

# AutoMonX Sensor Pack for Microsoft Azure

Date	Change	Author
20.01.2019	Initial Release	AutoMonX
04.07.2019	Updated to cover the latest features	AutoMonX
09.08.2019	Updated to cover the latest features, how to get your Azure credentials appendix	AutoMonX
05.09.2019	Updated troubleshooting, new billing update functionality, Firewall and anti-virus requirements, License handling	AutoMonX
05.11.2019	Updated new UI functionality	AutoMonX
06.01.2020	Added the upgrade procedure	AutoMonX
06.02.2020	Added Microsoft Insights monitoring	AutoMonX
28.04.2020	Updated UI, Billing sensor updates	AutoMonX
15.05.2020	Updates of troubleshooting, debugging and upgrade sections	AutoMonX
26.05.2020	Updated Upgrade procedure, Monitoring Automation Timeout handling, Performance improvements	AutoMonX
08.06.2020	Supported platforms, New debug capabilities	AutoMonX
12.09.2020	Added support for PRTG native SNMP auto-discovery for Azure Virtual Machines. Support for new Sensor types (Service Plan and Logic App), Support for PRTG Cloud. Refreshed UI. Windows 2008R2 servers are no longer supported, Introduction of Hybrid Sensors	AutoMonX
24.01.2021	<b>Multi-Tenant</b> support, 8 new sensor types, UI Updates, Connection Profiles encryption, Upgrade procedures	AutoMonX
03.07.2021	App Secret Key sensor added	AutoMonX
04.08.2021	Azure Kubernetes (AKS) Sensor was added. Our UI now requires PRTG username and password instead of passhash.	AutoMonX
23.09.2021	Updated security requirements	AutoMonX
24.03.2022	Azure VM Multi-Disk sensor was added.	AutoMonX

<b>22.07.2022</b>	New integration with PRTG, New discovery modes, Exclude sensors functionality, auto-deletion functionality	AutoMonX
<b>13.10.2022</b>	New upgrade mode with installer.	AutoMonX
<b>22.01.2023</b>	Multiple granular discovery options, Discovery and Automation can use Azure tags to add relevant resources to PRTG, New CLI options, configuration values, REST API service, SQL Elastic Pools sensor	AutoMonX
<b>24.04.2023</b>	Multiple per subscription sensors introduced, as well as File Share and Host Pools sensors. New monitoring service for the AutoMonX processes.	AutoMonX
<b>18.10.2023</b>	Added certificates for app registrations, Memory channels added to license sensor, UI added test connection button to PRTG, new sensors: Alerts, AKS Upgrade Profiles, Topic Subscriptions, Auto Discovery speed improvements	AutoMonX
<b>06.11.2023</b>	Added support for 7 new sensor types: Firewalls, Network Interfaces, Connections, Virtual Network Gateways, WAFs, Load Balancers, Network Watchers	AutoMonX
<b>28.07.2024</b>	Added support for sensors: SAML Certificate expiration, Site to Site VPN state. Azure Gov support	AutoMonX
<b>03.11.2024</b>	Updated KQL sensor appendix	AutoMonX
<b>01.10.2025</b>	Multiple new sensors added, evaluation automatic license features, include/exclude functionality, API updates	AutoMonX
<b>18.12.25</b>	Added support for Key Vault Certificates and Keys resource	AutoMonX
<b>22.1.25</b>	Fixed Discovery crash, Improved logging of Key Vault	AutoMonX

## Table of Contents

<b>1</b>	<b>PURPOSE</b>	<b>8</b>
<b>2</b>	<b>AUTOMONX SENSOR PACK FOR MICROSOFT AZURE OVERVIEW</b>	<b>8</b>
<b>3</b>	<b>HOW DOES IT WORK?</b>	<b>9</b>
3.1	THE AZURE SENSOR PACK ARCHITECTURE	10
3.2	THE AZURE SENSOR PACK – INTEGRATION WITH PRTG	10
3.3	THE AZURE SENSOR PACK – ESSENTIAL TERMINOLOGY	11
<b>4</b>	<b>GETTING STARTED WITH AZURE SENSOR PACK</b>	<b>12</b>
4.1	SUPPORTED SOFTWARE VERSIONS	12
4.2	AZURE SENSOR PACK - PORT REQUIREMENTS	12
4.3	AZURE SENSOR PACK – ANTI-VIRUS REQUIREMENTS	13
4.4	DOWNLOADING THE AZURE SENSOR PACK	13
4.5	INSTALLING THE AZURE SENSOR PACK	14
4.6	THE DIRECTORY STRUCTURE OF THE AZURE SENSOR PACK	18
4.7	LOOKUP FILE HANDLING (ON-PREM PRTG)	22
4.8	LOOKUP FILE HANDLING (PRTG CLOUD)	23
4.9	REQUESTING A LICENSE FOR THE AUTOMONX SENSOR PACK FOR AZURE	26
4.10	ACTIVATING THE AZURE SENSOR PACK LICENSE	27
<b>5</b>	<b>AZURE SENSOR PACK CONFIGURATION</b>	<b>29</b>
5.1	PREPARING FOR CONFIGURING THE AUTOMONX SENSOR PACK FOR AZURE	29
5.2	PROXY SERVER CONNECTION CONFIGURATION	30
5.3	CONFIGURING THE AUTOMONX SENSOR PACK FOR AZURE	30
5.4	AZURE SENSOR PACK - CONFIGURATION CHECK	34
5.5	AZURE SENSOR PACK – UPGRADE INSTRUCTIONS	35
5.6	AZURE SENSOR PACK SERVICE – INI CONFIG	40
5.7	CONFIGURING AZURE API ENDPOINTS	43
5.8	ENABLING HTTPS SENSORS	44
<b>6</b>	<b>INTRODUCING MULTI-TENANT AZURE MONITORING</b>	<b>45</b>
6.1	MULTI-TENANT LICENSE TYPES EXPLAINED	45
6.2	CONFIGURING MULTI-TENANT DISCOVERY	46
6.3	ENCRYPTION OF CONNECTION PROFILE DETAILS	47
6.4	THE AZURE SENSOR HIERARCHY IN PRTG	47
6.5	MIGRATION TO MULTI-TENANT VERSION	49
<b>7</b>	<b>AUTO DISCOVERY AND MONITORING AUTOMATION</b>	<b>51</b>
7.1	AUTOMATIC DISCOVERY OF AZURE RESOURCES	51
7.2	PREVIOUS DISCOVERY RESULTS HANDLING	52
7.3	SENSOR TYPES CREATED BY THE AZURE SENSOR PACK	53
7.4	SELECTING AZURE SENSORS FOR MONITORING	54
7.5	CONFIGURING PRTG GROUP SETTINGS	56

7.6	AUTO-DISCOVERY OF AZURE VIRTUAL MACHINES SNMP SENSORS	57
7.7	PREPARING FOR AUTO-DISCOVERY OF AZURE VIRTUAL MACHINES	57
7.8	RUNNING AUTO-DISCOVERY OF AZURE VIRTUAL MACHINES	59
7.9	AUTOMATICALLY ADDING AZURE SENSORS TO PRTG	60
7.10	RESUMING ADDING SENSORS IN CASE OF TIMEOUTS	61
7.11	AUTOMATIC DISCOVERY OF AZURE APPLICATION INSIGHTS	62
7.12	USING CLI TO DISCOVER AZURE RESOURCES	65
7.13	AZURE SENSOR PACK DISCOVERY MODES EXPLAINED	66
7.14	AZURE RESOURCES DISCOVERY – CLI OPTIONS FOR THOROUGH DISCOVERY	67
7.15	QUICK DISCOVERY CLI OPTIONS	67
7.16	AZURE RESOURCES DISCOVERY REPORT	68
7.17	AZURE DELTA DISCOVERY REPORT	69
7.18	AZURE CHANNEL LIMITS REPORT	69
7.19	MONITORING AUTOMATION FILES	69
7.20	USING THE MONITORING AUTOMATION CLI	69
7.21	ADDING THE AZURE RESOURCE URL TO PRTG COMMENTS	70
7.22	MULTI-TENANT DISCOVERY SCHEDULER	70

## **8 SUPPORTED SENSOR TYPES ..... 71**

8.1	SQL DATABASE	71
8.2	SQL ELASTIC POOLS	71
8.3	CERTIFICATE	72
8.4	APP SERVICE PLAN (SERVER FARMS)	72
8.5	WEB SITES	73
8.6	STORAGE ACCOUNTS	73
8.7	CLOUD SERVICES	74
8.8	DATABASE ACCOUNTS	74
8.9	SERVICE BUS NAMESPACE	75
8.10	BATCH ACCOUNTS	75
8.11	REDIS	76
8.12	DATA FACTORY	76
8.13	SCHEDULER (DEPRECATED)	77
8.14	SEARCH SERVICES	77
8.15	SERVICE BUS QUEUES	78
8.16	SERVICE HEALTH	78
8.17	VIRTUAL MACHINES	79
8.18	AZURE BILLING	80
8.19	AZURE APPLICATION INSIGHTS	80
8.20	AUTOMONX LICENSE	81
8.21	AZURE LOGIC APP	81
8.22	AZURE DISKS	82
8.23	AZURE NOTIFICATION HUBS	82
8.24	AZURE NETWORK INTERFACES	83
8.25	AZURE EXPRESSROUTE CONNECTIONS	83
8.26	AZURE VPN GATEWAYS METRICS	84

8.27	AZURE BACKUP JOBS	84
8.28	AZURE APP SECRET KEY AND/OR CERTIFICATE STATUS	85
8.29	AZURE KUBERNETES SERVICES (AKS)	86
8.30	AKS CLUSTER METRICS	86
8.31	AKS DEPLOYMENTS	86
8.32	AKS INTERNAL DEPLOYMENTS	87
8.33	AZURE VM MULTI-DISK	87
8.34	AZURE STORAGE FILE SHARE	87
8.35	AZURE SERVICE ISSUES	88
8.36	AZURE PLANNED MAINTENANCE	88
8.37	AZURE HEALTH ADVISORIES	89
8.38	AZURE ADVISOR	89
8.39	AZURE DEFENDER	89
8.40	AZURE QUOTAS	90
8.41	AZURE HOST POOL	90
8.42	AZURE RESERVATIONS	91
8.43	AZURE ALERTS	91
8.44	AKS UPGRADE PROFILES	92
8.45	AZURE SERVICE BUS TOPIC SUBSCRIPTIONS	92
8.46	AZURE FIREWALLS	93
8.47	AZURE APPLICATION GATEWAY (WAF)	93
8.48	AZURE (VPN) CONNECTIONS	94
8.49	IP ADDRESS	95
8.50	AZURE LOAD BALANCER	95
8.51	AZURE NETWORK WATCHER (DEPRECATED)	96
8.52	AZURE LOG ANALYTICS CUSTOM SENSOR	96
8.53	AZURE KUBERNETES COMBINED SENSOR	97
8.54	AZURE SHAREPOINT STATISTICS	97
8.55	AZURE ENTERPRISE SECRETS AND CERTIFICATES	98
8.56	AZURE SAML ENTERPRISE CERTIFICATES	98
8.57	AZURE ARC MACHINES	99
8.58	AZURE DEVOPS PIPELINES	99
8.59	AZURE SQL MANAGED INSTANCES	100
8.60	AZURE MANAGED DATABASES	100
8.61	AZURE APPLICATION GATEWAY BACKEND SERVERS	101
8.62	CONTAINER APPS	101
8.63	KEY VAULTS	101
8.64	AZURE ENTRADOMAIN HEALTH	103
8.65	AZURE WINDOWS GUEST METRICS	103
<b>9</b>	<b>TROUBLESHOOTING</b> .....	<b>105</b>
9.1	TROUBLESHOOTING THE AZURE SENSOR PACK INSTALLATION	105
9.2	TROUBLESHOOTING THE AZURE SENSOR CONFIGURATION	107
9.3	TROUBLESHOOTING AZURE DISCOVERY CONNECTION ERRORS	109
9.4	TROUBLESHOOTING AZURE DISCOVERY - PERMISSIONS	109

