

Kelverion Automation

Joiners Movers Leavers Application for Azure Automation

Deployment Guide

Version 1.3

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
3.2	Kelverion Items Required	4
3.3	Other Products Required.....	4
3.4	Installation Steps.....	4
3.5	Persistent Data Store.....	5
3.6	Configure the Automation Portal.....	5
3.7	Create an Azure user for starting runbooks	5
3.8	Load Integration Modules	6
3.9	Configure your Automation account using the Runbook Studio.....	6
3.9.1	<i>Design Time (Smart Connections).....</i>	<i>6</i>
3.9.2	<i>Azure Variables.....</i>	<i>8</i>
3.9.3	<i>Azure Credentials.....</i>	<i>9</i>
3.9.4	<i>Azure Runtime connections.....</i>	<i>11</i>
3.10	Import Runbooks using the Runbook Studio	13
3.11	Runbook Customisation.....	15
3.11.1	<i>High Level Overview.....</i>	<i>15</i>
3.12	Runbook Process Flow	16
3.12.1	<i>Data Storage (PDS Design).....</i>	<i>16</i>
3.12.2	<i>Gathering the Data</i>	<i>17</i>
3.12.3	<i>Transposing the Data</i>	<i>18</i>
3.12.4	<i>Using the Data.....</i>	<i>19</i>
3.12.5	<i>Returning the Results.....</i>	<i>20</i>
3.13	Logic Apps - Scheduling runbook execution.....	21
3.13.1	<i>Gather Request Logic App.....</i>	<i>21</i>
3.13.2	<i>Return Status Logic App.....</i>	<i>25</i>
3.13.3	<i>Discovery Logic App</i>	<i>28</i>
3.14	Testing.....	32

2 Overview

The processing of new joiners to a business is a hugely complex and time-consuming process for Service Desks and IT departments. Processing leavers is an equally complex task but for security and commercial sensitivity is arguably even more time critical.

Businesses would like to give team leaders and managers access to process their own Joiners and Leavers 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 Joiners Movers Leavers (JML) Application for Azure Automation.

The JML Application for Azure Automation provides an easy to use interface using the Kolverion Automation Portal, users are provided the rights to access portal services via group membership, but these rights are isolated from the IT infrastructure like Active Directory and Office 365, so it's simple to provide portal facilities without providing rights to directly manage users and licenses.

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

All the data that drives the requests in the automation portal is discovered and maintained by the Azure Automation runbooks, so the automation portal is always in sync with the IT management experience.

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 JML application package contains the following elements:

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

3.2 Keverion Items Required

The application requires the following Keverion products:

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

If you do not already have Keverion Integration Modules, Keverion Automation Portal, or the Keverion 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.

3.3 Other Products Required

The two following Microsoft PowerShell modules are required:

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

These modules are available from the PowerShell Gallery

3.4 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. Create Managed System 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

3.5 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_JML.sql*) from the package to Create the database tables and views.

3.6 Configure the Automation Portal

Create a new Service to store the portal offerings. Import the offerings Offering_JML_XXXXXXX.export files into the service.

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

- AzureAD.AssignedGroups
- AzureAD.AvailableGroups
- AzureAD.Groups
- AzureAD.v_AvailableSKUs
- AzureAD.v_enabledUsers
- AzureAD.v_CurrentUsers
- StandardTasks-GetAD
- StandardTasks-GetADGroup
- StandardTasks-GetOU

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

3.7 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 an "Automation Operator" 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 Logic App so make a note of these values.

3.8 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
- MSOnline
- MicrosoftTeams


3.9 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.

3.9.1 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.

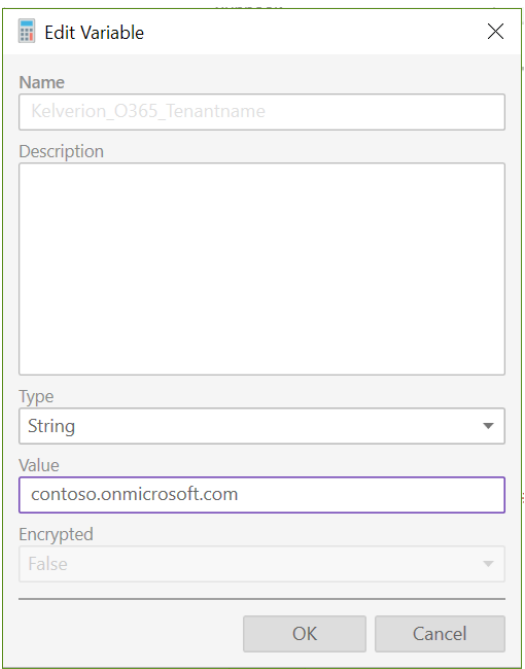
Create the following smart connections within the Runbook Studio.

<p>AutomationData</p>	<div><div><div> Edit Smart Connection</div><div><div><div>×</div></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>OK</div><div>Cancel</div></div></div>
-----------------------	---

3.9.2 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.

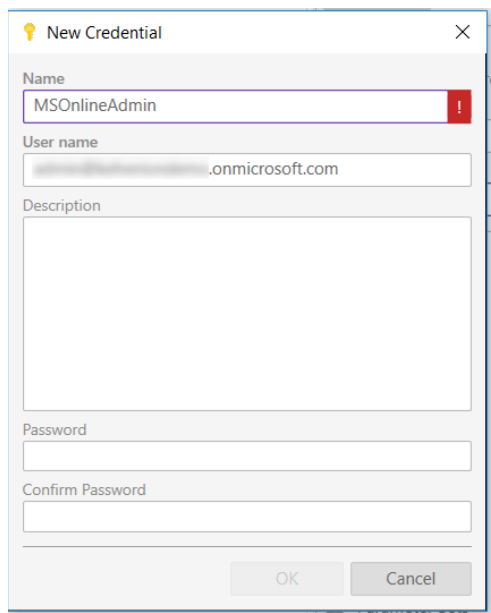
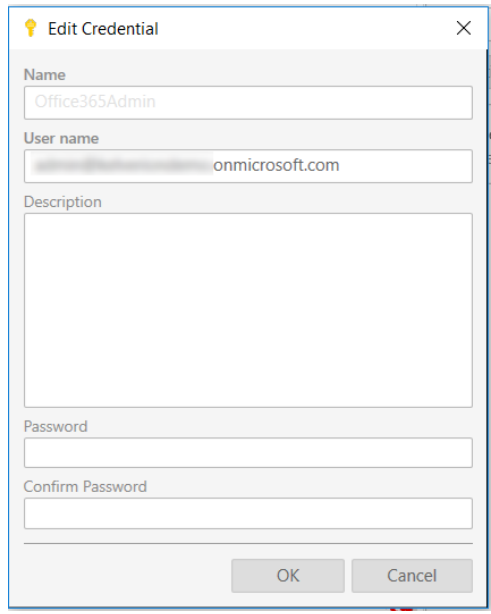
Create the following variables in the Automation Account using the Runbook Studio.

