RSA Identity Governance and Lifecycle

Solution Integration Guide

Configuring WildFly Clustering

RSA Identity Governance and Lifecycle Version 7.0.1

Contact Information

Go to the RSA corporate website for regional Customer Support telephone and fax numbers: www.emc.com/domains/rsa/index.htm.

Trademarks

License agreement

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 EMC software described in this publication requires an applicable software license.

EMC 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." EMC CORPORATION 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.

March 2017

Table of Contents

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, where 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.

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 7.0.1

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 in the following configurations:

- RSA Identity Governance and Lifecycle Hardware Appliance – Where the hardware appliance will be converted to a domain controller in the cluster and host the RSA Identity Governance and Lifecycle database.
- RSA Identity Governance and Lifecycle Soft Appliance – Where multiple soft appliance installations will participate in a cluster configuration pointing to a remote database setup.

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

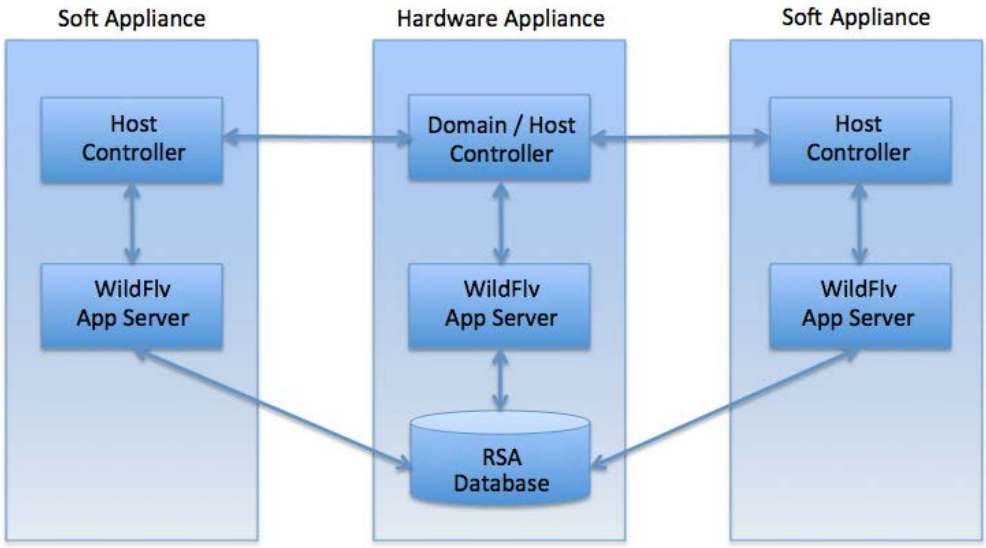
Caveats Understanding the Doc

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

RSA Identity Governance and Lifecycle Hardware Appliance

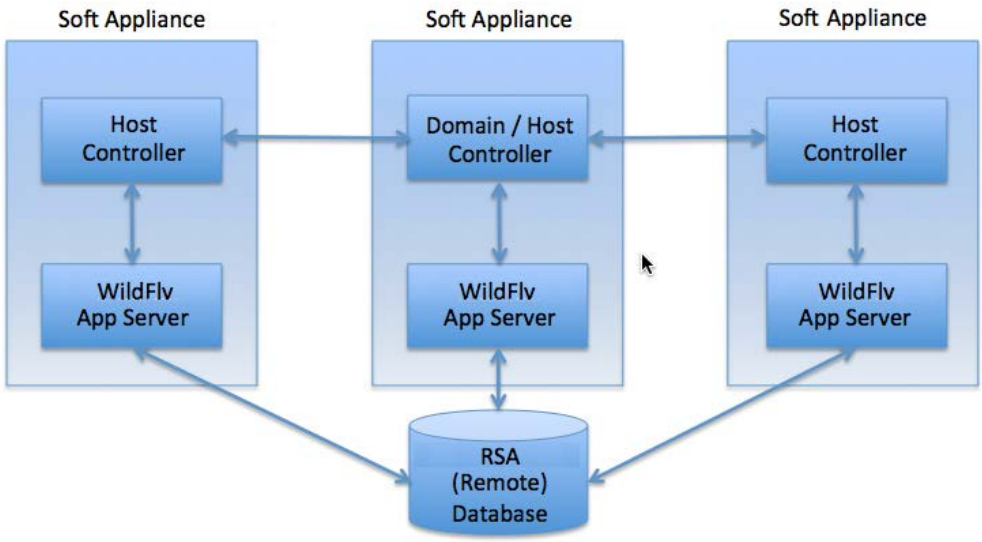
In this configuration, an RSA Identity Governance and Lifecycle hardware appliance has been purchased and the business requirement has been identified to add additional application server nodes (setup in a cluster configuration) as part of the solution. In this scenario, the hardware appliance hosts the database, domain controller for the cluster, and an application server. Additional soft appliances are added to the configuration as slave nodes in the cluster. These soft appliances will only host the application servers and participate in the cluster.

