



INTEGRATION MODULE FOR AZURE MONITOR

For Keverion Runbook Studio and Azure Automation

User Guide

Version 1.1



Kelverion Integration Module for Azure Monitor

Copyright © 2021 Kelverion Inc. All rights reserved.

Published: July 2026

The Kelverion Integration Module for Azure Monitor is Microsoft Azure Certified.

Feedback

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

Contents

Getting Started.....	4
System Requirements.....	4
Installing the Integration Module.....	4
Using the PowerShell Gallery.....	4
Manual Installation	5
Licensing the Integration Module	6
Creating Your Azure AD Application.....	7
Working with Activities in Runbook Studio.....	8
Smart Connections	8
Global Connection Assets.....	9
Smart Discovery.....	9
Activity Properties	10
Smart Parameters.....	10
Smart Filters	12
Retry Behavior	12
Additional Parameters.....	13
Get-AzureKaAlert	14
Get-AzureKaAlertHistory.....	17
Outputs	17
Set-AzureKaAlertState	18

Getting Started

The following sections outline how to deploy and configure the Kolverion Integration Module for Azure Monitor.

System Requirements

The Integration Module for Azure Monitor requires the following software to be installed and configured prior to implementing the integration.

- Kolverion Runbook Studio 5.9
- Microsoft .NET Framework 4.7.2
- Microsoft Azure Subscription
- Azure Automation account
- Azure AD Application and Security Principal

Installing the Integration Module

The easiest way to install and deploy the Integration Module for Azure Monitor is from the PowerShell Gallery, but you can also download the module from Kolverion 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 Kolverion Integration Module for Azure Monitor 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 you the PowerShellGet module installed.
2. Start a PowerShell window as Administrator and run the command:
Install-Module -Name Kolverion.Azure.Monitor -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 Keverion and deploy it manually to your local computer, hybrid workers and Automation Accounts.

The download package from Keverion 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.

Install the Integration Module on your local computer or hybrid worker:

1. Copy the **Keverion.Azure.Monitor.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 **Keverion.Azure.Monitor.zip** file.
5. Copy the **Keverion.Azure.Monitor** folder to a location in the `%PsModulePath%` path.

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

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 **Keverion.Azure.Monitor.zip** file that you downloaded.
6. Click **OK**. Importing the module may take several minutes.

Licensing the Integration Module

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

Register an Integration Module license with Runbook Studio:

1. Open **Keverion Runbook Studio**.
2. On 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**.

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

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 **Home** tab, 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.

Creating Your Azure AD Application

To access the resources it needs for monitoring, the Integration Module for Azure Monitor, requires that you create an **Azure AD application** and **service principal**. The article referenced by the following link shows how to create a single-tenant application and security principal.

<https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal#create-an-azure-active-directory-application>

Important: You will need to record your **Subscription ID**, **Tenant ID**, **Application ID**, and **Client Secret** so that you can setup connections for your runbooks in Runbook Studio.

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 Keverion Integration Azure Monitor.

The integration module includes the following activities:

Get-AzureKaAlert	Get a list of Azure alerts.
Get-AzureKaAlertHistory	Get alert history for a specific Azure alert.
Set-AzureKaAlertState	Modify the alert state of an existing alert.

Important: The advanced discovery capabilities provided by the activities in this integration module are only supported when authoring runbooks in Keverion Runbook Studio.

When you publish your runbooks from Keverion Runbook Studio to Azure Automation, Runbook Studio will automatically convert the dynamically generated parameters and filters of Smart activities into the parameters provided by the underlying command activities.

Smart Connections

In Keverion Runbook Studio you can configure one or more Smart Connections to establish reusable links between Runbook Studio and specific Azure subscriptions. You can create as many Smart Connections as you require, specifying links to multiple databases. You can also create multiple Smart Connections to the same database to allow for differences in security privileges for different user accounts.

You will need your **Subscription ID** and **Tenant ID** as well as the **Application ID** and **Client Secret** from the Azure AD application that you created for the integration module. For more information see [Creating Your Azure AD Application](#).

