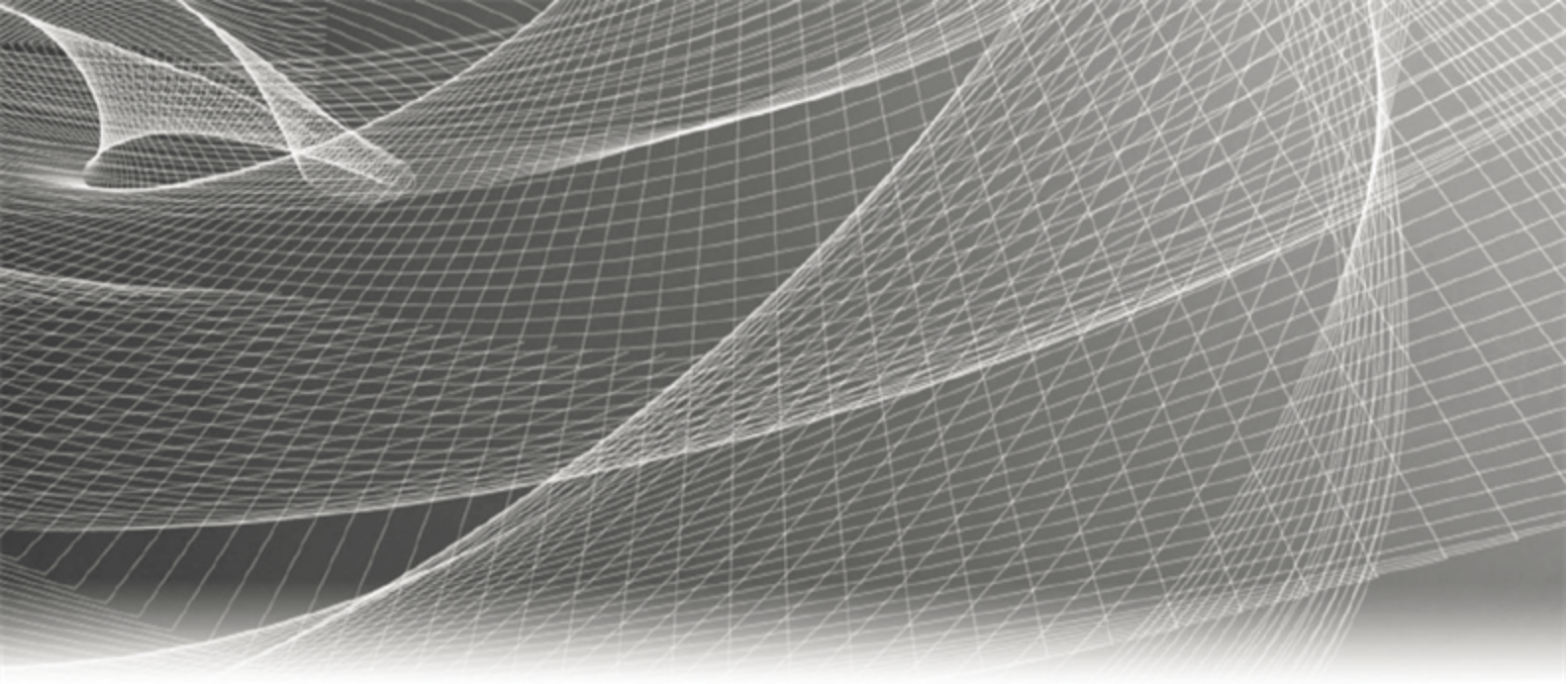

RSA IDENTITY GOVERNANCE AND LIFECYCLE

Configure WildFly Cluster to Use TCP

V7.1

Contact Information

RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

Dell, RSA, the RSA Logo, EMC and other trademarks, are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be trademarks of their respective owners. For a list of RSA trademarks, go to www.emc.com/legal/emc-corporation-trademarks.htm#rsa.

License agreement

This software and the associated documentation are proprietary and confidential to Dell Inc. or its subsidiaries, are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by Dell Inc.

Third-party licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed by launching the RSA Identity Governance and Lifecycle product and selecting the About menu. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on encryption technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any Dell software described in this publication requires an applicable software license.

Dell Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." DELL INC. MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

October 2018

Contents

Preface

Documentation Set

The latest product documentation is always available at https://community.rsa.com/community/products/governance-and-lifecycle.

Document	Description
Release Notes	What's new in the release, fixed issues, known issues and workarounds.
Installation Guide	Product installation instructions.
Upgrade and Migration Guide	Instructions for upgrading your product version and data.
Database Setup and Management Guide	Instructions for setting up and managing a customer-supplied Oracle database for RSA Identity Governance and Lifecycle.
Online Help	All concepts and instructions you need to configure and use the product.
Public Database Schema Reference	The public view of the database schema.

