



HP ArcSight SmartConnector Load Balancer

Software Version: 1.0

Configuration Guide

August 28, 2015



Configuration Guide

SmartConnector Load Balancer

August 28, 2015

Copyright ©2015 Hewlett-Packard Development Company, L.P. Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Follow this link to see a complete statement of ArcSight's copyrights, trademarks and acknowledgements:
<http://www.hpenterprisesecurity.com/copyright>.

The network information used in the examples in this document (including IP addresses and hostnames) is for illustration purposes only.

This document is confidential.

Revision History

Date	Description
08/28/2015	General availability of the HP ArcSight SmartConnector Load Balancer.

Contents

Chapter 1 — SmartConnector Load Balancer	5
Overview.....	5
Why Use Load Balancer?.....	5
Load Balancer Features	6
How Load Balancer Works	6
Syslog-based Load Balancing	6
Support for Single-line Events	6
For TCP	6
For UDP.....	7
File-based Load Balancing	7
Routing Policies.....	7
Load Balancer Modes.....	7
Chapter 2 — Installation and Configuration	9
System Requirements	9
General Setup	9
Hardware Requirements.....	9
Software or Platform Requirements.....	9
SmartConnector Requirements	9
Downloading Load Balancer	10
Verifying Your Files	10
Preparing for Deployment.....	10
Configuring the Ethernet Connection.....	10
Installing the Load Balancer	11
Installing the Load Balancer in Console Mode.....	12
Installing the Load Balancer in GUI Mode	14
Uninstalling Load Balancer.....	17
Configuring Load Balancer	17
Configuring Load Balancer in Standalone Mode.....	18
Configuring Load Balancer in HA Mode.....	19
Configuration Parameters	20
Configuration Examples	29
Configuring MemberHosts in Standalone Mode	29
Configuring MemberHosts as Peer.....	29
Configuring MemberHosts as Primary-Secondary	31
Syslog Load Balancing Routing Rule Example.....	32
File Load Balancing Routing Rule Example.....	34
Sample Configuration File	38
Starting Load Balancer	41
Installing Load Balancer as a Service.....	41
Starting or Stopping the Load Balancer Service	43

Load Balancer Service Commands	43
Load Balancer Service-related Logs	44
Interpreting Logs	44
Chapter 3 — Load Balancer Troubleshooting.....	45
Troubleshooting.....	45
Load Calculators Not Initialized/Destination Monitoring Not Working.....	45
Destination Configured with SCP Protocol but File Delivery Fails.....	45
Sources Relocated Away from [x] of [y] Destinations in Routing Rule	45
Appendix A –Configuration File Templates with Callout Information	47
Standalone Mode Configuration Template File	47
HA Mode Configuration Template File	52

Chapter 1 — SmartConnector Load Balancer

This guide provides information about downloading, installing and configuring the HP ArcSight SmartConnector Load Balancer application for use with event collection for ArcSight products.

Overview

HP ArcSight SmartConnector Load Balancer provides a “connector-smart” load balancing mechanism by monitoring the status and load of SmartConnectors. Currently it supports two types of event sources and SmartConnectors. One distributes the syslog input stream to syslog connectors using TCP or UDP protocol and the other downloads files from a remote server and distributes them to the file-based connectors.

 [File-based load balancing is a beta feature for this release.](#)

Load Balancer ensures efficiency by distributing the load to a pool of SmartConnectors. Load Balancer supports high availability configuration with active and standby nodes. It distributes the events received to one or more SmartConnectors predefined in the SmartConnector pool.

Load Balancer is aware of the following information for SmartConnectors defined as the SmartConnector pool:

- **Availability (up or down)** – Load Balancer monitors SmartConnectors for availability. Events are not forwarded to a SmartConnector if it is not running (down). Instead, events are forwarded to the next available SmartConnector in the pool per the defined load-balancing algorithm rules.
- **SmartConnector Load** - CPU usage, memory usage, and queue drop rate for events

Why Use Load Balancer?

Often a varying volume of events from the event source makes it difficult to configure the connector and there could be an outage in continuous event collection if any connectors go down. Load Balancer addresses these problems by distributing the events across SmartConnectors and by redistributing the events to available connectors if any connectors are down.

Load Balancer provides support to devices generating varying volumes of events, where:

- Overloaded connectors result in event loss and delayed collection
- Under-utilized connectors result in wasted resources
- Manual and tedious sizing and maintenance is necessary
- One connector becomes a single point of failure

Load Balancer is an advantageous solution for:

- Connector-smart load balancing
- Load balancing for TCP protocol without keeping the sessions sticky

-
- Load balancing for files
 - An aggregation-preferred routing policy, which sends events from a single device to the same connector up to a certain threshold

Load Balancer together with the SmartConnector pool provides availability, reliability, and scalability.

- Load Balancer supports High Availability (HA). If the active Load Balancer node is down, a passive Load Balancer node becomes an active node and continuously collects the events.
- If a SmartConnector is down, Load Balancer forwards the events to the next available connector in the SmartConnector pool per the load balancing rules.

Load Balancer Features

Load Balancer supports the following:

- High availability (HA) mode, which can be configured with two hosts
- Syslog type of input stream or batch files (beta) on FTP server
- Syslog-based and file-based (beta) SmartConnectors as destinations
- TCP and UDP protocol for syslog-type input or connectors
- Three routing policies — round robin, weighted round robin, and aggregation preferred
- Event batching (TCP only) for better aggregation at the destination connector and better network throughput
- Email notification for up/down status on member hosts and destination connectors
- Load and health monitoring of connector destinations
- Load Balancer running as a service

How Load Balancer Works

Load Balancer supports syslog-based load balancing, file-based load balancing, and several types of routing rules.

Syslog-based Load Balancing

Support for Single-line Events

The load balancer parses the input stream into a line but not to an event. It supports single-line event stream but not multi-lined events.

For TCP

When the source is a syslog-based network process and configured to use TCP protocol, the input stream is parsed into event lines and bundled into a batch. Then the batch is distributed to one of the destinations in the destination pool.

For UDP

If a routing rule is configured to use UDP protocol, event batching does not happen. Instead, each incoming event is distributed into one of the destinations configured in the routing policy.

File-based Load Balancing

Load Balancer downloads files from an FTP server and distributes them to one of the locations associated with the file connectors. It supports batch files. File-based connectors that read and process files are good candidates for this feature.

Routing Policies

Routing policies are a set of rules that define the data distribution from a source to a set of destinations. Eligible sources are syslog servers or FTP servers. A destination pool consists of one or more syslog or file connector destinations, all of the same type. Connector types cannot be mixed within a single destination pool. Routing policies define event or file distribution rules from a source to destination pool.

The routing policies supported in Load Balancer are:

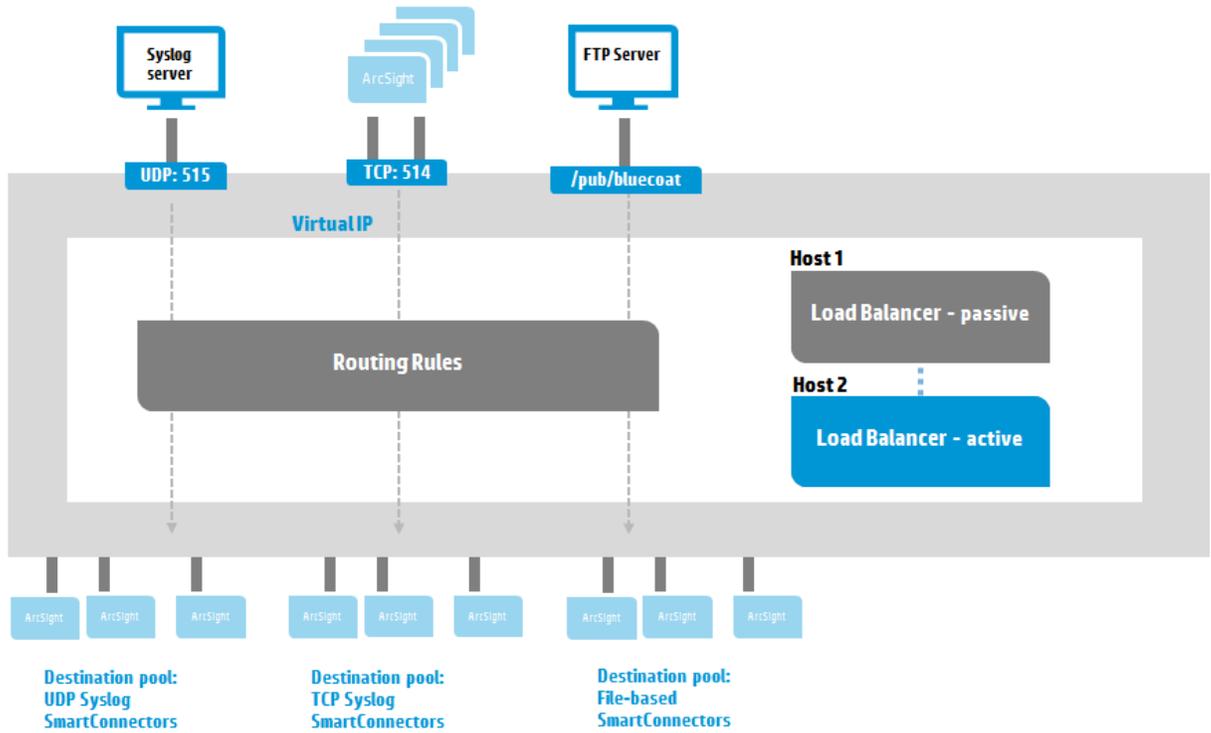
- **Round Robin:** Distributes events, batches, or files to each available destination in the destination pool in round robin fashion, beginning again at the start in a circular manner. File-based load balancing supports only the Round Robin policy.
- **Weighted Round Robin:** Distributes events in a round-robin fashion, but sends more events or batches to lightly loaded destinations.
- **Aggregation Preferred:** Events from the same source are sent to the same destination until a threshold is reached. Then, it will switch the routing to another destination. This routing policy is better suited if aggregation is enabled on connector destinations where events are sent to the same destination until certain load thresholds are met.

Load Balancer Modes

Load Balancer can be configured to run in three modes. To use the high availability feature, Load Balancer should be installed on two separate hosts sharing a virtual IP address. See the details at [Configuring Load Balancer in HA Mode](#).

- **Standalone mode:** Load Balancer runs as a single host without supporting the high availability feature. One host with a single static IP address is required to run Load Balancer in this mode.
 - **HA mode as peer:** Load Balancer runs with two hosts. The host that starts first becomes active and another host runs as passive until the first host goes down. The second host becomes active and stays active, even if the first host comes back up again.
 - **HA mode as primary-secondary:** Load Balancer runs with two hosts. One host can be designated as the preferred active host. In this mode, the host marked as primary runs as active node whenever it becomes available.
-  [The High Availability feature, which is available using primary-secondary or peer mode, currently works only within the same subnet.](#)

Load Balancer can be deployed between any syslog source, including SmartConnectors configured with CEF syslog or raw syslog destinations, or file source and SmartConnectors. The following diagram shows a Load Balancer deployment example running in HA mode. Both hosts share the common virtual IP address to handle the connection failover when an active Load Balancer host goes down. As shown in the diagram, Load Balancer can be used for three different types of input sources and destination pool types.



When configuring the routing rule, source and destination types must match. If the source is TCP syslog, the connectors in the destination pool must be TCP syslog connectors. Likewise, if the source is a file type, the connectors on the destination must be file-based connectors that expect to handle files.

When the routing rule is configured with TCP protocol, events received from the same source IP and port number are bundled into event batches. Event batching happens when any of the following conditions are met: buffer size, number of events, and batching interval. The bundled event batch is persisted on the hard drive before it is sent to the destination connector in the `/${ARCSIGHT_HOME}/user/loadbalancer/lbdata/persistence/{source}` directory of the currently active node. Note that persisted event batches are not shared across the member hosts and any unprocessed event batches awaiting bundling during the shutdown are sent when Load Balancer starts up again.

Chapter 2 — Installation and Configuration

This section describes system requirements and getting started with Load Balancer, including pre-deployment requirements, Ethernet configuration, Load Balancer installation, and Load Balancer configuration.

 Load Balancer is an independent component, not packaged with SmartConnectors.

System Requirements

The following section outlines the minimum system requirements for ArcSight SmartConnector Load Balancer.

General Setup

Servers should be dedicated to load balancing (not running other applications.)

For high availability (HA), there should be two separate servers, one for the active or primary Load Balancer and another for standby or secondary Load Balancer. They will share a Virtual IP address, so they should be in the same network location.

In addition, use the standard hardware required to deploy more than one SmartConnector to create the pool of SmartConnectors. See the SmartConnector documentation for details.

Hardware Requirements

- CPU: 2 CPU X 4 Cores each (2 x Intel E5620, quad core, 2.4 Ghz or better)
- RAM: 16 GB
- Disk: 60 GB
- Number of network interfaces—1 Dedicated Gig Ethernet interface

 To achieve better performance, use a server with higher system specifications.

