Kelverion

INTEGRATION MODULE FOR AZURE INFRASTRUCTURE

For Kelverion Runbook Studio and Azure Automation

USER GUIDE

Version 1.1

Microsoft Azure

Certified

Kelverion Integration Module for Azure Infrastructure

Published: February 2022

Feedback

Send suggestions and comments about this document to support@kelverion.com

Contents

Getting Started

The following sections outline how to deploy and configure the Kelverion Integration Module for Azure Infrastructure.

System Requirements

The Integration Module for Azure Infrastructure requires the following software to be installed and configured prior to implementing the integration. For more information on installing Kelverion Runbook Studio, please refer to the Kelverion Runbook Studio User Guide.

- Kelverion Runbook Studio 4.8
- Microsoft .NET Framework 4.7.2

Installing the Integration Module

The easiest way to install and deploy the Integration Module for Azure Infrastructure is from the PowerShell Gallery, but you can also download the module from Kelverion and perform the steps manually.

You must install and deploy the Integration Module to each Azure Automation Account and Hybrid Worker host system that you plan to use to run your runbooks. You must also install the Integration Module on any Runbook Studio host systems that you will be using to build and manage your runbooks.

Using the PowerShell Gallery

Using the commands in the **PowerShellGet** module, you can download the Kelverion Integration Module for Azure Infrastructure from the PowerShell Gallery and install it on your local computer. You can also deploy the module directly from the PowerShell Gallery to any of your Azure Automation Accounts.

Install the Integration Module on your local computer or Hybrid Worker:

1. Confirm that the latest PowerShellGet module is installed.
2. Start a PowerShell window as Administrator and run the command:

```
Install-Module -Name Kelverion.Azure.Infrastructure -Scope
AllUsers
```

Upload the Integration Module to an Azure Automation Account:

1. Go to the PowerShell Gallery.
2. Click the **Azure Automation** tab.
3. Click **Deploy to Azure Automation**. You will be directed to Microsoft Azure.
4. Select the **Automation Account** that you want to deploy the module to.
5. Click **OK**.

Manual Installation

Alternatively, you can download the Integration Module package from Kelverion and deploy it manually to your local computer, hybrid workers and Automation Accounts.

The download package from Kelverion includes a **.zip** file containing the Integration Module as well as the User Guide and Release Notes. The following instructions assume that you have unzipped the download package and have access to the **.zip** file containing the Integration Module.

Important: When installing the Integration Module on a Hybrid Worker, you must use a location that is accessible to all users of the computer.

Install the Integration Module on your local computer or any Hybrid Workers:

1. Copy the **Kelverion.Azure.Infrastructure.zip** file to your local computer.
2. Right-click on the file and select **Properties**.
3. Click the **General** tab. If necessary, click **Unblock**, and then click **OK**.
4. Unzip the **Kelverion.Azure.Infrastructure.zip** file.
5. Copy the **Kelverion.Azure.Infrastructure** folder to a location in the `%PsModulePath%` path.

Upload the Integration Module to an Azure Automation Account:

1. Sign into Microsoft Azure.
2. Open the Automation Account that you want to upload the module to.
3. Click **Modules** under Shared Resources. The list of installed modules is displayed.
4. Click **Add a module** at the top of the list.
5. In the **Upload File** box, select the **Kelverion.Azure.Infrastructure.zip** file that you downloaded.
6. Click **OK**. Importing the module may take several minutes.

Licensing the Integration Module

Licenses for Kelverion Integration Modules are managed and deployed using the *Kelverion Runbook Studio* and *Automation Connection Assets*.

Important: Entitlements will not display until after the Integration Module has been installed on the Runbook Studio computer.

Register an Integration Module license with Runbook Studio:

1. Open **Kelverion Runbook Studio**.
2. In the **File** tab, click **About**.
3. Click **License Information**.
4. Click the **Integration Modules** tab, and then click **Add License.**
5. Select the integration module license file (.kaml) and click **Open**.
6. You should see your entitlements displayed in the list.
7. Click **OK**.

Create a Connection Asset with a license key and upload it to Azure:

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** field, enter a name to identify the connection.
6. In the **Connection Type** field, select the desired connection type.
7. Enter the appropriate connection information in the provided fields.
8. Click **OK**.

Update all Connection Assets license keys and upload them to Azure:

1. On the ribbon, click the **Home** tab, and then click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the Explorer panel, click the **Azure (Online)** group.
4. Right-click the Azure Automation Account that contains the connection assets you want to update, and then click **Update License Keys**. A summary is displayed.