Support and Service

You can access community and support information on RSA Link at https://community.rsa.com/community/products/governance-and-lifecycle. RSA Link contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

You can also access the RSA Identity Governance and Lifecycle community at https://community.rsa.com/community/products/governance-and-lifecycle/client-partner-community. This private community is only available to RSA Identity Governance and Lifecycle customers, partners, and internal RSA staff.

Migrating a WildFly Cluster to TCP

By default, WildFly clusters in RSA Identity Governance and Lifecycle 7.1 use multicast for communication between the nodes. If needed, you can modify a WildFly cluster environment to use TCP instead of multicast.

You should only configure your 7.1 deployment to use TCP instead of multicast if it is appropriate for your environment and network.

Edit the Aveksa Cluster File

To configure the cluster to use TCP instead of multicast, you must modify the **aveksa_cluster** file.

Procedure

1. In the **/etc/init.d/** directory, open the **aveksa_cluster** file for editing.
2. On each node in the cluster, add the JBoss bind IP address option to the start-up options:
   a. Look for the following block of text:

```
if [ "${NODE_TYPE}" == "DOMAIN" ] ; then
echo This system will start in Domain mode
CLUSTER_COMMAND_STRING="${AVEKSA_WILDFLY_
HOME}/bin/domain.sh -b $(hostname -i) \
-Djboss.bind.address.management=$(hostname -i) \
-Djboss.messaging.group.address=${JMS_MULTICAST_IP} \
${JAVA_OPTS}"
else
echo This system will start in Slave mode
CLUSTER_COMMAND_STRING="${AVEKSA_WILDFLY_
HOME}/bin/domain.sh -b $(hostname -i) \
-Djboss.domain.master.address=${DOMAIN_MASTER} \
-Djboss.bind.address.management=$(hostname -i) \
-Djboss.messaging.group.address=${JMS_MULTICAST_IP} \
${JAVA_OPTS}"
fi
```

   b. Edit this block to include the following bolded lines:

```
if [ "${NODE_TYPE}" == "DOMAIN" ] ; then
echo This system will start in Domain mode
CLUSTER_COMMAND_STRING="${AVEKSA_WILDFLY_
HOME}/bin/domain.sh -b $(hostname -i) \
-Djboss.bind.address.management=$(hostname -i) \
-Djboss.messaging.group.address=${JMS_MULTICAST_IP} \
-Djboss.bind.address.private=${DOMAIN_MASTER} \
${JAVA_OPTS}"
else
echo This system will start in Slave mode
CLUSTER_COMMAND_STRING="${AVEKSA_WILDFLY_
HOME}/bin/domain.sh -b $(hostname -i) \
-Djboss.domain.master.address=${DOMAIN_MASTER} \
-Djboss.bind.address.management=$(hostname -i) \
-Djboss.messaging.group.address=${JMS_MULTICAST_IP} \
-Djboss.bind.address.private=${SLAVE_HOST} \
```

```
${JAVA_OPTS}"
fi
```

3. On the host controller only, add the SLAVE_HOST variable with a value of the IP address of the host controller.
   For example:

   `SLAVE_HOST="HostControllerIP"`

   Where *HostControllerIP* is the IP address of the host controller.

Update the Firewall Configuration

For WildFly messaging in a cluster setup to communicate successfully using TCP, you must add the required TCP ports and remove the unneeded UDP ports.

Procedure

1. In a SuSE environment, edit /etc/sysconfig/SuSEfirewall2.
2. Add TCP ports 7600 and 57600 to the operating systems firewall configuration using the FW_SERVICES_EXT_TCP variable. For example:

```
FW_SERVICES_EXT_TCP="21 22 1158 1555 8080 8443 8444 9999 7600
57600"
```

3. Remove the UDP ports 9876, 45688 and 55200 from the list of ports opened in firewall configuration under the FW_SERVICES_EXT_UDP variable. For example:

```
FW_SERVICES_EXT_UDP=""
```

Next steps

After configuring the ports, restart the firewall to apply the changes.

Edit Multicast References in domain.xml

You must edit the **domain.xml** file to use TCP instead of multicast.