Software or Platform Requirements

- RHEL 6.6 and 7.0 (64-bit only)
- CentOS 6.6 and 7.0 (64-bit only)

SmartConnector Requirements

- HP ArcSight SmartConnector release 7.1.3.7445 or later
- Syslog daemon and/or file-based SmartConnectors

Downloading Load Balancer

Download the 64-bit executable and the *HP ArcSight SmartConnector Load Balancer Configuration Guide* from the Support Web site (<https://softwaresupport.hp.com/>).

For a successful Load Balancer installation, follow the installation procedures documented in the *Installing the Load Balancer* section of this chapter.

Verifying Your Files

HP provides a digital public key for you to verify that the signed software you received is indeed from HP and has not been manipulated in any way by a third party.

Visit the following site for information and instructions:

<https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning>

Preparing for Deployment

To run in HA mode, the following pre-requisites must be met:

- Have two hosts with static IP addresses to install Load Balancer.
- Have a single, unused address for the VIP to run Load Balancer in HA mode.
- Create an Ethernet configuration file to support failover migration.

Configuring the Ethernet Connection

Begin with creating an Ethernet configuration file to support failover migration.

To configure the Ethernet connection:

1. Before installation, identify the machine or machines where Load Balancer will be installed. To enable HA, two machines and one virtual IP address are needed (all in the same subnet).
2. If Load Balancer is run by a non-root user, be sure to give sudo capability to the user. For example, if `arcsight` is the user that installs and runs Load Balancer, add the `arcsight` user and add sudoer capability with `NOPASSWD`. See the following example:

 Ignore this step if Load Balancer is installed as root.

```
# adduser arcsight // Creates arcsight group and adds the user to the group.  
# sudo visudo
```

```
// Add the following line, and exit.  
arcsight ALL=(ALL) NOPASSWD:ALL
```

 Step 3 and 4 can be skipped if Load Balancer is deployed in standalone mode. Note that step 3 and 4 can vary depending on the OS version and flavor. Use the instructions as a reference. HP advises getting help from the network administrator to execute the steps below.

3. When using two machines for HA, create a network profile or Ethernet configuration file on each machine. In the supported distributions of Linux, this file is usually located in the `/etc/sysconfig/network-scripts` directory.

- a. Go to the directory and verify that the file has the primary network interface (usually 'eth0') configuration. The `IPADDR` value of this file should show the IP address assigned to this machine. A similar configuration file needs to be created for the virtual IP address.

- b. Log on as a privileged administrator and go to the directory where the Ethernet profiles are located.

```
# cd /etc/sysconfig/network-scripts
```

- c. Copy the default `eth0` configuration to `eth0:1`.

```
# cp ifcfg-eth0 ifcfg-eth0:1
```

- d. Edit `ifcfg-eth0:1` to modify `DEVICE` to `eth0:1` and `IPADDR` to a virtual IP address and save the file.

 `ONBOOT` must be set to `no` in order to prevent the VIP address from being bound to the host automatically upon system reboot. Otherwise, the virtual IP address needs to be released manually when another host is running as the active node or it will lose the connection from the source devices.

```
DEVICE=eth0:1
```

```
IPADDR=<virtual-ip-address> # for example, 192.168.1.255
```

```
ONBOOT=no
```

4. Verify the full path of the `ifup` command, usually `/sbin/ifup`. Make note of the full path of the `ifup` command and Ethernet profile.

Installing the Load Balancer

Before beginning your installation, obtain the Load Balancer binary (see [Downloading Load Balancer](#)) and configure the Ethernet connection (see [Configuring the Ethernet Connection](#)), if needed. If Load Balancer will be running in HA mode, install Load Balancer on each host.

The installer runs both in console mode and GUI mode. Follow the instructions in one of the following sections for the appropriate mode:

- ◆ [Installing the Load Balancer in Console Mode](#)
- ◆ [Installing the Load Balancer in GUI Mode](#)

Installing the Load Balancer in Console Mode

To install the Load Balancer files in console mode:

1. Obtain the Load Balancer binary file and copy it to the desired location.
2. Run the installer.

 The '-i console' mode is automatically selected by default if you do not use graphical display or if the DISPLAY variable is not set. It can also be specifically invoked using the -i console switch as shown here.

```
# sh ArcSightSCLoadBalancer-<build-number>.bin -i console
```

```
Preparing to install...
```

```
Extracting the JRE from the installer archive...
```

```
Unpacking the JRE...
```

```
Extracting the installation resources from the installer archive...
```

```
Configuring the installer for this system's environment...
```

```
Launching installer...
```

```
Graphical installers are not supported by the VM. The console mode will be used instead...
```

```
=====  
ArcSight SmartConnector Load Balancer          (created with InstallAnywhere)  
-----
```

```
Preparing CONSOLE Mode Installation...
```

```
=====  
Introduction  
-----
```

```
The ArcSight installer guides you through the installation of the ArcSight SmartConnector Load Balancer.
```

```
ArcSight recommends that you quit all other programs before continuing with this installation.
```

```
Click the 'Next' button to proceed to the next window. If you want to change something on a previous window, click the 'Previous' button. To cancel this installation at any time, click the 'Cancel' button.
```

```
PRESS <ENTER> TO CONTINUE:
```

3. Type an absolute path or just press **Enter** to accept the default location.

Choose Install Folder

Select an installation folder. When upgrading from a previous version, select the folder that contains the currently installed ArcSight SmartConnector Load Balancer

Where to install:

Default Install Folder: /home/arcsight/ArcSightSCLoadBalancer

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT

:

4. Type the option number that corresponds with the shortcut or link to be created for Load Balancer, if any. Press **Enter**.

Choose Link Location

Where would you like to create links?

->1- Default: /home/arcsight

2- In your home folder

3- Choose another location...

4- Don't create links

ENTER THE NUMBER OF AN OPTION ABOVE, OR PRESS <ENTER> TO ACCEPT THE DEFAULT

:

5. Check the pre-installation summary before proceeding to the installation, then press **Enter** to start the installation.

Pre-Installation Summary

Review the following information before continuing:

Product Name:

ArcSight SmartConnector Load Balancer

Install Folder:

/home/arcsight/ArcSightSCLoadBalancer

Link Folder:

/home/arcsight

Install Set:

Typical

PRESS <ENTER> TO CONTINUE:

6. Upon completion, the screen displays the location where Load Balancer is installed.

Installing...

```
-----  
[===== | ===== | ===== | ===== ]  
[----- | ----- | ----- | ----- ]  
-----
```

Installation Complete

```
-----
```

The core components of the ArcSight SmartConnector Load Balancer have been successfully installed to:

```
/home/arcsight/ArcSightSCLoadBalancer
```

PRESS <ENTER> TO EXIT THE INSTALLER:

7. Press **Enter** to exit.

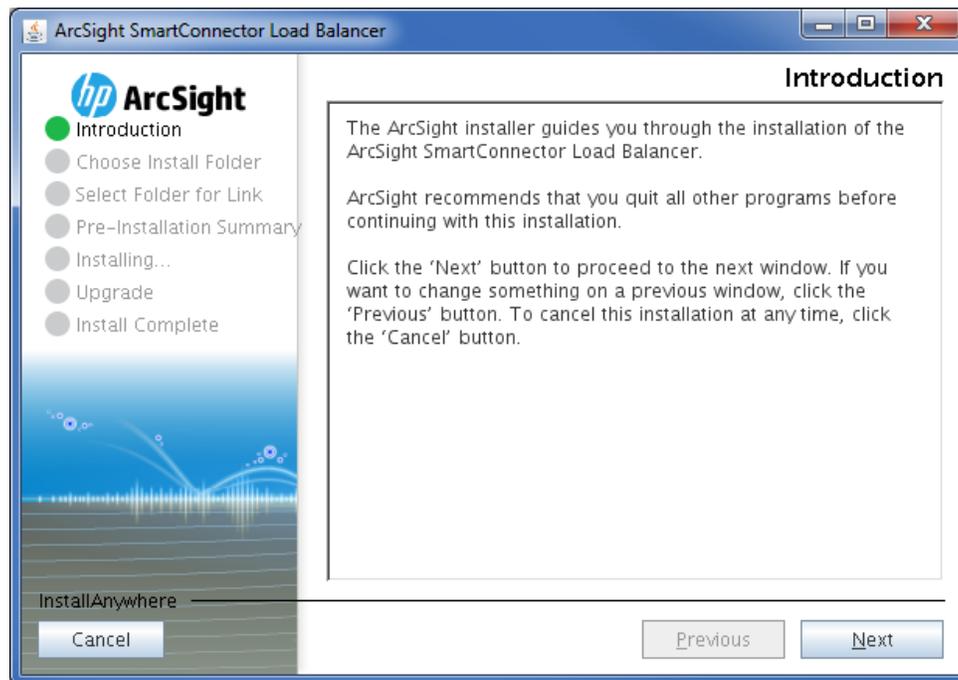
Installing the Load Balancer in GUI Mode

To install the Load Balancer files in GUI mode:

1. Copy the Load Balancer binary file to the desired location.
2. Run the installer.

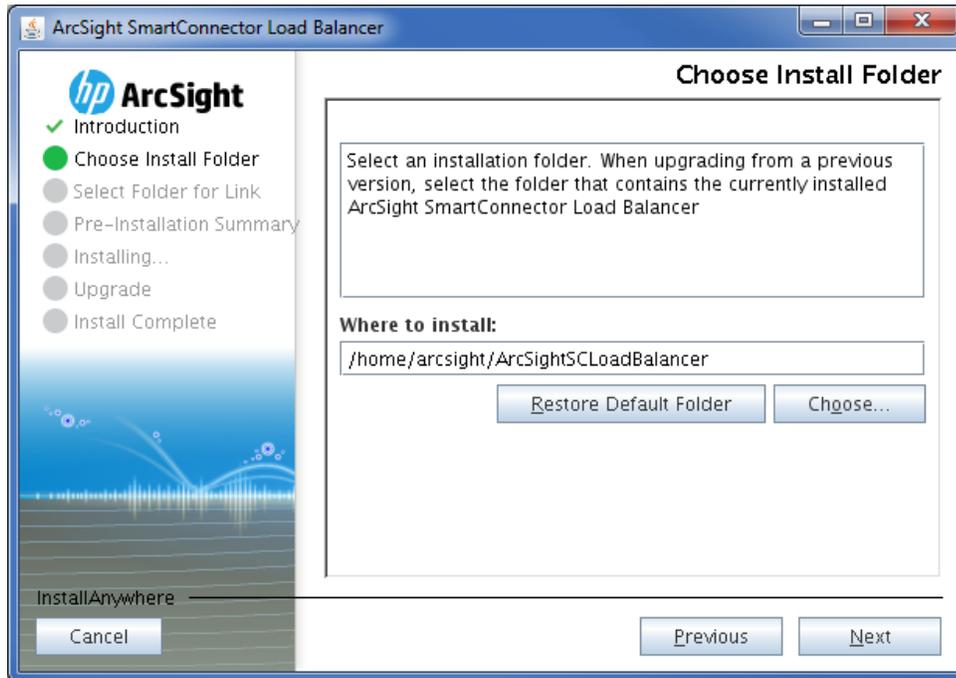
```
# sh ArcSightSCLoadBalancer-<build-number>.bin
```

The installer loads and the Introduction screen is displayed. Click **Next**.