The following illustration depicts an RSA Identity Governance and Lifecycle deployment in a clustered application server environment using a hardware appliance:



RSA Identity Governance and Lifecycle Soft Appliance

In this configuration, the soft appliances are set up in a clustered environment leveraging a remote database setup. One of the soft appliances 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 soft appliances connecting to a remote database deployment:



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 the table in the following section.
- A database environment that is prepared for running RSA Identity Governance and Lifecycle 7.0.x – 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 7.0.x appliance model.
- For the additional nodes that will participate in the cluster, RSA Identity Governance and Lifecycle 7.0.x is installed with the remote database option.

Machine Information

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

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 as described in this section.

Un-deploy 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 EARs (aveksa.ear, aveksaWFArchitect.ear) deployed in this configuration are no longer needed in a cluster configuration. To avoid accidental deployment and conflict with these EARs and to conserve space, un-deploy the EARs.

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 shown below:

```
Service: aveksa_server status

/home/oracle/wildfly/bin/jboss-cli.sh -c --command="undeploy aveksa.ear"
/home/oracle/wildfly/bin/jboss-cli-sh –c ––command="undeploy aveksaWFArchitect.ear"
```

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

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --command="deployment-info --
name=aveksa.ear"

JBAS014807: Management resource '[("deployment" => "aveksa.ear")]' not found

/home/oracle/wildfly/bin/jboss-cli.sh -c --command="deployment-info --
name=aveksaWFArchitect.ear"

JBAS014807: 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 all non-clustered standalone instances of RSA Identity Governance and Lifecycle. You need to stop and disable these services before you create clustered domain instances. If you plan to implement AFX as part of this setup, the AFX services can only be installed and running on one of the machines.

Log in as `root` on all identified cluster nodes and run the following commands:

AFX will <u>not</u> be deployed/running on the server

```
# Stop the services
service aveksa_watchdog stop
service aveksa_server stop

# Unregister the services
chkconfig aveksa_watchdog off
chkconfig aveksa_server off

# remove executable permissions from service scripts
chmod 400 /etc/init.d/aveksa_watchdog
chmod 400 /etc/init.d/aveksa_server
```

AFX will be deployed/running on the server

```
# Stop the services
service aveksa_watchdog stop
service aveksa_server stop

# Unregister the services
chkconfig aveksa_watchdog off
chkconfig aveksa_server off

# remove executable permissions from service scripts
chmod 400 /etc/init.d/aveksa_watchdog
chmod 400 /etc/init.d/aveksa_server
```

Login as <AFX_User> on all identified cluster nodes and run the following commands (where AFX_user is a non-root account that owns the AFX server files):

AFX will not be deployed/running on the server

```
# Stop the services
service afx_server stop

# Unregister the services
chkconfig afx_server off

# remove executable permissions from service scripts
chmod 400 /etc/init.d/afx_server
```

AFX will be deployed/running on the server

```
# Stop the services
service afx_server stop
```

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. Set the permissions on `aveksa_cluster` using the following command:

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

3. Edit `/etc/init.d/aveksa_cluster` using a text editor.
4. 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`. For example:

```
Post-Upgrade Cleanup of the /home/oracle Directory on an Appliance#NODE_TYPE=DOMAIN
NODE_TYPE=SLAVE
```

   If the server is the domain controller, then do not change the `NODE_TYPE`.

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

```
DOMAIN_MASTER="10.101.250.7"
```

6. 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
```

7. Set the **`JMS_MULTICAST_IP`** variable to the IP address reserved for multicast on this cluster. The nodes in the cluster should communicate over a unique multicast address. If you have multiple WildFly clusters with the default messaging-group multicast address of 231.7.7.7, the clusters will conflict with each other. Change the multicast address to a unique address. Check with your network administrator for the address. For example:

```
JMS_MULTICAST_IP=231.7.7.7
```

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

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 perm 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 perm memory. For example:

```
Recommended Cluster Options Settings (in MB)
WILDFLY_HEAP_MEM : 9686
WILDFLY_PERM_MEM : 1709

Calculations based off of
ORACLE_MEM : 31344 MB for installed Database system on this machine
AFX_MEM : 3072 MB for installed AFX system on this machine
OS_MEM= : 2411  MB for general operating system overhead
```

Edit `/etc/init.d/aveksa_cluster` in a text editor and set the **`WILDFLY_HEAP_MEM`** and **`WILDFLY_PERM_MEM`** variables to the recommended values returned by the above command. For example, using the lowest common setting :

```
WILDFLY_HEAP_MEM=9686
WILDFLY_PERM_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 the UDP port 9876 to the operating systems firewall setup.