9.5	COLLECTING THE DISCOVERY FILES FOR AUTOMONX SUPPORT	109
9.6	TROUBLESHOOTING SENSORS LOCALLY	110
9.7	TROUBLESHOOTING THE DISCOVERY OF AZURE METRICS	111
9.8	TROUBLESHOOTING THE DISCOVERY OF AZURE SERVICE HEALTH	112
9.9	COLLECTING AZURE SERVICE DEBUG INFORMATION	113
9.10	COLLECTING AZURE SENSOR DEBUG INFORMATION	113
9.11	COLLECTING IN-DEPTH AZURE SENSOR DEBUG INFORMATION	114
<b>10</b>	<b>COMMAND LINE OPTIONS (CLI).....</b>	<b>115</b>
10.1	THE AZURE SENSOR PACK COMMAND LINE OPTIONS REFERENCE	115
10.2	FULLY AUTOMATED AZURE MONITORING	116
10.2.1	<i>Automated Discovery and Monitoring</i> .....	116
10.2.2	<i>Automated Clean-Up of Un-Needed Resources from Monitoring</i> .....	118
10.2.3	<i>Automatically Pausing Un-Needed Resources</i> .....	119
10.2.4	<i>Automated Inclusion and Exclusion of Sensors and Channels</i> .....	120
10.2.5	<i>Automated Scan-Now Functionality</i> .....	124
10.2.6	<i>Automated Addition and Removal of Tenants</i> .....	125
<b>11</b>	<b>REST API SERVICE .....</b>	<b>126</b>
11.1	FUNCTION GET_TENANT_ID	126
11.2	FUNCTION INITIATE DISCOVERY VIA API	127
11.3	FUNCTION ADD_TENANT	127
11.4	FUNCTION REMOVE_TENANT	128
11.5	FUNCTION TO CHECK AUTO DISCOVERY STATUS	128
11.6	FUNCTION ADD_TO_PRTG	129
11.7	FUNCTION DISCOVER_AND_ADD	129
11.8	FUNCTION UPDATE_KEY	130
11.9	ALLOWING REMOTE CONNECTION TO API SERVER	130
11.10	ENABLING HTTPS CONNECTION TO THE API SERVICE	131
<b>12</b>	<b>MONITORING THE AUTOMONX SENSOR PACK FOR AZURE PROCESSES.....</b>	<b>132</b>
<b>13</b>	<b>APPENDIX A - OBTAINING THE AZURE APPLICATION AND TENANT IDS.....</b>	<b>133</b>
13.1	RETRIEVING THE AZURE SECRET KEY	137
13.2	CREATING LIMITED MONITORING ROLE	138
<b>14</b>	<b>APPENDIX B - ADDING PERMISSIONS .....</b>	<b>143</b>
14.1	ENABLING APP SECRET KEYS FOR MONITORING	143
14.2	ENABLING RESERVATIONS FOR MONITORING	144
14.3	REGISTERING RESOURCE PROVIDERS	145
14.4	ENABLING DEVOPS FOR MONITORING	146
<b>15</b>	<b>APPENDIX C - ACTIVATING LOG ANALYTICS ON A VM .....</b>	<b>147</b>
<b>16</b>	<b>APPENDIX D – CREATING CUSTOM KQL LOG ANALYTICS SENSORS .....</b>	<b>148</b>
<b>17</b>	<b>APPENDIX E – MONITORING VM ACTIVE SESSIONS.....</b>	<b>155</b>

**18 APPENDIX E – TROUBLESHOOTING KEY VAULT ISSUES ..... 157**

## 1 Purpose

The purpose of this document is to provide a detailed explanation of the AutoMonX Sensor Pack for Microsoft Azure and how to deploy it.

## 2 AutoMonX Sensor Pack for Microsoft Azure Overview

The AutoMonX Sensor Pack for Azure is aimed at monitoring Microsoft Azure cloud environments for IT teams and service providers (MSPs and CSPs). The Azure Sensor pack can discover and monitor Azure resources located across multiple Azure tenants and multiple subscriptions. These unique sensors monitor the various aspects of Microsoft Azure's resources and services and have a tight integration with PRTG. The Azure Sensor pack currently supports auto-discovery and monitoring of 59 Azure resource types as seen below:

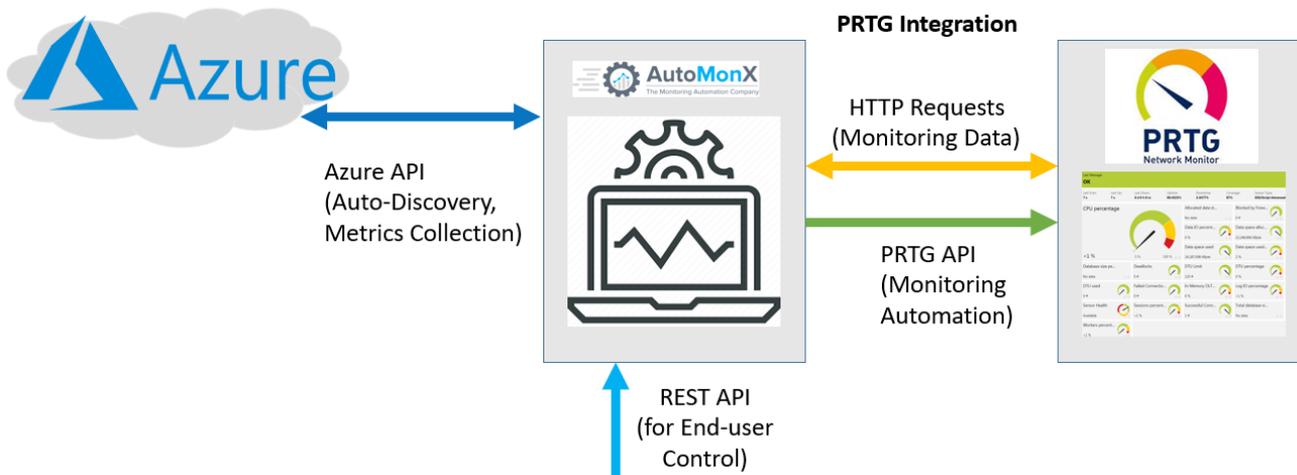
- SQL Databases (SAAS)
- Certificates
- App Service Plan (Server Farms)
- Web Sites
- Storage Accounts
- Cloud Services
- Database Accounts
- Service Bus Namespace
- Batch Accounts
- Redis
- Data Factory
- Scheduler
- Search Services
- Service Bus Queues
- Service Health
- Virtual Machines
- Azure Billing
- Application Insights
- Logic App
- Disks
- Notification Hubs
- Network Interfaces
- ExpressRoute Connections
- Virtual Network Gateways
- Backup Jobs
- App Secret Keys & Certificates
- Azure Kubernetes Cluster Metrics
- Azure Kubernetes Deployments
- VM Multi-Disk
- SQL Elastic Pools
- Storage File Share
- Service Issues
- Planned Maintenance
- Health Advisories
- Azure Advisor
- Quotas
- Host Pool
- Reservations
- Firewalls
- Application Gateway (WAF)
- (VPN) Connections
- IP Address
- Load Balancer
- Network Watcher
- Alerts
- AKS Upgrade Profiles
- Service Bus Topic Subscriptions
- Custom KQL queries
- SharePoint
- Enterprise App Secrets
- Enterprise SAML Certificates
- Azure Defender
- Arc Machines
- DevOps Pipelines
- SQL Managed Instances
- Application Gateway backend servers

- AKS Internal Deployments
- Container Apps
- Windows Guest Metrics **NEW**
- Key Vaults
- Entra Domain Health

The Azure Sensor pack supports auto-discovery and monitoring of many additional Azure resource types that their monitoring metrics are enabled via the generic API access.

### 3 How Does It Work?

The AutoMonX Sensor Pack for Azure connects via REST API to the Microsoft Azure management environment and collects metrics, values and additional information. It reports back to the Azure Sensor application server the gathered data. The Azure Sensor Pack is tightly integrated with PRTG and provides metrics and custom error limits in a form that is understandable by PRTG. In order to integrate with the Azure Sensor pack, our Monitoring Automation auto-configures PRTG by deploying HTTP Data Advanced sensors. These sensors connect to the Azure Sensor pack application server. As a result, PRTG displays the information gathered from Microsoft Azure by the Azure Sensor pack application server in a readable and clear way as seen in the picture below.



The EXE/Script Advanced sensor is the legacy option to integrate with PRTG and it also supported for backward compatibility.

✓ Sensor Azure App Metrics ★★★★★



### 3.1 The Azure Sensor Pack Architecture

The AutoMonX Sensor Pack for Azure application server sends multiple requests to the Microsoft Azure Management API and therefore needs a managing service to efficiently handle the requests while minimizing the PRTG probe load. The managing service harnesses the advantages of threading technology to efficiently queue the sensor requests to the Microsoft Azure Management API in order to provide reliable, swift, and lightweight performance. The Azure Sensor Pack is highly scalable and flexible thus can monitor thousands of resources spread across multiple Azure Tenants, subscriptions and regions.

### 3.2 The Azure Sensor Pack – Integration with PRTG

The AutoMonX Sensor Pack for Azure is tightly integrated with PRTG via two sensor types: the HTTP Advanced sensor and the custom EXE/XML sensor (for backward compatibility). Starting with version 4.2.x, HTTP Advanced sensor is the recommended (and the default) way to integrate PRTG and the Azure Sensor pack. Such integration is automatically created by the Monitoring Automation feature of the Azure Sensor Pack that pushes all the required configuration settings into PRTG.

### 3.3 The Azure Sensor Pack – Essential Terminology

The Azure Sensor Pack automatically integrates with the PRTG web interface for displaying Microsoft Azure resources. Below are some essential terms that are used through this deployment guide:

**Group** – PRTG group of devices. The Azure Sensor pack monitoring automation automatically organizes the Azure resources it discovers by creating groups of Azure resources (SQL, WebFarm, Network Interface etc) Read more about the automatically created hierarchy in [section 6.4](#).

