Kelverion Automation

Web Page Checks

Deployment Guide

Version 1.1

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

1. Table of Contents

2. Overview

Gauging user experience when interacting with your web pages is a complex affair. The web page check solution from Kelverion brings a simple and flexible approach to gathering Key Performance metrics by leveraging the power of the Kelverion Runbook Suite.

The solution unites the following components

- Kelverion Runbook Studio
- Kelverion Automation Portal
- Kelverion Integration Module for Web Automation
- Microsoft Azure Automation
- Microsoft Azure Log Analytics

These combine to build a straightforward framework to enable web page monitoring from an end user perspective using synthetic transactions executed from within a standard web browser.

The results of those transactions are measured, and the data pushed into Microsoft Log Analytics. Then the KPI data can be consolidated with multiple other data sources to provide a holistic view of web page health and performance. Allowing you to pre-empt impacting issues, raise Incidents when web performance issues are detected and enrich the data streams that are used for troubleshooting.

3. General Configuration Steps

3.1. Pre-Installation Information

The Web Page Check Package contains the following elements:

- Persistent Data Store (PDS) SQL configuration script
- PDS definition spreadsheet
- 3 Azure Automation Runbooks
- Kelverion Automation Portal Service export file.

Kelverion Items Required

The solution requires the following Kelverion products:

- Kelverion Runbook Studio
- Kelverion Automation Portal
- Kelverion Integration Module for SQL Server
- Kelverion Integration Module for Web Automation

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

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 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 PDS – WebCheckSolution.sql*) from the package to Create the database tables, views and stored procs.

3.3. Hybrid worker

The Kelverion Integration Module for Web Automation spawns a web browser and controls that browser to interact with web pages.

As the Azure Automation "sandpit" where runbooks are executed is limited (by design) a browser cannot be executed within it. This means that the solution MUST be configured to execute on a hybrid worker.

The hybrid worker must be configured and added to the automation account. If this hybrid worker is an Azure Virtual Machine, then its recommended that the VM is configured as a D2 or higher due to the CPU and memory load from spawning Web Browsers.

For a proof of concept, a B2 machine may be suitable, however you should monitor the CPU credits and balance with the execution frequency of the runbooks to ensure that the VM is not starved of compute power. 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.

3.4. 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 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.

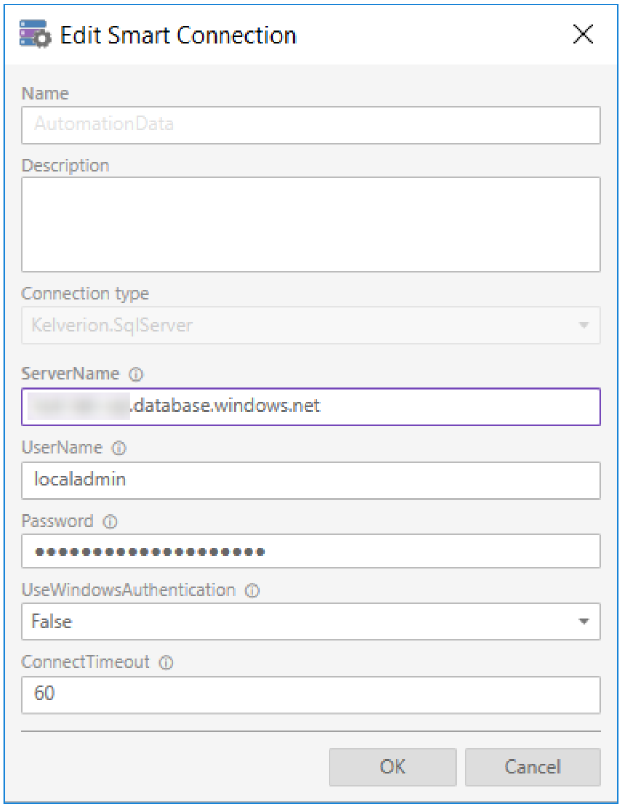
3.5. Configure your Automation account using the Runbook Studio

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.

3.5.1. 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.