Procedure

1. Open the file **/home/oracle/wildlfly/domain/configuration/domain.xml** for editing.
2. Change the default communication stack from UDP to TCP by replacing all references of <channel name="ee" stack="udp"/> to <channel name="ee" stack="tcp"/>.
3. Change the JMS messaging group configuration:
   a. Search for the broadcast-group section, which contains the following text:

      ```
      <broadcast-group name="bg-group1" jgroups-channel="activemq-cluster" connectors="http-connector"/> <broadcast-group name="aveksa-broadcast-group" socket-binding="messaging-group" connectors="http-connector"/> <discovery-group name="dg-group1" jgroups-channel="activemq-cluster"/> <discovery-group name="aveksa-discovery-group" socket-binding="messaging-group"/>
      ```

   b. Remove the default broadcast-group with the name bg-group1 and the default discover-group with the name dg-group1 from this text block by deleting the following lines:

      <broadcast-group name="bg-group1" jgroups-channel="activemq-cluster" connectors="http-connector"/>

      and

      <discovery-group name="dg-group1" jgroups-channel="activemq-cluster"/>

   c. Replace the broadcast-group section with the following text, removing the reference to messaging-group which uses multicast and default groups in the system. Changes from the original text are bold:

      ```
      <broadcast-group name="aveksa-broadcast-group" jgroups-channel="tcp" connectors="http-connector" jgroups-stack="tcp"/> <discovery-group name="aveksa-discovery-group" jgroups-channel="tcp" jgroups-stack="tcp"/>
      ```

4. Include TCPPING for discovery:

   a. Search for the following content, which appears twice:

```
<stack name="tcp">
<transport type="TCP" socket-binding="jgroups-tcp"/>
<protocol type="MPING" socket-binding="jgroups-mping"/>
<protocol type="MERGE3"/>
<protocol type="FD_SOCK" socket-binding="jgroups-tcp-fd"/>
<protocol type="FD"/>
<protocol type="VERIFY_SUSPECT"/>
<protocol type="pbcast.NAKACK2"/>
<protocol type="UNICAST3"/>
<protocol type="pbcast.STABLE"/>
<protocol type="pbcast.GMS"/>
<protocol type="MFC"/>
<protocol type="FRAG2"/>
</stack>
```

   b. Comment-out the MPING section and add the TCPPING property as shown in bold in the following:

```
<stack name="tcp">
<transport type="TCP" socket-binding="jgroups-tcp"/>
<protocol type="TCPPING">
<property name="initial_hosts">${jboss.cluster.tcp.initial_hosts}
</property>
</protocol>
<!--protocol type="MPING" socket-binding="jgroups-mping"/-->
<protocol type="MERGE3"/>
<protocol type="FD_SOCK" socket-binding="jgroups-tcp-fd"/>
<protocol type="FD"/>
<protocol type="VERIFY_SUSPECT"/>
<protocol type="pbcast.NAKACK2"/>
<protocol type="UNICAST3"/>
<protocol type="pbcast.STABLE"/>
<protocol type="pbcast.GMS"/>
<protocol type="MFC"/>
<protocol type="FRAG2"/>
</stack>
```

5. Include the initial_hosts property:

   a. Search for the following text:

```
<socket-binding-group ref="full-ha-sockets"/>
</server-group>
```

b. Add the initial_hosts text to this section, as shown in bold, where the value is a list of all hosts to include in the cluster, including the domain controller and all host controllers.

```
<socket-binding-group ref="full-ha-sockets"/>
<system-properties>
<property name="jboss.cluster.tcp.initial_hosts" value="<IP address of host1>[7600],<IP address of host2>[7600]" />
</system-properties>
</server-group>
```

For example:

```
<property name="jboss.cluster.tcp.initial_hosts" value="10.31.66.130[7600],10.31.66.129[7600]" />
```

Note: The default port of 7600 works for deployments with one node per host. If a single host contains multiple nodes, different ports must be provided and added to the firewall configuration.

Restart the Aveksa Cluster

On the domain controller, restart the aveksa_cluster.

Procedure

1. Enter the following command:

   ```
   service aveksa_cluster stop
   service aveksa_cluster start
   ```

2. Check **/home/oracle/wildfly/domain/log/stdout.log** for errors.
   If you see an "Unable to parse XML file" error, ensure that the values entered in the **domain.xml** or **host.xml** files are correct.

Troubleshooting Multicast to TCP Migration

This section contains several steps you can take to verify the functionality of your WildFly cluster migration from multicast to TCP.

Confirm Communication

To confirm that the cluster setup is working, search the domain controller and host controller log files for the string Bridge ClusterConnectionBridge. The log files are located under **/home/oracle/wildfly/domain/log/stdout.log**.

Verify that messages like the following appear in the log files of both the domain controller and the host controllers.

