

# Kelverion

## Operations Manager 2022 Routing and Remediation

### User's Guide

Version 2.0

## 1. Table of Contents

|   |                                     |
|---|-------------------------------------|
| <b>2. Upgrade Warning .....</b>   | <b>3</b>                            |
| <b>3. Overview.....</b>   | <b>3</b>                            |
| <b>4. Runbook Design Overview .....</b>   | <b>5</b>                            |
| 4.1. SCOM to a Service Desk Operation .....   | 5                                   |
| <b>5. General Configuration Steps .....</b>   | <b>7</b>                            |
| 5.1. Pre-Installation Information .....   | 7                                   |
| Persistent Data Store.....  | 8                                   |
| 5.1.1. Location.....  | 8                                   |
| 5.1.2. Database version.....  | 8                                   |
| 5.1.3. Collation.....   | 8                                   |
| 5.1.4. Sizing .....   | 8                                   |
| 5.1.5. Other options .....  | 9                                   |
| 5.2. PDS Creation .....   | 9                                   |
| 5.3. PDS Creation Steps .....   | 10                                  |
| 5.4. Solution Installation.....   | 10                                  |
| 5.5. SQL Server Integration Pack Configuration .....                                      | 10                                  |
| 5.6. Operations Manager Integration Pack Configuration.....                               | 10                                  |
| 5.7. Runbook Management Integration Pack Configuration .....                              | 11                                  |
| <b>6. Service Desk Event Enrichment .....</b>   | <b>12</b>                           |
| 6.1. Service Desk Runbook Configuration.....  | 14                                  |
| <b>7. Optional Diagnostics and Remediation Extension.....</b>                             | <b>17</b>                           |
| 7.1. Structure of a typical diagnostic runbook .....                                      | 18                                  |
| <b>8. Runbook Performance and Scalability.....</b>  | <b>19</b>                           |
| 8.1. Testing Environment.....   | 19                                  |
| 8.2. Results .....  | 21                                  |
| <b>9. Error Reporting.....</b>  | <b>22</b>                           |
| <b>10. Installing Temporary License of Kelverion Integration Pack for SQL Server.....</b> | <b>22</b>                           |
| <b>11. Notes.....</b>   | <b>Error! Bookmark not defined.</b> |

## 2. Upgrade Warning

This Runbook Solution is provided for installation in a clean Orchestrator environment. If you have deployed an earlier version of this Runbook Solution, then installing this version will overwrite any changes you have made to the currently deployed Runbooks.

You can either delete you existing Runbook deployment and then install this new Runbook Solution set or upgrade your existing deployment.

If you would like to upgrade your existing deployment of this Runbook Solution to take advantage of the updates in this release please contact [info@kelverion.com](mailto:info@kelverion.com) and we will be happy to discuss what specific updates you will need to make to your existing deployment.

## 3. Overview

The Operations Manager Routing and Remediation solution for Orchestrator has been widely deployed since 2012 and meets the needs of customers who wanted to forward Alerts from System Center Operations Manager into other event management or Service Desk products in a bi-directional manner. It also provided automatic routing of Incident Tickets based on SCOM Alert parameters and then the execution of automatic diagnostic and remediation Runbooks to update and ultimately resolve the Incident Ticket without any human interaction.

Kelverion have developed a set of Orchestrator Runbook Solutions that integrate Operations Manager 2022 to a number of Service Desks.

These Solution deliver a number of inter-connection functions;

- Event forwarding
- Automatic Incident Ticket creation
- Automatic update of the SCOM Event to record the Service Desk Ticket ID
- Monitoring of the target EM tool or Service Desk in order to automatically resolve the original Operations Manager Event once the Event is Resolved on the target system.
- 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

The Solution leverage the Kelverion Persistent Data Store design philosophy and SQL Server Integration Pack to provide a scalable and robust solution.

The Operations Manager 2022 Routing and Remediation supports the following Target systems;

**Service Desks**

BMC Helix ITSM

ServiceNow

Atlassian Jira

The Solution is provided ready built just requiring configuration to implement in your environment.

The Solution is available as a self-installation package or for customers who are less familiar with Orchestrator as a complete installation and configuration package, where we work with you to setup the connectors in your environment.

This document provides the guidance on how to setup and configure these Runbooks in your environment.