Connecting to Azure

To connect the Kelverion Integration Module for Azure Infrastructure to an Azure subscription, you will need to:

1. Create an Azure AD application.
2. Assign the application to a role.

Please follow the steps described in the link below to create an app and assign it to a role. You will need to record the subscription ID, tenant ID, application ID and key value to configure an IM connection afterwards.

https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal

Once you obtained all the required information you can configure an IM connection as follows:

Azure Value	**Integration Module Connection Property**
Subscription ID	SubscriptionId
Tenant ID	TenantId
Application ID	Client ID
Authentication Key	Client Secret

Connecting Runbook Studio to Azure Infrastructure

In Kelverion Runbook Studio you can configure one or more Smart Connections in order to establish reusable links between Runbook Studio and a specific Azure Tenant. You can create as many Smart Connections as you require, specifying links to multiple Azure AD applications.

Adding a Smart Connection to Kelverion Runbook Studio:

1. On the ribbon, go to the **Home** tab.
2. Click **Smart Connections.**
3. On the **Smart Connections** dialog, click **New.**
4. In the **Name** box, enter a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the configuration.
5. In the optional **Description** box, enter a description of the Smart Connection.
6. In the **Connection Type** box, select **Kelverion.Azure.Infrastructure**.
7. In the **SubscriptionId** box, specify the ID of the Azure subscription you want to connect to. Please refer to Connecting to Azure for details on how to obtain the subscription ID.
8. In the **TenantId** box, specify the ID of the Azure tenant you want to connect to. Please refer to Connecting to Azure for details on how to obtain the tenant ID.
9. In the **ClientId** box, specify the Application ID of the Azure client app that is being used to connect to Azure. Please refer to Connecting to Azure for details on how to obtain the application ID.

10. In the **ClientSecret** box, specify the client secret of the Azure client app that is being used to connect to Azure. Please refer to Connecting to Azure for details on how to obtain the client secret.
11. In the **Template Folder** box, specify the path of the folder containing ARM template JSON files.
12. Click **OK**.
13. Click **OK**.

Azure Global Connection Assets

The activities in the Kelverion Integration Module for Azure Infrastructure require connection information for licensing and to connect to Azure.

The recommended way to pass connection information to your activities in your runbooks is to use Global Connection Assets. Global connection assets let you securely define connection information in Azure which can then be retrieved on demand using either the **Get-AutomationConnection** activity or **Connection Asset** data source.

Adding a global connection asset to your Azure Automation Account:

1. On the **Resources** panel, go to the **Azure** group.
2. Expand the tree to find the desired Azure Subscription and Automation Account.
3. Right click **Connections** and click **Add New Connection.**
4. In the **Name** box, enter a name for the configuration. This could be the name of the instance or a descriptive name to distinguish the type of configuration.
5. In the optional **Description** box, enter a brief description describing the connection.
6. In the **Connection Type** box, select **Kelverion.Azure.Infrastructure**.
7. In the **SubscriptionId** box, specify the ID of the Azure subscription you want to connect to. Please refer to Connecting to Azure for details on how to obtain the subscription ID.
8. In the **TenantId** box, specify the ID of the Azure tenant you want to connect to. Please refer to Connecting to Azure for details on how to obtain the tenant ID.
9. In the **ClientId** box, specify the Application ID of the Azure client app that is being used to connect to Azure. Please refer to Connecting to Azure for details on how to obtain the application ID.
10. In the **ClientSecret** box, specify the client secret of the Azure client app that is being used to connect to Azure. Please refer to Connecting to Azure for details on how to obtain the client secret.
11. Click **OK**.

Working with Activities in Runbook Studio

The following sections outline some of the common configuration options that are available to you when working with the activities in the Kelverion Integration Module for Azure Infrastructure.

Activity Properties

All activities in the Kelverion Integration Module for Azure Infrastructure have the following properties:

Property	Description
Label	A unique label that identifies the activity in the runbook. Runbook Studio will provide a default name for each activity, but you can provide your own labels to make their role in the runbook more obvious.
Description	An optional description of the activity. Providing a description is a great way to let everyone understand the function of the activity in the runbook.
Checkpoint	Indicates whether or not a checkpoint is set in the runbook workflow after the activity runs. Checkpoints are only available for Graphical PowerShell Workflow runbooks. If the runbook uses Azure activities, you should follow best practices and follow a check-pointed activity with an *Add-AzureRMAccount* in case the runbook is suspended and restarts from this checkpoint on a different worker.