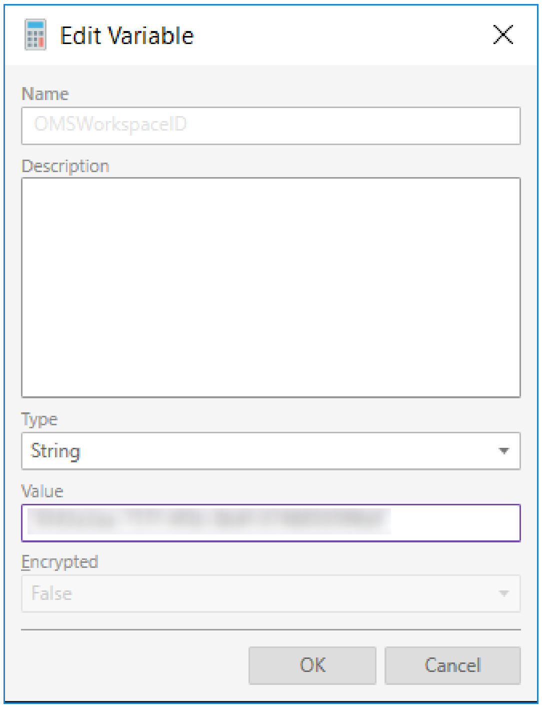
Create the following smart connections within the Runbook Studio.

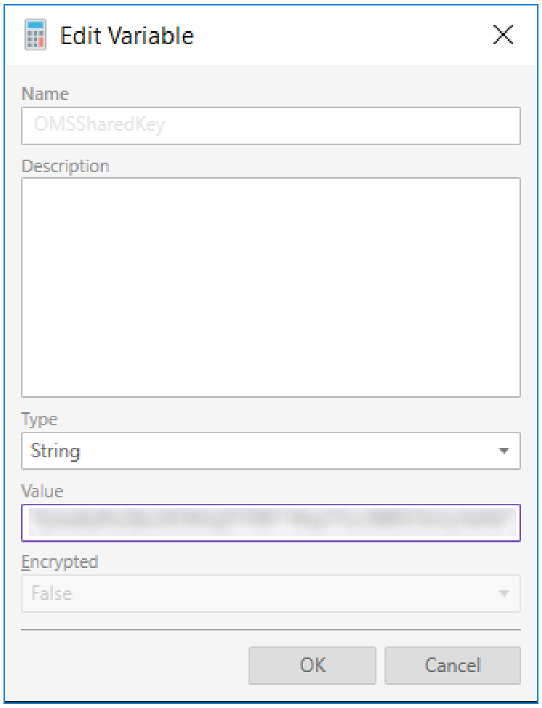
AutomationData	 Name: AutomationData Description: Connection type: Kelverion.SqlServer ServerName: [redacted].database.windows.net UserName: localadmin Password: •••••••••••••••••••• UseWindowsAuthentication: False ConnectTimeout: 60

3.5.2. Azure Variables

Azure variable allows 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 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

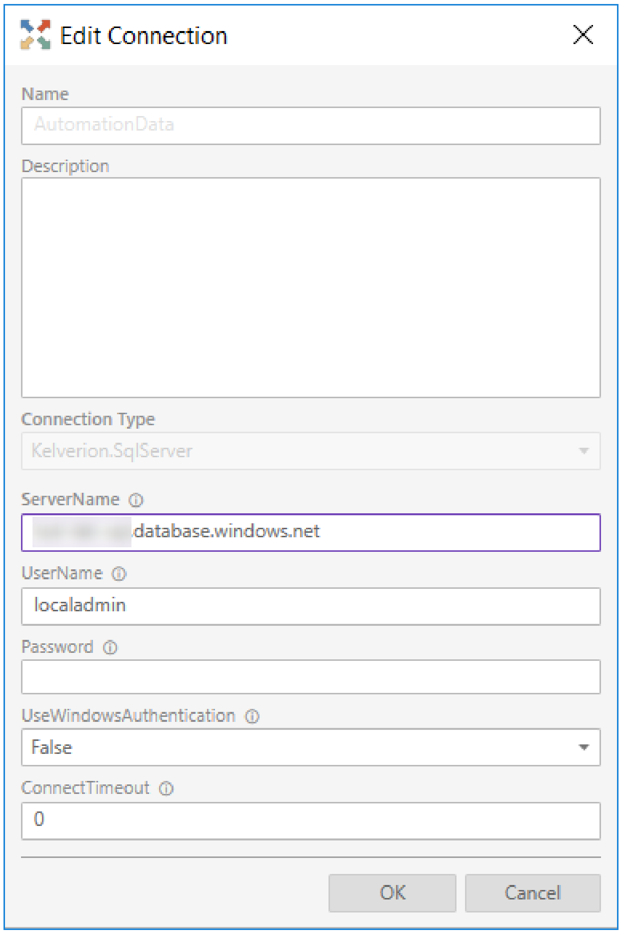
OMSWorkspaceID	The OMS workspace ID can be found in the "Advanced Settings" of the OMS workspace	 Edit Variable Name: OMSWorkspaceID Description: Type: String Value: Encrypted: False

