

**RSA** IDENTITY GOVERNANCE AND LIFECYCLE

## **Configuring WildFly Clustering**

**V7.1**

**Revision 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](http://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.

Copyright © 2018 Dell Inc. or its subsidiaries. All Rights Reserved.

February 2018

Revised: July 2018

## Contents

---

Preface .....	6
Documentation Set .....	6
Support and Service .....	6
<b>Revision History .....</b>	<b>7</b>
<b>Chapter 1: Overview .....</b>	<b>8</b>
Supported Platform Versions .....	8
Audience .....	8
Supported Configurations .....	8
About Using this Document .....	8
RSA Identity Governance and Lifecycle Software Bundle .....	9
<b>Chapter 2: Staging the Environment .....</b>	<b>10</b>
Prerequisites .....	10
Machine Information .....	10
Before You Begin .....	11
Undeploy aveksa.ear and aveksaWFArchitect.ear .....	11
Stop and Disable Services .....	12
Configure Services for the Cluster .....	13
Add and Register the Cluster Aware Service .....	13
Determine and Configure JMS Multicast Settings for the Cluster .....	14
Determine and Configure JVM Memory Settings for the Cluster .....	14
Open Required Ports on all Controllers .....	15
Disable Appliance Mode in Software Bundle Installation .....	16
<b>Chapter 3: Set Up and Configure the Domain Controller .....</b>	<b>18</b>
Create a Management User .....	18
Configure the Domain Configuration File .....	19
Update Management Timeout .....	19
Configure Server Group .....	20
Update JVM Memory Settings .....	20
Configure Cluster Authentication Credentials .....	21
Configure Domain Network Interfaces .....	21
Configure Host Configuration .....	21
Configure Server Name registered in RSA Identity Governance and Lifecycle .....	21
Verify local Domain Controller .....	22
Add Aveksa Security Realms .....	22
Configure Host Network Interfaces .....	23
Update JVM Memory Settings .....	23
Open Firewall Port for Management Interface Port .....	23
SuSE Environment .....	23
Red Hat Environment .....	24
Start the Domain Controller .....	24
Tune WildFly Configuration .....	25
Deploy RSA Identity Governance and Lifecycle .....	27
<b>Chapter 4: Setup and Configure the Host Controllers .....</b>	<b>28</b>

Create Host Configuration File .....	28
Configure Host Configuration File .....	28
Name the Host Controller .....	28
Add management user secret value .....	28
Add Aveksa Security Realms .....	29
Configure Domain Controller settings .....	29
Configure Server Name registered in RSA Identity Governance and Lifecycle .....	30
Configure Host Network Interfaces .....	30
Update JVM Memory Settings .....	30
Start the Host Controllers .....	31
<b>Chapter 5: Operations and Management .....</b>	<b>32</b>
Log File Location and Properties .....	32
Location .....	32
Log4J Property File .....	32
Server Operations Node (SON) .....	32
Assign a Server Operations Node .....	32
Reassigning node as next SON .....	32
Cluster Checklist .....	32
Confirm JMS messaging is working .....	32
Confirm server nodes in UI .....	33
Authentication Sources .....	34
Set Secure Cookies .....	34
Configure Logging .....	35
Initial Logging Setup .....	35
Modify Logging .....	37
Uninstall a Cluster .....	37
Backup and Restore .....	38
<b>Chapter 6: Upgrading Your Cluster .....</b>	<b>39</b>
Before You Begin .....	39
Copy Installation Files .....	39
Backup WildFly and AFX .....	39
Repackage EAR with Customizations .....	39
Undeploy aveksa.ear .....	40
Stop And Enable Services .....	41
Stop the AFX Server .....	41
Install Support Files .....	41
Upgrade Domain Controller .....	42
Deploy EAR files .....	42
Schema Migration .....	42
Upgrade Host Controllers .....	43
Upgrade AFX .....	43
Import Connectors .....	43
Start the AFX Server .....	44
<b>Chapter 7: Installing a Patch or Upgrade for RSA Identity Governance and Lifecycle .....</b>	<b>45</b>
Before You Begin .....	45
Procedure .....	46
Run the lockdown_privileges.sh Script .....	48

<b>Appendix A: Troubleshooting</b> .....	<b>49</b>
Unable to Authenticate Cluster .....	49
No Resource Definition is Registered for Address .....	49
Permission Errors in a Cluster Environment .....	49
RSA Identity Governance and Lifecycle Does Not Automatically Startup After a Reboot .....	50
RSA Identity Governance and Lifecycle Stops Responding Because of Default Limits .....	50