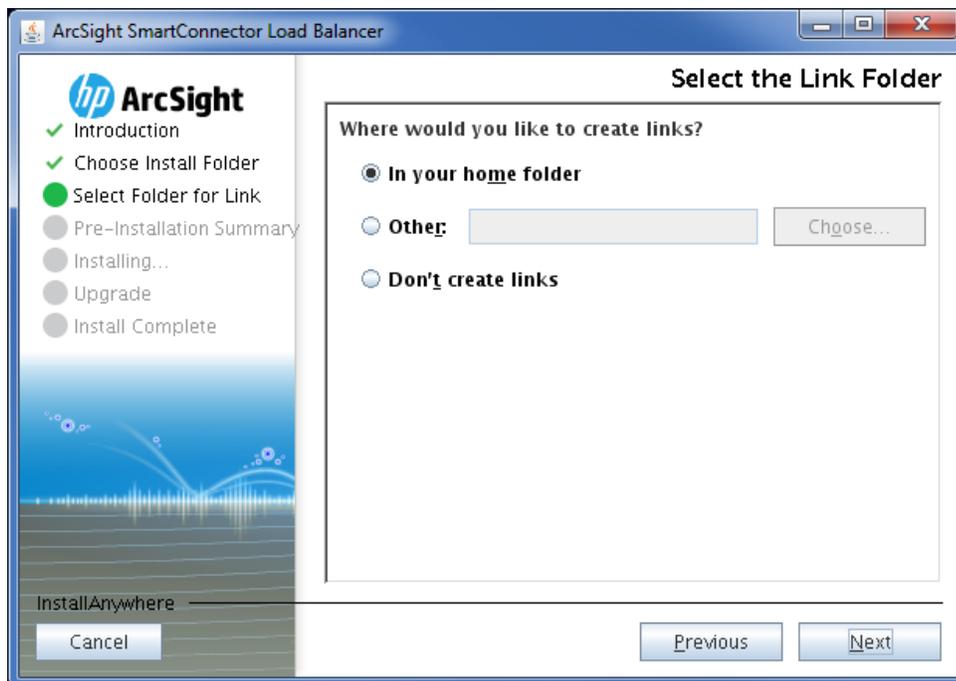


3. Accept the default path or enter a new path in the "Where to install" field. Click **Next**.

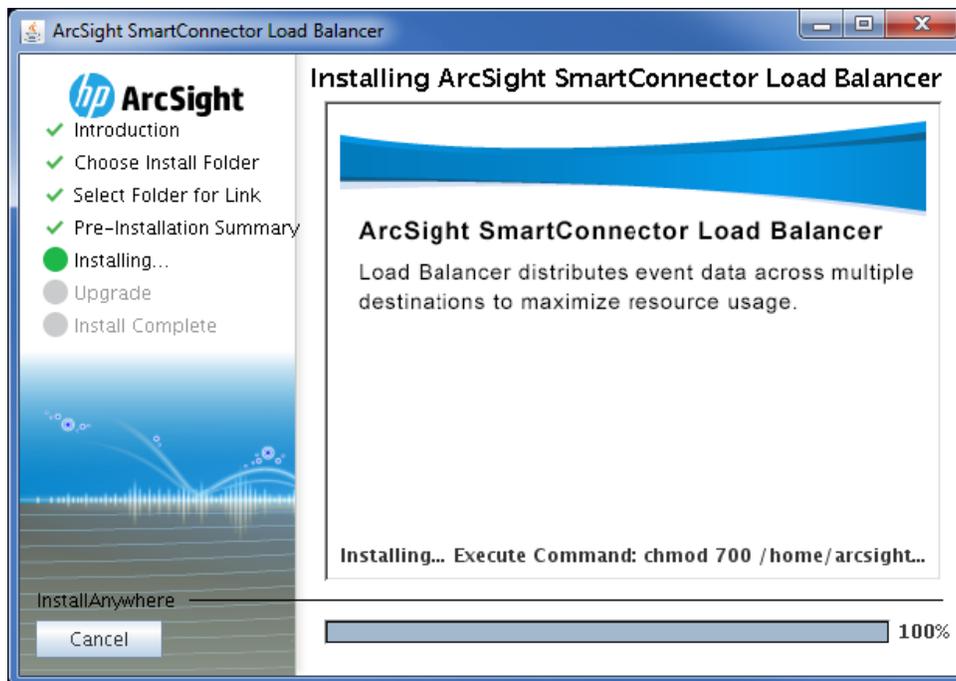
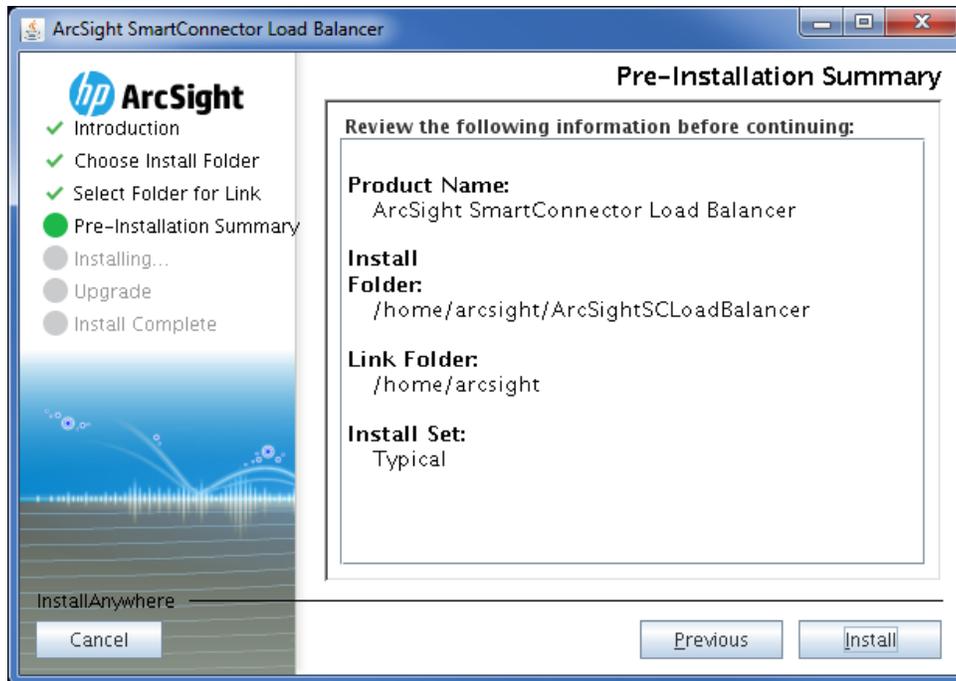
If you are logged in as `root`, the install path is `root /ArcSightSCLoadBalancer`.



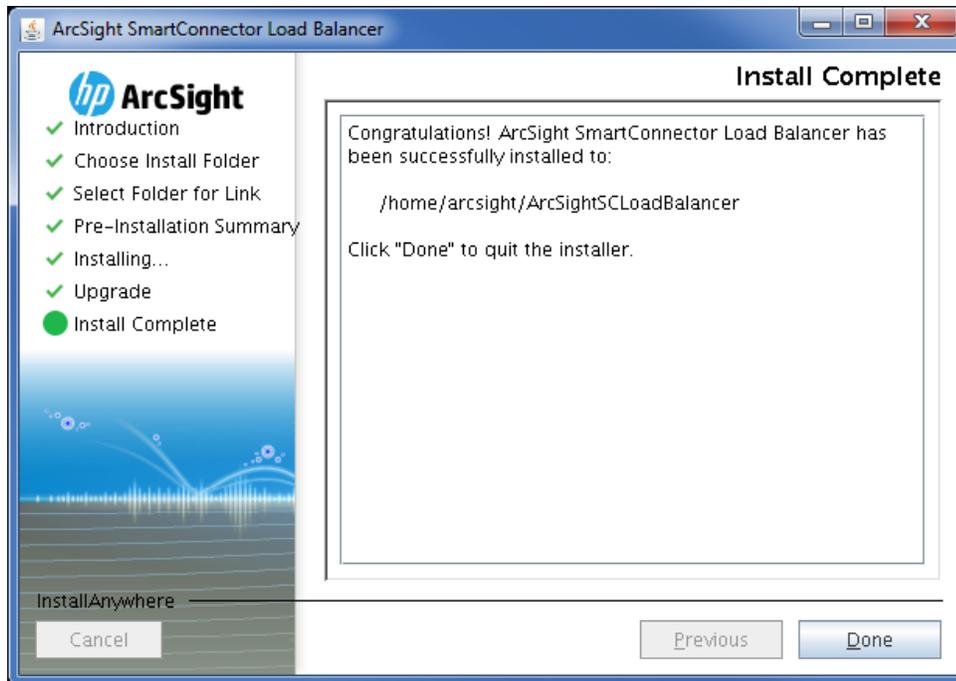
4. Select your preferred option for creating a link to Load Balancer. Click **Next**.



5. Review installation information. Click **Previous** to make changes or click **Install** to install Load Balancer.



6. Review the installation location and click **Done** to quit the installer.



Uninstalling Load Balancer

Load Balancer can be uninstalled using the GUI or console mode.

To uninstall Load Balancer in GUI mode:

1. Run the `./Uninstall_ArcSightSmartConnectorLB` script in the `$ARCSIGHT_HOME/current/UninstallerData` folder to launch the GUI uninstaller. For example:

```
./Uninstall_ArcSightSmartConnectorLB -i swing
```

2. Follow the screen prompts.

To uninstall Load Balancer in console mode:

1. Run the `./Uninstall_ArcSightSmartConnectorLB -i console` command to uninstall Load Balancer in console mode. For example:

```
./Uninstall_ArcSightSmartConnectorLB -i console
```

2. Add `'-i silent'` to launch the silent mode installation.

Configuring Load Balancer

Load Balancer can be configured for standalone mode or high availability. See the appropriate section:

- [Configuring Load Balancer in Standalone Mode](#)
- [Configuring Load Balancer in HA Mode](#)

Configuring Load Balancer in Standalone Mode

Only one host is needed when Load Balancer is run in standalone mode.

To configure Load Balancer in standalone mode:

1. Log on to the host.
2. Go to `$ARCSIGHT_HOME/config/loadbalancer`. Copy the `lbConfig.xml.template.standalone` file to the `$ARCSIGHT_HOME/user/loadbalancer/` directory to configure Load Balancer to run in standalone mode.
3. Go to `$ARCSIGHT_HOME/user/loadbalancer` directory and rename the file `lbConfig.xml`.
4. In the `lbConfig.xml` file, configure the host under the `memberHost` parameter.
 - a. List the host for Load Balancer.
 - b. Match `memberIdentity` and the `memberHosts/memberHost` name so that Load Balancer can identify itself. See [Configuring MemberHosts in Standalone Mode](#) for more information.
 - c. Set `isPrimary=true`.
5. Configure the routing and other parameters. Routing rules must be defined to have the events distributed from a source to a set of destinations (SmartConnectors). See [Syslog Load Balancing Routing Rule Example](#) or [File Load Balancing Routing Rule Example](#) for more information.
 - a. Configure destinations and destination pools.
 - b. Configure sources.
 - c. Configure routing rules.
6. Configure the web server.
7. Finish configuration. Refer to the [Configuration Parameters](#) and the [Sample Configuration File](#) for more information. Other optional configuration settings include:
 - ◆ Notification
 - ◆ Statistics Logging
8. Start the destination connectors before you start Load Balancer to ensure that Load Balancer can query the destinations for connector health and load.
9. Go to the `$ARCSIGHT_HOME` directory on the host and start Load Balancer.

```
# bin/arcsight loadbalancer
```

- 🔗 If there are any configuration errors, Load Balancer will not start. Instead, it logs the configuration error messages at `logs/loadbalancer.log`. If this happens, fix the issue associated with the error message and start Load Balancer again.

Configuring Load Balancer in HA Mode

If Load Balancer will be run in High Availability (HA) mode, first decide the type (peer or primary-secondary HA) and configure Ethernet file accordingly. (See [Preparing for Deployment](#).) Configure either the HA primary member host or the one that will be started first in peer mode with the full configuration in the XML configuration file. The secondary host (for primary-secondary configuration) or the passive host (for peer configuration) that starts second must be configured with the settings for member hosts and web service only. Other configuration settings will be synchronized when the second member host is started after the first member host.

 The host that starts first will overwrite the configuration file of the host that is started second. If the Load Balancer host with an incomplete configuration file is started first, the configuration can be lost. It is a good idea to make a backup of the completed configuration file before starting Load Balancer.

To configure Load Balancer in HA mode:

1. Log on to the primary member host or the one where the Load Balancer will start first (for peer configuration).
2. Go to `$ARCSIGHT_HOME/config/loadbalancer`. Copy the `lbConfig.xml.template` file to the `$ARCSIGHT_HOME/user/loadbalancer/` directory if Load Balancer will be running in HA mode.
3. Go to `$ARCSIGHT_HOME/user/loadbalancer` directory and rename the file `lbConfig.xml`.
4. In the `lbConfig.xml` file, configure both participating member hosts and the routing rules.
 - a. List both participating member hosts for Load Balancer. (For security reasons, Load Balancer only communicates with known hosts using configured ports.)
 - b. Match up `memberIdentity` with one of `memberHosts/memberHost/name` so that Load Balancer can identify itself. See [Configuring MemberHosts as Peer](#) and [Configuring MemberHosts as Primary-Secondary](#) for more information.
 - c. For primary-secondary configuration, set `isPrimary=true` on the primary host. On the secondary host, set `isPrimary=false`.
 - d. For peer mode, set both hosts to `isPrimary=false`.
 - e. Refer to [Configuration Parameters](#) to configure a secondary host and finish the host configuration.
5. Configure the routing and other parameters. Routing rules must be defined to have the events from a source be distributed to a set of destinations (SmartConnectors). See [Syslog Load Balancing Routing Rule Example](#) or [File Load Balancing Routing Rule Example](#) for more information.
 - a. Configure destinations and destination pools.
 - b. Configure sources.
 - c. Configure routing rules.

-
6. Configure the web server.
 7. Finish optional configuration. Refer to the [Configuration Parameters](#) and the [Sample Configuration File](#) for more information. Other optional configuration settings include:
 - ◆ Notification
 - ◆ Statistics Logging
 8. Log on to the second host and do the following:
 - a. Go to `$ARCSIGHT_HOME/config/loadbalancer` and copy the `lbConfig.xml.template` file to the `$ARCSIGHT_HOME/user/loadbalancer/` directory. Go to `$ARCSIGHT_HOME/user/loadbalancer` directory and rename the file `lbConfig.xml`.
 - b. Edit member hosts in the `lbConfig.xml` file. Make sure that `memberIdentity` is different from the one configured following step 4b.
 - c. The routing rule can be copied from the configuration file created in step 5 if preferred, but it is not required because the configuration values will be synchronized and persist after startup.
 9. Configure the firewall to open the ports on both hosts as needed to allow the two participating hosts to detect each other. Configure the firewall rule as needed. The configured ports include: `memberHosts/vipPingPort`, `memberHost/port` and all listening ports configured in `source` and outbound ports configured for `destinations`.
 10. Start the destination connectors before you start Load Balancer to ensure that Load Balancer can query the destination for connector health and load.
 11. Go to the `$ARCSIGHT_HOME` directory on the primary or active host and start Load Balancer.

```
# bin/arcsight loadbalancer
```

- 🔗 The virtual IP address is obtained by the host where Load Balancer is first started when Load Balancer is configured as peer. If Load Balancer is configured as primary-secondary, the virtual IP address will be used by the primary host.

