

Avaya Business Rules Engine 3.4 Cluster Management Guide



Avaya Business Rules Engine 3.4 Cluster Management Guide

IMPORTANT

Updated Dec 17, 2019.

This document describes how to configure an Avaya Business Rules Engine deployment model, including node and cluster properties.

- 1. Avaya Business Rules Engine 3.4 Cluster Management Guide 2
 - 1.1 Avaya Business Rules Engine 3.4 Configuring Internal Firewall - IPTables 26
 - 1.2 Avaya Business Rules Engine 3.4 Post Installation and Configuration Sanity Test Checklist 29
 - 1.3 Avaya Business Rules Engine Load Balancer Sanity Test Checklist 32

- 1 Avaya Business Rules Engine 3.4 Cluster Management Guide
- 2 Configure Avaya Business Rules Engine Installation Type and Deployment Model
 - 2.1 Configuration Script and Execution
- 3 Start Avaya Business Rules Engine Services
- 4 Load Avaya Business Rules Engine Default Data
- 5 Update Avaya Business Rules Engine Config DB Data
- 6 Stop Avaya Business Rules Engine
- 7 Check Status of Avaya Business Rules Engine Services
- 8 Start/Stop the CMS Connector
- 9 Configure the Traffic Limit
- 10 Cluster and Node Configuration Properties
 - 10.1 Cluster Properties
 - 10.2 Node Properties
 - 10.2.1 Kafka Mirrors
- 11 Cluster Manager Tool (CMT)
 - 11.1 Cluster Manager Tool Command Description and Usage
 - 11.2 Cluster Manager Tool Command Parameters
- 12 Using Avaya Business Rules Engine
- 13 Appendix A - (Optional) Importing Demo Data
- 14 Appendix B - Security Enable https
- 15 Appendix C - Replacing SSL certificates
- 16 Appendix D - Avaya Business Rules Engine Services
- 17 Appendix E - Disable TLS v1.1

Configure Avaya Business Rules Engine Installation Type and Deployment Model

There are two ways to configure the Avaya Business Rules Engine deployment model, either by:

- executing the **configuration script** to automatically set the parameters (**recommended**)
- running the Cluster Manager Tool to manually set individual parameters

The configuration script, provides an interactive way to setup the entire cluster in a simpler manner. This script is part of the Avaya Business Rules Engine installation package.

By default, the script configures Avaya Business Rules Engine to write the **RDRs** (Routing Decision Records) in the Management Services Nodes. To setup a different destination for the RDRs (e.g. external database), review the Avaya Business Rules Engine - RDR Reference Guide (part of the Integration Guides).

Configuration Script and Execution

The Avaya Business Rules Engine installer package includes the [*configureCluster.sh*](#) script, under the */Tools* directory, which provides an easy way to configure any Avaya Business Rules Engine deployment model. The script runs the required Cluster Manager Tool commands on behalf of the user.

The script can be run in one of two modes:

- Attended mode: asks questions during the execution before applying the changes
- Unattended mode: runs using a properties file that contains all required parameters

After receiving all parameters (from either user input or properties file), the script executes commands remotely, on all nodes. These remote commands are executed via SSH, using the user/password. Even in unattended mode, the user/password should be provided manually when requested. By default, the script uses the current user to access all nodes (e.g. if you are running the script as root, a root password is requested to access the nodes). A different user can be provided, as parameter, if needed. Before applying the configuration, the script asks for the password for each node, and installs a key on each one, to avoid asking for passwords later.

The script first provides the prompt for "Installation Type". Options are:

- 1) Production
- 2) Lab

Select the Installation Type (which in turn will require the corresponding BRE license).

The script then needs to know which Deployment Model should be used. Options are:

- 1) Single Node - Single Data Center - (Total - 1 Server)
- 2) Triple Node - Single Data Center - (Total - 3 Servers)
- 3) Single Node - Dual Data Center - (Total - 2 Servers)
- 4) Triple Node - Dual Data Center - (Total - 6 Servers)

Select the Deployment Model based on customer needs. Depending on the Deployment Model selected, the script will require other information such as:

- Data Center Names - a string that identifies the Data Center Name
- Node IPs or Hostnames - IPs/Hostnames of the nodes used for Avaya Business Rules Engine (**Note:** in Triple Node deployments, [the first two IPs should be the MGMT IPs](#))
- WebLM Address - IP/Hostname where the license server is installed (standalone WebLM or Avaya System Manager)
- WebLM Port - Port on the license server to access the Avaya Business Rules Engine License

[tmp/configureCluster.log](#) is a log file generated by the script containing:

- Selected Deployment Type
- Parameters for all nodes
- SSH connection status and errors
- All commands executed by the script

Note: the script only runs Avaya Business Rules Engine Cluster Manager commands; it does not change Linux configuration.

Before you begin:

- Read the Avaya Business Rules Engine Planning Guide and prepare the node(s) with all the prerequisites mentioned in it;
- Read the Avaya Business Rules Engine Installation Guide and install the Avaya Business Rules Engine software on all nodes;
- Keep on hand, all the User IDs/Passwords that are able to login to the nodes.

Procedure:

Note: the script needs to be run only once, on one node, to configure all nodes and both Data Centers (in the case of dual DC).

To run the script, first mount the Avaya Business Rules Engine installation ISO and then execute:

- [<MOUNT_POINT>/Tools/configureCluster.sh](#)
 - Parameters:
 - -u <user> (optional) - set a different user to run the Avaya Business Rules Engine Cluster Manager Tool commands (via SSH) (if not provided, current user is used)
 - -n <unattended_file> (optional) - run the script in unattended mode, using the properties in the named file
 - -c <certificate> (optional) - identify a certificate to connect to the nodes via SSH

The configuration script can be run without any parameters (any access keys needed will be generated dynamically). However, there is some special handling of the parameters needed, in some cases, which is described below.

Before running the configuration script, Linux should be configured to accept ssh with sroot user:

1. As sroot:

1. `cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bck`
2. `vi /etc/ssh/sshd_config`

1. Change the lines:

1. PermitRootLogin yes

1. from no to yes

2. DenyGroups root

1. Comment the line adding a # (#DenyGroups root)

3. `systemctl restart sshd`

2. Run the script, adding the user at the end: *<MOUNT_POINT>/Tools/configureCluster.sh -u sroot*

3. Revert the sshd configuration change:

1. `cp /etc/ssh/sshd_config.bck /etc/ssh/sshd_config`
2. `systemctl restart sshd`

Unattended Mode (requires setting of the Unattended_file Parameter)

To run in unattended mode, the script should receive a file with the configuration parameters, e.g.

```
INSTALLATION_TYPE=1
DEPLOYMENT_TYPE=1
WEBLM_IP=10.130.92.242
WEBLM_PORT=52233
DC1_NAME=DCA
DC1_MGMTS=10.130.92.195
DC1_CORES=10.130.92.195
DC1_MSGS=10.130.92.195
DC1_AGGS=10.130.92.195
DC1_RPTS=10.130.92.195
DC2_NAME=
DC2_MGMTS=
DC2_CORES=
DC2_MSGS=
DC2_AGGS=
DC2_RPTS=
DC_COUNT=
```

A sample of this file (containing instructions) is included in the Avaya Business Rules Engine installation package.

To execute the script in unattended mode enter *<MOUNT_POINT>/Tools/configureCluster.sh -n /tmp/config.properties*

Note: The sample unattended properties file, in the installer ISO, is read-only. To use it, copy the file to a read-write directory first, edit it to match your needs, and then point the script to it.

Note: If a password is required, the script requests it, even in unattended mode. To not require this input during installation, run in attended mode first, to add the certificates; subsequent executions will require no user interaction.

Using a Certificate (requires setting of the Certificate Parameter)

An already existing certificate can be used by the configuration script to connect to Avaya Business Rules Engine nodes. It can be used in environments like AWS. The "-c" parameter identifies the certificate.

Start Avaya Business Rules Engine Services

Once Avaya Business Rules Engine is installed and the Deployment Model configured, it is time to start up the services.

Run the command below:

```
systemctl start abre-services
```

Run command `su - avayadr -c "is-abre-available.sh"` to have real-time confirmation, when the User Interface is available.

The `/var/log/messages` file has the output of the services initialization. A tail on this log can be done to have a real time view of the services startup.

Load Avaya Business Rules Engine Default Data

It is necessary to run a command to load Avaya Business Rules Engine with the initial default data. This command should be executed in **ONLY ONE** of the nodes that has Management Services enabled. This means, on **ONLY one NODE and ONLY one Data Center** (otherwise the default data will have different IDs on each DC).

```
sudo su - avayadr -c "cluster-mgr.sh --command init-configuration" loads default data
```

IMPORTANT: On **Dual Data Center** deployments, this command should be executed **AFTER ALL SERVICES ON ALL NODES** are up and running.

IMPORTANT: You should **not** Load Default Data in an Upgrade scenario.

Update Avaya Business Rules Engine Config DB Data

It is necessary to run a command to update the Avaya Business Rules Engine's default data. This command should be executed in **ONLY ONE** of the nodes that has Management Services enabled. This means, on **ONLY one NODE and ONLY one Data Center**.

```
sudo su - avayadr -c "cluster-mgr.sh --command update-configuration" updates default data
```

IMPORTANT: On **Dual Data Center** deployments, this commands should be executed **AFTER ALL SERVICES ON ALL NODES** are up and running.