## Preface

### Documentation Set

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

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

### Support and Service

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

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

## Revision History

---

Revision Number	Date	Revision
1	June 2018	<ul style="list-style-type: none"><li>• Updated procedures to upgrade the domain controller and slave nodes.</li><li>• Updated the cluster upgrade procedure.</li><li>• Moved installing a patch or upgrade content to a new chapter.</li></ul>
2	July 2018	<ul style="list-style-type: none"><li>• Removed procedures for clustering with a hardware appliance, which is not supported.</li></ul>

## 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.

### 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 7.1

### 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 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

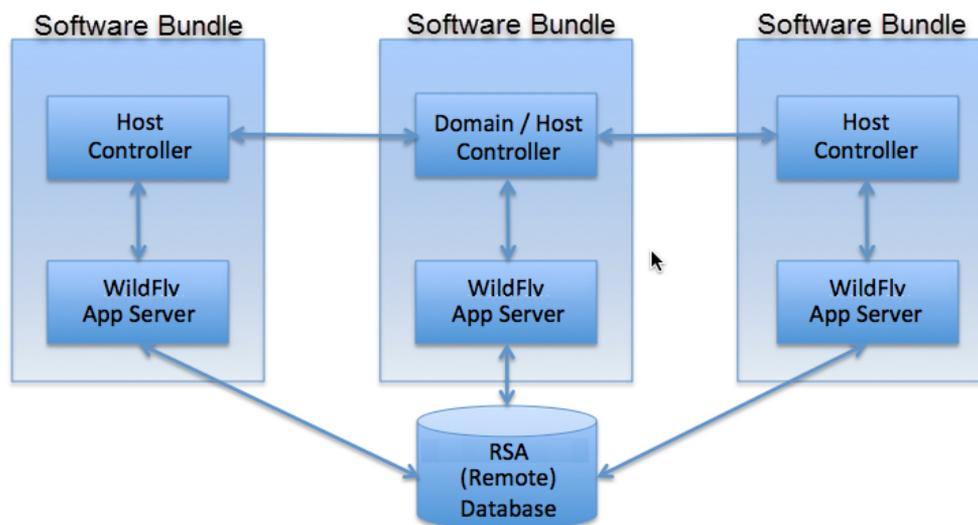
Throughout this document there are references that may include variable substitutions, such as: `$SAVEKSA_WILDFLY_HOME`. This variable represents the `/home/oracle` directory based on the initial installation and is not meant to imply that the home installation path can be changed during the setup/installation.

**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 Software Bundle

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 7.1 – 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 7.1 Software Bundle model.
- For the additional nodes that will participate in the cluster, install RSA Identity Governance and Lifecycle 7.1 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			

	Machine Name	IP Address	Systems Operations Node (Select One)
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

/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 file, verify that the EAR files do not exist by running the following commands:

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --command="deployment-
info --name=aveksa.ear"
WFLYCTL0216: Management resource '['deployment" =>
```

```
"aveksa.ear"]]' not found

/home/oracle/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
```

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 JMS Multicast Settings for the Cluster

### Before you begin

If the firewall is enabled, open the JMS port before performing the following procedure.

### Procedure

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

1. Determine if the default JMS multicast address can be used:

```
service aveksa_cluster testmulticast
```

OR

```
cd /etc/init.d
aveksa_cluster testmulticast
```

If the test returns PASS, then the default address can be used. If not, go to Step 2.

2. If the default IP address is already in use, test a new IP address:

```
service aveksa_cluster testmulticast <IP address>
```

OR

```
cd /etc/init.d
aveksa_cluster testmulticast <IP address>
```

Repeat as needed to find a usable JMS IP address.

3. As the **root** user, edit **/etc/init.d/aveksa\_cluster** and update **JMS\_MULTICAST\_IP** with the new address. For example:

```
JMS_MULTICAST_IP=231.0.0.4
```

## 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:

```
WILDFLY_HEAP_MEM: 9287
WILDFLY_PERM_META_MEM: 1639
```

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 `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
```

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

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

## Disable Appliance Mode in Software Bundle Installation

**CAUTION:** Perform this step only if you are using an 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 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

---

This section is for setting up and configuring the machine that has been identified as the ‘domain controller’ for the cluster.

