Kelverion Automation
Routing and Remediation for Azure Automation

Deployment Guide

Version 1.4

Email: info@kelverion.com
Web: www.kelverion.com

1. Table of Contents

2. Overview

Increasingly customers are looking to be more advanced and instead of just raising Incident Tickets in their Service Desk they want to route the tickets to the correct support group with the appropriate severity. Many customers would like to perform automated diagnostic and remediation, as part of their Azure Automation and OMS Automation offerings.

The Routing and Remediation Solution for Azure Automation takes the well-established patterns from the on premise SCOM connectors solution and moves the execution into the cloud with Azure Automation and Azure Hybrid workers to connect in to your on-premise SCOM environment. This allows you to leverage the flexible pricing and on-demand power of Azure Automation.

This Solution delivers the following inter-connection functions;

- Automatic Incident Ticket creation
- Automatic update of the SCOM Alert to record the Service Desk Ticket ID
- Monitoring of the target Service Desk in order to automatically resolve the original Operations Manager Alert once the incident is Resolved.
- Provides the ability to route tickets to correct support group
- Provides the ability to run diagnostics for known issues and updates the Incident Ticket with the results
- Provides the ability to automatically trigger remediation for known issues

3. General Configuration Steps

3.1. Pre-Installation Information

The Routing and Remediation Package contains the following elements:

- Persistent Data Store (PDS) SQL configuration script
- PDS definition spreadsheet
- 10 Azure Automation Runbooks
- Solution Test Script

Kelverion Items Required

The solution requires the following Kelverion products:

- Kelverion Runbook Studio
- Kelverion Integration Module for SQL Server
- Kelverion Integration Module for ServiceNow

If you do not already have Kelverion Integration Modules, or the Kelverion Runbooks Studio they can be downloaded for evaluation from our website.

System Center Products Required

The one of the two following System Center PowerShell modules are required:

- System Center 2012 R2 Operations Manager
- System Center 2016 Operations Manager

3.2. Persistent Data Store

The Persistent Data Store or PDS is a SQL Server database that is used by these runbooks to allow all of the actions that the runbooks take to be carried out in a robust way. The use of the database at each “step” allows us to design the runbooks such that each runbook is simple and can be considered a discrete unit. In programming terms, it allows the runbooks to be modular.

To best exploit the power and flexibility of Azure, the PDS should be deployed to a SQL instance within your Azure subscription.

We will be using a PDS on Azure using Azure's SQL offerings, rather than building a VM and installing SQL. This allows us to deploy the SQL instance and PDS database quickly and with the minimum of maintenance.

1. Create a new Azure SQL Server. Create the SQL Server in a New Resource Group "Automation", ensure that the "Allow azure services to access server" check box is ticked. This means that ALL Azure resources will be allowed to access the SQL Server through the firewall
2. Give your desktop access to the SQL Server through the firewall
3. Create a new Database "AutomationData". It's important to use this name for the PDS, as it is held within the runbooks. The **BASIC** tier is ample performance for testing and evaluation of the solution. As it is trivial to scale up or down the databases this should also be the starting point for your deployments unless you know that there is going to be a high volume of alerts right from the outset.
4. Connect to the database using SQL Management Studio
5. Execute the SQL script (*Create Azure Automation - SCOM Routing db objects.sql*) from the package to Create the database tables, views and stored procs.
6. Execute the second SQL Script (*Create Azure Automation - SCOM Routing - Example Enrichment Rules.sql*) to create a set of example enrichment rules within the PDS

3.3. Create an Azure user for starting runbooks

In Azure AD create a "local" Azure AD User.
You will need to login to the portal using the user account to set the password, as the password change flag is set when the account is created.

The account should be set as a "Contributor" for the Automation account (Access Control (IAM) blade under the automation account). The username and password for this account will need to be configured within your automation account so make a note of these values.

3.4. Hybrid worker

The solution integrates with SCOM using the native SCOM cmdlets. These commands need to be able to "reach in" to the SCOM environment. This is achieved by using a Hybrid worker.
The hybrid worker is configured as part of OMS, and must be added to the Automation account. It should be set to run using a domain "Run as" account, that has rights to connect to the SCOM console.
Visit https://docs.microsoft.com/en-us/azure/automation/automation-hybrid-runbook-worker

NOTE: you must add the Public IP Address of the hybrid worker to the SQL Firewall access rules.

