

Rackspace Failover Automation Update
DRBD & HA PowerPack

Prepared For:

Customer

Prepared By:

ScienceLogic Professional Services Team

Benjamin Leyland

August 16th 2024

Table of Contents

1 Revision History and Approvals 3
1.1 Revision History: 3
1.2 Approval: 3
2 Overview & Customer Needs 4
3 Business Value 4
4 Prerequisites and Assumptions 4
5 Previous Requirements 4
5.1 Original Run Book Action Type Requirements 5
6 New Requirements 7
6.1 New Run Book Action Type Requirements 7

1 Revision History and Approvals

This section lists the revisions made to this document, revision number and revised date, that will track changing information during the project's life cycle. The approval section will list the key stakeholders who have approved the document at a certain point. If several rounds of approval are required due to changing information, it will list several rows of approvals with different dates.

1.1 Revision History:

Revision Number	Revision Date	Revision Description
DRAFT Rev 1.0	08/16/2024	ScienceLogic drafted initial document version for all project owners review, input and agreement
Final Rev 2.0	[Date]	

1.2 Approval:

Company	Approver Name / Date Approved

2 Overview & Customer Needs

Rackspace ("Customer") requested assistance from the Professional Services team ("PS") at ScienceLogic, Inc. ("ScienceLogic") to provide a solution to automate the enablement/disablement of Dynamic Application collection when a cluster failover occurs.

The objective of this document is to provide the Customer with a list of PowerPack development specifications. The details in this document cover the work ScienceLogic will deliver during this part of the engagement with the Customer.

3 Business Value

Customer Owners:

Cassandra Peitersen

Ryan Speck

Business Value:

- Reduces manual effort in configuration after failover
- Improves accuracy of monitoring configuration

4 Prerequisites and Assumptions

The design of this PowerPack assumes the following:

- The PowerPack will be installed on SL1 version 12.1 or later
- The customer will be responsible for configuring automation actions and policies based on the provided action type and documentation
- The solution will be delivered as an update to the DRBD & HA PowerPack, which was delivered to Rackspace by the ScienceLogic Professional Services team as part of a previous project
    - The previous version of the PowerPack deployed in is version 1.2, the version delivered for this solution will be version 2.0
    - No changes will be made to the other content in the PowerPack

5 Previous Requirements

The solution will be based on the existing Failover Monitoring Settings run book action type. This section lists the original requirements for this run book action type, for reference.

5.1 Original Run Book Action Type Requirements

The Run Book Action type in this PowerPack must disable or enable SL1 system process monitors based on collected Dynamic Application data. For example, when the DRBD status object for a device is changes from "Secondary" to "Primary", monitoring of the mysqld process must be enabled.

Basic Properties

Name	Failover Monitoring Settings
Version	1.0
State	Enabled
Organization	System
Description	This action type updates SL1 monitoring settings (enable/disable) when failover occurs in DRBD & HA systems.
Execution Environment	DRBD & HA

Input Parameters

Name	Type	Purpose
config_object_id	string	The configuration object id (e.g. o_1) that collects the DRBD or HA status
enable_value	string	The value of the collection object that indicates the device is currently primary (e.g. "Primary", "all", "Yes", etc).
disable_value	string	The value of the collection object that indicates the device is currently primary (e.g. "Secondary", "none", "No", etc).
process_monitors	string	A comma-separated list of process monitors that need to be disabled or enabled when the DRBD or

		HA status changes. Can be an empty string (no process monitors are edited)
mount_point	string	Either the collection object id (e.g. o_1) that collects a mount point OR the actual mount point (e.g. /data) for which monitoring must be enabled or disabled when the DRBD or HA status changes. Can be an empty string (no file system monitoring settings are changed)

Output Parameters

Name	**Type**	**Purpose**
message	string	A string output indicating the result of the automation action

Snippet Logic

When a run book action of this type is executed, the following logic is performed:

- If the process_monitors and mount_point inputs are both empty strings, the snippet execution ends with a human-readable error message in the "message" output parameter.
- The config_object_id input and the device ID for which the action is running are used to determine the DRBD or HA status, i.e. the value of that collection object ID for that device. Note that this must be the last collected value; SL1 does not store collected configuration data for each poll if the value does not change.
    - If the collection object has not been collected for this device, the snippet execution ends with a human-readable error message in the "message" output parameter.
- The collected DRBD or HA status value is matched against the enable_value and disable_value strings:
    - If the enable_value is matched, the snippet will be enabling process monitors and/or file system monitoring.
    - If the disable_value is matched, the snippet will be disabling process monitors and/or file system monitoring.
    - If neither value is matched, the snippet execution ends with a human-readable error message in the "message" output parameter.
- If the snippet execution has not yet ended, an output string is initialized as an empty string. All output messages for the remaining execution are appended to this string
- If the process_monitors input is a non-empty string:
    - The process_monitors string is comma-separated, it is first split in to multiple strings
    - Each sub-string is matched against the process names of the System Process Monitoring policies aligned to the device.

        - If a process monitoring policy matches, the policy state is changed based on the enable_value or disable_value matching logic. The process name, policy ID, and new state value is appended to the output string.
        - If a process monitoring policy does not match, a human readable error is appended to the output string. This does not stop the snippet from processing additional processes or file systems.
- If the mount_point input is a non-empty string:
    - The file system mount point is determined:
        - If the mount_point input is in SL1 collection object format (i.e. begins with "o_"), the value of that collection object ID for that device is determined. Note that this must be the last collected value; SL1 does not store collected configuration data for each poll if the value does not change.
        - Otherwise, the value of the mount_point input is the mount point.
    - The mount point is matched against the file systems monitored for that device:
        - If a file system matches, the monitoring state is changed based on the enable_value or disable_value matching logic. The file system and new state value is appended to the output string.
        - If a file system does not match, a human readable error is appended to the output string. This does not stop the snippet from processing additional processes or file systems.
- The output string is returned in the "message" output parameter.