IMPORTANT: You should **not** Update Default Data in a Fresh Install scenario.

Stop Avaya Business Rules Engine

The user can follow the command below to stop all Avaya Business Rules Engine services.

Run the command below:

```
systemctl stop abre-services
```

Check Status of Avaya Business Rules Engine Services

The user can follow the command below to check the status of all the Avaya Business Rules Engine services.

Run the command below:

```
sudo su - avayadr abre-services-status.sh
```

Start/Stop the CMS Connector

The CMS Connector is required to feed real-time metrics to Avaya Business Rules Engine. The CMS Connector has its own service. The following table shows how to control it.

Application	Service	How to use
CMS Connector	cms-connector	<i>systemctl start cms-connector [start / stop / restart / status]</i>

Configure the Traffic Limit

ABRE 3.4 (or later) requires a new license. In conjunction with this license, the new System Property (SP) *maxTransactionsPerHour* defines the max traffic (N x 1000 requests/hour) that ABRE will process. Each Data Center acquires the traffic bandwidth prescribed by the SP from the license server (license >= SP). Therefore, the license must have sufficient ability to serve the System Property request. Traffic sent to BRE that is higher than the SP limit, will raise an alarm. After 30 days, if the traffic is still higher than the SP limit, the traffic over-the-limit will be refused by BRE.

IMPORTANT: after installation configure the System Property (SP) *maxTransactionsPerHour* to an appropriate value, to meet the traffic needs, and within the license limits.

Cluster and Node Configuration Properties

There are many configuration properties that determine how services operate in the Avaya Business Rules Engine nodes and across the clusters. The parameters are grouped as either:

- cluster properties or
- node properties

and are described below.

Cluster Properties

The command line to configure cluster properties via the Cluster Manager Tool is the following:

```
sudo su - avayadr -c "cluster-mgr.sh --command update-cluster-configuration --set <property> --value <new_value>"
```

For example:

```
sudo su - avayadr -c "cluster-mgr.sh --command update-cluster-configuration --set lookup-hosts --value 10.130.125.30"
```

Note: The property value, in some cases, can be a string starting with dash '-'. In order to avoid the Cluster Manager Tool interpreting this as another command parameter, the value needs to be provided using single quotes ', and needs to start with a space. For example: to set the below property with value **--disable**, the command should be:


```
sudo su - avayadr -c "cluster-mgr.sh --command update-cluster-configuration --set propertyName --value ' --disable'"
```

The following line would be incorrect. Cluster Manager Tool would interpret it not as a value, but another parameter.

```
sudo su - avayadr -c "cluster-mgr.sh --command update-cluster-configuration --set propertyName --value --disable"
```

All cluster properties are contained in a file called [\[JAVA_HOME\]/abre/config/cluster.properties](#).

The table below lists all the cluster properties supported by the Cluster Manager Tool and gives an example of how to use them.

 *It is very important that you do NOT include blank spaces in the parameters. For example, for a Data Center Name, you can separate words using "-" or "_". If you enter a DC name like "DC A", the cluster will not recognize the blank space properly and the configuration will not be correct.*

Property	Description	Needs Restart	Cluster Manager Tool example
name	Name of the Cluster	Yes	<code>--command update-cluster-configuration --set name --value DCA</code>
lookup-hosts	IP Addresses of Nodes hosting lookup service (separated by commas)	Yes	<code>--command update-cluster-configuration --set lookup-hosts --value 10.130.125.30</code>
weblm	URL to the WebLM server	Yes	<code>--command update-cluster-configuration --set weblm --value https://10.130.125.30:52233/WebLM/LicenseServer</code> Note: Depending on the WebLM version, the URL can change. The example above illustrates a connection with a WebLM hosted by an Avaya System Manager version 7.X
mgm-servers-count	Number of Nodes hosting MANAGEMENT zone	Yes	<code>--command update-cluster-configuration --set mgm-servers-count --value 2</code>
core-servers-count	Number of Nodes hosting CORE zone	Yes	<code>--command update-cluster-configuration --set core-server-count --value 3</code>
zk-local-connection-string	String containing IP Address and Port of Local Zookeeper	Yes	<code>--command update-cluster-configuration --set zk-local-connection-string --value 127.0.0.1:2181/local</code>
zk-consolidated-connection-string	String containing IP Address and Port of local Consolidated Zookeeper	Yes	<code>--command update-cluster-configuration --set zk-consolidated-connection-string --value 127.0.0.1:2181/consolidated</code>
has-remote-cluster	Flag indicating whether deployment includes a Remote Cluster or not. This configuration is required for WAN Gateway.	No	<code>--command update-cluster-configuration --set has-remote-cluster --value false</code>
remote-lookup-hosts	IP Addresses of Nodes hosting MANAGEMENT service, on Remote Cluster (separated by commas). This configuration is required for WAN Gateway.	Yes	<code>--command update-cluster-configuration --set remote-lookup-hosts --value 10.130.122.24</code>
remote-cluster-name	Name of Remote Cluster. This configuration is required for WAN Gateway.	Yes	<code>--command update-cluster-configuration --set remote-cluster-name --value DCB</code>

elastic-servers	IP Addresses and Ports of Nodes hosting Elasticsearch service (separated by commas)	Yes	--command update-cluster-configuration --set elastic-servers --value 10.130.125.30:9300,10.130.125.31:9300
logstash-kafka-servers	IP Addresses and Ports of Nodes hosting Kafka service (separated by commas)	Yes	--command update-cluster-configuration --set logstash-kafka-servers --value 10.130.125.30:9092,10.130.125.31:9092
logstash-elastic-servers	IP Addresses and Ports of Nodes hosting Elasticsearch -Logstash (separated by commas)	Yes	--command update-cluster-configuration --set logstash-elastic-servers --value 10.130.125.30:9200,10.130.125.31:9200
zk-hosts	IP Addresses and Ports of Nodes hosting Zookeepers (separated by commas)	Yes	--command update-cluster-configuration --set zk-hosts --value 10.130.125.30:3191:4191,10.130.125.30:3192:4192,10.130.125.31:3191:4191,10.130.125.31:3192:4192
kafka-local-hosts	IP Addresses and Ports of Nodes hosting Local Kafka (separated by commas)	Yes	--command update-cluster-configuration --set kafka-local-hosts --value 10.130.125.30:9092,10.130.125.31:9092
kafka-consolidated-hosts	IP Addresses and Ports of Nodes hosting Consolidated Kafka (separated by commas)	Yes	--command update-cluster-configuration --set kafka-consolidated-hosts --value 10.130.125.30:19092,10.130.125.31:19092
decisions-persistence-kafka-servers	IP Addresses and Ports of Nodes hosting local Kafka (separated by commas) for decisions topic persistence	Yes	--command update-cluster-configuration --set decisions-persistence-kafka-servers --value 10.130.125.30:9092,10.130.125.31:9092
auditlog-persistence-kafka-servers	IP Addresses and Ports of Nodes hosting local Kafka (separated by commas) for Audit Log topic persistence	Yes	--command update-cluster-configuration --set auditlog-persistence-kafka-servers --value 10.130.125.30:9092,10.130.125.31:9092
destination-metrics-consumer-brokers	IP Addresses and Ports of Nodes hosting local Kafka (separated by commas) for Metrics Consumers	Yes	--command update-cluster-configuration --set destination-metrics-consumer-brokers --value <ip01:port01,ip02:port02,...,ipN:portN>
metrics-adjustment-producer-brokers	IP Addresses and Ports of Nodes hosting local Kafka (separated by commas) for Metrics Adjustment Producers	Yes	--command update-cluster-configuration --set metrics-adjustment-producer-brokers --value <ip01:port01,ip02:port02,...,ipN:portN>
metrics-adjustment-consumer-brokers	IP Addresses and Ports of Nodes hosting local Kafka (separated by commas) for Metrics Adjustment Consumers	Yes	--command update-cluster-configuration --set metrics-adjustment-consumer-brokers --value <ip01:port01,ip02:port02,...,ipN:portN>

Node Properties

The command line to configure node properties via the Cluster Manager Tool is the following:

```
sudo su - avayadr -c "cluster-mgr.sh --command update-node-configuration --set <property> --value <new_value>"
```

All node properties are contained in the file called [\[AVA YA_HOME\]/abre/config/node.properties](#).

The table below lists all the node properties supported by the Cluster Manager Tool and gives an example of how to use it.

