

Kelverion Automation

Standard Tasks Application for Azure Automation

Deployment Guide

Version 2.5

Email: info@kelverion.com

Web: www.kelverion.com

1 Table of Contents

2	Overview.....	3
3	General Configuration Steps	3
3.1	Pre-Installation Information	3
3.1.1	<i>Kelverion Items Required</i>	<i>3</i>
3.1.2	<i>Other Products Required</i>	<i>4</i>
3.2	Installation Steps.....	4
3.3	Persistent Data Store.....	5
3.4	Configure the Automation Portal.....	5
3.5	Create an Azure system assigned managed identity for starting runbooks.....	6
3.6	Load Integration Modules	6
3.7	Configure your Automation account using the Runbook Studio.....	6
3.7.1	<i>Design Time (Smart Connections).....</i>	<i>6</i>
3.7.2	<i>Azure Variables.....</i>	<i>8</i>
3.7.3	<i>Azure Credentials.....</i>	<i>9</i>
3.7.4	<i>Azure Runtime connections.....</i>	<i>10</i>
3.8	Import Runbooks using the Runbook Studio	12
3.9	Runbook Customisation.....	15
3.9.1	<i>High Level Overview.....</i>	<i>15</i>
3.10	Runbook Process Flow	16
3.10.1	<i>Data Storage (PDS Design).....</i>	<i>16</i>
3.10.2	<i>Gathering the Data</i>	<i>17</i>
3.10.3	<i>Transposing the Data</i>	<i>18</i>
3.10.4	<i>Using the Data.....</i>	<i>19</i>
3.10.5	<i>Returning the Results.....</i>	<i>20</i>
3.11	Logic Apps - Scheduling runbook execution.....	21
3.11.1	<i>Gather Request Logic App.....</i>	<i>21</i>
3.11.2	<i>Return Status Logic App.....</i>	<i>25</i>
3.11.3	<i>Discovery Logic App</i>	<i>29</i>
3.12	Testing.....	33

2 Overview

Businesses would like to give team leaders and managers access to create their own users and groups via self-service facilities to reduce the workload on the heavily utilised service desk and front-line support teams. Equally with I.T. Security being in the forefront of everyone's minds it is a challenge that requires careful balancing between providing the right facilities to end users, and not compromising the security or integrity of your configuration.

Therefore, companies are increasingly looking for Automation to improve consistency and supportability, along with the self-service facilities whilst maintaining the security controls. To meet this need we have built the Standard Tasks Application for Azure Automation.

The Kolverion Standard Tasks application for Azure Automation has been developed based on our experiences of common IT tasks. It is designed to meet the needs of customers who want to provide a simplified experience of carrying out common IT tasks as well as make use of Microsoft's hybrid cloud and on-premises architecture.

All the actions are carried out by your Azure Automation account, so every change is logged and carried out in a consistent way.

The Runbooks have been written using the Runbook Studio authoring application and leverage the integration and smart discovery capabilities provided by the Integration Module for SQL Server. These Integration Modules are also available in the PowerShell Gallery.

3 General Configuration Steps

3.1 Pre-Installation Information

The Standard Tasks application package contains the following elements:

- Persistent Data Store (PDS) SQL configuration script
- Standard Tasks Azure Automation Runbooks
- Azure Automation Service Export

3.1.1 Kolverion Items Required

The application requires the following Kolverion products:

- Kolverion Runbook Studio
- Kolverion Integration Module for SQL Server
- Kolverion Automation Portal

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

This guide assumes that you have already installed the Runbook Studio and the Automation Portal. If you have not yet installed those products, then please do so before you continue.

Each of the product downloads contains its own documentation to guide you through the initial configuration.

3.1.2 Other Products Required

The following Microsoft PowerShell modules are required:

- Azure AD
- Az.Accounts
- Az.Automation
- MSOnline

These modules are available from the PowerShell Gallery

3.2 Installation Steps

As a guide the steps taken are as followed:

1. Configure the PDS database
2. Import and configure the Portal Service (If using the Keverion Automation Portal)
3. Configure the Service Offerings (If not using the Keverion Automation Portal)
4. Create the Azure components
 - a. Resource Group
 - b. Automation Account
 - c. Managed Identity
 - d. Load Integration Modules
 - e. Import Runbooks
 - f. Create Smart Connections
 - g. Create Azure Variables
5. Create the Gather Runbook 90-XX (If not using the Keverion Automation Portal)
6. Create the Return Runbook 97-XX (If not using the Keverion Automation Portal)
7. Configure the Convert Runbook 95-XX
8. Create the Logic Apps

3.3 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 application. 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 (*PDS_STSK.sql*) from the package to Create the database tables and views.

3.4 Configure the Automation Portal

Import the service request definition `Kelverion_STSK_Services.export`

The following queries must be updated to point to the customers PDS

- StandardTasks-GetAD
- StandardTasks-GetADUser2
- StandardTasks-GetOU
- StandardTasks-GetADGroup
- StandardTasks-GetADGroups2
- StandardTasks-GetADGroupMembers
- StandardTasks-GetNonADGroupMembers
- StandardTasks-ShowADGroupsNotIn
- StandardTasks-ShowADGroupsIn
- StandardTasks-GetServers
- StandardTasks-GetServices
- StandardTasks-GetMultipleServers
- StandardTasks-GetVirtualServers
- StandardTasks-GetHybridWorkers
- StandardTasks-GetExchangeDomain
- StandardTasks-GetExchangeMbld

- StandardTasks-GetExchangeMbDb

The SQL instance AND authentication details will need to be updated for each query.

3.5 Create an Azure system assigned managed identity for starting runbooks

In the automation account, go to Identity. Enable the System assigned managed identity.

Allocate the role of “Automation Operator” for the resource group that the automation account is in.

3.6 Load 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 modules can easily be installed on the machine from the PowerShell Gallery.

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

The following modules are required

- Keverion.SqlServer
- AzureAD
- Az.Accounts
- Az.Automation


3.7 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.

3.7.1 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.

<p>AutomationData</p>	<div><div><div> Edit Smart Connection</div><div><div><div><div>Name</div><div>AutomationData</div></div><div><div>Description</div><div></div></div><div><div>Connection type</div><div>Kelverion.SqlServer</div></div><div><div>ServerName ⓘ</div><div>████████████████████database.windows.net</div></div><div><div>UserName ⓘ</div><div>localadmin</div></div><div><div>Password ⓘ</div><div>●●●●●●●●●●●●●●●●</div></div><div><div>UseWindowsAuthentication ⓘ</div><div>False</div></div><div><div>ConnectTimeout ⓘ</div><div>60</div></div></div><div><div>OK</div><div>Cancel</div></div></div></div></div>
-----------------------	---

3.7.2 Azure Variables

Azure variables 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 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.

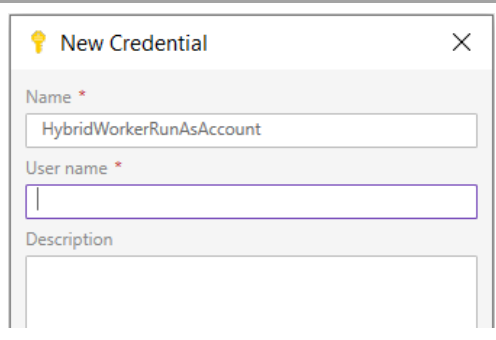
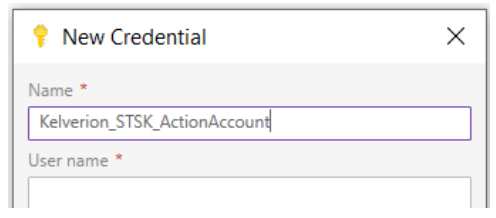
Variable Name	Value
Kelverion_STSK_AA	The Name of the Automation Account
Kelverion_STSK_RG	The Resource Group that contains the Automation Account
TenantID	The TenantID of the host environment
Subscription	The name of the subscription that you are launching the runbooks in

3.7.3 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.

Any runbook in this application that is launched on a Hybrid Worker (for on premise actions) will use an account linked to the Hybrid Worker Group associated with the automation account.

Create the following Credential Assets using the Runbook Studio.