If there are any configuration errors, Load Balancer will not start. Instead, it logs the configuration error messages at `logs/loadbalancer.log`. If this happens, fix the issue associated with the error message and start Load Balancer again.

12. Log on to the secondary or passive host and start Load Balancer.

Configuration Parameters

Configure Load Balancer with the following parameters. They fall into several basic categories:

- `memberIdentity`

-
- `memberHosts`
 - `notification`
 - `routing`
 - `statisticsLogging`
 - `webServer`
 - `globalParameters`

`memberIdentity`

`memberHost` in the `memberHosts` section defines the list of hosts that run Load Balancer, where each member host must have a unique name. The value of `memberIdentity` should be configured with the member host name that identifies the current host.

Verify that:

- The valid name consists of alphanumeric characters without spaces.
- The matching names are found in `memberHosts/memberHost/name`.
- The host configuration is the configuration for the current node.

`memberHosts`

Configure the list of member hosts that participate in load balancing in this section. The load balancer mode is determined by this configuration. Up to two member hosts are allowed.

- `vipAddress`: Specifies the virtual IP address when running Load Balancer with two hosts to enable HA mode.
- `vipPingPort`: Specifies the port used internally to detect the virtual IP binding status. Change the value if this port is being used by another application. (Default port is 9090.)
- `memberHost`: Configures the participating host where Load Balancer will be installed and running.
 - ◆ `name`: Specifies a unique name that identifies the host.
 - ◆ `address`: Specifies the IP address of the participating host. Load Balancer must be installed on this host.
 - ◆ `port`: Specifies the port number used by the underlying library for HA support.
 - ◆ `isPrimary`: Specifies the running mode for Load Balancer.
 - Set this value to `true` to designate a primary host when Load Balancer is running in primary-secondary mode.
 - Only one host can be configured as the designated primary host.
 - To run Load Balancers in peer mode, set this value to `false` for both member hosts.
 - ◆ `vipBindCommand`: Specifies the full command used to bind the virtual IP address to this host. Prior to configuring this, the Ethernet connection virtual IP address should have been configured. See [Configuring the Ethernet Connection](#) for details.

-
- In Linux, `/sbin/ifup` shows the Ethernet configuration.
 - Be sure to use the absolute path when specifying the command. For example, if the virtual IP address profile is located in:
`/etc/sysconfig/network-scripts/ifcfg-eth0:1`

specify:

```
sudo /sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1.
```

 If Load Balancer is running as the root user, remove 'sudo' .

- ◆ `vipUnbindCommand`: Specifies the full command used to unbind the virtual IP address from this host. It defines the counter command for binding. Refer to the details in the previous `vipBindCommand`.

notification

The configuration information provided in this section sets up notifications when certain events occur, such as when a member host goes up or down, or when a destination host goes down or up.

- `enable`: Specifies whether the sending of notification is enabled or disabled. Set this value to `true` to enable notifications.
- `enabledNotification`: Specifies the events for which notifications are sent. Notifications are supported for the following types of events:
 - ◆ `MemberHostUp`
 - ◆ `MemberHostDown`
 - ◆ `DestinationUp`
 - ◆ `DestinationDown`

Notifications are sent only for specified supported events. For example, if only the `MemberHostDown` event is listed, the notification will be sent only when one of the configured hosts is down.

- ◆ `event`: Specifies the events for which a notification will be sent.
 - `name`: Specifies the event name.
 - `message`: Specifies the custom message. If undefined, a default message is used.
- `email`: Configures the email sender, receiver, prefix, and SMTP server.
 - ◆ `prefix`: Configures the value used to tag the notification message in the subject line. When not configured, the subject will not have a prefix tag.
 - ◆ `recipients`: Specifies a list of one or more valid email addresses of the recipients. Separate each email address by a space.
 - ◆ `sender`: Specifies the sender's email address.
 - ◆ `smtpServer`: Specifies the SMTP server configuration. If this value is not configured, the email will not be sent.

routing

Use this section to define the routing rules. Data will be received from the source machine and distributed to the destinations in the destination pool. In routing configuration, every name should be unique whether the name is used for source or destination. The source cannot be referenced in more than one routing rule. The destination can be referenced in more than one destination pool.

When configuring a routing rule, the incoming and outgoing protocol used for one routing rule should be the same. For example, if routing rule A has source configured with TCP, destinations in the destination pool in routing rule A must be configured with the same TCP. Likewise if the source is configured with UDP, destinations in the same routing rule must be configured with UDP.

- **sources:** List of sources
 - ◆ **source:** Specifies the data ingress.
 - **name:** Specifies the unique name that identifies the source.
 - **type:** Specifies the source type. Valid source types are `file` and `syslog`.
 - Specify `syslog` if the events are fed from a syslog server or syslog connector.
 - Specify `file` if files are to be distributed to a destination.
 - **host:** If the source is configured as a `file` type, specify the host IP address, host name, or FQDN from which Load Balancer will download the files.
 - **protocol:** Specifies the protocol that Load Balancer will use to listen:
 - For `syslog` type, `tcp` and `udp` are supported.
 - For `file` type, `ftp` is supported.
 - **port:** Specifies the port used:
 - For the `syslog` type, specify the port where Load Balancer will be listening. All port numbers must be unique or there will be a binding error.
 - For the `file` type, this specifies the FTP server port to which Load Balancer connects to download files. Skip the configuration of this value if FTP server is configured with the default port. (The default FTP port is 21.)

The following configuration values apply only to the `file` type source.

- **path:** Specifies the path where the files are located in the FTP server. This path should be based off the FTP root directory. For example, if the FTP root directory is configured as `/ftp/pub` and the files are located under `/ftp/pub/source`, specify `/source` to this value.
- **username:** Specifies the username used to log into the FTP server.
- **password:** Specifies the password configured for the user. The plaintext password is encrypted and persisted during the Load Balancer startup.
- **fileFilter:** Specifies the Java-style regular expression used to filter the files to download from the specified path. For example, `.*log` will filter the files with names ending with 'log'. To download the files with names starting with 'Simple' and ending with 'log', use `Simple.*log`.

-
- When uploading files to the source FTP server, be sure to use a temporary file name and specify the filter in the `fileFilter` parameter to filter out temporary files. After file upload is complete, rename the file to an original name.
 - `recursive`: Specifies that Load Balancer downloads the file recursively from subdirectories when set to `true`. By default, it is set to `true`.
 - Note that Load Balancer cannot handle a file larger than 1 GB – 1 byte (approximately 1GB). When Load Balancer detects a file that exceeds the maximum size, it logs an error message.
 - `localWorkDirectory`: Specifies the existing path to the directory where Load Balancer will place the downloaded files temporarily before they are actually sent out to the destination. Configuring this field is required when FTP is configured with the `file` source type.
 - `moveToDirectory`: Specifies the directory to which the files should be moved after the files are successfully delivered to the destination. This directory can be created as a sub-directory under the source directory that is specified in `path` or can be located in another location. In the case of moving the files to a sub-directory of the source files, be sure to provide a name that starts with dot so that it is treated as a hidden directory such as `.done`. Otherwise it will recursively download the files from sub-directories and move them to another nested sub-directory unless `recursive` is set to `false`. See the option to turn off recursive search. If this is set to `blank` or omitted, files are deleted instead.
 - `passive`: FTP server can be configured to run in passive or active mode. Set this value to `true` if FTP server is running in passive mode. Otherwise set to `false`.
- `destinations`: List of destinations
- ◆ `destination`: Specifies a destination where events or file will be sent.
 - ◆ Only connectors are supported as destinations. Identify the connectors to be used as destinations and configure the following values:
 - `name`: Specifies a unique name that identifies the destination.
 - `type`: Specifies the destination type. Valid destination types are `file` and `syslog`. The type should match the connector type. If a `syslog` connector is used as a destination, configure `type` as `syslog`. If the connector is reading files from certain directory such as a Bluecoat connector, configure `type` as `file`.
 - `host`: Specifies the destination address where the connector is installed. Applicable whether the `type` is set to `syslog` or `file`.
 - `protocol`: Specify the protocol used to send data to the destination connector.
 - For `syslog` type, `tcp` and `udp` are supported.
 - For `file` type, `ftp` and `scp` are supported. When `ftp` is used, the connector installation host should be running an FTP server to receive the file from Load Balancer.
 - `port`: For `syslog` destinations, this value specifies the configured port where the connector listens for events. This port number should match the port number found in the `agent.properties` file of the destination connector. For `file` type, this can be skipped if the default ports are being used. The default FTP port is 21 and SCP is 22.

The following configuration values are applicable only to `file` type source.

- `username`: Specifies the username used to log into the FTP server or to run an `scp` command.
- `password`: Specifies the password set for the user. This password will also be encrypted when the Load Balancer starts up.
- `path`: Specifies the path to where the files will be moved. If protocol is configured as `ftp`, this path should be relative to the FTP root directory. For `scp`, it should be a full path.
- `knownHostsFile`: Specifies the file path of known hosts file if the destination type is `file` and protocol is `scp`. This file should contain the host key used for SSH connections, which is usually added to `$HOME/.ssh/known_hosts` on an initial SSH connection to a specified host. Specify the path of default known hosts file or the one created for Load Balancer testing, if it exists. Note that `$ARCSIGHT_HOME/user/loadbalancer` is assumed to be the base directory if the path does not start with `"/`. Currently `ssh-rsa` or `ssh-dsa` are accepted as valid algorithms.

Specify the destination connector information using in the following section to enable Load Balancer to communicate with the connector and to check the health and load. Before configuring the following values, first go to the connector installation directory on the machine where the connector is installed and ensure that remote management is enabled and get the port number configured for remote management. Corresponding property names for these two values in `agent.properties` are `remote.management.enabled` and `remote.management.listener.port`. If the destination is a file connector, the agent name— which is specified near the end of the installation wizard process during the connector configuration step and persisted into destination descriptor— will be needed. Note that agent name within the container should be unique when more than one connector is configured within the container.

- `additionalParameters/properties`
 - `remote.management.listener.port`: Specifies the value of `remote.management.listener.port` of the destination connector.
 - `agent.name`: Provide this value only if the destination is a file connector.

When you install the connector, the default port and username are specified under the connector installation in `config/agent/agent.defaults.properties` as `remote.management.listener.port` and `remote.management.user`. If you change these property values to anything other than the default, then these properties are present in `agents.properties` under `user/agent`. In that case, you must update these values to the correct values in the `agents.properties` file.

For management of certificates from destinations, Load Balancer creates a directory on the Load Balancer machine such as `<ARCSIGHT_HOME>/certs` and populates downloaded certificates from the destinations under this folder. The automatic download and import of certificates is done in the background.

 **IMPORTANT:**

On remote connector installations (destination connectors), turn on the remote management enabled flag. In the `user/agent/agents.properties` file, add `remote.management.enabled=true`. Do this before starting the connectors.

 Start the destination connectors before starting Load Balancer. Doing so ensures Load Balancer is able to query destinations for usage, load, and health statistics. More importantly, this also lets Load Balancer contact the connector host on the provided port and download the certificates for the connectors, which then enables the destination monitoring. If the connector is not up when Load Balancer starts, Load Balancer will check periodically to see if the connector comes up and then includes it for destination monitoring.

- `destinationPools`: List of destination pools
 - ◆ `destinationPool`: Specifies the destination group that can handle the same type of events. All destinations in one destination pool must be of the same type.
 - `name`: Specifies a unique name that identifies the destination pool.
 - `destinations`: Specifies comma-separated destination names. Valid destination names are the ones already configured.

 Only destination names configured under the `destination` section can be used here.

- `routingRules`: List of routing rules
 - ◆ `routingRule`: Specifies the routing rule that defines the data flow. Data received on the source will be distributed to the destinations in the destination pool.
 - `name`: Specifies a unique name that identifies the routing rule.
 - `sourceName`: Specifies the source name that is configured in `sources`.
 - `destinationPoolName`: Specifies the name of the destination pool that is configured in `destinationPools`.
 - `routingPolicy`: Specifies the routing policy algorithm. Valid routing policies are `RoundRobin`, `WeightedRoundRobin`, and `AggregationPreferred` for `syslog` type. For `file` type, only `RoundRobin` is supported.
 - `enabled`: Specifies activation of the routing rule if set to `true`. Otherwise, the routing rule will not be applied.

statisticsLogging

- `logInterval`: Specifies the statistics logging interval in milliseconds. By default, the statistics are logged every minute (60,000 ms).

webServer

 This configuration is required per Load Balancer installation.

- `httpsPort`: Specify the HTTPS port. By default, it uses 8443 as the listening port.
- `certificatePath`: Specifies the file path and name where the certificate is created or loaded for Load Balancer's web service. It assumes `user/loadbalancer` as the default path. Note that the directory is not created at runtime. So to have certificate in a non-default directory, the directory must exist beforehand.