Add a Smart Connection in Keverion Runbook Studio:

1. On the **Home** tab, click **Smart Connections**. The Smart Connections dialog appears.
2. Click **Add a connection** at the top of the list.
3. In the **Name** box, type the name for the connection.
4. In the **Connection Type** box, select *Keverion.Azure.Monitor*.
5. In the **SubscriptionId** box, type your Azure subscription ID.
6. In the **TenantId** box, type your Azure AD Tenant ID.
7. In the **ClientId** box, type the Application ID of your Azure AD application.
8. In the **ClientSecret** box, type the client secret of your Azure AD application.
9. Click **OK**, and then click **OK** again.

Global Connection Assets

The activities in the Keverion Integration Module for Azure Monitor require connection information to connect to an Azure subscription. 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* cmdlet or Connection Asset Data Source.

You will need your **Subscription ID** and **Tenant ID** as well as the **Application ID** and **Client Secret** from the Azure AD application that you created for the integration module. For more information see [Creating Your Azure AD Application](#).

Add a global connection asset in Runbook Studio:

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** box, enter a name for the configuration.
6. In the **Connection Type** box, select *Keverion.Azure.Monitor*.
7. In the **SubscriptionId** box, type your Azure subscription ID.
8. In the **TenantId** box, type your Azure AD Tenant ID.
9. In the **ClientId** box, type the Application ID of your Azure AD application.
10. In the **ClientSecret** box, type the client secret of your Azure AD application.
11. Click **OK**.

Smart Discovery

When designing runbooks in Keverion Runbook Studio, you will notice that the activities in the Keverion Integration Module for Azure Monitor include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the Keverion Integration Module for Azure Monitor support interactive design-time configuration. All activities in the Keverion Integration Module for Azure Monitor have a **Connection** option on the **Discovery** panel which lets you specify the design-time connection for the activity.

Activity Properties

All activities in the Keverion Integration Module for Azure Monitor have the following properties:

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 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 cmdlets, 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 Parameters

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

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 Keverion Integration Module for Azure Monitor include a **Connection** parameter which is used to specify runtime connection information. This is used when the activity executes as part of a runbook running in Azure or on the 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 Monitor 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 • Boolean • Char • Byte • SByte • Int16 • Int32 • Int64 • UInt16 • UInt32 • UInt64 • Decimal • Double • Float • SwitchParameter When assigning a constant DateTime value, Runbook Studio assumes the value is in UTC.
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 <i>System.String</i>
Null	A null (\$null) value will be assigned to the parameter. Available when the parameter type is a reference type.
PowerShell expression	Specify a <i>simple</i> 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.

Smart Filters

Some of the activities in the Keverion Integration Module for Azure Monitor include a Filters panel which lets you specify filters that can be used to retrieve a sub-set of objects.

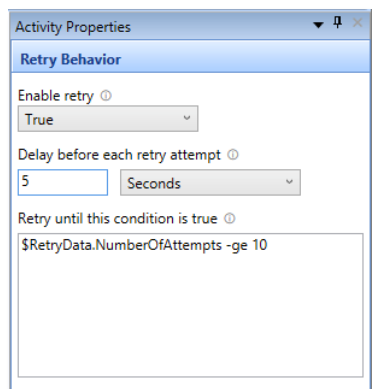
If you do not provide any filters, the activity will retrieve all objects for that activity.

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 used 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 Keverion Integration Module for Azure Monitor can be configured to run multiple times until a particular 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.

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.

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 Monitor 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**

Get-AzureKaAlert

The **Get-AzureKaAlert** activity is used in a runbook to retrieve Azure alert records for a limited time interval. The following tables list the parameters, filters, and outputs for this activity.

Discovery Parameters

You can use the following discovery options to connect to Azure and configure the 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. • Alert ID – you specify an ID to retrieve a specific alert.
Time Range	When retrieving multiple alerts, it specifies if you want to use a time interval to filter results further.
Custom Time Range	When using Time Range, it specifies whether you want to use a custom date and time interval, or if you want to use one of the preset time range values.