The default port for multicast messaging in WildFly is 9876.

SuSE Environment

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

FW_SERVICES_EXT_UDP="9876"
```

Red Hat Environment

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

-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --dport 9876 –j ACCEPT
```

The cluster nodes need to communicate via JGroups, a reliable multicasting toolkit for UDP. These ports are set in the `socket-binding-group` in the `domain.xml` file. The current port default settings are 45688 and 55200. Add these ports to the respective operating system's firewall setting.

SuSE Environment

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

FW_SERVICES_EXT_UDP="9876 45688 55200"
```

Red Hat Environment

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

-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --dport 45688 –j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m state --state NEW -m udp --dport 55200 –j ACCEPT
```

Messaging requires TCP port 8080 to be open. Open the port.

SuSE Environment

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

FW_SERVICES_EXT_TCP="22 8080 8443 8444"
```

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 - udp --dport 8080 –j ACCEPT
```

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

SuSE Environment

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

Red Hat Environment

```
/etc/init.d/iptables restart
```

Disable Appliance Mode

Perform this step only if you are using an appliance for the master server, where the Oracle database was provided by RSA and installed on the same machine.

If you are converting a hardware appliance 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;
```

Set Up and Configure the Domain Controller

This section is for setting up and configuring the machine that has been identified as the ‘domain controller’ for the cluster. Note: Only a single machine can be identified to be the “domain controller” in the cluster. Complete this section before configuring the “host controllers”

Create 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 `/home/oracle/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>
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 '/home/oracle/wildfly-
8.2.0.Final/standalone/configuration/mgmt-users.properties'
Added user 'AveksaClusterAdmin' to file '/home/oracle/wildfly-
8.2.0.Final/domain/configuration/mgmt-users.properties'
Added user 'AveksaClusterAdmin' with groups to file '/home/oracle/wildfly-
8.2.0.Final/standalone/configuration/mgmt-groups.properties'
Added user 'AveksaClusterAdmin' with groups to file '/home/oracle/wildfly-
8.2.0.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" />
```

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:

JBAS015267: Password must have at least 1 non-alphanumeric symbol. 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 `/home/oracle/wildfly/domain/configuration` and make the following changes:

Configure Server Group

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

```
<server-groups>
        <server-group name="img-server-group" profile="full-ha">
            <jvm name="default">
                <heap size="1024m" max-
size="${jboss.memory.max.heap.size}"/>
                <permgen size="256m" max-
size="${jboss.memory.max.permgen.size}"/>
                <jvm-options>
                    <option value="-server"/>
                </jvm-options>
            </jvm>
            <socket-binding-group ref="full-ha-sockets"/>
        </server-group>
</server-groups>
```

Configure HornetQ Cluster Credentials

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

```
<subsystem xmlns="urn:jboss:domain:messaging:2.0">
```

Replace the password line (2 rows below) to look like the following:

```
<cluster-password>some_password</cluster-password>
```

And add the following right below the password line:

```
<cluster-user>some_username</cluster-user>
```

Note: The some_username and some_password that you add here are not used in the subsequent configuration of the host controllers, but you should be record them as part of your implementation documentation.

Configure Host Configuration

Edit the `host.xml` file located at `/home/oracle/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>
      <engine enabled-cipher-suites="TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA"/>
      <keystore path="/home/oracle/keystore/server.keystore" keystore-
password="Av3k5a15num83r0n3" alias="server" key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/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>
      <engine enabled-cipher-suites= "TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA"/>
      <keystore path="/home/oracle/keystore/aveksa.keystore" keystore-
password="Av3k5a15num83r0n3" alias="server" key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/keystore/aveksa.keystore" keystore-
password="Av3k5a15num83r0n3"/>
   </authentication>
  </security-realm>
```

Open Firewall Port for Management Interface Port

The domain controller communicates with the host controllers in the cluster using the management interface port 9999. This port needs to be accessible on the domain controller machine. Perform the following operations to make port 9999 accessible.

SuSE Environment

As `root`, edit `/etc/sysconfig/SuSEfirewall2` and find the `FW_SERVICES_EXT_TCP` setting. Add **9999** to the end of the setting as shown in the following example.