Property	Description	Needs Restart	Cluster Manager Tool example
mgm-enabled	Flag indicating whether the Management Zone is enabled in this Node	Yes	--command update-node-configuration --set mgm-enabled --value true
core-enabled	Flag indicating whether the CORE Zone is enabled in this Node	Yes	--command update-node-configuration --set core-enabled --value true
mgm-memory	Total amount of Memory (in MB) to be allocated to the (GigaSpaces) Management Zone in this Node	Yes	--command update-node-configuration --set mgm-memory --value 1024
core-memory	Total amount of Memory (in MB) to be allocated to the (GigaSpaces) Core Zone in this Node	Yes	--command update-node-configuration --set core-memory --value 1024
host-address	IP Address of this Node	Yes	--command update-node-configuration --set host-address --value 10.120.120.124
audit-purge-db-user	DB User used by Audit Log to save records into the Audit Log table	Yes	--command update-node-configuration --set audit-purge-db-user --value dynamicrouting
audit-purge-db-pwd	DB Password used by Audit Log to save records into the Audit Log table	Yes	--command update-node-configuration --set audit-purge-db-pwd --value 123456
audit-purge-retention-period	Retention period (in days or hours) that Audit Log Records will be kept in the Database (older ones will be deleted) <ul style="list-style-type: none">• [1-23] h - for hours, e.g. 1h is 1 hour• d - for days, e.g. 1d is 1 day	No	--command update-node-configuration --set audit-purge-retention-period --value 30d #kept for 30 days --command update-node-configuration --set audit-purge-retention-period --value 10h #kept for 10 hours

audit-purge-schedule	<p>Schedule pattern (CRON) to periodically purge Audit Log Records. The supported patterns are:</p> <ul style="list-style-type: none"> • m - to execute every minute: 'm [1...]' (every X minutes) • h - for hourly executions: 'h [1-23]' (every X hours) • d - for daily executions: 'd [0-23]' (every day at X o'clock) • w - for weekly executions: 'w [0-23] [0-6]' (every week at X o'clock on Y day; where [0-23] is the hour of the day and [0-7] is the day of the week Sunday is either 0 or 7. note: day of week can also be specified as: sun, mon, tue, wed, thu, fri, sat) • M - for monthly executions: 'm [0-23] [1-31]' (every month at X o'clock on Y day; where [0-23] is the hour of the day and [1-31] is the day of the month) 	No	<pre>--command update-node-configuration --set audit-purge-schedule --value 'm 5' #every 5 minutes --command update-node-configuration --set audit-purge-schedule --value 'h 12' #every 12 hours starting when the command was run --command update-node-configuration --set audit-purge-schedule --value 'd 20' #every day at 8 PM --command update-node-configuration --set audit-purge-schedule --value 'w 3 1' #every Monday at 3 AM --command update-node-configuration --set audit-purge-schedule --value 'w 0 sat' #every Saturday at midnight --command update-node-configuration --set audit-purge-schedule --value 'M 12 1' #every first day of the month at noon</pre>
max-rdr-records	The maximum number of records to keep in the RDR database		<pre>--command update-node-configuration --set max-rdr-records --value 10000 # will allow up to 10000 records in the rdr database</pre>
rdr-purge-schedule-max-records	<p>Schedule pattern (CRON) to periodically purge RDR records. The supported patterns are:</p> <ul style="list-style-type: none"> • m - to execute every minute: 'm [1...]' (every X minutes) • h - for hourly executions: 'h [1-23]' (every X hours) • d - for daily executions: 'd [0-23]' (every day at X o'clock) • w - for weekly executions: 'w [0-23] [0-6]' (every week at X o'clock on Y day; where [0-23] is the hour of the day and [0-7] is the day of the week Sunday is either 0 or 7. note: day of week can also be specified as: sun,mon,tue,wed,thu,fri,sat) • M - for monthly executions: 'm [0-23] [1-31]' (every month at X o'clock on Y day; where [0-23] is the hour of the day and [1-31] is the day of the month) 	No	<pre>--command update-node-configuration --set rdr-purge-schedule-max-records --value 'm 5' #every 5 minutes --command update-node-configuration --set rdr-purge-schedule-max-records --value 'h 23' #every 23 hours starting when the command was run --command update-node-configuration --set rdr-purge-schedule-max-records --value 'd 23' #every day at 11 PM --command update-node-configuration --set rdr-purge-schedule-max-records --value 'w 0 0' #every Sunday at midnight --command update-node-configuration --set rdr-purge-schedule-max-records --value 'w 1 wed' #every Wednesday at 1 AM --command update-node-configuration --set rdr-purge-schedule-max-records --value 'M 16 28' #every month on day 28 at 4 PM</pre>
elastic-purge-retention-period	<p>Retention period (in days or hours) that Elasticsearch Records will be kept (older ones will be deleted)</p> <ul style="list-style-type: none"> • [1-23] h - for hours, e.g. 10h is 10 hours • d - for days, e.g. 31d is 31 days 	No	<pre>--command update-node-configuration --set elastic-purge-retention-period --value 23h #kept for 23 hours</pre>

elastic-purge-schedule	<p>Schedule pattern (CRON) to periodically purge Elasticsearch Records. The supported patterns are:</p> <ul style="list-style-type: none"> • m - to execute every minute: 'm [1...]' (every X minutes) • h - for hourly executions: 'h [1-23]' (every X hours) • d - for daily executions: 'd [0-23]' (every day at X o'clock) • w - for weekly executions: 'w [0-23] [0-6]' (every week at X o'clock on Y day; where [0-23] is the hour of the day and [0-7] is the day of the week Sunday is either 0 or 7. note: day of week can also be specified as: sun,mon,tue,wed,thu,fri,sat) • M - for monthly executions: 'm [0-23] [1-31]' (every month at X o'clock on Y day; where [0-23] is the hour of the day and [1-31] is the day of the month) 	No	<pre>--command update-node-configuration --set elastic-purge-schedule --value 'm 5' #every 5 minutes --command update-node-configuration --set elastic-purge-schedule --value 'h 10' #every 10 hours starting when the command was run --command update-node-configuration --set elastic-purge-schedule --value 'd 11' #every day at 11 AM --command update-node-configuration --set elastic-purge-schedule --value 'w 13 1' #every Monday at 1 PM --command update-node-configuration --set elastic-purge-schedule --value 'w 17 thu' #every Thursday at 5 PM --command update-node-configuration --set elastic-purge-schedule --value 'M 5 1' #every month on day 1 at 5 AM</pre>
zookeeper-local-enabled	Flag indicating whether the specified Zookeeper Service (Local or Consolidated) is enabled or not.	Yes	<pre>--command update-node-configuration --set zookeeper-local-enabled --value true</pre>
zookeeper-consolidated-enabled	<p>A Consolidated instance is required to copy data from other clusters (using mirror process) in case of multi CORE deployments.</p> <p>(default : local = true, consolidated = false)</p>		<pre>--command update-node-configuration --set zookeeper-consolidated-enabled --value false</pre>
zookeeper-default-count	Number of Instances of Zookeeper Services (default : 3)	Yes	<pre>--command update-node-configuration --set zookeeper-default-count --value 3</pre>
zookeeper-default-basenum	Number to use to start naming the Zookeeper Instances (default : 0)	Yes	<pre>--command update-node-configuration --set zookeeper-default-basenum --value 0</pre>
zookeeper-local-baseclientport	Number to use to start numbering the Zookeeper Client Ports. If this is N, and instances.count is M, Zookeeper Local Service will use ports from N to N+M. (default : 2181)	Yes	<pre>--command update-node-configuration --set zookeeper-local-baseclientport --value 2181</pre>
zookeeper-consolidated-baseclientport	Number to use to start numbering the Zookeeper Client Ports. If this is N, and instances.count is M, Zookeeper Local Consolidated Service will use ports from N to N+M. (default : 12181)	Yes	<pre>--command update-node-configuration --set zookeeper-consolidated-baseclientport --value 12181</pre>
zookeeper-local-baseleaderport	Number to use to start numbering the Zookeeper Leader Ports. If this is N, and instances.count is M, Zookeeper Service will use ports from N to N+M. (default : 3191)	Yes	<pre>--command update-node-configuration --set zookeeper-local-baseleaderport --value 3191</pre>
zookeeper-consolidated-baseleaderport	Number to use to start numbering the Zookeeper Leader Ports. If this is N, and instances.count is M, Zookeeper Service will use ports from N to N+M. (default : 13191)	Yes	<pre>--command update-node-configuration --set zookeeper-consolidated-baseleaderport --value 13191</pre>

zookeeper-local-baseelectionport	Number to use to start numbering the Zookeeper Election Ports. If this is N, and instances.count is M, Zookeeper Service will use ports from N to N+M. (default : 4191)	Yes	--command update-node-configuration --set zookeeper-local-baseelectionport --value 4191
zookeeper-consolidated-baseelectionport	Number to use to start numbering the Zookeeper Election Ports. If this is N, and instances.count is M, Zookeeper Service will use ports from N to N+M. (default : 14191)	Yes	--command update-node-configuration --set zookeeper-consolidated-baseelectionport --value 14191
kafka-local-enabled	Flag indicating whether the specified Local Kafka Service is enabled or not.	Yes	--command update-node-configuration --set kafka-local-enabled --value true
kafka-consolidated-enabled	Flag indicating whether the specified Consolidated Kafka Service is enabled or not. A Consolidated instance is required to copy data from other clusters (using mirror process) in case of multi CORE deployments.	Yes	--command update-node-configuration --set kafka-consolidated-enabled --value false
kafka-default-count	Number of Instances of Kafka Service (default : 1)	Yes	--command update-node-configuration --set kafka-default-count --value 1
kafka-default-basenum	Number to use to start naming the Kafka Instances (default : 0)	Yes	--command update-node-configuration --set kafka-default-basenum --value 0
kafka-local-baseclientport	Number to use to start numbering the Kafka Client Ports. If this is N, and instances.count is M, Kafka Service will use ports from N to N+M (default : 9092)	Yes	--command update-node-configuration --set kafka-local-baseclientport --value 9092
kafka-consolidated-baseclientport	Number to use to start numbering the Kafka Client Ports. If this is N, and instances.count is M, Kafka Service will use ports from N to N+M (default : 19092)	Yes	--command update-node-configuration --set kafka-consolidated-baseclientport --value 19092
kafkamirror-enabled	Flag indicating whether the Kafka Mirrors need to be started or not A Consolidated instance - both for Zookeeper and Kafka - are required to copy data from other clusters (using mirror processes) in case of multi CORE deployments. (default : false)	Yes	--command update-node-configuration --set kafkamirror-enabled --value false
kafkamirror-instances	List of string values separated by ",". Each value will be a Kafka Mirror Configuration. (default : fromRemoteToLocal, FromLocalToRemote)	Yes	--command update-node-configuration --set kafkamirror-instances --value fromRemoteToLocal, FromLocalToRemote
kafka-mirror-mirrordefault-consumer-cons_timeout	Default value for Consumer configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This should contain the entire declaration for the Kafka Mirror Consumer property (default : consumer.timeout.ms =-1)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-consumer-cons_timeout --value consumer.timeout.ms =-1