Detailed output, produced using the Kelverion Runbook Surveyor product, has also been provided to show you how each Runbook and Activity has been configured should you need to reset a Runbook or Activity back to its delivered state after following any modifications you make to the Runbooks.

## 4. Runbook Design Overview

The Runbooks are based around our experience of building these types of solutions for many customers. The Runbook are designed as a solution which will meet 80% of the customer requirements out of the box but may need extension and modification to tailor the solution to exact customer requirements.

### 4.1. SCOM to a Service Desk Operation

The Runbooks perform the following tasks:

1. Monitor SCOM for New Events and record them in the PDS.
2. Monitor the PDS for New Events, pick those up and then Create an Incident Ticket
3. Record the Incident Ticket ID in the PDS and pass this back to the SCOM updating the Alert
4. Monitor the Service Desk for Resolved Incident Tickets and record in the PDS
5. Monitor the PDS for Resolved Incident Tickets and then go to SCOM and mark the appropriate Alert as Closed.
6. Monitor SCOM for Alerts marked as Closed and check the PDS to see if the Incident Ticket has been Resolved. If not record the Closed Alert in the PDS.
7. Monitor the PDS for Closed Alerts, pick those up and then mark the appropriate Incident Ticket as Resolved.
8. Under very high SCOM load conditions occasionally Orchestrator does not capture every new Alert at creation. There is therefore a Runbook which periodically gets all Alerts created since the last time it ran and checks the PDS to see if each Alert was processed correctly at creation time. If the Alert is not in the PDS it was missed and so the Runbook adds it to the PDS as a new Alert to be processed like any other.

Every implementation will want to filter for only those SCOM Alerts which they actually want to be forwarded to a Service Desk. The criteria for Alert forwarding is specific to each customer therefore it will require customisation for each implementation.

Without being able to predetermine how customers will ultimately want the forwarding to work, the Runbooks initially forward all SCOM Alerts to a Service Desk. This is ideal for Proof of Concept and evaluation purposes where the emphasis is on proving the capability of SCOM and Orchestrator rather than setting up complex SCOM Alert categorisations.

Also, you may want to create Enriched Service Desk Incident Tickets, for example automatically assigning the Ticket to the correct Resolution Group.

By using the Persistent Data Store approach any complex decision logic can be handled using the database and Orchestrator Runbooks. You can extend the

Runbooks yourself or we can provide consultancy to help you with the design update and Runbook modifications.

## 5. General Configuration Steps

### 5.1. Pre-Installation Information

#### Operations Manager 2022 Routing and Remediation Package Contents

Operations Manager 2022 Routing and Remediation Package contains the following elements:

- Persistent Data Store database creation script (Microsoft SQL Server DB)
- SCOM Routing and Remediation Runbook Set
- Operations Manager 2022 Routing and Remediation Users Guide

#### Integration Packs Required

The solutions requires the following Integration Packs:

##### **Microsoft**

- System Center 2022 Operations Manager Integration Pack

##### **Kelverion**

- SQL Server Integration Pack
- Runbook Management KIP

Plus one of the following:

- ServiceNow Integration Pack
- BMC Helix ITSM Integration Pack
- Atlassian Jira Integration Pack

Before importing any Runbooks please insure these Integration Packs are installed in Orchestrator. If you do not already have Kelverion Integration Packs they can be downloaded for evaluation from our website.

#### System Center Products Required

The following System Center products are required:

- System Center 2022 Operations Manager

## 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.

In your environment there may be a number of constraints that control the creation of a new database. For example, the location of the log and data files, the recovery options that should be used, and the collation of the server. These requirements are typically specified by the DBA responsible for your database server. These options do not affect the runbooks so please use the appropriate options for your environment.

### 5.1.1. Location

Typically, we create the PDS on the same database instance as is used for the Orchestrator database. There is no specific requirement that this must be the case. In environments where there is very high load you may find that creating the PDS on a different database instance advantageous.

### 5.1.2. Database version

The connector runbooks have been tested against SQL 2019 with the latest patches / updates applied.

### 5.1.3. Collation

The runbooks have all been developed on systems using **Case Insensitive** collations, the specific collation setting used for your environment must be case insensitive other than that though the setting can be chosen as appropriate for your environment.