-
- `keystorePath`: Specifies the file path and name where the keystore will be created and loaded for Load Balancer's web service. It assumes `user/loadbalancer` as its default path. Note that the directory is not created at runtime. To have certificate in a non-default directory, the directory must exist beforehand.

globalParameters

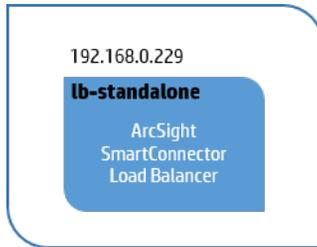
- `batch.bufferSize`: Specifies the maximum buffer size in bytes that can be used for the batch criteria. Load Balancer creates an event batch right before the total event size limit is reached.
 - `batch.eventCount`: Specifies the total number of events that can be used as the batch cut-off criteria.
 - `batch.timeout`: Specifies the timeout in milliseconds. A new batch will be created if the time reaches this value after the last batching and at least one event is waiting in the buffer.
-  Load Balancer applies these three batch parameters together using whichever condition is met first.
- `trust.store.relative.location`: Specifies the location of the trust store as a relative path in relation to the Load Balancer installation. The default value is `jre/lib/security/`, which translates to, for example:
`$ARCSIGHT_HOME/jre/lib/security`
 - `trust.store.name`: Specifies the name of the trust store specified under `trust.store.relative.location`. The default location is `cacerts`, which internally translates to, for example:
`$ARCSIGHT_HOME/jre/lib/security/cacerts` (using the trust store relative location.)
 - `trust.store.password`: Specifies the password for the trust store.
 - `destination.monitoring.interval.ms`: Specifies the time interval in milliseconds for the destinations being monitored. This applies to all destinations across all destination pools. The status and health information of the destination is queried once every time interval. The default value is `60,000ms = 1 minute`. It is not recommended to reduce this time period as excessive destination querying impairs Load Balancer's performance.
 - `weighted.interval.max`: Specifies the maximum time interval before re-calculating the load distribution for Weighted Round Robin mode. This should be at least a few times the target interval.
 - `weighted.interval.target`: Specifies the ideal time interval before re-calculating the load distribution for Weighted Round Robin mode. As event rates through Load Balancer always vary slightly, this will not be an exact number.
 - `aggregation.connector.event.ratio`: Specifies the ratio of EPS-to-queue rate below which sources will be reallocated away from a connector in Aggregation Preferred mode. This is specified as a ratio, for example, `0.9` to indicate 90%.
 - `aggregation.connector.fail.periods`: Specifies the number of consecutive monitoring intervals in which a connector must fail the above check before action is taken in Aggregation Preferred mode to prevent transient issues from causing unneeded reallocations. Note that this is based on the statistics logger configuration, not `destination.monitoring.interval.ms`.

-
- `aggregation.reallocated.warn.ratio`: Specifies the ratio of the number of connectors that must be reallocated in Aggregation Preferred mode before a warning message is sent to the log file that says there is insufficient capacity. This is specified as a ratio, for example, 0.9 to indicate 90%.
 - `aggregation.reallocation.max.ratio`: Specifies the maximum ratio of events from the previous monitoring interval to reallocate away from a connector in a single monitoring interval in Aggregation Preferred mode. This is specified as a ratio, for example, 0.9 to indicate 90%. Reallocation of sources continues until either this or the `trust.store.relative.location` limit is reached, depending on which occurs first.
 - `aggregation.reallocation.max.sources`: Specifies the maximum number of sources to reallocate away from a connector in a single monitoring interval in Aggregation Preferred mode. Reallocation of sources continues until either this or the `aggregation.reallocation.max.ratio` limit is reached, whichever occurs first.
 - `queue.max.consumer`: (TCP only) Specifies the maximum queue size for the per event source buffer, before batches are created, as a number of events. If there are a number of sources each sending a small amount of events, this should be somewhat low. If there are a few sources each sending a large amount of events, this should be somewhat high. Tune this to be roughly 1-2 seconds' worth of events, given the batching parameters and expected event rate. Note that these queues are usually nearly empty and are just to absorb load. Events that are in this queue are not persisted anywhere and will be lost if Load Balancer terminates unexpectedly. When this queue is full, no further events are read off of the socket, so event sources will experience TCP backpressure.
 - `queue.max.producer`: Specifies the maximum queue size for the per destination asynchronous buffer as a number of batches (TCP) or events (UDP). While this buffer is full, TCP will buffer batches on disk up to `persistent.queue.disk.limit` bytes, and UDP will buffer events in memory up to the `udp.events.queue.capacity` value before starting to drop events. Events in this queue are still persisted on disk for TCP.
 - `udp.consumer.threads`: Specifies the number of threads used to read UDP packets from the network.
 - `tcp.consumer.threads`: Specifies the number of threads to use to dispatch TCP event batches. If files are appearing in the `$ARCSIGHT_HOME/user/loadbalancer/lbdata/persistence/RULE-NAME` directory faster than they are disappearing, consider increasing this value, up to the number of destinations in a single destination pool.
 - `persistent.queue.disk.limit`: The maximum size, in bytes, of the persisted event batches for each TCP routing rule. This includes internal overhead, but not file system overhead. For example, if you have two TCP routing rules, you need at least two times this much disk space available, plus a margin for file system metadata and to account for block sizes. This should be at least 1073741824 (1 GiB), and must be at least twice the expected throughput per second.

Configuration Examples

Configuring MemberHosts in Standalone Mode

Load Balancer can be configured to run in standalone mode as shown in the diagram.

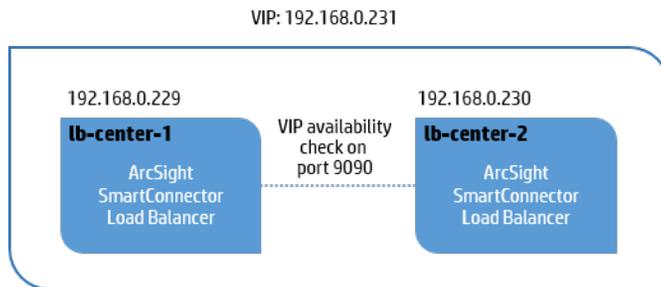


```
<memberHosts vipAddress="192.168.0.229" vipPingPort="9090">  
  <memberHost name="lb-standalone" host="192.168.0.229" port="6702" isPrimary="false" vipBindCommand="sudo  
/sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-  
scripts/ifcfg-eth0:1"/>  
</memberHosts>  
<memberIdentity>lb-standalone</memberIdentity>
```

 `vipBindCommand` and `vipUnbidCommand` have `sudo` in the command because Load Balancer is not running as root.

Configuring MemberHosts as Peer

When Load Balancer is deployed to run as peer, Load Balancer is installed on two hosts sharing the same virtual IP address. In the diagram below lb-center-1 and lb-center-2 are running as peer. The member host that starts first will run as the active member and pushes the configuration value to the other member host. Note that `isPrimary` is set to false in both configurations.



For lbConfig.xml on **lb-center-1**:

```
<memberHosts vipAddress="192.168.0.231" vipPingPort="9090">
  <memberHost name="lb-center-1" host="192.168.0.229" port="6702" isPrimary="false" vipBindCommand="sudo
/sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-
scripts/ifcfg-eth0:1"/>
</memberHosts>
<memberHost name="lb-center-2" host="192.168.0.230" port="6702" isPrimary="false" \
  vipBindCommand="sudo /sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" \
  vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
</memberHosts>
<memberIdentity>lb-center-1</memberIdentity>
```

For lbConfig.xml on **lb-center-2**:

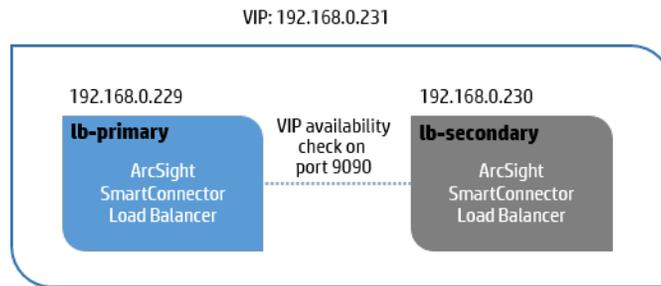
```
<memberHosts vipAddress="192.168.0.231" vipPingPort="9090">
  <memberHost name="lb-center-1" host="192.168.0.229" port="6702" isPrimary="false" vipBindCommand="sudo
/sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-
scripts/ifcfg-eth0:1"/>
</memberHosts>
<memberHost name="lb-center-2" host="192.168.0.230" port="6702" isPrimary="false" vipBindCommand="sudo
/sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-
scripts/ifcfg-eth0:1"/>
</memberHosts>
<memberIdentity>lb-center-2</memberIdentity>
```

 [vipBindCommand and vipUnbidCommand have sudo in the command because Load Balancer is not running as root.](#)

Configuring MemberHosts as Primary-Secondary

When Load Balancer is going to be deployed as primary-secondary, Load Balancer is installed in two hosts sharing the same virtual IP address. In the diagram below lb-primary is designated as the primary load balancer and the value for `isPrimary` is set to `true` for `memberHost`, while it is set to `false` for lb-secondary. Always start lb-primary member host first to have configuration synchronized to lb-secondary member host.

 `vipBindCommand` and `vipUnbidCommand` do not have `sudo` in the command because Load Balancer is running as root.



For `lbConfig.xml` on **lb-primary**:

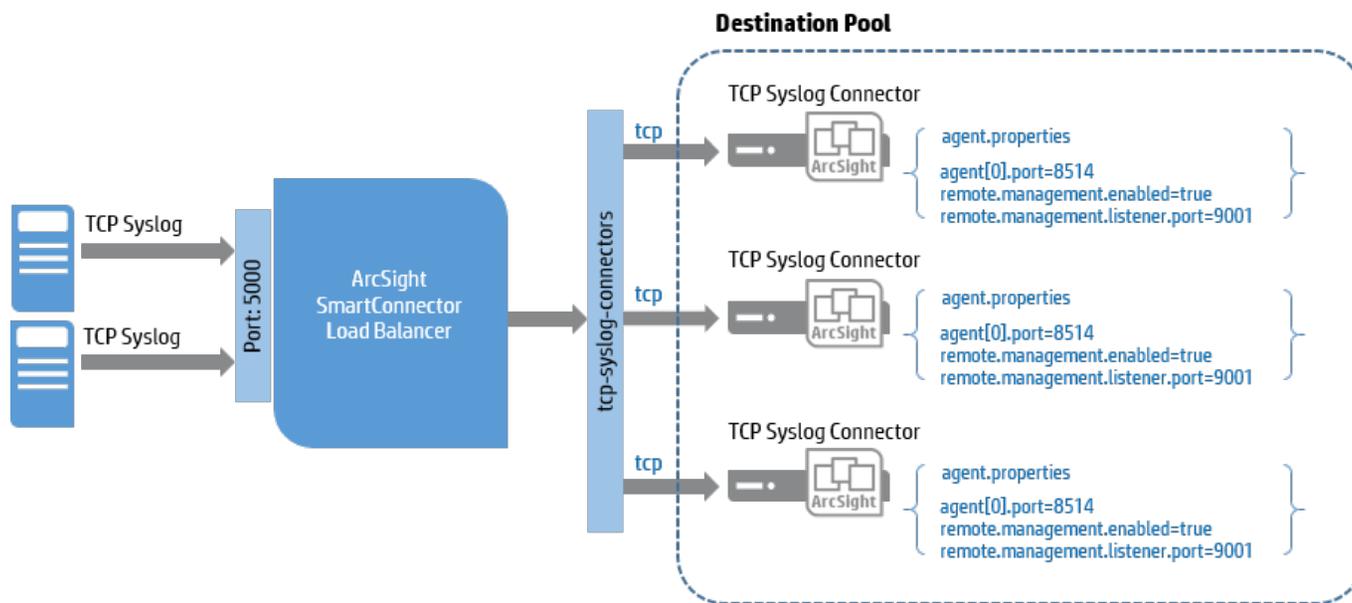
```
<memberHosts vipAddress="192.168.0.231" vipPingPort="9090">
  <memberHost name="lb-primary" host="192.168.0.229" port="6702" isPrimary="true" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-
eth0:1" />
</memberHosts>
<memberHost name="lb-center-2" host="192.168.0.230" port="6702" isPrimary="false" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-
eth0:1" />
</memberHosts>
<memberIdentity>lb-primary</memberIdentity>
```

For lbConfig.xml on **lb-secondary**:

```
<memberHosts vipAddress="192.168.0.231" vipPingPort="9090">
  <memberHost name="lb-center-1" host="192.168.0.229" port="6702" isPrimary="true" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-
eth0:1"/>
  </memberHosts>
  <memberHost name="lb-secondary" host="192.168.0.230" port="6702" isPrimary="false"
vipBindCommand="/sbin/ifup /etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown
/etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
  </memberHosts>
</memberIdentity>lb-secondary</memberIdentity>
```

Syslog Load Balancing Routing Rule Example

The following diagram illustrates two syslog servers feeding an input stream into Load Balancer on port 5000 using a TCP connection. In this scenario, Load Balancer distributes the events to three TCP syslog connectors which are grouped as one destination pool called 'tcp-syslog-connectors'. The following configuration file shows an example of how to configure the routing rule used in this scenario. Note that `remote.management.listener.port` is configured per destination. This information is used to detect the health and load of the connectors in destination pool and the connector is considered down if it is configured with an incorrect value.



