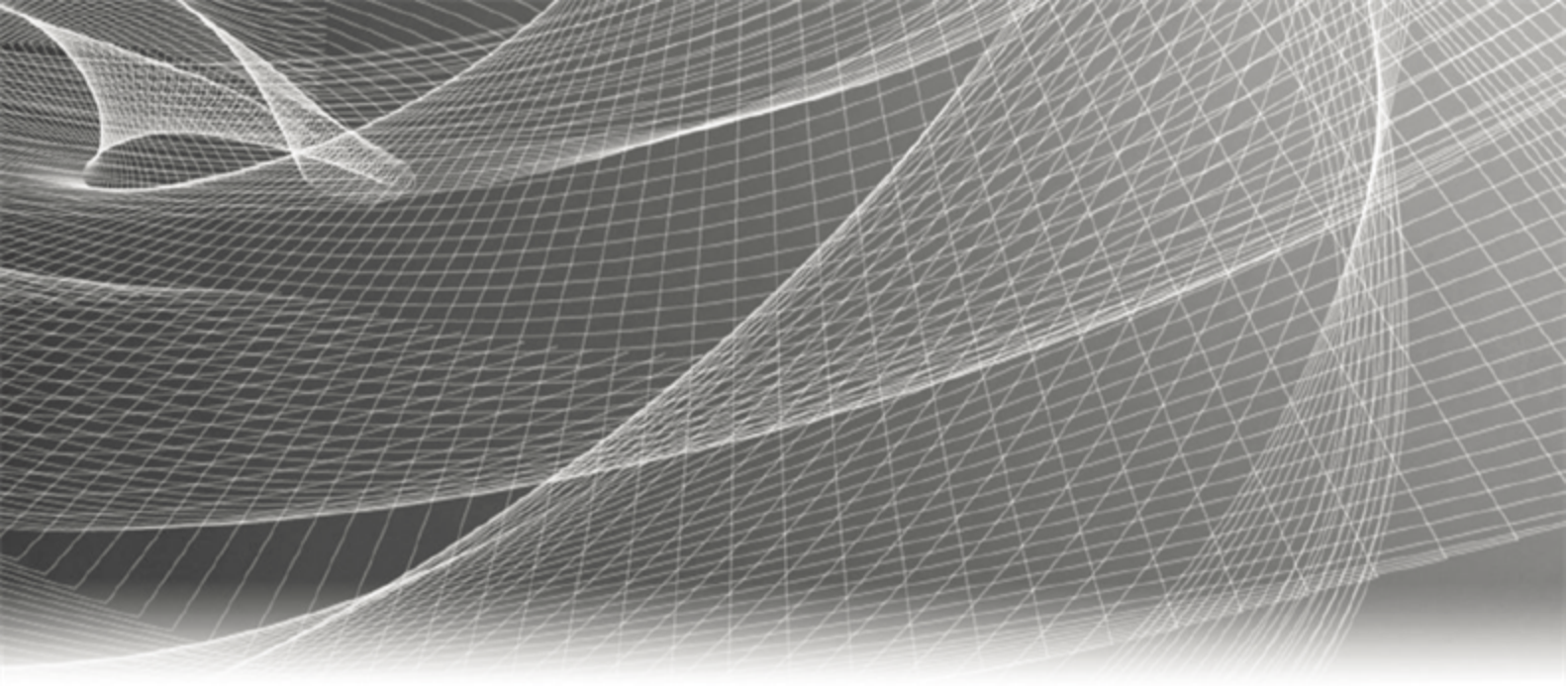


RSA Identity Governance and Lifecycle
Configuring WildFly Clustering

Version 7.2

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.

February 2020

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.
Configuring WildFly Clusters	Instructions to set up and configure a WildFly application server cluster in an RSA Identity Governance and Lifecycle deployment.
Online Help	All concepts and instructions you need to configure and use the product.
Administrator's Guide	How to configure and manage RSA Identity Governance and Lifecycle. Contains a subset of the information provided in the Online Help.
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.

Chapter 1: Overview

This solutions integration guide provides the steps required to set up and configure a WildFly application server cluster in an RSA Identity Governance and Lifecycle deployment.

In order to provide client load balancing, a front-end load balancer setup is required. The load balancer must send a client to the same WildFly server during a session. The setup of the WildFly application servers in a cluster configuration does not provide high availability load balancing services.

This guide does NOT provide information about configuring a front-end load balancer. That is outside the scope of this guide.

Note: All machines included in a clustered deployment, both domain and hosts, must be deployed as the root user, and the option to install the RSA Identity Governance and Lifecycle as a service must be selected when configured in the standalone mode.

Supported Platform Versions

This solutions integration guide is published for the following RSA Identity Governance and Lifecycle versions only. Please contact your support representative if there are questions for versions other than what is listed below.

- RSA Identity Governance and Lifecycle - Version 7.2

Audience

The following is the target audience for this guide:

- RSA Identity Governance and Lifecycle Installer & Administrator or appropriate user with network / administration rights to install and configure the RSA Identity Governance and Lifecycle application.

Supported Configurations

WildFly clustering is supported for the RSA Identity Governance and Lifecycle Software Bundle, in which multiple software bundle installations participate in a cluster configuration pointing to a remote database setup.

Note: RSA Identity Governance and Lifecycle hardware appliances do not support WildFly clustering.

Additional details for the supported configuration are provided in the following sections.

About Using this Document

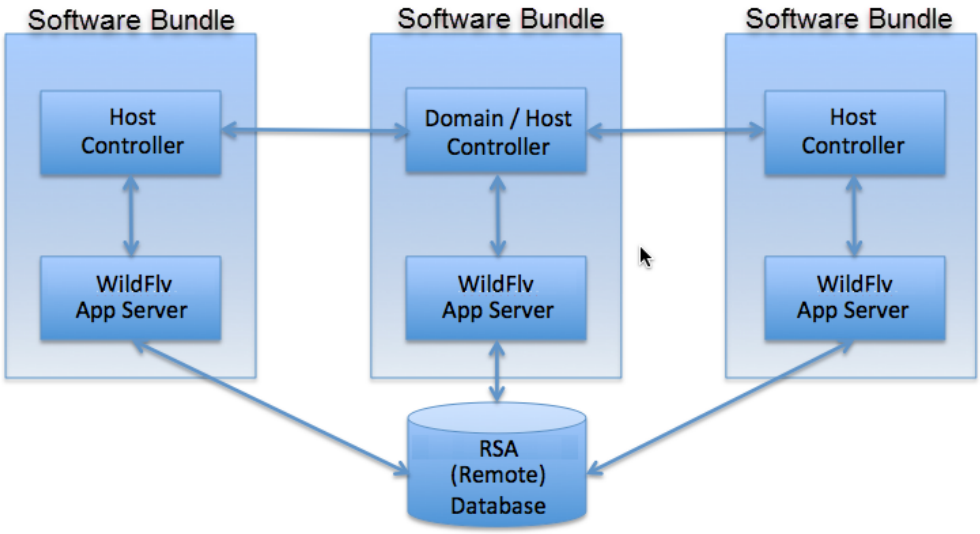
Throughout this document there are references that may include variable substitutions, such as: *$AVEKSA_WILDFLY_HOME*. This variable represents the **$AVEKSA_HOME** directory based on the initial installation and is not meant to imply that the home installation path can be changed during the setup/installation.

Note: When using the commands included in this document, ensure that you do not enter any extra spaces or line breaks.

RSA Identity Governance and Lifecycle Soft Appliance

In this configuration, the software bundle deployments are set up in a clustered environment leveraging a remote database setup. One of the software bundle deployments hosts the domain controller for the cluster and a separate machine (known as the remote database) hosts the database.

The following illustration depicts an RSA Identity Governance and Lifecycle implementation in a clustered WildFly environment using software bundle deployments connecting to a remote database deployment:



Chapter 2: Staging the Environment

Prerequisites

- Identify the following machines that will be part of this cluster setup:
    - The Domain Controller
    - The Host Controller(s)
    - The Database Server
    - The identified Systems Operation Node
- Record information about these machines in Machine Information Worksheet.
- Prepare a database environment for running RSA Identity Governance and Lifecycle 7.2 – Oracle 12.1.0.2 (latest patch). This can be a remote database environment or an RSA-supplied database environment provided in an RSA Identity Governance and Lifecycle 7.2 appliance model.
- For the additional nodes that will participate in the cluster, install RSA Identity Governance and Lifecycle 7.2 with the remote database option.
- Verify the timezone settings within the remote database:

    i. As the **sysdba** user, run the following SQL statements:

    ```
    SELECT DBTIMEZONE FROM DUAL;

    SELECT avuser.Utilities_Pkg.Get_DBTimezone_Value FROM DUAL;
    ```

    ii. If the values do not match, run the following SQL statements:

    ```
    alter database set time_zone='<second query value>';

    shutdown immediate;

    startup;
    ```

Machine Information

	Machine Name	IP Address	Systems Operations Node (Select One)
Domain Controller			
Host Controller(s)			
(Remote) Database Server			N/A

Note: When your database serves multiple application server nodes, multiply the process number by the number of healthy and active nodes.