Kelverion_O365_Tenantname	This is the name of your office 365 tenant e.g. contoso.onmicrosoft.com	
SuscriptionID	The ID of the subscription that the runbooks are imported into	
TenantId	The tenant ID of your Azure subscription that the runbook application is imported into	

3.9.3 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.

MSONlineAdmin	This credential will be used to configure the MS Online accounts. This will be an Office 365 administrator account.	
Office365Admin	This credential will be used to configure the Office 365 users \ groups. Will need to be an Office 365 Administrator	

EmailSenderAccount	This credential will be used to email temporary passwords for the “Reset Password” offering.	<div><div>⚡ Edit Credential</div><div><div>Name</div><div>EmailSenderAccount</div></div><div><div>User name</div><div>@kelverion.com</div></div><div><div>Description</div><div></div></div><div><div>Password</div><div></div></div><div><div>Confirm Password</div><div></div></div><div><div>OK</div><div>Cancel</div></div></div>
--------------------	--	---

3.9.4 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	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.	<div><div>Edit Connection</div><div><div>Name</div><div>AutomationPortal</div></div><div><div>Description</div><div></div></div><div><div>Connection Type</div><div>Kolverion.SqlServer</div></div><div><div>ServerName ⓘ</div><div>localhost\sqlservr\KolverionAutomationPortalDatabase</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>0</div></div><div><div>OK</div><div>Cancel</div></div></div>
------------------	---	---

3.10 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_JML_30-1_New_Joiner	Parent runbook for the New Joiner process
Kelverion_JML_30-2_Move_User	Parent runbook for the Move User process
Kelverion_JML_30-3_Leaver	Parent runbook for the Leaver process
Kelverion_JML_90_01_KAP_Get_Request	Gathers the request data from the Kelverion Automation Portal. Requires a Logic App to trigger
Kelverion_JML_95_01_KAP_ConvertToRequestData	Converts the gathered data to the correct format for the runbooks
Kelverion_JML_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
Kelverion_0365_10-4_Worker-Disable-User	0365 Child runbook
Kelverion_0365_10-5_Worker-Assign-License	0365 Child runbook
Kelverion_0365_10-6_Worker-Unassign-License	0365 Child runbook
Kelverion_0365_10-12_Add-User-ToGroups	0365 Child runbook
Kelverion_0365_10-13_Remove-User-FromGroups	0365 Child runbook
Kelverion_0365_10-14_Enable-Litigation	0365 Child runbook
Kelverion_0365_10-16_Create-User	0365 Child runbook
Kelverion_0365_30-2_PDSDData-Users	0365 PDS discovery runbook
Kelverion_0365_30-3_PDSDData-SKUs	0365 PDS discovery runbook
Kelverion_0365_30-4_PDSDData-GroupMembers	0365 PDS discovery runbook
Kelverion_0365_30-5_PDSDData-MSOLUser	0365 PDS discovery runbook
Kelverion_0365_30-8_PDSDData-Mailbox	0365 PDS discovery runbook

Kelverion_STSK_20-1_Worker_AD-Create-User	STSK Child runbook
Kelverion_STSK_23-1_Worker_AD-Enable-Disable	STSK Child runbook
Kelverion_STSK_24-0_Worker_AD-Remove-Group-Members	STSK Child runbook
Kelverion_STSK_24-1_Worker_AD-Remove-Group-Members	STSK Child runbook
Kelverion_STSK_25-0_Worker_AD-Add-User-To-Groups	STSK Child runbook
Kelverion_STSK_25-1_Worker_AD-Add-User-To-Groups	STSK Child runbook
Kelverion_STSK_40-1_Worker_Add-User-Mailbox	STSK Child runbook
Kelverion_STSK_80-1_Discovery-AD-Groups	STSK PDS discovery runbook
Kelverion_STSK_80-2_Discovery-AD-Users	STSK PDS discovery runbook
Kelverion_STSK_80-3_Discovery-AD-GroupMembers	STSK PDS discovery runbook