Smart Discovery

When designing runbooks in Kelverion Runbook Studio, you will notice that the activities in the Kelverion Integration Module for Azure Infrastructure include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the Kelverion Integration Module for Azure Infrastructure support interactive discovery of the Azure Infrastructure assets in your environments.

All activities in the Kelverion Integration Module for Azure Infrastructure have a **Connection** option on the **Discovery** panel which lets you specify how Runbook Studio should connect to Azure.

When connected to Azure Infrastructure, Runbook Studio will provide additional discovery options, such as **Resource Group** and **Template File**, which can be used to specify the resources that you want to integrate with. Once you have filled in the discovery options Runbook Studio will provide additional parameters and, in some cases, filters which can be used to configure the activity.

Smart Parameters

Unlike standard command activities, whose parameters are determined by the Parameter Set that is selected, the parameters in the Kelverion Integration Module for Azure Infrastructure are determined by the Discovery options that you specify.

For example, when using the **New-AzureKaResourceDeployment** activity, the Discovery panel will contain options for selecting a template source. Once you have selected a template, Runbook Studio will provide you with mandatory and optional parameters that coincide with the fields in the template. If you select another template, Runbook Studio will provide you with a different set of parameters automatically.

You must configure all mandatory parameters. To view the optional parameters that are associated with an activity, click **Optional** at the top of the Parameters tab.

In addition, all activities in the Kelverion Integration Module for Azure Infrastructure include a **Connection** parameter which is used to specify information that the activity will use to connect to Azure when it is executed as part of a runbook running in Azure or on a Hybrid Worker. Typically, you will assign a Connection Asset data source to this parameter so that the activity can securely use connection information stored in Azure.

The Connection parameter should not be confused with the similarly named Connection option on the Discovery panel which is used to specify how Runbook Studio connects to Azure Infrastructure in order to provide design-time configuration options.

Several factors determine the data sources that are available when configuring a parameter. They include: the parameter's data type, whether it is linked to another activity and whether the runbook has any input parameters.

Runbook studio supports the following data sources.

Activity output	Specify activity whose output will be assigned to the parameter. You may also provide an optional Path to select a specific property of the output objects that are generated by the activity. Available when the activity is linked to a source activity.
Not configured	Clears any value that was previously configured. You must configure all mandatory parameters.
Certificate asset	Specify the name of the global certificate asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the certificates that are available.
Credential asset	Specify the name of the global credential asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the credentials that are available.
Constant	Specify a constant value to assign to the parameter. Available for parameters that have the following data types: • String • DateTime • Timespan

	• Decimal • Double
Connection asset	Specify the name of the global connection asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the connections that are available.
Empty string	An empty string will be assigned to the parameter. Available when the parameter is type *System.String*
Null	A null ($null) value will be assigned to the parameter. Available when the parameter type is a reference type.
PowerShell expression	Specify a *simple* PowerShell expression whose output will be assigned to the parameter. You can use variables in the expression to access the output of an activity or a runbook parameter.
Runbook input	Specify the name of the runbook input parameter whose value will be assigned to the parameter. Available when the runbook has one or more input parameters.
Variable asset	Specify the name of the global variable asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the variables that are available.

Important: When assigning a constant DateTime and Time values, Runbook Studio assumes the value is in UTC.

Smart Filters

Some of the activities in the Kelverion Integration Module for Azure Infrastructure include a **Filters** panel which lets you specify filters that can be used to retrieve specific Azure resources.

To add a filter to your activity, select the **Filters** panel and click **Add.** Filters have the following properties.

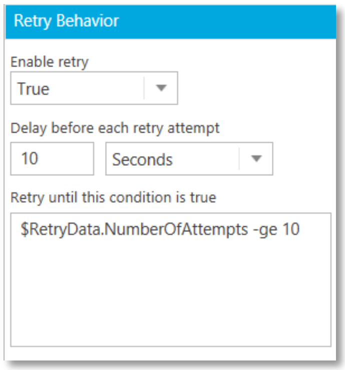
Filter	The name of the filter.

Operation	The operation used to evaluate the filter. Different operators will be provided based on the filter that is selected. Possible filter operators include: • Equals • Does not equal • Is less than • Is less than or equal to • Is greater than • Is greater than or equal to • Contains • Does not contain • Matches • Does not match • Starts with • Ends with
Value	The data source used to retrieve the value to use to evaluate the filter. The value used to evaluate the filter will be obtained. For more information on data sources, please refer to the Parameters section for more information on configuring data sources.

Retry Behavior

The activities in the Kelverion Integration Module for Azure Infrastructure can be configured to run multiple times until a condition, which you specify, is satisfied. You can use the retry behavior options to configure activities that should run multiple times, that are error prone or may need more than one attempt for success.

