Kelverion Automation

Automated Patching for Azure Automation User Guide

User Guide

Version 1.1

Email: info@kelverion.com

Web: www.kelverion.com

1. Table of Contents

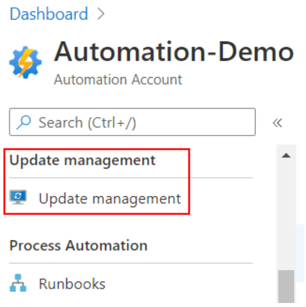
2. Introduction

The Kelverion Automated Patching Application provides a set of automation runbooks linked to the Kelverion Automation Portal and Azure Update Management. This guide will help you to identify which parts of the Kelverion Automated Patching Application fits into Azure Update Management so you can easily control your patching deployments.

Parts of this guide will cover sections that may require an Azure Administrator to access or a SQL Administrator for the backend database that the runbooks access called the Persistent Data Store (PDS).

3. Azure Update Management

You can use Update Management in Azure Automation to manage operating system updates for your Windows and Linux virtual machines in Azure, in on-premises environments, and in other cloud environments. You can quickly assess the status of available updates on all agent machines and manage the process of installing required updates for servers.



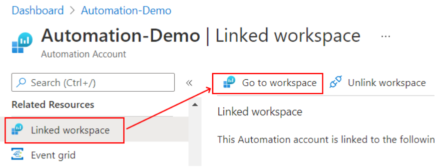
When installed you will find Update Management part of an Automation Account in Azure.

3.1. Update Management Requirements

The Update Management feature in Azure requires the following components:

- Automation account
- Log Analytics Workspace

The Automation account will provide the Update Management UI for an Azure Administrator. The Log Analytics Workspace provides the data collected from the clients and can be used to query and provide reporting.



This Automation account is linked to the followin

You can find the workspace through the 'Linked Workspace' option in Update Management.

Then click on 'Go to workspace'.

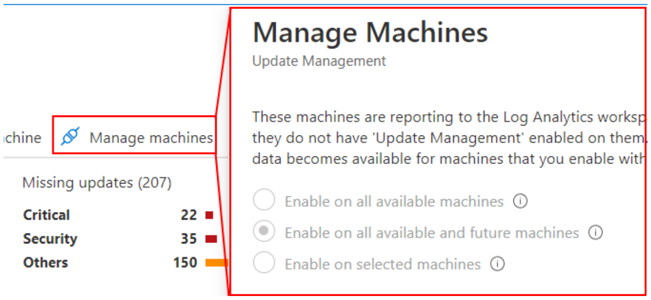
3.2. Clients

The Update Management feature will install a system Hybrid Runbook Worker on each client device. Only specific operating systems are supported by Microsoft for Update Management. An up-to-date list of supported operating systems can be found here:

https://docs.microsoft.com/en-us/azure/automation/update-management/overview#supported-operating-systems

Update Management uses the Microsoft Management Agent (MMA) as a client.

Once the MMA is installed you can set Update Management to automatically enable itself on devices that are linked to the workspace. You can see the clients active and responding in the Update Management section of the Azure Portal. You can also directly query the Log Analytics Workspace to find which devices are currently available.


chine
Manage machines

Missing updates (207)

Critical	22
Security	35
Others	150

Manage Machines

Update Management

These machines are reporting to the Log Analytics worksp
they do not have 'Update Management' enabled on them
data becomes available for machines that you enable with

○ Enable on all available machines
● Enable on all available and future machines
○ Enable on selected machines

The application offers an automated installation of the MMA client, if devices are in an Azure resource group.

Otherwise, for non-Azure devices, you will need to manually install the client.

Instructions for doing so are listed in the Microsoft article:

https://docs.microsoft.com/en-gb/azure/azure-monitor/agents/log-analytics-agent#installation-options

3.3. Client configuration – Source Server

Windows servers will use the Windows Update client to collect updates for Update Management. Any group policy in place will take precedence over Update Management. You can use this to define the service location of your updates if required, by pointing your devices at a WSUS server.

If you are using an internal WSUS server, you must ensure that any selected updates are approved. If the updates are not approved in WSUS, the patch deployment will fail.

Likewise for Linux servers you can define a local or public repository and they will use this location for Update Management.

4. Automated Patching Application

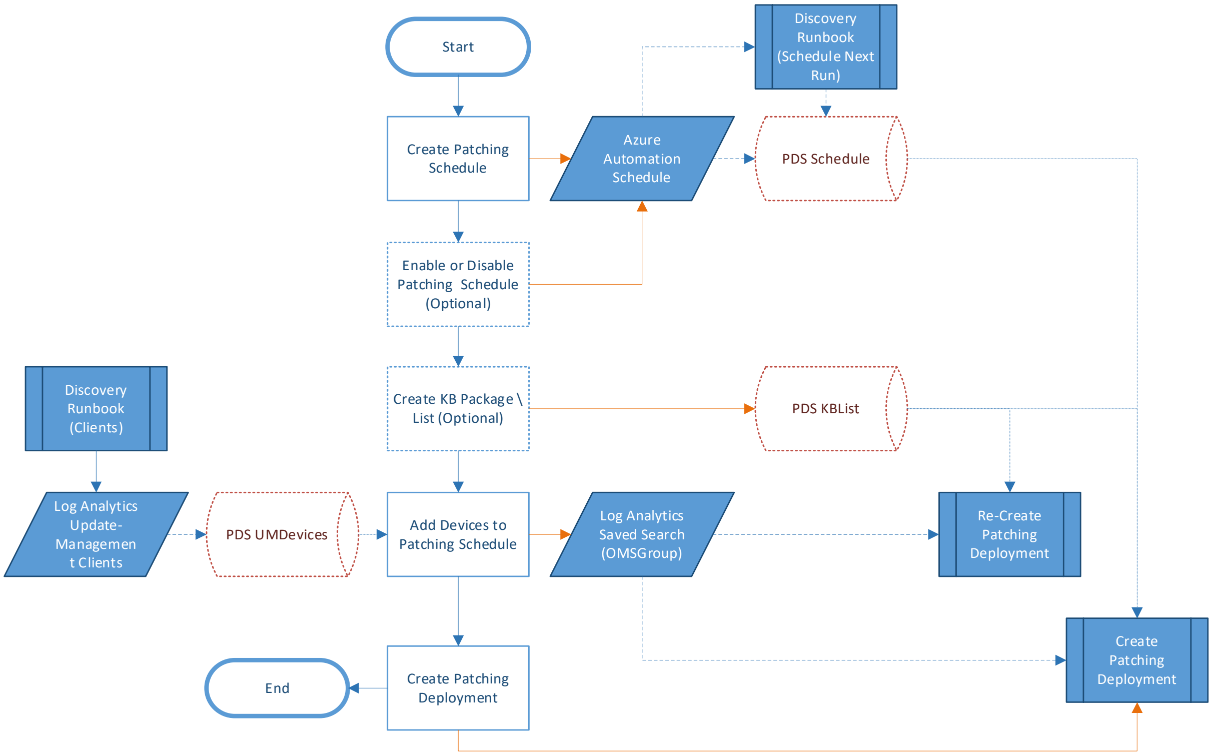
The Kelverion Automated Patching Application for Azure Update Management has several components. By default, the application is installed with the Kelverion Automation Portal that provides a front-end UI that hosts a selection of service offerings. The user will use these offerings to manage the Update Management features in Azure.