3.5. Load Kelverion Integration Modules

The Integration modules will need to be installed in the following locations

- The Automation Account ("Assets" > "Modules")
- The machine where you are running the Runbook Studio.
- The Hybrid worker.

The best practice for managing PowerShell modules is to create a new easy to find directory (for example c:\Modules) and extract all of the modules you require into that location. Once you have done so, you should update the environmental variable PSModulePath to include the new location. This will ensure that the modules can be found by any commands that need them.

Visit https://msdn.microsoft.com/en-us/library/dd878326(v=vs.85).aspx for more information on the PSModulePath.

Visit https://docs.microsoft.com/en-us/azure/automation/automation-update-azure-modules for more information about loading Integration modules into your Automation Account.

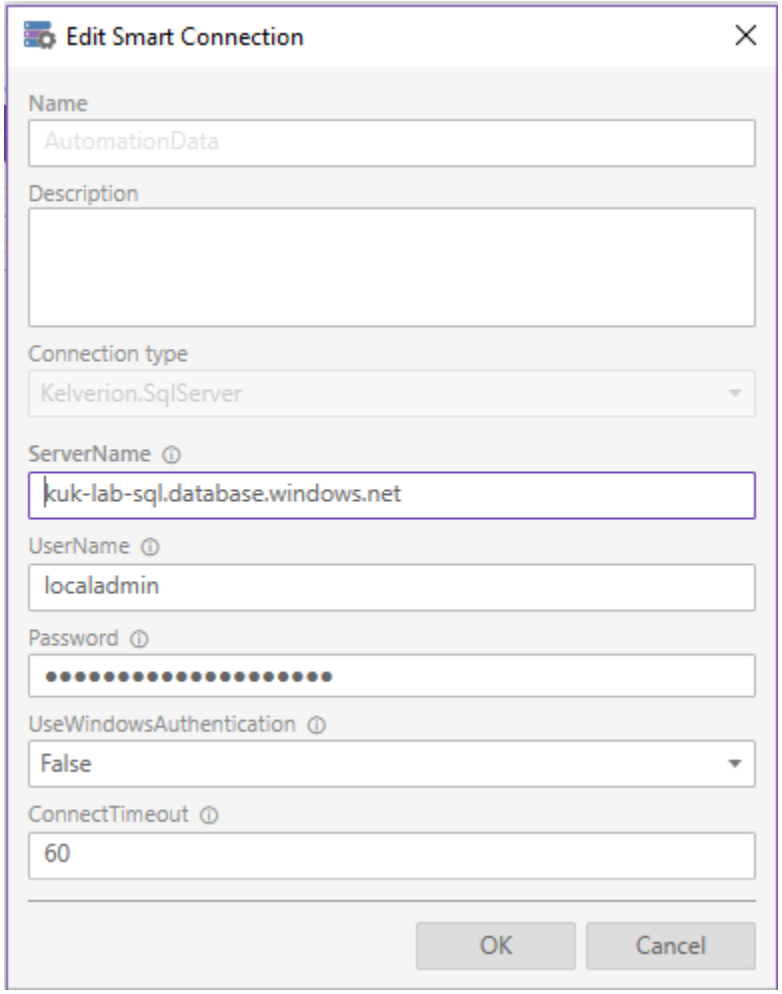
3.6. Configure your Automation account using the Runbook Studio