Before You Begin

Before starting the configuration for setting up a WildFly cluster environment, you must un-deploy the current RSA Identity Governance and Lifecycle application and disable specific services.

Undeploy aveksa.ear and aveksaWFArchitect.ear

The WildFly configuration that is set up by the standard install script is deployed in a standalone mode configuration. The EAR files (aveksa.ear, aveksaWFArchitect.ear) deployed in this configuration are no longer needed in a cluster configuration. To avoid accidental deployment, conflict, and to conserve space, undeploy the EAR files.

Log in as **oracle** on each machine that is going to be part of the cluster, make sure WildFly is running, and run the **undeploy** command as follows:

```
service aveksa_server status

$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --command="undeploy
aveksa.ear"
$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --command="undeploy
aveksaWFArchitect.ear"
```

After undeploying each EAR file, verify that the EAR files do not exist by running the following commands:

```
$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --command="deployment-
info --name=aveksa.ear"
WFLYCTL0216: Management resource '[("deployment" =>
"aveksa.ear")]' not found

$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --command="deployment-
info --name=aveksaWFArchitect.ear"
WFLYCTL0216: Management resource '[("deployment" =>
"aveksaWFArchitect.ear")]' not found
```

The output of this command should report that the EAR is not found.

Stop and Disable Services

Stop and Disable AFX Services

If used, AFX services are installed on only one of the servers in the cluster.

1. Identify the server used as the AFX Engine.
2. Log in as **AFX_user** on that server.
3. Stop the AFX service:

   ```
   service afx_server stop
   ```

4. Log in as **root** on all other nodes to disable the AFX service with the following commands.

a. Stop the AFX services:

```
service afx_server stop
```

b. Unregister the services:

```
chkconfig afx_server off
```

c. Remove executable permissions from the service scripts:

```
chmod 400 /etc/init.d/afx_server
```

Stop and Disable RSA Identity Governance and Lifecycle Services

Stop and disable RSA Identity Governance and Lifecycle services before you create clustered domain instances.

1. Log in as **root** on an identified cluster node.
2. Run the following commands.

a. Stop the services:

```
service aveksa_watchdog stop
service aveksa_server stop
```

b. Unregister the services:

```
chkconfig aveksa_watchdog off
chkconfig aveksa_server off
```

c. Remove executable permissions from the service scripts:

```
chmod 400 /etc/init.d/aveksa_watchdog
chmod 400 /etc/init.d/aveksa_server
```

3. Repeat for every cluster node.

Configure Services for the Cluster

Perform these steps as the **root** user on all servers that are going to be part of the cluster.

Add and Register the Cluster Aware Service

On all servers (except the remote database) that are going to be part of the cluster setup, complete the following steps:

1. As the **root** user, copy the supplied **aveksa_cluster** file to **/etc/init.d**.
2. Back up **/etc/init.d/aveksa_cluster**.
3. Set the permissions on **aveksa_cluster** using the following command:

```
chmod 755 /etc/init.d/aveksa_cluster
```

4. Edit **/etc/init.d/aveksa_cluster** using a text editor.

5. If the server is a host controller node, change the **NODE_TYPE** variable to **SLAVE**. You can do this by uncommenting the line **#NODE_TYPE=SLAVE** and commenting the line **NODE_TYPE=DOMAIN**. If the server is the domain controller, then do not change the **NODE_TYPE**. For example:

   Domain Controller

   ```
   NODE_TYPE=DOMAIN
   #NODE_TYPE=SLAVE
   ```

   Host Controller

   ```
   #NODE_TYPE=DOMAIN
   NODE_TYPE=SLAVE
   SLAVE_HOST="HostControllerIP"
   ```

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

6. Set the **DOMAIN_MASTER** variable to the IP address of the domain controller. For example:

   ```
   DOMAIN_MASTER="10.101.250.7"
   ```

7. Set the **HOST_XML_NAME** variable to the name of the host. The name can be found in **$AVEKSA_WILDFLY_HOME/domain/configuration/host.xml** as the 'name' attribute of the **<host>** element. For example:

   ```
   HOST_XML_NAME=master
   ```

8. Save and close **/etc/init.d/aveksa_cluster**.

9. Run the following commands to register the service:

   ```
   cd /etc/init.d
   chkconfig --add aveksa_cluster
   chkconfig --level 35 aveksa_cluster on
   ```

Determine and Configure JVM Memory Settings for the Cluster

As the root user, run the following command on all servers that will participate in the cluster to determine the recommended amount of memory for the WildFly heap and meta space. Record the results and use the lowest setting when updating the **aveksa_cluster** service.

```
service aveksa_cluster getmem
```

The output displays the recommended heap and meta memory values. For example:

```
Total Memory: 16047 MB
Reserved for OS: 2048 MB
Reserved by AFX: 3072 MB
Available for WildFly: 10927 MB

 Recommended Cluster Options Settings (in MB)
WILDFLY_HEAP_MEM: 9287
WILDFLY_PERM_META_MEM: 1639
```

Edit **/etc/init.d/aveksa_cluster** in a text editor and set the **WILDFLY_HEAP_MEM** and **WILDFLY_PERM_**

META_MEM variables to the recommended values returned by the above command. For example, using the lowest common setting:

```
WILDFLY_HEAP_MEM=9686
WILDFLY_PERM_META_MEM=1709
```

Save and close **/etc/init.d/aveksa_cluster**.

Open Required Ports on all Controllers

For WildFly messaging in a cluster setup to communicate successfully, add TCP ports 7600 and 57600 to the operating systems firewall setup.

SuSE Environment

Edit /etc/sysconfig/SuSEfirewall2 and include ports 7600 and 57600 to the existing list of open ports, as shown in the following:

```
FW_SERVICES_EXT_TCP="21 22 5802 5902 80 8080 8081 8082
8161 8443 8444 8445 9999 7600 57600"
```

Red Hat Environment

Edit /etc/sysconfig/iptables and add the following lines in the correct location:

```
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m
tcp --dport 7600 -j ACCEPT

-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -m
tcp --dport 57600 -j ACCEPT
```

The cluster nodes need to communicate using JGroups using TCP protocol. The opened ports 7600 and 57600 are used by JGroups.

Open the TCP ports 8443, 8444, and 9999 using the following commands:

SuSE Environment

Edit **/etc/sysconfig/SuSEfirewall2** and set:

```
FW_SERVICES_EXT_TCP="22 8443 8444 9999"
```

Red Hat Environment

Edit **/etc/sysconfig/iptables** and add the following line in the correct location:

```
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -
tcp --dport 8443 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -
tcp --dport 8444 -j ACCEPT
-A RH-Firewall-1-INPUT -p tcp -m state --state NEW -
tcp --dport 9999 -j ACCEPT
```

Note: If other ports are in use, add them as well with the appropriate commands for your platform.

Save the file and run the following commands to apply the changes.

SuSE 11 Environment

```
/etc/init.d/SuSEfirewall2_init restart
/etc/init.d/SuSEfirewall2_setup restart
```

SuSE 12 Environment

```
systemctl restart SuSEfirewall2
```

Red Hat Environment

```
systemctl restart iptables
```

Disable Appliance Mode in Software Bundle Installation

Note: Perform this step only if you are using a software bundle for the master server where the Oracle database was provided by RSA and installed on the same machine.

If you are converting a software bundle deployment with an RSA-provided database to participate in a cluster, you are changing the database to effectively be a remote database and switching the deployment on WildFly from standalone mode to domain mode. Log into the database as **AVUSER** and run the following SQL commands:

```
update t_system_settings set value='N' where
parameter='isAppliance';
commit;
```

Chapter 3: Set Up and Configure the Domain Controller

You must up and configure the machine that has been identified as the domain controller for the cluster.
CAUTION: Only a single machine can be identified as the domain controller in the cluster. Complete this section before configuring the host controllers.

Create a Management User

The domain controller requires a management user to authenticate a host controller. The management user will be configured on the domain controller. Execute **add-user.sh** script under **$AVEKSA_HOME/wildfly/bin**. The following are values provided to the various options in the script. RSA recommends that you use a strong password for your production implementation.

```
oracle@vm-adap-10:~/wildfly/bin> ./add-user.sh
What type of user do you wish to add?
 a) Management User (mgmt-users.properties)
 b) Application User (application-users.properties)
(a): a


Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing
property files.
Username : AveksaClusterAdmin

Password recommendations are listed below. To modify these
restrictions edit the add-user.properties configuration file.
 - The password should not be one of the following restricted
values {root, admin, administrator}
 - The password should contain at least 8 characters, 1
alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol
(s)
 - The password should be different from the username
Password : <YOUR_PASSWORD_CHOICE>
Are you sure you want to use the password entered yes/no? yes
Re-enter Password : <YOUR_PASSWORD_CHOICE>
What groups do you want this user to belong to? (Please enter a
comma separated list, or leave blank for none)[  ]:
Leave blank and press Enter
About to add user 'AveksaClusterAdmin' for realm
'ManagementRealm'
Is this correct yes/no? yes
Added user 'AveksaClusterAdmin' to file '$AVEKSA_HOME/wildfly-
10.1.0.Final/standalone/configuration/mgmt-users.properties'
Added user 'AveksaClusterAdmin' to file '$AVEKSA_HOME/wildfly-
```

```
10.1.0.Final/domain/configuration/mgmt-users.properties'
Added user 'AveksaClusterAdmin' with groups to file '$AVEKSA_
HOME/wildfly-10.1.0.Final/standalone/configuration/mgmt-
groups.properties'
Added user 'AveksaClusterAdmin' with groups to file '$AVEKSA_
HOME/wildfly-10.1.0.Final/domain/configuration/mgmt-
groups.properties'
Is this new user going to be used for one AS process to connect
to another AS process?
e.g. for a slave host controller connecting to the master or for
a Remoting connection for server to server EJB calls.
yes/no? yes
To represent the user add the following to the server-identities
definition <secret value="QXZla3NhMTIz" />
```

Note: Do not use the dollar sign ($) as part of your password. The scripts interpret the symbol as a new variable and will not be able to read your password as a result.