When you enable retry for an activity, you can configure the runbook to wait a specified number of minutes or seconds before running the activity again. If no delay is specified the runbook will run the activity again, immediately after it completes.


Retry Behavior

Enable retry: True

Delay before each retry attempt: 10 Seconds

Retry until this condition is true: $RetryData.NumberOfAttempts -ge 10

The retry condition lets you specify a PowerShell expression that the runbook will evaluate after each time the activity runs. If the result of the expression is true the activity does not run again, and the runbook moves on to the next child activity in the runbook.

When defining the retry conditions for your activity you can take advantage of a global variable called **$RetryData**. Specific information about the last time the activity ran can be accessed using the following properties.

Property	Description
NumberOfAttempts	Number of times that the activity has ran
Output	Output that was generated by the activity the last time that it ran
TotalDuration	Time elapsed since the activity was started
StartedAt	Time in UTC when the activity was first started

The following are some examples of activity retry conditions.

```
# Run the activity exactly 5 times
$RetryData.NumberOfAttempts -eq 5

# Run the activity until it produces some output
$RetryData.Output.Count -ge 1

# Run the activity until at least 2 minutes has elapsed
$RetryData.TotalDuration.TotalMinutes -ge 2
```

Additional Parameters

The activities in the Kelverion Integration Module for Azure Infrastructure let you specify additional PowerShell parameters that you can use to control the behavior of the activity.

For example, to output detailed information about the operation performed by an activity you would specify **`-Verbose:$True`**

Azure Resource IDs

Azure resource IDs are strings representing the fully qualified path of the resource in the resource hierarchy, with the following format:

```
/subscriptions/{subscription-guid}
/resourceGroups/{resource-group-name}
/providers/{resource-provider-namespace}
[/{parent-resource-type}/{parent-resource-name}]*
/{resource-type}/{resource-name}
```

For example, for a VM with name “VM-Test1” located in a resource group “Kelverion-Azure-VM”, the fully qualified resource ID is:

```
/subscriptions/d514b975-8729-4453-8527-122ad6a4a37d
/resourceGroups/Kelverion-Azure-VM
/providers/Microsoft.Compute
/virtualMachines/VM-Test1
```

The ID for a public IP address resource with name “VM-Test1-IP”, would be:

```
/subscriptions/d514b975-8729-4453-8527-122ad6a4a37d
/resourceGroups/Kelverion-Azure-VM
/providers/Microsoft.Network
/publicIPAddresses/VM-Test1-IP
```

The ID for a disk resource with name “VM-Test1_DataDisk_1”, would be:

```
/subscriptions/d514b975-8729-4453-8527-122ad6a4a37d/
resourceGroups/Kelverion-Azure-VM
/providers/Microsoft.Compute
/disks/VM-Test1_DataDisk_1
```

Resource Group IDs

Fully qualified resource group IDs follow the same structure, up to including the resource group name:

```
/subscriptions/d514b975-8729-4453-8527-122ad6a4a37d/
resourceGroups/Kelverion-Azure-VM
```

Resource Types

Resource types consist of the resource provider namespace, parent resource type (if any) and resource type of the resource:

```
{resource-provider-namespace}[/{parent-resource-type}]*/{resource-type}
```

Some example resource types:

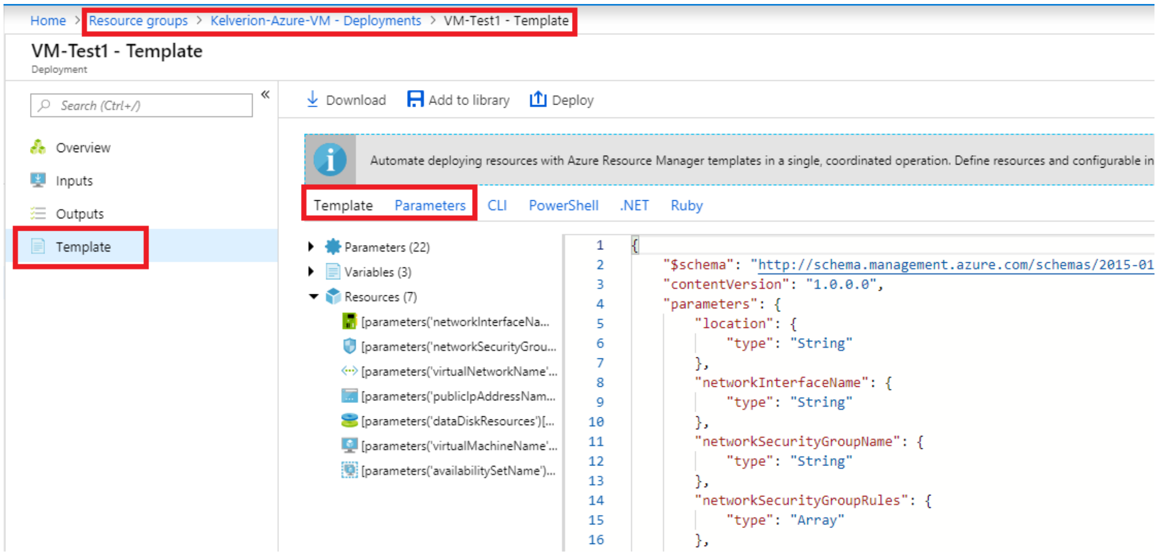
```
Microsoft.Compute/virtualMachines
Microsoft.Network/publicIPAddresses
Microsoft.Compute/disks
```