kafka-mirror-mirrordefault-consumer-part_assign	Default value for Consumer partition assignment configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This should contain the entire declaration for the Kafka Mirror Consumer property (default : partition.assignment.strategy=org.apache.kafka.clients.consumer.RoundRobinAssignor)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-consumer-part_assign --value partition.assignment.strategy=org.apache.kafka.clients.consumer.RoundRobinAssignor
kafka-mirror-mirrordefault-producer-serializer	Default value for Producer serializer configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This should contain the entire declaration for the Kafka Mirror Producer property (default : serializer.class=kafka.serializer.DefaultEncoder)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-producer-serializer --value serializer.class=kafka.serializer.DefaultEncoder
kafka-mirror-mirrordefault-producer-timeout	Default value for Producer timeout configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This should contain the entire declaration for the Kafka Mirror Producer property (default : queue.enqueueTimeout.ms=-1)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-producer-timeout --value queue.enqueueTimeout.ms=-1
kafka-mirror-mirrordefault-producer-server	Default value for Producer server configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This should contain the entire declaration for the Kafka Mirror Producer property (default : localhost:19092)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-producer-server --value localhost:19092
kafka-mirror-mirrordefault-service_config-streams	Default value for custom Kafka Mirror configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This is a command sent to the Kafka Mirror command line. (default : --num.streams=4)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-service_config-streams --value ' --num.streams=4' See Important Note above.
kafka-mirror-mirrordefault-service_config-topics	Default value for custom Kafka Mirror configuration for all the Kafka Mirrors (can be overwritten by changing the "mirrordefault" by the name of the specified mirror instance). This is a command sent to the Kafka Mirror command line. (default : --whitelist=.)	Yes	--command update-node-configuration --set kafka-mirror-mirrordefault-service_config-topics --value ' --whitelist=.*' See Important Note above.
kafka-mirror-fromlocaltolocal-consumer-server	Specific configuration for the mirror named "fromLocalToLocal", for the Consumer property called server. (default : bootstrap.servers=localhost:9092). This is where this mirror will consume the records.	Yes	--command update-node-configuration --set kafka-mirror-fromlocaltolocal-consumer-server --value bootstrap.servers=localhost:9092
kafka-mirror-fromlocaltolocal-consumer-group	Specific configuration for the mirror named "fromLocalToLocal", for the Consumer property called group. (default : group.id=MIRRORMAKER-DECISIONS1). This is a string naming the group of processes; this allows to have more than 1 mirror for resiliency purposes.	Yes	--command update-node-configuration --set kafka-mirror-fromlocaltolocal-consumer-group --value group.id=MIRRORMAKER-DECISIONS1

kafka-mirror-fromremotetolocal-consumer-server	Specific configuration for the mirror named "fromRemoteToLocal", for the Consumer property called server. (default :bootstrap.servers=remoteServer:9092 This needs to be changed to the remote server local kafka). This is where this mirror will consume records.	Yes	--command update-node-configuration --set kafka-mirror-fromremotetolocal-consumer-server --value bootstrap.servers=remoteServer:9092
kafka-mirror-fromremotetolocal-consumer-group	Specific configuration for the mirror named "fromLocalToLocal", for the Consumer property called group. (default : group.id =MIRRORMAKER-DECISIONS-REMOTE-1). This is a string naming the group of processes; this allows to have more than 1 mirror for resiliency purposes.	Yes	--command update-node-configuration --set kafka-mirror-fromremotetolocal-consumer-group --value group.id=MIRRORMAKER-DECISIONS-REMOTE-1
kafka-consumer-group-id	Id of the Kafka Consumer group	Yes	--command update-node-configuration --set kafka-consumer-group-id --value group001
kafka-consumer-db-user	DB User used by the Kafka Consumer to save decisions into the RDR table	Yes	--command update-node-configuration --set kafka-consumer-db-user --value dynamicrouting
kafka-consumer-db-pwd	DB Password used by the Kafka Consumer to save decisions into the RDR table	Yes	--command update-node-configuration --set kafka-consumer-db-pwd --value 123456
kafka-consumer-db-connection-url	JDBC Connection URL used by the Kafka Consumer to save decisions into the RDR table	Yes	--command update-node-configuration --set kafka-consumer-db-connection-url --value jdbc:postgresql://127.0.0.1:5555/dynamicrouting_rdr_db
kafka-consumer-db-connection-testquery	Test statement used by the Kafka Consumer db pool to validate DB connections	Yes	--command update-node-configuration --set kafka-consumer-db-connection-testquery --value 'select 1'
kafka-consumer-db-hibernate-dialect	Hibernate Dialect used by the Kafka Consumer to save decisions into the RDR table	Yes	--command update-node-configuration --set kafka-consumer-db-hibernate-dialect --value org.hibernate.dialect.PostgreSQL82Dialect
kafka-consumer-db-driver-class	DB vendor specific JDBC Driver Class used by the Kafka Consumer to save decisions into RDR table	Yes	--command update-node-configuration --set kafka-consumer-db-driver-class --value org.postgresql.Driver
kafka-consumer-kafka-servers	Kafka servers used by Kafka Consumer to consume from.	Yes	--command update-node-configuration --set kafka-consumer-kafka-servers --value 10.135.7.227:9092,10.135.7.228:9092,10.135.7.229:9092
kafka-local-retention-time	Local Kafka retention time. If the property is not defined, the kafka-default-retention-time is used instead . Once the value is defined, it cannot be deleted, only updated.	Yes	--command update-node-configuration --set kafka-local-retention-time --value 5h

kafka-consolidated-retention-time	Local Consolidated Kafka retention time. If the property is not defined, the kafka-default-retention-time is used instead. Once the value is defined, it cannot be deleted, only updated.	Yes	--command update-node-configuration --set kafka-consolidated-retention-time --value 2d
kafka-default-retention-time	The default retention time used by both Local and Consolidated Kafka instances. A default value is not provided, so should be configured by the user.	Yes	--command update-node-configuration --set kafka-default-retention-time --value 1h
kafka-decisions-recovery-schedule	<p>Schedule pattern (CRON) to periodically recover rejected decisions and insert them into the RDR DB. The supported patterns are:</p> <ul style="list-style-type: none"> • h - for hourly executions: 'h [1-23]' (every X hours) • d - for daily executions: 'd [0-23]' (every day at X o'clock) • w - for weekly executions: 'w [0-23] [0-7]' (every week at X o'clock on Y day; where [0-23] is the hour of the day and [0-7] is the day of the week Sunday is either 0 or 7. note: day of week can also be specified as: sun,mon,tue,wed,thu,fri,sat) • m - for monthly executions: 'm [0-23] [1-31]' (every month at X o'clock on Y day; where [0-23] is the hour of the day and [1-31] is the day of the month) <p>NOTE: This command needs to be issued on every node in which we have Kafka Consumer running, since these nodes keep track of their own list of rejected decisions.</p>	No	<pre>--command update-node-configuration --set kafka-decisions-recovery-schedule --value 'm 20' #every 20 minutes --command update-node-configuration --set kafka-decisions-recovery-schedule --value 'h 23' #every 23 hours starting when the command was run --command update-node-configuration --set kafka-decisions-recovery-schedule --value 'd 23' #every day at 11 PM --command update-node-configuration --set kafka-decisions-recovery-schedule --value 'w 0 0' #every Sunday at midnight --command update-node-configuration --set kafka-decisions-recovery-schedule --value 'M 16 28' #every month on day 28 at 4 PM --command update-node-configuration --set kafka-decisions-recovery-schedule --value 'w 1 wed' #every Wednesday at 1 AM</pre>

kafka-auditlog-recovery-schedule	<p>Schedule pattern (CRON) to periodically recover rejected Audit Log records and insert them into the audit_log table. The supported patterns are:</p> <ul style="list-style-type: none"> h - for hourly executions: 'h [1-23]' (every X hours) d - for daily executions: 'd [0-23]' (every day at X o'clock) w - for weekly executions: 'w [0-23] [0-7]' (every week at X o'clock on Y day; where [0-23] is the hour of the day and [0-7] is the day of the week Sunday is either 0 or 7. note: day of week can also be specified as: sun,mon,tue,wed,thu,fri,sat) m - for monthly executions: 'm [0-23] [1-31]' (every month at X o'clock on Y day; where [0-23] is the hour of the day and [1-31] is the day of the month) <p>NOTE: This command needs to be issued on every node in which we have Kafka Audit Log Consumer running, since these nodes keep track of their own list of rejected Audit Log records.</p>	No	<pre>--command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'm 20' #every 20 minutes --command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'h 23' #every 23 hours starting when the command was run --command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'd 23' #every day at 11 PM --command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'w 0 0' #every Sunday at midnight --command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'M 16 28' #every month on day 28 at 4 PM --command update-node-configuration --set kafka-auditlog-recovery-schedule --value 'w 1 wed' #every Wednesday at 1 AM</pre>
elastic-search-queue-size	Queue size of Elasticsearch for search (query) operations. The default value is 1000.	Yes	<pre>--command update-node-configuration --set elastic-search-queue-size --value 2000</pre>
elastic-bulk-queue-size	Queue size of Elasticsearch for bulk operations. The default value is 200.	Yes	<pre>--command update-node-configuration --set elastic-bulk-queue-size --value 400</pre>