After you add the user, note the value of the secret. In the previous example, it is "QXZla3NhMTIz". You will need this secret when configuring the host controllers:

```
To represent the user add the following to the server-identities
definition: <secret value="QXZla3NhMTIz" />
```

If you choose a password that does not meet WildFly's current complexity requirements, you may receive messages similar to the following:

```
WFLYDM0099: Password should have at least 8 characters! Are you
sure you want to use the password entered yes/no?
```

Type **yes** and press return if you want to continue with your current password choice and it will prompt you to re-enter the same password.

Configure the Domain Configuration File

Edit **domain.xml** file located at **$AVEKSA_HOME/wildfly/domain/configuration** and make the following changes:

Update Management Timeout

The **jboss.as.management.blocking.timeout** value defaults to 300 seconds. This property is a timeout on container stability. If **jboss.as.management.blocking.timeout** is reached during startup, all applications are undeployed and the container is shut down.

The following example shows the commands to set a custom value, such as 900 seconds:

```
<system-properties>
<property name="jboss.as.management.blocking.timeout"
value="900"/>
</system-properties>
```

Configure Server Group

Remove all **<server-group>** entries under the **<server-groups>** setting and add a new **<server-group>** as shown below.

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

```
<server-groups>
<server-group name="img-server-group" profile="full-ha">
<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>
 </server-groups>
```

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.

Update JVM Memory Settings

For optimal performance, add the following settings to the new server group made in Configure Server Group:

```
<server-group name="img-server-group" profile="full-ha">
  <jvm name="default">
  <heap size="${jboss.memory.max.heap.size}" max-size="${jboss.memory.max.heap.size}"/>
  <jvm-options>
    <option value="-server"/>
    <option value="-XX:MaxMetaspaceSize=${jboss.memory.max.permgen.size}"/>
    <option value="-XX:+UseG1GC"/>
    <option value="-XX:+ExplicitGCInvokesConcurrent"/>
    <option value="-XX:+ParallelRefProcEnabled"/>
    <option value="-XX:+AlwaysPreTouch"/>
    <option value="-XX:+UseStringDeduplication"/>
    <option value="-XX:InitiatingHeapOccupancyPercent=10"/>
    <option value="-XX:+HeapDumpOnOutOfMemoryError"/>
    <option value="-XX:HeapDumpPath=$AVEKSA_HOME/wildfly/domain/log/"/>
  </jvm-options>
  </jvm>
  <socket-binding-group ref="full-ha-sockets"/>
</server-group>
</server-groups>
```

Configure Cluster Authentication Credentials

1. Search for the **<profile name="full-ha">** entry and find the following:

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">
```

2. Find the password line below:

```
<cluster
password="${jboss.messaging.cluster.password:CHANGE
ME!!}"/>
```

3. Replace the password line to look like the following:

```
<cluster user="some_username" password="some_password"/>
```

The *some_username* and *some_password* that you add here will update the appropriate values by the **configureWildfly.sh** script.

Configure Domain Network Interfaces

Define the network interfaces in **domain.xml** as follows:

```
<interfaces>
        <interface name="management"/>
        <interface name="public"/>
        <interface name="unsecure">
            <inet-address
value="${jboss.bind.address.unsecure:127.0.0.1}"/>
        </interface>
        <interface name="private">
            <inet-address
value="${jboss.bind.address.private:127.0.0.1}"/>
        </interface>
    </interfaces>
```

Include TCP Ping for Discovery

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

Procedure

1. Open the file **$AVEKSA_HOME/wildlfly/domain/configuration/domain.xml** for editing.
2. 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>
```

3. 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>
```

4. 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.

Configure Host Configuration

Edit the **host.xml** file located at **$AVEKSA_HOME/wildfly/domain/configuration** and make the following changes.

Configure Server Name registered in RSA Identity Governance and Lifecycle

Remove all **<server>** entries under **<servers>** and add a new **<server>** as shown below. The **<server>** entry is an instance of WildFly application server that hosts an IMG application.

The name of the server should be unique in the group. In this example, this host manages **img-server-1**.

```
<servers>
    <server name="img-server-1" group="img-server-group" />
</servers>
```

Verify Local Domain Controller

The **<domain-controller>** setting has the following configuration in this file. The **<local/>** entry identifies this host as the domain controller.

```
<domain-controller>
    <local/>
    <!-- Alternative remote domain controller configuration with
a host and port -->
    <!-- <remote host="${jboss.domain.master.address}"
port="${jboss.domain.master.port:9999}"
security-realm="ManagementRealm"/> -->
</domain-controller>
```

Add Aveksa Security Realms

In a standalone mode (appliance mode) the installation automatically configures the security realms **AveksaAgentRealm** and **AveksaRealm**. The **AveksaAgentRealm** contains the key and trust store for the agent. The **AveksaRealm** contains the key and trust store for web access. In a clustered setup this is not part of the domain configuration. Do not configure it in **domain.xml**, instead configure these settings in the **host.xml** file.

Edit **host.xml** and add the following **AveksaAgentRealm** security realm under security-realms element, for example:

```
<security-realm name="AveksaAgentRealm">
  <server-identities>
    <ssl>
      <keystore path="$AVEKSA_HOME/keystore/server.keystore" keystore-password="Av3k5a15num83r0n3" alias="server"
key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
```

```
  <authentication>
    <truststore path="$AVEKSA_HOME/keystore/server.keystore" keystore-password="Av3k5a15num83r0n3"/>
  </authentication>
</security-realm>
```

AveksaRealm will be customer specific setup. For our testing purpose we will be using the **aveksa.keystore**.

Add the following **AveksaRealm** security realm under **security-realms** element.

```
<security-realm name="AveksaRealm">
  <server-identities>
    <ssl>
      <keystore path="$AVEKSA_HOME/keystore/aveksa.keystore" keystore-password="Av3k5a15num83r0n3" alias="server"
key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="$AVEKSA_HOME/keystore/aveksa.keystore" keystore-password="Av3k5a15num83r0n3"/>
   </authentication>
  </security-realm>
```

Update JVM Memory Settings

Use the following recommended values for JVM memory settings:

```
<jvms>
        <jvm name="default">
            <heap size="2048m"
max-size="${jboss.memory.max.heap.size}"/>
            <jvm-options>
                <option value="-server"/>
                <option value="-XX:MetaspaceSize=2048m"/>
                <option value="-XX:MaxMetaspaceSize=2048m"/>
            </jvm-options>
        </jvm>
    </jvms>
```

These settings can be further tuned to meet your environment's performance needs.

Start the Domain Controller

Log in as **oracle** on the domain controller machine. Start the domain controller:

```
service aveksa_cluster start
```

Look for the following **started** message in **$AVEKSA_WILDFLY_HOME/domain/log/stdout.log**. This message indicates that the domain controller has started.

```
=========================================================================
=========
  JBoss Bootstrap Environment
```

```
  JBOSS_HOME: $AVEKSA_HOME/wildfly
  JAVA: /usr/lib64/jvm/java-1.8.0-openjdk-1.8.0/bin/java
  JAVA_OPTS: -Xms64m -Xmx512m -XX:MaxPermSize=256m -
Djava.net.preferIPv4Stack=true -
Djboss.modules.system.pkgs=org.jboss.byteman -
Djava.awt.headless=true
=========================================================================
=========
14:02:01,871 INFO  [org.jboss.modules] (main) JBoss Modules
version 1.3.3.Final
14:02:02,159 INFO  [org.jboss.as.process.Host Controller.status]
(main) JBAS012017: Starting process 'Host Controller'
[Host Controller] 14:02:03,415 INFO  [org.jboss.modules] (main)
JBoss Modules version 1.3.3.Final
[Host Controller] 14:02:03,725 INFO  [org.jboss.msc] (main)
JBoss MSC version 1.2.2.Final
[Host Controller] 14:02:03,838 INFO  [org.jboss.as] (MSC service
thread 1-3) JBAS015899: WildFly 10.1.0.Final starting
[Host Controller] 14:02:05,264 INFO  [org.xnio] (MSC service
thread 1-1) XNIO version 3.3.0.Final

...