```
FW_SERVICES_EXT_TCP="22 8080 8443 8444 9999"
```

Run these commands after saving the file to apply the changes:

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

Red Hat Environment

As `root`, edit `/etc/sysconfig/iptables` and add the following line to open port 9999, as shown in the following example:

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

After saving the file, run the following command to apply the changes:

```
/etc/init.d/iptables restart
```

Start the Domain Controller

Log in as `oracle` on the domain controller machine. Start the domain controller by executing the following command:

If it does not exist, you need to create the log folder ($AVEKSA_WILDFLY_HOME/domain/log/) before running the command.

```
service aveksa_cluster start
```

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

```
=========================================================================
  JBoss Bootstrap Environment
  JBOSS_HOME: /home/oracle/wildfly
  JAVA: /etc/alternatives/java_sdk_1.7.0/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 8.2.0.Final "Tweek" 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)
JBAS015874: WildFly 8.2.0.Final "Tweek" (Host Controller) started in 10164ms -
Started 44 of 46 services (13 services are lazy, passive or on-demand)
```

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>
```

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_USERNAME=AveksaClusterAdmin -DDOMAIN_USER_PASSWORD=Aveksa123"
```

The following script uses the configuration in `/home/oracle/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
```

You may see exceptions at this point. If these are seen, they are a one-time load error exception that is taken care of with the restart when the ear is deployed. You may also see a message to “Restart WildFly after running this command.” Restarting after the ear is deployed will also restart WildFly.

Deploy RSA Identity Governance and Lifecycle

The two ear files, `aveksa.ear` and `aveksaWFArchitect.ear` are deployed on the domain controller machine. The domain controller then propagates each ear to all the servers.

The ears are located in `/tmp/repackaged_ear_dir`. The ears are deployed to the server-group `img-server-group`.

To deploy the aveksa.ear, log in as the `oracle` user and execute the following command:

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --controller=localhost

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

To check that the EAR is deployed, monitor the log file and verify that the run-
once log shows the patch number and the database updates are performed.
```

Setup and Configure the Host Controllers

> 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 `/home/oracle/wildfly/domain/configuration` folder, run the following commands to copy the `host-slave.xml` to `host.xml`:

```
cd /home/oracle/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 display in the logs and in the WildFly management console.

Note 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:2.1">
```

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 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`.

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="/home/oracle/keystore/server.keystore" keystore-
password="Av3k5a15num83r0n3" alias="server" key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/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="/home/oracle/keystore/aveksa.keystore" keystore-
password="Av3k5a15num83r0n3" alias="server" key-password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/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. 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 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>
```

Start the Host Controllers

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

> If it does not exist, you need to create the log folder ($AVEKSA_WILDFLY_HOME/domain/log/) before running the command.

```
service aveksa_cluster start
```

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) JBAS010918: Registered remote slave host "vm-adap-42", WildFly
8.2.0.Final "Tweek"
```

If you get this time out message, open port 9999 on the domain controller machine.

```
[Host Controller] 15:34:38,040 WARN [org.jboss.as.host.controller]
(Controller Boot Thread) JBAS010900: Could not connect to remote domain
controller at remote://10.101.249.10:9999 -- java.net.ConnectException:
JBAS012144: Could not connect to remote://10.101.249.10:9999. The
connection timed out
```

Operations and Management

Startup Checklist

After the domain and host controller servers are running look for the "***Bridge Cluster.. is connected***" message in the server.log. This will confirm the HornetQ messing is working in a cluster.

The server.log is found in `/home/oracle/wildfly/domain/servers/<server-name>/log`. For example: `/home/oracle/wildfly/domain/servers/img-server-one/log`. An example of the messaging cluster connection is shown below.