Working with ARM Templates

An Azure Resource Manager (ARM) Template is a JSON formatted script that defines one or more resources to deploy to a resource group or subscription. Currently the integration module only supports deploying to a resource group.

The template defines a set of parameters which are used for customizing the template deployment. The template parameters must be provided as a separate JSON script when deploying the template. The template can also define a set of outputs which are returned after the deployment is completed.

To see some example template and template parameters in your Azure environment, navigate to a resource group which contains some deployments, select a deployment, and then select Template in the left pane. In the menu bar you can now select Template to see the template JSON and Parameters to see the parameters JSON.


Home > Resource groups > Kelverion-Azure-VM - Deployments > VM-Test1 - Template
VM-Test1 - Template
Deployment
Search (Ctrl+/)
Overview
Inputs
Outputs
Template
Download
Add to library
Deploy
Template
Parameters
CLI
PowerShell
.NET
Ruby
Parameters (22)
Variables (3)
Resources (7)
[parameters('networkInterfaceNa...
[parameters('networkSecurityGrou...
[parameters('virtualNetworkName'...
[parameters('publicIpAddressNam...
[parameters('dataDiskResources')[...
[parameters('virtualMachineName'...
[parameters('availabilitySetName')...

Automate deploying resources with Azure Resource Manager templates in a single, coordinated operation. Define resources and configurable in

```
{
    "$schema": "http://schema.management.azure.com/schemas/2015-01
    "contentVersion": "1.0.0.0",
    "parameters": {
        "location": {
            "type": "String"
        },
        "networkInterfaceName": {
            "type": "String"
        },
        "networkSecurityGroupName": {
            "type": "String"
        },
        "networkSecurityGroupRules": {
            "type": "Array"
        },
```

For more details on ARM template file structure, please refer to Azure template documentation:

https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/syntax

The **New-AzureResourceDeployment** activity integrates the template deployment process into automation runbooks. The activity analyzes a specified template and provides activity input and output parameters dynamically, based on what parameters and outputs are defined in that template. As a user, you no longer have to provide a custom parameters JSON file when running the deployment. Instead, you can parametrize your deployment with data from your runbook via activity input parameters. The activity internally compiles all the input data into the parameters JSON structure and provides it to deployment execution, along with the template.

Template Parameter Types

Currently the following types of parameters can be defined in an ARM template:

- string
- securestring
- int
- bool
- object
- secureObject
- array

The **New-AzureResourceDeployment** activity supports these types but not in the same way. Simple parameter types, such as string, securestring, int and bool, can be integrated seamlessly as activity input parameters. However, for complex parameter types (object, secureObject, array), the entire JSON structure representing the parameter value must be provided.

For example, for a VM deployment the following parameters can be defined in the template JSON:

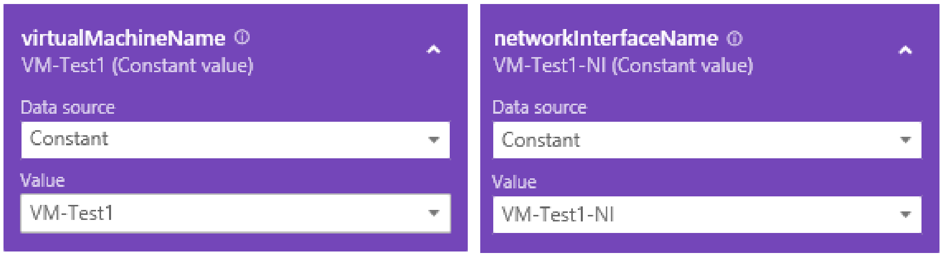
```
"parameters": {
    ...
    "virtualMachineName": {
        "type": "String"
    },
    "networkInterfaceName": {
        "type": "String"
    },
    "dataDisks": {
        "type": "Array"
    },
    ...
}
```