**Device** – Each Azure resource is represented in PRTG as a device (for example a Virtual Machine, an SQL database, or a Network Interface).

**Sensor** – Created under every PRTG device. The main sensors that are available for most Azure resource types are Azure App Metrics and Azure Service Health. Additional sensor types are available, read more about it in [section 7.3](#).

**Channel** – The PRTG sensor channels (App Metrics) represent a single Azure resource performance metric.

**Azure Tags** – These are Labels in the Microsoft Azure Portal for logical grouping of resources. The AutoMonX Sensor Pack for Azure can utilize these tags to include or exclude the desired resources. Read more in [section 10.2.4](#).

## 4 Getting Started with Azure Sensor Pack

### 4.1 Supported Software versions

The Azure Sensor pack has been tested to support the following software:

Software Type	Versions	Comments
Windows OS	2012R2, 2016, 2019	Standard and Enterprise editions
Virtual Infrastructure	VMWare Cloud or on-prem, Azure VM, AWS VM	
PRTG Core and Probe deployments	20.x, 21.x, 22.x, 23.x, 24.x, 25.x	All On-Prem PRTG license types supported
PRTG Hosted (Cloud)	Supported with Azure Sensor pack 3.31 and higher	

### 4.2 Azure Sensor Pack - Port requirements

The AutoMonX Sensor Pack for Azure requires the following ports to be open for it to function correctly. Please make sure that the local firewall / anti-virus and the external firewalls are configured as required to allow the Sensor Pack to function correctly.

Port / URL	Purpose	Direction
<a href="https://management.azure.com">https://management.azure.com</a> <a href="https://login.windows.net">https://login.windows.net</a> <a href="https://login.microsoftonline.com">https://login.microsoftonline.com</a> <a href="https://management.core.windows.net">https://management.core.windows.net</a> <a href="https://graph.microsoft.com">https://graph.microsoft.com</a> <a href="https://api.applicationinsights.io">https://api.applicationinsights.io</a> <a href="https://api.loganalytics.io">https://api.loganalytics.io</a> <a href="https://app.vssps.visualstudio.com">https://app.vssps.visualstudio.com</a> <a href="https://dev.azure.com">https://dev.azure.com</a>	Azure API connection	From PRTG Probe to Azure
TCP 443, 80	Connect to Azure, PRTG API	From PRTG Probe to Azure and PRTG Core
SNMP (UDP 161), ICMP	For monitoring Azure-based Virtual Machines	From PRTG Probe to Azure VMs
TCP 443 (Optional, for Evaluation requests)	Connect to our license server auto.automonx.net	From PRTG Probe to our license server in Azure

TCP 8148 TCP 8092 TCP 8075	Internal service ports. Make sure these ports are not occupied by other programs on the server.	No need to open FW rules.
----------------------------------	--	---------------------------

### 4.3 Azure Sensor Pack – Anti-Virus Requirements

The AutoMonX Sensor Pack for Azure initiates many processes and threads during its normal execution. Configure your anti-virus and/or anti-malware software to exclude the AutoMonX directory in `<drive>:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML` from on-access scanning. This would greatly improve the general performance of the Azure Sensor Pack.

### 4.4 Downloading the Azure Sensor pack

Obtain the software by downloading it from the AutoMonX web site at <http://www.automonx.com/downloads>

#### Production Releases

Product Name	Version	Updated On	Download	Comments	MD5 Fingerprint
3PAR/Primera Sensor Pack	2.7.4.6.1	28.12.2021	<a href="#">Download</a>		a3a9aaa4ffff6034c9e95bf8d4e4da50
Azure Sensor Pack	4.2.8 (Previous)	23.10.2022	<a href="#">Download</a>	New License for versions prior to v4.0.26	21e8f673711df30460f5307967a7c7ac
Azure Sensor Pack	4.3.10 (Stable)	28.08.2023	<a href="#">Download</a>	New License for versions prior to v4.0.26	5e61415d244fab6aabc211fe8515cc6d
Cisco ACI Sensor Pack	2.2.10	20.08.2024	<a href="#">Download</a>		2b89ebbd7ce6b1bb410353b41cf13bc7

The Azure Sensor Pack new version will first appear in the Preview section of the download page

#### Preview Releases

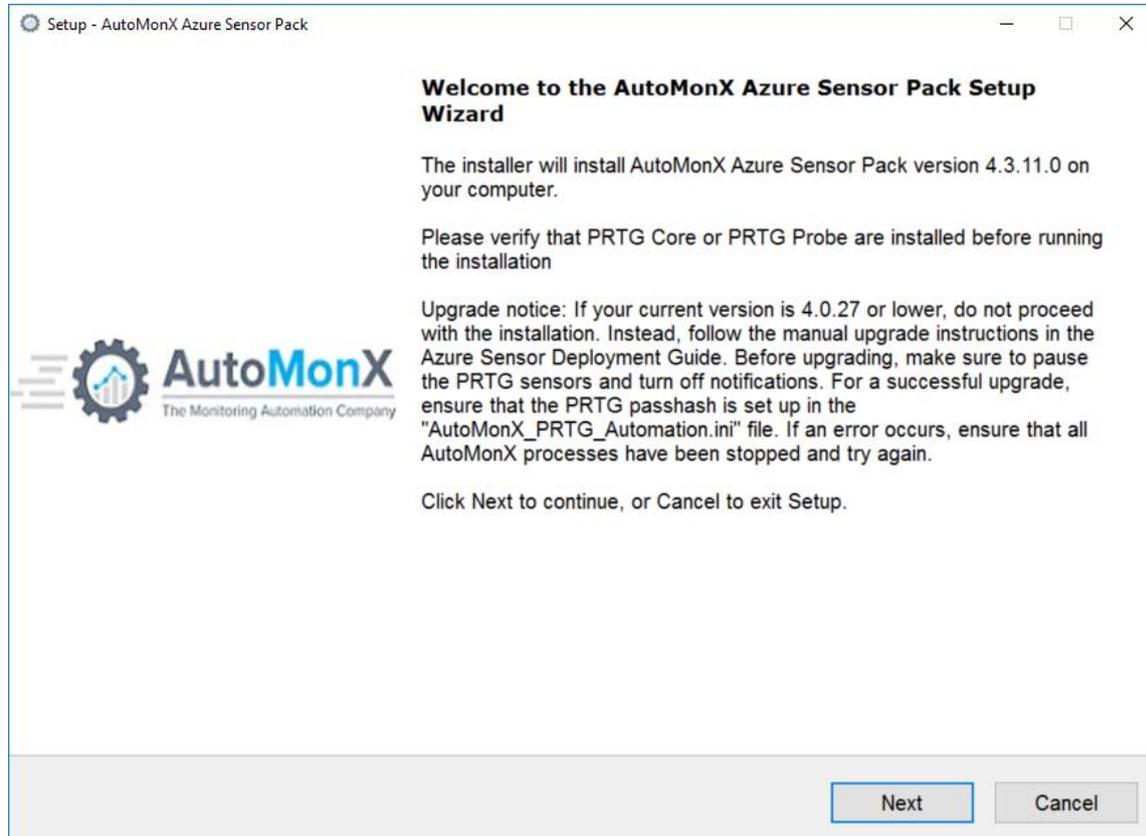
Product Name	Version	Updated On	Download	Comments	MD5 Fingerprint
3Par Sensor Pack	v2.8.1.1	08.10.2024	<a href="#">Download</a>	Preview Release!	807626d50c8693307f7a4e31deb6aa08
Azure Sensor Pack	v4.3.28.1	11.11.2024	<a href="#">Download</a>	Preview Release!	15e52c582228f814fef71c7a6706d826

## 4.5 Installing the Azure Sensor Pack

Download the latest Azure Sensor Installer from

<https://www.automonx.com/downloads>

Start the installer and follow the instructions.



Configure the PRTG connection information. Make sure to mark "Enable HTTPS" if relevant.

Setup - AutoMonX Azure Sensor Pack

### PRTG Web Credentials

This information is critical for the immediate success of this installation

User Name:

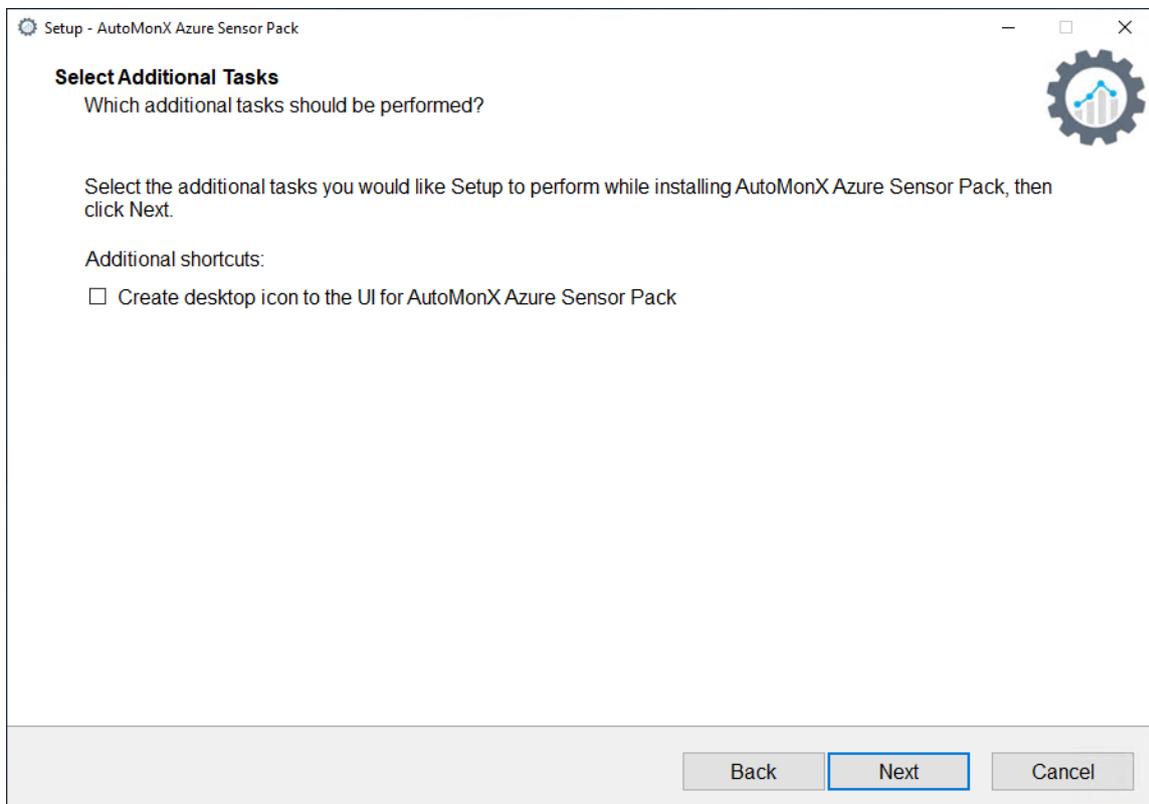
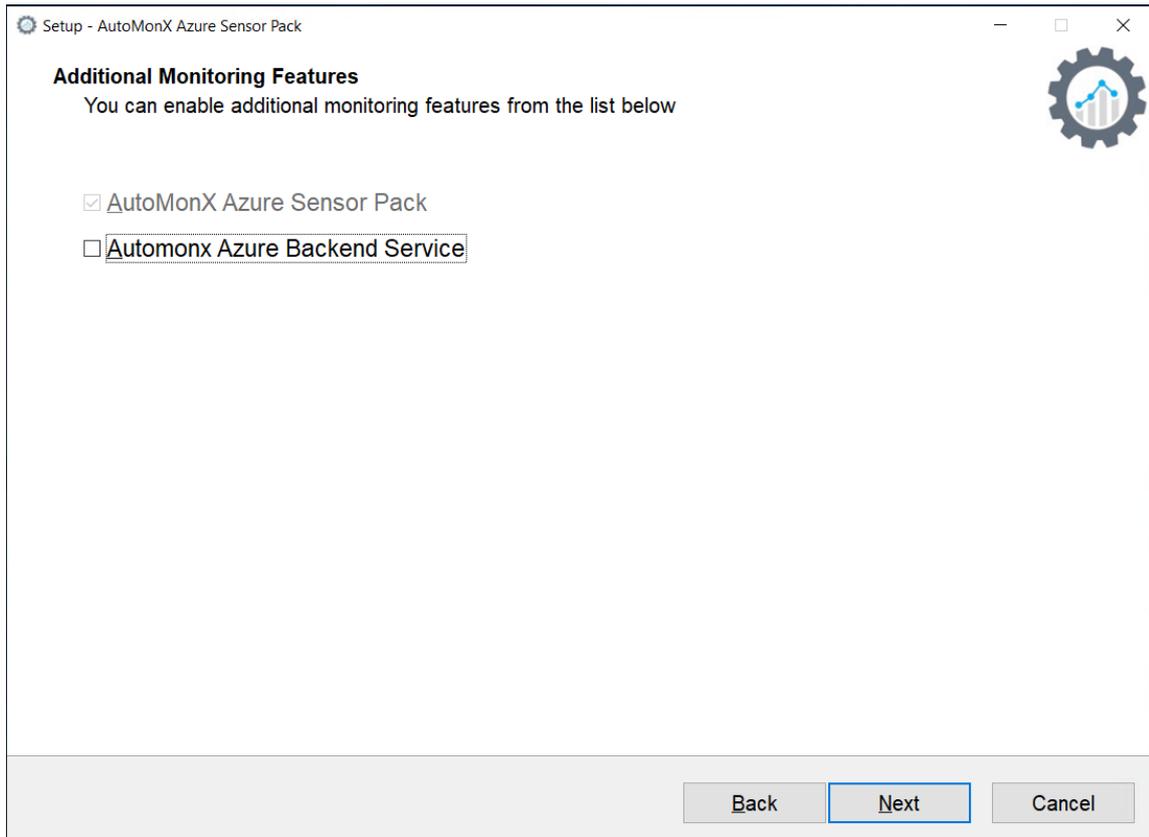
Password:

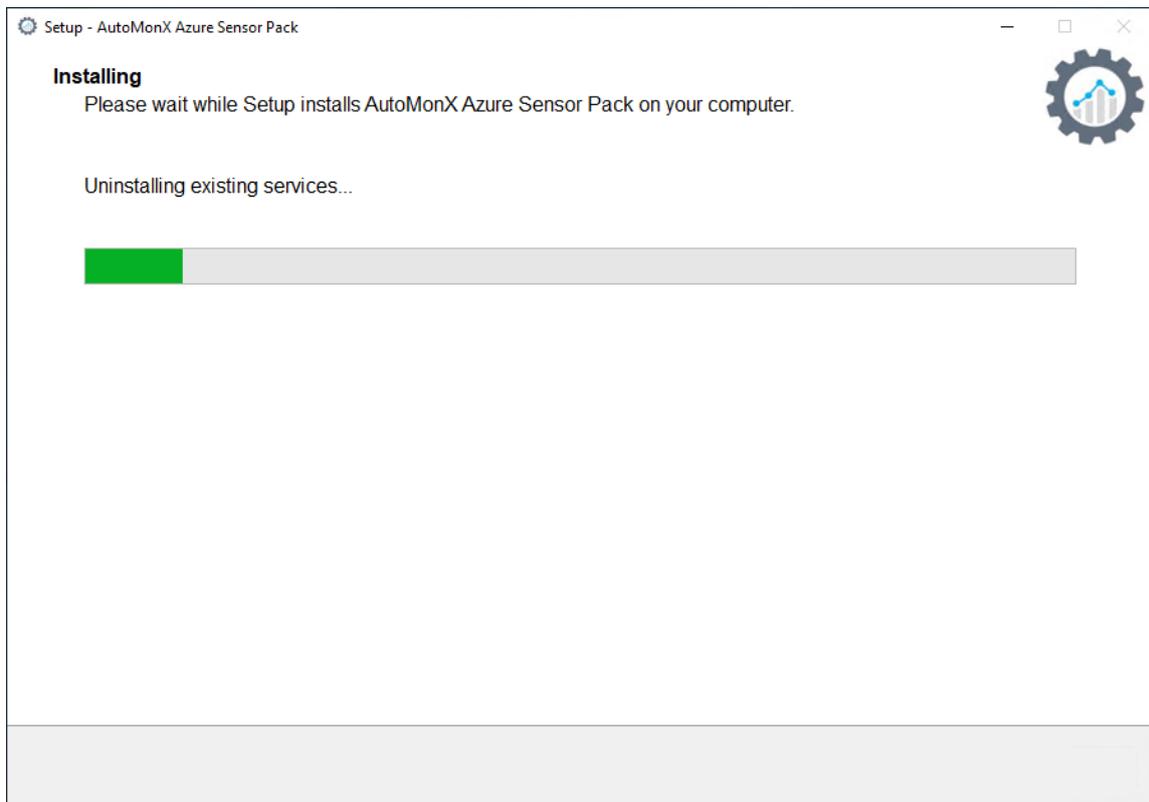
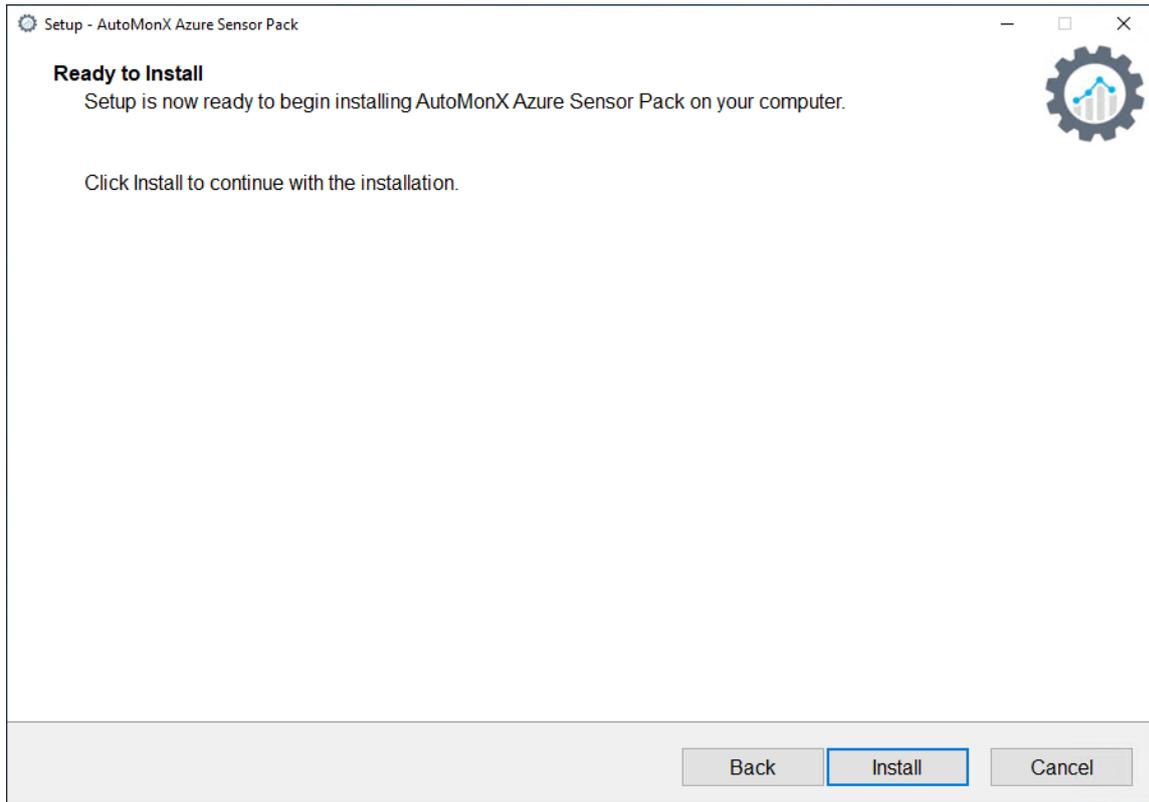
IP:

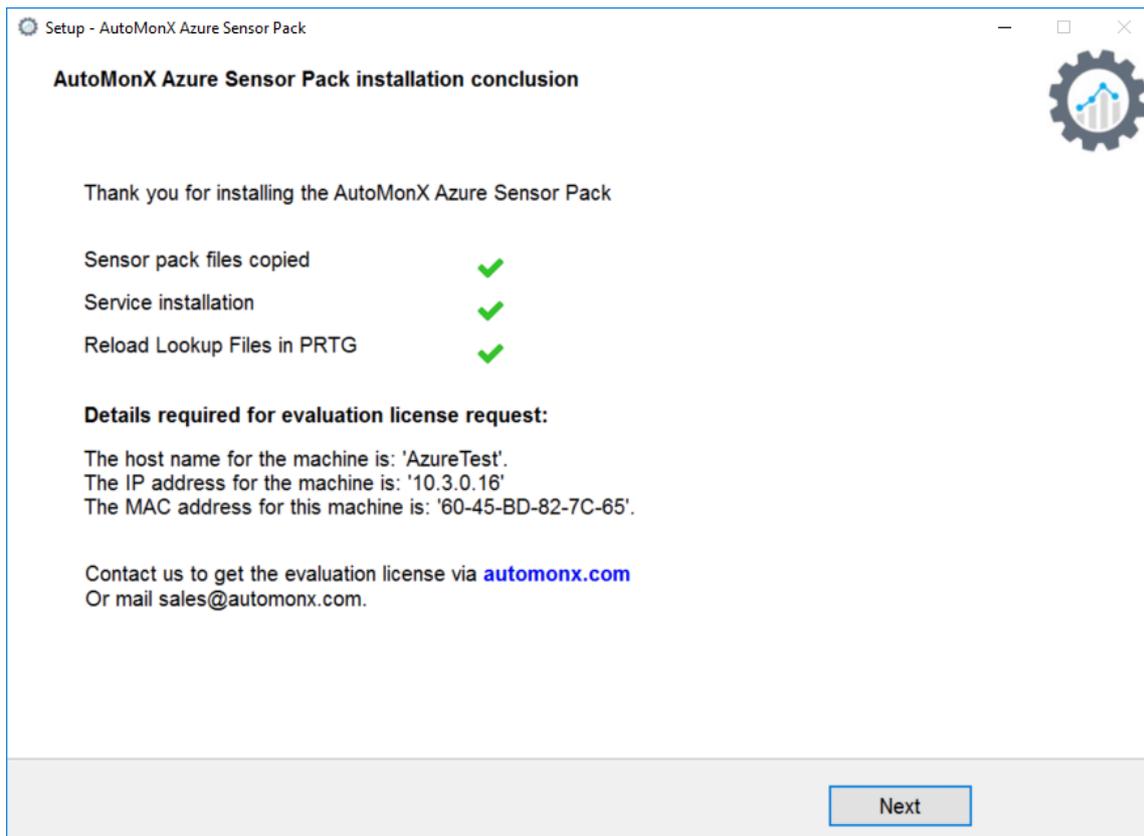
Port:

Enable HTTPs

Choose if you wish to install the additional Backend Service (Monitors the Sensor Pack and provides API capability).