<routing>

<destinationPools>

```
<destinationPool name="tcp-syslog-connectors" \
  destinations="tcp-syslog-1,tcp-syslog-2,tcp-syslog-3"/>
```

</destinationPools>

<destinations>

```
<destination name="tcp-syslog-1" type="syslog" host="192.168.0.1" \
  port="8514" protocol="tcp">
  <additionalParameters type="connector">
    <properties>
      <property key="remote.management.listener.port" value="9001"/>
    </properties>
  </additionalParameters>
</destination>
<destination name="tcp-syslog-2" type="syslog" host="192.168.0.2" \
```

```

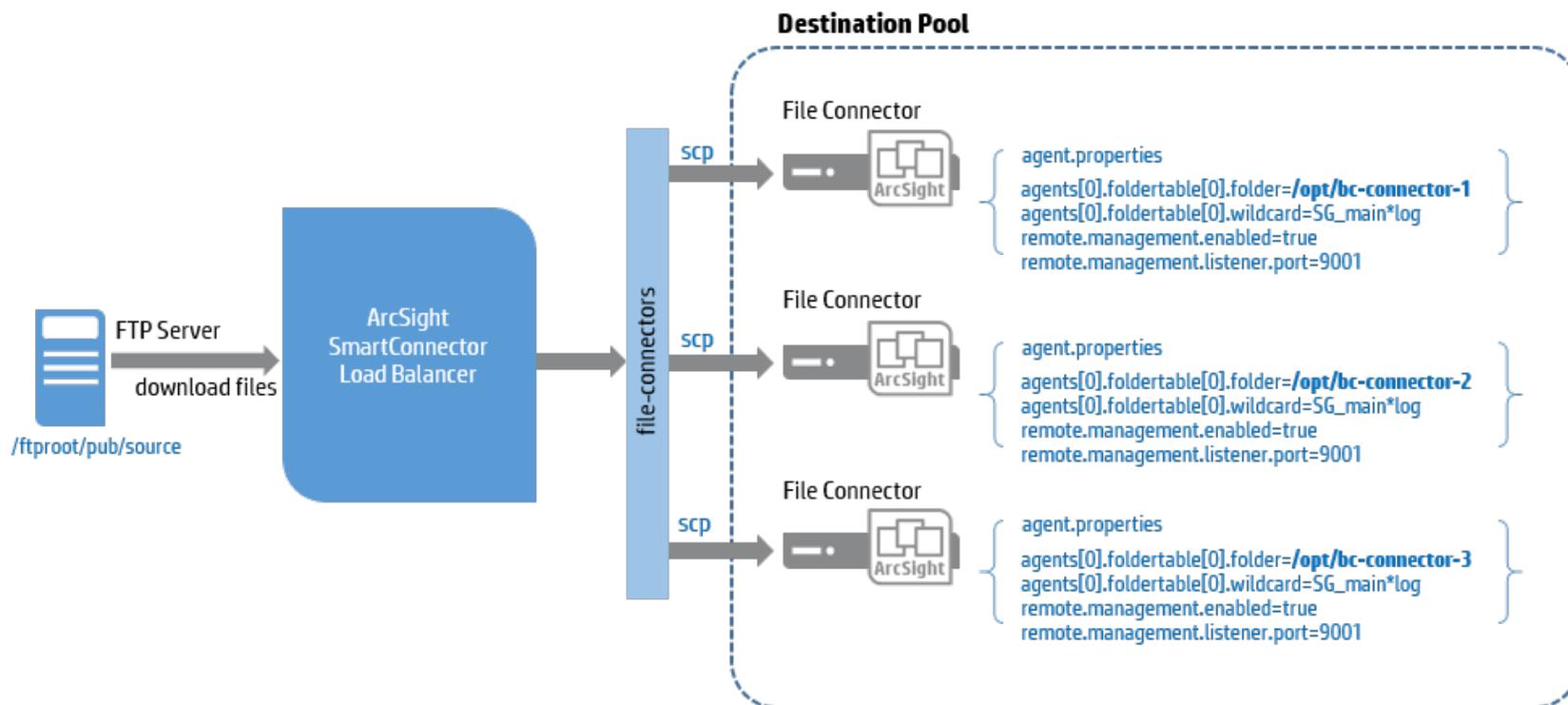
        port="8514" protocol="tcp">
    <additionalParameters type="connector">
        <properties>
            <property key="remote.management.listener.port" value="9001"/>
        </properties>
    </additionalParameters>
</destination>
<destination name="tcp-syslog-3" type="syslog" host="192.168.0.3" \
    port="8514" protocol="tcp">
    <additionalParameters type="connector">
        <properties>
            <property key="remote.management.listener.port" value="9001"/>
        </properties>
    </additionalParameters>
</destination>
</destinations>
<routingRules>
    <routingRule name="firewall-syslog" sourceName="syslog-receiver" \
        destinationPoolName="tcp-syslog-connectors" \
        routingPolicy="WeightedRoundRobin" enabled="true"/>
</routingRules>
<sources>
    <source name="syslog-receiver" type="syslog" port="5000" \
        protocol="tcp"/>
</sources>
</routing>

```

File Load Balancing Routing Rule Example

The following diagram shows a configuration example of a routing rule for file load balancing. To define the source, an FTP host address, the credentials, and the path need to be defined. Here, the FTP root directory is `/ftproot/pub`, so the actual location to FTP client should be the `source` directory. The rest of the routing rule configuration is similar to syslog routing rule configuration.

- When uploading files to the source FTP server, be sure to use a temporary file name and specify the filter in the `fileFilter` parameter to filter out temporary files. After file upload is complete, rename the file to an original name.
- Note that the agent name for file connectors is specified in this configuration.



```
<routing>
  <destinationPools>
    <destinationPool name="file-connectors" \
      destinations="file-connector-1,file-connector-2,file-connector-3"/>
  </destinationPools>
</routing>
```

```
</destinationPools>
<destinations>
  <destination name="file-connector-1" type="file" \
    path="/opt/bc-connector-1" host="192.168.0.1" protocol="scp" \
    username="admin" password="password" \
    knownHostsFile="/home/arcsight/.ssh/known_hosts">
    <additionalParameters type="connector">
      <properties>
        <property key="remote.management.listener.port" value="9001"/>
        <property key="agent.name" value="bc-connector-1"/>
      </properties>
    </additionalParameters>
  </destination>
  <destination name="file-connector-2" type="file" \
    path="/opt/bc-connector-2" host="192.168.0.2" protocol="scp" \
    username="admin" password="password" \
    knownHostsFile="/home/arcsight/.ssh/known_hosts">
    <additionalParameters type="connector">
      <properties>
        <property key="remote.management.listener.port" value="9001"/>
        <property key="agent.name" value="bc-connector-2"/>
      </properties>
    </additionalParameters>
  </destination>
  <destination name="file-connector-3" type="file" \
    path="/opt/bc-connector-3" host="192.168.0.3" protocol="scp" \
    username="admin" password="password" \
    knownHostsFile="/home/arcsight/.ssh/known_hosts">
```

```
<additionalParameters type="connector">
  <properties>
    <property key="remote.management.listener.port" value="9001"/>
    <property key="agent.name" value="bc-connector-3"/>
  </properties>
</additionalParameters>
</destination>
</destinations>
<routingRules>
  <routingRule name="file-rule" sourceName="file-watcher" \
    destinationPoolName="file-connectors" routingPolicy="RoundRobin" \
    enabled="true"/>
</routingRules>
<sources>
  <source name="file-watcher" type="file" path="/source" \
    host="192.168.1.225" protocol="ftp" username="admin" \
    password="OBFUSCATE.1:B8R3Ts5XXui0aBjFn1Js7Q==" \
    moveToDirectory=".done" fileFilter="SG_main.*log" \
    localWorkDirectory="/tmp" recursive="true" passive="false" />
</sources>
</routing>
```

Sample Configuration File

The following example shows the Load Balancer configuration file configured for syslog load balancing. The template provided by Load Balancer is shown in Appendix A.

 This is a sample configuration file with passwords obfuscated since it was captured after Load Balancer started.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lbConfiguration>
  <memberHosts vipAddress="192.168.5.231" vipPingPort="9090">
    <memberHost name="primary-node" host="192.168.5.229" port="7701" isPrimary="false" vipBindCommand="sudo /sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="sudo /sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-
eth0:1"/>
    <memberHost name="secondary-node" host="192.168.5.230" port="7701" isPrimary="false" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
  </memberHosts>
  <memberIdentity>primary-node</memberIdentity>
  <notification enable="true">
    <enabledNotification>
      <event name="MemberHostUp" message="Member node is up"/>
      <event name="MemberHostDown" message="Member node is down"/>
      <event name="DestinationUp" message="Destination is up"/>
      <event name="DestinationDown" message="Destination is down"/>
    </enabledNotification>
  </notification>
  <email>
    <prefix>[Load Balancer]</prefix>
    <recipients>chris.smith@example.com</recipients>
    <sender>chris.smith@example.com</sender>
    <smtpServer>englab-mail.arst.usa.example.com</smtpServer>
  </email>
</notification>
<routing>
  <destinationPools>
    <destinationPool name="tcp-syslog-connectors" destinations="tcp-syslog-1,tcp-syslog-2"/>
    <destinationPool name="bc-file-connectors" destinations="bc-connector-1,bc-connector-2"/>
  </destinationPools>
  <destinations>
    <destination name="tcp-syslog-1" type="syslog" host="192.168.0.11" port="8514" protocol="tcp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="9001"/>
        </properties>
      </destination>
    </destinations>
  </routing>
</lbConfiguration>
```

```

        </additionalParameters>
    </destination>
    <destination name="tcp-syslog-2" type="syslog" host="192.168.0.12" port="8514" protocol="tcp">
        <additionalParameters type="connector">
            <properties>
                <property key="remote.management.listener.port" value="9001"/>
            </properties>
        </additionalParameters>
    </destination>
    <destination name="bc-connector-1" type="file" path="/opt/ArcSightSmartConnectors/bc-connector/current/file-
feeds" host="192.168.0.1" protocol="scp" username="admin" password="OBFUSCATE.1:wAeVeZlW8m4Cq2wLEupkJA==" recursive="false"
passive="false" knownHostsFile="/home/arcsight/.ssh/known_hosts">
        <additionalParameters type="connector">
            <properties>
                <property key="remote.management.listener.port" value="9001"/>
                <property key="agent.name" value="bc-connector"/>
            </properties>
        </additionalParameters>
    </destination>
    <destination name="bc-connector-2" type="file" path="/opt/ArcSightSmartConnectors/bc-connector/current/file-
feeds" host="192.168.0.2" protocol="scp" username="admin" password="OBFUSCATE.1:wAeVeZlW8m4Cq2wLEupkJA==" recursive="false"
passive="false" knownHostsFile="/home/arcsight/.ssh/known_hosts">
        <additionalParameters type="connector">
            <properties>
                <property key="remote.management.listener.port" value="9001"/>
                <property key="agent.name" value="bc-connector"/>
            </properties>
        </additionalParameters>
    </destination>
</destinations>
<routingRules>
    <routingRule name="syslog-tcp-rule" sourceName="syslog-tcp" destinationPoolName="tcp-syslog-connectors"
routingPolicy="WeightedRoundRobin" enabled="true"/>
    <routingRule name="bc-file-rule" sourceName="bc-ftp-server" destinationPoolName="bc-file-connectors"
routingPolicy="RoundRobin" enabled="true"/>
</routingRules>
<sources>
    <source name="syslog-tcp" type="syslog" host="192.168.8.12" port="8002" protocol="tcp"/>
    <source name="bc-ftp-server" type="file" path="bc-files" host="192.168.8.34" protocol="ftp" username="arcsight"
password="OBFUSCATE.1:y05cvjSnFlVybZFBBE0HiQ==" recursive="true" passive="true" localWorkDirectory="/tmp"
fileFilter="*.log"/>
</sources>
</routing>
<statisticsLogging logInterval="60000" />

```

```
<webServer httpsPort="8443" certificatePath="loadbalancer.cer" keystorePath="loadbalancer.p12"/>
</lbConfiguration>
```

Starting Load Balancer

Start Load Balancer from the host that has been fully configured first (primary or active) to allow the configuration to correctly sync to the passive or secondary node. Otherwise, the configuration information will not be passed properly if the passive or secondary host is started first.

To start Load Balancer:

1. Go to \$ARCSIGHT_HOME directory and start Load Balancer with the following command:

```
# bin/arcsight loadbalancer
```

2. Log on to the secondary machine, if one exists, and start Load Balancer.

 In primary-secondary HA mode, if the secondary node starts before the primary node, the configuration will not be copied over properly.

Installing Load Balancer as a Service

To run Load Balancer as a service, the initial installation must be completed and there must be a working configuration file. Validate the working configuration file by running Load Balancer as a standalone application, as shown in [Starting Load Balancer](#).

To install the files needed to run Load Balancer as a service:

1. Go to the \$ARCSIGHT_HOME directory and run the following command as root:

```
# bin/arcsight loadbalancer_service -i
```

This command installs the files necessary for running Load Balancer as a service.

2. Run the command without any switches to see the usage:

```
# bin/arcsight loadbalancer_service
```

```
ServiceTool - ArcSight SmartAgent Service Tool
```

```
Version : 1.0
```

```
Copyright (c) 2001-2015 Hewlett-Packard Development Company, L.P.
```

```
Confidential commercial computer software. Valid license required.
```

```
Usage: ServiceTool <parameters>
```

```
Optional Parameters:
```

```
-sd <description> Service/Script description (Install only) (Load Balancer for  
Arcsight SmartConnectors)
```

```
-sn <name> Service/Script name (Install only) (connlb)
```

Options:

-h help - Get help for this command
-i install - Installs the SmartConnectors LoadBalancer as a service
-r remove - Removes the SmartConnectors LoadBalancer Service

 If you are not the root user, an error message displays when invoking service tool:

```
# bin/arcsight loadbalancer_service
Assuming ARCSIGHT_HOME: /home/arcsight/betal/current
Assuming JAVA_HOME: /home/arcsight-1/betal/current/jre

ArcSight Load Balancer Service Tool starting...
*****
ERROR: This program should be run as [root]. Exiting...
*****
```

3. Once the service `arc_connlb` has been created, it can be accessed with service commands. For example:

```
# service arc_connlb status

Running as root

Output will be logged to $ARCSIGHT_HOME/current/logs/lb.out.wrapper.log
Getting status of ArcSight Load Balancer for Arcsight SmartConnectors...
ArcSight Load Balancer for Arcsight SmartConnectors is not running.
```

4. (Optional) To give the service a different name, use the `-sn` switch during service installation. The line below shows the service name changed to `arc_loadbalancer`. (The `arc_` is added before all services names.) If no other name is suggested, the default service name is `'connlb'`.