The first parameter defines the VM name, the second defines the VM network interface name and the third defines an array for specifying VM data disk information. When deploying this template, the provided parameters JSON structure must contain values for template defined parameters:

```
"parameters": {
    ...
    "virtualMachineName": {
        "value": "VM-Test1"
    },
    "networkInterfaceName": {
        "value": "VM-Test1-NI"
    },
    "dataDisks": {
        "value": [
        {
            "lun": 0,
            "createOption": "attach",
            "caching": "None",
            "writeAcceleratorEnabled": false,
            "id": null,
            "name": "VM-Test1_DataDisk_0",
            "storageAccountType": null
        },
        {
            "lun": 1,
            "createOption": "attach",
            "caching": "None",
            "writeAcceleratorEnabled": false,
            "id": null,
            "name": "VM-Test1_DataDisk_1",
            "storageAccountType": null
          }
        ]
    },
    ...
}
```

The first two parameters specify the VM name as “VM-Test1”, and the VM network interface name as “VM-Test1-NI”. The third parameter specifies an array of two objects, each one containing information for a data disk.

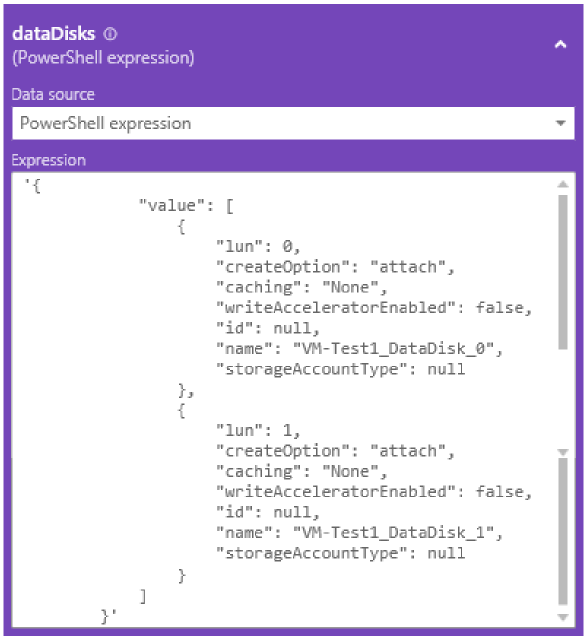
The IM can seamlessly integrate with string type parameters, so the first two parameters can easily be configured in the Runbook Studio activity parameters:



virtualMachineName
VM-Test1 (Constant value)
Data source: Constant
Value: VM-Test1

networkInterfaceName
VM-Test1-NI (Constant value)
Data source: Constant
Value: VM-Test1-NI

For the dataDisks parameter however, since it’s a complex type, the entire JSON structure must be provided as a PowerShell expression:



```
'{
            "value": [
                {
                    "lun": 0,
                    "createOption": "attach",
                    "caching": "None",
                    "writeAcceleratorEnabled": false,
                    "id": null,
                    "name": "VM-Test1_DataDisk_0",
                    "storageAccountType": null
                },
                {
                    "lun": 1,
                    "createOption": "attach",
                    "caching": "None",
                    "writeAcceleratorEnabled": false,
                    "id": null,
                    "name": "VM-Test1_DataDisk_1",
                    "storageAccountType": null
                }
            ]
        }'
```

The parameter Data source must be specified as “PowerShell expression” and the expression value must be the dataDisks JSON property value. **Make sure to include the opening and closing braces, and to enclose the entire expression in single quotes to denote a PowerShell string.**

```
"dataDisks": {
    "value": [
    {
        "lun": 0,
        "createOption": "attach",
        "caching": "None",
        "writeAcceleratorEnabled": false,
        "id": null,
        "name": "VM-Test1_DataDisk_0",
        "storageAccountType": null
    },
    {
        "lun": 1,
        "createOption": "attach",
        "caching": "None",
        "writeAcceleratorEnabled": false,
        "id": null,
        "name": "VM-Test1_DataDisk_1",
        "storageAccountType": null
      }
    ]
},
```

Activity Reference

The following sections describe how to configure the activities in the Kelverion Integration Module for Azure Infrastructure in conjunction with Kelverion Runbook Studio.