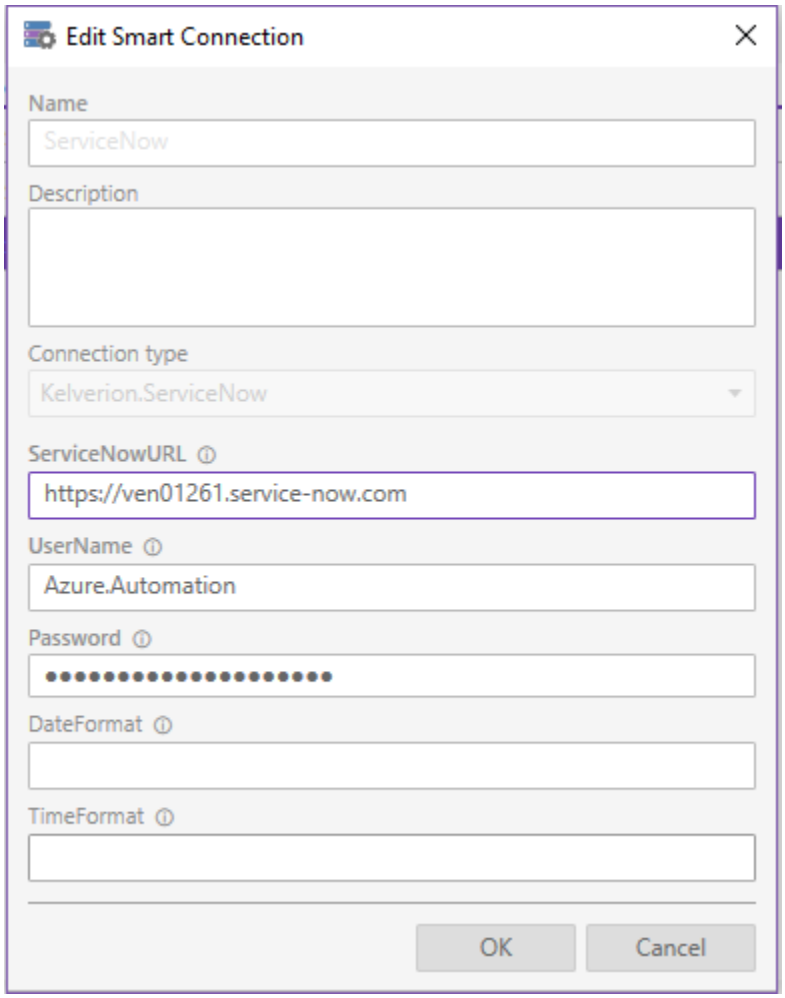
Connect the Runbook studio to your target Azure subscription. You can find more information on the initial configuration of the runbook studio on pages 5 and 6 of the User's Guide.

3.6.1. Design Time (Smart Connections)

The runbook studio needs to have connections configured for use at design time, these smart connections are used by the discovery process within the Runbook Studio to allow the Runbook Studio to discover information about your target systems. This discovery process accelerates the process of runbook development.

Create the following smart connections within the Runbook Studio.

AutomationData	

ServiceNow	

3.6.2. Azure Variables

Azure variable allow us to remove static configuration information from the code in our runbooks and store the information in an easy to access place. This helps us to build consistent configuration between all of our runbooks. These values are accessed at design time, and at runtime by both the Hybrid workers, and the Azure Worker sandboxes.

Create the following variables in the Automation Account using the runbook Studio

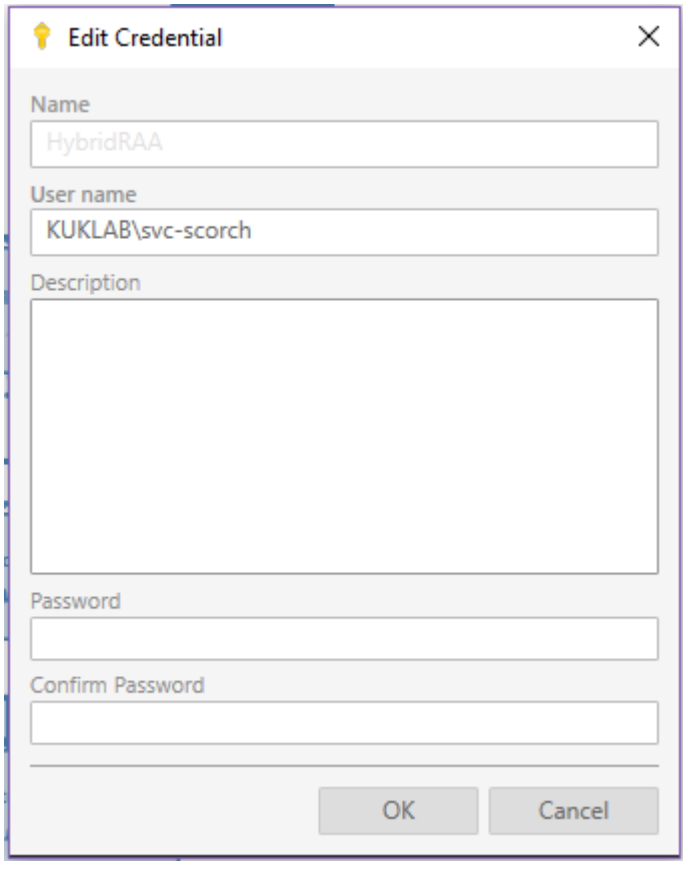
DignosticAutomationAccount	This is the account that contains the diagnostic runbooks. The Automation account name is used by the runbook *"New - Monitor-PDSActionQueueReady*" to find and start the runbooks specified in the EER_ACTION_MAP