```
# bin/arcsight loadbalancer_service -sn loadbalancer -i
```

5. Use the `-sd` switch to give a different service description. For example, change the service description to `'LBService'`:

```
# bin/arcsight loadbalancer_service -i -sd LBService
```

6. Using the service description change shown step 5, the status command displays as follows:

```
# service arc_connlb status

Running as root

Output will be logged to $ARCSIGHT_HOME/current/logs/lb.out.wrapper.log
Getting status of ArcSight LBService...
ArcSight LBService is not running.
```

7. To remove the service files, use the following command:

```
# bin/arcsight loadbalancer_service -r
```

Starting or Stopping the Load Balancer Service

After you have installed Load Balancer as a service, you can start or stop the service at any time. You are expected to be a `root` user to run Load Balancer as a service.

To start or stop the Load Balancer service:

 Only root users may run Load Balancer as a service.

1. To start the Load Balancer service, use the following command:

```
# /etc/init.d/arc_connlb start
```

or

```
service arc_connlb start
```

 If you changed the default service name (`connlb`), use that name in place of `'connlb'`.

2. To stop the Load Balancer service, use the following command:

```
# /etc/init.d/arc_connlb stop
```

or

```
service arc_connlb stop
```

Load Balancer Service Commands

Other commands available when running Load Balancer as a service include:

- `start` — Starts Load Balancer as a service.
- `stop` — Stops Load Balancer as a service.
- `restart` — Stops the service, if it's running, then restarts it.
- `dump` — Captures the current JVM state including all the running threads and their states. The output will be present in `lb.out.wrapper.log`. The Load Balancer service continues to run normally after the dump. This command needs Load Balancer to be running or the command will have no effect.
- `console` — Runs the Load Balancer service as an application from the current window, which can be stopped with a `Ctrl + c` or with the `stop service` command from another window. The log will be displayed on the console.

Usage is displayed if no command is given. For example:

```
# service arc_connlb
```

```
Running as root
```

```
Output will be logged to $ARCSIGHT_HOME/current/logs/lb.out.wrapper.log
```

```
Usage: $ARCSIGHT_HOME/current/bin/lb.wrapper.sh { start | stop | restart
| dump | status | console }
```

Load Balancer Service-related Logs

The log will be redirected to `lb.out.wrapper.log` under the `logs` directory.

Interpreting Logs

Statistics are divided by:

- Routing rule—each rule is on its own line
- Locations—from sources or to destinations
- Metrics—bytes, events, and batches
- Time units—average per second SLC, total SLC, and total since startup

For each routing rule, the combined totals from every source are listed first, followed by the combined totals to every destination, and the individual statistics per destination.

Overall statistics:

```
2015-08-24 08:43:25,556
[INFO][statisticsLogging][com.arcsight.lb.stats.StatLoggingTask][run] - Load
Balancer statistics {metric=average per second-SLC/SLC/Total}:
2015-08-24 08:43:25,557
[INFO][statisticsLogging][com.arcsight.lb.stats.StatLoggingTask][run] -
Routing rule=[syslog-tcp-rule-1]:
[src=(total),bytesRcvd=1636923/16369232/188733127,eventsRcvd=10000/100000/115
3000,batchesRcvd=10/100/1153],
[dest=(total),bytesSent=1636923/16369232/196591275,eventsSent=10000/100000/12
01000,batchesSent=10/100/1201],[dest=tcp-syslog-
1,bytesSent=818452/8184521/98377554,eventsSent=5000/50000/601000,batchesSent=
5/50/601],[dest=tcp-syslog-
2,bytesSent=818471/8184711/98213721,eventsSent=5000/50000/600000,batchesSent=
5/50/600]]
2015-08-24 08:43:25,557
[INFO][statisticsLogging][com.arcsight.lb.stats.StatLoggingTask][run] -
Routing rule=[syslog-tcp-rule-1]: In: EPS=[10000] Bytes/s=[1636923] Out:
EPS=[10000] Bytes/s=[1636923]
```

Chapter 3 — Load Balancer Troubleshooting

This chapter contains information about troubleshooting for common issues.

Troubleshooting

Load Calculators Not Initialized/Destination Monitoring Not Working

Issue: Load calculators are not initialized and destination monitoring is not working, but the certificate import is fine.

```
2015-07-02 12:16:39,266 [ERROR][com.arcsight.lb.b.b][intialize] - Please check
the credentials for Connector tcp-syslog-connector-12 Error Message [; nested
exception is:
```

```
    java.net.ConnectException: Connection refused]
```

```
2015-07-02 12:16:39,266 [ERROR][com.arcsight.lb.b.c][initDestinationLoadCalculator]
- Failed to initialize Load Calculator for the destination [tcp-syslog-connector-12]
```

Answer: This problem can be resolved by:

1. Verifying that connector can be managed remotely by checking the value of `remote.management.enabled` in the `agent.properties` file. This value should be set to `true`.
2. Verifying that the `destination/additionalParameters/properties/property@remote.management.listener.port` matches the value of `remote.management.listener.port`.

Destination Configured with SCP Protocol but File Delivery Fails

Issue: Load Balancer cannot successfully log into the destination host with SCP.

Answer: It may log an exception. There are several possible reason for this error:

- `knownHostsFile` is not specified in the `lbConfig.xml` configuration file.
- `knownHostsFile` is configured, but the host key for this specific host was not found.
- `knownHostsFile` is configured and the host key was found, but the algorithm generated for the host key is neither RSA nor DSA. Currently Load Balancer supports only these two types of algorithms. If the host is configured to use another algorithm in generating a host key, regenerate the host key using one of the accepted algorithms.

Sources Relocated Away from [x] of [y] Destinations in Routing Rule

Issue: "Sources were reallocated away from [x] of [y] destinations in the routing rule [my-routing-rule]. You may wish to add more destinations" displays even though the "There was no incoming data" error message is being displayed.

Answer: The destination overloaded message is triggered by examining the Connector's internal statistics, regardless of the traffic that Load Balancer is sending it. It is therefore possible for a connector to be deemed overloaded even though Load Balancer has not yet sent it any traffic.

Appendix A – Configuration File Templates with Callout Information

Standalone Mode Configuration Template File

```
<?xml version="1.0" encoding="UTF-8"?>
<lbConfiguration>
  <!-- Identify the current host among the memberHosts. -->
  <memberIdentity>primary-node</memberIdentity>

  <!-- Load Balancer can run in standalone mode. -->
  <!-- To run Load Balancer in standalone mode, configure one memberHost. vipAddress and -->
  <!-- vipPingPort cannot be null but it won't be referenced. -->
  <memberHosts vipAddress="192.168.1.253" vipPingPort="9090">
    <!-- 'host' is the host address where Load Balancer is installed and 'port' is internally -->
    <!-- used to communicate with another Load Balancer to detect the health for HA support. -->
    <!-- Standalone mode still requires valid port number to be specified. -->
    <!-- If Load Balancer is running as non-root, add Load Balancer user to sudoer list and -->
    <!-- prefix 'vipBindCommand' and 'vipUnbindCommand' with 'sudo' such as 'sudo /sbin/ifup..' -->
    <memberHost name="primary-node" host="192.168.1.253" port="6702" isPrimary="true" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
  </memberHosts>
  <!-- To get an email notification when the Load Balancer member host is down or up, or when -->
  <!-- the destination is down or up, set enable to 'true' and configure the email section. -->
  <notification enable="true">
    <enabledNotification>
      <event name="MemberHostUp" message="Member node is up." />
      <event name="MemberHostDown" message="Member node is down." />
      <event name="DestinationUp" message="Destination is up." />
      <event name="DestinationDown" message="Destination is down." />
    </enabledNotification>
    <email>
      <!-- Create a prefix for the subject line. -->
      <prefix>[Load Balancer]</prefix>
      <!-- Separate multiple recipients with a space. -->
      <recipients>jane.doe@abc.com john.doe@abc.com</recipients>
      <sender>admin@abc.com</sender>
      <smtpServer>smtp.abc.com</smtpServer>
    </email>
  </notification>
  <routing>
    <!-- All names in the routing section must be unique. -->
    <destinationPools>
      <destinationPool name="tcp-syslog-connectors" destinations="syslog-connector-1,syslog-connector-2"/>
    </destinationPools>
  </routing>
</lbConfiguration>
```

```

        <destinationPool name="udp-syslog-connectors" destinations="syslog-connector-3,syslog-connector-4"/>
        <destinationPool name="file-connectors" destinations="file-connector-1,file-connector-2"/>
</destinationPools>
<destinations>
  <!-- Examples of configuring TCP connectors as destinations. -->
    <!-- 'host' is the host address where the tcp connector is running and 'port' is connector's listening
-->
    <!-- port which can be found from agent.properties. 'tcp' corresponds to 'Raw TCP' in agent.properties.
-->
    <destination name="syslog-connector-1" type="syslog" host="192.168.1.12" port="513" protocol="tcp">
      <!-- Specify the connection configuration value here with key and matching value. Load Balancer need -->
      <!-- the information to perform the connector health check and to obtain the load information.
-->
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <destination name="syslog-connector-2" type="syslog" host="192.168.1.13" port="513" protocol="tcp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <!-- Examples of configuring UDP connectors as destinations -->
    <destination name="syslog-connector-3" type="syslog" host="192.168.1.12" port="514" protocol="udp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <destination name="syslog-connector-4" type="syslog" host="192.168.1.13" port="514" protocol="udp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <!-- Examples of configuring file-based connectors as destination. -->
    <!-- Supported protocols on the destination side are ftp and scp. Each protocol requires a different -->

```

```

->         <!-- set of configuration values as shown in the examples below. Plaintext passwords are persisted as -
->         <!-- encrypted value when Load Balancer starts. -
->         <!-- In order to use 'scp' protocol, file that has ssh host key should be provided to 'knownHostsFile'. -
->         <!-- Load Balancer does not populate this file automatically. In order to obtain the host key, ssh to -
->         <!-- the destination manually and specify the full path of the system default known_hosts file or copy -
->         <!-- the file to another location and give that path to 'knownHostsFile'. -
->
->         <destination name="file-connector-1" type="file" path="/opt/connector-1/input" host="192.168.0.1" protocol="scp"
username="admin" password="password" knownHostsFile="/root/.ssh/known_hosts">
->         <!-- Configure the information about this connector before starting Load Balancer. -->
->         <additionalParameters type="connector">
->         <properties>
->             <!-- If the destination connector is file-based connector, specify the following two -->
->             <!-- values so that Load Balancer can handshake with the connector. -->
->             <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
->             <property key="agent.name" value="{agent name configured for the destination in the final stage of
agent setup}"/>
->         </properties>
->         </additionalParameters>
->         </destination>
->         <!-- Configure the port if FTP server is configured with a non-default port. Default port 21 is used if not
specified. -->
->         <!-- 'host' is the host address of FTP server and 'username' and 'password' is the user credential who
has access to -->
->         <!-- FTP server. Plaintext password is encrypted and persisted to this file as soon as Load Balancer
starts. -->
->         <destination name="file-connector-2" type="file" host="192.168.0.2" protocol="ftp" username="admin"
password="password" path="landing">
->         <additionalParameters type="connector">
->         <properties>
->             <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
->             <property key="agent.name" value="{agent name configured for the destination in the final stage of
agent setup}"/>
->         </properties>
->         </additionalParameters>
->         </destination>
->     </destinations>
->     <routingRules>
->         <!-- Supported routing policies are RoundRobin and WeightedRoundRobin. -->

```