### 5.1.4. Sizing

With every environment where the connectors have been implemented the sizing requirements have been different. The amount of space required will depend on the two following factors:

- Number of alerts
- Housekeeping frequency

Based on previous implementations of the connectors in varying environments we recommend that you allocate 1Mb of space for every 10 events that you expect to hold in your PDS. Remember that the events in the PDS are transitory so we also recommend that you perform housekeeping on these events to ensure that any events that have been processed are deleted or archived.

**Example:**

Our example environment receives 10 events per minute (or 14400) events per day, and our housekeeping removes events that have been closed after 7 days. Therefore, after 7 days there will typically be c.100,000 events in the PDS.

As a minimum for this environment, we would allocate 10Gb of storage to the data files.

We like to allow a margin of error in our calculations in case of periods of heavy load, so we actually allocate 12Gb of space (or an extra 20%.)

For the transaction logs we allocate 25% of the space allocated to the data files, so for this environment we allocate a further 3Gb.

We do not allow database auto-growth in this environment, and we use full logging. Again these choices will probably be made by your DBA.

#### **5.1.5. Other options**

For the remaining database options we use the defaults specified by SQL Server.

## **5.2. PDS Creation**

Each Runbook Solution uses a set of common tables within the PDS Database and a set of tables specific to itself.

As part of each solution package you are provided with a SQL script which will generate the PDS database tables required for the solution. When the SQL scripts are executed they check for the existence of each table they required in the PDS database. If this is a new installation they will create both the Common Tables and their Solution Specific database tables. If you are already using a Runbook Solution then the script will detect that some of the tables this solution requires already exist and the script skips these table creation steps and creates only the tables which do not exist in your installation.

This means you can easily deploy one or more Kelverion Runbook Solutions to an existing PDS database without damaging and tables which already exist. This also means when installing multiple Runbook Solutions in a new installation of the PDS you can run each Solution PDS creation script in order you like and know that when done you will have all the tables you need for all your Runbook Solutions to operate.

### 5.3. PDS Creation Steps

1. Create a New Database on your SQL Server called PDS\_Live or connect to your existing PDS\_LIVE database
2. Then execute the SQL Script provided within the PDS\_Live database you created.
3. Once the PDS\_Live database is created you must ensure the Orchestrator Runbook Server Service Account has as a minimum Read and Write Access permissions to the PDS\_Live database.

### 5.4. Solution Installation

The installation steps are as follows:

1. Install the Kelverion SQL Server Integration Pack, Microsoft System Center 2022 Operations Manager Integration Pack and the appropriate Service Desk Integration Packs into Orchestrator.
2. Install the Microsoft System Center 2022 Operations Manager Console onto the Runbook Server and machine that the Orchestrator Runbook Designer is installed on.
3. Import the SCOM Connect Runbook Set you wish to use.

### 5.5. SQL Server Integration Pack Configuration

In the Options Menu of Runbook Designer find the KA SQL Server settings. In here you will see a default database connection (PDS\_LIVE), which is required for the IP operation against the PDS.

Leaving the default Database Name (PDS\_LIVE), modify the default database connection by changing the 'Server Name' and authentication details to values appropriate to your environment.

In the Authentication scheme field, select how the connection request will be authenticated.

If you selected *SQL Server Authentication*, type the SQL user credentials into the User name and User password fields.

### 5.6. Operations Manager Integration Pack Configuration

In the Options Menu of Runbook Designer find the Microsoft Operations Manager Integration Pack settings. In here you will see a default Operations Manager connection (SCOM), which is required for the IP operation against Operations Manager.

You should update that configuration with the appropriate values for your environment and ensure that you can successfully “test connection” before you proceed.

### **5.7. Runbook Management Integration Pack Configuration**

In the Options Menu of Runbook Designer find the KA Runbook Management settings. In here you will see a default connection (RM), which is required for the IP operation against the Orchestrator Web Service.

Leaving the default (RM), modify the connection to values appropriate to your environment.

## 6. Service Desk Event Enrichment

The Service Desk event enrichment allows you to route alerts to a specific assignment team with the desired severity based on the event parameters.

The enrichment process is driven by a set of SQL “rules” that allow all the processing work to be offloaded on to the Database engine and allow fine control of the alert enrichment without impact upon the runbook design.