SCOM Server Name	This is the name of the SCOM Management server that the runbooks will use when they get or update SCOM alerts.
DignosticResourceGroup	This is the resource group that contains the diagnostic runbooks. The Automation account name is used by the runbook *"New - Monitor-PDSActionQueueReady"* to find and start the runbooks specified in the EER_ACTION_MAP
DiagnosticHybridWorkerGroup	This is the Hybrid Worker Name

3.6.3. Azure Credentials

Azure Credential assets allow us to create and manage credential objects that can be utilised throughout our runbooks. These credentials are accessed by our runbooks at runtime.

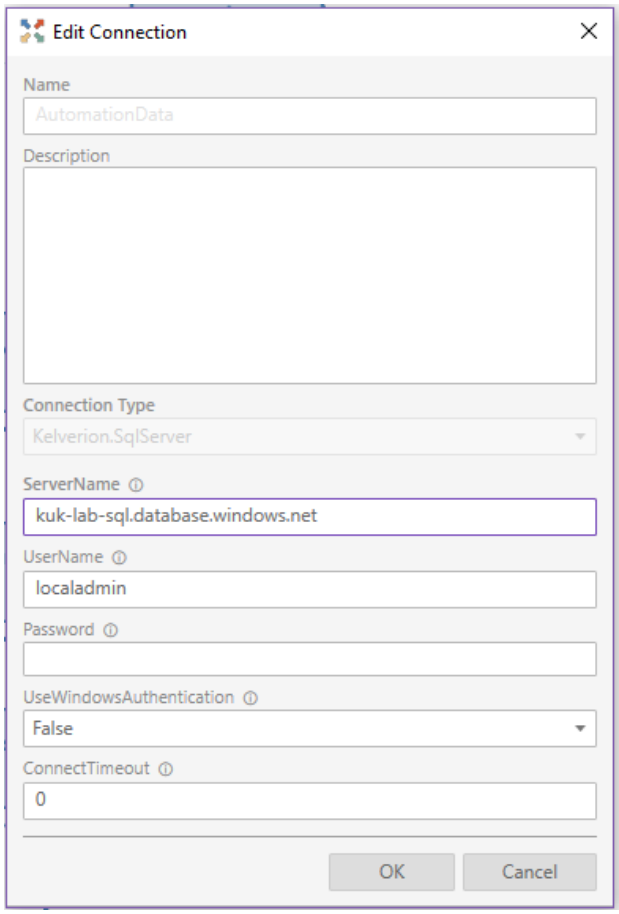
Create the following Credential Assets using the Runbook Studio