The application is built using:

- Azure Automation Runbooks
- Azure SQL Server
- Azure SQL Database(s) (Kelverion Automation Portal \ Persistent Data Store)
- Azure Web App (Kelverion Automation Portal)
- Azure Logic Apps

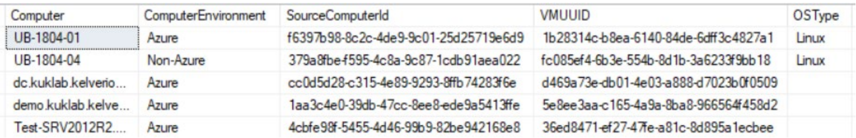
4.1. High level patch process

The diagram below shows the correct process to deploy updates using the Kelverion Automated Patching Application. It also shows the PDS table names and related runbooks.


Start
Discovery Runbook (Schedule Next Run)
Create Patching Schedule
Azure Automation Schedule
PDS Schedule
Enable or Disable Patching Schedule (Optional)
Discovery Runbook (Clients)
Create KB Package \ List (Optional)
PDS KBList
Log Analytics Update-Management Clients
PDS UMDevices
Add Devices to Patching Schedule
Log Analytics Saved Search (OMSGroup)
Re-Create Patching Deployment
Create Patching Deployment
End
Create Patching Deployment

4.2. Client Discovery

The application runs a discovery runbook against the log analytics workspace, to determine which devices have a client configured for Update Management. These devices are added to the PDS table [AzureUM].[UMDevices].



Computer	ComputerEnvironment	SourceComputerId	VMUUID	OSType
UB-1804-01	Azure	f6397b98-8c2c-4de9-9c01-25d25719e6d9	1b28314c-b8ea-6140-84de-6dff3c4827a1	Linux
UB-1804-04	Non-Azure	379a8fbe-f595-4c8a-9c87-1cdb91aea022	fc085ef4-6b3e-554b-8d1b-3a6233f9bb18	Linux
dc.kuklab.kelverio...	Azure	cc0d5d28-c315-4e89-9293-8ffb74283f6e	d469a73e-db01-4e03-a888-d7023b0f0509	
demo.kuklab.kelve...	Azure	1aa3c4e0-39db-47cc-8ee8-ede9a5413ffe	5e8ee3aa-c165-4a9a-8ba8-966564f458d2	
Test-SRV2012R2....	Azure	4cbfe98f-5455-4d46-99b9-82be942168e8	36ed8471-ef27-47fe-a81c-8d895a1ecbee	

The [AzureUM].[UMDevices] also stores which group the device has been added to via the offering ‘Add device to patch schedule’.

Each group is related to a patch schedule. A device can only be a member of a single group (patch schedule).

5. Application Offerings

The Kelverion patching application for Azure Automation comes with several service offerings in the Automation Portal. This section will detail how each of these offerings works and how it interacts with Azure Update Management.

The offerings are split into what are considered administrative and user (device owner) based offerings:

Patch Administrative Offerings:

- Create Patching Schedule
- Create Patching Deployment
- Enable or Disable Patching Schedule
- Create KB \ Package List
- Deploy Update Management Clients
- Remove Patching Deployment

User (Device Owner) Offering:

- Add Devices to Patching Schedule

5.1. Create Patching Schedule

Before you can deploy any updates to client devices, you will need to create at least one schedule. The schedule provides the backbone of the patching deployment.

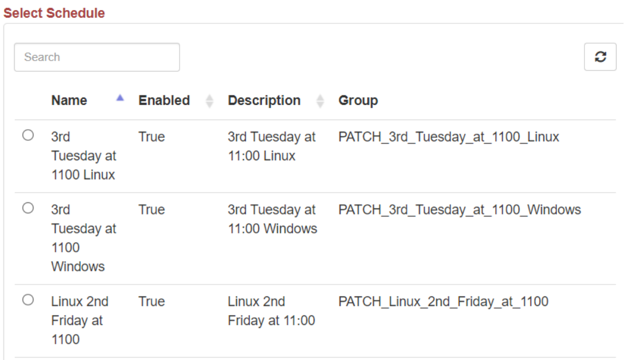
In the Kelverion application the chosen schedule name also becomes the deployment name when it is created. It is advisable to select a name that reflects the parameters chosen, as this will help identify it in reporting:

e.g., **AUTO Windows 3rd Tuesday No Reboot**

Prefixing the name with AUTO will differentiate the schedule from any manually created schedules.

Adding in the OS type helps the end users select the correct schedule to add their chosen devices too. Adding in the chosen day of the month and reboot options that will take place adds further information for the end user.

When you go to add a device to a chosen schedule it will make it easier to find the correct schedule.

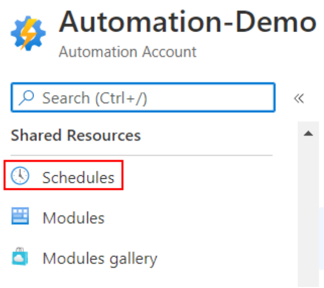


	Name	Enabled	Description	Group
○	3rd Tuesday at 1100 Linux	True	3rd Tuesday at 11:00 Linux	PATCH_3rd_Tuesday_at_1100_Linux
○	3rd Tuesday at 1100 Windows	True	3rd Tuesday at 11:00 Windows	PATCH_3rd_Tuesday_at_1100_Windows
○	Linux 2nd Friday at 1100	True	Linux 2nd Friday at 11:00	PATCH_Linux_2nd_Friday_at_1100

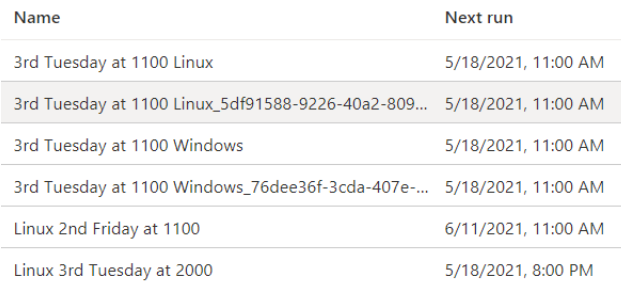
When creating a schedule, you will need to provide the following:

Field Name	Description
Name	The name of the schedule
Description	Optional descriptive field
Start Time	The start time
Day of Week	The day of the week the schedule occurs
Weekly Occurrence	Which week of the month the schedule occurs
Monthly Interval	1 = Every Month. 2 = Every 2 months etc.
Expire	Sets the schedule to expire after it has first run. It will not repeat past the expiry date \ time.
Expiry Date	Required if you set Expire to True
Expiry Time	Required if you set Expire to True
Operating System	Windows or Linux

5.1.1. Azure Schedules



The patching schedule is really a schedule associated with the automation account that has the Update Management feature. An Azure administrator can view these in the automation account.