The Integration Module for Azure Infrastructure includes the following activities:

Get-AzureKaResource	Retrieves Azure resources.
Get-AzureKaResourceGroup	Retrieves Azure resource groups.
Move-AzureKaResource	Moves an Azure resource to another resource group.
New-AzureKaResourceDeployment	Deploys an Azure Resource Manager (ARM) template.
Remove-AzureKaResource	Removes an Azure resource.
Remove-AzureKaResourceTag	Removes a resource tag from an Azure resource.
Remove-AzureKaVm	Removes an Azure VM.
Set-AzureKaResourceTag	Sets a resource tag on an Azure resource.

Get-AzureKaResource

The **Get-AzureKaResource** activity is used in a runbook to retrieve and filter Azure resources for the configured subscription.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.
Search By	Specifies how Azure resources will be retrieved. • **Filters**: you can specify filters to narrow down the result set. • **Resource ID**: you specify an ID to retrieve a specific resource. • **Tag**: you specify a resource tag name and value to retrieve all resources associated with that tag.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Resource ID	Specifies the ID of a resource to be retrieved, when searching by Resource ID. Please refer to Azure Resource IDs for details on how to format the resource ID. Available when **Search By** is set to **Resource ID**.
Tag Name	Specifies the name of a resource tag, when retrieving resources by Tag. Available when **Search By** is set to **Tag**.
Tag Value	Specifies the value of a resource tag, when retrieving resources by Tag. Available when **Search By** is set to **Tag**.

Filters

When the **Search By** option is set to **Filters**, the activity will provide the following filters that you can use to select which resource objects to retrieve.

Location	Filter by resource location.
Name	Filter by resource name.
PlanName	Filter by resource plan name.
PlanProduct	Filter by resource plan product.
PlanPublisher	Filter by resource plan publisher.
Resource Group	Filter by resource group name.
Resource Type	Filter by resource type. Please refer to Resource Types for details on how to format the resource type.

Outputs

This activity outputs objects that represent the resources that were retrieved. Each object has the following properties.

ResourceId	Resource identifier.
Location	Resource location.
ManagedBy	Resource ID of the resource managing this resource.
Name	Resource name.
PlanName	Resource plan name.
PlanProduct	Resource plan product.
PlanPublisher	Resource plan publisher.
ResourceGroup	Resource group name.
SkuName	Resource SKU name.
SkuTier	Resource SKU tier.
Tags	Tags associated with this resource. (Dictionary)
ResourceType	Resource type.

Get-AzureKaResourceGroup

The **Get-AzureKaResourceGroup** activity is used in a runbook to retrieve Azure resource groups for the configured subscription.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.
Search By	Specifies how Azure resource groups will be retrieved. • **Name**: you specify a resource group name to retrieve a specific resource group. • **Tag**: you specify a resource group tag name and value to retrieve all groups associated with that tag. When this property is not specified, the activity will retrieve all resource groups for the configured subscription.

Required Parameters

You must configure the following parameters.

Name	Description
Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Name	Specifies the name of a resource group to be retrieved, when retrieving groups by Name. Available when **Search By** is set to **Name.**
Tag Name	Specifies the name of a resource group tag, when retrieving groups by Tag. Available when **Search By** is set to **Tag**.
Tag Value	Specifies the value of a resource group tag, when retrieving groups by Tag. Available when **Search By** is set to **Tag**.

Outputs

This activity outputs objects that represent the resource groups that were retrieved. Each object has the following properties.

Name	Resource group name.
ProvisioningState	Provisioning state for the resource group.
RegionName	Resource group region name.
ResourceGroupId	Resource group ID.
Tags	Tags associated with the resource group.

Move-AzureKaResource

The **Move-AzureKaResource** activity is used in a runbook to move existing Azure resources to another resource group. The activity moves a single resource, if there are other resources related to the resource that is being moved, those resources will have to be moved separately. Note that tools and scripts associated with moved resources will have to be updated with the new resource ID.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
New Resource Group ID	Specifies the ID of the resource group where the resource is to be moved. **This resource ID will no longer be valid after the move operation is complete. The activity will publish the new resource ID for the resource.** Please refer to Resource Group IDs for details on how to format the resource group ID.
Resource ID	Specifies the ID of a resource to be retrieved, when searching by Resource ID. Please refer to Azure Resource IDs for details on how to format the resource ID.

Outputs

The activity publishes the new ID of the resource after being moved to the new resource group.