```
2015-08-03 10:06:56,727 INFO  [org.hornetq.core.server] (Thread-19 (HornetQ-server-
HornetQServerImpl::serverUUID=f484f65c-39e0-11e5-ac09-a336eec6a45d-2142161218))
HQ221027: Bridge ClusterConnectionBridge@6e29f674 [name=sf.my-cluster.f7baf34b-39e0-
11e5-950b-113c8b43d750, queue=QueueImpl[name=sf.my-cluster.f7baf34b-39e0-11e5-950b-
113c8b43d750, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=f484f65c-39e0-11e5-ac09-a336eec6a45d]]@acceae5
targetConnector=ServerLocatorImpl (identity=(Cluster-connection-
bridge::ClusterConnectionBridge@6e29f674 [name=sf.my-cluster.f7baf34b-39e0-11e5-
950b-113c8b43d750, queue=QueueImpl[name=sf.my-cluster.f7baf34b-39e0-11e5-950b-
113c8b43d750, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=f484f65c-39e0-11e5-ac09-a336eec6a45d]]@acceae5
targetConnector=ServerLocatorImpl
[initialConnectors=[TransportConfiguration(name=http-connector, factory=org-hornetq-
core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-upgrade-
endpoint=http-acceptor&host=10-101-249-179&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@500825358[nodeUUID=f484f65
c-39e0-11e5-ac09-a336eec6a45d, connector=TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory)
?port=8080&host=10-101-249-178&http-upgrade-endpoint=http-acceptor&http-upgrade-
enabled=true, address=jms, server=HornetQServerImpl::serverUUID=f484f65c-39e0-11e5-
ac09-a336eec6a45d])) [initialConnectors=[TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-
upgrade-endpoint=http-acceptor&host=10-101-249-179&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]] is connected
```

Also on the domain's server log you would see messages similar to the following as you turn on the slave machines. For example:

```
[Server:domain1.aveksa.local] 12:23:10,064 INFO  [org.hornetq.core.server] (Thread-
21 (HornetQ-server-HornetQServerImpl::serverUUID=336b7a38-4295-11e5-ad44-
3bfe8e031bc4-2129622477)) HQ221027: Bridge ClusterConnectionBridge@5735dbfe
[name=sf.my-cluster.cce7e915-429c-11e5-9bf4-e1c1926cbcd1,
queue=QueueImpl[name=sf.my-cluster.cce7e915-429c-11e5-9bf4-e1c1926cbcd1,
postOffice=PostOfficeImpl [server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-
ad44-3bfe8e031bc4]]@10fe2d6d targetConnector=ServerLocatorImpl (identity=(Cluster-
connection-bridge::ClusterConnectionBridge@5735dbfe [name=sf.my-cluster.cce7e915-
429c-11e5-9bf4-e1c1926cbcd1, queue=QueueImpl[name=sf.my-cluster.cce7e915-429c-11e5-
9bf4-e1c1926cbcd1, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-ad44-
3bfe8e031bc4]]@10fe2d6d targetConnector=ServerLocatorImpl
[initialConnectors=[TransportConfiguration(name=http-connector, factory=org-hornetq-
core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-upgrade-
endpoint=http-acceptor&host=192-168-21-11&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@712816214[nodeUUID=336b7a3
8-4295-11e5-ad44-3bfe8e031bc4, connector=TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory)
?port=8080&host=192-168-21-10&http-upgrade-endpoint=http-acceptor&http-upgrade-
enabled=true, address=jms, server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-
ad44-3bfe8e031bc4])) [initialConnectors=[TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-
upgrade-endpoint=http-acceptor&host=192-168-21-11&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]] is connected
[Host Controller] 12:27:15,369 INFO  [org.jboss.as.domain] (Host Controller Service
Threads - 31) JBAS010918: Registered remote slave host "slave2", WildFly 8.2.0.Final
"Tweek"
[Server:domain1.aveksa.local] 12:27:36,690 INFO  [org.hornetq.core.server] (Thread-
27 (HornetQ-server-HornetQServerImpl::serverUUID=336b7a38-4295-11e5-ad44-
3bfe8e031bc4-2129622477)) HQ221027: Bridge ClusterConnectionBridge@a860fce
[name=sf.my-cluster.c76ab720-4298-11e5-8e98-91a8ffd87111,
queue=QueueImpl[name=sf.my-cluster.c76ab720-4298-11e5-8e98-91a8ffd87111,
postOffice=PostOfficeImpl [server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-
ad44-3bfe8e031bc4]]@6777a2c7 targetConnector=ServerLocatorImpl (identity=(Cluster-
connection-bridge::ClusterConnectionBridge@a860fce [name=sf.my-cluster.c76ab720-
4298-11e5-8e98-91a8ffd87111, queue=QueueImpl[name=sf.my-cluster.c76ab720-4298-11e5-
8e98-91a8ffd87111, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-ad44-
3bfe8e031bc4]]@6777a2c7 targetConnector=ServerLocatorImpl
[initialConnectors=[TransportConfiguration(name=http-connector, factory=org-hornetq-
core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&host=192-168-21-12&http-
upgrade-endpoint=http-acceptor&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@712816214[nodeUUID=336b7a3
8-4295-11e5-ad44-3bfe8e031bc4, connector=TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory)
?port=8080&host=192-168-21-10&http-upgrade-endpoint=http-acceptor&http-upgrade-
enabled=true, address=jms, server=HornetQServerImpl::serverUUID=336b7a38-4295-11e5-
ad44-3bfe8e031bc4])) [initialConnectors=[TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory)
?port=8080&host=192-168-21-12&http-upgrade-endpoint=http-acceptor&http-upgrade-
enabled=true], discoveryGroupConfiguration=null]] is connected
```

Log File Location and Properties

Location

The application server log files are located on each machine at `/home/oracle/wildfly/domain/servers/<server-name>/log` directory, where `<server-name>` is the name of the application server. In the previous example, it is `img-server-1`.

The domain/host controller log files are located in the `/home/oracle/wildfly/domain/log` directory.

Log4J Property File

On a cluster setup the `aveksa-log4j.properties` file is located in `/home/oracle/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.

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 all server nodes are connected in the cluster

In the log file look for a message similar to the following:

```
2015-07-09 07:09:16,789 INFO
[org.infinispan.remoting.transport.jgroups.JGroupsTransport] (Incoming-1,shared=udp)
ISPN000094: Received new cluster view: [master:img-server-1/ejb|1] (2) [master:img-
server-1/ejb, acmr620-07:img-server-2/ejb]
```

where

- ***[master:img-server-1/ejb|1] (2)*** indicates 2 nodes are connected in the cluster.
- ***[master:img-server-1/ejb, acmr620-07:img-server-2/ejb]*** is a comma separated list of nodes connected in the cluster.

Confirm JMS messaging is working

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

```
2015-07-09 07:14:21,719 INFO  [org.hornetq.core.server] (Thread-1 (HornetQ-server-
HornetQServerImpl::serverUUID=ece149af-25f9-11e5-ab43-1d33c64eb3be-1144939731))
HQ221027: Bridge ClusterConnectionBridge@4b496d61 [name=sf.my-cluster.081e723a-25fb-
11e5-b5e4-bb0b73dd86b3, queue=QueueImpl[name=sf.my-cluster.081e723a-25fb-11e5-b5e4-
bb0b73dd86b3, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=ece149af-25f9-11e5-ab43-
1d33c64eb3be]]@41ac21e4 targetConnector=ServerLocatorImpl (identity=(Cluster-
connection-bridge::ClusterConnectionBridge@4b496d61 [name=sf.my-cluster.081e723a-
25fb-11e5-b5e4-bb0b73dd86b3, queue=QueueImpl[name=sf.my-cluster.081e723a-25fb-11e5-
b5e4-bb0b73dd86b3, postOffice=PostOfficeImpl
[server=HornetQServerImpl::serverUUID=ece149af-25f9-11e5-ab43-
1d33c64eb3be]]@41ac21e4 targetConnector=ServerLocatorImpl
[initialConnectors=[TransportConfiguration(name=http-connector, factory=org-hornetq-
core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-upgrade-
endpoint=http-acceptor&host=10-101-250-27&http-upgrade-enabled=true],
discoveryGroupConfiguration=null]]::ClusterConnectionImpl@1691175884[nodeUUID=ece149
af-25f9-11e5-ab43-1d33c64eb3be, connector=TransportConfiguration(name=http-
connector, factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory)
?port=8080&host=10-101-250-11&http-upgrade-endpoint=http-acceptor&http-upgrade-
enabled=true, address=jms, server=HornetQServerImpl::serverUUID=ece149af-25f9-11e5-
ab43-1d33c64eb3be])) [initialConnectors=[TransportConfiguration(name=http-connector,
factory=org-hornetq-core-remoting-impl-netty-NettyConnectorFactory) ?port=8080&http-
upgrade-endpoint=http-acceptor&host=10-101-250-27&http-upgrade-enabled=true],
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 the "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>:9999
```

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:/home/oracle/wildfly/bin # ./jboss-cli.sh -c --
controller=10.101.250.7:9999
[domain@10.101.250.7:9999 /] /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@10.101.250.7:9999 /] exit
```