6 New Requirements

This section describes the new requirements that will be met by the updated PowerPack. Differences between the new requirements and the previous requirements are highlighted in bold.

These changes are not backwards-compatible with automation actions created using the previous version(s) of the run book action type. Therefore, the run book action type described in this section must be delivered as a new run book action type, i.e. a Save As operation must be performed on the existing run book action type prior to the necessary changes being made. The PowerPack will be delivered with both the existing and new versions of this run book action type.

6.1 New Run Book Action Type Requirements

The Run Book Action type in this PowerPack must disable or enable SL1 system process monitors, **Dynamic Application collections**, and file systems based on collected Dynamic Application data. For example, when the DRBD status object for a device is changes from "Secondary" to "Primary", monitoring of the mysqld process must be enabled.

Basic Properties

Name	Failover Monitoring Settings
Version	**2.0**

State	Enabled
Organization	System
Description	This action type updates SL1 monitoring settings (enable/disable) when failover occurs in DRBD & HA systems.
Execution Environment	DRBD & HA

Input Parameters

Name	Type	Purpose
config_object_id	string	The configuration object id (e.g. o_1) that collects the DRBD or HA status
enable_value	string	The value of the collection object that indicates the device is currently primary (e.g. “Primary”, “all”, “Yes”, etc).
disable_value	string	The value of the collection object that indicates the device is currently primary (e.g. “Secondary”, “none”, “No”, etc).
process_monitors	string	A comma-separated list of process monitors that need to be disabled or enabled when the DRBD or HA status changes. Can be an empty string (no process monitors are edited)
dynamic_apps	**string**	**A comma-separated list of Dynamic Application IDs that need to be disabled or enabled when the DRBD or HA status changes. Can be an empty string (no Dynamic Application collections are edited)**
edit_user_id	**id**	**A user account ID in the system. When Dynamic Applications are enabled/disabled, this user ID will be recorded as the edit user.**

mount_point	string	Either the collection object id (e.g. o_1) that collects a mount point OR the actual mount point (e.g. /data) for which monitoring must be enabled or disabled when the DRBD or HA status changes. Can be an empty string (no file system monitoring settings are changed)

Output Parameters

Name	Type	Purpose
message	string	A string output indicating the result of the automation action

Snippet Logic

When a run book action of this type is executed, the following logic is performed:

- If the process_monitors, **dynamic_apps**, and mount_point inputs are empty strings, the snippet execution ends with a human-readable error message in the "message" output parameter.
- The config_object_id input and the device ID for which the action is running are used to determine the DRBD or HA status, i.e. the value of that collection object ID for that device. Note that this must be the last collected value; SL1 does not store collected configuration data for each poll if the value does not change.
    - If the collection object has not been collected for this device, the snippet execution ends with a human-readable error message in the "message" output parameter.
- The collected DRBD or HA status value is matched against the enable_value and disable_value strings:
    - If the enable_value is matched, the snippet will be enabling process monitors, **Dynamic Application collections**, and/or file system monitoring.
    - If the disable_value is matched, the snippet will be disabling process monitors, **Dynamic Application collections**, and/or file system monitoring.
    - If neither value is matched, the snippet execution ends with a human-readable error message in the "message" output parameter.
- If the snippet execution has not yet ended, an output string is initialized as an empty string. All output messages for the remaining execution are appended to this string
- If the process_monitors input is a non-empty string:
    - The process_monitors string is comma-separated, it is first split in to multiple strings
    - Each sub-string is matched against the process names of the System Process Monitoring policies aligned to the device.

        - If a process monitoring policy matches, the policy state is changed based on the enable_value or disable_value matching logic. The process name, policy ID, and new state value is appended to the output string.
        - If a process monitoring policy does not match, a human readable error is appended to the output string. This does not stop the snippet from processing additional processes or file systems.
- **If the dynamic_apps input is a non-empty string:**
    - **The dynamic_apps string is comma-separated, it is first split in to multiple strings**
    - **Each Dynamic Application ID is matched against the Dynamic Applications aligned to the device.**
        - **If a Dynamic Application matches, the collection state of all collection objects is changed based on the enable_value or disable_value matching logic. The edit_user_id value is used as the editing user when changing the collection object state. The Dynamic Application ID, name, and new state value is appended to the output string.**
        - **If a Dynamic Application does not match, a human readable error is appended to the output string. This does not stop the snippet from processing additional processes or file systems.**
- If the mount_point input is a non-empty string:
    - The file system mount point is determined:
        - If the mount_point input is in SL1 collection object format (i.e. begins with "o_"), the value of that collection object ID for that device is determined. Note that this must be the last collected value; SL1 does not store collected configuration data for each poll if the value does not change.
        - Otherwise, the value of the mount_point input is the mount point.
    - The mount point is matched against the file systems monitored for that device:
        - If a file system matches, the monitoring state is changed based on the enable_value or disable_value matching logic. The file system and new state value is appended to the output string.
        - If a file system does not match, a human readable error is appended to the output string. This does not stop the snippet from processing additional processes or file systems.
- The output string is returned in the "message" output parameter.