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.
Alert ID	Specifies the ID of an alert to be retrieved when searching by Alert ID. When Search By is set to <i>Alert ID</i> . The ID must be in the form: <i>/subscriptions/{subscription-guid}</i> <i>/providers/Microsoft.AlertsManagement/alerts/{alert-guid}</i>
End Time	Specifies the end time of a custom time interval for filtering alerts. The time interval must be within 30 days of the time when the activity runs.
Start Time	Specifies the start time of a custom time interval for filtering alerts. The time interval must be within 30 days of the time when the activity runs.
Time Range	Specifies a time interval for filtering alerts. Allowed values are: • 1h – one hour • 1d – one day • 7d – seven days • 30d – thirty days

Optional Parameters

You can use the following parameters to control the behavior of the activity.

Sort By	Specifies an alert field for sorting results. Allowed values are: • AlertState • LastModifiedDateTime • MonitorCondition
----------------	--

	• Name • Severity • StartDateTime • TargetResourceGroup • TargetResourceName • TargetResourceType
Sort Order	Specifies the sort order when a Sort By field is specified. Allowed values are: • Ascending • Descending

Filters

When **Search By** is set to *Filters*, the activity provides the following filters that you can use to control which alerts to retrieve.

AlertRuleId	Filter by AlertRuleId
AlertState	Filter by AlertState
MonitorCondition	Filter by MonitorCondition
MonitorService	Filter by MonitorService
Severity	Filter by Severity
TargetResourceGroup	Filter by TargetResource Group
TargetResourceID	Filter by TargetResource ID
TargetResourceType	Filter by TargetResource Type

Outputs

The activity outputs objects that represent the alerts that were retrieved. Each alert object has the following properties.

AlertId	The ID of the alert.
AlertRuleId	The ID of the alert rule for this alert.
AlertState	The alert state.
Description	The description of the alert
IsSuppressed	Indicates if the alert is suppressed or not.
LastModifiedDateTime	The date and time when this alert record was last modified.
MonitorCondition	The alert monitor condition.
MonitorService	The name of the monitor service.
Name	The name of the alert
Severity	The severity of the alert
SignalType	The signal type

SmartGroupId	The alert smart group ID.
SmartGroupingReason	The alert smart grouping reason.
SourceCreatedId	The ID of the source that created the alert
StartDateTime	The date and time when that the alert was created
TargetResourceId	The ID of the resource targeted by the alert
TargetResourceGroup	The resource group of the resource targeted by the alert
TargetResourceName	The name of the resource targeted by the alert
TargetResourceType	The type of the resource targeted by the alert

Get-AzureKaAlertHistory

The **Get-AzureKaAlertHistory** activity is used in a runbook to retrieve alert history records for a specific alert. Alert history captures any monitor condition changes (i.e., Fired/Resolved) and alert state changes (i.e., New/Acknowledged/Closed). The following tables list the parameters and outputs for this activity.

Discovery Parameters

You can use the following discovery options to connect to Azure and configure the 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.
Alert ID	Specifies the ID of the alert for which history is to be retrieved. The ID must be in the form: <code>/subscriptions/{subscription-guid}</code> <code>/providers/Microsoft.AlertsManagement/alerts/{alert-guid}</code>

Outputs

The activity outputs objects that represent an alert history record. Each object has the following properties.

AlertId	The alert ID.
AlertHistoryId	The ID for this alert history record.
AlertName	The alert name.
Comments	Alert history record comments.
Description	Alert history record description.
ModificationEvent	The modification reason
ModifiedAt	The date and time that the modification took place.
ModifiedBy	The name off the principal client that modified the alert
NewAlertSate	The new alert state.
OldAlertState	The old alert state.

Set-AzureKaAlertState

The **Set-AzureKaAlertState** activity is used in a runbook to modify the alert state for an existing Azure alert. The following tables list the parameters and outputs for this activity.

Discovery Parameters

You can use the following discovery options to connect to Azure and configure the 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.
Alert ID	Specifies the ID of an alert to be modified. The ID must be in the form: <i>/subscriptions/{subscription-guid}</i> <i>/providers/Microsoft.AlertsManagement/alerts/{alert-guid}</i>
New State	Specifies the new alert state.

Outputs

The activity outputs the alert ID as a string.