If an error occurred while updating the Lookup files, update them manually [as explained in section 4.6.](#)

## 4.6 The Directory Structure of the Azure Sensor Pack

The installer will extract all the Azure Sensor pack files and sub directories to the following directory on the PRTG Probe server:

"<drive>:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML"

The Azure Sensor pack will not function anywhere else. The extracted files will create a directory structure as seen below. The root directory under EXEXML will look like the picture below:

Name	Date modified	Type	Size
AutoMonX	23/07/22 1:51 PM	File folder	
Automonx_AzureSensor.cmd	28/05/19 6:54 PM	Windows Comma...	1 KB
Automonx_AzureSensorEngine.cmd	15/08/21 5:09 PM	Windows Comma...	1 KB

AutoMonX directory content:

**AutoMonX Ltd © 2025 All Rights Reserved**

Web : <http://www.automonx.com>

Email : [support@automonx.com](mailto:support@automonx.com)

- Contrib
- Data
- Logs
- Queue
- Reports
- ResponderREQ
- Scheduler

**AutoMonX** Root directory would include the following files

Filename	Purpose
AutoMonX_AzureSensorEngine.cmd AutoMonX_AzureSensor.cmd	Used for invoking the Sensor by PRTG

**Common** directory would include the following files:

Filename	Purpose
ExecutableActivation.dll ExecutableActivation.pdb FileHelpers.dll FileHelpers.xml Newtonsoft.Json.dll Newtonsoft.Json.xml Renci.SshNet.dll Renci.SshNet.xml SensorAutoDisco_UI.exe SensorAutoDisco_UI.exe.config SensorAutoDisco_UI.ini SensorAutoDisco_UI.Lib.dll SensorAutoDisco_UI.Lib.pdb SensorAutoDisco_UI.pdb	Discovery and monitoring User Interface files
LicDetailsLocator.exe	Utility to gather the required details for license generation
AutoMonX_PRTG_Automation.exe AutoMonX_PRTG_Automation.ini AMX_PRTG_sensors_issues.exe	Monitoring Automation module files
exclude_mon - Example.csv exclude_mon.csv include_mon.csv down_sensors_filter.ini pause_sensors_filter.ini	Filtering logic files for Monitoring Automation
AutoMonX_ReqFetch.dll libcrypto-1_1-x64.dll libgcc_s_seh-1.dll libssh2-1.dll libssl-1_1-x64.dll	DLLs required for Monitoring Automation

libstdc++-6.dll libwinpthread-1.dll zlib1.dll	
---	--

**Azure** directory would include the following files:

Filename	Purpose
Contrib Data Logs Queue Reports ResponderREQ Scheduler	Sub-directories required for the Azure sensor operation
AutoMonX_AzureCollector.exe AutoMonX_AzureResponder.exe AutomonX_AzureSensor.exe AutoMonX_AzureSensorRun.exe automonx_azuresensorsumm_starter.exe Azure_Backend_Service.exe	Azure Sensor Pack executables
AutoMonX_AzureSensor.ini AzureConnProfiles.ini	Azure Sensor Pack main configuration file and Azure Connection profiles configuration file (for multiple Tenants)
AutoMonX_AzureLicense.dat AutoMonX_AzureTenantLicense.dat	Azure sensor license file – Sensors Azure Tenants license file
libcrypto-1_1-x64.dll libgcc_s_seh-1.dll libssl-1_1-x64.dll libstdc++-6.dll libwinpthread-1.dll zlib1.dll	Azure sensor DLL files

**OVL** directory content:

Filename	Purpose
automonx.azure.availability.ValueLookup.ovl automonx.azure.availabilitymetric.ValueLookup.ovl automonx.azure.certificate.ValueLookup.ovl automonx.azure.connectionstatus.ValueLookup.ovl automonx.azure.defender.ValueLookup.ovl automonx.azure.dfv2pipeline.ValueLookup.ovl automonx.azure.jobbackupstatus.ValueLookup.ovl automonx.azure.mdbstatus.ValueLookup.ovl automonx.azure.quota.ValueLookup.ovl automonx.azure.status.ValueLookup.ovl automonx.azure.taskstatus.ValueLookup.ovl	PRTG custom lookup file for the Azure Sensor Pack

**Contrib** directory content:

Filename	Purpose
Amx_Discovery_Addition.cmd Amx_Discovery_Addition.ps1 Amx_Sensor_Deletion.cmd	Example scripts for running automated discovery and addition to PRTG. Please don't edit these files – instead copy to a different folder, as they will be overwritten upon upgrade.

**Reports** directory content – here you will find the Auto Discovery reports in HTML format.

**Scheduler** directory content:

Filename	Purpose
Scheduler.csv	Sample settings for the scheduler functionality that allows running API commands for the Azure Sensor Pack automation such as discovery, monitoring automation and more

## 4.7 Lookup File Handling (on-prem PRTG)

You need the AutoMonX Azure Sensor Lookup files to properly display the sensor output in PRTG. Copy the following files from the OVL folder, located in the zip file to the following folder on the **PRTG Core server**.

"<drive>:\Program Files (x86)\PRTG Network Monitor\lookups\custom"

Below are the OVL files that need to be in the folder:

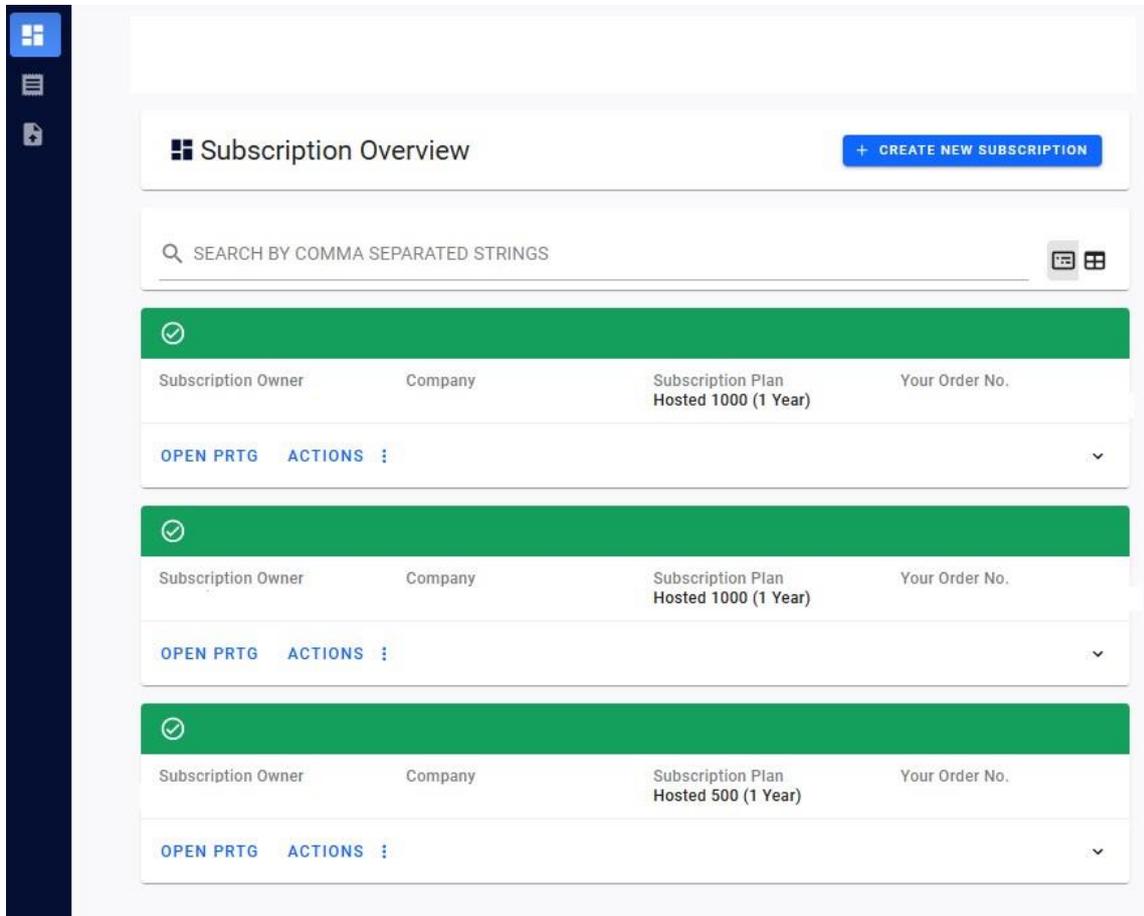
- automonx.azure.availability.ValueLookup.ovl
- automonx.azure.availabilitymetric.ValueLookup.ovl
- automonx.azure.certificate.ValueLookup.ovl
- automonx.azure.connectionstatus.ValueLookup.ovl
- automonx.azure.defender.ValueLookup.ovl
- automonx.azure.dfv2pipeline.ValueLookup.ovl
- automonx.azure.jobbackupstatus.ValueLookup.ovl
- automonx.azure.mdbstatus.ValueLookup.ovl
- automonx.azure.quota.ValueLookup.ovl
- automonx.azure.status.ValueLookup.ovl
- automonx.azure.taskstatus.ValueLookup.ovl

After copying the Lookup files to the PRTG Core Server, you would need to reload the PRTG Lookup database with the following action:

From the PRTG upper menu -> Setup -> System Administration -> Administrative Tools -> Reload Lookup Files -> Go!

