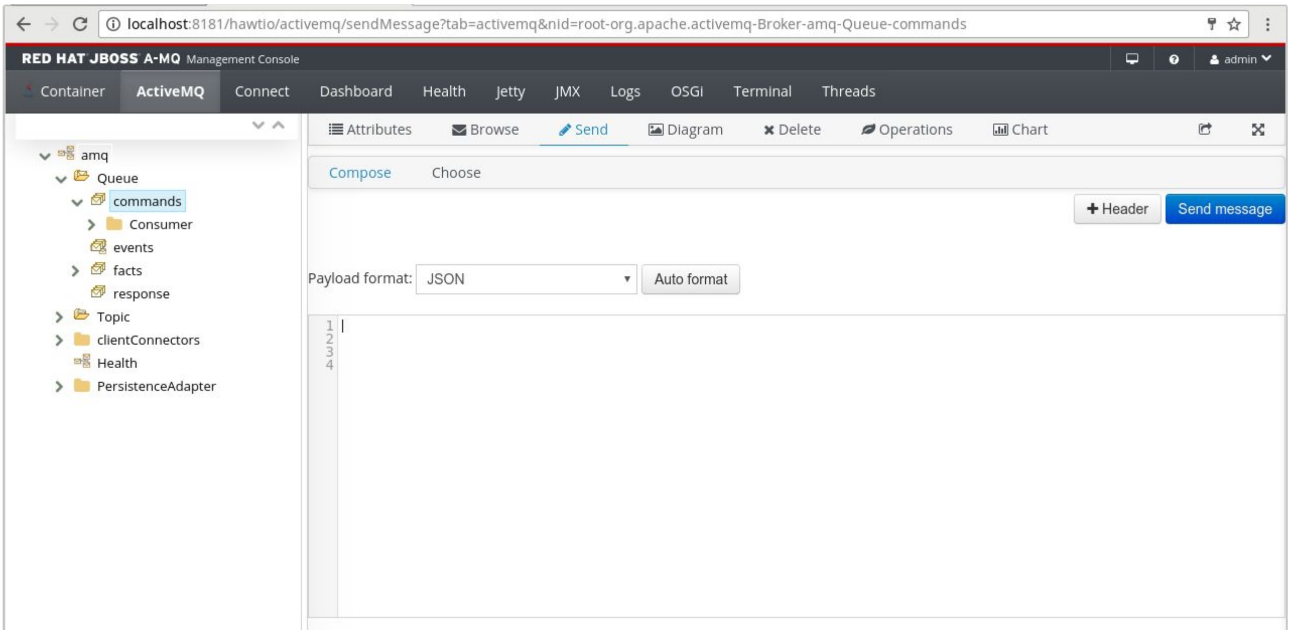
In order to start the rules upgrade process we'll send a json command **upgrade**: open a browser tab and go to `http://localhost:8181`

Log-in with username: `admin` and password: `admin` in the AMQ broker administration console:

1. Click on **activemq** tab
2. Then on **queue**
3. Then on **commands**
4. And finally on **send**

You should see something similar to the following picture:

HACEP: Highly Available and Horizontally Scalable Complex Event Processing



Payload format: JSON

Now copy and paste the following command using JSON as a payload format:

```
{
  "command":"UPGRADE",
  "params": [{"key": "RELEASE_ID","value": "it.redhat.jdg.examples:hacep-rules:2.0"}]
}
```

And finally click “Send Message”.

Now HACEP should stop the flow of messages, update the system to the latest version of the rules (picking the new rule directly from your `~/m2/repository`), recompile the rule in real time, update the serialized sessions in the cache (if any) and restart the flow of messages.

Restart the JMS client from its terminal:

```
[root@hacep hacep-perf-client]# mvn -s ../../../settingsEAP7-merge.xml -Psupported-GA-ee7
-o exec:java -Dduration=480 -Dconcurrent.players=20 -Ddelay.range=5 -Devent.interval=1
-Dtest.preload=true -Dtest.messages=10 -Dbroker.host="localhost:61616"
-Dbroker.authentication=true -Dbroker.usr=admin -Dbroker.pwd=admin
-Dorg.apache.activemq.SERIALIZABLE_PACKAGES="*"
```

Observe in the GUI that the rules are back to the previous version, which updates the player level every 10 player points. All of this, on a live system.

Task 3.6: Close the browser, stop the client and the 4 servers

When you are done, close again the browser, stop the JMS client and stop the nodes too (from the CLI).

```
[domain@localhost:9990 /] /server-group=hacep:stop-servers
```

HACEP: Highly Available and Horizontally Scalable Complex Event Processing

Links and References

This section contains useful links and resources

References

Links to useful material

1. HACEP Reference Architecture: https://access.redhat.com/articles/2542881

This Document

This Document was created for the Red Hat Summit 2017.