Kafka Mirrors

As Kafka Mirrors are added or configured, this may require new properties to be configured, which are not part of the standard configuration. An override mechanism is available to add these Kafka properties.

As an example, if another mirror is needed (more than 2 clusters) or resilient mirrors are needed, new mirrors need to be configured, but their properties do not exist yet.

For Kafka Mirrors, if a property is called KAFKA-MIRROR-REMOTE3-CONSUMER-GROUP, that property will be applied only to the mirror called REMOTE3. Other properties under "MIRRORDEFAULT" name will be used too.

If a specific mirror - example: "FROMREMOTETOLOCAL" - needs to override a default property value, it can be accomplished in the following way:

The default property must be overridden: KAFKA-MIRROR-**MIRRORDEFAULT**-PRODUCER-SERIALIZER

To do so, the property with the name KAFKA-MIRROR-**FROMREMOTETOLOCAL**-PRODUCER-SERIALIZER must be set.

Cluster Manager Tool (CMT)

Avaya Business Rules Engine is delivered with default settings. However, some settings need to be overwritten by the customer. The Cluster Manager Tool (CMT) can be used to override both individual settings of each node and cluster settings. CMT can be run manually to update each property. Additionally, CMT can be used to perform certain operations.

The script to run the tool is available at [\[AVAYA_HOME\]/abre/bin](#), and this section illustrates how to use it.

```
sudo su - avayadr -c "cluster-mgr.sh --command <command>"
```

Help details are printed on the console whenever the utility is executed with no parameters.

Cluster Manager Tool Command Description and Usage

Command	Optional parameters	Mandatory parameters	Description
deploy-zone	--skip-gsc-check value	--zone value	Starts GSC and deploys PUs for specific zone, if enabled. The system checks if GSC is already running before creating a new one
undeploy-zone		--zone value	Undeploys PUs for specific zone.
create-config-db			Creates the configuration database and loads default configuration data
init-configuration			Loads both properties and default configuration data into the system
is-enabled		--service value	Checks whether a given service is enabled or not
log-level		--component value --level value	Switches the logging filter of a component to a desired level
load-demo-data			Loads the demo data into the configuration database
load-properties			Loads properties into the system
cs-replication-status			Displays current synchronization and WAN Gateway status
cs-enable-replication			Deploys WAN Gateway to Configuration Store, enabling data replication across clusters
cs-shutdown-replication			Removes the replication nature of the Avaya Business Rules Engine installation
clean-config-db			Deletes all records stored in configuration database and set cluster to perform a bootstrap during next startup.
replication-space-compare			Compares Configuration Store space content between data centers.
purge-status			Lists all the available purges run by CRON: Audit Log, RDR and Elasticsearch
rebalance-kafka-local			Attempts to re-balance current Local Kafka cluster
rebalance-kafka-consolidated			Attempts to re-balance current Consolidated Kafka cluster
backup-db			Creates a backup of the Config DB at [AVAYA_HOME]/abre/backup folder
restore-db		--file fileName	Restores the Config DB from the dump file provided. e.g --command restore-db --file /opt/Avaya/abre/backup/abre_backup_1526047931232.dump

check-db-mode			Shows the current mode (Primary or Standby) of the Config DB on the node.
http-config		--action value	<p>Enable/Disable HTTP for Avaya Business Rules Engine (otherwise use HTTPS).</p> <ul style="list-style-type: none"> • Enable --command http-config --action enable • Disable --command http-config --action disable
test-rdr-connection			Troubleshoot the RDR DB Connection (attempt DB and show privileges or provide error)
view-snmp-targets			View SNMP properties
add-snmp-target		--targetParameters values (comma separated)	<p>Add an SNMP target server for Alarms. Example: --command add-snmp-target --targetParameters version=v3, trapDestination=10.130.92.200 /10162,oid=1.3.6.1.4.1.6889.2.78, community=asmcpe,user.name=initial,user.securityName=initial,user.authenticationProtocol=MD5, user.authenticationPassword=12345678, user.privacyProtocol=DES,user.privacyPassword=12345678</p> <p>Note: targetId should not be supplied</p>
update-snmp-target		--targetId value --targetParameters value (comma separated)	<p>Update one or more properties of an SNMP target server</p> <p>Example: --command update-snmp-target --targetId 1 --targetParameters trapDestination=10.130.92.200 /10162,oid=1.3.6.1.4.1.6889.2.78, community=asmcpe, user.name=initial,user.securityName=initial,user.authenticationProtocol=MD5, user.authenticationPassword=12345678, user.privacyProtocol=DES,user.privacyPassword=12345678"</p> <p>Note: protocol version cannot be updated</p>
delete-snmp-target		--targetId value	<p>Delete an SNMP target server</p> <p>Example: --command delete-snmp-target --targetId 1</p>
restore-snmp-properties			Restore previously saved snmp.properties file when upgrading ABRE release

update-memory		--component component_name --memory memory_type --amount value	<p>Updates either the minimum or maximum amount of memory dedicated to a given component.</p> <p>Supported values for the parameter component are:</p> <ul style="list-style-type: none"> • elastic • kafka • logstash • zookeeper <p>Supported values for the parameter memory are min or max.</p> <p>Value assigned to the parameter amount must follow the pattern: <any _integer>[g G m M k K] (examples: 9g, 256m)</p> <p>Example:</p> <pre>--command update-memory -- component kafka --memory min -- amount 1024M" --command update-memory -- component kafka --memory max -- amount 1024M"</pre>
update-abre-services-memory			<p>Updates the memory used by ABRE's services according to the amount of memory in the host.</p> <p>Example:</p> <pre>--command update-abre-services- memory</pre>
show-metrics-adjustment-configuration			<p>Displays all current Metrics Adjustment configurations</p> <pre>--command show-metrics- adjustment-configuration</pre>
update-metrics-adjustment-expression		--metric-name --adjustment-expression	<p>Adds or updates a mathematical expression to be used for Metrics Auto-adjustment</p> <p>metric-name - is the name of the metric configured in the CMS Connector's cmsReportConfiguration.xml file e.g. AVAILABLE, EWTHIGH, EWTMEDIUM, CALLSOFFERED, EWTLOW</p> <p>adjustment-expression - is a mathematical expression used to calculate the metric adjustment and apply it to the metric after each decision request. Any existing metric can be used as an argument in the expression. All basic operators are supported (+, -, /, *, ^), also a number of complex operations can be used (see http://mathparser.org) e.g. 'AVAILABLE - 1', 'EWTMEDIUM + WAT'. WAT is provided by the CMS, and used to adjust EWT.</p> <pre>--command update-metrics- adjustment-expression --metric- name AVAILABLE --adjustment- expression 'AVAILABLE - 1'</pre>
remove-metrics-adjustment-expression		--metric-name	<p>Removes a Metrics Adjustment expression for a metric.</p> <pre>--command remove-metrics- adjustment-expression --metric- name AVAILABLE</pre>

Cluster Manager Tool Command Parameters

Parameter	Description
-----------	-------------

--component value	<p>Name of the component whose log level is about to be changed. Valid values are:</p> <p>context-store : Context Store execution</p> <p>configstore : Configuration Store</p> <p>configstore-cluster-mgr : Cluster Manager operations on Configuration Store</p> <p>db-utils</p> <p>lb : load-balancer messages</p> <p>global</p> <p>metrics : metrics service</p> <p>rt-metrics : real-time metrics operation</p> <p>routing : routing service messages</p> <p>routing-scripts : scripts used to calculate routing / make decisions</p> <p>routing-comm : routing communication listeners</p> <p>routing-business : business operations on routing service</p> <p>routing-core : routing core operations</p> <p>web-admin: Web Admin operation</p> <p>admin-api : Admin API execution</p> <p>rt-mon : real-time monitoring messages at runtime</p> <p>wlmconn : web license manager connectivity</p> <p>weblm : web licensing manager general messages</p> <p>weblm-audit : audit messages from weblm</p> <p>weblm-oper : operation messages from weblm</p> <p>weblm-security : security messages from weblm</p> <p>audit : Avaya Business Rules Engine generic audit messages</p> <p>audit-log : audit log processing unit</p> <p>alarm : messages related to alarms raised by Avaya Business Rules Engine</p> <p>kafka : kafka persistence messages</p> <p>rejected-decisions : messages indicating rejected decision at runtime</p> <p>cluster-mgr: Cluster Manager operation</p> <p>logstash: Logstash, applicable only with the command update-memory</p> <p>elastic: ElasticSearch, applicable only to the command update-memory</p>
--help	Displays help and exits
--mirrorName value	Name of the mirror to start/stop