## 4.8 Lookup File Handling (PRTG Cloud)

You can now manually add the custom Automonx lookup files into hosted PRTG. After installation on the probe, you can find them in the folder: PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\OVL

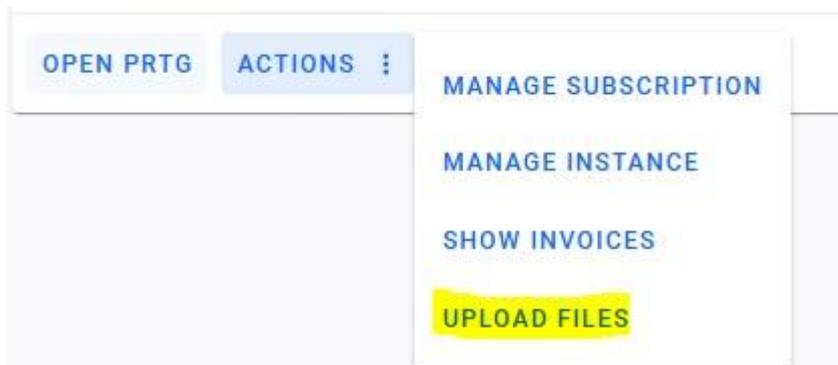


**Subscription Overview** + CREATE NEW SUBSCRIPTION

SEARCH BY COMMA SEPARATED STRINGS

Subscription Owner	Company	Subscription Plan	Your Order No.
		Hosted 1000 (1 Year)	
		Hosted 1000 (1 Year)	
		Hosted 500 (1 Year)	

Each row includes 'OPEN PRTG' and 'ACTIONS' buttons.



**OPEN PRTG** **ACTIONS** ⋮

- MANAGE SUBSCRIPTION
- MANAGE INSTANCE
- SHOW INVOICES
- UPLOAD FILES**

Upload Custom Files

**My Custom Files**

Upload your custom files. Upload your own [created custom device templates](#) here if the predefined custom templates you need are not available in the list above.

**i** You cannot upload files that are larger than 1MB.

- MIB/ .MIB, .mib, .my +
- devicetemplates/ .odt +
- lookups/custom/ .ovl +
- snmplibs/ .oidlib +
- webroot/icons/devices/ .svg, .png +

UPLOAD

### Custom Files

1. Click  next to the ovl file type and browse for the path to the file in the **File Explorer**.
2. Select the file and click **Open**.

**My Custom Files**

Upload your custom files. Upload your own [created custom device templates](#) here if the predefined custom templates you need are not available in the list above.

**i** You cannot upload files that are larger than 1MB.

- MIB/ .MIB, .mib, .my +
- devicetemplates/ .odt +
- lookups/custom/ .ovl +
- snmplibs/ .oidlib +
  -  custom\_file.oidlib
- webroot/icons/devices/ .svg, .png +

UPLOAD

### Custom File

3. Click **Upload** to upload the custom ovl files.

Successfully uploaded to your PRTG. **CLOSE**

### Custom File Upload Successful

Previous way to handle hosted PRTG lookups:

Since custom lookup files are not available in the PRTG Cloud edition, the Azure Sensor pack can also function with the built-in lookup files. To enable the PRTG Cloud mode, you need to edit the `AutoMonX_AzureSensor.ini` file.

Open the `AutoMonX_AzureSensor.ini` file in Notepad and modify the following line from false to true (support PRTG Cloud)

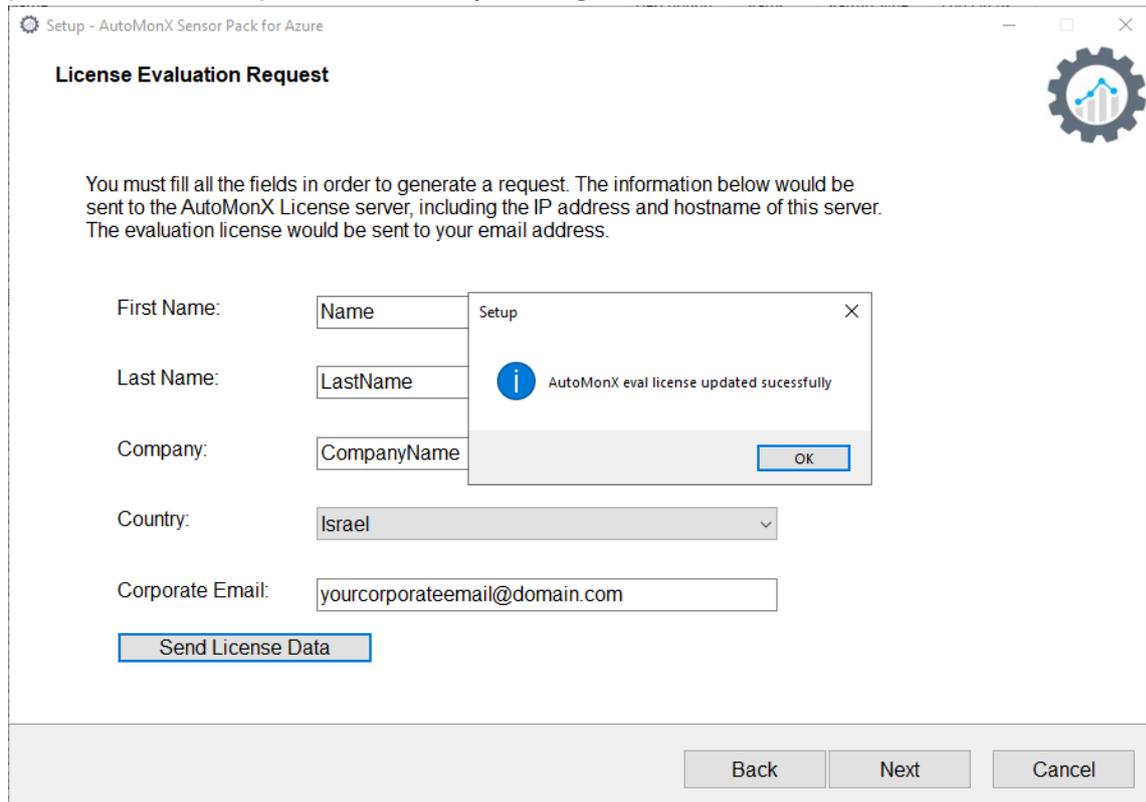
```
CLOUD_VLOOKUP=true
```

Save the file and restart the AutoMonX Azure Sensor Service.

## 4.9 Requesting a License for the AutoMonX Sensor Pack for Azure

The initial license file used by the Azure Sensor Pack, part of the downloaded installer, is empty and functions as a place holder. You must activate the sensor pack by obtaining a license.

You can request an evaluation license which would automatically activate the product for a trial period of 30 days using this screen:



**License Evaluation Request**

You must fill all the fields in order to generate a request. The information below would be sent to the AutoMonX License server, including the IP address and hostname of this server. The evaluation license would be sent to your email address.

First Name:

Last Name:

Company:

Country:

Corporate Email:

(on sub-dialog)

Filling all fields is mandatory.

Fill all fields as shown and click on "Send License Data" after a few seconds the Automonx\_AzureLicense.dat file would be updated automatically with the evaluation license, and you can proceed further with the installation while skipping (clicking next) on the manual license input screen.

You can also request the license string manually via email:

To successfully activate manually the Azure Sensor Pack:

1. Contact AutoMonX either by filling in the license request form at <http://www.automonx.com/azure>
2. Or send an email to [sales@automonx.com](mailto:sales@automonx.com) and provide the following information:
  - Your first and last name
  - Your contact details (email, phone)
  - Your business addresses.
  - The hostname of the PRTG Probe server machine
  - The IP address of the PRTG Probe server

**Important:** The hostname is case sensitive. Please use the LicDetailsLocator.exe utility to obtain the hostname and IP address

AutoMonX would provide you with a fully functional software evaluation license (two strings) valid for 30 days.

At the end of the evaluation period, you would need to purchase a license to continue monitoring your Azure infrastructure.

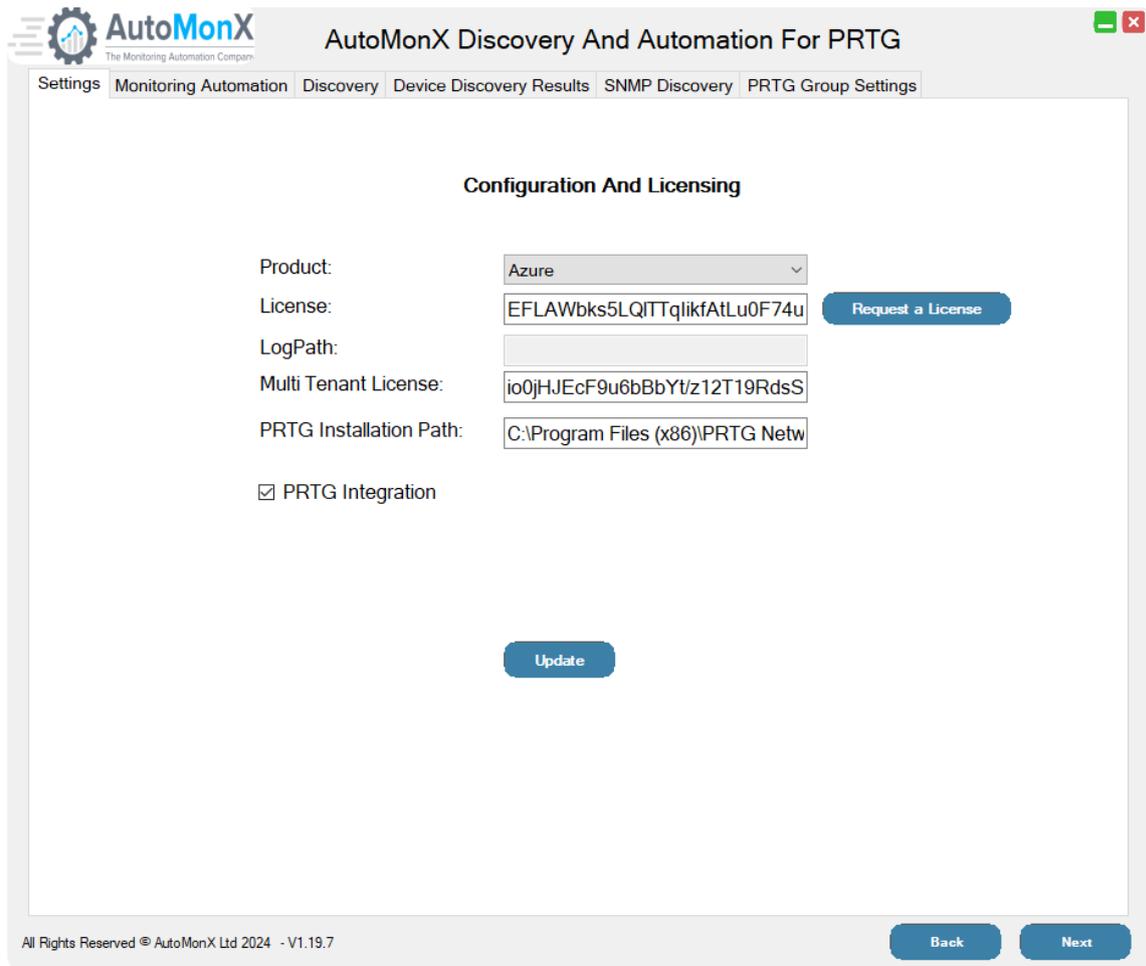
#### 4.10 Activating the Azure Sensor pack License

In version 3.23 and higher, you can activate the licenses of the Azure Sensor Pack by [opening our UI](#) and selecting the Settings Tab. Select “Azure” from the Product drop-down list (if not selected) and paste the license string you have received via email.