New-AzureResourceDeployment

The **New-AzureResourceDeployment** activity is used in a runbook to deploy an Azure Resource Manager (ARM) template and create one or more Azure resources. The activity will dynamically provide inputs and outputs corresponding to the ARM template selected during discovery.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.
Template Source	Specifies where the activity will retrieve the ARM template that will be used for the new deployment. • **Local File:** The template will be retrieved from a local JSON file. • **Azure Deployment:** The template will be retrieved from a previous deployment.
Template File	Specifies the name of the JSON file containing the ARM template to be deployed. The file must be located in the **Template Folder** specified by the Kelverion.Azure.Infrastructure smart connection. This property is available only when **Template Source** is set to **Local File.**
Deployment Resource Group	Specifies the name of the resource group containing a previous deployment whose template will be used for the new deployment. This property is available only when **Template Source** is set to **Azure Deployment.**
Deployment	Specifies the name of a previous deployment whose template will be used for the new deployment. This property is available only when **Template Source** is set to **Azure Deployment.**
Location	Specifies the location for the new deployment. When specified, location template parameters will be set to this value.
Resource Group	Specifies the resource group name for the new deployment.

Required Parameters

You must configure the following parameters. Additional parameters will be provided, based on the selected template.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Deployment Name	The name for the new deployment. Deployment names do not have to be unique, but if an existing deployment name is specified, the new deployment will overwrite the existing deployment.
Resource Group	Specifies the resource group name for the new deployment. This parameter is available only when the **Resource Group** discovery property is not specified.

Note: When configuring parameters with a Constant Data source, you may be provided with a drop-down list. If the value you want is not on the list, change the Data source to PowerShell expression and enter the value in single quotes.

Outputs

The activity outputs an object that contain information about the new resource deployment. The properties of the object are determined by the selected ARM template.

Remove-AzureKaResource

The **Remove-AzureKaResource** activity is used in a runbook to remove an existing Azure resource.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Resource ID	Specifies the ID of a resource to be removed, when removing resource. Please refer to Azure Resource IDs for details on how to format the resource ID.

Outputs

The activity outputs objects that contain information about the resources that were removed. Each object has the following properties.

ApiVersion	API version used to delete the resource.
DeleteStatus	Indicates whether the delete operation has succeeded or failed.
Error	Error information when deleting the resource fails.
ResourceId	ID of the deleted resource.
ResourceName	Name of the deleted resource.
ResourceType	Type of the deleted resource.

Remove-AzureKaResourceTag

The **Remove-AzureKaResourceTag** activity is used in a runbook to remove a resource tag from an existing Azure resource.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Resource ID	Specifies the ID of the resource for which the tag is to be removed. Please refer to Azure Resource IDs for details on how to format the resource ID.
Tag Name	Specifies the name of the tag that is to be removed.

Optional Parameters

You can configure the following optional properties as necessary.

Wait (seconds)	Number of seconds the activity will wait while verifying that the tag was removed. When not configured, or when set to 0, the activity does not wait.

Outputs

The activity publishes the ID of the resource that the tag was removed for.

Remove-AzureKaReVm

The **Remove-AzureKaVm** activity is used in a runbook to remove an existing Azure virtual machine.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Remove Disks	Specifies that the activity should also remove all disks used by a VM.
Remove NICs	Specifies that the activity should also remove all NICs used by a VM.
Remove Public IPs	Specifies that the activity should also remove all public IP addresses associated with NICs used by a VM.
VM ID	Specifies the ID of a VM to be removed. Please refer to Azure Resource IDs for details on how to format the VM ID.

Outputs

This activity outputs objects that contain information about the resources that were removed. Each object has the following properties.

ApiVersion	API version used to delete the resource.
DeleteStatus	Indicates whether the delete operation has succeeded or failed.
Error	Error information when deleting the resource fails.
ResourceId	ID of the deleted resource.
ResourceName	Name of the deleted resource.
ResourceType	Type of the deleted resource.

Set-AzureKaResourceTag

The **Set-AzureKaResourceTag** activity is used in a runbook to set a resource tag value on an existing Azure resource. The resource tag is created if it doesn't exist.

Discovery Options

You can use the following options to connect to Azure and configure the smart activity.

Connection	The name of the smart connection used for design-time activity configuration.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing runtime connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Resource ID	Specifies the ID of the resource for which the tag is to be set. Please refer to Azure Resource IDs for details on how to format the resource ID.
Tag Name	Specifies the name of the tag that is to be set.

Optional Parameters

You can configure the following optional parameters as necessary.

Tag Value	Specifies the new tag value.
Wait (seconds)	Number of seconds the activity will wait while verifying that the tag was set. When not configured, or when set to 0, the activity does not wait.

Outputs

The activity outputs the ID of the resource that was updated.