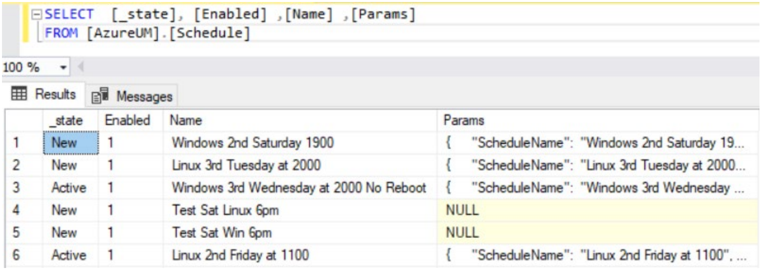
Name	Next run
3rd Tuesday at 1100 Linux	5/18/2021, 11:00 AM
3rd Tuesday at 1100 Linux_5df91588-9226-40a2-809...	5/18/2021, 11:00 AM
3rd Tuesday at 1100 Windows	5/18/2021, 11:00 AM
3rd Tuesday at 1100 Windows_76dee36f-3cda-407e-...	5/18/2021, 11:00 AM
Linux 2nd Friday at 1100	6/11/2021, 11:00 AM
Linux 3rd Tuesday at 2000	5/18/2021, 8:00 PM

In this example, you can see that there are duplicates of various schedule names with a GUID appended on the end. The schedules with a GUID have an associated patching deployment and are created automatically when a patch deployment is created.

N.B. You must not delete the schedules with a GUID as it will break the patch deployment job.

5.1.2. PDS AzureUM.Schedule

When a patching schedule is created, it is stored into the PDS table [AzureUM].[Schedule]



```
SELECT  [_state], [Enabled] ,[Name] ,[Params]
FROM [AzureUM].[Schedule]
```

	_state	Enabled	Name	Params
1	New	1	Windows 2nd Saturday 1900	{ "ScheduleName": "Windows 2nd Saturday 19...
2	New	1	Linux 3rd Tuesday at 2000	{ "ScheduleName": "Linux 3rd Tuesday at 2000...
3	Active	1	Windows 3rd Wednesday at 2000 No Reboot	{ "ScheduleName": "Windows 3rd Wednesday ...
4	New	1	Test Sat Linux 6pm	NULL
5	New	1	Test Sat Win 6pm	NULL
6	Active	1	Linux 2nd Friday at 1100	{ "ScheduleName": "Linux 2nd Friday at 1100", ...

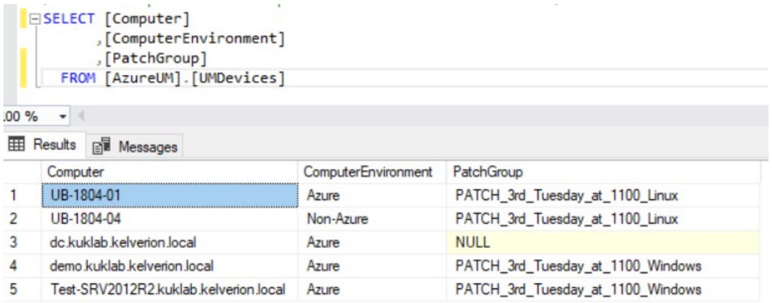
5.2. Add Devices to Patching Schedule

To use a Patching Schedule in a Patching Deployment you must have added some devices to it. For the Kelverion application we are simply adding the devices to a Log Analytics Saved Search, that is associated with our schedule.

A device can only be a member of a single Patching Schedule. When this offering runs, it will remove a device from any previous schedule and add it to the newly selected schedule.

5.2.1. PDS AzureUM Devices – Group

A SQL administrator can view the database to see which device is in which group.



```
SELECT [Computer]
      ,[ComputerEnvironment]
      ,[PatchGroup]
  FROM [AzureUM].[UMDevices]
```

	Computer	ComputerEnvironment	PatchGroup
1	UB-1804-01	Azure	PATCH_3rd_Tuesday_at_1100_Linux
2	UB-1804-04	Non-Azure	PATCH_3rd_Tuesday_at_1100_Linux
3	dc.kuklab.kelverion.local	Azure	NULL
4	demo.kuklab.kelverion.local	Azure	PATCH_3rd_Tuesday_at_1100_Windows
5	Test-SRV2012R2.kuklab.kelverion.local	Azure	PATCH_3rd_Tuesday_at_1100_Windows

If you wish to see this without using SSMS, then it is advisable to use this data via a PowerBI report to view the membership of each chosen schedule.

5.2.2. Determine Device Owners

The list of devices listed in this offering can be filtered based on the logged-on user. This allows you to only allow users to see their selected list of devices associated with their team.

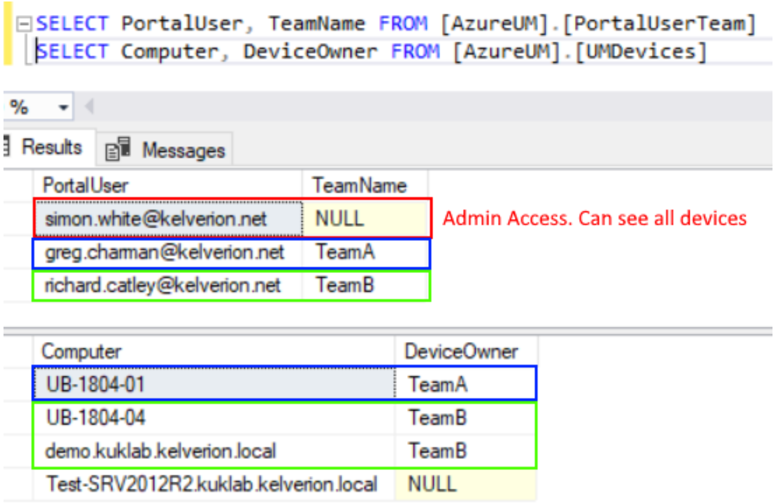
This is managed through two tables in the PDS.

- [AzureUM].[PortalUserTeam]
- [AzureUM].[UMDevices]

Table [PortalUserTeam], defines the account that is used for the Kelverion Automation Portal with the users specified team.

Table [UMDevices], is populated by discovery, but will have an empty column called [DeviceOwner] that will need to match the required team name.

Any user that does not have a defined TeamName in [PortalUserTeam], will be able to see all the devices.



```
SELECT PortalUser, TeamName FROM [AzureUM].[PortalUserTeam]
SELECT Computer, DeviceOwner FROM [AzureUM].[UMDevices]
```

PortalUser	TeamName
simon.white@kelverion.net	NULL
greg.charman@kelverion.net	TeamA
richard.catley@kelverion.net	TeamB

Admin Access. Can see all devices

Computer	DeviceOwner
UB-1804-01	TeamA
UB-1804-04	TeamB
demo.kuklab.kelverion.local	TeamB
Test-SRV2012R2.kuklab.kelverion.local	NULL

5.3. Create Patching Deployment

A patching deployment will require the following:

- An enabled schedule
- Target device(s)

If you have not created a schedule or added devices to that schedule, then this offering will fail. Azure Automation Update Management cannot create a patching deployment without target devices. Please see the previous sections on how to create a schedule and add devices to it.