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-handler=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=FILE/: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:/home/oracle/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-handler=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=FILE/: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, the `rotate-size` is `300m` and the `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:/home/oracle/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
```

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:
    1. Go to RSA Link, then click Log In and enter your user name and password.
    2. Click RSA Identity Governance and Lifecycle.
    3. Click Downloads > RSA Identity Governance and Lifecycle 7.0.1
    4. Click on Additional Downloads.
    5. Click Access Certification Manager.
    6. Click Download Software (may take a few moments to find downloads).
    7. Download the patch or upgrade file:
        Aveksa_<VersionNumber>_P<PatchNumber>.tar.gz
2. Save the patch or upgrade file to the domain controller machine.
3. Ensure that the server running on the domain controller machine is configured as the Systems Operation Node.
4. Make sure the application is running (so you can undeploy and redeploy the two EAR files: the `aveksa.ear`, and the `aveksaWFArchitect.ear` files).
5. Ensure all nodes are running.

Procedure

1. Log into the domain controller server as the `oracle` user.
2. Stage the currently installed EAR files:

```
cd /home/oracle/deploy

At the CLI prompt, enter:

./customizeACM.sh –c

Follow the prompts to unpack the EAR to /tmp/customizeACM.
```

3. Untar the patch or upgrade file:

```
tar zvxf Aveksa_<VersionNumber>_P<PatchNumber>.tar.gz
```

   The files are unpacked into a new directory named Aveksa_<VersionNumber>_P<PatchNumber>.

4. Use the following command to change directories to the new directory created in step 3:

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

5. Use the following command to copy all untarred files under aveksa.ear to the /tmp/customizeACM directory:

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

6. Zip files in /tmp/customizeACM by running the following commands:

```
cd /tmp/customizeACM
jar cvf /home/oracle/archive/aveksa.ear *
```

7. Undeploy the existing ear:

```
cd /home/oracle/wildfly/bin/
./jboss-cli.sh -c --controller=localhost

At the CLI prompt, enter:

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

To check that aveksa.ear is undeployed, enter:

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

Exit the CLI prompt.
```

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

```
service aveksa_cluster stop
```

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

```
/home/oracle/AFX/afx stop
```

10. Deploy the patch or upgrade:

Deploy the updated ear file on the identified domain controller only.

Deploy the `aveksa.ear` file:

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

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

Deploy the `aveksaWFArchitect.ear` file:

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

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

11. 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.

12. Restart the domain controller server by running the following commands:

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

13. Start the AFX server.

```
/home/oracle/AFX/afx start
```

14. Restart the domain controller and all host controllers.

```
service aveksa_cluster start
```

15. Rename the `aveksa.ear` file in /home/oracle/archive to include the patch number and date.

```
cd /home/oracle/archive
cp archive aveksa.ear aveksa_7_0.1_P01-2016-Oct-2018.ear
```

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
./uninstallAFX.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.

Hardware Appliance

If the domain controller is a hard appliance (the database server is on the same machine as the domain controller), then you can back up by running the following command on the domain controller server as the oracle user:

```
sudo -u oracle $AVEKSA_HOME/database/DBA/AVDB/scripts/AVDB_Export_AVUSER.sh -t
<filename>
```

For example:

```
sudo -u oracle $AVEKSA_HOME/database/DBA/AVDB/scripts/AVDB_Export_AVUSER.sh -t 2015-
08-19-17-30-00
```

To restore the database, log in as root to stop RSA Identity Governance and Lifecycle on all nodes, and then restart the database on the domain controller:

```
# First do this on all cluster nodes
service aveksa_cluster stop

# Then do this only on the domain controller
service aveksa_cluster stopdb
service aveksa_cluster startdb
```

As the `oracle` user, run the following command on the domain controller server to restore the database:

```
sudo -u oracle $AVEKSA_HOME/database/DBA/AVDB/scripts/AVDB_Import_AVUSER.sh -t
<filename>
```

For example:

```
sudo -u oracle $AVEKSA_HOME/database/DBA/AVDB/scripts/AVDB_Import_AVUSER.sh -t 2015-
08-19-17-30-00
```

See the Identity Governance and Lifecycle product documentation for more information on the backup and restore scripts.

Soft Appliance

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

Upgrading Your Cluster

This section explains how to upgrade RSA Identity Governance and Lifecycle 7.0 to the base version 7.0.1 without patches installed on a WildFly application server cluster.

Prerequisites

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:

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

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

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

   ```
   /home/oracle/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

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

   ```
   /home/oracle/AFX/afx stop
   ```

Stop Cluster Service

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

```
service aveksa_cluster stop
```

Copy Installer Binaries

Copy the 7.0.1 GA installer binaries to `/tmp/aveksa/staging` on all machines.

Upgrade Domain Controller

The following sections explain how to upgrade the domain controller.