--level value	<p>Log level to be applied to a certain component. Possible values are:</p> <p>TRACE : informational events of very low importance</p> <p>INFO : informational messages highlighting overall progress of the application</p> <p>DEBUG : informational events of lower importance, more relevant for troubleshooting purposes</p> <p>WARN : potentially harmful situations</p> <p>ERROR : error events which may or not be fatal to the application</p> <p>OFF : to turn logging off</p>
--service value	<p>Name of a Avaya Business Rules Engine service. Valid values are:</p> <p>mgm : the management zone</p> <p>core : the core zone</p> <p>zookeeper: ZooKeeper service, used to orchestrate the various, distributed Kafka instances in the cluster</p> <p>kafka: Kafka, the stream processing platform used to handle Avaya Business Rules Engine's real-time data feeds</p> <p>elastic: Elasticsearch engine</p>
--skip-confirmation	Skip the confirmation message on properties to which restart is required
--skip-gsc-check value	<p>Values: true false.</p> <p>When set to true, zone deployment will always spawn a new GSC, even if one is already running.</p>
--zone value	Which zone to manage. Valid values are mgm and core
--memory	The memory limit to be updated, for a particular component (acceptable values are min or max)
--amount	Amount of memory to be dedicated to a particular component

Using Avaya Business Rules Engine

Your main interfaces as an Avaya Business Rules Engine user are:

- Administer the Avaya Business Rules Engine entity configuration using **Web Admin**
 - Refer to the Avaya Business Rules Engine Administration Guide
- Send routing requests to the Avaya Business Rules Engine using **Decision API (REST)**
 - Refer to the Avaya Business Rules Engine Integration Guide
- Administer the configuration data via **Admin API (REST)**
 - Refer to the Avaya Business Rules Engine Integration Guide

Appendix A - (Optional) Importing Demo Data

When using Avaya Business Rules Engine in the lab, or for Demo purposes, it is helpful to rely on a user-friendly sample configuration.

This utility provides end to end configuration of a fictitious company called ***Alive***, with the following characteristics:

- More than 10 sample call segments for 3 packages (products): Basic, Family, and Premium
- More than 40 sample Agent Groups across 10 ACDs, 10+ Locations and 4 Companies.
- Examples of both in-house ACDs (A1, A2 ... A4 at Alive), and 6 external ACDs in fictitious (outsourced) companies: Teleperformance, Accenture and Blue.

The following block shows how to run this step. To do so, run the command:

```
sudo -u avayadr /opt/Avaya/abre/bin/cluster-mgr.sh --skip-confirmation true --command load-demo-data
```

Load Demo Data

- This step can be **only** executed the **first time** you install Avaya Business Rules Engine
- The above load command has to be executed after starting the services
- Execute this import on **only one of the nodes that runs Management Services**
- Inspect Demo Data using the Web Admin

Additionally, you can import demo data into one of the tenants. In order to do so you have to run the command this way:

```
sudo -u avayadr /opt/Avaya/abre/bin/cluster-mgr.sh --skip-confirmation true --command load-demo-data --tenant tenant_full_name
```

Note that:

Load Demo Data

- You should first create the tenant, into which you want to upload the demo data
- If you want to load demo data to an LoB, you have to provide the full Name "parent_tenant_name/LoB_name"
- A Tenant and it's LoBs share a common pool of entities names, so you cannot load the demo data into both the Tenant and its LoBs.

Appendix B - Security Enable https

In order to tighten security, http can be disabled and https enforced instead.

Procedure:

1. Run the following CMT command
 1. **sudo su - avayadr -c "cluster-mgr.sh --command http-config --action <ACTION>"**, where action is either enable or disable
 2. To enable http (disable https)
 1. **sudo su - avayadr -c "cluster-mgr.sh --command http-config --action enable"**
 3. To disable http (enable https)
 1. **sudo su - avayadr -c "cluster-mgr.sh --command http-config --action disable"**

Note: The http-config CMT command restarts the NGINX service, so there is no need to restart it.

Appendix C - Replacing SSL certificates

If you want to change the SSL certificate and key, follow the procedure below.

Procedure:

1. Put the new certificate and private key file(s) in a directory of your preference
2. Open the file `/etc/nginx/conf.d/abre-ssl.conf` for editing
3. Change the property `ssl_certificate`, to point to the full path of the file that contains the new SSL certificate (e.g.: `/etc/nginx/ssl/new-abre-cert.crt`)
4. Change the property `ssl_certificate_key`, to point to the full path of the file that contains the new SSL private key (e.g.: `/etc/nginx/ssl/new-abre-key.key`)
5. Save the file
6. Restart NGINX (`systemctl restart nginx`)

Important notes

- It is not mandatory to keep certificates and private keys in separate files (although it is a good practice)
- Usually, the certificates and keys used by NGINX in Avaya Business Rules Engine are saved under directory `/etc/nginx/ssl`. It is recommended, although not mandatory, that you save your new files in this directory

Appendix D - Avaya Business Rules Engine Services

Although to start/stop Avaya Business Rules Engine services the user only needs to run `abre-services`, each service can be started/stopped manually following the commands below:

- `systemctl start zookeeper [start | stop | restart | status]` Zookeeper
- `systemctl start kafka-instance@local [start | stop | restart | status]` Kafka Local
- `systemctl start kafka-instance@consolidated [start | stop | restart | status]` Kafka Consolidated
- `systemctl start kafka-instance-mirror@fromLocalToLocal [start | stop | restart | status]` Kafka mirror instance fromLocalToLocal
- `systemctl start kafka-instance-mirror@fromRemoteToLocal [start | stop | restart | status]` Kafka mirror instance fromRemoteToLocal
- `systemctl start logstash [start | stop | restart | status]` Logstash
- `systemctl start elastic [start | stop | restart | status]` Elasticsearch
- `systemctl start dr-postgres [start | stop | restart | status]` PostgreSQL

Appendix E - Disable TLS v1.1

The TLS v1.2 and TLS v1.1 are supported by default. If you want to disable TLS v1.1, please follow the procedure below.

Procedure:

1. Open the file `/etc/nginx/conf.d/abre-ssl.conf` for editing
2. Remove "TLSv1.1" from the line `ssl_protocols TLSv1.2 TLSv1.1;`
3. Save the file
4. Restart NGINX (`systemctl restart nginx`)

Avaya Business Rules Engine 3.4 Configuring Internal Firewall - IPTables

To secure the nodes (servers) using the internal Linux firewall (IPTables), Avaya Business Rules Engine provides a script to configure the communication between the nodes that are part of the Avaya Business Rules Engine cluster and also the communication with the external environment. The user should run this script on **EACH** Avaya Business Rules Engine nodes of the cluster, manually.

The IPTables configuration script ([configureIPTables.sh](#)) can be found in the installation ISO file under [/Tools](#) and:

- Performs backup of the current IPTables configuration
- Prompts the user for input to guide the configuration, such as:
 - whether to keep port 80 open to the external environment
 - Port 443 is the default port provided for Administration, APIs and Routing Requests, but the user can also use port 80 if required
 - the entire list of IP Addresses for the Avaya Business Rules Engine Cluster, including Core Services, Management Services, Messaging Services, Aggregation Services, Reporting Services, and CMS Connectors
 - whether the server where the script is being executed will receive data from the CMS
 - If the answer is yes, the script opens port 7200 to receive data from the external environment. If the CMS will send the feed via a different port, the firewall should be manually configured, to open the desired port
- Confirms the user input and rebuilds the IPTables based on this input

IMPORTANT

- This script **flushes** the current IPTables configuration. After script execution, only the Avaya Business Rules Engine input will remain on the node.
- The list of HTTP enablement ports and IP Addresses **should be the same on all the Avaya Business Rules Engine nodes**. The only variation may be regarding the CMS feed, where the user should enable this option only on the servers that will receive data from a CMS.
- Whenever port 80 is closed to the external environment, Web Admin will no longer be accessible via HTTP. The user must use **HTTPS** instead.