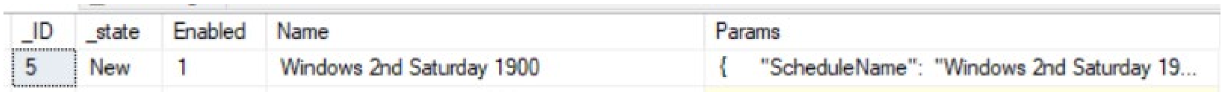
When creating a patching deployment, you will need to provide the following:

Field Name	Description
Schedule	Select from the chosen list of created schedules
Duration	This is the maintenance window for the patching. At least 20mins is reserved for any reboots. If patching takes longer than expected and there is less than 20mins left, the device will not reboot. This has a maximum value of 6hrs.

Patching Classifications	Select which classification of updates are to be deployed
Include KBs	Select a pre-created list of updates that must be deployed
Exclude KBs	Select a pre-created list of updates that must not be deployed
Reboot Setting	Decide if the devices should reboot after patching. See notes above about the 'Duration' if you have a lot of updates to deploy.

N.B. If you create a second patching deployment for a schedule that already has a deployment, then the application will remove the old patching deployment and replace it with the new deployment.

When the runbook runs it will create a deployment straight away and update the PDS table [AzureUM].[Schedules] with the deployment parameters.



_ID	_state	Enabled	Name	Params
5	New	1	Windows 2nd Saturday 1900	{ "ScheduleName": "Windows 2nd Saturday 19...

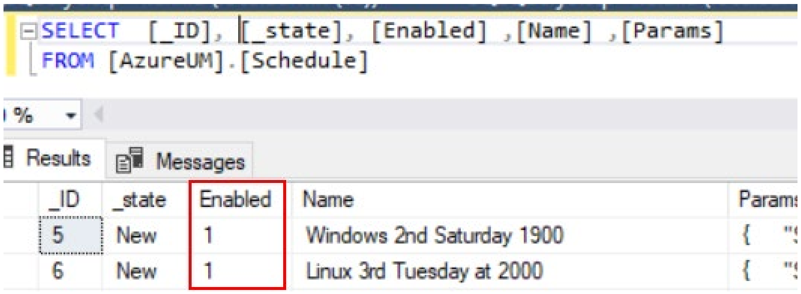
At 1 hour before the patching deployment is due to run a runbook is called via a 'Logic App' to re-create the patching deployment. This is done to ensure that the most up to date group membership is used in the deployment.

5.4. Enable or Disable Patching Schedule

Each schedule can be enabled or disabled. When a schedule is created it is automatically enabled. You may however wish to remove a schedule option temporarily from the end users. To do this you can use this offering to disable a schedule.

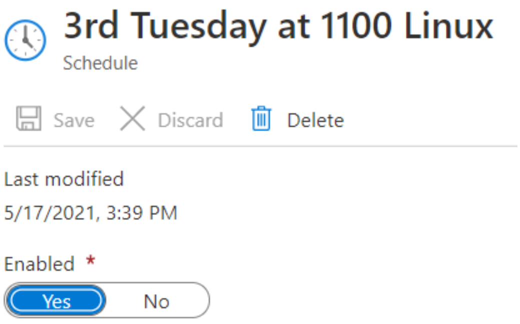
N.B. Disabling a schedule does not remove or move any of the associated devices that are added to its group.

The offering will update the Azure schedule in the automation account and update the PDS table [AzureUM].[Schedule].



```
SELECT [_ID], [_state], [Enabled] ,[Name] ,[Params]
FROM [AzureUM].[Schedule]
```

_ID	_state	Enabled	Name	Params
5	New	1	Windows 2nd Saturday 1900	{ "
6	New	1	Linux 3rd Tuesday at 2000	{ "



3rd Tuesday at 1100 Linux

Schedule

Last modified
5/17/2021, 3:39 PM

Enabled *
Yes No

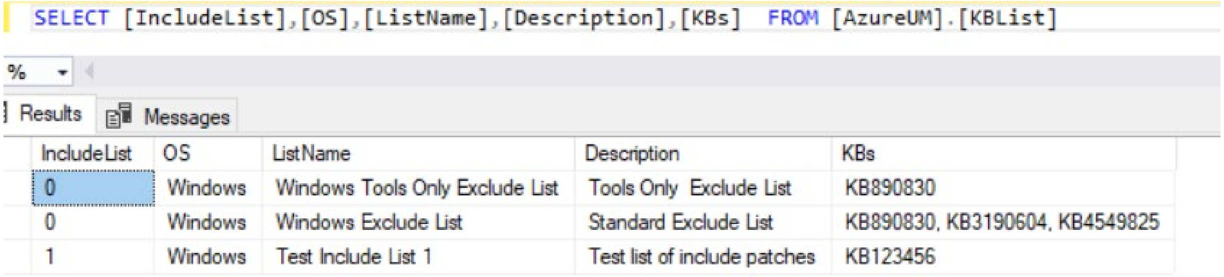
Each schedule enable \ disable status can also be viewed in the Azure portal.

5.5. Create KB \ Package List

This offering allows you to create a pre-defined comma separated list of updates to include or exclude from a patching deployment.

5.5.1. PDS AzureUM.KBList

The KB \ Package lists are stored in the PDS table [AzureUM].[KBList].



```
SELECT [IncludeList],[OS],[ListName],[Description],[KBs]  FROM [AzureUM].[KBList]
```

IncludeList	OS	ListName	Description	KBs
0	Windows	Windows Tools Only Exclude List	Tools Only Exclude List	KB890830
0	Windows	Windows Exclude List	Standard Exclude List	KB890830, KB3190604, KB4549825
1	Windows	Test Include List 1	Test list of include patches	KB123456

5.6. Remove Patching Deployment

This offering allows you to remove a patching deployment. The offering will only show you schedules that have previously had a deployment created for them.

Once the runbook completes the schedule will be removed from the list, as the stored deployment parameters are removed from the PDS table [AzureUM].[Schedules].

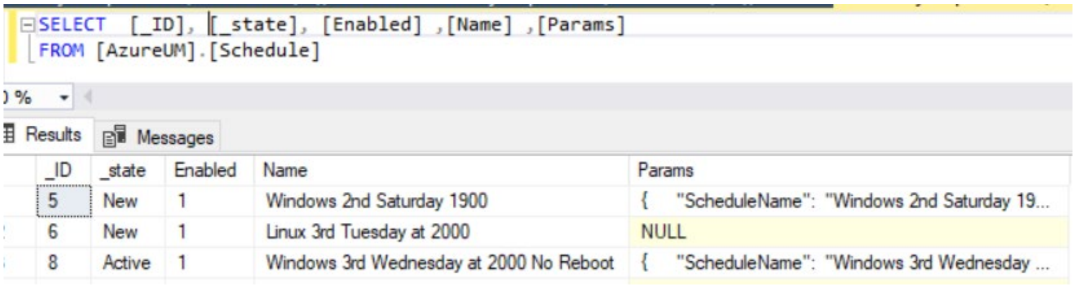
The [AzureUM].[Schedules] table below shows the differing states for each Schedule:

e.g.

_ID = 5 “Windows 2nd Saturday 1900” has a deployment that is due to run in the future.