[Host Controller] 14:02:12,193 INFO
[org.jboss.as.domain.controller.mgmt] (Remoting "vm-adap-
22:MANAGEMENT" task-4) JBAS010920: Server [Server:img-server-1]
connected using connection [Channel ID 0f7d74e9 (inbound) of
Remoting connection 442b578f to /10.101.249.22:27320]
[Host Controller] 14:02:12,512 INFO  [org.jboss.as] (Controller
Boot Thread) JBAS015961: Http management interface listening on
http://10.101.249.22:9990/management
[Host Controller] 14:02:12,513 INFO  [org.jboss.as] (Controller
Boot Thread) JBAS015951: Admin console listening on
http://10.101.249.22:9990
[Host Controller] 14:02:12,514 INFO  [org.jboss.as] (Controller
Boot Thread) WFLYSRV0025: WildFly Full 10.1.0.Final (WildFly
Core 2.2.0.Final) started in 83234ms - Started 44 of 46 services
(13 services are lazy, passive or on-demand)
```

Note: If it does not exist, you need to create the log folder **$AVEKSA_WILDFLY_HOME/domain/log/** before running the command. This directory must have the same owner:group permissions of the **$AVEKSA_WILDFLY_HOME** directory.

Tune WildFly Configuration

Use SSH to connect to the domain controller and log in as **root**.

Edit **configureWildfly.sh** located in **/tmp/aveksa/staging/deploy** and add the following arguments to **WILDFLY_CONFIG_OPTIONS**:

```
-DOperatingMode=domain
-DDOMAIN_CONTROLLER=<IP address of the domain controller host>
-DDOMAIN_USERNAME=<management-username>
-DDOMAIN_USER_PASSWORD=<management-user-password>
-DDOMAIN_HOSTNAME=<HOST_XML_NAME value from aveksa_cluster
script>
```

In the above step, for <*management-username*> and <*management-user-password*> use the same clear text username and password that you used in the previous Create Management User section.

For example:

```
CONFIG_OPTIONS="-DAVEKSA_PASS_ENCRYPTED=${AVEKSA_PASS_ENCRYPTED}
-DAVEKSA_REPORTS_PASS_ENCRYPTED=${AVEKSA_REPORTS_PASS_ENCRYPTED}
-DAVEKSA_PUBLIC_DB_PASS_ENCRYPTED=${AVEKSA_PUBLIC_DB_PASS_
ENCRYPTED}
-DAVEKSA_AVPERF_PASS_ENCRYPTED=${AVEKSA_AVPERF_PASS_ENCRYPTED}
-DOperatingMode=domain -DDOMAIN_CONTROLLER=10.101.249.183 -
DDOMAIN_HOSTNAME=master
-DDOMAIN_USERNAME=AveksaClusterAdmin -DDOMAIN_USER_
PASSWORD=Aveksa123"
```

The following script uses the configuration in **$AVEKSA_HOME/Aveksa_System.cfg** for the remote database configuration.

Ensure that the following settings are correctly configured in this file:

- REMOTE_ORACLE=Y
- REMOTE_ORACLE_IP=<Remote Oracle database instance IP address>
- REMOTE_ORACLE_PORT=<Remote Oracle database instance port number>
- AVEKSA_PASS
- AVEKSA_REPORTS_PASS,
- AVEKSA_PUBLIC_DB_PASS,
- AVEKSA_AVPERF_PASS

Once you have verified that the settings are correct, run the following command:

```
cd /tmp/aveksa/staging/deploy

./configureWildfly.sh
```

This script will setup datasources, messaging properties and SSL properties for the application. Any failures should be closely examined. If the script fails, follow these steps before you re-run the script:

1. Stop the **aveksa_cluster** service:
    - For SuSE 11, enter:

        ```
        service aveksa_cluster stop
        ```

- For SuSE 12, enter:

```
systemctl stop aveksa_cluster
```

2. Edit **domain.xml** to remove the jdbc sources.
3. Start the **aveksa_cluster** service:
   - For SuSE 11, enter:

```
service aveksa_cluster start
```

   - For SuSE 12, enter:

```
systemctl start aveksa_cluster
```

If the script runs successfully, restart the domain controller when finished:

SuSE 11 commands

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

SuSE 12 commands

```
systemctl stop aveksa_cluster
systemctl start aveksa_cluster
```

Deploy RSA Identity Governance and Lifecycle

The **aveksa.ear** and **aveksaWFArchitect.ear** files are deployed on the domain controller machine. The domain controller then propagates each EAR to all of the servers.

Before you begin

Identify the full path of the **aveksa.ear** and **aveksaWFarchitect.ear** files. On a standalone machine that is being added to a cluster, the EAR files are located in **/tmp/repackaged_ear_dir**. The EAR files are deployed to **img-server-group**.

Procedure

1. To deploy EAR files, log in as **oracle** user and execute the following command:

```
$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --controller=<IP
address of domain controller>
```

2. At the CLI prompt, enter:

```
[domain@localhost:9990]: deploy /tmp/repackaged_ear_
dir/aveksa.ear --server-groups=img-server-group
[domain@localhost:9990]: deploy /tmp/repackaged_ear_
dir/aveksaWFArchitect.ear --server-groups=img-server-group
```

After you finish

To check that the EAR files are deployed, monitor the **$AVEKSA_HOME/wildfly/domain/log/stdout.log** file.

For more information, see Log File Location and Properties on page 32.

Chapter 4: Set Up and Configure the Host Controllers

Note: This section is for setting up and configuring each host controller machine that you plan to be part of the cluster.

The next step is to configure the servers that will be "host controllers" in the cluster setup.

Create Host Configuration File

The **host-slave.xml** is a template that is used to configure this machine as a host controller. Within the **$AVEKSA_HOME/wildfly/domain/configuration** folder, run the following commands to copy the **host-slave.xml** to **host.xml**:

```
cd $AVEKSA_HOME/wildfly/domain/configuration

cp host-slave.xml host.xml
```

Configure Host Configuration File

In this step, edit the **host.xml** file to set the unique name for each host controller and set up the security settings to participate in the cluster.

Name the Host Controller

It is a good practice to name your host controllers. The names appear in the logs and in the WildFly management console.

This name must be unique for each host controller. Add the name attribute to the host element.

```
<?xml version='1.0' encoding='UTF-8'?>
<host name="<server name>" xmlns="urn:jboss:domain:4.2">
```

Add management user secret value

In this file locate the **<server-identities>** under **<security-realm name="ManagementRealm">**. Change the value of the secret element to the value you noted when creating the management user on the domain controller.

```
<management>
        <security-realms>
            <security-realm name="ManagementRealm">
                <server-identities>
                     <!-- Replace this with either a base64
password of your own, or
    use a vault with a vault expression -->
                     <secret value="QXZla3NhMTIz">
                </server-identities>
```

Add Aveksa Security Realms

The **AveksaAgentRealm** contains the key and trust store for the agent. The **AveksaRealm** contains the key and trust store for the web access. In standalone mode (appliance mode), the security realms **AveksaAgentRealm** and **AveksaRealm** are configured automatically. In a clustered setup, this is not part of the domain configuration. You do not configure these in **domain.xml**. Instead, you configure the **host.xml**.

Note: Perform these changes on all machines where WildFly is installed and will be part of the cluster setup.

Edit **host.xml** and add the following **AveksaAgentRealm** security realm under the **security-realms** element:

```
<security-realm name="AveksaAgentRealm">
  <server-identities>
    <ssl>
      <keystore path="$AVEKSA_HOME/keystore/server.keystore" keystore-password="Av3k5a15num83r0n3" alias="server"
key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="$AVEKSA_HOME/keystore/server.keystore" keystore-password="Av3k5a15num83r0n3"/>
  </authentication>
</security-realm>
```

AveksaRealm will be a customer specific setup. For testing and documentation purposes we use the **aveksa.keystore**. Add the following **AveksaRealm** security realm under the **security-realms** element.

```
<security-realm name="AveksaRealm">
  <server-identities>
    <ssl>
      <keystore path="$AVEKSA_HOME/keystore/aveksa.keystore" keystore-password="Av3k5a15num83r0n3" alias="server"
key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="$AVEKSA_HOME/keystore/aveksa.keystore" keystore-password="Av3k5a15num83r0n3"/>
  </authentication>
</security-realm>
```

Configure Domain Controller Settings

Each host controller requires the domain controller and the management user credentials to authenticate. To the remote element, add the username attribute and set its value as username of the management user you added during the domain controller setup and configuration.

In this example the value of username is **AveksaClusterAdmin**.

```
<domain-controller>
  <remote host="${jboss.domain.master.address}" port="${jboss.domain.master.port:9999}" username="AveksaClusterAdmin"
security-realm="ManagementRealm"/>
</domain-controller>
```

Configure Server Name registered in RSA Identity Governance and Lifecycle

Remove all **<server>** entries under **<servers>** and add a new **<server>** as shown below. The **<server>** entry is an instance of the WildFly application server that will host the RSA Identity Governance and Lifecycle application. The name of the server should be unique in the group. The name should not be the hostname because WildFly allows multiple servers running on the same host. In this example, this host manages **img-server-2**.