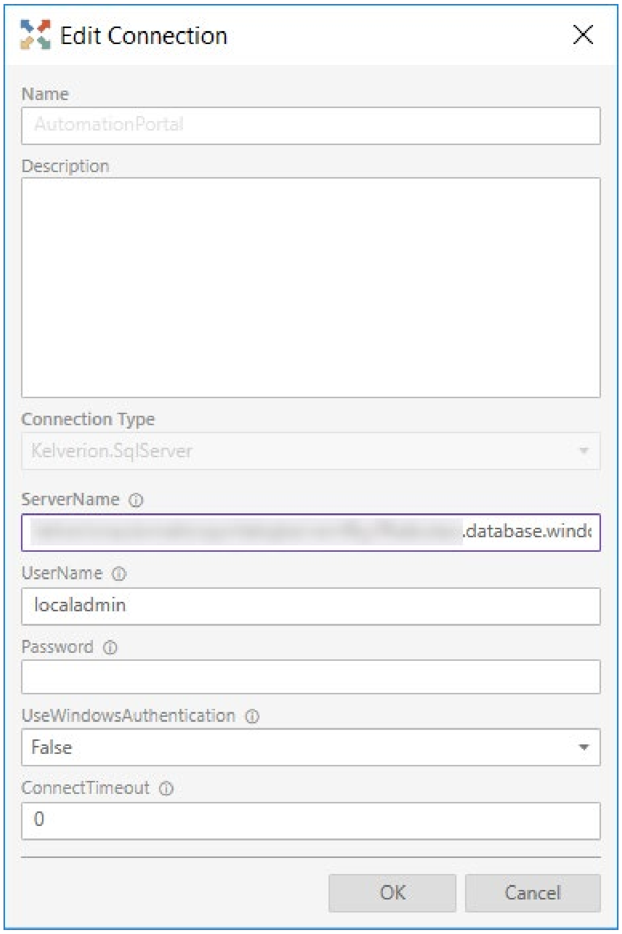
OMSSharedKey	The OMS Shared key can be found in the "Advanced Settings" of the OMS workspace. Either the Primary or secondary key for the account can be used.	 Edit Variable Name: OMSSharedKey Description: Type: String Value: [redacted] Encrypted: False

3.5.3. 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 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.	 Edit Connection Name: AutomationData Description: Connection Type: Kelverion.SqlServer ServerName: [redacted].database.windows.net UserName: localadmin Password: UseWindowsAuthentication: False ConnectTimeout: 0

AutomationPortal	The "AutomationPortal" connection asset in Azure defines the connection that will be used at runtime for the runbooks to connect to the Kelverion Automation Portal Database.	 Name: AutomationPortal Description: Connection Type: Kelverion.SqlServer ServerName: [redacted].database.wind UserName: localadmin Password: UseWindowsAuthentication: False ConnectTimeout: 0

3.6. 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.

Create_OMS_Event.runbook	Utility Runbook to Create a custom log entry within OMS using the OMS Rest API
Check-Web-Page-Get-MasterURLs.runbook	Kelverion Automation Portal integration runbook. This runbook gets new requests from the Automation Portal requests table and extracts the master URL requested and adds it to the PDS

Check-Web-Page.runbook	The core runbook of the solution that loads each of the webpages in the browser and times how long it takes to load. The load time is sent to OMS, and the page is parsed for URLS to be added to the PDS.

3.7. Testing

The following steps allow you to prove that all the components have been configured correctly. Testing the components should take place before the runbooks are scheduled for repeated execution.

1. Using the Automation Portal Create a request for a new Master URL
2. Using the Azure Portal Start runbook:
3. Check-Web-Page-Get-MasterURLs
4. When the runbook has completed check the state of the request in the portal it should now be "completed"
5. Check the PDS table MasterURLs, there should be a new record for your request.
6. Start Runbook Check-Web-Page
7. When the runbook has completed check the ProcessedURLs and ActivityTrace tables for new records.

Any data posted to OMS will take about 15 minutes before it is visible.
The data will show up as the LogType "CheckWebPage_CL"

3.8. 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. To work around this issue, we recommend the use of Azure Logic Apps for scheduling the runbook execution. In a typical scenario we would schedule the runbooks to execute on a 15 minute cycle, however you can increase or decrease the timings to balance the execution cost versus the granularity of performance data.

Please note:
The runbooks **Create_OMS_Event** is called on demand and does not need to be scheduled.

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