_ID = 6 “Linux 3rD Tuesday at 2000” has NULL parameters, so has no deployments created for it

_ID = 8 “Windows 3rd Wednesday at 2000 No Reboot” has a _state = “Active”, so it is either going to run within the hour or has just run.



```
SELECT  [_ID], [_state], [Enabled] ,[Name] ,[Params]
FROM [AzureUM].[Schedule]
```

_ID	_state	Enabled	Name	Params
5	New	1	Windows 2nd Saturday 1900	{ "ScheduleName": "Windows 2nd Saturday 19...
6	New	1	Linux 3rd Tuesday at 2000	NULL
8	Active	1	Windows 3rd Wednesday at 2000 No Reboot	{ "ScheduleName": "Windows 3rd Wednesday ...

5.7. Deploy Update Management Clients

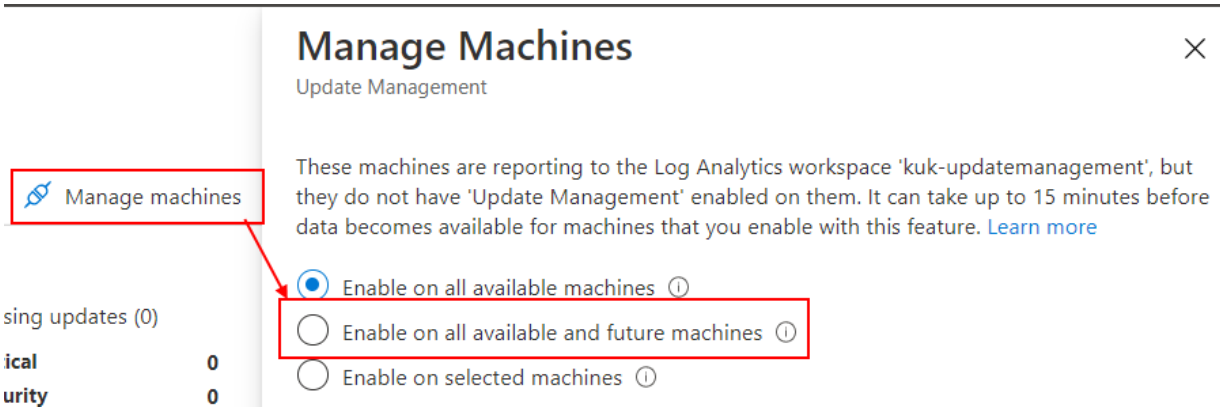
This offering will automatically deploy the Update Management client to a specified resource group of VMs in Azure.

The client is the Microsoft Monitoring Agent (MMA) that is linked to the Log Analytics workspace for Update Management. If the device already has an MMA agent installed, this offering will add the Log Analytics workspace to that agent.

N.B. Devices need to be powered on for this offering to complete.

5.7.1. How do I automatically enable clients for Update Management in Azure?

In the Azure portal for Update Management, there is a 'Manage Machines' option. You can select an option to 'Enable on all available and future machines'.



Manage Machines

Update Management

These machines are reporting to the Log Analytics workspace 'kuk-updatemanagement', but they do not have 'Update Management' enabled on them. It can take up to 15 minutes before data becomes available for machines that you enable with this feature. Learn more

● Enable on all available machines ⓘ

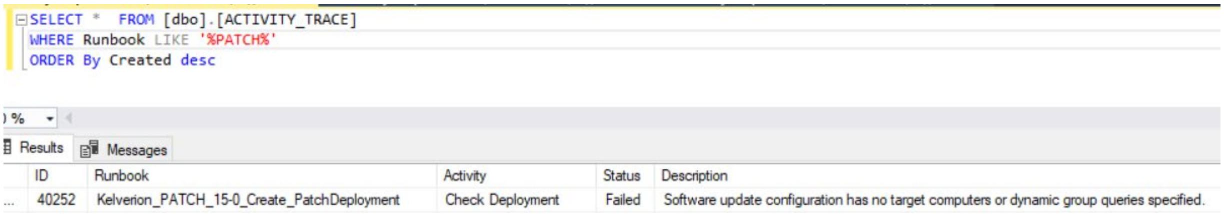
○ Enable on all available and future machines ⓘ

○ Enable on selected machines ⓘ

6. Troubleshooting

6.1. Application error logging

The runbooks are designed to pick up errors when activities fail. When an activity does fail it will attempt to write the error to the PDS table [dbo].[ACTIVITY_TRACE].



```
SELECT *  FROM [dbo].[ACTIVITY_TRACE]
WHERE Runbook LIKE '%PATCH%'
ORDER By Created desc
```

	ID	Runbook	Activity	Status	Description
...	40252	Kelverion_PATCH_15-0_Create_PatchDeployment	Check Deployment	Failed	Software update configuration has no target computers or dynamic group queries specified.

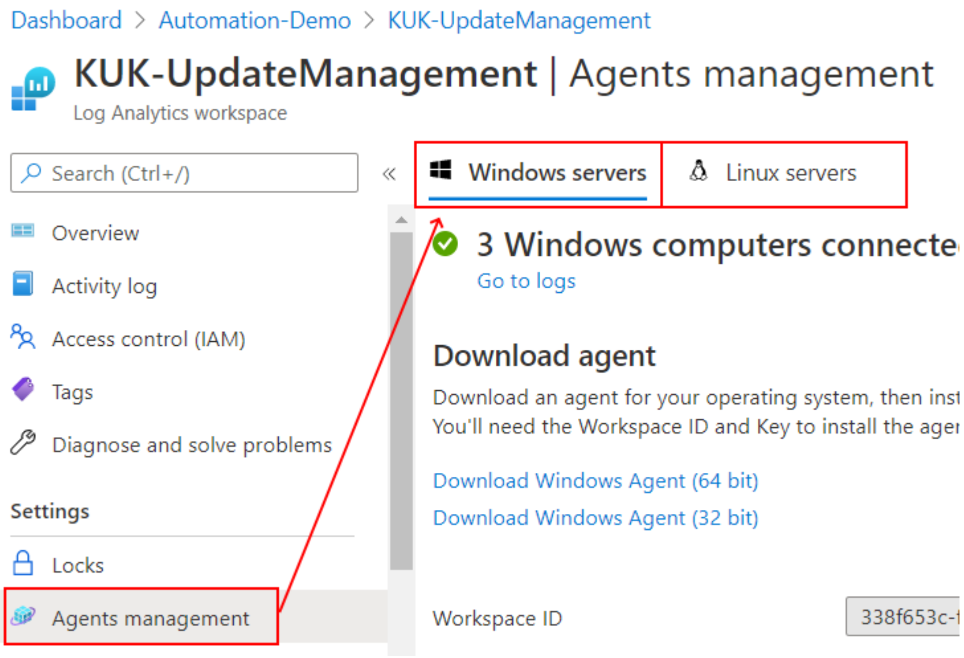
6.2. How do I deploy a client to a Non-Azure device?

Deploying clients that are not in a resource group (e.g. Non-Azure devices) will require administrative access to the device(s).

Microsoft provide a detailed guide here on installing the client: https://docs.microsoft.com/en-gb/azure/azure-monitor/agents/log-analytics-agent#installation-options

You can also go to your Log Analytics Workspace in Azure and browse to the ‘Agents management’ section.