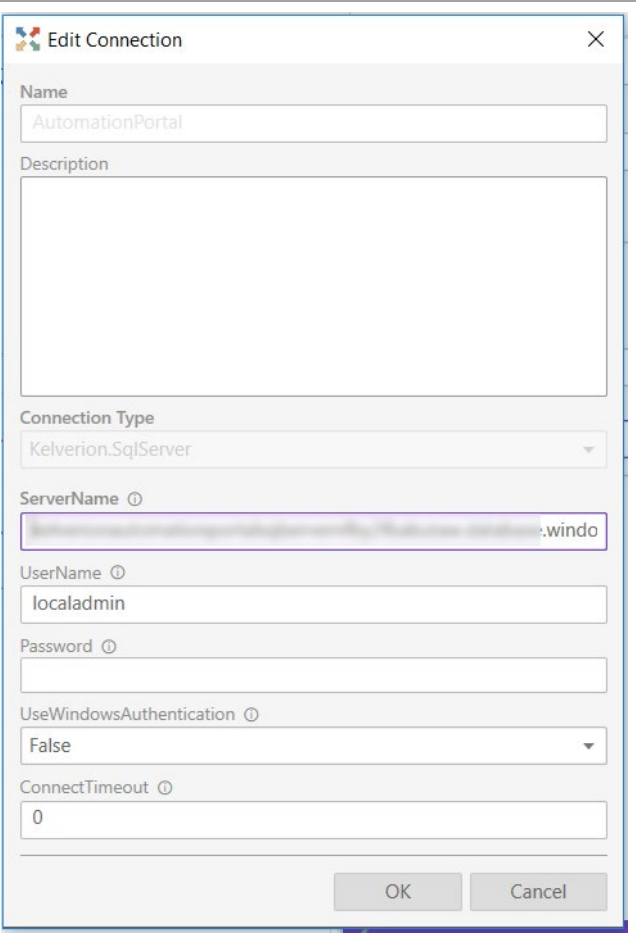
HybridWorkerRunAs Account	This credential will be used to launch the runbooks on the Hybrid Worker	
Kelverion_STSK_ActionAccount	Account used to carry out remote actions on Servers	

3.7.4 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 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.	
----------------	---	--

AutomationPortal	<p>The "AutomationPortal" connection asset in Azure defines the connection that will be used at runtime for the runbooks to connect to the Kolverion Automation Portal Database.</p> <p>N.B. Only required if you are using the Kolverion Automation Portal as a front end for the service.</p>	
------------------	---	---

3.8 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.

Runbook Name	Brief Description
Kelverion_STSK_10-1_Worker_Restart-Service	Restarts a service on a selected device
Kelverion_STSK_11-1_Worker_Remote-Reboot	Reboots a selected device
Kelverion_STSK_12-1_Worker_Remote-Shutdown	Shutdown one or more devices
Kelverion_STSK_13-0_Process-Start-AzureVM	Start one or more Azure virtual machines
Kelverion_STSK_14-0_Process-Stop-AzureVM	Stop one or more Azure virtual machines
Kelverion_STSK_15-0_Process-Restart-AzureVM	Restart one or more Azure virtual machines
Kelverion_STSK_20-0_Util-GetRunbookData	This runbook will gather Request data from the Automation Portal
Kelverion_STSK_20-1_Worker_AD-Create-User	Creates a new AD user account
Kelverion_STSK_21-0_Worker_AD-Add-Group-Members	Parent runbook to add AD Group members and then runs discovery to update the PDS
Kelverion_STSK_21-1_Worker_AD-Add-Group-Members	Adds group members to an AD group
Kelverion_STSK_22-1_Worker_AD-Reset-Password	Resets an AD user account password
Kelverion_STSK_23-1_Worker_AD-Enable-Disable	Enables or Disables an AD user account

Kelverion_STSK_24-0_Worker_AD-Remove-Group-Members	Parent runbook to remove AD Group members and then runs discovery to update the PDS
Kelverion_STSK_24-1_Worker_AD-Remove-Group-Members	Removes group members from an AD group
Kelverion_STSK_25-0_Worker_AD-Add-User-To-Groups	Parent runbook to add an AD user to groups and then runs discovery to update the PDS
Kelverion_STSK_25-1_Worker_AD-Add-User-To-Groups	Adds an AD user to AD groups
Kelverion_STSK_26-0_Worker_AD-Remove-User-From-Groups	Parent runbook to remove an AD user from groups and then runs discovery to update the PDS
Kelverion_STSK_26-1_Worker_AD-Remove-User-From-Groups	Removes an AD user from AD groups
Kelverion_STSK_30-1_Worker_Ping-Check	Sends an ICMP request to one or more devices
Kelverion_STSK_31-1_Worker_Ping-Url	Checks a specified URL from a chosen location (HybridWorker)
Kelverion_STSK_32-1_Worker_Tracert	Runs a Tracert against a specified IP address from a chosen location (HybridWorker)
Kelverion_STSK_33-1_Worker_Diags-Events	Searches Windows Event Logs on selected chosen device(s). Optional filters available
Kelverion_STSK_35-1_Worker_Clear-Diskpace	Deletes files from (Recycle Bin \ Downloads \ Custom Folder) on one or more selected devices
Kelverion_STSK_40-1_Worker_Add-User-Mailbox	Creates an AD user account and matching Exchange mailbox
Kelverion_STSK_41-1_Worker_Add-Shared-Mailbox	Creates a shared mailbox in Exchange
Kelverion_STSK_42-1_Worker_Delete-Mailbox	Removes or permanently deletes an Exchange mailbox

Kelverion_STSK_90-01_KAP_Get_Request	Gathers the request data from the Kelverion Automation Portal. Requires a Logic App to trigger
Kelverion_STSK_95-01_ConvertToRequestData	Converts the gathered data to the correct format for the runbooks
Kelverion_STSK_97-01_KAP_Return_Status	Collects processed rows of the PDS for returning data back to the Kelverion Automation Portal. Requires a Logic App to trigger

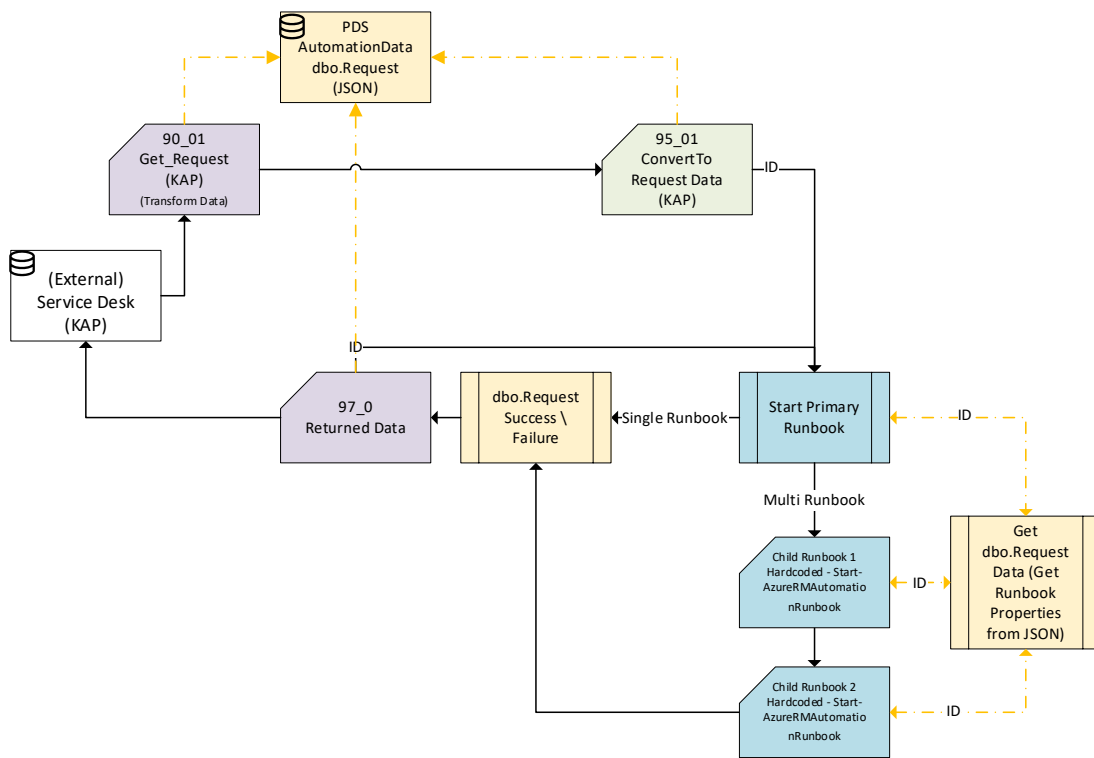
3.9 Runbook Customisation

This application is designed to allow flexibility across multiple service desk applications. The initial runbooks are configured for using the Keverion Automation Portal as the main user portal for initiating any requests and receiving the status updates.