```
<servers>
        <server name="img-server-2" group="img-server-group"/>
</servers>
```

Update JVM Memory Settings

Use the following recommended values for JVM memory settings:

```
<jvms>
        <jvm name="default">
            <heap size="2048m"
max-size="${jboss.memory.max.heap.size}"/>
            <jvm-options>
                <option value="-server"/>
                <option value="-XX:MetaspaceSize=2048m"/>
                <option value="-XX:MaxMetaspaceSize=2048m"/>
            </jvm-options>
        </jvm>
    </jvms>
```

These settings can be further tuned to meet your environment's performance needs.

Start the Host Controllers

Login using the oracle user on each host controller machine. Start the host controller by executing the following command:

SuSE 11 Command

```
service aveksa_cluster start
```

SuSE 12 Command

```
sudo systemctl start aveksa_cluster.service
```

Note: If it does not exist, you need to create the log folder **$AVEKSA_WILDFLY_HOME/domain/log/** before running the command. This directory must have the same owner:group permissions of the **$AVEKSA_WILDFLY_HOME** directory.

You will notice following message in **$AVEKSA_WILDFLY_HOME/domain/log/stdout.log** on the domain controller.

```
[Host Controller] 16:27:42,141 INFO  [org.jboss.as.domain] (Host
Controller Service Threads - 27) WFLYHC0019: Registered remote
slave host "img-server-2", JBoss WildFly Full 10.1.0.Final
(WildFly 2.2.0.Final)
```

Note: If you get a timeout message, open port 9999 on the domain controller machine.

Chapter 5: Operations and Management

Log File Location and Properties

Location

The application server log files are located on each machine at **$AVEKSA_HOME/wildfly/domain/servers/**<*server-name*>**/log** directory, where <*server-name*> is the name of the application server.

The domain/host controller log files are located in the **$AVEKSA_HOME/wildfly/domain/log** directory.

Log4J Property File

On a cluster setup the aveksa-log4j.properties file is located in **$AVEKSA_HOME/wildfly/domain/servers/**<*server-name*>**/configuration** directory, where <*server-name*> is the name of the application server.

Server Operations Node (SON)

Assign a Server Operations Node

Login to RSA Identity Governance and Lifecycle user interface with an admin (e.g. AveksaAdmin) account to select one of the server modes as the System Operations Node. Click Admin > System, and select the Server Nodes tab. Click the "Make Next SON" button for one of the server to designate it as the System Operations Node.

CAUTION: In the Server Nodes tab, if you see entries that are not nodes in the cluster (for example, duplicate entries), you should delete those entries.

Reassigning node as next SON

You can designate a single node only as the SON role in a cluster. All other nodes must be general nodes.

If a general node is in an active state, you can designate it to be the next SON when it is restarted and the currently assigned SON has been shut down.

If a general node is in an inactive state, you should shut down the current SON and then designate the inactive general node to be the next SON when it is restarted.

Cluster Checklist

Confirm JMS messaging is working

In the log file look for a message similar to the following example This message appears on both the Host Controller and the Domain Controller:

```
[Server:img-server-1] 23:09:48,413 INFO
[org.apache.activemq.artemis.core.server] (Thread-6 (ActiveMQ-
server-
```

```
org.apache.activemq.artemis.core.server.impl.ActiveMQServerImpl$
2@73fc1c8b-728162539)) AMQ221027: Bridge
ClusterConnectionBridge@3d1ca940 [name=sf.my-cluster.0dcec9a2-
c2be-11e7-851a-2795e894791f, queue=QueueImpl[name=sf.my-
cluster.0dcec9a2-c2be-11e7-851a-2795e894791f,
postOffice=PostOfficeImpl
[server=ActiveMQServerImpl::serverUUID=5a936be7-bfeb-11e7-b254-
01ee8de4c9e5]]@2c545b0e targetConnector=ServerLocatorImpl
(identity=(Cluster-connection-
bridge::ClusterConnectionBridge@3d1ca940 [name=sf.my-
cluster.0dcec9a2-c2be-11e7-851a-2795e894791f, queue=QueueImpl
[name=sf.my-cluster.0dcec9a2-c2be-11e7-851a-2795e894791f,
postOffice=PostOfficeImpl
[server=ActiveMQServerImpl::serverUUID=5a936be7-bfeb-11e7-b254-
01ee8de4c9e5]]@2c545b0e 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=10-101-249-164],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@885477
902[nodeUUID=5a936be7-bfeb-11e7-b254-01ee8de4c9e5,
connector=TransportConfiguration(name=http-connector,
factory=org-apache-activemq-artemis-core-remoting-impl-netty-
NettyConnectorFactory)
?httpUpgradeEnabled=true&httpPpgradeEndpoint=http-
acceptor&port=8080&host=10-101-249-101, address=jms,
server=ActiveMQServerImpl::serverUUID=5a936be7-bfeb-11e7-b254-
01ee8de4c9e5])) [initialConnectors=[TransportConfiguration
(name=http-connector, factory=org-apache-activemq-artemis-core-
remoting-impl-netty-NettyConnectorFactory ?
httpUpgradeEnabled=true&httpPpgradeEndpoint=http-
acceptor&port=8080&host=10-101-249-164],
discoveryGroupConfiguration=null]] is connected
```

Confirm server nodes in UI

The format of the server nodes names in a cluster is <*hostname*>-<*wildfly-server-name*>.

Login to RSA Identity Governance and Lifecycle, click **Admin** > **System** and select the **Server Nodes** tab. Delete the server nodes that are unresponsive and have only **hostname** as the server node name.

Authentication Sources

You can create/update/delete an Authentication Source by clicking **Admin** > **System** and selecting the **Authentication** tab. In a clustered environment, if you configure the Authentication Source from a node that is not on the same server as the domain controller, the configuration does not take effect until you restart WildFly on the domain controller server.

Set Secure Cookies

By default in the clustered setup, secure cookies are not enabled. As a best practice, enable secure cookies so that a user can only log into RSA Identity Governance and Lifecycle over HTTPS.

In a clustered setup, you cannot toggle the secure cookie setting from the application UI (by clicking **Admin** > **System**, and selecting the **Security** tab). If you try to update the setting from the UI, you see the following error in **aveksaServer.log**:

```
ERROR (default task-60)
[com.aveksa.server.authentication.AuthProviderUtils] Failed to
change secure session cookie value to true Error message:
"JBAS014883: No resource definition is registered for address [
    (\"subsystem\" => \"undertow\"),
    (\"servlet-container\" => \"default\"),
    (\"setting\" => \"session-cookie\")
]"
```

To set secure cookies in a clustered setup, log into the domain controller server and change to the **<AVEKSA_HOME>/wildfly/bin** directory. Then connect to the JBoss CLI using the following command:

```
./jboss-cli.sh -c --controller=<domain-controller-ip-address>
```

At the CLI command prompt, issue the following command:

```
/profile=full-ha/subsystem=undertow/servlet-
container=default/setting=session-cookie:write-attribute
(name="secure",value="true")
```

You should receive a response that starts with "outcome" => "success". It is only necessary to do this on the domain controller. After setting the value through the CLI, restart the RSA Identity Governance and Lifecycle application. For example:

```
acmr620-02:$AVEKSA_HOME/wildfly/bin # ./jboss-cli.sh -c --
controller=10.101.250.7:9990
[domain@10.101.250.7:9990 /] /profile=full-
ha/subsystem=undertow/servlet-container=default/setting=session-
cookie:write-attribute(name="secure",value="true")
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"img-server-group" => {"host" => {
        "acmr620-03" => {"img-server-2" => {"response" => {
            "outcome" => "success",
            "result" => undefined,
            "response-headers" => {
                "operation-requires-reload" => true,
                "process-state" => "reload-required"
            }
```

```
            }}},
            "master" => {"img-server-1" => {"response" => {
                "outcome" => "success",
                "response-headers" => {
                    "operation-requires-reload" => true,
                    "process-state" => "reload-required"
                }
            }}}
        }}}
}
[domain@10.101.250.7:9990 /] exit
```

After you finish

After you set the value using the CLI, restart the RSA Identity Governance and Lifecycle application on all nodes of the cluster.

Configure Logging

In the clustered setup, you cannot configure the logging setting in the application UI (by clicking **Admin** > **System**, and selecting the **Logging** tab). Use the following sections to configure logging in the cluster.

Initial Logging Setup

To initially configure the logging settings, log into the domain controller server and change directories to **<AVEKSA_HOME>/wildfly/bin**. Then connect to the JBoss CLI using the following command:

```
./jboss-cli.sh -c --
controller=<domain-controller-ip-address>:9999
```

At the CLI command prompt, issue the following command:

```
/profile=full-ha/subsystem=logging/periodic-rotating-file-handle
r=FILE/:remove
```

You should receive a response that starts with "outcome" => "success".

Then issue the following command:

```
/profile=full-ha/subsystem=logging/size-rotating-file-handler=FI
LE/:add(rotate-size=100m,named-formatter=PATTERN,file=
{relative-to=>jboss.server.log.dir,path=>server.log},max-backup-
index=5)
```

You should receive a response that starts with "outcome" => "success". This configures the server.log to rotate when the size reaches 100m and keeps up to 5 rolled over files.

For example:

```
acmr620-02:$AVEKSA_HOME/wildfly/bin # ./jboss-cli.sh -c --
controller=10.101.250.7:9999
[domain@10.101.250.7:9999 /]
```

```
/profile=full-ha/subsystem=logging/periodic-rotating-file-handle
r=FILE/:remove
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"img-server-group" => {"host" => {
        "acmr620-03" => {"img-server-2" => {"response" => {
            "outcome" => "success",
            "result" => undefined
        }}},
        "master" => {"img-server-1" => {"response" => {"outcome"
=> "success"}}}
    }}}
}
[domain@10.101.250.7:9999 /]
/profile=full-ha/subsystem=logging/size-rotating-file-handler=FI
LE/:add(rotate-size=100m,named-formatter=PATTERN,file=
{relative-to=>jboss.server.log.dir,path=>server.log},max-backup-
index=5)
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"img-server-group" => {"host" => {
        "acmr620-03" => {"img-server-2" => {"response" => {
            "outcome" => "success",
            "result" => undefined
        }}},
        "master" => {"img-server-1" => {"response" => {"outcome"
=> "success"}}}
    }}}
}
[domain@10.101.250.7:9999 /] exit
```

Modify Logging

Once the initial logging is set up using the above steps, you can modify the max log file size and the number of rolled over files to keep. To modify those settings, log into the domain controller server and change directories to **<AVEKSA_HOME>/wildfly/bin**. Then connect to the JBoss CLI using the following command:

```
./jboss-cli.sh -c --controller=<domain-controller-ip-
address>:9999
```

Issue the following command and provide the values that you want to set for **rotate-size** and **max-backup-index**. In this example, **rotate-size** is 300m and **max-backup-size** is 10:

```
/profile=full-ha/subsystem=logging/size-rotating-file-
handler=FILE:update-properties(rotate-size="300m",max-backup-
index="10")
```

For example:

```
acmr620-02:$AVEKSA_HOME/wildfly/bin # ./jboss-cli.sh -c --
controller=10.101.250.7:9999
[domain@10.101.250.7:9999 /] /profile=full-
ha/subsystem=logging/size-rotating-file-handler=FILE:update-
properties(rotate-size="300m",max-backup-index="10")
{
    "outcome" => "success",
    "result" => undefined,
    "server-groups" => {"img-server-group" => {"host" => {
        "acmr620-03" => {"img-server-2" => {"response" => {
            "outcome" => "success",
            "result" => undefined
        }}},
        "master" => {"img-server-1" => {"response" => {"outcome"
=> "success"}}}
    }}}
}
[domain@10.101.250.7:9999 /] exit
```

Uninstall a Cluster

Use the following procedure to uninstall a cluster.

1. Log in as **root** on all machines in the cluster.
2. Run the following commands:

```
cd /tmp/aveksa/staging/deploy
./uninstall.sh
```

Backup and Restore

In a clustered environment, there is only one database instance. The server nodes all connect to this single instance, so RSA strongly recommends backing up the database.

If the database is remote to the domain controller, then see the chapter entitled *Maintaining the Database* in the Database Setup and Management Guide.

Chapter 6: Upgrading Your Cluster

This section explains how to upgrade RSA Identity Governance and Lifecycle 7.0 to the base version 7.2 without patches installed on a WildFly application server cluster.

Prerequisites

Download and decompress the installation files Aveksa_<*VersionNumber*>_P<*PatchNumber*>.tar.gz and, if necessary, upgradeJDK<*version*>_<*revision*>.tar. For instructions, see "Before you Begin" at Installing a Patch or Upgrade for RSA Identity Governance and Lifecycle on page 46.

Use your Oracle credentials to log in on each machine in the cluster and make sure that the WildFly cluster is running.

Undeploy aveksa.ear

1. Log in as **oracle** to the domain controller machine, and run the following command:

   ```
   $AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --controller=<ip-
   address> --command="undeploy aveksa.ear --server-groups=img-
   server-group"
   ```

   Where <*ip-address*> is the IP address of domain controller machine.

2. After undeploying the EAR file, run the following commands to verify that the EAR does not exist:

   ```
   $AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --controller=<ip-
   address> --command="deployment-info --server-group=img-server-
   group"
   ```

   Where <*ip-address*> is IP address of domain controller machine. You should see the following output:

   ```
   WFLYCTL0216: Management resource '[("deployment" =>
   "aveksa.ear")]' not found
   ```

3. Run the following command to stop the AFX server:

   ```
   $AVEKSA_HOME/AFX/afx stop
   ```

Stop And Enable Services

Log in as **oracle** on all machines (master and slaves), and run the following command:

```
service aveksa_cluster stop
```

Set executable permissions from the service scripts:

```
chmod 755 /etc/init.d/aveksa_watchdog
```

```
chmod 755 /etc/init.d/aveksa_server
```

Unregister the following services:

```
chkconfig aveksa_watchdog on
chkconfig aveksa_server on
```

Copy Installation Files

Copy the RSA Identity Governance and Lifecycle 7.1 installer binaries to **/tmp/aveksa/staging** and packages to **/tmp/aveksa/packages** on all machines.

Backup WildFly and AFX

The installer will have the option to back up the current WildFly and AFX installation on each node.

For a manual backup of WildFly, backup **wildfly-10.1.0.Final** on each node and remove its symbolic link.

Upgrade Domain Controller

The following steps explain how to upgrade the domain controller.

1. Go to **/tmp/aveksa/staging/deploy**.
2. Run **install.sh** using the same database server.
3. When prompted for database migration to 7.2, select YES.
4. After installation, stop the **aveksa_server** service.
5. Reconfigure the newly installed aveksa_cluster service using the steps in Configure Services for the Cluster on page 12.
6. Perform all steps in Set Up and Configure the Domain Controller on page 16.
7. Perform all steps in Configure Global Modules on page 39.
8. Run the Schema Migration on page 42.

Configure Global Modules

Log in as oracle to the domain controller machine, and configure global modules.

Configure Aveksa JDBC

Run the following commands:

```
cd /tmp/aveksa/staging/deploy/wildfly-conf/lib/
cp aveksa-jboss.jar $AVEKSA_
HOME/wildfly/modules/com/aveksa/jdbc/main
```

Configure Oracle JDBC

1. Run the following commands:

```
cd /tmp/aveksa/staging/deploy/oracle
```

```
cp ojdbc8.jar $AVEKSA_HOME/wildfly/modules/com/oracle/main
cp xdb6.jar $AVEKSA_HOME/wildfly/modules/com/oracle/main
cp xmlparserv2_sans_jaxp_services.jar $AVEKSA_
HOME/wildfly/modules/com/oracle/main
cd $AVEKSA_HOME/wildfly/modules/com/oracle/main
```

2. If you see **ojdbc5.jar**, execute the following command:

   ```
   rm ojdbc5.jar
   ```

   If you see **ojdbc6.jar**, execute the following command:

   ```
   rm ojdbc6.jar
   ```

   If you see **ojdbc7.jar**, execute the following command:

   ```
   rm ojdbc7.jar
   ```

3. Edit **module.xml**, and replace its contents with the following XML:

   ```
   <?xml version="1.0" ?>
   <module xmlns="urn:jboss:module:1.1" name="com.oracle">
       <resources>
           <resource-root path="ojdbc8.jar"/>
           <resource-root path="xdb6.jar"/>
           <resource-root path="xmlparserv2_sans_jaxp_
   services.jar"/>
       </resources>
       <dependencies>
           <module name="javax.api"/>
           <module name="javax.resource.api"/>
       </dependencies>
   </module>
   ```

Configure RSA Crypto Module

1. Run the following commands:

   ```
   cd $AVEKSA_HOME/wildfly/modules/com/rsa
   rm -rf cryptoj cryptojce cryptojcommon
   mkdir main
   cd /tmp/aveksa/staging/deploy/wildfly-conf/lib
   cp cryptojce.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
   cp cryptojcommon.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
   cp jcmFIPS.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
   cd $AVEKSA_HOME/wildfly/modules/com/rsa/main
   ```

2. Create a file named **module.xml**.

3. Add the following contents and save the file:

   ```
   <?xml version="1.0" ?>
   <module xmlns="urn:jboss:module:1.1" name="com.rsa">
   ```

```
    <resources>
        <resource-root path="jcmFIPS.jar"/>
        <resource-root path="cryptojcommon.jar"/>
        <resource-root path="cryptojce.jar"/>
    </resources>

    <dependencies>
        <module name="javax.api"/>
    </dependencies>
</module>
```

Global Module Changes in Domain Configuration

Edit **$AVEKSA_HOME/wildfly/domain/configuration/domain.xml** as follows:

1. Replace the global-modules element with these values:

```
<global-modules>
                <module name="javax.wsdl4j.api" slot="main"/>
                <module name="com.oracle" slot="main"/>
                <module name="net.sf.jasperreports"
slot="main"/>
                <module name="com.aveksa.jdbc" slot="main"/>
                <module name="com.aveksa.http" slot="main"/>
                <module name="com.rsa" slot="main"/>
 </global-modules>
```

2. Under the system-properties element, add the following property:

```
<property name="rsavialg.security.keydir" value="$AVEKSA_
HOME/security"/>
```

Start the Domain Controller Server

Run the following command:

```
service aveksa_cluster start
```

Keep the domain controller and the server that is configured as SON running.

Deploy EAR files

Log in as **oracle**, and run the following commands:

1. Deploy the **aveksa.ear** file:

```
$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --controller=<ip-
address> --command="deploy /tmp/aveksa/staging/aveksa.ear --
server-groups=img-server-group"
```

Where <ip-address> is the IP address of the domain controller machine.

2. Deploy the **aveksaWFArchitect.ear** file:

```
$AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --controller=<ip-
address> --command="deploy
/tmp/aveksa/staging/aveksaWFArchitect.ear --server-groups=img-
server-group"
```

Where <*ip-address*> is the IP address of the domain controller machine.

Stop and Start Domain Controller Server

Run the following commands:

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

Schema Migration

Access the main page of the server that is designated as a Systems Operation Node (SON). You will see the option to perform schema migration.

Do the following:

1. Enter the schema migration authorization password and click **Migrate Schema**.
2. Click the **Follow Output** link to view the progress of the schema migration. When the migration is complete, a message reads "Initialization operations completed. Please restart the application server."
3. Log in as **oracle**, and run the following commands to stop and start the domain controller:

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

4. Log in to the application using your administrator credentials, and verify that the log in is successful.

Upgrade Slave Nodes

The following steps are for upgrading all slave nodes.

1. Go to **/tmp/aveksa/staging/deploy**.
2. Run **install.sh** using the same database server.
3. When prompted for database migration to 7.2, select YES.
4. After installation, stop the **aveksa_server** service.
5. Reconfigure the newly installed aveksa_cluster service using the steps in Configure Services for the Cluster on page 12.
6. Perform all steps in Set Up and Configure the Host Controllers on page 28.
7. Perform all steps in Configure Global Modules on page 42.

Configure Global Modules

On each slave node, log in as **oracle**, and configure the following global modules.

Configure Aveksa JDBC

Run the following commands:

```
cd /tmp/aveksa/staging/deploy/wildfly-conf/lib/
cp aveksa-jboss.jar $AVEKSA_
HOME/wildfly/modules/com/aveksa/jdbc/main
```

Configure Oracle JDBC

1. Run the following commands:

```
cd /tmp/aveksa/staging/deploy/oracle
cp ojdbc8.jar $AVEKSA_HOME/wildfly/modules/com/oracle/main
cp xdb6.jar $AVEKSA_HOME/wildfly/modules/com/oracle/main
cp xmlparserv2_sans_jaxp_services.jar $AVEKSA_
HOME/wildfly/modules/com/oracle/main
cd $AVEKSA_HOME/wildfly/modules/com/oracle/main
```

2. If you see **ojdbc5.jar**, execute the following command:

```
rm ojdbc5.jar
```

   If you see **ojdbc6.jar**, execute the following command:

```
rm ojdbc6.jar
```

   If you see **ojdbc7.jar**, execute the following command:

```
rm ojdbc7.jar
```

3. Edit **module.xml** , and replace its contents with the following XML:

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
    <resources>
        <resource-root path="ojdbc8.jar"/>
        <resource-root path="xdb6.jar"/>
        <resource-root path="xmlparserv2_sans_jaxp_
services.jar"/>
    </resources>
    <dependencies>
        <module name="javax.api"/>
        <module name="javax.resource.api"/>
    </dependencies>
</module>
```

Configure RSA Crypto Module

1. Run the following commands:

```
cd $AVEKSA_HOME/wildfly/modules/com/rsa
rm -rf cryptoj cryptojce cryptojcommon
mkdir main
cd /tmp/aveksa/staging/deploy/wildfly-conf/lib
cp cryptojce.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
cp cryptojcommon.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
```

```
cp jcmFIPS.jar $AVEKSA_HOME/wildfly/modules/com/rsa/main
cd $AVEKSA_HOME/wildfly/modules/com/rsa/main
```

2. Create a file named **module.xml**.
3. Add the following contents and save the file:

```
<?xml version="1.0" ?>
<module xmlns="urn:jboss:module:1.1" name="com.rsa">
    <resources>
        <resource-root path="jcmFIPS.jar"/>
        <resource-root path="cryptojcommon.jar"/>
        <resource-root path="cryptojce.jar"/>
    </resources>

    <dependencies>
        <module name="javax.api"/>
    </dependencies>
</module>
```

Upgrade AFX

The following steps are for upgrading AFX.

Stop the AFX Server

1. Connect to the AFX server host using the afx account.
2. Stop AFX by using the afx admin script located in the top level AFX installation directory:

```
<path-to-AFX>/afx stop
```

Example 1: **/home/afxuser/AFX/afx stop**

Example 2: **$AVEKSA_HOME/AFX/afx stop**

Import Connectors

Download the **AFX-**<*product-version*>**-Standard-Connectors.zip** file for this RSA Identity Governance and Lifecycle release from RSA Link at https://community.rsa.com/community/products/governance-and-lifecycle to a host that you can access with RSA Identity Governance and Lifecycle using a web browser.

1. Log on to RSA Identity Governance and Lifecycle.
2. Select **AFX** > **Import**.
3. Browse to the **AFX-**<*product-version*>**-Standard-Connectors.zip** file.
4. Select **Next**.
5. Check the **Select all** items box to select all connector templates listed for import.
6. Select **Import** to load all standard connector template packages for this released version into RSA Identity Governance and Lifecycle.

7. If you are licensed for one or more AFX Premium Connectors, repeat steps 1 through 5 for **AFX-**<*product-version*>**-Premium-Connectors.zip** (also located in the packages directory for RSA Identity Governance and Lifecycle v6.9.x and later).

Start the AFX Server

1. Connect to the AFX server host using the "afx account."
2. Start AFX by using the "afx" admin script located in the top level AFX installation directory:

   ```
   <path-to-AFX>/afx start
   ```

   Example 1: **/home/afxuser/AFX/afx start**

   Example 2: **$AVEKSA_HOME/AFX/afx start**

Chapter 7: Installing a Patch or Upgrade for RSA Identity Governance and Lifecycle

To install a patch or upgrade for RSA Identity Governance and Lifecycle deployed in a WildFly cluster, you need to undeploy the existing EAR (Enterprise Application Archive) files, then deploy the EAR files provided in the patch or upgrade files.

Before You Begin

1. Download the patch or upgrade file:
    a. Log in to RSA Link, and click RSA Identity Governance and Lifecycle.
    b. Click **Downloads** > **RSA Identity Governance and Lifecycle<*Version*>**, where <*Version*> is the version number of the product that you are patching.
    c. Click **Version Upgrades**.
    d. Click the **Upgrade** link for your licensed RSA Identity Governance and Lifecycle asset.
    e. Click **Continue**.
    f. On the Order Detail page, click the menu icon and select **Product List**.
    The **Current** tab provides the most current release or patch, and the **Archive** tab provides previous patches and releases.
    g. Click the appropriate tab, and select the name of the patch to download.
    h. Download the following files:

    Aveksa_<*VersionNumber*>_P<*PatchNumber*>.tar.gz

    i. Copy Aveksa_<*VersionNumber*>_P<*PatchNumber*>.tar.gz to $AVEKSA_HOME, and unzip the file using the following command:

    ```
    tar -xvf Aveksa_<VersionNumber>_P<PatchNumber>.tar.gz
    ```

2. Ensure that the server running on the domain controller machine is configured as the Systems Operation Node (SON). To do this, in RSA Identity Governance and Lifecycle, go to **Admin** > **System** > **Server Nodes**.
3. Copy the patch to the Domain Controller/SON.
4. Make sure that RSA Identity Governance and Lifecycle is running so you can undeploy and redeploy the two EAR files: **aveksa.ear** and **aveksaWFArchitect.ear**.
5. Make sure that all host controller nodes are running.

Procedure

Repackage EAR with Customizations

1. Log into the domain controller server as the **oracle** user.
2. Copy the currently installed EAR file for the cluster to archive directory:
    i. Locate the EAR file:

    ```
    cd $AVEKSA_HOME//wildfly/domain/servers/<server
    ```

```
name>/tmp/vfs/temp/<latest tempXXXXXX
directory>/<latest content-XXXXXXXXX directory
containing larger content file>
```

ii. Copy the larger content file to archive directory:

```
cp content $AVEKSA_HOME/archive/aveksa_cluster.ear
```

3. Extract customized content:

i. Run the command:

```
cd $AVEKSA_HOME/deploy
```

ii. At the prompt, enter:

```
./customizeACM.sh -c $AVEKSA_HOME/archive/aveksa_
cluster.ear
```

iii. Follow the prompts to unpack the EAR to **/tmp/customizeACM**.