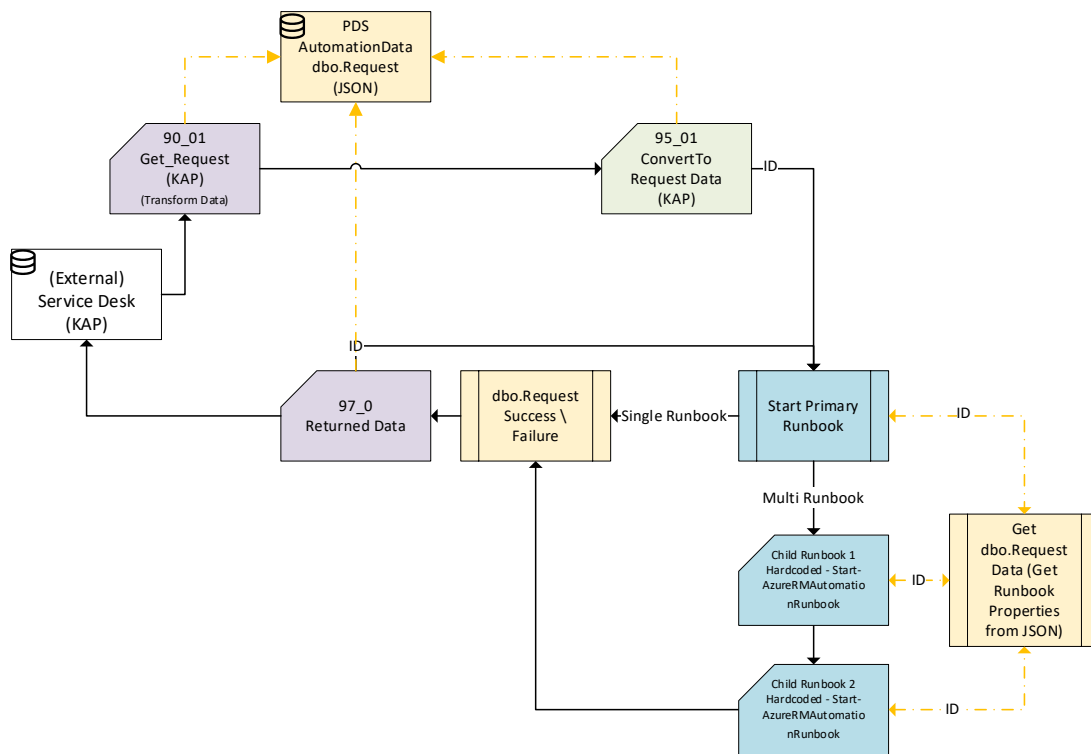
3.11 Runbook Customisation

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).

3.11.1 High Level Overview

The main process flow is as follows:



3.12 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.

3.12.1 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

3.12.1.1 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

3.12.2 Gathering the Data

The base applications come 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 will have a 90-0x_XXX_Get_Request runbook that can be configured for the required Service Desk portal.

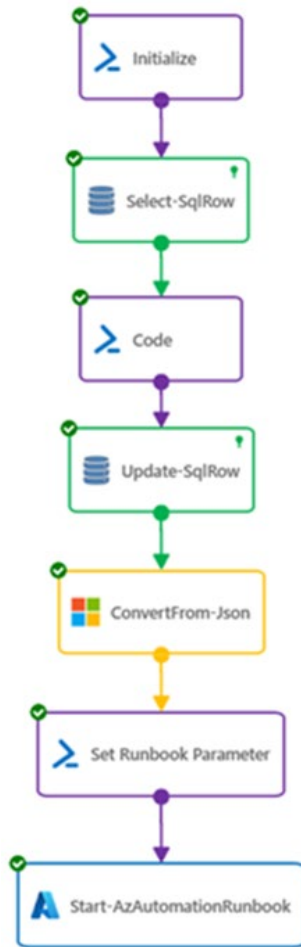
This runbook has a code block that will transpose the Automation Portal request into JSON format and store it into the PDS dbo.Request table. Other service desks will require different activities to store the code in the corresponding JSON format.

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...

3.12.3 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 (Kolverion_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 (e.g. Standard Tasks Application).

The example runbook here is from the Office 365 application: Kolverion_0365_95-01_ConvertToRequestData

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 = 'Kolverion_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-AzAutomationRunbook activity.

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

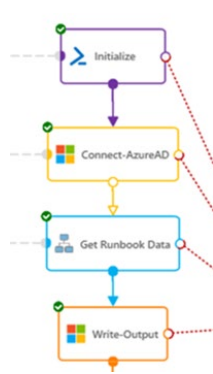
RunbookOwner	Data	ServiceName	OfferingName	ExternalId	State	OutputData
Kolverion_STSK_21-1_Work...	{ "First Name": "Bob", ...	Business User...	New Joiner	306	Complete	{ "Runbook": "Kolverion_JML_30-...
Kolverion_0365_10-4_Worke...	{ "Reason for Blocking ...	Office 365	Disable Login	305	Complete	{ "ObjectId": "240ba31e-6a1a-46d...
Kolverion_0365_10-2_Worke...	{ "User": { "..."	Office 365	Enable User	304	Complete	{ "ObjectId": "240ba31e-6a1a-46d...

3.12.4 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

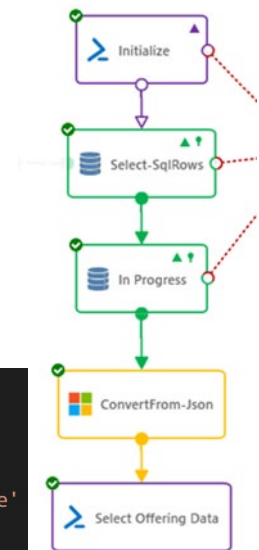
N.B. More complex array outputs may require a PowerShell code block to process them, rather than a Write-Output.