If you are not using the Keverion Automation Portal, then the installation engineer will be required to create a gathering runbook (90_XX) and a return runbook (97_XX).

3.9.1 High Level Overview

The main process flow is as follows:



3.10 Runbook Process Flow

This section covers a more detailed description of how the runbook logic fits together. You should be able to use it to configure and customise the application.

3.10.1 Data Storage (PDS Design)

The applications use a SQL database (PDS) to store the request \ offerings from the Service Desk. The table dbo.Request stores data per request from the source Service Desk. Each row has a unique ID that is used as a reference for the worker runbooks.

RunbookOwner	Data	Message	ServiceName	OfferingName	ExternalId	State	OutputData
Kelverion_STSK...	{ "First Nam...	User **Orchestrator....	Business Us...	New Joiner	306	Complete	{ "Runbook": "Ke
Kelverion_O365_...	{ "Reason fo...	User disabled	Office 365	Disable Login	305	Complete	{ "ObjectId": "240
Kelverion_O365_...	{ "User": { ...	User enabled	Office 365	Enable User	304	Complete	{ "ObjectId": "240

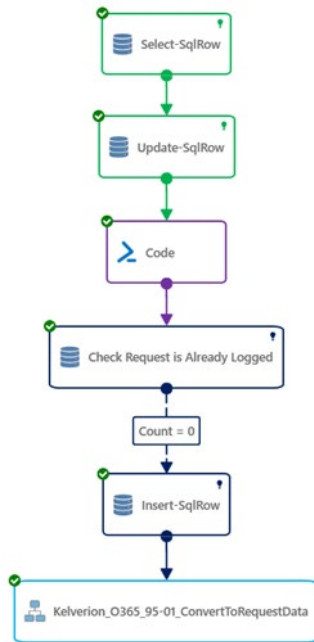
3.10.1.1 Request Table

Description of the column use in the dbo.Request table.

Column	Description
ID	Unique column ID. Created when data is inserted.
Created	Create time stamp
Updated	Updated time stamp
RequestedBy	Service Offering Requestor
Data	JSON Data dump from the request offering
Message	Return message to the service desk
Deleted	(Bit) 1 = request deleted
ServiceID	Numerical ID of the Service Name (Optional and service desk dependent)
ServiceName	Service Name from the service desk request
OfferingID	Numerical ID of the Offering Name (Optional and service desk dependent)
OfferingName	Offering Name from the service desk request
ExternalId	Service Desk reference ID
State	Worker Runbook state
RowVersion	SQL Row version. Generated automatically on data entry
ServiceDesk	Name of the service desk being used
OutputData	JSON Data used by the worker runbooks

3.10.2 Gathering the Data

The base application comes with a runbook designed to use the Kelverion Automation Portal. Other Service Desk portals with available Kelverion Integration Modules can be used too, but a gathering runbook will need to be created.



The runbooks are driven from an Azure Logic App, to either:

- Monitor the Service Desk portals database (SQL)

or

- Launch the gather runbook every 'x' minute(s)

Each application has a 90-0x_XXX_Get_Request runbook that can be configured for the required Service Desk portal.

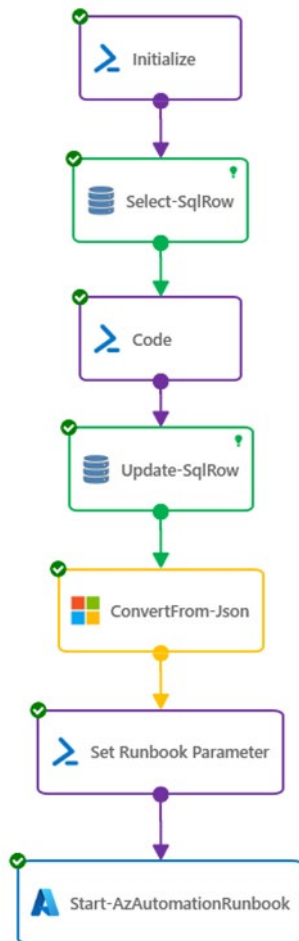
This runbook has a code block that will transpose the Automation Portal request into JSON format and store it into the PDS dbo.Request table. Other service desks will require different activities to store the code in the corresponding JSON format.

RunbookOwner	Data	ServiceName	OfferingName	ExternalId	State	OutputData
Kelverion_STSK_21-1_Work...	{ "First Name": "Bob", ...	Business User...	New Joiner	306	Complete	{ "Runbook": "Kelverion_JML_30-...
Kelverion_O365_10-4_Worke...	{ "Reason for Blocking ...	Office 365	Disable Login	305	Complete	{ "ObjectId": "240ba31e-6a1a-46d...
Kelverion_O365_10-2_Worke...	{ "User": { "..."	Office 365	Enable User	304	Complete	{ "ObjectId": "240ba31e-6a1a-46d...

3.10.3 Transposing the Data

The modular runbooks are designed to use JSON.

However, they require the JSON to be in a specific format. Each application has a runbook (Kelverion_XXX_95-01_ConvertToRequestData) to transpose the data into the required format for the worker runbooks.



These runbooks will all look similar. This depends on if the worker runbooks require the use of a Hybrid Worker. In this case you may see additional logic at the end of the Runbook.

The only activity that will require modification is the “Code” activity. This PowerShell activity has an input of JSON (Data) and transposes it to JSON (OutputData).

Example Code for an offering with a single worker runbook.

```

"Delete User" {
    $out = [PSCustomObject]@{
        Service Desk Offering Name
        Runbook = "Kelverion_0365_10-3_Worker-Delete-User"
        Worker Runbook Offering Name
        "Delete User" = [PSCustomObject]@{
            UserPrincipalName = $inputConverted.'User'. 'UserPrincipalName'
            Worker Runbook Inputs
            ObjectId = $inputConverted.'User'. 'ObjectId'
        }
    }
}
  
```

Each service desk offering must pass through the name of the ‘Parent Worker Runbook’. This can be a hidden field in the service desk portal.

The activity Set Runbook Parameter passes the ID column, from the Request table, and feeds it into the Start-AzureRMAutomationRunbook activity.

Start-AzureRMAutomationRunbook launches the Parent Worker Runbook defined from the Service Desk offering.

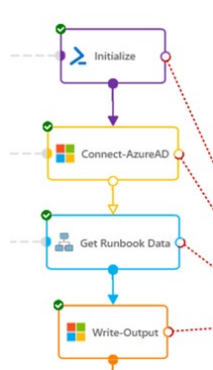
RunbookOwner	Data	ServiceName	OfferingName	ExternalId	State	OutputData
Kelverion_STSK_21-1_Work...	{ "First Name": "Bob", ...	Business User...	New Joiner	306	Complete	{ "Runbook": "Kelverion_JML_30-...
Kelverion_O365_10-4_Worke...	{ "Reason for Blocking ...	Office 365	Disable Login	305	Complete	{ "ObjectId": "240ba31e-6a1a-46d...
Kelverion_O365_10-2_Worke...	{ "User": { "..."	Office 365	Enable User	304	Complete	{ "ObjectId": "240ba31e-6a1a-46d...

3.10.4 Using the Data

Each worker runbook has been designed with the following:

- Single input of ID (from the dbo.Request table)
- Calls a child runbook Kelverion_XXXX_20-0_Util-GetRunbookData to gather the required runbook inputs

This allows the correct input data to be gathered back from the PDS.



Here is an example start of a worker runbook.

It will:

- Set some initial variables (Initialize)
- Connect to Azure (Connect-AzureAD)
- Call 'Get Runbook Data' runbook
- Output the returned data (Write-Output) for use in the main runbook activities

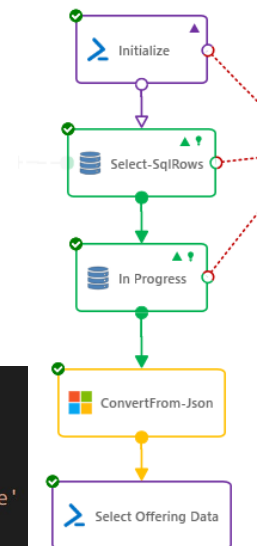
N.B. More complex array outputs may require a PowerShell code block to process them, rather than a Write-Output.

'Get Runbook Data' has 2 required inputs:

1. ID
2. OfferingName (Worker Runbook Offering Name)

The activity 'Get Runbook Data' uses the ID input to go and retrieve the OutputData from the PDS.