**CAUTION:** Only a single machine can be identified to be 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 **/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>
Are you sure you want to use the password entered yes/no? yes
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" />
```

**CAUTION:** 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 **/home/oracle/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 `<server-groups>` setting and add a new `<server-group>` as shown below.

```

<server-groups>
  <server-group name="img-server-group" profile="full-ha">
<socket-binding-group ref="full-ha-sockets"/>
  </server-group>
</server-groups>

```

## 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=/home/oracle/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>
```

## 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 Aveska Security Realms

In a standalone mode (appliance mode) the installation automatically configures the security realms **AveskaAgentRealm** and **AveskaRealm**. The **AveskaAgentRealm** contains the key and trust store for the agent. The **AveskaRealm** 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 **AveskaAgentRealm** security realm under **security-realms** element, for example:

```
<security-realm name="AveskaAgentRealm">
  <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>
```

**AveskaRealm** will be customer specific setup. For our testing purpose we will be using the **aveska.keystore**.

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

```
<security-realm name="AveskaRealm">
  <server-identities>
    <ssl>
      <keystore path="/home/oracle/keystore/aveska.keystore" keystore-password="Av3k5a15num83r0n3" alias="server" key-
password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/keystore/aveska.keystore" keystore-password="Av3k5a15num83r0n3"/>
  </authentication>
</security-realm>
```

**CAUTION:** Verify that there is a space after each comma in the value for engine enabled-cipher-suites, to ensure successful server startup.

## Configure Host Network Interfaces

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

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

## 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.

## 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"
```

In SuSE 11, run these commands after saving the file to apply the changes:

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

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

In SuSE 12, run this command after saving the file to apply the changes:

```
systemctl restart SuSEfirewall2
```

## 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:

### SuSE 11 command

```
service aveksa_cluster start
```

### SuSE 12 command

```
systemctl start aveksa_cluster
```

**CAUTION:** If it does not exist, you need to create the log folder **\$SAVEKSA\_WILDFLY\_HOME/domain/log/** before running the command.

Look for the **started** message in **\$SAVEKSA\_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 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)
```

## 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 `/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
```

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:

```
/home/oracle/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
```

### Next steps

To check that the EAR files are deployed, monitor the **aveksaServer.log** file. See *Log File Location and Properties* for more information.

## Chapter 4: Setup and Configure the Host Controllers

---

**CAUTION:** 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.

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 Aveska Security Realms

The **AveskaAgentRealm** contains the key and trust store for the agent.

The **AveskaRealm** contains the key and trust store for the web access. In standalone mode (appliance mode), the security realms **AveskaAgentRealm** and **AveskaRealm** 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**.

**CAUTION:** 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 **AveskaAgentRealm** security realm under the **security-realms** element:

```
<security-realm name="AveskaAgentRealm">
  <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>
```

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

```
<security-realm name="AveskaRealm">
  <server-identities>
    <ssl>
      <keystore path="/home/oracle/keystore/aveska.keystore" keystore-password="Av3k5a15num83r0n3" alias="server" key-
password="Av3k5a15num83r0n3"/>
    </ssl>
  </server-identities>
  <authentication>
    <truststore path="/home/oracle/keystore/aveska.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 **AveskaClusterAdmin**.

```
<domain-controller>
  <remote host="{jboss.domain.master.address}" port="{jboss.domain.master.port:9999}" username="AveskaClusterAdmin"
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>
```

## Configure Host Network Interfaces

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

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

## 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
```

**CAUTION:** If it does not exist, you need to create the log folder `$AVEKSA_WILDFLY_HOME/domain/log/` before running the command.

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)
```

**CAUTION:** If you get a time out 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 `/home/oracle/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 `/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.

**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@8854779
02 [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>: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, **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:/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
```

## 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, 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 7.1. To apply a 7.1 patch to a version lower than 7.1, you must first follow the steps in this chapter, and then perform the procedures in [Installing a Patch or Upgrade for RSA Identity Governance and Lifecycle](#).

### Before You Begin

Gather the following details for the existing cluster. This information is used while performing the upgrade.

- Domain Controller, which should also be set up as the SON. To verify this, in RSA Identity Governance and Lifecycle, go to **Admin > System > Server Nodes**.
- Host Controllers. To determine this, in RSA Identity Governance and Lifecycle, go to **Admin > System > Server Nodes**.
- The database server IP address and port.
- The Cluster Management user name and password.
- The JMS Multicast IP and port (aveksa\_cluster script).

### Copy Installation Files

Download and extract the installation files from **Aveksa\_<VersionNumber>\_P<PatchNumber>.tar.bz2** to the default location **/tmp/aveksa/staging**, and copy the packages to **/tmp/aveksa/packages** on the domain controller and all host controllers.

### Backup WildFly and AFX

For a manual backup of WildFly, back up **wildfly-<installed\_version\_number>.Final** on each node and remove its symbolic link.

Back up the **/etc/init.d/aveksa\_cluster** script.

Back up AFX, which is typically installed on the domain controller at **/home/oracle/AFX**.

### 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 /home/oracle//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 /home/oracle/archive/aveksa_cluster.ear
```

