ScienceLogic

Rackspace Failover Automation

The DRBD & HA PowerPack for Rackspace contains a run book action type that can be used to create automations that change the monitoring state of processes, file systems, and Dynamic Applications when a clustered device changes state (e.g. from Primary to Secondary or Secondary to Primary).

The automation actions created using this action type are intended to be triggered on the Dynamic Application configuration change event for a collection that indicates the cluster state (e.g. a collection object that collects "Primary" or "Secondary", "Active" or "Passive", etc). The automation then enables or disables one or more process monitors, file system monitors, and/or Dynamic Applications on the associated device.

Configuring an Automation Action

To configure an automation action using the action type in the PowerPack:

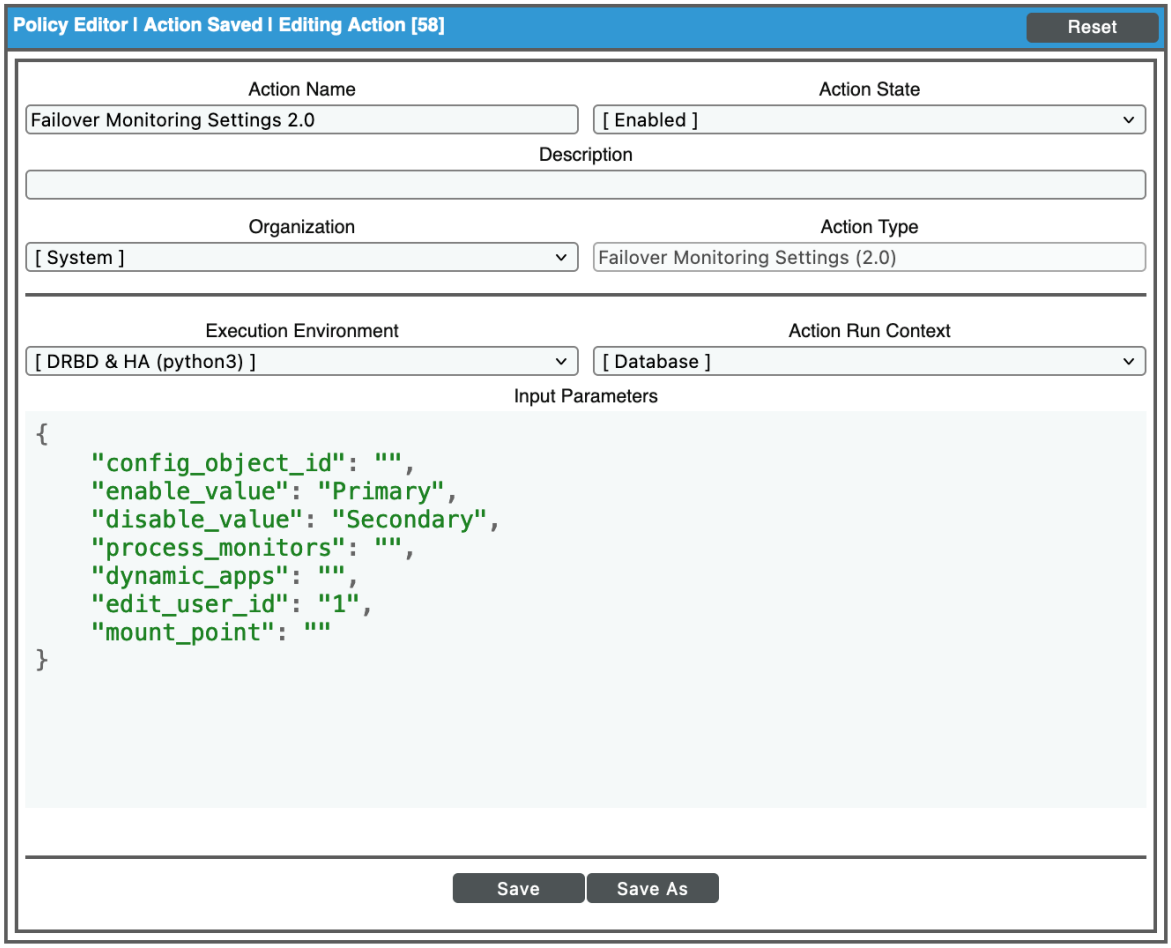
1. Go to Registry > Run Book > Actions.
2. Search for "Failover Monitoring Settings 2.0" and click its wrench icon.
3. Supply a new name in the "Action Name" field
4. In the Input Parameters field, update parameter values:
    a. config_object_id. The numeric ID of the Dynamic Application that collects the cluster state, i.e. the value that will determine whether the automation is enabling or disabling collection.
    b. enable_value. The collected value (e.g. "Primary", "Active", etc) that indicates that the device is a primary/active cluster node. If the current value of the collection object specified in the config_object_id field matches this value, the specified monitors will be enabled.
    c. disable_value. The collected value (e.g. "Secondary", "Passive", etc) that indicates that the device is a secondary/passive cluster node. If the current value of the collection object specified in the config_object_id field matches this value, the specified monitors will be disabled.
    d. process_monitors. A comma-separated list of process names. This list of process monitors will be enabled or disabled by the automation. This parameter can be set to an empty string to skip process monitoring enable/disablement.
    e. dynamic_apps. A comma-separated list of Dynamic Application IDs. The collection objects in these Dynamic Applications will be enabled or disabled by the automation. This parameter can be set to an empty string to skip Dynamic Application enable/disablement.

f. edit_user_id. The numeric ID number of an SL1 user account. When the automation changes the state of Dynamic Application collection objects, this user will be set as the editing user, i.e. the UI will show this user edited those collection object states.

g. mount_point. A comma-separated list of file system mount point names OR collection object IDs (beginning with o_). This list of file systems will be enabled (unhidden) or enabled (hidden) by the automation. If a collection object ID is specified in this list, the current value of that object will be used as the file system name. This parameter can be set to an empty string to skip file system monitoring enable/disablement.

Note: Although the process_monitors, dynamic_apps, and mount_point parameters can be an empty string, you must supply a non-empty value in at least one of the three parameters.

5. Click the "Save As" button



Output and Logging

The automation action result will include one of the following:

- A single message indicating that the whole automation failed, e.g. if empty or invalid parameters are supplied.
- A list of message for each monitored element (process, Dynamic Application, file system) that matched the configuration. These messages will indicate either a change was made or that no change was made (typically because the element already has that setting applied).

The debug log for this automation is /var/log/sl1/failover_monitoring_settings.log. The automation executes on the primary Database Server.