The data is converted back from JSON and the 'Select Offering Data' filters out the required data based on the worker runbook offering name.



```

'Delete User' {
  Service Desk      $out = [PSCustomObject]@{
  Offering Name     Runbook = 'Kelverion_0365_10-3_Worker-Delete-User'
  Worker Runbook    'Delete User' = [PSCustomObject]@{
  Offering Name     UserPrincipalName = $inputConverted.'User'.UserPrincipalName
  Worker Runbook Inputs ObjectId = $inputConverted.'User'.ObjectId
  }
}
  
```

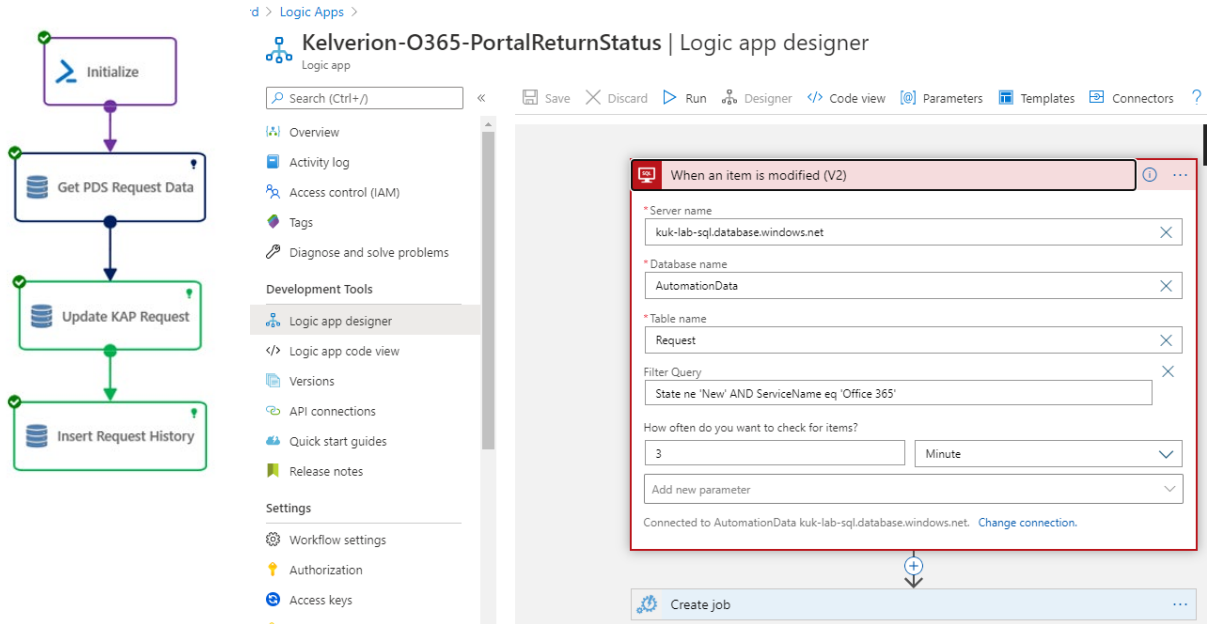
Annotations in the image:

- Service Desk** points to the outer object name 'Delete User'.
- Offering Name** points to the 'Runbook' property value.
- Worker Runbook** points to the inner object name 'Delete User'.
- Offering Name** points to the 'UserPrincipalName' property value.
- Worker Runbook Inputs** points to the 'ObjectId' property value.
- Parent Runbook Name** points to the 'Runbook' property value.

3.10.5 Returning the Results

Each application has a return runbook that gathers updates to the PDS table and returns the message and state back to the required Service Desk portal.

For the Kolverion created applications, this uses the Kolverion Automation Portal. The runbook will be launched via a Logic App in Azure that detects the changes in the dbo.Request table.



N.B. The Logic App filter can be modified if more than one ServiceName exists so that a single Logic App can be used for multiple applications.

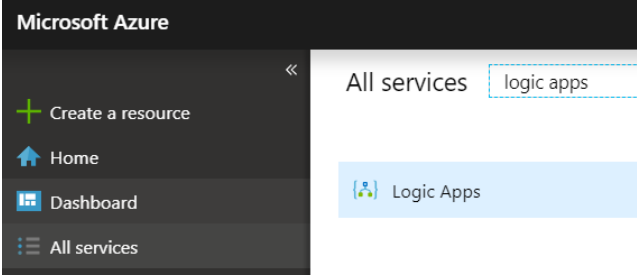
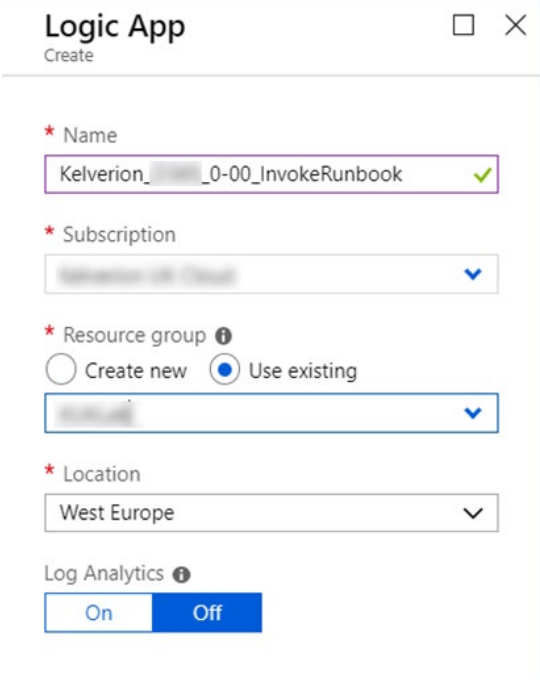
3.11 Logic Apps - Scheduling runbook execution

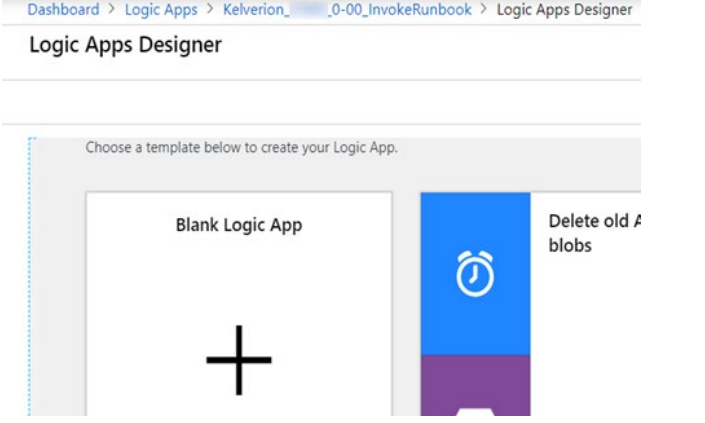
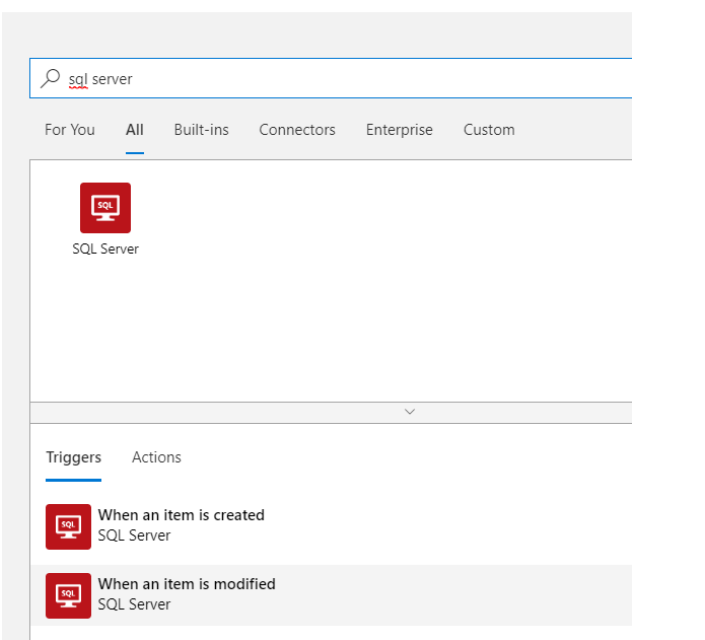
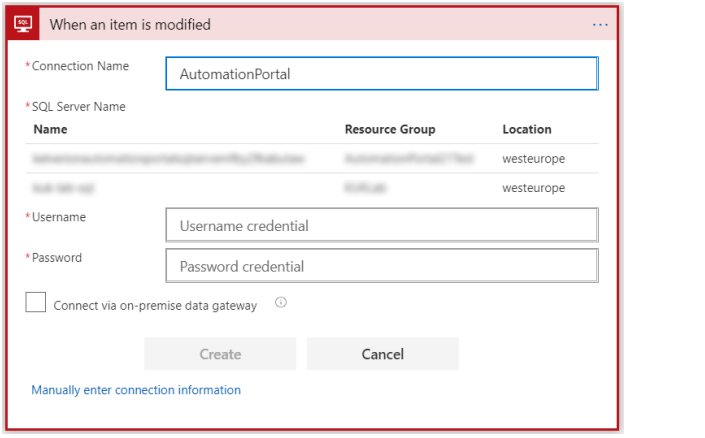
Once all the runbooks (and other assets) are in place and the runbooks have been tested you will need to schedule the runbooks for repeated execution.