The solution is configured with a single default rule which will assign all Incident Tickets to a single support group you define with a given Severity and Priority.

We also provide some sample rules which are sufficient for a proof of concept but will require analysis of your events and assignment requirements before being implemented into a production environment.

The integration into the solution is straightforward thanks to the ability of the SQL integration pack to retrieve data from a SQL View. If you have examined the PDS you will see that it contains a table called SCOM\_EVENTS. This table holds the SCOM events that have been detected by orchestrator and metadata (processing state) for those events. The solution also contains a VIEW v\_SCOM\_EVENTS. This view contains the enriched event data, including Priority, Assignment group, and the Enrichment rule id.

The view itself is derived from a complex join between two tables the SCOM\_EVENTS table and the SCOM\_EER table.

The configuration rules use SQL Like expressions to match events using the following fields in the SCOM\_EVENTS table:

- Name
- Description
- MonitoringObjectFullName
- PrincipalName
- Parameters

If required the SCOM\_EER table and the definition of the rules can easily be modified to change the fields that are used, however this configuration has served well in our previous implementations.

To allow greater flexibility in the definition of the rules each rule has a Priority which is used to resolve conflicts between the different rules, with the highest priority rule “winning”.

For example, if you consider the two highlighted rules in the SCOM\_EER table below:

| SD_Priority | SD_Assignee        | EER_Priority | EER_Description                    | EER_Action | Name | Description   | MonitoringObjectFullName                     | PrincipalName           | Parameters |
|-------------|--------------------|--------------|------------------------------------|------------|------|---|--|-------------------------|------------|
| 3 - Low     | Service Desk       | 1            | Default Assignment rule            | 0          | %    | %   | %  | %                       | %          |
| 3 - Low     | Database           | 2            | NULL                               | 1          | %    | %   | Microsoft.SqlServer.DBEngine:%               | %                       | %          |
| 3 - Low     | ITSM Engineering   | 4            | Health Service Errors              | 1          | %    | %   | Microsoft.SystemCenter.HealthService:%       | %                       | %          |
| 1 - High    | Database           | 5            | NULL                               | 1          | %    | %   | %  | SQL.lab.kelverion.local | %          |
| 2 - Medium  | Database San Diego | 10           | Database Errors for San Diego Team | 1          | %    | %   | Microsoft.SqlServer.DBEngine:%;PRODUCTION    | %                       | %          |
| 3 - Low     | Database Atlanta   | 10           | Database Errors for Atlanta Team   | 1          | %    | %   | Microsoft.SqlServer.DBEngine:%;PREPRODUCTION | %                       | %          |
| 3 - Low     | Service Desk       | 10           | Email alert failures               | 0          | %    | Event Tab error Unable to send email message after 3 attempts thrown on SCO | Microsoft.SystemCenter.HealthService:%       | %                       | %          |

You can see that the first rule will match all of the events that the second rule would match, however because it is a less specific rule it has been given a lower priority.

You can also see that the first rule in the table (with the description Default assignment rule) has the most generic search criteria and therefore the lowest priority. The effect is that this rule will only be used when all of the other rules have been exhausted.

The rules can be created and amended by using the SQL management studio.

## 6.1. Service Desk Runbook Configuration

To use the SCOM to Service Desk Connectors you will need to configure the Runbooks which are appropriate to the Service Desk you wish to use: BMC Helix ITSM and Service-Now (Default).

Runbooks have been supplied for each Service Desk and configuration is a three stage process. You should check carefully that the default configuration meets your requirements.

1. Modify Runbook 15.1\_Forward\_New\_SCOM\_Alerts - change the Invoke Runbook Activity to invoke the Service Desk Runbook you want 15.1.1, 15.1.2 or 15.1.3.  
*Ensure that the ID parameter is passed to the runbook that gets Invoked.*

Raise Service Desk Incident Properties

**Details Information**  
Define the Runbook and Initial Data for the Invoke Runbook Activity.

General

Runbook: /ice\_Now\_Record/15.1.2\_Create\_Service\_Now\_Record ...

☐ Invoke by path ☐ Wait for completion

Parameters:

ID {Id from Monitor PDS New SCOM}

Runbook Servers:

Finish Cancel Help

Modify Runbook 15.2\_Forward\_Closed\_SCOM\_Alerts – change the Invoke Runbook Activity to invoke the Service Desk Runbook you want 15.2.1, 15.2.2 or 15.2.3.