You can also activate the Azure Sensor Pack by editing the following files via Notepad, pasting the relevant license string(s) you have received via email and saving the files.

AutoMonX\_AzureLicense.dat – For Azure Sensor pack resources monitoring license for single Tenant scenarios.

AutoMonX\_AzureTenantLicense.dat – For Azure Sensor pack multi-tenant license for multiple Azure tenants monitoring scenarios.



The screenshot shows a web interface for configuring AutoMonX for PRTG. The page title is "AutoMonX Discovery And Automation For PRTG". The navigation menu includes "Settings", "Monitoring Automation", "Discovery", "Device Discovery Results", "SNMP Discovery", and "PRTG Group Settings". The main content area is titled "Configuration And Licensing" and contains the following fields and options:

- Product: Azure (dropdown menu)
- License: EFLAWbks5LQITtqlikfAtLu0F74u (text input) with a "Request a License" button
- LogPath: (empty text input)
- Multi Tenant License: io0jHJEcF9u6bBbYtz12T19RdsS (text input)
- PRTG Installation Path: C:\Program Files (x86)\PRTG Netw (text input)
- PRTG Integration

At the bottom of the form is an "Update" button. The footer of the page contains "All Rights Reserved © AutoMonX Ltd 2024 - V1.19.7" and "Back" and "Next" buttons.

## 5 Azure Sensor Pack Configuration

### 5.1 Preparing for Configuring the AutoMonX Sensor Pack for Azure

The AutoMonX PRTG Azure Sensor pack connects to Microsoft Azure via a service principal, that at least must have read permissions. You need to obtain the following information for the AutoMonX Sensor Pack for Azure to properly function:

- APP Identification key (Application Id) - AZURE\_AD\_ID
- AD password KEY (SECRET KEY) - AZURE\_PASS
- Tenant Identification key - AZURE\_TENANT

The connection information of the first Azure tenant added to monitoring is always stored in AutoMonX\_AzureSensor.ini.

Starting with Azure Sensor Pack version 4.0, when monitoring a multi-tenant Azure environment, the location of the connection parameters for any additional Azure tenant(s) has moved to a new file called AzureConnProfiles.ini.

The Azure connection settings are modified by using the AutoMonX UI. Check the [Configuring the Azure Sensor Pack](#) section for more information.

**Note:** The Azure Sensor Pack is backward compatible and will be able to read the Azure connection details from their previous location.

The Tenant ID is the Directory ID which can be found in the Properties sub-menu in the Azure Active Directory resource.

Use the following guide to create a service principal to get an Application Id and password:

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal>

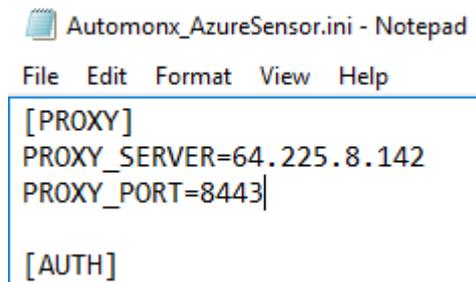
It is recommended to follow the entire guide in order to prepare all the relevant settings for the AutoMonX Sensor Pack for Azure.

## 5.2 Proxy Server Connection Configuration

The following is required to configure the proxy data. In the file “AutoMonX\_AzureSensor.ini” add the following lines:

```
[PROXY]
PROXY_SERVER=
PROXY_PORT=
```

For example:

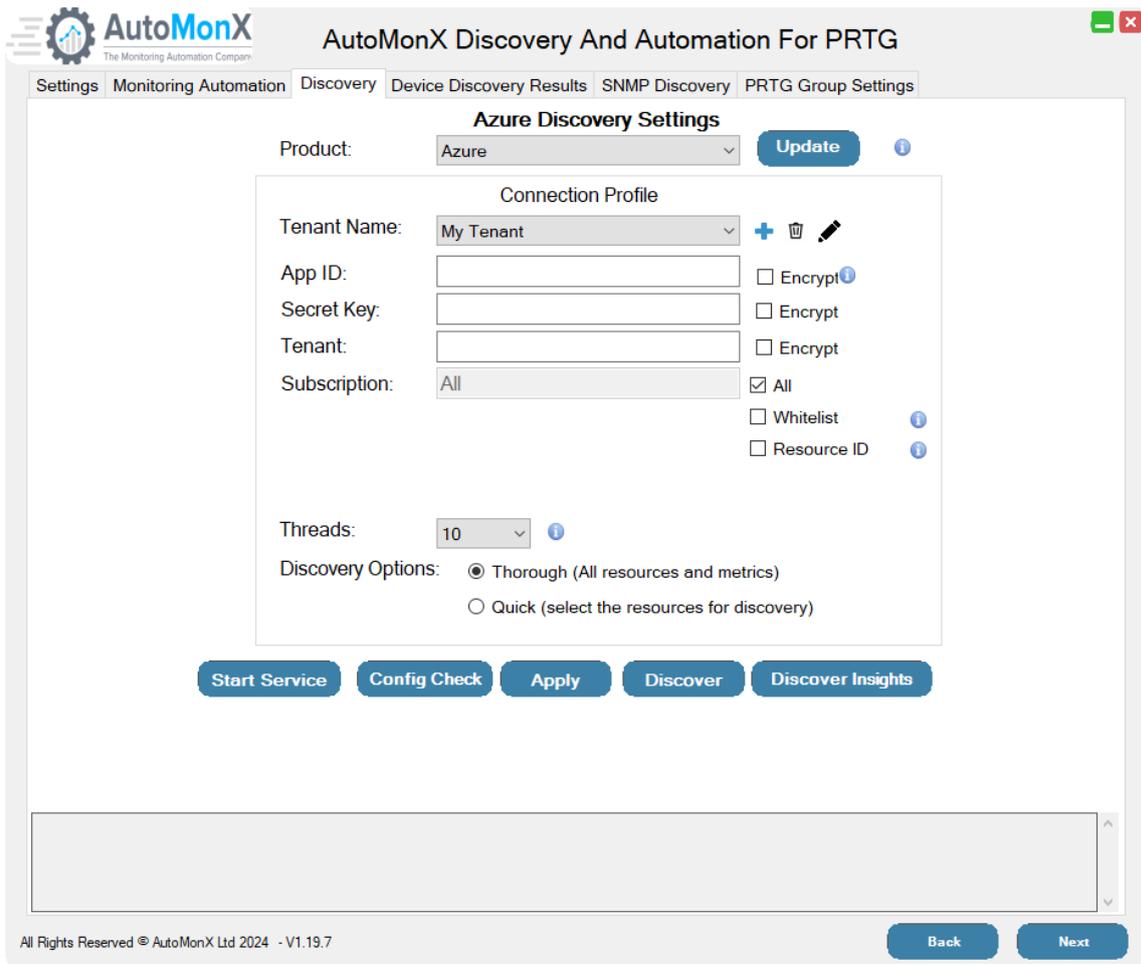


```
Automonx_AzureSensor.ini - Notepad
File Edit Format View Help
[PROXY]
PROXY_SERVER=64.225.8.142
PROXY_PORT=8443
[AUTH]
```

## 5.3 Configuring the AutoMonX Sensor Pack for Azure

You need to start the Azure Sensor pack configuration UI by running as Administrator a file called SensorAutoDisco\_UI.exe from the AutoMonx\Common folder.

Use the configuration UI to fill in the required details for the Azure Sensor Pack to connect to Azure API as the initial ini file contains dummy data.



The screenshot shows the 'Azure Discovery Settings' configuration page within the AutoMonX interface. The page is titled 'AutoMonX Discovery And Automation For PRTG' and has a navigation bar with tabs for 'Settings', 'Monitoring Automation', 'Discovery', 'Device Discovery Results', 'SNMP Discovery', and 'PRTG Group Settings'. The 'Discovery' tab is active.

The main configuration area is titled 'Azure Discovery Settings' and includes the following fields and options:

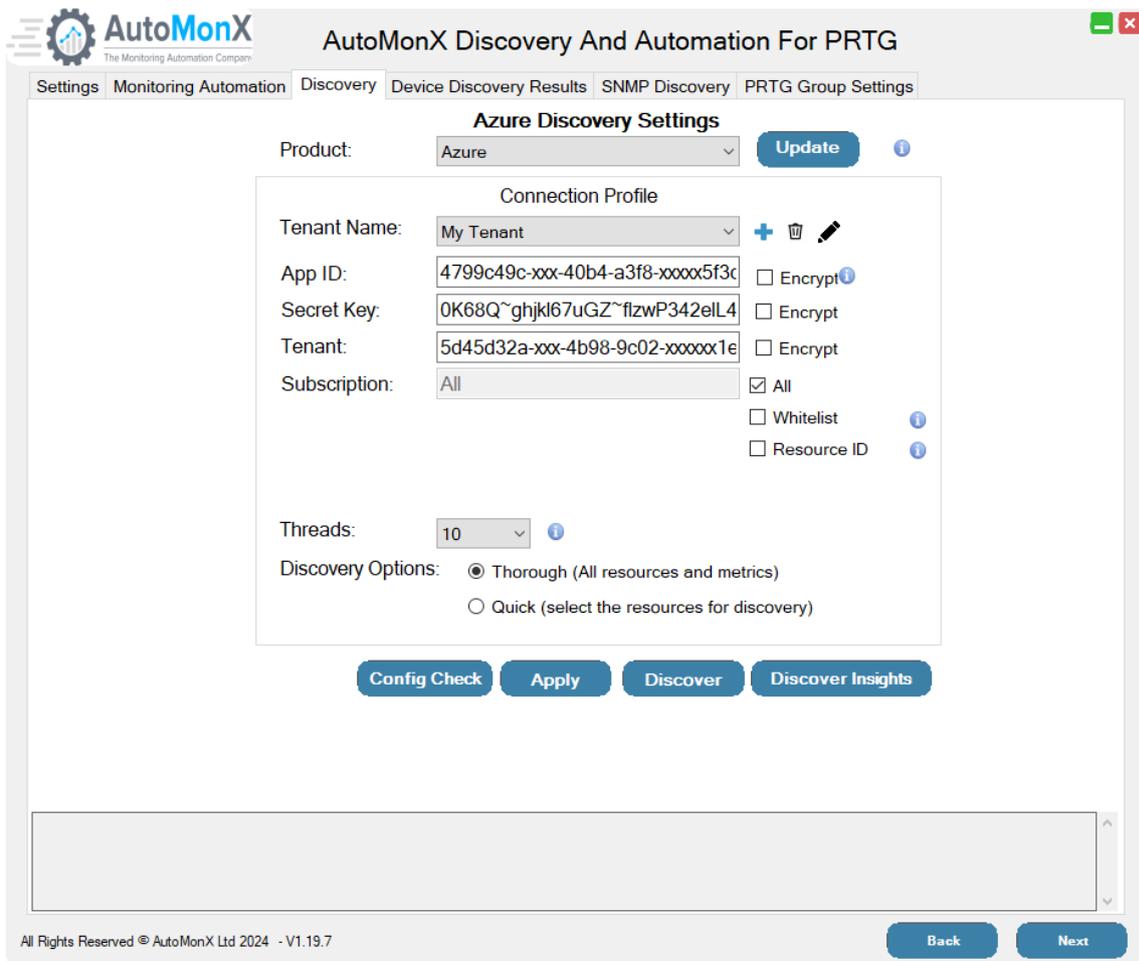
- Product:** A dropdown menu set to 'Azure' with an 'Update' button and an information icon.
- Connection Profile:** A section containing:
  - Tenant Name:** A dropdown menu set to 'My Tenant' with add, delete, and edit icons.
  - App ID:** A text input field with an 'Encrypt' checkbox and an information icon.
  - Secret Key:** A text input field with an 'Encrypt' checkbox.
  - Tenant:** A text input field with an 'Encrypt' checkbox.
  - Subscription:** A dropdown menu set to 'All' with a checked 'All' checkbox, and 'Whitelist' and 'Resource ID' checkboxes, each with an information icon.
- Threads:** A dropdown menu set to '10' with an information icon.
- Discovery Options:** Radio buttons for 'Thorough (All resources and metrics)' (selected) and 'Quick (select the resources for discovery)'.