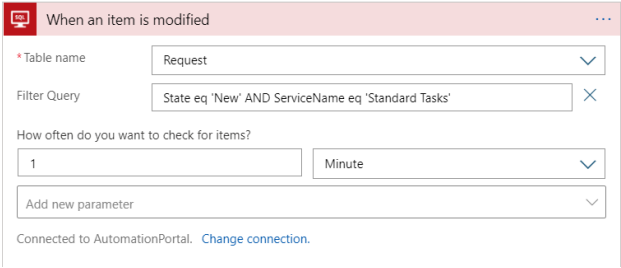
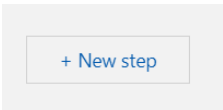
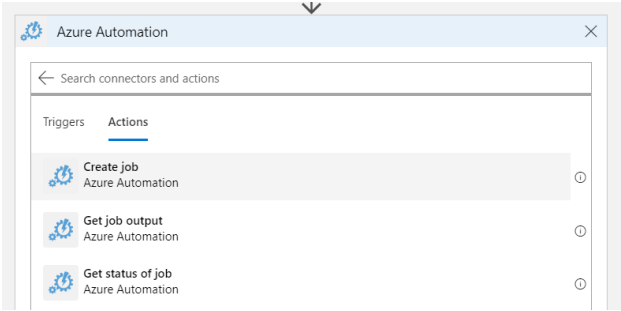
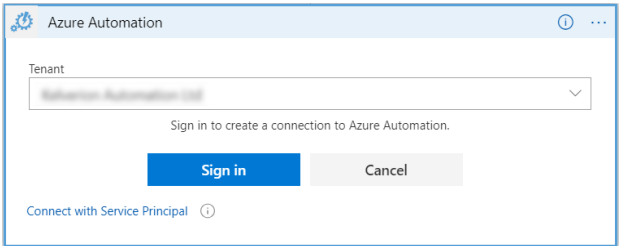
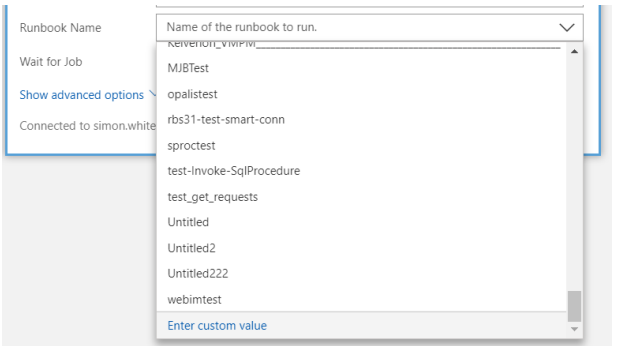
Two Logic Apps are required. One to gather the information from the required Service Desk application (in this case the Automation Portal) and one for returning the status of the runbook to the Service Desk application.

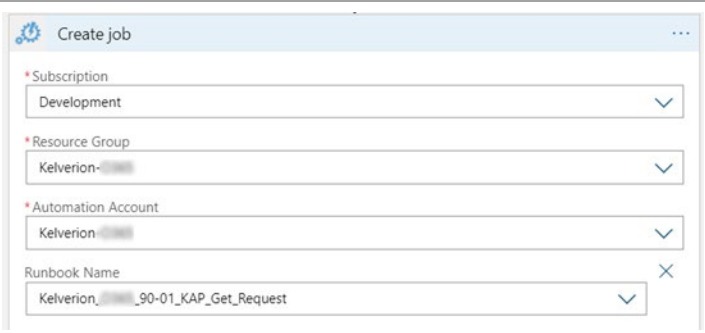

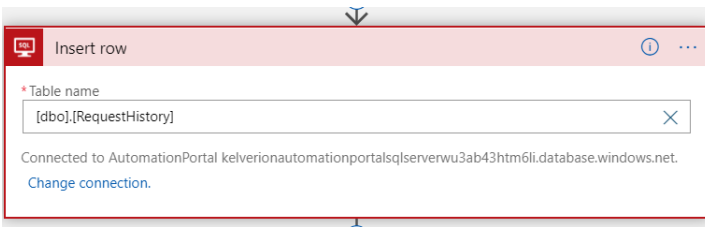
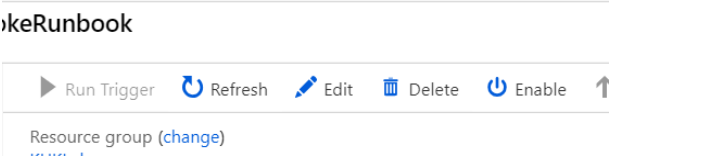
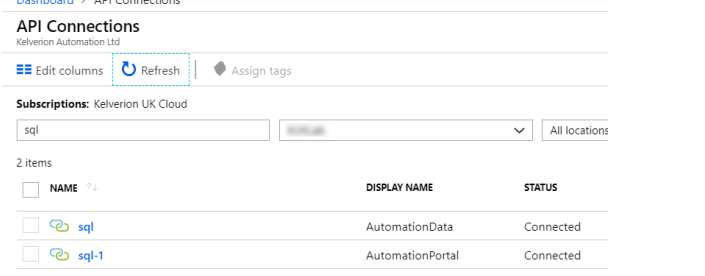
3.11.1 Gather Request Logic App

This Logic App will call the runbook Kolverion_STSK_90-01_KAP_Get_Request.

Step	
Login to the Azure Portal and go to "Logic Apps"	 The screenshot shows the Microsoft Azure portal interface. On the left is a dark sidebar with navigation links: 'Create a resource', 'Home', 'Dashboard', and 'All services'. On the right, the 'All services' page is displayed, with a search bar containing 'logic apps'. Below the search bar, the 'Logic Apps' service is highlighted in blue.
Create a New Logic App and name it appropriately for this application. e.g. Kolverion_STSK_0-00_InvokeRunbook Add it to the same Resource Group that you have deployed the runbooks too	 The screenshot shows the 'Create Logic App' form in the Azure portal. The form has a title bar with 'Logic App' and a 'Create' button. It contains several fields: 'Name' with the value 'Kolverion_0-00_InvokeRunbook' and a green checkmark; 'Subscription' with a dropdown arrow; 'Resource group' with radio buttons for 'Create new' and 'Use existing' (selected), and a dropdown arrow; 'Location' with a dropdown arrow showing 'West Europe'; and 'Log Analytics' with 'On' and 'Off' buttons.