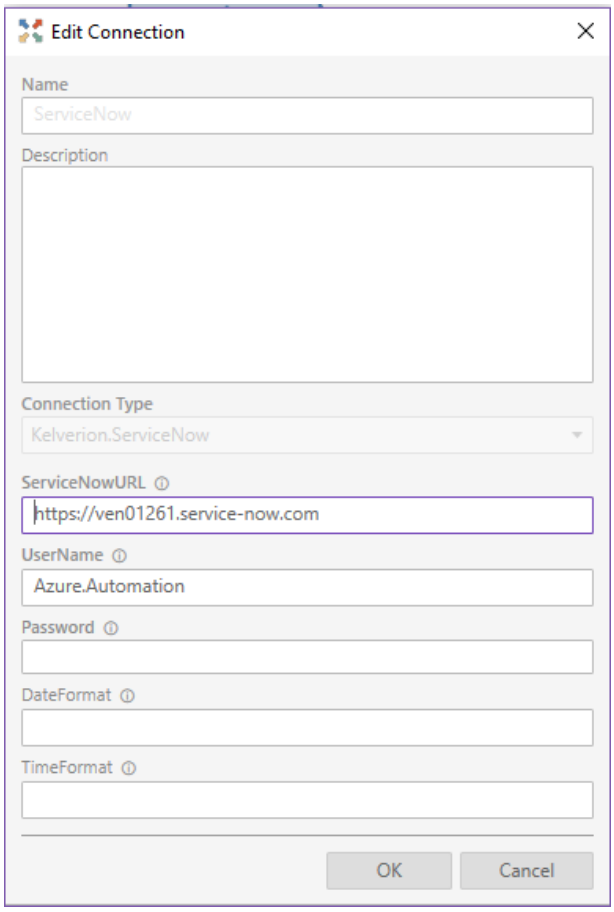
HybridRAA	The HybridRAA (Hybrid worker run as account) is used in the configuration of the hybrid worker, it is not referenced directly by the runbooks. This is the account that the hybrid worker will execute each of the runbooks as. The account will require access to the Operations Manager Console so it can search for and update SCOM events.	

3.6.4. Azure Runtime connections

Azure connection assets are used at runtime to define a reusable connection configuration. The connection types available are dependent upon module that have been loaded into your Automation Account. If you cannot see the connection types listed below when you attempt to create the connection assets, then this indicates an issue with the modules that are loaded into your automation account. Please verify that all of the required modules are loaded.

Create the following Connection in Azure using the Runbook Studio

AutomationData	The " AutomationData " connection asset in Azure defines the connection that will be used at runtime for the runbooks to connect to the pds database.	

ServiceNow	The "ServiceNow" connection asset in Azure defines the connection that will be used at runtime for the runbooks to connect to the ServiceNow Instance. The license server URL show below Must only be used for internal implementations. Any customer implementations must have their own management server for controlling Integration Module licenses	

3.7. Import Runbooks using the Runbook Studio

Connect to the Automation Account using the runbook studio.

Check that you have the correct Automation Account set as the default target (if there is more than one Automation account associated with the subscription)

Open each of the following runbooks, then "publish draft" and then "publish" the runbooks.

Kelverion_OMSN_00-1_Worker_Diagnostic1	Sample diagnostic runbook which is launched based upon the diagnostic rules held with the EER_Action_Map and the SCOM_EER (Event Enrichment Rules) tables.	Execute on Hybrid worker or Azure. It's likely that the processing should take

	This runbook can be imported several times using the Runbook Studio (providing you update the Runbook Name. You should also update the "self" activity to reflect the updated runbook name	place on the hybrid worker to allow access to the source of the event, however this is a sample diagnostic.
Kelverion_OMSN_10-1_Monitor_OMForNewAlert	The runbook will search for all events where the ModifiedTime = CreatedTime, and the ModifiedTime is after the time of the last New event that was detected. The Operations Manager PowerShell uses a "long time" format that includes thousandths of a second. The PDS also uses the "long time" format, but it is necessary to use the CONVERT function in SQL to ensure that the full-time resolution is used when querying the database.	This runbook must be executed on the Hybrid worker as it uses the Operations Manager PowerShell Module to query SCOM.
Kelverion_OMSN_10-2_Monitor_CreateIncident	This runbook monitors the PDS SCOM_EVENTS table for rows which indicate that a NEW scom event has been created. The runbook will then execute the stored procedure "sp_ProcessEnrichmentData" which will return the desired severity and resolver group for the incident, using the same logic as the view used in the Orchestrator version of the solution. The stored PROC will create any entries in the EER_ACTION_QUEUE if they are required. The runbook will then create a new incident within ServiceNow, and update the PDS with the SysID and Ticket Number.	This runbook can be executed either in Azure, or on the Hybrid Worker.
Kelverion_OMSN_10-3_Monitor_UpdateAlert	This runbook monitors the PDS SCOM_EVENTS table for rows that	This runbook must be executed on