'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
  }
}
  
```

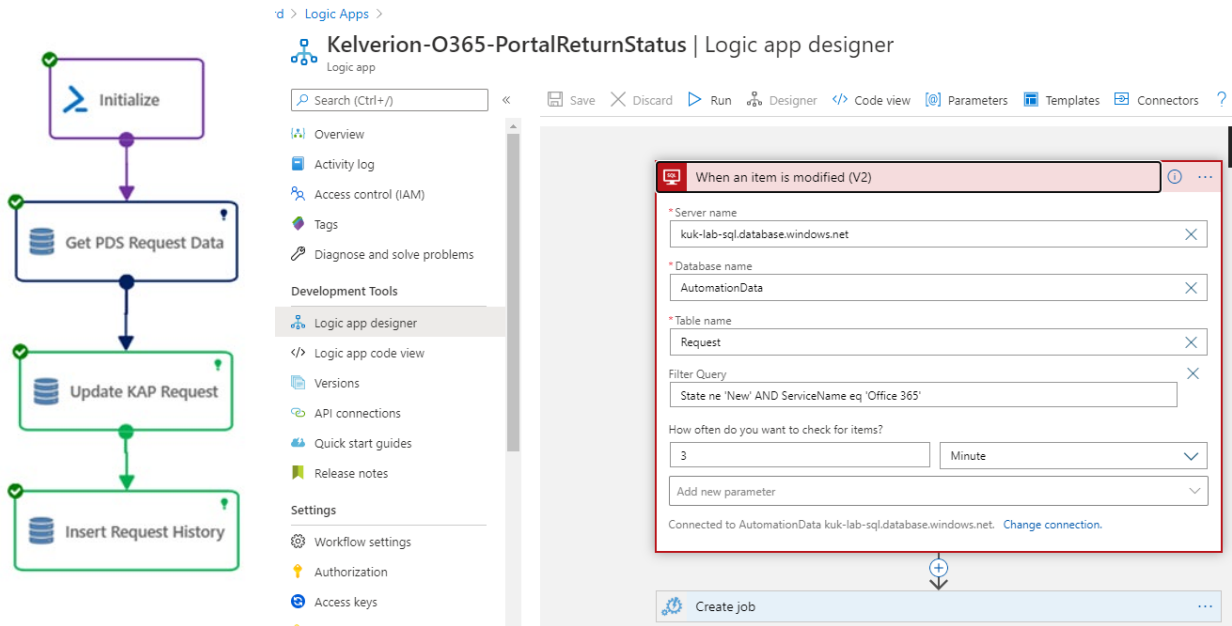
Annotations in the image point to:

- Service Desk**: Points to the outer function name 'Delete User'.
- Offering Name**: Points to the 'Runbook' property value.
- Worker Runbook**: Points to the inner function 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.

3.12.5 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.

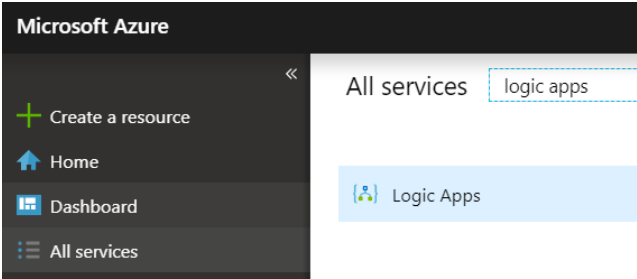
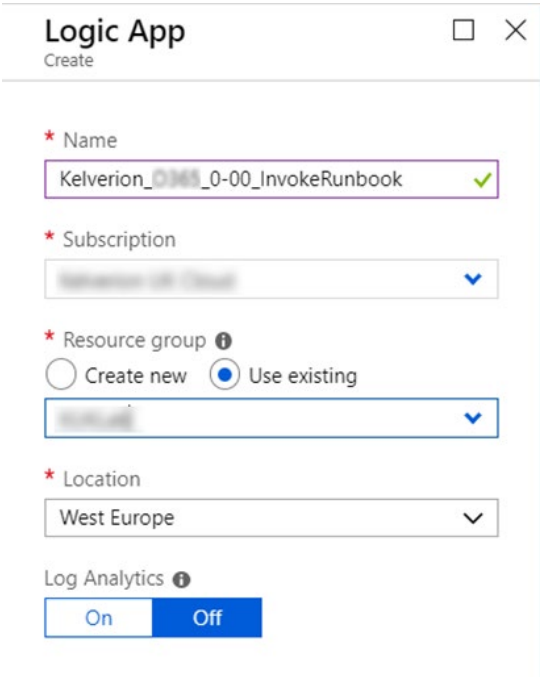
3.13 Logic Apps - Scheduling runbook execution

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.

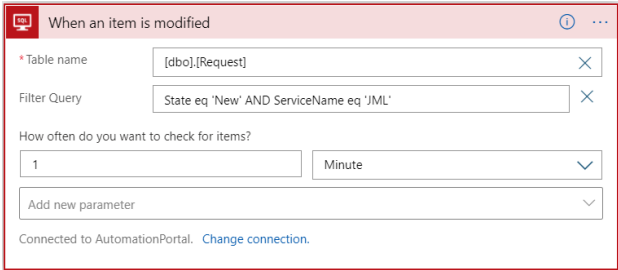
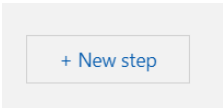
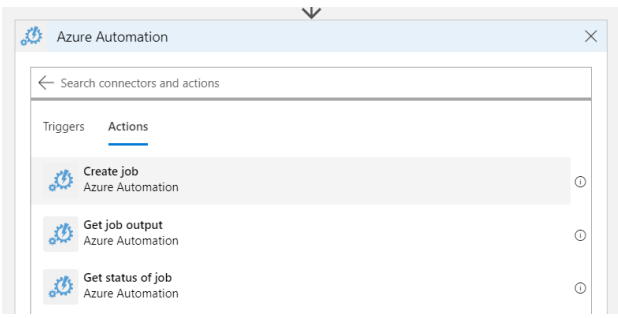
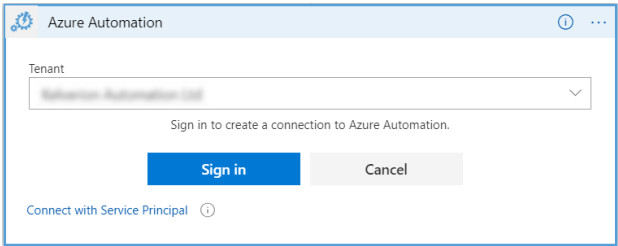
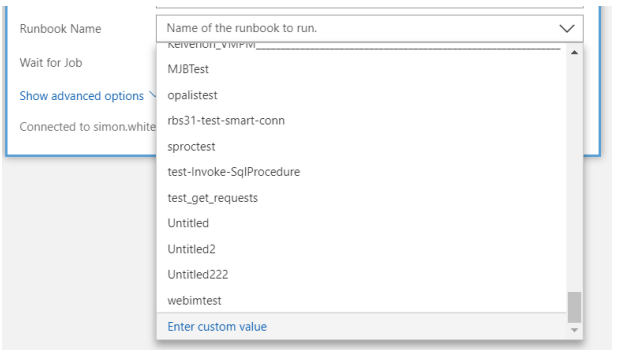
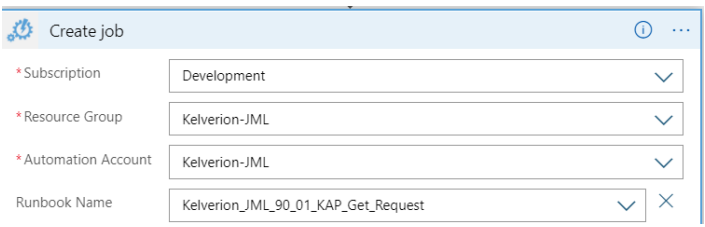
Two Logic Apps are required. One to gather the information from the required Service Desk application (in this case the Automation Portal) and one for returning the status of the runbook to the Service Desk application.


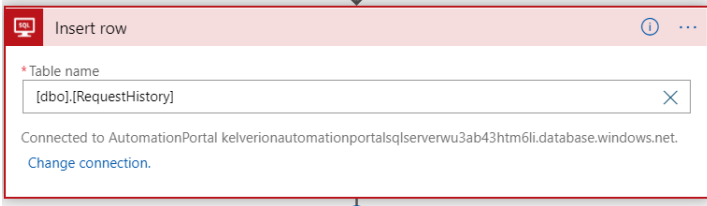

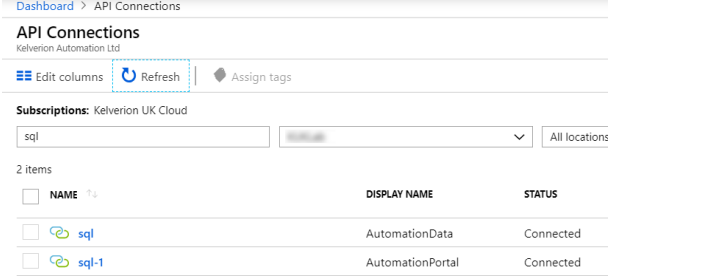
3.13.1 Gather Request Logic App

This Logic App will call the runbook Kolverion_JML_90-01_KAP_Get_Request.

Step	
Login to the Azure Portal and go to "Logic Apps"	
Create a New Logic App and name it appropriately for this application. e.g. Kolverion_JML_0-00_InvokeRunbook Add it to the same Resource Group that you have deployed the runbooks too	

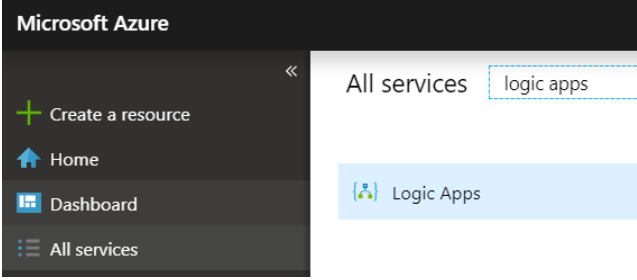
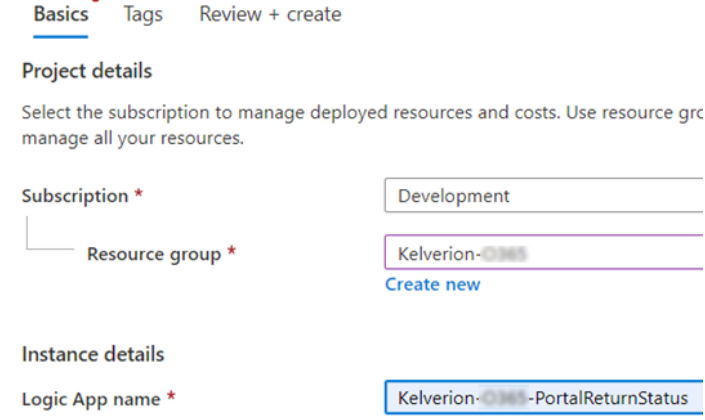
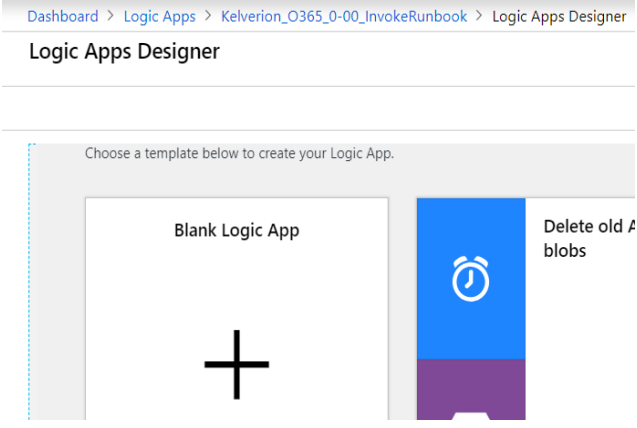
<p>Open the newly created Logic App and select “Blank Logic App”</p>	<div><div>Dashboard > Logic Apps > Kelverion_O365_0-00_InvokeRunbook > Logic Apps Designer</div><div>Logic Apps Designer</div><div><div>Choose a template below to create your Logic App.</div><div><div>Blank Logic App</div><div><div>+</div></div></div><div><div><div>Delete old blobs</div><div><div></div></div></div></div></div></div>									