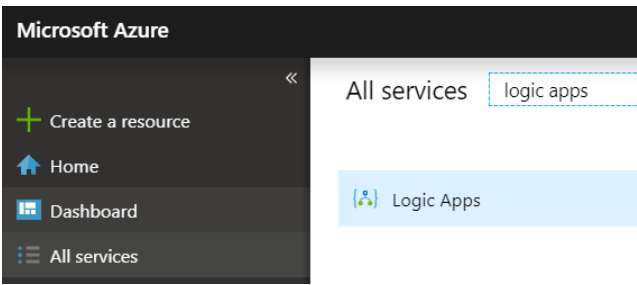
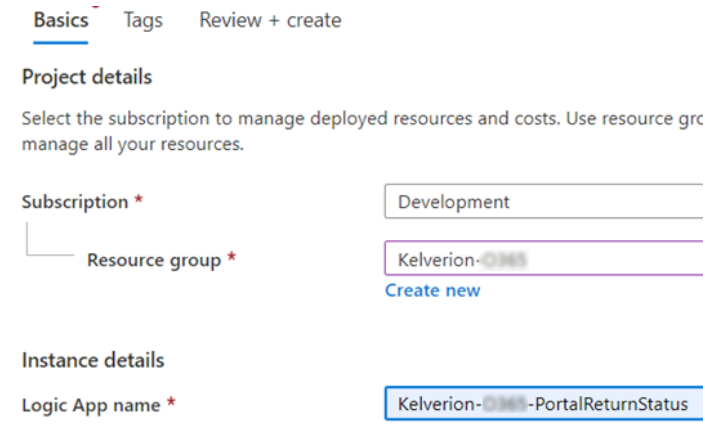
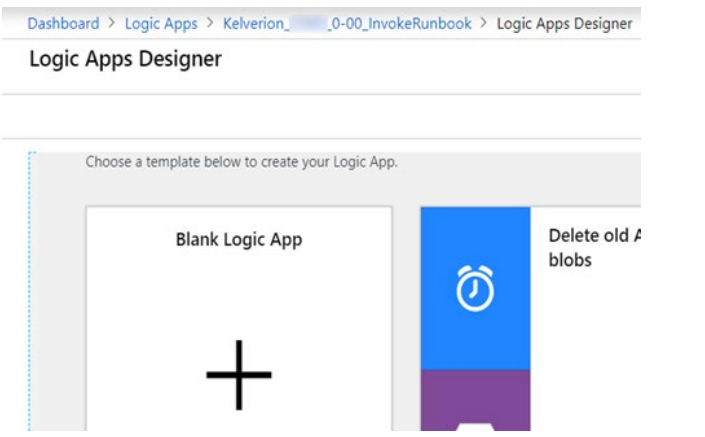
<p>Open the newly created Logic App and select “Blank Logic App”</p>	
<p>Search for “Sql Server” and select the Trigger “When an item is modified”</p>	
<p>If you do not already have a connection (API Connection) then you will need to set one up for your Automation Portal database.</p> <p>Enter the appropriate connection details for the database.</p>	

<p>Configure the activity as shown. Table = Request</p> <p>Filter Query: State eq 'Approved' AND ServiceName eq 'Standard Tasks'</p> <p>Interval = 1 Minute</p>	
<p>Click on New Step to add another activity</p>	
<p>Search for "Azureautomation" and select the Action "Create Job"</p>	
<p>If you have not done so before, you will need to create an API Connection to your tenant. Use the required Azure login details to make the connection.</p>	
<p>Enter the appropriate: Subscription \ Resource Group \ Automation Account For Runbook Name, scroll to the bottom and select "Custom Value"</p>	

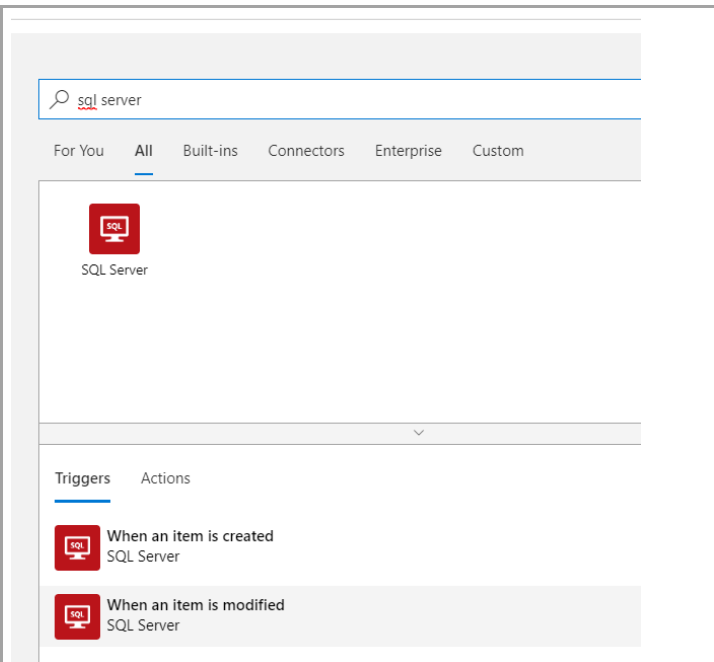
<p>Select the Runbook from the drop down list as:</p> <p>Kelverion_STSK_90-01_KAP_Get_Request</p>	
<p>Add Runbook Parameters. Where ID is from the List of Items from the previous activity</p>	
<p>Automation Portal Only: Add a new activity to write a line to the RequestHistory table</p>	
<p>Ensure the Logic App is active by clicking on Enable</p>	
<p>If you need to change connection details, you should be able to find your connection information in “API Connections”</p>	

3.11.2 Return Status Logic App

This Logic App will call the runbook Kelverion_STSK_97-01_KAP_Return_Status.

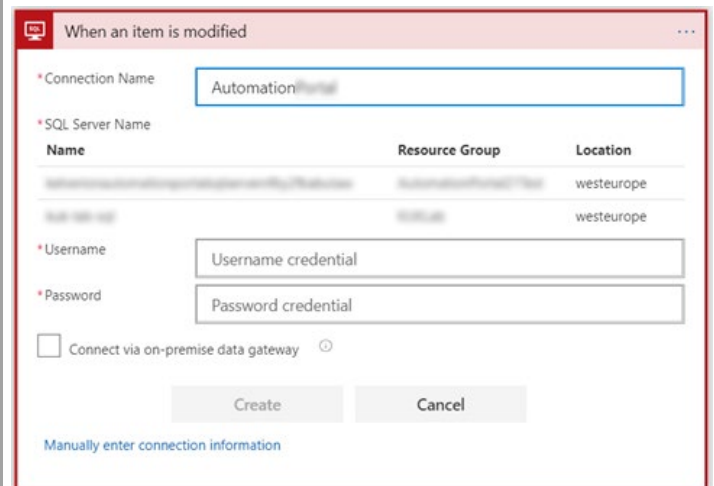
Step	
Login to the Azure Portal and go to "Logic Apps"	 <p>The screenshot shows the Microsoft Azure portal interface. On the left is a dark sidebar with navigation links: 'Create a resource', 'Home', 'Dashboard', and 'All services'. The main area shows 'All services' with a search bar containing 'logic apps'. Below the search bar, 'Logic Apps' is highlighted in a light blue box.</p>
<p>Create a New Logic App and name it appropriately for this application. e.g. Kelverion-STSK-PortalReturnStatus</p> <p>Add it to the same Resource Group that you have deployed the runbooks too</p>	 <p>The screenshot shows the 'Create new Logic App' wizard. The 'Basics' tab is selected. Under 'Project details', the 'Subscription' is set to 'Development' and the 'Resource group' is 'Kelverion-0345'. Under 'Instance details', the 'Logic App name' is 'Kelverion-0345-PortalReturnStatus'. There are 'Create new' links for both the subscription and resource group.</p>
Open the newly created Logic App and select "Blank Logic App"	 <p>The screenshot shows the Logic Apps Designer interface. The breadcrumb path is 'Dashboard > Logic Apps > Kelverion_0345_0-00_InvokeRunbook > Logic Apps Designer'. The title is 'Logic Apps Designer'. Below, it says 'Choose a template below to create your Logic App.' There are two templates: 'Blank Logic App' with a large plus sign icon, and 'Delete old blobs' with a clock icon.</p>