*Ensure that the ID parameter is passed to the runbook that gets Invoked.*

Forward Closed SCOM Alerts Properties

**Details Information**  
Define the Runbook and Initial Data for the Invoke Runbook Activity.

General

Runbook: /vice\_Now\_Record/15.2.2\_Close\_Service\_Now\_Record ...

☐ Invoke by path ☐ Wait for completion

Parameters:

ID {Id from Monitor PDS Closed SCOM}

Runbook Servers:

Finish Cancel Help

2. Configure the Create Incident Ticket Activity in the Service Desk Runbook you are going to use 15.1.1, 15.1.2 or 15.1.3 and map the fields from SCOM to the Incident Ticket Fields.

By using the Persistent Data Store approach any complex field mappings can be

handled using the database and Orchestrator Runbooks. You can extend the Runbooks yourself or we can provide consultancy to help you with the design update and Runbook modifications.

The screenshot shows the 'Insert Record Properties' dialog box with the 'Properties' tab selected. The 'Configuration' section shows 'Name: ServiceNow'. The 'Properties' table lists the following fields and values:

| Table Name          | Value                            |
|---------------------|----------------------------------|
| Incident [incident] |                                  |
| Company             | ACME                             |
| Caller              | Orchestrator                     |
| Severity            | 2 - Medium                       |
| Urgency             | {SD_Priority from "Select Rows"} |
| Work notes          | Computer: {NetbiosComputerName f |
| Assignment group    | {SD_Assignee from "Select Rows"} |

Buttons at the bottom: Finish, Cancel, Help.

Sample ServiceNow Incident Creation

Configure the Close Incident Ticket Activity in the Service Desk Runbook you are going to use 15.2.1, 15.2.2 or 15.2.3.

The screenshot shows the 'Update Record Properties' dialog box with the 'Properties' tab selected. The 'Configuration' section shows 'Name: ServiceNow'. The 'Properties' table lists the following fields and values:

| Table Name          | Value                               |
|---------------------|-------------------------------------|
| Incident [incident] |                                     |
| Sys ID              | {_ServiceDesk_ID from "Select Rows" |
| Work notes          | Source SCOM event has been resolv   |
| Close code          | Closed/Resolved by Caller           |
| Close notes         | Source SCOM event has been resolv   |
| State               | Closed                              |
| Additional comments | Source SCOM event has been resolv   |

Buttons at the bottom: Finish, Cancel, Help.

Sample ServiceNow Incident Closed

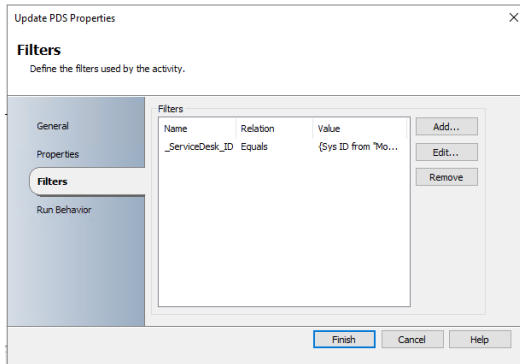
3. Configure the Monitor Activity in the Monitor Service Desk Runbook you are going to use 20.1.1, 20.1.2 or 20.1.3 and map the Incident Ticket ID Published Data to the Filter Criteria for the Update PDS Activity.

The screenshot shows the 'Monitor Records Properties' dialog box with the 'Properties' tab selected. The 'Configuration' section shows 'Name: ServiceNow'. The 'Properties' table lists the following fields and values:

| Table Name              | Value |
|-------------------------|-------|
| Incident [incident]     |       |
| Monitor Interval        | 15    |
| Monitor New Records     | True  |
| Monitor Updated Records | True  |

Buttons at the bottom: Finish, Cancel, Help.

Sample ServiceNow Monitor Request Activity



Sample Update PDS Activity showing Published Data mapped to Filter Criteria

## 7. Optional Diagnostics and Remediation Extension

The solution also includes the capability to launch Orchestrator runbooks to perform diagnostic actions and update the incidents, or even perform automated remediation, again with data passed into the incidents to ensure that your standard ITIL processes and reporting have visibility of the issues and the resolution. We find that many of our customers prefer this approach to “recovery actions” over the native SCOM recovery actions because of the improved visibility to service desk personnel.