```

        <routingRule name="syslog-tcp-rule" sourceName="syslog-tcp" destinationPoolName="tcp-syslog-connectors"
routingPolicy="RoundRobin" enabled="true"/>
        <routingRule name="syslog-udp-rule" sourceName="syslog-udp" destinationPoolName="udp-syslog-connectors"
routingPolicy="WeightedRoundRobin" enabled="true"/>
        <routingRule name="file-routing-rule" sourceName="file-watcher" destinationPoolName="file-connectors"
routingPolicy="RoundRobin" enabled="true"/>
    </routingRules>
    <sources>
        <!-- When the source is syslog type, set 'type' to 'syslog' and configure the protocol accordingly. -->
        <!-- Supported protocols are 'udp' and 'tcp'. 'port' is the listening port on Load Balancer and -->
            <!-- source syslog server should be configuredly to send the events to this port. -->
        <source name="syslog-tcp" type="syslog" port="513" protocol="tcp"/>
        <source name="syslog-udp" type="syslog" port="514" protocol="udp"/>
        <!-- 'file' type source can be used with the file based connector and Load Balancer downloads -->
        <!-- the files from FTP server and distribute them to the defined destinations in destination pool. -->
        <!-- Supported protocol is 'ftp' and 'host' is the host address where FTP server is running. -->
        <!-- It assumes a default FTP port 21. If other port is configured, add 'port' attribute port and -->
        <!-- specify the port number. 'path' should a relative path to FTP root directory. -->
        <!-- Specify the credential for one who has a permission in accessing files from FTP server with -->
        <!-- a full permission since files will need to moved to another directory or deleted after -->
        <!-- the file is downloaded. When 'moveToDirectory' is configured, the downloaded file will be -->
        <!-- to a specified directory or when the value is empty, file will be deleted afterwards. -->
        <!-- If the specified path is located under 'path', be sure to give the path as hidden directory -->
        <!-- starting with '.'. Otherwise it will attempt to download the files from the directory again -->
        <!-- since it will recursively look for sub-directories. To disable recursive lookup, set false to -->
        <!-- 'recursive' attribute. -->
        <!-- User credential who has access to FTP server and path should be specified in 'username' and -->
            <!-- 'password'. Plaintext password will be converted as encrypted value as soon as Load Balancer -->
            <!-- starts. -->
        <!-- 'localWorkDirectory' is where the file is temporarily kept before the file is sent to one of -->
        <!-- the destinations. This is required value and it assumes that the directory exists already. -->
        <source name="file-watcher" type="file" host="192.168.0.2" protocol="ftp" path="landingzone" username="admin"
password="password" fileFilter="*.log" moveToDirectory=".done" recursive="true" passive="false" localWorkDirectory="/tmp" />
    </sources>
</routing>
<!-- logInterval is in milliseconds. -->
<statisticsLogging logInterval="60000"/>
<!-- WebServer must be configured for all nodes listed as member hosts in Load Balancer. -->
<!-- Load Balancer creates certificate and keystore on startup on specified location. The base -->
    <!-- directory is user/loadbalancer unless a full path is specified. -->
<webServer httpsPort="8443" certificatePath="loadbalancer.cer" keystorePath="loadbalancer.p12"/>
<!-- Uncomment and configure this section in order to customize the configuration. -->
<!-- Refer to the configuration guide for the details. -->
<!-- globalParameters>
    <properties>
        <property key="batch.buffer.size" value="102400" />
    </properties>

```

```
</globalParameters -->  
</lbConfiguration>
```

HA Mode Configuration Template File

```
<?xml version="1.0" encoding="UTF-8"?>
<lbConfiguration>
  <!-- Identify the current host among the memberHosts. -->
  <memberIdentity>primary-node</memberIdentity>

  <!-- Load Balancer can run in HA mode: two hosts can run primary-secondary or peer. -->
  <!-- (1) To run Load Balancer as primary-secondary, configure two memberHosts and -->
  <!-- for one of the hosts, and set isPrimary to 'true'. -->
  <!-- (2) To run Load Balancer as peer, configure two memberHosts and set isPrimary to -->
  <!-- 'false' for both. -->
  <!-- 'vipAddress' is the virtual IP address that will be shared between two member hosts to -->
  <!-- handle seamless failover of member host. 'vipPingPort' is internally used to check if -->
  <!-- VIP address is still bound to one of the member hosts for continuous event collection. -->
  <!-- Specify any unused port to 'vipPingPort'. -->
  <memberHosts vipAddress="192.168.1.255" vipPingPort="9090">
    <!-- 'host' is the host address where Load Balancer is installed and 'port' is internally -->
    <!-- used to communicate with another Load Balancer to detect the health for HA support. -->
    <!-- If Load Balancer is running as non-root, add Load Balancer user to sudoer list and -->
    <!-- prefix 'vipBindCommand' and 'vipUnbindCommand' with 'sudo' such as 'sudo /sbin/ifup..'. -->
    <memberHost name="primary-node" host="192.168.1.253" port="6702" isPrimary="true" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
    <memberHost name="secondary-node" host="192.168.1.254" port="6702" isPrimary="false" vipBindCommand="/sbin/ifup
/etc/sysconfig/network-scripts/ifcfg-eth0:1" vipUnbindCommand="/sbin/ifdown /etc/sysconfig/network-scripts/ifcfg-eth0:1"/>
  </memberHosts>
  <!-- To get an email notification when the Load Balancer member host is down or up, or when -->
  <!-- the destination is down or up, set enable to 'true' and configure the email section. -->
  <notification enable="true">
    <enabledNotification>
      <event name="MemberHostUp" message="Member node is up." />
      <event name="MemberHostDown" message="Member node is down." />
      <event name="DestinationUp" message="Destination is up." />
      <event name="DestinationDown" message="Destination is down." />
    </enabledNotification>
  </notification>
  <email>
    <!-- Create a prefix for the subject line. -->
    <prefix>[Load Balancer]</prefix>
    <!-- Separate multiple recipients with a space. -->
    <recipients>jane.doe@abc.com john.doe@abc.com</recipients>
    <sender>admin@abc.com</sender>
    <smtpServer>smtp.abc.com</smtpServer>
  </email>
  </notification>
  <routing>
    <!-- All names in the routing section must be unique. -->
    <destinationPools>
```

```

    <destinationPool name="tcp-syslog-connectors" destinations="syslog-connector-1,syslog-connector-2"/>
    <destinationPool name="udp-syslog-connectors" destinations="syslog-connector-3,syslog-connector-4"/>
    <destinationPool name="file-connectors" destinations="file-connector-1,file-connector-2"/>
</destinationPools>
<destinations>
  <!-- Examples of configuring TCP connectors as destinations. -->
    <!-- 'host' is the host address where the tcp connector is running and 'port' is connector's listening
-->
    <!-- port which can be found from agent.properties. 'tcp' corresponds to 'Raw TCP' in agent.properties.
-->
    <destination name="syslog-connector-1" type="syslog" host="192.168.1.12" port="513" protocol="tcp">
      <!-- Specify the connection configuration value here with key and matching value. Load Balancer need -->
      <!-- the information to perform the connector health check and to obtain the load information.
-->
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <destination name="syslog-connector-2" type="syslog" host="192.168.1.13" port="513" protocol="tcp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <!-- Examples of configuring UDP connectors as destinations -->
    <destination name="syslog-connector-3" type="syslog" host="192.168.1.12" port="514" protocol="udp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <destination name="syslog-connector-4" type="syslog" host="192.168.1.13" port="514" protocol="udp">
      <additionalParameters type="connector">
        <properties>
          <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
        </properties>
      </additionalParameters>
    </destination>
    <!-- Examples of configuring file-based connectors as destination.
-->

```

```

    <!-- Supported protocols on the destination side are ftp and scp. Each protocol requires a different -->
    <!-- set of configuration values as shown in the examples below. Plaintext passwords are persisted as -
->
    <!-- encrypted value when Load Balancer starts. -
->
    <!-- In order to use 'scp' protocol, file that has ssh host key should be provided to 'knownHostsFile'. -
->
    <!-- Load Balancer does not populate this file automatically. In order to obtain the host key, ssh to -
->
    <!-- the destination manually and specify the full path of the system default known_hosts file or copy -
->
    <!-- the file to another location and give that path to 'knownHostsFile'. -
->
    <destination name="file-connector-1" type="file" path="/opt/connector-1/input" host="192.168.0.1" protocol="scp"
username="admin" password="password" knownHostsFile="/root/.ssh/known_hosts">
    <!-- Configure the information about this connector before starting Load Balancer. -->
    <additionalParameters type="connector">
    <properties>
    <!-- If the destination connector is file-based connector, specify the following two -->
    <!-- values so that Load Balancer can handshake with the connector. -->
    <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
    <property key="agent.name" value="{agent name configured for the destination in the final stage of
agent setup}"/>
    </properties>
    </additionalParameters>
    </destination>
    <!-- Configure the port if FTP server is configured with a non-default port. Default port 21 is used if not
specified. -->
    <!-- 'host' is the host address of FTP server and 'username' and 'password' is the user credential who
has access to -->
    <!-- FTP server. Plaintext password is encrypted and persisted to this file as soon as Load Balancer
starts. -->
    <destination name="file-connector-2" type="file" host="192.168.0.2" protocol="ftp" username="admin"
password="password" path="landing">
    <additionalParameters type="connector">
    <properties>
    <property key="remote.management.listener.port" value="{remote.management.listener.port from
agent.properties}"/>
    <property key="agent.name" value="{agent name configured for the destination in the final stage of
agent setup}"/>
    </properties>
    </additionalParameters>
    </destination>
</destinations>
<routingRules>
    <!-- Supported routing policies are RoundRobin and WeightedRoundRobin. -->

```

```

        <routingRule name="syslog-tcp-rule" sourceName="syslog-tcp" destinationPoolName="tcp-syslog-connectors"
routingPolicy="RoundRobin" enabled="true"/>
        <routingRule name="syslog-udp-rule" sourceName="syslog-udp" destinationPoolName="udp-syslog-connectors"
routingPolicy="WeightedRoundRobin" enabled="true"/>
        <routingRule name="file-routing-rule" sourceName="file-watcher" destinationPoolName="file-connectors"
routingPolicy="RoundRobin" enabled="true"/>
    </routingRules>
    <sources>
        <!-- When the source is syslog type, set 'type' to 'syslog' and configure the protocol accordingly. -->
        <!-- Supported protocols are 'udp' and 'tcp'. 'port' is the listening port on Load Balancer and -->
            <!-- source syslog server should be configuredly to send the events to this port. -->
        <source name="syslog-tcp" type="syslog" port="513" protocol="tcp"/>
        <source name="syslog-udp" type="syslog" port="514" protocol="udp"/>
        <!-- 'file' type source can be used with the file based connector and Load Balancer downloads -->
        <!-- the files from FTP server and distribute them to the defined destinations in destination pool. -->
        <!-- Supported protocol is 'ftp' and 'host' is the host address where FTP server is running. -->
        <!-- It assumes a default FTP port 21. If other port is configured, add 'port' attribute port and -->
        <!-- specify the port number. 'path' should a relative path to FTP root directory. -->
        <!-- Specify the credential for one who has a permission in accessing files from FTP server with -->
        <!-- a full permission since files will need to moved to another directory or deleted after -->
        <!-- the file is downloaded. When 'moveToDirectory' is configured, the downloaded file will be -->
        <!-- to a specified directory or when the value is empty, file will be deleted afterwards. -->
        <!-- If the specified path is located under 'path', be sure to give the path as hidden directory -->
        <!-- starting with '.'. Otherwise it will attempt to download the files from the directory again -->
        <!-- since it will recursively look for sub-directories. To disable recursive lookup, set false to -->
        <!-- 'recursive' attribute. -->
        <!-- User credential who has access to FTP server and path should be specified in 'username' and -->
            <!-- 'password'. Plaintext password will be converted as encrypted value as soon as Load Balancer -->
            <!-- starts. -->
        <!-- 'localWorkDirectory' is where the file is temporarily kept before the file is sent to one of -->
        <!-- the destinations. This is required value and it assumes that the directory exists already. -->
        <source name="file-watcher" type="file" host="192.168.0.2" protocol="ftp" path="landingzone" username="admin"
password="password" fileFilter="*.log" moveToDirectory=".done" recursive="true" passive="false" localWorkDirectory="/tmp" />
    </sources>
</routing>
<!-- logInterval is in milliseconds. -->
<statisticsLogging logInterval="60000"/>
<!-- WebServer must be configured for all nodes listed as member hosts in Load Balancer. -->
<!-- Load Balancer creates certificate and keystore on startup on specified location. The base -->
    <!-- directory is user/loadbalancer unless a full path is specified. -->
<webServer httpsPort="8443" certificatePath="loadbalancer.cer" keystorePath="loadbalancer.p12"/>
<!-- Uncomment and configure this section in order to customize the configuration. -->
<!-- Refer to the configuration guide for the details. -->
<!-- globalParameters>
    <properties>
        <property key="batch.buffer.size" value="102400" />
    </properties>

```

```
</globalParameters -->  
</lbConfiguration>
```