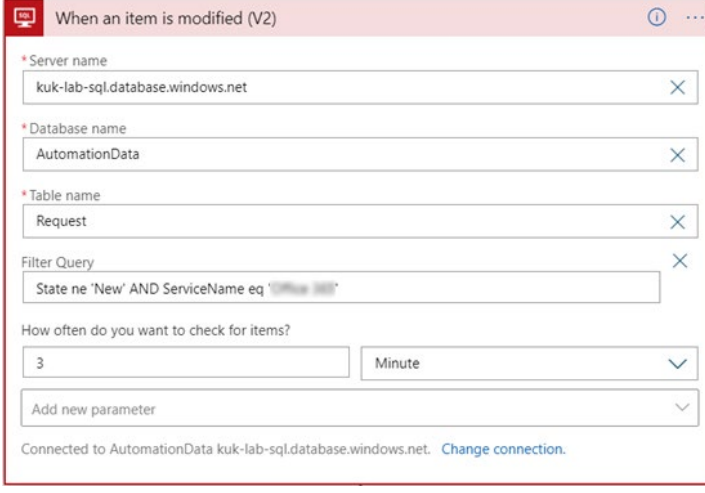
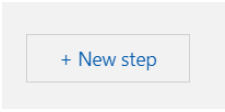
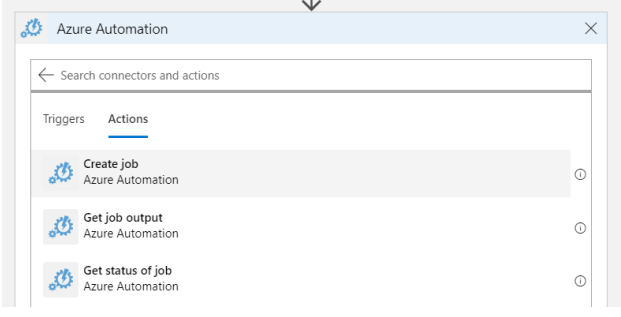
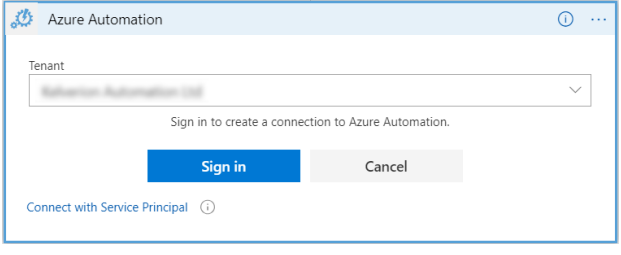
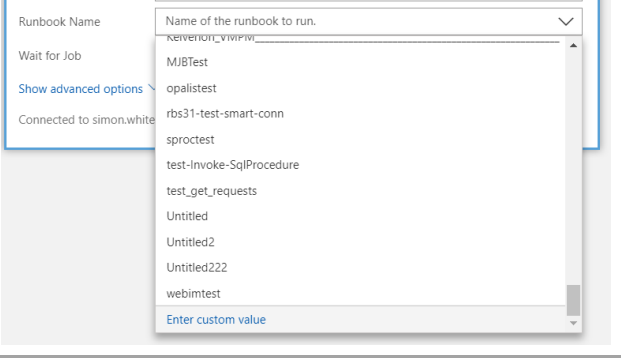
Search for “Sql Server”
and
select the Trigger
“When an item is modified”

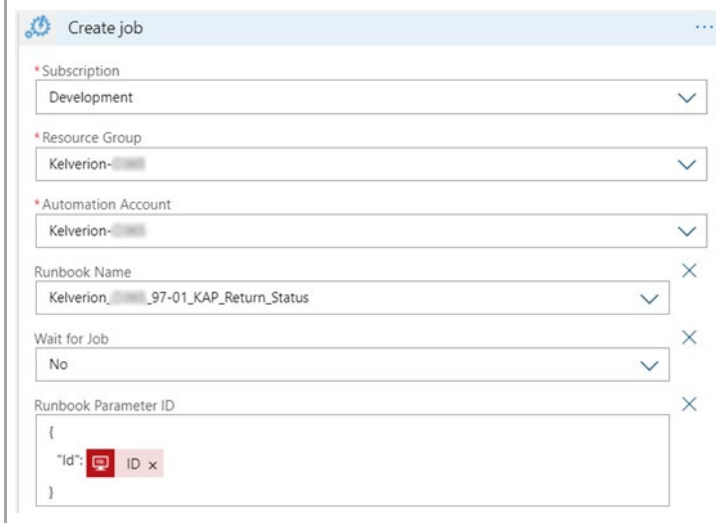

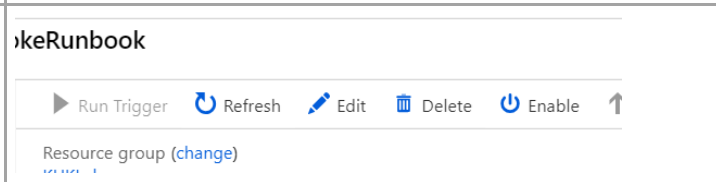
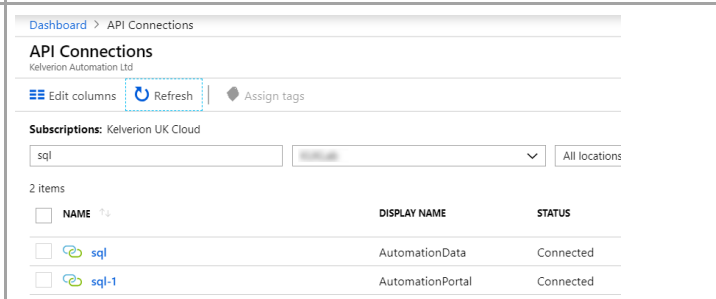


If you do not already have a
connection (API Connection)
then you will need to set one
up for your AutomationData
database.

Enter the appropriate
connection details for the
database.

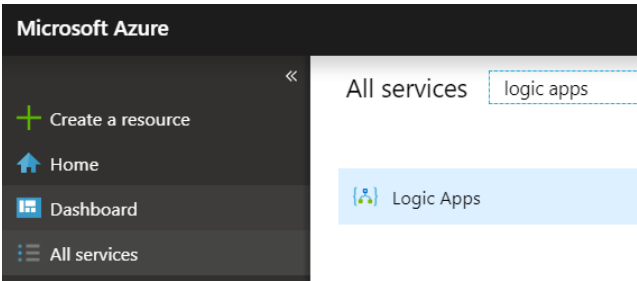
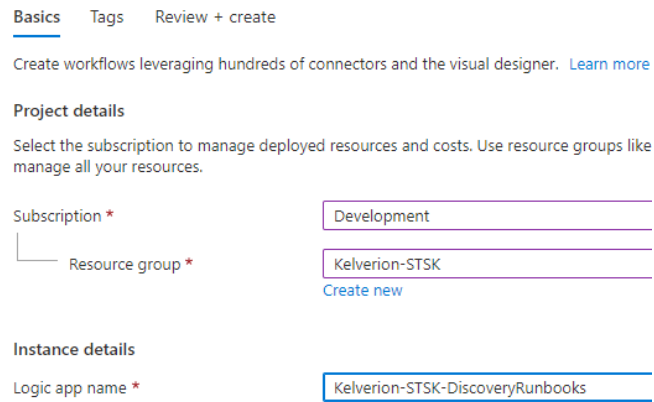
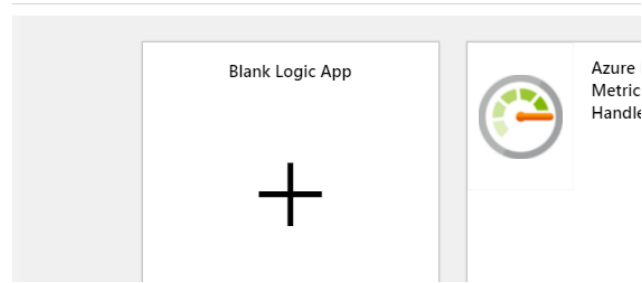



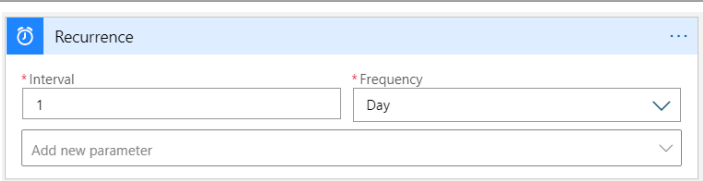
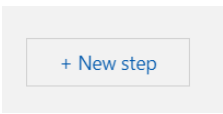
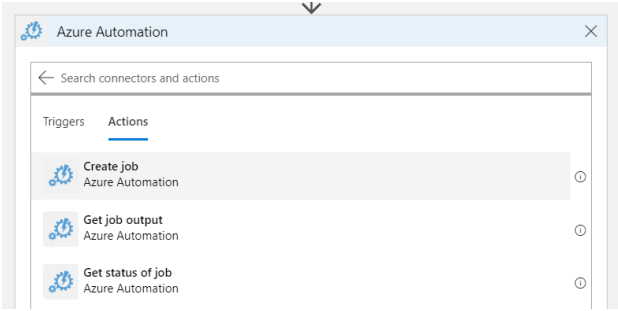
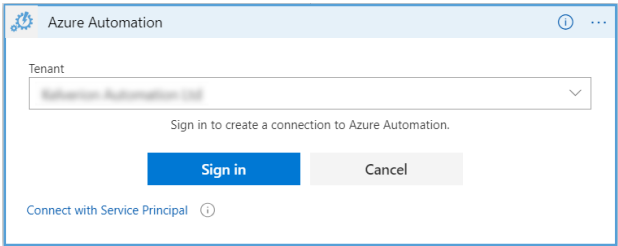
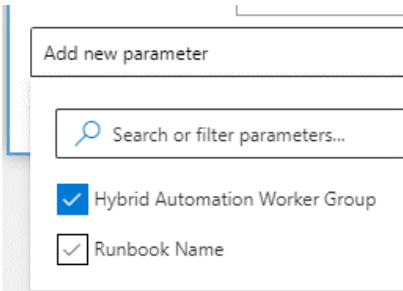
<p>Configure the activity as shown.</p> <p>Table = Request</p> <p>Filter Query: State ne 'New' AND ServiceName eq 'Standard Tasks'</p> <p>Interval = 1 Minute</p>	
<p>Click on New Step to add another activity</p>	
<p>Search for "Azureautomation" and select the Action "Create Job"</p>	
<p>If you have not done so before, you will need to create an API Connection to your tenant. Use the required Azure login details to make the connection.</p>	
<p>Enter the appropriate: Subscription \ Resource Group \ Automation Account</p> <p>For Runbook Name, scroll to the bottom and select "Custom Value"</p>	