Here you can download clients for your Windows servers:



Dashboard > Automation-Demo > KUK-UpdateManagement
KUK-UpdateManagement | Agents management
Log Analytics workspace
Search (Ctrl+/)
«
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Locks
Agents management
Windows servers
Linux servers
Go to logs
Download Windows Agent (64 bit)
Download Windows Agent (32 bit)

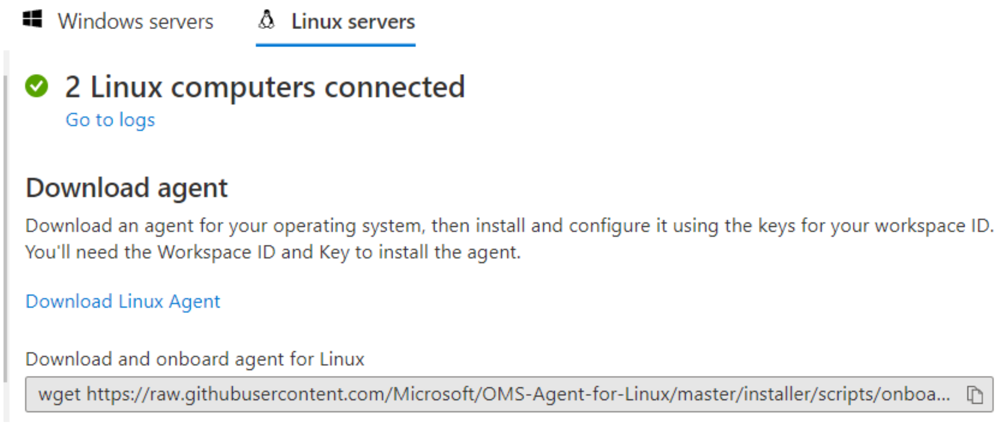
3 Windows computers connecte

Download agent

Download an agent for your operating system, then inst
You'll need the Workspace ID and Key to install the ager

Workspace ID 338f653c-

Here you can find the correct command for your Linux servers:



2 Linux computers connected

Go to logs

Download agent

Download an agent for your operating system, then install and configure it using the keys for your workspace ID. You'll need the Workspace ID and Key to install the agent.

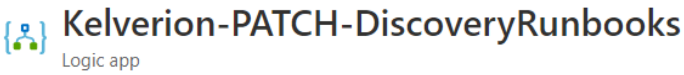
Download Linux Agent

Download and onboard agent for Linux

```
wget https://raw.githubusercontent.com/Microsoft/OMS-Agent-for-Linux/master/installer/scripts/onboa...
```

6.3. I cannot see a device to select in the Automation Portal.

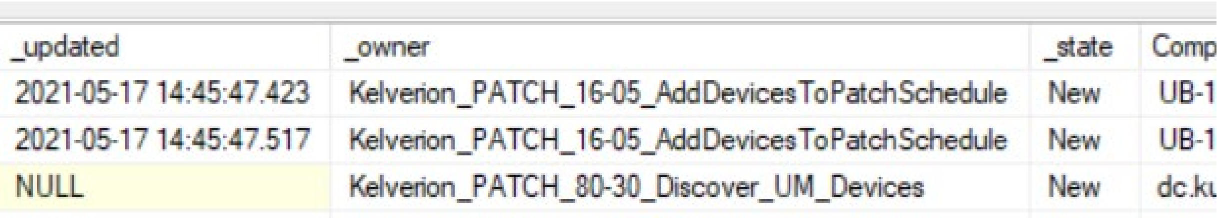
Devices are discovered by a discovery runbook launched from a ‘Logic App’. By default, this will run on a daily schedule.



Kelverion-PATCH-DiscoveryRunbooks

Logic app

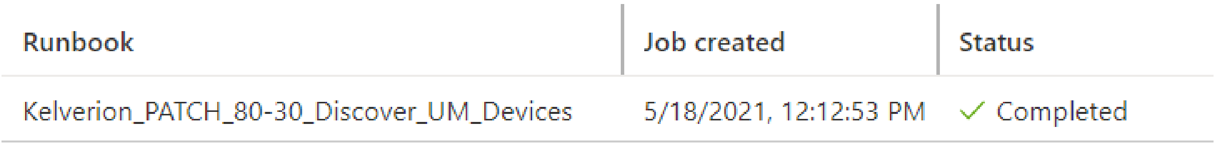
The discovery runbook will query the associated Log Analytics workspace and gather any updated client information. This will then be added to the [AzureUM].[UMDevices] table in the PDS.



_updated	_owner	_state	Comp
2021-05-17 14:45:47.423	Kelverion_PATCH_16-05_AddDevicesToPatchSchedule	New	UB-1
2021-05-17 14:45:47.517	Kelverion_PATCH_16-05_AddDevicesToPatchSchedule	New	UB-1
NULL	Kelverion_PATCH_80-30_Discover_UM_Devices	New	dc.ku

Newly added devices will have the _owner set to the discovery runbook. First check here to see if the device has been added.

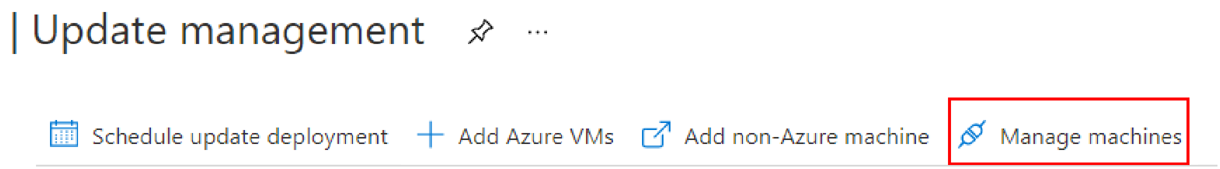
You can check the automation account to see if the discovery runbook has run and completed with any errors:



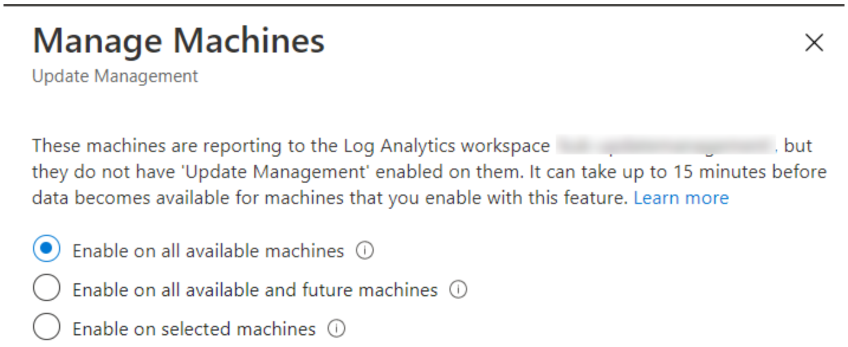
Runbook	Job created	Status
Kelverion_PATCH_80-30_Discover_UM_Devices	5/18/2021, 12:12:53 PM	✓ Completed

If it has not run, then check the Logic App is enabled and functioning correctly.