<p>Search for “Sql Server” and select the Trigger “When an item is modified”</p>	<div><div>sql server</div><div><div>For You</div><div>All</div><div>Built-ins</div><div>Connectors</div><div>Enterprise</div><div>Custom</div></div><div><div><div>SQL Server</div><div>SQL Server</div></div></div><div><div>Triggers</div><div>Actions</div></div><div><div><div>When an item is created SQL Server</div><div>When an item is modified SQL Server</div></div></div></div>									
<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>	<div><div>When an item is modified</div><div><div>* Connection Name</div><div>AutomationPortal</div></div><div><div>* SQL Server Name</div><table><thead><tr><th>Name</th><th>Resource Group</th><th>Location</th></tr></thead><tbody><tr><td>AutomationPortalDB</td><td>AutomationPortalDB</td><td>westeurope</td></tr><tr><td>AutomationPortalDB</td><td>AutomationPortalDB</td><td>westeurope</td></tr></tbody></table></div><div><div>* Username</div><div>Username credential</div></div><div><div>* Password</div><div>Password credential</div></div><div><div><input type="checkbox"/> Connect via on-premise data gateway ⓘ</div></div><div><div>Create</div><div>Cancel</div></div><div><div>Manually enter connection information</div></div></div>	Name	Resource Group	Location	AutomationPortalDB	AutomationPortalDB	westeurope	AutomationPortalDB	AutomationPortalDB	westeurope
Name	Resource Group	Location								
AutomationPortalDB	AutomationPortalDB	westeurope								
AutomationPortalDB	AutomationPortalDB	westeurope								

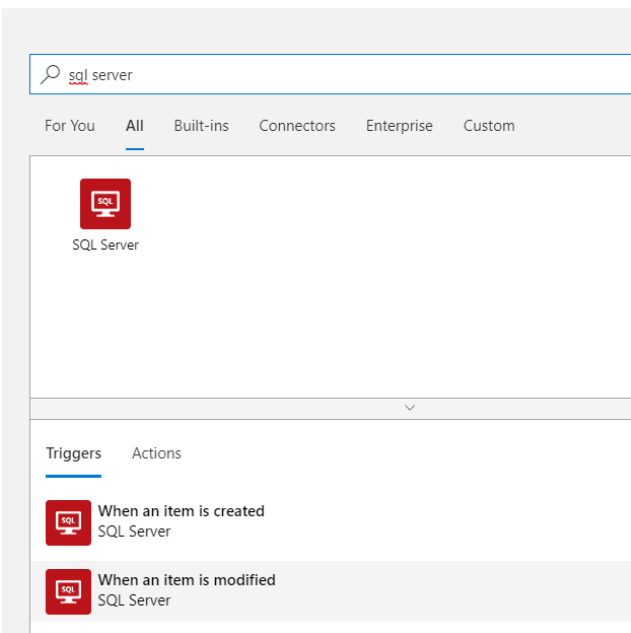
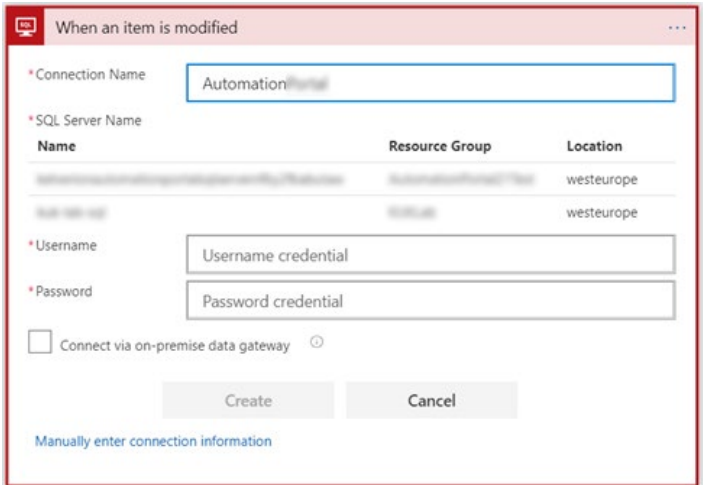
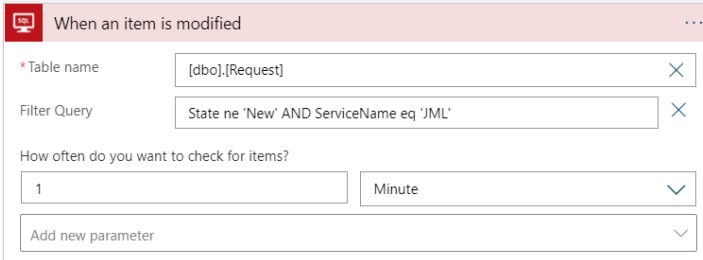
<p>Configure the activity as shown. Table = Request</p> <p>Filter Query: State eq 'Approved' AND ServiceName eq 'JML'</p> <p>Interval = 1 Minute</p>	
<p>Click on New Step to add another activity</p>	
<p>Search for "Azureautomation" and select the Action "Create Job"</p>	
<p>If you have not done so before, 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: Kelverion_JML_90-01_KAP_Get_Request</p>	

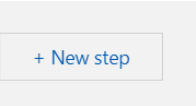
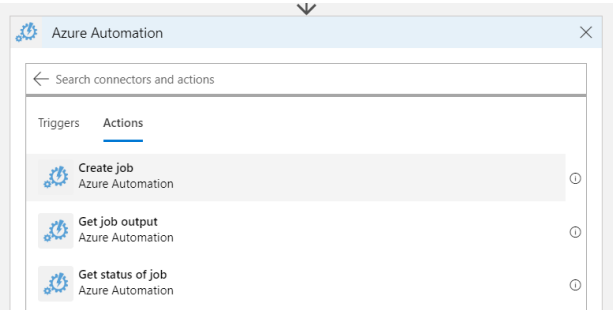
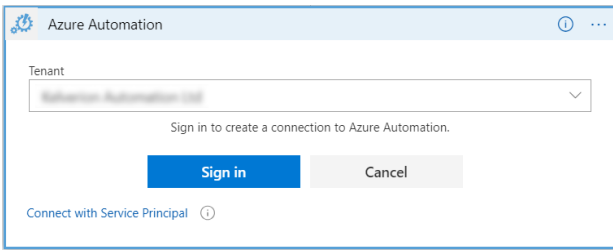
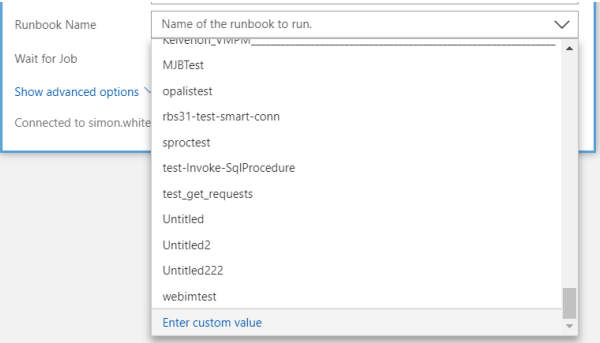
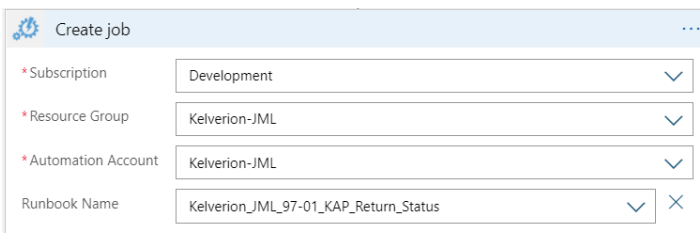