The enhanced solution comes with 5 template runbooks for diagnostics and 1 template runbook for remediation. These runbooks are intended to be examples that show how to utilise PowerShell remoting from the solution to execute a variety of command types on the machine that generated the error.

The diagnostic and remediation facilities build upon the enrichment rules. For each of the rules in the SCOM\_EER table the EER\_Action column controls whether an event is evaluated for further actions to be applied after an incident has been raised.

For events that match an enrichment event where the EER\_Action flag is set to 1, the runbooks will query the EER\_ACTION\_MAP for **all of the actions** that match the \_ID of the enrichment Rule. Each of the matching actions will then be added to the EER\_Action\_Queue with the path to the runbook from the EER\_ACTION\_MAP and the \_ID of the event from the SCOM\_Events table.

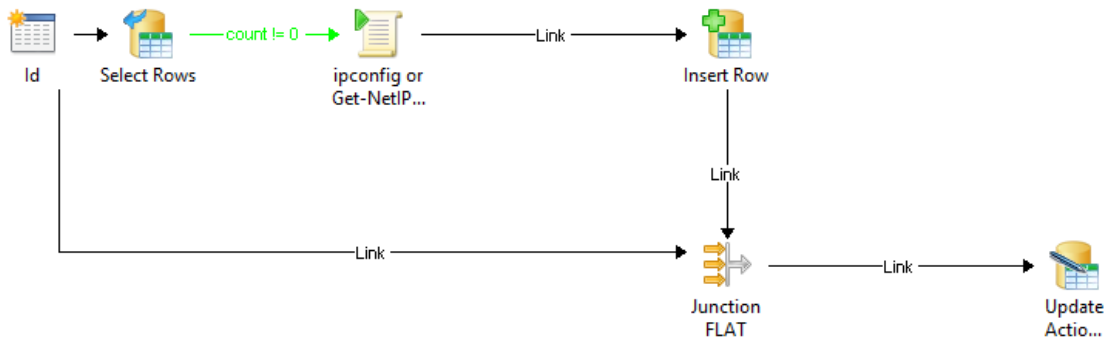
The EER\_ACTION\_QUEUE is monitored by the 30.2\_Monitor\_Action\_Queue runbook which will launch the specific runbooks using the Kelverion Runbook management integration pack. The child runbook that is launched is responsible for updating the PDS with the result of the diagnostics and setting its completion state.

Finally, the runbook 30.3\_Monitor\_Action\_Complete will update the incident with the result of the diagnostics. This runbook is currently configured to update ServiceNow. If you are using one of the alternative help desks that the solution supports, then this runbook will need to be updated.

### 7.1. Structure of a typical diagnostic runbook

The diagnostic runbooks must be authored using an approach that is attuned with the rest of the solution. The example runbooks in the solution are intended as a starting point for your specific diagnostic and remediation steps.

If we look at a typical diagnostic runbook.



The important features to note are:

- 1) The “initialise data” activity at the start of the runbook **always** has a single parameter, the Id of the SCOM event from the SCOM\_EVENTS table. The Runbook Management integration pack invokes the Orchestrator Web service to start the diagnostic runbook, so each of the diagnostic runbooks must have exactly the same “inputs”.
- 2) The SQL IP is used to retrieve the SCOM event from the SCOM\_EVENTS table, so that data from the event can be used to drive the diagnostic/ remedial action. For example, connect to the correct server, and interrogate the database specified in the SCOM event.
- 3) The result of the action is INSERTED into the UNIFIED\_DIAGNOSTICS table so that it can be added into the Incident’s history.
- 4) The EER\_Action\_Queue is updated to show that the action is complete.

## 8. Runbook Performance and Scalability

### 8.1. Testing Environment

As part of the creation of these SCOM Connector Runbooks Kelverion undertook extensive Performance and Scalability tests in our development environment.