Upgrade JDK

Log in as root on the domain controller machine and run the following commands:

```
cd /tmp/aveksa/staging/deploy
./installJDK.sh $JAVA_HOME
```

Install Support Files

1. Copy `installSupportFiles.sh` script from the distribution to `/tmp/aveksa/staging/deploy` folder.
2. Log in as root and run the following commands:

   ```
   cd /tmp/aveksa/staging/deploy
   chmod 750 installSupportFiles.sh
   ./installSupportFiles.sh
   ```

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 /home/oracle/wildfly/modules/com/aveksa/jdbc/main
```

Configure Oracle JDBC

Run the following commands:

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

cp ojdbc6.jar /home/oracle/wildfly/modules/com/oracle/main

cd /home/oracle/wildfly/modules/com/oracle/main

rm ojdbc5.jar
```

Edit `module.xml` and replace reference to ojdbc5.jar with ojdbc6.jar

Configure RSA Crypto Module

1. Run the following commands:

   ```
   cd /home/oracle/wildfly/modules/com/rsa

   rm -rf cryptoj cryptojce cryptojcommon

   mkdir main

   cd /tmp/aveksa/staging/deploy/wildfly-conf/lib

   cp cryptojce.jar /home/oracle/wildfly/modules/com/rsa/main

   cp cryptojcommon.jar /home/oracle/wildfly/modules/com/rsa/main

   cp jcmFIPS.jar /home/oracle/wildfly/modules/com/rsa/main
   ```

```
cd /home/oracle/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 `/home/oracle/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="/home/oracle/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:

```
/home/oracle/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:

   ```
   /home/oracle/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
```

Migration

Access the main page of the server that is designated as a Systems Operation Node (SON). You will see the Initialization Status page 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.

Upgrade JDK

Log in as root and run the following commands:

```
cd /tmp/aveksa/staging/deploy
./installJDK.sh $JAVA_HOME
```

Install Support Files

1. Obtain the installSupportFiles.sh script from the Downloads section of RSA Link, at https://community.rsa.com/community/products/governance-and-lifecycle.
2. Copy the installSupportFiles.sh script to `/tmp/aveksa/staging/deploy` folder.
3. Log in as root and run the following commands:

   ```
   cd /tmp/aveksa/staging/deploy
   chmod 750 installSupportFiles.sh
   ./installSupportFiles.sh
   ```

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
/home/oracle/wildfly/modules/com/aveksa/jdbc/main
```

Configure Oracle JDBC

Run the following commands:

```
cd /tmp/aveksa/staging/deploy/oracle
cp ojdbc6.jar /home/oracle/wildfly/modules/com/oracle/main
cd /home/oracle/wildfly/modules/com/oracle/main
rm ojdbc5.jar
```

Edit `module.xml` and replace reference to `ojdbc5.jar` with `ojdbc6.jar`.

Configure RSA Crypto Module

1. Run the following commands:

   ```
   cd /home/oracle/wildfly/modules/com/rsa
   rm -rf cryptoj cryptojce cryptojcommon
   mkdir main
   ```

```
cd /tmp/aveksa/staging/deploy/wildfly-conf/lib
cp cryptojce.jar /home/oracle/wildfly/modules/com/rsa/main
cp cryptojcommon.jar
/home/oracle/wildfly/modules/com/rsa/main
cp jcmFIPS.jar /home/oracle/wildfly/modules/com/rsa/main
cd /home/oracle/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>
```

Start the Slave Server

1. Log in as oracle, and run the following command:

   `service aveksa_cluster start`

2. Access the application page to verify that RSA Identity Governance and Lifecycle.

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: `/home/oracle/AFX/afx stop`

Import Connectors

Download the AFX-<product-version>-Standard-Connectors.zip file for this RSA Identity Governance and Lifecycle release from 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: `/home/oracle/AFX/afx start`

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.hornetq.core.server] (default I/O-1) HQ224018:
Failed to create session:
HornetQClusterSecurityException[errorType=CLUSTER_SECURITY_EXCEPTION
message=HQ119099: Unable to authenticate cluster user: HORNETQ.CLUSTER.ADMIN.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 `hornetq-server` ensure `cluster-user` and `cluster-password` properties are configured. If they are missing add them by choosing any name for `user` and value for `password`.

The following is an example of this configuration:

```
<subsystem xmlns="urn:jboss:domain:messaging:2.0">
                <hornetq-server>
                    <cluster-user>wfadmin</cluster-user>
                    <cluster-password>testing2@</cluster-password>
                    <journal-file-size>102400</journal-file-size>
                    <connectors>
```

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 on page 26.

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 /home/oracle/wildfly/domain/servers/img-server-
1/data/content/0c is not writable
java.io.FileNotFoundException: /home/oracle/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 /home/oracle/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
```