	indicate an incident has been raised for a New SCOM event. The runbook will then update the SCOM event with the Ticket ID from the service desk	the Hybrid worker, as it uses the Operations Manager PowerShell Module to query SCOM.
Kelverion_OMSN_20-1_Monitor_OMForClosedAlert	The runbook will use a search criteria to specify all closed events that have been updated since the last time closed events were found, and NOT closed by the user account that is used to connect to SCOM. This ensures that the runbooks do not create a processing loop where events get stuck within the solution.	This runbook must be executed on the Hybrid worker as it uses the Operations Manager PowerShell Module to query SCOM.
Kelverion_OMSN_20-2_Monitor_CloseIncident	This runbook monitors the PDS SCOM_EVENTS table for rows that indicate a SCOM Event has been resolved. The runbook will then update the incident and set its state to resolved.	This runbook can be executed either in Azure, or on the Hybrid Worker.
Kelverion_OMSN_30-1_Monitor-ClosedIncidents	This runbook monitors ServiceNow for any incidents that have been closed (resolved) since it last detected a closed event. The criteria used for the search exclude incidents that have been closed by the user account that is used for the solutions connection to Service Now. When a matching incident is detected the SCOM_EVENTS table is updated.	This runbook can be executed either in Azure, or on the Hybrid Worker.
Kelverion_OMSN_30-2_Monitor_CloseAlert	This runbook monitors the PDS SCOM_EVENTS table for rows indicating that events, which previously have had an incident raised, have now been resolved in the service desk. The runbook updates the SCOM event to mark it as closed	This runbook must be executed on the Hybrid worker as it uses the Operations Manager

		PowerShell Module to query SCOM.
Kelverion_OMSN_40-1_Monitor_PDSForActionReady	This runbook can be executed either in Azure, or on the Hybrid Worker. This runbook monitors the PDS table EER_ACTION_QUEUE for entries that signify that a diagnostic or remedial action should be executed. The entries are written to the EER_ACTION_QUEUE by the sql Stored procedure "sp_ProcessEnrichmentData" during the processing of new SCOM events. This is a change from the Orchestrator solution, that is largely driven by the fact that the SQL integration module does not currently support selecting from SQL Views.	This runbook can be executed either in Azure, or on the Hybrid Worker.
Kelverion_OMSN_40-2_Monitor_PDSForActionComplete	This runbook monitors the PDS table EER_ACTION_QUEUE for entries that signify the diagnostic (or remediation) action has completed. When an appropriate row is detected the data will be retrieved from the UNIFIED_DIGNOSTIC table and written to the Service Now incident.	This runbook can be executed either in Azure, or on the Hybrid Worker.

3.8. Testing

Each of the runbooks can be tested using the PowerShell script "**test Azure Routing and remediation.ps1"** which should be in the package directory
Open the script in the ISE and update the password on line 14, then execute the script and follow the prompts on screen.

You could easily achieve the same results by running the monitoring runbook via the Azure Portal, starting the runbooks from the script using PowerShell is much easier.

3.9. Scheduling runbook execution.

Once all the runbooks (and other assets) are in place and the runbooks have been tested you will need to schedule the runbooks for repeated execution.

Azure Automation has a schedule facility for runbooks. However, the minimum repeat interval with the runbook schedules is 1 hour. As we need much finer granularity for our solution, we recommend using Azure Logic Apps with to trigger the execution on a recurring interval.

Using an execution interval or 1 minute means that the solution will consume a large number of Automation minutes each month, however it gives the most responsive configuration. The runbooks are written in such a way that, if the customer prefers a low Azure Resource utilisation, they can be executed with a lower frequency (obviously with a trade off in responsiveness)

1 Lea Business Park,
Lower Luton Road,
Harpenden
AL5 5EQ
Email: info@kelverion.com
Web: www.kelverion.com