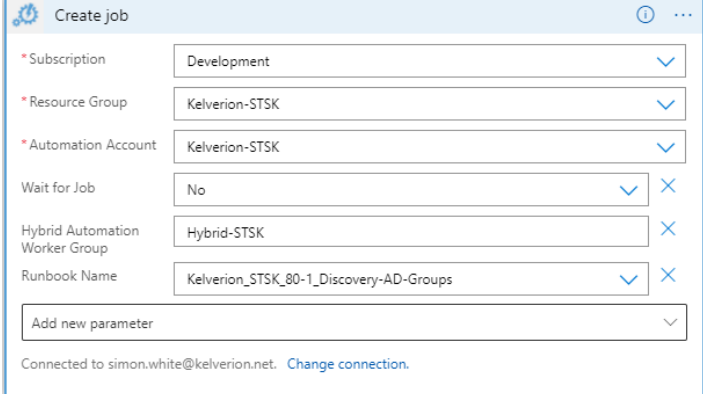
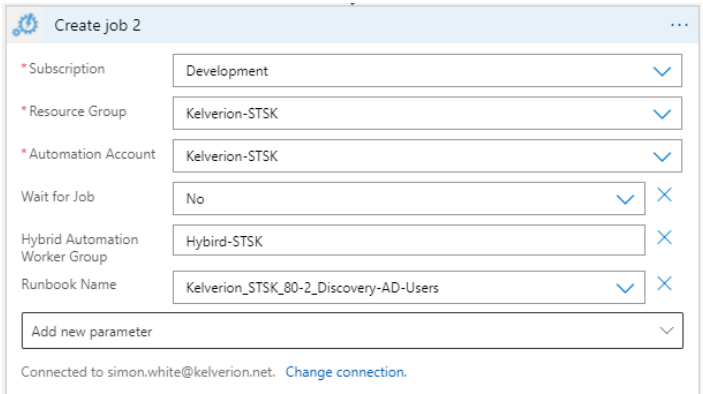
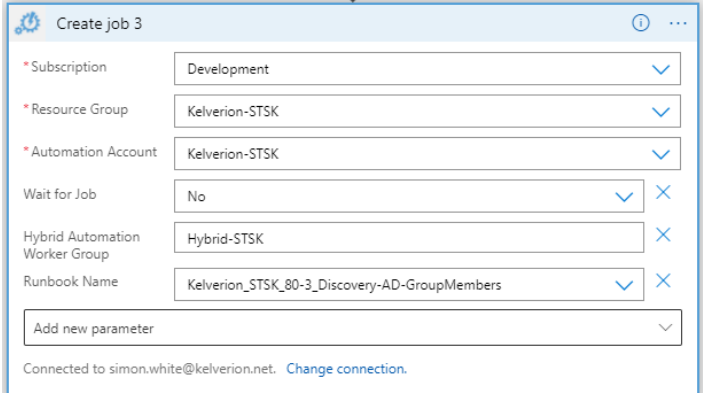
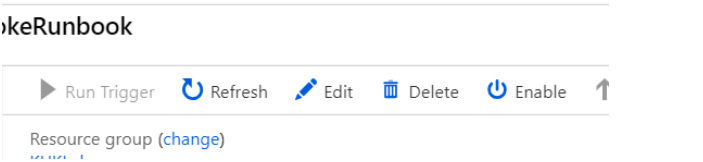
<p>Ensure that the activity is pointing at the correct Runbook:</p> <p>Kelverion_STSK_97-01_KAP_Return_Status</p>																												
<p>Add Runbook Parameters. Where ID is from the List of Items from the previous activity.</p>																												
<p>Ensure the Logic App is active by clicking on Enable</p>																												
<p>If you need to change connection details, you should be able to find your connection information in “API Connections”</p>	 <table><tr><th colspan="3">API Connections</th></tr><tr><td colspan="3">Kelverion Automation Ltd</td></tr><tr><td>Edit columns</td><td>Refresh</td><td>Assign tags</td></tr><tr><td colspan="3">Subscriptions: Kelverion UK Cloud</td></tr><tr><td><input type="text" value="sql"/></td><td><input type="text" value=""/></td><td>All locations</td></tr><tr><td colspan="3">2 items</td></tr><tr><th><input type="checkbox"/> NAME</th><th>DISPLAY NAME</th><th>STATUS</th></tr><tr><td><input type="checkbox"/> sql</td><td>AutomationData</td><td>Connected</td></tr><tr><td><input type="checkbox"/> sql-1</td><td>AutomationPortal</td><td>Connected</td></tr></table>	API Connections			Kelverion Automation Ltd			Edit columns	Refresh	Assign tags	Subscriptions: Kelverion UK Cloud			<input type="text" value="sql"/>	<input type="text" value=""/>	All locations	2 items			<input type="checkbox"/> NAME	DISPLAY NAME	STATUS	<input type="checkbox"/> sql	AutomationData	Connected	<input type="checkbox"/> sql-1	AutomationPortal	Connected
API Connections																												
Kelverion Automation Ltd																												
Edit columns	Refresh	Assign tags																										
Subscriptions: Kelverion UK Cloud																												
<input type="text" value="sql"/>	<input type="text" value=""/>	All locations																										
2 items																												
<input type="checkbox"/> NAME	DISPLAY NAME	STATUS																										
<input type="checkbox"/> sql	AutomationData	Connected																										
<input type="checkbox"/> sql-1	AutomationPortal	Connected																										

3.11.3 Discovery Logic App

This Logic App will call the discovery runbooks on a schedule.

Step	
Login to the Azure Portal and go to "Logic Apps"	
<p>Create a New Logic App and name it appropriately for this application. e.g. Kelverion-STSK-DiscoveryRunbooks</p> <p>Add it to the same Resource Group that you have deployed the runbooks too</p>	
Open the newly created Logic App and select "Blank Logic App"	
Search for "Schedule" and select the Trigger "Recurrence"	

Set the chosen interval. Daily should be enough to cover most device discovery. You may need a more frequent discovery for your environment.	
Click on New Step to add another activity	
Search for "Azure Automation" and select the Action "Create Job"	
If you have not done so before, you will need to create an API Connection to your tenant. Use the required Azure login details to make the connection.	
Enter the appropriate: Subscription \ Resource Group \ Automation Account. You will need to add the parameter 'Hybrid Automation Worker Group' and 'Runbook Name'	

<p>Ensure that the Hybrid Worker is set to your Hybrid Worker name:</p> <p>And the Runbook should be:</p> <p>Kelverion_STSK_80-1_Discovery-AD-Groups</p>	
<p>Repeat the above step for the 2nd Discovery runbook</p> <p>Kelverion_STSK_80-2_Discovery-AD-Users</p>	
<p>Repeat the above step for the 3rd Discovery runbook</p> <p>Kelverion_STSK_80-3_Discovery-AD-GroupMembers</p>	
<p>Ensure the Logic App is active by clicking on Enable</p>	

If you need to change connection details, you should be able to find your connection information in “API Connections”

Dashboard > API Connections

API Connections

Kelverion Automation Ltd

Edit columns

Refresh

Assign tags

Subscriptions: Kelverion UK Cloud

sql

All locations

2 items

<input type="checkbox"/>	NAME	DISPLAY NAME	STATUS
<input type="checkbox"/>	sql	AutomationData	Connected
<input type="checkbox"/>	sql-1	AutomationPortal	Connected

3.12 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 Create User
2. Start the runbook Kolverion_STSK_20-1_Worker_AD-Create-User using the Azure Portal
3. Monitor the runbook to ensure it completes without errors
4. Login to the Hybrid Worker and check for the user account
5. Check the status of the request using the Automation portal
6. Attempt to log in to AD using the details supplied in the automation portal

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