You can also check the Update Management UI and select “Manage machines”.



This allows your Azure Administrator to set how clients with a correctly configured MMA can connect to Update Management.

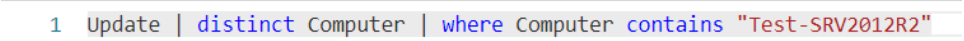

Manage Machines

Update Management

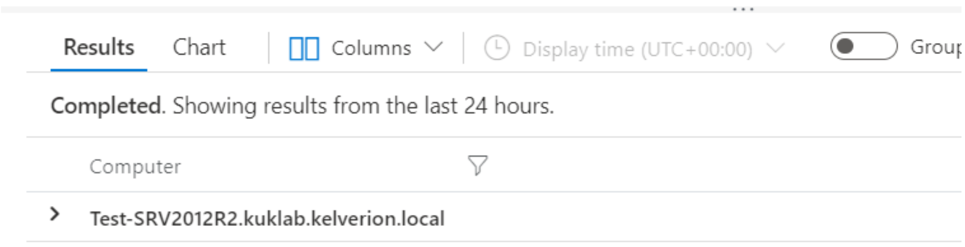
These machines are reporting to the Log Analytics workspace [redacted], but they do not have 'Update Management' enabled on them. It can take up to 15 minutes before data becomes available for machines that you enable with this feature. Learn more

● Enable on all available machines
○ Enable on all available and future machines
○ Enable on selected machines

You can also directly query the log analytics workspace for the chosen client name:



```
Update | distinct Computer | where Computer contains "Test-SRV2012R2"
```



Completed. Showing results from the last 24 hours.

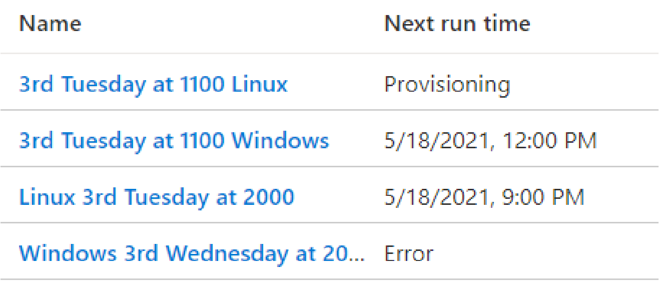
Computer
Test-SRV2012R2.kuklab.kelverion.local

6.4. Why has my patch deployment not run?

There are several reasons why a patch deployment may not have run.

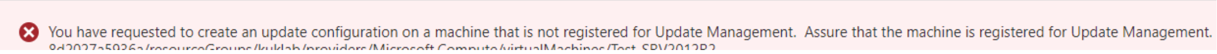
Issue: Deployment Error:

You can ask your Azure Administrator to check the Update Management deployments for any errors:



Name	Next run time
3rd Tuesday at 1100 Linux	Provisioning
3rd Tuesday at 1100 Windows	5/18/2021, 12:00 PM
Linux 3rd Tuesday at 2000	5/18/2021, 9:00 PM
Windows 3rd Wednesday at 20...	Error

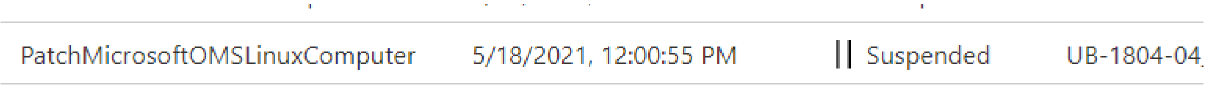
If there is an Error, then clicking on the deployment will provide more information:



You have requested to create an update configuration on a machine that is not registered for Update Management. Assure that the machine is registered for Update Management.

Generally, the deployments fail when they have no clients to connect to. At the time of provisioning the deployment, Update Management will need to connect the client device to setup the patch deployment runbook.

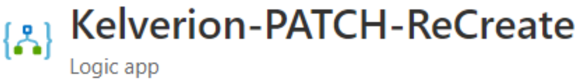
You may see the following in the automation account job log when a targeted device is not currently available.



PatchMicrosoftOMSLinuxComputer 5/18/2021, 12:00:55 PM Suspended UB-1804-04

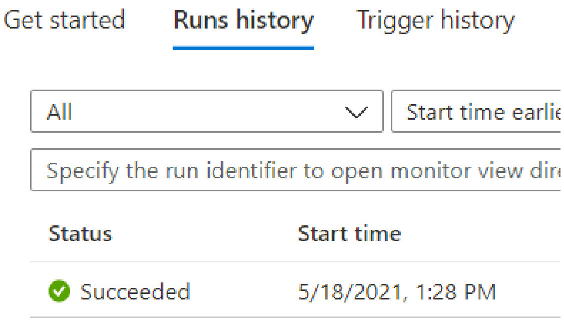
Issue: Logic App Failure:

1 hour before a patch deployment is to run, the application will re-create the deployment to ensure that the correct set of target devices have been added. This re-creation is driven from a ‘Logic App’.



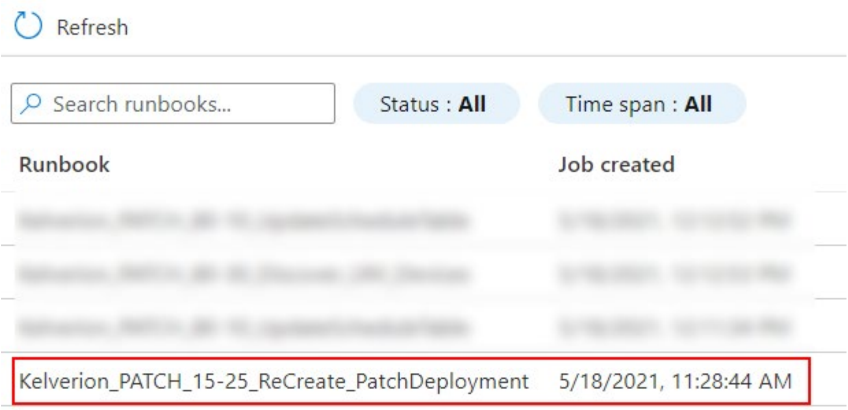
Kelverion-PATCH-ReCreate
Logic app

You should check that this ‘Logic App’ is enabled. You can also check the run history to see when it last triggered.



Status	Start time
✓ Succeeded	5/18/2021, 1:28 PM

This should be launching the Patch Re-Create runbook. You can check the status of this in the jobs log of the automation account.

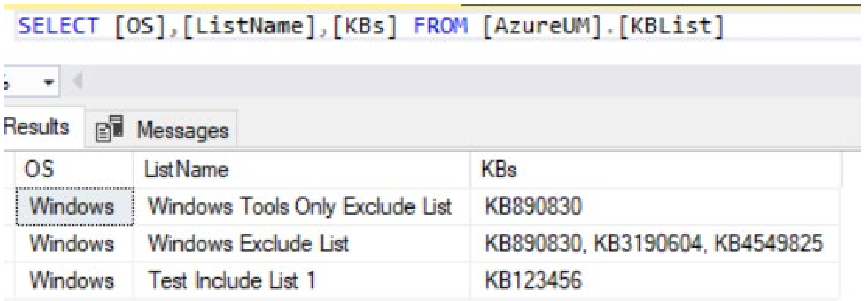


Runbook	Job created
Kelverion_PATCH_15-25_ReCreate_PatchDeployment	5/18/2021, 11:28:44 AM

Issue: Incorrectly created KB list:

It is optional to add an include or exclude list to a patch deployment. You should check to make sure that the updates included in your comma separated list are valid.