Script Execution - Example

This example configures the Avaya Business Rules Engine Cluster deployed as 3 Nodes in 2 Data Centers with 1 CMS Connector per Data Center, disabling the HTTP port:

- Data Center 1
 - Nodes
 - 192.168.0.2
 - 192.168.0.3
 - 192.168.0.4
 - CMS Connector
 - 192.168.0.5
- Data Center 2
 - Nodes
 - 192.168.1.2
 - 192.168.1.3
 - 192.168.1.4
 - CMS Connector
 - 192.168.1.5

Executing in the Avaya Business Rules Engine Nodes (192.168.0.2, 192.168.0.3, 192.168.0.4, 192.168.1.2, 192.168.1.3, 192.168.1.4)

```
[root@dc1nd1 ~]# ./configureIPTables.sh
1) All ports open between the servers, only http(s) and ssh open to
external network
2) Exit
Please, select the option: 1
This option will clean the current iptables configuration and add the
Avaya Business Rules Engine IPTables required configuration
Confirm the input? (y/n)?y
  - Performing backup of the current iptables configuration. Backup file:
/etc/sysconfig/iptables.20171212170313
All ports open between the servers, only http(s) and ssh open to external
network
  -- Leave http port 80 open(non secure) open? (y/n): n
  -- Enter the IPs of all servers, including the ones in remote Data
Centers and all CMS Connectors (comma separated):
192.168.0.2,192.168.0.3,192.168.0.4,192.168.0.5,192.168.1.2,192.168.1.3,192
.168.1.4,192.168.1.5
  -- Leave the 7200 port open to receive CMS Feed (Agent Group Metrics) in
this server? (y/n): n
The script is about to apply the changes on this server.
Confirm the input? (y/n)?y
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Unloading modules: [ OK ]
iptables: Applying firewall rules: [ OK ]
```

Executing in the CMS Connector Nodes (192.168.0.5, 192.168.1.5)

```

[root@dclcmsconn1 ~]# ./configureIPTables.sh
1) All ports open between the servers, only http(s) and ssh open to
external network
2) Exit
Please, select the option: 1
This option will clean the current iptables configuration and add the
Avaya Business Rules Engine IPTables required configuration
Confirm the input? (y/n)?y
  - Performing backup of the current iptables configuration. Backup file:
/etc/sysconfig/iptables.20171212170313
All ports open between the servers, only http(s) and ssh open to external
network
  -- Leave http port 80 open(non secure) open? (y/n): n
  -- Enter the IPs of all servers, including the ones in remote Data
Centers and all CMS Connectors (comma separated):
192.168.0.2,192.168.0.3,192.168.0.4,192.168.0.5,192.168.1.2,192.168.1.3,192
.168.1.4,192.168.1.5
  -- Leave the 7200 port open to receive CMS Feed (Agent Group Metrics) in
this server? (y/n): y
The script is about to apply the changes on this server.
Confirm the input? (y/n)?y
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]
iptables: Setting chains to policy ACCEPT: filter [ OK ]
iptables: Flushing firewall rules: [ OK ]
iptables: Unloading modules: [ OK ]
iptables: Applying firewall rules: [ OK ]

```

Enabling other CMS Connector Ports

If the CMS will send the feed to a port other than 7200, the user should run this command on the CMS Connectors that will receive the feed:

```
iptables -A INPUT -p tcp -m tcp --dport <PORT> -j ACCEPT
```

Note:

Change the **<PORT>** to the desired port number.

Avaya Business Rules Engine 3.4 Post Installation and Configuration Sanity Test Checklist

The purpose of this checklist is to perform a quick sanity-test of Avaya Business Rules Engine, after an installation or maintenance procedure.

Pre-requisites: Avaya Business Rules Engine has been successfully installed and configured. All services are up & running (for the topology selected).

1) **Log into Web Admin.** Go to the Web Admin Home page.

- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ (For first login only) Enter the user/password from init data (admin/Avaya123)
- ☐ (For first login only) You are asked to change the password to a new one that complies with security standards.
- ☐ Enter the new user/password. You are logged into Web Admin successfully.

2) **Create a Location.** Go to the Location option on the left pane, and click on the '+' button to create a new Location entity.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter **testLocation** for both the Name and Description fields.
- ☐ Click on the Save button. The entity is created.

3) **Create a Company.** Go to the Company option on the left pane, and click on the '+' button to create a new Company entity.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter **testCompany** for both the Name and Description fields.
- ☐ Click on the Save button. The entity is created.

4) **Create an ACD.** Go to the ACD option on the left pane, and click on the '+' button to create a new ACD entity.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter "1" for the Native ID field
- ☐ Enter **testACD** for both the Name and Description fields
- ☐ Click the check-box "Metrics Available"
- ☐ Click on the Save button. The entity is created.

5) **Create an Agent Group.** Go to the Agent Group option on the left pane, and click on the '+' button to create a new Agent Group entity.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter **testAG** for the Agent Group Name field
- ☐ Enter "1" for the Native ID field
- ☐ Select the available ACD value from the combo-box
- ☐ Select the available Company value from the combo-box
- ☐ Select the available Location value from the combo-box
- ☐ Enter testAddress for Address field
- ☐ Click on the Save button. The entity is created.

6) **Create a Segmentation Table.** Go to the Segmentation Table option on the left pane, and click on the '+' button to create a new Segmentation Table entity.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter **testSegTable** for both the Name and Description fields
- ☐ Add a Dimension Name "Dim1" and select DataType "String".
- ☐ Enter "1" for the Min Destinations Allowed field
- ☐ Enter "10" for the Max Destinations Allowed field
- ☐

- ☐ Check radio-button "Display Destination Count"
- ☐ Check radio-button "No Strategy"
- ☐ Check radio-button "Row Priority Mode"
- ☐ Click on the Save button. The entity is created.

7) **Create a Segmentation Rule.** Go to the Segmentation Table option on the left pane, and select the Segmentation Table you created in step 6. Click on the '+' button to create a new Segmentation Rule entity, associated with that Segmentation Table.

- ☐ A pop-up window with empty fields to populate for the new entity is opened.
- ☐ Enter **testRule** for the Name field.
- ☐ Enter "Value1" for the Value field (for Dimension Name "Dim1")
- ☐ Click on the Save button. The entity is created.

8) **Add a Destination to a SegRule.** Click on the Destination field for the Segmentation Rule you created in step 7.

- ☐ A pop-up window is opened to select the Destination for that SegRule.
- ☐ Select the Agent Group you created in step 5 "testAG" for Destination.
- ☐ Click on the Save button. The Destination is associated with the Segmentation Rule.

9) **Send a Decision Request.** We are using the Decision Tracer, in this example, but you can also use the Decision API or jMeter.

- ☐ Go to the Decision Tracer option on the left pane.
- ☐ Check radio-button "Track Decision" (to register and count the decision request)
- ☐ Select "Standard_DF" in the Decision Function combo-box.
- ☐ Select "testSegTable" in the Segmentation Table combo-box.
- ☐ At the Segmentation Attributes, you shall see the Destination Name "Dim1"
- ☐ Enter "Value1" as the attribute value.
- ☐ Click on the "Send Test Request" button
- ☐ You shall have Result Code 10, which means the Destination was reached OK.
- ☐ You shall see the Segmentation Information (Segmentation Table and Segmentation Rule)
- ☐ You shall see the Selected Destination (Name, Address, Company, Location)
- ☐ You can click again on the "Send Test Request" button, to perform several decision requests to the same Destination.

10) **Check Counters for the Segmentation Table.** After performing the decision requests, go to Segmentation Table.

- ☐ The Segmentation Table used for the decision requests has proper values for counters "Last 30 Min Traffic" and "Today Traffic" (counts how many times the Segmentation Table was reached)
- ☐ Those requests older than 30 min will be cleaned from the counter "Last 30 Min Traffic".
- ☐ Select the Segmentation Table used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached all the Segmentation Rules in that Segmentation Table.

11) **Check Traffic for Segmentation Rule.** After performing the decision requests, go to Segmentation Table. Select the Segmentation Table used for the decision requests to access its Segmentation Rules. Select the Segmentation Rule used for the decision requests.

- ☐ Select the Segmentation Rule used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached all the Destinations in that SegRule.

12) **Check Traffic for Agent Group.** After performing the decision requests, go to the Agent Group option on the left pane.

- ☐ Select the AG used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached that Destination.

13) **Check Traffic for Companies.** After performing the decision requests, go to Companies option on the left pane.

- ☐ Select the Company used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached that Company.

14) **Check Traffic for Locations.** After performing the decision requests, go to Locations option on the left pane.

- ☐ Select the Location used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached that Location.

15) **Check Traffic for ACDs.** After performing the decision requests, go to ACDs option on the left pane.

- ☐ Select the ACDs used for the decision requests and click check the radio-button "Traffic"
- ☐ You shall see time-line graphics and table details of the decision requests that reached that ACD.

16) **Check Metrics for Agent Groups.** After performing the decision requests, go to Agent Group option on the left pane.

- ☐ Select the AG used for the decision requests and click check the radio-button "Metrics"
- ☐ You shall see time-line graphics for metrics available.

17) **Check Replication.** After adding the entities, open Web Admin on the second Data Center and check:

- ☐ All new elements were replicated to the second Data Center
- ☐ The same requests made to Data Center 1 also work on Data Center 2
- ☐ Repeat the steps 1-15 on a Management Server installed with the second Config DB

18) **Check RDRs.** After sending decision requests:

- ☐ Based on RDR Reference documentation, check if the RDR for the requests were generated on the configured RDR DB

19) **Confirm the System Property *maxTransactionsPerHour* is set correctly.**

- ☐ The System Property (SP) *maxTransactionsPerHour* defines the max traffic (N x 1000 requests/hour) that ABRE will process. Each Data Center acquires the traffic bandwidth prescribed by the SP from the license server (license must be \geq SP). Details in the ABRE Admin Guide.