<p>Add Runbook Parameters. Where ID is from the List of Items from the previous activity</p>	<p>Runbook Parameter ID  ID x</p>
<p>Automation Portal Only: Add a new activity to write a line to the RequestHistory table</p>	
<p>Ensure the Logic App is active by clicking on Enable</p>	
<p>If you need to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

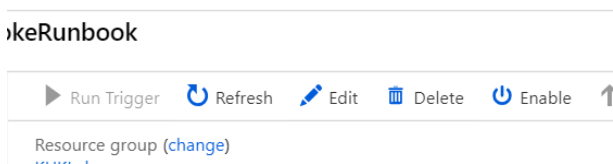
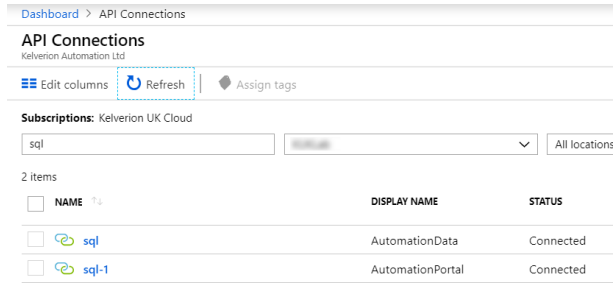
3.13.2 Return Status Logic App

This Logic App will call the runbook Kelverion_JML_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-JML-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' form. At the top are tabs for 'Basics', 'Tags', and 'Review + create'. Under 'Project details', there's a description and fields for 'Subscription' (set to 'Development') and 'Resource group' (set to 'Kelverion-O365'). Under 'Instance details', the 'Logic App name' field is filled with 'Kelverion-O365-PortalReturnStatus'.</p>
Open the newly created Logic App and select "Blank Logic App"	 <p>The screenshot shows the 'Logic Apps Designer' interface. The breadcrumb trail at the top reads: 'Dashboard > Logic Apps > Kelverion_O365_0-00_InvokeRunbook > Logic Apps Designer'. Below the title 'Logic Apps Designer', there's a section 'Choose a template below to create your Logic App.' with two options: 'Blank Logic App' (represented by a large plus sign) and 'Delete old blobs' (represented by a clock icon).</p>

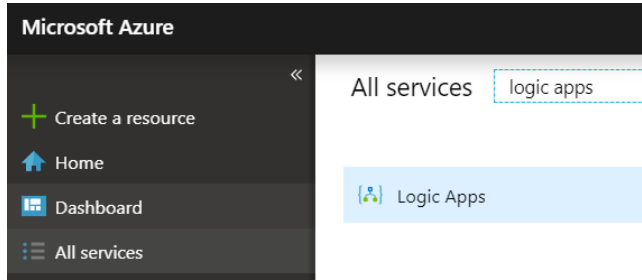
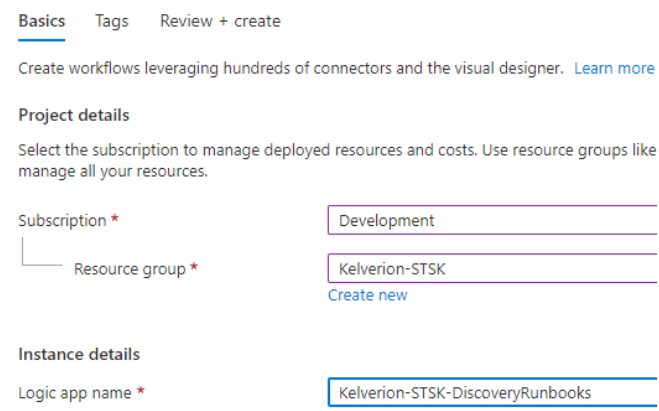
<p>Search for “Sql Server” and select the Trigger “When an item is modified”</p>	
<p>If you do not already have a connection (API Connection) then you will need to set one up for your AutomationData database.</p> <p>Enter the appropriate connection details for the database.</p>	
<p>Configure the activity as shown.</p> <p>Table = Request</p> <p>Filter Query: State ne 'New' AND ServiceName eq 'JML'</p> <p>Interval = 1 Minute</p>	

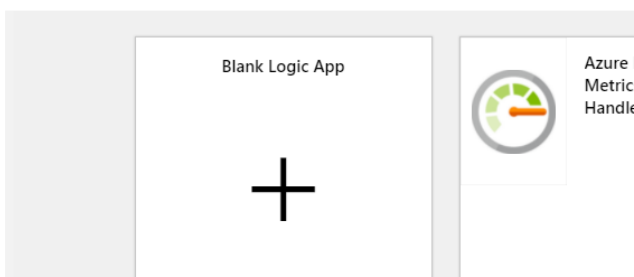




Click on New Step to add another activity	
Search for “Azureautomation” and select the Action “Create Job”	