You can check this in the PDS table [AzureUM].[KBList]



```
SELECT [OS],[ListName],[KBs] FROM [AzureUM].[KBList]
```

OS	ListName	KBs
Windows	Windows Tools Only Exclude List	KB890830
Windows	Windows Exclude List	KB890830, KB3190604, KB4549825
Windows	Test Include List 1	KB123456

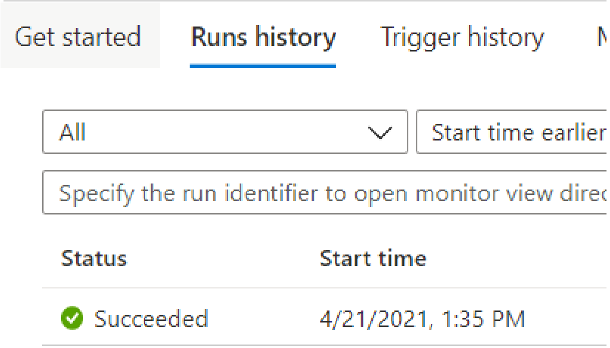
6.5. Why has my request not progressed from ‘New’?

The Kelverion Automation Portal requests are monitored by an Azure ‘Logic App’ called **Kelverion-Patch-PortalIntegration**. [1]

The Logic App must be enabled to allow it to check the backend database for new requests. By default, new requests are checked every minute.

This then calls a runbook that updates the status of the request in the portal.

You can check the Logic App trigger history and the corresponding runbook **Kelverion_PATCH_90-01_KAP_Get_Request** that is called in the automation account:



Status	Start time
Succeeded	4/21/2021, 1:35 PM

N.B: The Logic App will only trigger when it detects a new row in the database. It is this check that is done every 1 minute. You should not expect to see a trigger log for every minute.

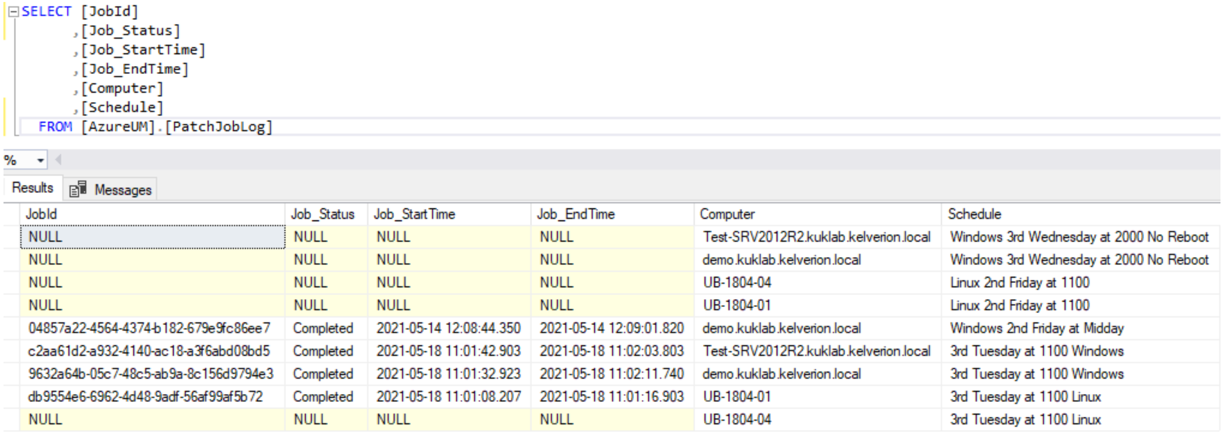
Notes:

1: If you have multiple service offerings in your Kelverion Automation Portal, you may have a differently name ‘Logic App’ that is collecting data from all offerings.

6.6. How do I know if a patch job ran on a device?

The application has a discovery runbook that will check for patch jobs, up to 3hrs after a patch maintenance window has ended.

A Status of Completed or Failed, will be logged back to the PDS table [AzureUM].[PatchJobLogs] for any detected device. If a device is not powered on during the maintenance window, then there will be no patch job logged for it.



```sql
SELECT [JobId]
      ,[Job_Status]
      ,[Job_StartTime]
      ,[Job_EndTime]
      ,[Computer]
      ,[Schedule]
  FROM [AzureUM].[PatchJobLog]
```

JobId	Job_Status	Job_StartTime	Job_EndTime	Computer	Schedule
NULL	NULL	NULL	NULL	Test-SRV2012R2.kuklab.kelverion.local	Windows 3rd Wednesday at 2000 No Reboot
NULL	NULL	NULL	NULL	demo.kuklab.kelverion.local	Windows 3rd Wednesday at 2000 No Reboot
NULL	NULL	NULL	NULL	UB-1804-04	Linux 2nd Friday at 1100
NULL	NULL	NULL	NULL	UB-1804-01	Linux 2nd Friday at 1100
04857a22-4564-4374-b182-679e9fc86ee7	Completed	2021-05-14 12:08:44.350	2021-05-14 12:09:01.820	demo.kuklab.kelverion.local	Windows 2nd Friday at Midday
c2aa61d2-a932-4140-ac18-a3f6abd08bd5	Completed	2021-05-18 11:01:42.903	2021-05-18 11:02:03.803	Test-SRV2012R2.kuklab.kelverion.local	3rd Tuesday at 1100 Windows
9632a64b-05c7-48c5-ab9a-8c156d9794e3	Completed	2021-05-18 11:01:32.923	2021-05-18 11:02:11.740	demo.kuklab.kelverion.local	3rd Tuesday at 1100 Windows
db9554e6-6962-4d48-9adf-56af99af5b72	Completed	2021-05-18 11:01:08.207	2021-05-18 11:01:16.903	UB-1804-01	3rd Tuesday at 1100 Linux
NULL	NULL	NULL	NULL	UB-1804-04	3rd Tuesday at 1100 Linux

This data can be displayed using a PowerBI dashboard.

6.7. How is a device compliance worked out?

Update Management scans managed machines for data using the following rules. It can take between 30 minutes and 6 hours for the dashboard to display updated data from managed machines.

- Each Windows machine - Update Management does a scan twice per day for each machine.
- Each Linux machine - Update Management does a scan every hour.

The average data usage by Azure Monitor logs for a machine using Update Management is approximately 25 MB per month. This value is only an approximation and is subject to change, depending on your environment. We recommend that you monitor your environment to keep track of your exact usage.

The scan is dependent on the upstream server that your device is pointing too. If you have a Group Policy in place, that is pointing your devices at an internal WSUS server, then it will determine its compliance from that server.

Unit 31, Thrales End Business Centre
Thrales End Lane
Harpenden
Hertfordshire
AL5 3NS
Email: info@kelverion.com
Web: www.kelverion.com