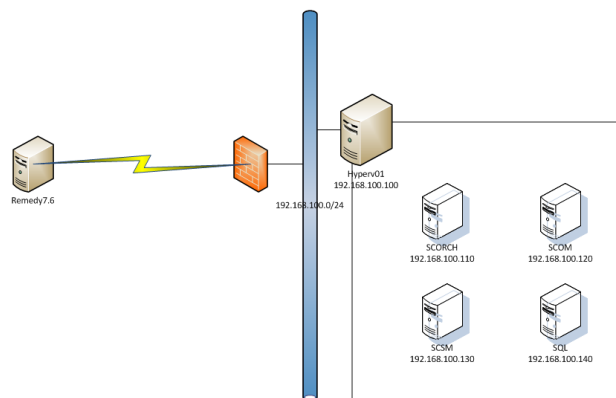
The test environment was spread across two regional data centers in the USA and the UK replicating a not unusual customer implementation. Our BMC Remedy and BEM Systems were hosted remotely and connected via VPN. The VPN connection introduces a network latency between 500 and 1000ms.

Our test system comprised of a Windows Server with:

- A single Core i7 quad core processor with hyperthreading enabled.
- 16Gb Ram.
- Dual Gigabit Interfaces.
- iSCSI storage running on a NAS with a pair of teamed gigabit interfaces. The underlying LUN was a pair of SATA disks configured as a large concat (RAID0)
- HyperV role enabled

This server was running 4 virtual machines, each machine was running Windows with the VHD stored on a separate iSCSI LUN.

1. Domain controller for the domain  
Single vCPU  
4Gb Ram statically Allocated.
2. MS SQL Hosting all of the databases for the test systems.  
Single vCPU  
2-6Gb Ram Dynamically Allocated.
3. System Center Operations manager  
Single vCPU  
2-6Gb Ram Dynamically Allocated.
4. System Center Orchestrator  
4 vCPU  
2-6 Gb Ram Dynamically Allocated



SCOM was configured to monitor the application event log for all events with a source of “TEST”; the duplicate detection for this rule was set such that any distinct event message would lead to the creation of an alert. Any rules that lead to the suppression of alerts were disabled, so we could ensure a consistent high volume of alert from SCOM.

## 8.2. Results

Tests were conducted with batches of SCOM alerts being raised and forwarded. 20 new alerts were raised every minute for 11 hours to generate sustained heavy load and the average time take for the process.

The results are as follows:

| SCOM to Service Desk Connector                                       |                      |
|--|----------------------|
| Action   | Time Average         |
| SCOM Event created -> Creation of Remedy Incident                    | Less than 40 Seconds |
| Creation of Remedy Incident -> propagation of Ticket ID back to SCOM | Less than 20 Seconds |
| Closure within SCOM -> propagation of closure to Remedy              | Less than 40 Seconds |
| Closure within Service Desk -> Closure of event in SCOM              | Less than 40 Seconds |

## 9. Error Reporting

The Operations Manager 2022 Routing and Remediation Runbooks record critical errors to a table in the Persistent Datastore database which under pins the solution. There is a pre-built Runbook which picks these errors up from the database and writes these errors to the Windows Event Log (or as Orchestrator platform events). You can then use your existing Event Management System to monitor the Windows Event Log on the Orchestrator Runbook Server to pick up these errors.

The faults are logging the faults to the Windows Event Log because for reliability there needs to be a fault communication path to the Event Management System which is independent of the Integration Packs being used in Orchestrator. The reason being if it is actually an Integration Pack communication error then Orchestrator has no way to communicate that out if the use of the offending Integration Pack is its only error reporting mechanism.

## 10. Installing Temporary License of Kelverion Integration Pack for SQL Server

To use the Operations Manager 2022 Routing and Remediation you will need a license key for Kelverion SQL Server and Runbook Management Integration Packs. Having downloaded the solution you will receive a temporary evaluation key which will run for 14 days from the date of installation.

The license files will need to be copied into the folder C:\Program Files\Kelverion Automation\Licenses. If this folder does not already exist on your system, please first create the folder C:\Program Files\Kelverion Automation\Licenses and then copy the attached files into it.

The license key is regularly updated as it includes a specific license end date after which the product will no longer work. If you have a license or date format error on trying to run this product please contact [info@kelverion.com](mailto:info@kelverion.com) detailing date of download and error details.

To purchase a license please contact your Kelverion representative, reseller or email [info@kelverion.com](mailto:info@kelverion.com)