4. Extract aveksa.ear from the patch, and change to the patch directory:

```
cd Aveksa_<VersionNumber>_P<PatchNumber>
```

5. Copy all decompressed files under **aveksa.ear** to the **/tmp/customizeACM** directory:

```
cp -pr aveksa.ear/* /tmp/customizeACM/
```

6. Zip files in **/tmp/customizeACM**:

i. `cd /tmp/customizeACM`

ii. `jar cvf $AVEKSA_HOME/archive/aveksa.ear ./*`

7. Extract **aveksaWFArchitect.ear** from the patch:

i. `cd $AVEKSA_HOME/Aveksa_<VersionNumber>_P<PatchNumber>`

ii. `cp aveksaWFArchitect.ear/aveksaWFArchitect.ear $AVEKSA_HOME/archive`

Redeploy the EAR

1. Log in to all host controller machines and stop servers:

```
service aveksa_cluster stop
```

2. Log in to the domain controller machine and stop the AFX server:

```
$AVEKSA_HOME/AFX/afx stop
```

3. Undeploy the existing EAR files:

   i. Issue the following commands:

   ```
   cd $AVEKSA_HOME/wildfly/bin/
   ./jboss-cli.sh -c --controller=<ip_address in cluster
   mode>
   ```

   ii. At the CLI prompt, enter:

   ```
   [domain@localhost:9990]: undeploy aveksa.ear --server-
   groups=img-server-group
   [domain@localhost:9990]: undeploy
   aveksaWFArchitect.ear --server-groups=img-server-group
   ```

   iii. To check that **aveksa.ear** and **aveksaWFArchitect.ear** are undeployed, enter:

   ```
   [domain@localhost:9990]: deployment-info --server-
   group=img-server-group
   ```

   iv. Exit the CLI prompt.

4. Deploy the patch.

   Note: Deploy the updated EAR files on the identified domain controller only.

   i. Deploy the **aveksa.ear** file, where <*ip-address*> is the IP address of the domain controller machine:

   ```
   $AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --
   controller=<ip-address> --command="deploy $AVEKSA_
   HOME/archive/aveksa.ear --server-groups=img-server-
   group"
   ```

   ii. Deploy the **aveksaWFArchitect.ear** file, where <*ip-address*> is the IP address of the domain controller machine:

   ```
   $AVEKSA_HOME/wildfly/bin/jboss-cli.sh -c --
   controller=<ip-address> --command="deploy $AVEKSA_
   HOME/archive/aveksaWFArchitect.ear --server-
   groups=img-server-group"
   ```

5. To check that each EAR is deployed, monitor the log file and verify that the run-once log shows the new patch number and that the database updates are completed.

6. Restart the domain controller server:

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

7. Restart the domain controller and all host controllers:

```
service aveksa_cluster start
```

8. Start the AFX server:

```
$AVEKSA_HOME/AFX/afx start
```

9. Rename the **aveksaWFArchitect.ear** and **aveksa.ear** files in **home/oracle/archive** to include the patch number and date. For example:

```
cp aveksa.ear aveksa_7.1.1_<DateTime>.ear
cp aveksaWFArchitect.ear aveksaWFArchitect_7.1.1_
<DateTime>.ear
```

Run the lockdown_privileges.sh Script

Note: This procedure must be applied to every node in the cluster.

1. Change to the patch directory **$AVEKSA_HOME/Aveksa_<VersionNumber>_P<PatchNumber>/deploy** and copy **lockdown_privileges.sh** to **$AVEKSA_HOME/deploy** in the cluster node:

```
cd $AVEKSA_HOME/Aveksa_<VersionNumber>_
P<PatchNumber>/deploy
cp -rvp lockdown_privileges.sh $AVEKSA_HOME/deploy
```

2. Go to **$AVEKSA_HOME/Aveksa_<VersionNumber>_P<PatchNumber>/etc** in the patch directory and copy **sudoers.snippet** to **$AVEKSA_HOME/deploy>/etc** in the cluster node:

```
cd $AVEKSA_HOME/Aveksa_<VersionNumber>_P<PatchNumber>/etc
cp -rvp sudoers.snippet $AVEKSA_HOME/deploy/etc
```

3. Login as **root** and go to **$AVEKSA_HOME/deploy** to execute the **lockdown_privileges.sh** script:

```
$AVEKSA_HOME/deploy chmod 750 lockdown_privileges.sh
./lockdown_privileges.sh
```

4. Wait for the following confirmation:

```
Privilege lockdown
Change deploy area ownership to root:oinstall
updating file
Privilege lockdown complete
```

Chapter 8: Migrating a WildFly Cluster to TCP

By default, WildFly clusters in RSA Identity Governance and Lifecycle 7.1.1 use TCP for communication between the nodes. Perform the procedures in this chapter to change the communication protocol for an existing cluster from multicast to TCP.

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=""
   ```

After you finish

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

Edit Multicast References in domain.xml

Use the following procedure to edit the **domain.xml** file to change the default communication stack from multicast to TCP.

Procedure

1. Open the file **$AVEKSA_HOME/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. Include TCPPING instead of multicast:
    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. In both places, 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>
```

4. 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 **$AVEKSA_HOME/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 Migrating a Cluster to Use TCP

The following procedures can help you verify that a cluster has been successfully configured to use TCP instead of multicast.

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 **$AVEKSA_HOME/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@103591
8432[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 on page 55.

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 $AVEKSA_HOME/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)
   ```

Troubleshooting

Unable to Authenticate Cluster

In the WildFly server log you may see the following message:

```
HOrnetQ Cluster Security Exception at initial starting up
cluster:
2015-08-13 15:11:49,238 ERROR
[org.apache.activemq.artemis.core.server] (default I/O-6)
AMQ224018: Failed to create session: ActiveMQSecurityException
[errorType=SECURITY_EXCEPTION message=AMQ119031: Unable to
validate user]
```

To resolve this issue, edit the **domain.xml** located under **${AVEKSA_WILDFLY_HOME}/domain/configuration**.

Look for **messaging subsystem** under profile **full-ha**. Under the cluster, ensure that the user and password are configured correctly:

```
<subsystem xmlns="urn:jboss:domain:messaging-activemq:1.0">
                <server name="default">
                    <security enabled="false"/>
                    <cluster user="guest" password="guest"/>
```

No Resource Definition is Registered for Address

When you see this message in the server log, you tried to update the secure session cookie value through the application UI by clicking **Admin** > **System**, and selecting the **Security** tab. In a clustered environment you cannot use the application UI to modify this setting. You must run a WildFly CLI command on the domain controller machine. For more information on setting session cookie see "Set Secure Cookies".

Permission Errors in a Cluster Environment

You may see permission errors like the one shown below when trying to start RSA Identity Governance and Lifecycle using the **aveksa_cluster** service.

```
JBAS014922: Directory $AVEKSA_HOME/wildfly/domain/servers/img-
server-1/data/content/0c is not writable
java.io.FileNotFoundException: $AVEKSA_
HOME/wildfly/domain/servers/img-server-1/log/server.log
(Permission denied)
```

This is likely because RSA Identity Governance and Lifecycle was initially started as **root** rather than the **oracle** user and now the **oracle** user does not have the needed write permission on certain files.

You can resolve this by running the following command to change the ownership to **oracle**:

```
chown -hR oracle $AVEKSA_HOME/wildfly/domain
```

RSA Identity Governance and Lifecycle Does Not Automatically Startup After a Reboot

You can configure the **aveksa_cluster service** to start up at boot time, but it does not start if Oracle hasn't started yet. In this case, it passes itself off to the **aveksa_watchdog** service, which is not a configured service on a cluster node.

You may see the following message logged during boot up:

```
<notice – Aug 18 15:23:11.568139000> aveksa_cluster start
Cannot connect to the database. The watchdog will start the
server when it can connect to the database.
<notice – Aug 18 15:23:14.958043000>
'aveksa_cluster start' exits with status 1
```

You can resolve this by manually starting RSA Identity Governance and Lifecycle following a reboot:

```
service aveksa_cluster start
```

RSA Identity Governance and Lifecycle Stops Responding Because of Default Limits

RSA Identity Governance and Lifecycle can stop responding or report errors when the default soft and hard limits are low for assets such as currently open file descriptors and the maximum number of file handlers. You can configure these limits to help keep the server stable and functional.

Note: The appliance, local AFX, and remote components (AFX and Database) each have limits to configure.

For recommended general settings, apply the following limits:

- nofile - greater than 1024 soft, greater than 65536 hard
- nproc - greater than 16384 soft, greater than 16384 hard
- process stack segment size - greater than 10240 KB soft, 10240 to 32768 KB hard

Procedure

1. Execute the following command in your editor:

   ```
   vi /etc/security/limits.conf
   ```

2. Scroll to the value column you want to edit and insert the new values as necessary.
3. Finalize the changes with the **:wq** command.