20) **Confirm ABRE Alarms are being forwarded to an SNMP server**

- ☐ Alarms should be forwarded to System Manager or similar SNMP server. Details in the ABRE Integration Guide for the configuration required.

21) **Confirm that the application that is sending requests to BRE has error handling in place**

- ☐ Should the BRE not respond to a Decision Request, there should be error handling in the application calling BRE, to route the call to a default destination.

IMPORTANT:

If any of the tests fails, please read the Avaya Business Rules Engine Maintenance Guide.

Avaya Business Rules Engine Load Balancer Sanity Test Checklist

The purpose of this checklist is to perform a quick sanity-test of the load balancer configuration .

Pre-requisites:

- Avaya Business Rules Engine has been successfully installed and configured. All services are up and running (for the topology selected).
- Customer provided Load Balancer configured
- To guide the process, select one of the Data Centers to be the Data Center 1 and the other one to be the Data Center 2

Before start:

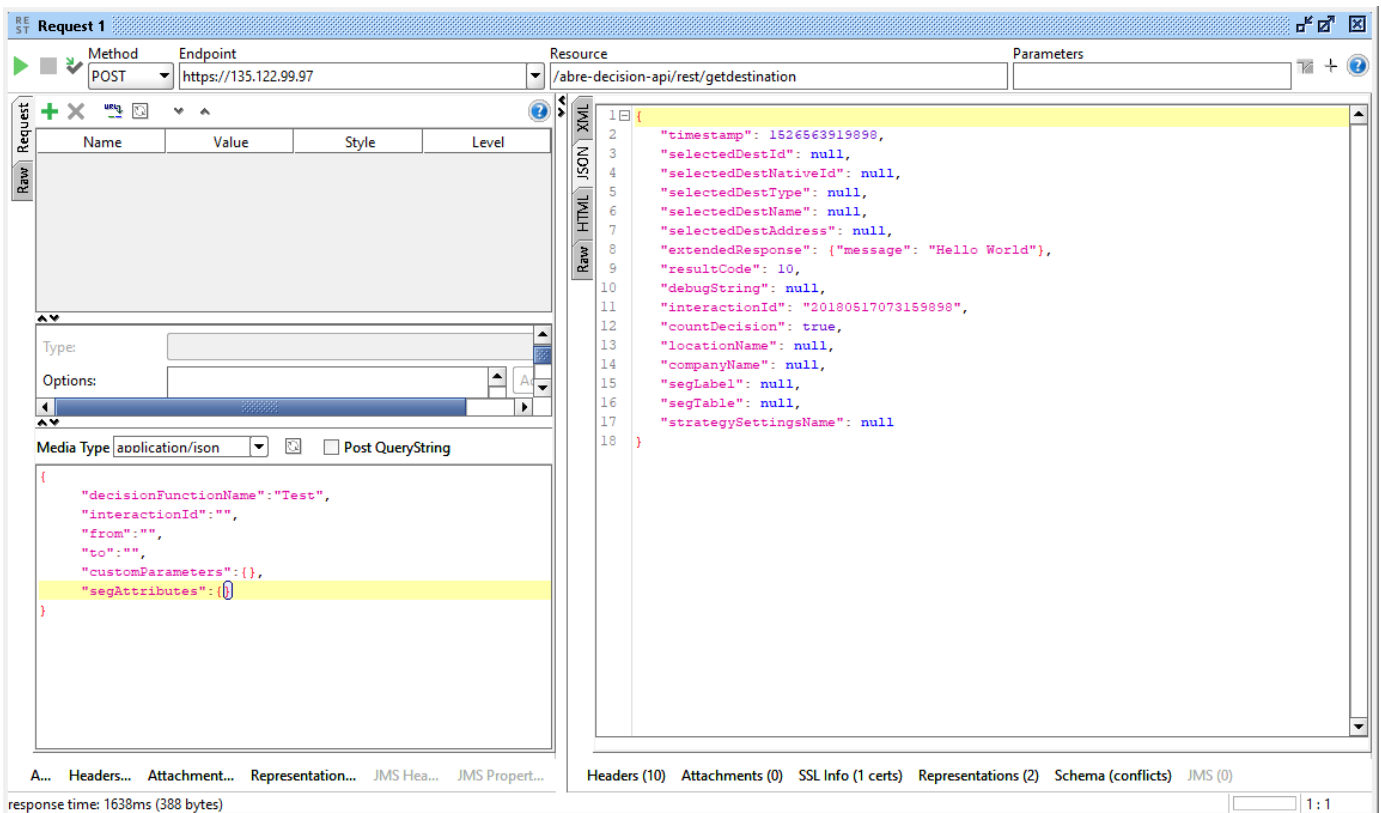
To simulate the service down in the Avaya Business Rules Engine servers, the service nginx should be stopped/started. Stopping this service, the server will not accept requests anymore.

To stop/start the service:

- Stop: systemctl stop nginx
- Start: systemctl start nginx

The tests will require to send some route requests to Avaya Business Rules Engine.

The user can use commercial tools to send the HTTP REST request to Avaya Business Rules Engine. The example below uses SoapUI to place the requests:



The expected successful response is also illustrated into the screenshot above.

Depending on the installed topology, the load balancer configuration should be different. This checklist will describe the tests to be executed for each deployment type.

Note: Some deployment types do not require external load balancer.

Deployment: 2 Data Centers - 3 Nodes per Data Center

This deployment type supports N-1 failure. If one Data Center loses one single server, nothing should be done and the two remaining servers of that specific Data Center will keep working with no impact. If two servers of the same Data Center stop to work, then the load balancer should move the entire traffic to the other Data Center.

When the Data Center has only one server working on there, that server should be "disabled" by the load balancer. One single server in the data center cannot handle all the traffic/features.

Administration Virtual IP/Hostname - All servers up and running

1) **Open the Avaya Business Rules Engine Web Admin** using the address: https://<virtual_ip_or_hostname/>abre-admin/

- ☐ The load balancer is showing all services up and running
- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ The web requests are going to the Data Center 1

Administration Virtual IP/Hostname - Simulating one server in Data Center 1 failure (Data Center 2 - 100% up and running)

1) **Open the Avaya Business Rules Engine Web Admin** using the address: https://<virtual_ip_or_hostname/>abre-admin/

- ☐ The load balancer is showing one server in Data Center 1 with the Administration Services down
- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ The web requests are going to the Data Center 1

Administration Virtual IP/Hostname - Simulating two servers in Data Center 1 failure (Data Center 2 - 100% up and running)

1) **Open the Avaya Business Rules Engine Web Admin** using the address: https://<virtual_ip_or_hostname/>abre-admin/

- ☐ The load balancer is showing two servers in Data Center 1 with the Administration Services down
- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ The web requests are going to the Data Center 2. The Data Center 1 should not receive requests anymore, even having one server up

Administration Virtual IP/Hostname - Simulating one server in Data Center 2 failure (Data Center 1 - 100% up and running)

1) **Open the Avaya Business Rules Engine Web Admin** using the address: https://<virtual_ip_or_hostname/>abre-admin/

- ☐ The load balancer is showing one server in Data Center 2 with the Administration Services down
- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ The web requests are going to the Data Center 2

Administration Virtual IP/Hostname - Simulating two servers in Data Center 2 failure (Data Center 1 - 100% up and running)

1) **Open the Avaya Business Rules Engine Web Admin** using the address: https://<virtual_ip_or_hostname/>abre-admin/

- ☐ The load balancer is showing two servers in Data Center 2 with the Administration Services down
- ☐ The Login page appears. You are asked to enter User and Password.
- ☐ The web requests are going to the Data Center 1. The Data Center 2 should not receive requests anymore, even having one server up

Routing Requests Virtual IP/Hostname - All servers up and running

1) **Place a decision request using the virtual IP/hostname**

- ☐ The load balancer is showing all services up and running
- ☐ Place a routing request and check if the web requests are going to the Data Center 1

Routing Requests Virtual IP/Hostname - Simulating one server in Data Center 1 failure (Data Center 2 - 100% up and running)

1) Place a decision request using the virtual IP/hostname

- ☐ The load balancer is showing one server in Data Center 1 with the Routing Request Services down
- ☐ Place a routing request and check if the web requests are going to the Data Center 1

Routing Requests Virtual IP/Hostname - Simulating two servers in Data Center 1 failure (Data Center 2 - 100% up and running)

1) Place a decision request using the virtual IP/hostname

- ☐ The load balancer is showing two servers in Data Center 1 with the Routing Request Services down
- ☐ Place a routing request and check if the web requests are going to the Data Center 2

Routing Requests Virtual IP/Hostname - Simulating one server in Data Center 2 failure (Data Center 1 - 100% up and running)

1) Place a decision request using the virtual IP/hostname

- ☐ The load balancer is showing one server in Data Center 2 with the Routing Request Services down
- ☐ Place a routing request and check if the web requests are going to the Data Center 2

Routing Requests Virtual IP/Hostname - Simulating two servers in Data Center 2 failure (Data Center 1 - 100% up and running)

1) Place a decision request using the virtual IP/hostname

- ☐ The load balancer is showing two servers in Data Center 2 with the Routing Request Services down
- ☐ Place a routing request and check if the web requests are going to the Data Center 1

IMPORTANT:

If one of the tests fail, please read the Avaya Business Rules Engine Planning Guide.