For example, the string to look for is bold in the following log entry:

```
[Server:img-server-1] ^[[0m^[[0m09:13:22,147 INFO
[org.apache.activemq.artemis.core.server] (Thread-14 (ActiveMQ-
server-
org.apache.activemq.artemis.core.server.impl.ActiveMQServerImpl$2
@553f4fb7-1481379788)) AMQ221027: Bridge
ClusterConnectionBridge@26d0c5ef [name=sf.my-cluster.d3c66dc6-
7481-11e8-9533-0f7eebfd76f1, queue=QueueImpl[name=sf.my-
cluster.d3c66dc6-7481-11e8-9533-0f7eebfd76f1,
postOffice=PostOfficeImpl
[server=ActiveMQServerImpl::serverUUID=a5fbc63a-7470-11e8-a611-
d99efcf3c169]]@206c8d45 targetConnector=ServerLocatorImpl
(identity=(Cluster-connection-
bridge::ClusterConnectionBridge@26d0c5ef [name=sf.my-
cluster.d3c66dc6-7481-11e8-9533-0f7eebfd76f1, queue=QueueImpl
[name=sf.my-cluster.d3c66dc6-7481-11e8-9533-0f7eebfd76f1,
postOffice=PostOfficeImpl
[server=ActiveMQServerImpl::serverUUID=a5fbc63a-7470-11e8-a611-
d99efcf3c169]]@206c8d45 targetConnector=ServerLocatorImpl
[initialConnectors=[TransportConfiguration(name=http-connector,
factory=org-apache-activemq-artemis-core-remoting-impl-netty-
NettyConnectorFactory)
?httpUpgradeEnabled=true&httpPpgradeEndpoint=http-
acceptor&port=8080&host=172-24-216-26],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@1035918
432[nodeUUID=a5fbc63a-7470-11e8-a611-d99efcf3c169,
connector=TransportConfiguration(name=http-connector,
factory=org-apache-activemq-artemis-core-remoting-impl-netty-
NettyConnectorFactory)
?httpUpgradeEnabled=true&httpPpgradeEndpoint=http-
```

```
acceptor&port=8080&host=172-24-216-20, address=jms,
server=ActiveMQServerImpl::serverUUID=a5fbc63a-7470-11e8-a611-
d99efcf3c169])) [initialConnectors=[TransportConfiguration
(name=http-connector, factory=org-apache-activemq-artemis-core-
remoting-impl-netty-NettyConnectorFactory)
?httpUpgradeEnabled=true&httpPpgradeEndpoint=http-
acceptor&port=8080&host=172-24-216-26],
discoveryGroupConfiguration=null]] is connected^
```

Note that the message appears only after both the domain controller and host controller are up and running.

Confirm Communication Protocol

Connect to the nodes using SSH, and run the following command to confirm that there is a process listening on ports 7600 and 57600:

```
netstat -anp | grep 7600
```

An example of the output of the above command is as follows:

```
tcp 0 0 10.31.66.130:57600 0.0.0.0:* LISTEN 16679/java
tcp 0 0 10.31.66.130:7600 0.0.0.0:* LISTEN 16679/java
tcp 0 0 10.31.66.130:40209 10.31.66.129:57600 ESTABLISHED
16679/java
tcp 0 0 10.31.66.130:7600 10.31.66.129:16634 ESTABLISHED
16679/java
tcp 0 0 10.31.66.130:57600 10.31.66.129:49785 ESTABLISHED
16679/java
```

The above example output that ran on the machine 10.31.66.130 shows that the java process with PID 16679 is listening on 7600 and 57600 for new connections, and has two connections established from 10.31.66.129 on this port. In this example, 10.31.66.130 is the domain controller and 10.31.66.129 is the host controller.

Confirm that the java process PID is the RSA Identity Governance and Lifecycle instance running on the node to verify that 7600 and 57600 are not used by any other process.

If the communication protocol was not correctly configured, the netstat command would not show the LISTEN section. Check the configuration in domain.xml. After each change, restart the domain controller first and then restart the host controller. For instructions, see Restart Nodes After Changes.

If there is a connection issue between the nodes, the netstat command would not show the ESTABLISHED message. If the ESTABLISHED message does not appear, verify that the firewall is running and check for any connectivity issue between the nodes. To verify connectivity, telnet <IP Address> 7600 and confirm that there is communication.

Restart Nodes After Changes

You must restart all nodes any time you make a change to the **domain.xml** file on the domain controller.

Procedure

1. Stop RSA Identity Governance and Lifecycle on all nodes using the following command:

   ```
   service aveksa_cluster stop
   ```

2. After you have saved any changes to **domain.xml**, start the aveksa_cluster on the domain controller first using the following command:

   ```
   service aveksa_cluster start
   ```

3. After the domain controller has started and the stdout.log file located in /home/oracle/wildfly/domain/log directory reports that RSA Identity Governance and Lifecycle has started, start the host controller service.

   When the host controller is running, a message appears in domain controller **stdout.log** that a remote slave was registered.

   For example, for a host controller connected to a domain controller over the admin port 9999, the domain controller stdout.log file would provide the following information:

   ```
   WFLYHC0019: Registered remote slave host "slave", JBoss WildFly Full
   10.1.0.Final (WildFly 2.2.0.Final)
   ```