3. Stage the EAR file:

- i. Run the command:

```
cd /home/oracle/deploy
```

- ii. At the prompt, enter:

```
./customizeACM.sh -c /home/oracle/archive/aveksa_
cluster.ear
```

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

4. Extract **aveksa.ear** to the **/tmp/customizeACM** directory:

```
cd /tmp/customizeACM
jar xvf /tmp/aveksa/staging/aveksa.ear
```

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

- i. `cd /tmp/customizeACM`
- ii. `jar cvf /home/oracle/archive/aveksa.ear`

6. Copy **aveksaWFArchitect.ear** from **/tmp/aveksa/staging** to the **/home/oracle/archive** directory.

## 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-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:

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --controller=<ip-
address> --command="deployment-info --server-groups=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:

```
/home/oracle/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
```

## 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**

## Install Support Files

Log in as root and run the following commands:

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

## 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 from 7.0.2 to 7.1.0, select NO.
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](#).
6. Perform all steps in [Set Up and Configure the Domain Controller](#) except "Deploy EAR Files".

## Deploy EAR files

This procedure deploys the EAR files created in [Repackage EAR with Customizations](#).

### Procedure

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

1. Deploy the `aveksa.ear` file:

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

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

2. Deploy the `aveksaWFArchitect.ear` file:

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

Where `<ipaddress>` 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.

## Procedure

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 Host Controllers

Perform the following steps to upgrade all host controllers.

### Procedure

1. Go to `/tmp/aveksa/staging/deploy`.
2. Run **install.sh** using the same database server.
3. When prompted for database migration from 7.0.2 to 7.1.0, 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](#).
6. Perform all steps in [Setup and Configure the Host Controllers](#).

## Upgrade AFX

The following steps are for upgrading AFX.

### 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**

This runs the AFX auto upgrader.

## 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 `/home/oracle`, 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 /home/oracle//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 /home/oracle/archive/aveksa_cluster.ear
```

3. Extract customized content:

- i. Run the command:

```
cd /home/oracle/deploy
```

- ii. At the prompt, enter:

```
./customiseACM.sh -c /home/oracle/archive/aveksa_
cluster.ear
```

- iii. Follow the prompts to unpack the EAR to **/tmp/customiseACM**.

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 /home/oracle/archive/aveksa.ear

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

- i. cd /home/oracle/Aveksa\_<VersionNumber>\_P<PatchNumber>

- ii. cp aveksaWFArchitect.ear/aveksaWFArchitect.ear /home/oracle/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:

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

3. Undeploy the existing EAR files:

- i. Issue the following commands:

```
cd /home/oracle/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-  
groups=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:

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --controller=<ip-  
address> --command="deploy /home/oracle/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:

```
/home/oracle/wildfly/bin/jboss-cli.sh -c --controller=<ip-  
address> --command="deploy  
/home/oracle/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:

```
/home/oracle/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_7.1.0_P01_<DateTime>.ear
cp aveksaWFArchitect.ear aveksaWFArchitect_7.1.0_P01_
<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 **/home/oracle/Aveksa\_<VersionNumber>\_P<PatchNumber>/deploy** and copy **lockdown\_privileges.sh** to **/home/oracle/deploy** in the cluster node:

```
cd /home/oracle/Aveksa_<VersionNumber>_P<PatchNumber>/deploy
cp -rvp lockdown_privileges.sh /home/oracle/deploy
```

2. Go to **/home/oracle/Aveksa\_<VersionNumber>\_P<PatchNumber>/etc** in the patch directory and copy **sudoers.snippet** to **/home/oracle/deploy/etc** in the cluster node:

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

3. Login as **root** and go to **/home/oracle/deploy** to execute the **lockdown\_privileges.sh** script:

```
/home/oracle/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
```

## Appendix A: 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 /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
```

## 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 Software Bundle installation, 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

## Configuring WildFly Clustering

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.