If you have not done so before, you will need to create an API Connection to your tenant. Use the required Azure login details to make the connection.	
Enter the appropriate: Subscription \ Resource Group \ Automation Account For Runbook Name, scroll to the bottom and select “Custom Value”	
Ensure that the activity is pointing at the correct Runbook: Kelverion_JML_97-01_KAP_Return_Status	
Add Runbook Parameters. Where ID is from the List of Items from the previous activity.	

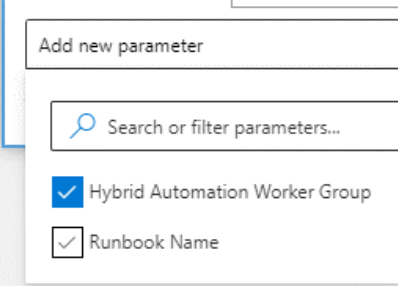
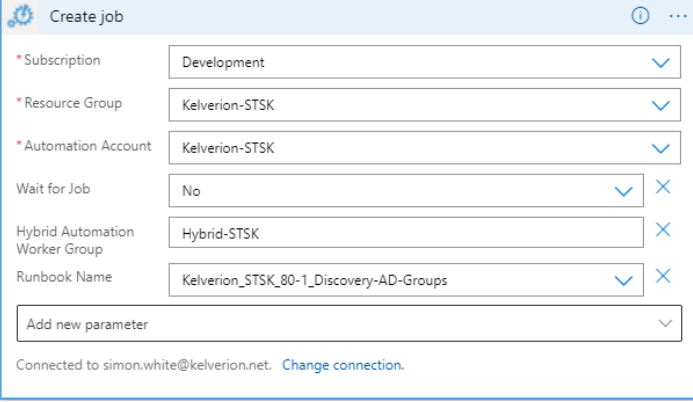
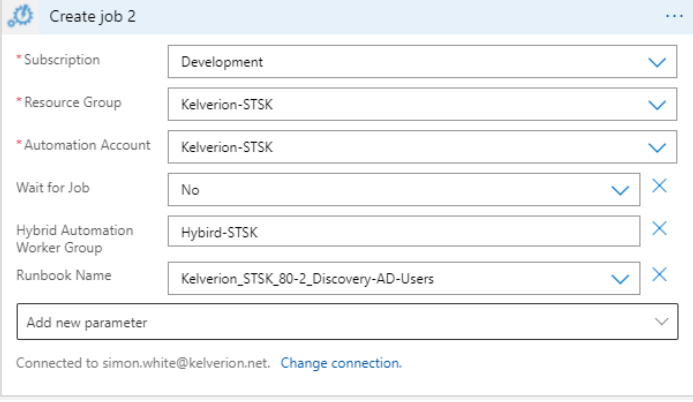
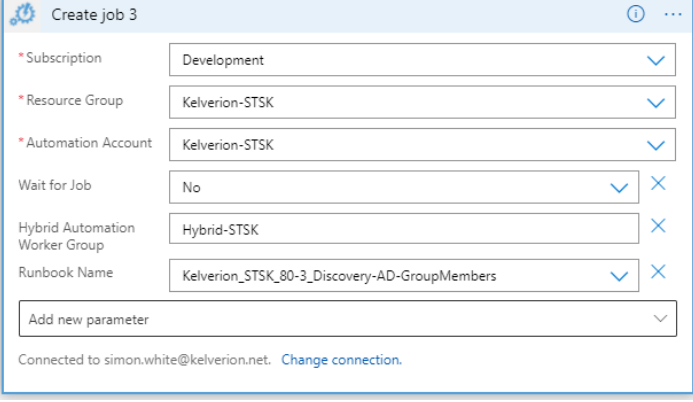
Ensure the Logic App is active by clicking on Enable	
If you need to change connection details, you should be able to find your connection information in “API Connections”	

3.13.3 Discovery Logic App

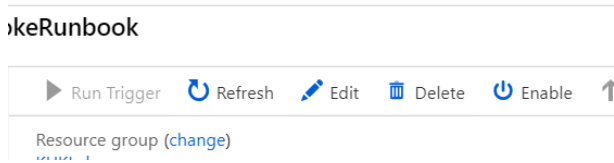
This Logic App will call the discovery runbooks on a schedule.

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”	<div>Dashboard > Microsoft.EmptyWorkflow > Kelverion-STSK-DiscoveryRunbooks ></div> <div>Logic Apps Designer ...</div> <div></div>
Search for “Schedule” and select the Trigger “Recurrence”	<div>Triggers Actions</div> <div> Recurrence Schedule</div>
Set the chosen interval. Daily should be enough to cover most device discovery. You may need a more frequent discovery for your environment.	<div>Recurrence</div> <div>* Interval <input type="text" value="1"/> * Frequency <input type="text" value="Day"/></div> <div>Add new parameter</div>
Click on New Step to add another activity	<div>+ New step</div>
Search for “Azure Automation” and select the Action “Create Job”	<div>Azure Automation</div> <div>Search connectors and actions</div> <div>Triggers Actions</div> <div> Create job Azure Automation</div> <div> Get job output Azure Automation</div> <div> Get status of job Azure Automation</div>
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.	<div>Azure Automation</div> <div>Tenant <input type="text" value="Kelverion Automation Ltd"/></div> <div>Sign in to create a connection to Azure Automation.</div> <div><input type="button" value="Sign in"/> <input type="button" value="Cancel"/></div> <div>Connect with Service Principal ⓘ</div>

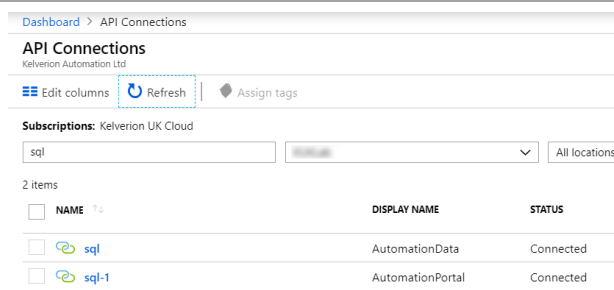
<p>Enter the appropriate: Subscription \ Resource Group \ Automation Account. You will need to add the parameter 'Hybrid Automation Worker Group' and 'Runbook Name'</p>	
<p>Ensure that the Hybrid Worker is set to your Hybrid Worker name:</p> <p>And the Runbook should be: Kelverion_STSK_80-1_Discovery-AD-Groups</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>	

Ensure the Logic App is active by clicking on **Enable**



The screenshot shows the 'Runbook' interface for a Logic App. At the top, there is a horizontal bar with several icons and labels: a play button for 'Run Trigger', a circular arrow for 'Refresh', a pencil for 'Edit', a trash can for 'Delete', a power button for 'Enable', and an upward arrow. Below this bar, the text 'Resource group (change)' is displayed with a link to 'change'.

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



The screenshot shows the 'API Connections' page in the Azure portal. The page has a breadcrumb 'Dashboard > API Connections'. The main heading is 'API Connections' with the subtitle 'Kelverion Automation Ltd'. Below this, there are three buttons: 'Edit columns', 'Refresh', and 'Assign tags'. The 'Refresh' button is highlighted with a dashed blue box. Below the buttons, it says 'Subscriptions: Kelverion UK Cloud'. There is a search bar with 'sql' entered and a dropdown menu showing 'All locations'. Below this, it says '2 items'. There is a table with three columns: 'NAME', 'DISPLAY NAME', and 'STATUS'. The table has two rows: one for 'sql' with display name 'AutomationData' and status 'Connected', and one for 'sql-1' with display name 'AutomationPortal' and status 'Connected'.

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

3.14 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_0365_10-16_Create-User using the Azure Portal
3. Monitor the runbook to ensure it completes without errors
4. Login to Office 365 Admin Center and check for the user account
5. Check the status of the request using the Automation portal
6. Attempt to log in to office 365 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