At the bottom of the configuration area, there are five buttons: 'Start Service', 'Config Check', 'Apply', 'Discover', and 'Discover Insights'. Below the configuration area is a large empty text box. At the very bottom of the window, there is a footer with 'All Rights Reserved © AutoMonX Ltd 2024 - V1.19.7' and 'Back' and 'Next' buttons.

**Important:** Press the Apply button to save your changes and press the “Install Service” button to install the Azure Sensor Pack service.

**Azure Billing Information:** Starting with version 4.2.8 of the Azure Sensor pack, it is no longer required to add the Azure billing information. The Azure billing information will be automatically discovered.

Below is an example of how the UI would look like if the service is already installed and started.



In some cases, where the Azure Sensor Pack service is not started, the UI would make visible the “Start Service” button so you can start the Sensor Pack service from our UI.


**AutoMonX**  
The Monitoring Automation Company
AutoMonX Discovery And Automation For PRTG
✕

Settings
Monitoring Automation
Discovery
Device Discovery Results
SNMP Discovery
PRTG Group Settings

### Azure Discovery Settings

Update
i

Product: Azure

Connection Profile

Tenant Name: My Tenant + 🗑️ ✎️

App ID: 4799c49c-xxx-40b4-a3f8-xxxx5f3c  Encrypt i

Secret Key: 0K68Q~ghjkl67uGZ~flzwP342eLL4  Encrypt

Tenant: 5d45d32a-xxx-4b98-9c02-xxxxx1e  Encrypt

Subscription: All  All

Whitelist i

Resource ID i

Threads: 10 i

Discovery Options:  Thorough (All resources and metrics)

Quick (select the resources for discovery)

Start Service
Config Check
Apply
Discover
Discover Insights

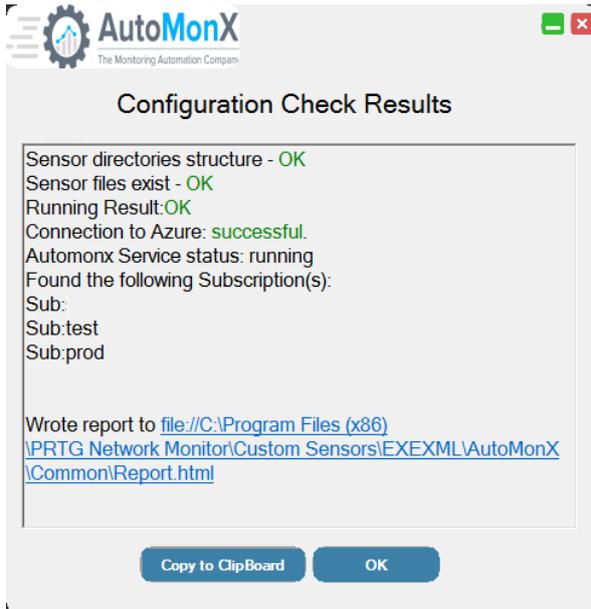
All Rights Reserved © AutoMonX Ltd 2024 - V1.19.7

Back
Next

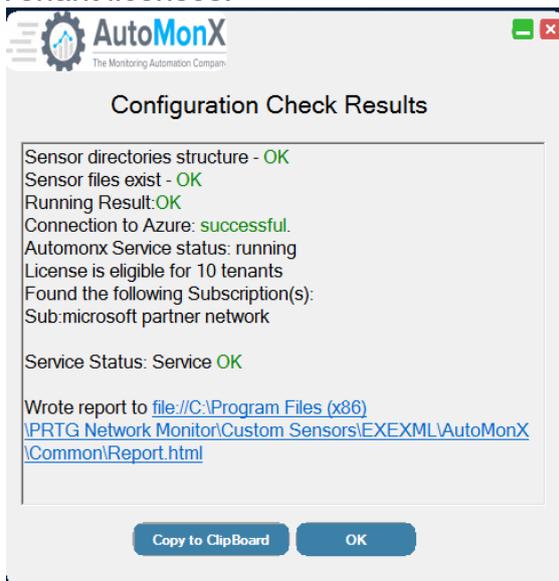
## 5.4 Azure Sensor pack - Configuration Check

To verify your Azure Sensor Pack installation, start the UI, go to AutoMonX\Azure\Common directory and run as Administrator the SensorAutoDisco\_UI.exe file.

Fill in all the required information. Press the Config Check button to initiate a self-check to make sure everything was configured correctly. Successful test will look like the screen below:



When a Multi-Tenant license is available, Config check will report the number of Tenant licenses:



## 5.5 Azure Sensor pack – Upgrade Instructions

If you are required to upgrade an existing installation of the AutoMonX Sensor Pack for Azure, please follow the steps below.

### Upgrade Notes:

The license of any version prior to **v4.0.28**, needs to be migrated. Please contact [sales@automonx.com](mailto:sales@automonx.com) before upgrading to the latest releases.

When upgrading from version **4.0.17** or lower, update the OVL files in the relevant directory and reload them via the PRTG Web Administration page. Delete all the AutoMonX\_Azure\_types.dat files and run full discovery, otherwise the metrics polling will not function properly. This is automated with the installer in later versions.

Starting from version **4.2.0**, auto-discovery by default uses the HTTP Data Advanced sensors as the integration point with PRTG. This is by design as it replaces the previous EXE/Script Advanced sensor. You can fallback to EXE type sensors by changing the HTTP\_SENSOR\_MODE value in the AutoMonX\_AzureSensor.ini file.

Make sure to follow the upgrade procedure carefully and avoid copying INI and the license (.dat) files from the zip file unless instructed. It is suggested to pause the Azure Sensor Pack sensors until the upgrade is completed.

Using the Azure Sensor Pack Installer is highly recommended. The installer automatically upgrades all the Sensor pack files. Automatic upgrade to the latest version is supported starting from version 4.0.17 of the Azure Sensor pack.

- Download the latest Azure Sensor Installer from <https://www.automonx.com/downloads>
- Make sure to pause the Azure root group in PRTG.
- Add the PRTG passhash to the configuration file (To smoothly update the lookup files. You may delete this later). For example:

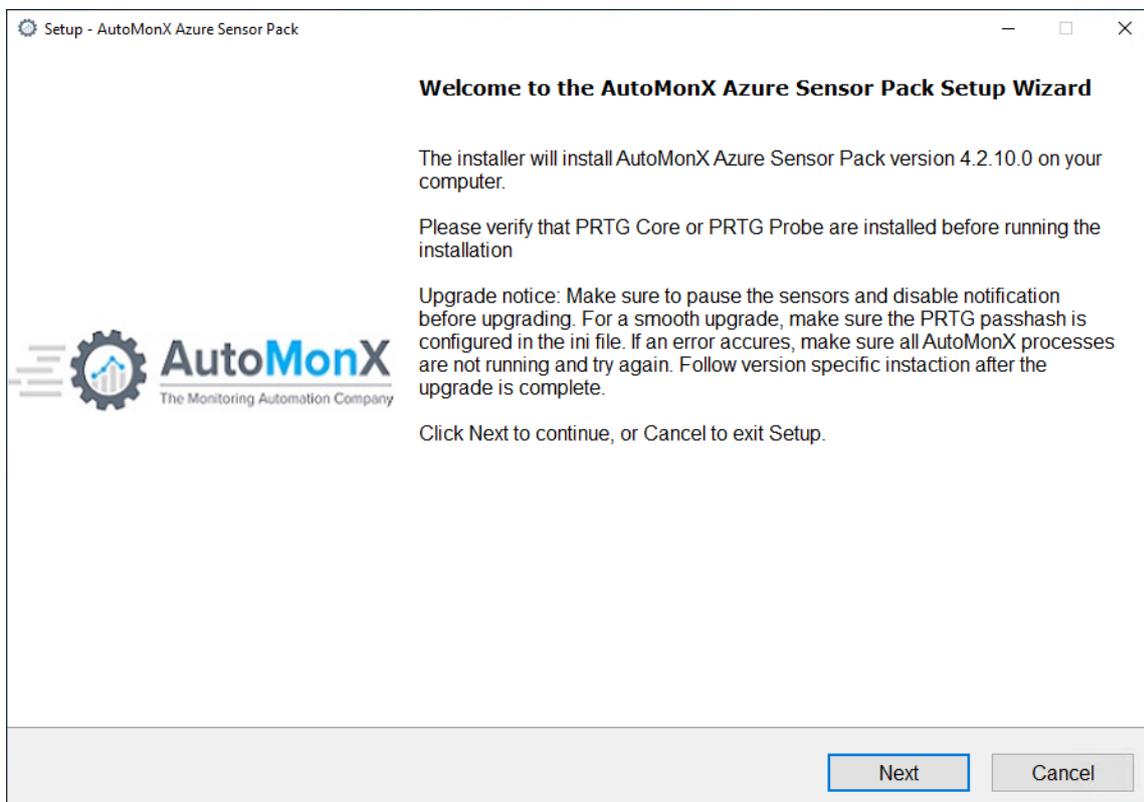
AutoMonX\_PRTG\_Automation.ini - Notepad

File Edit Format View Help

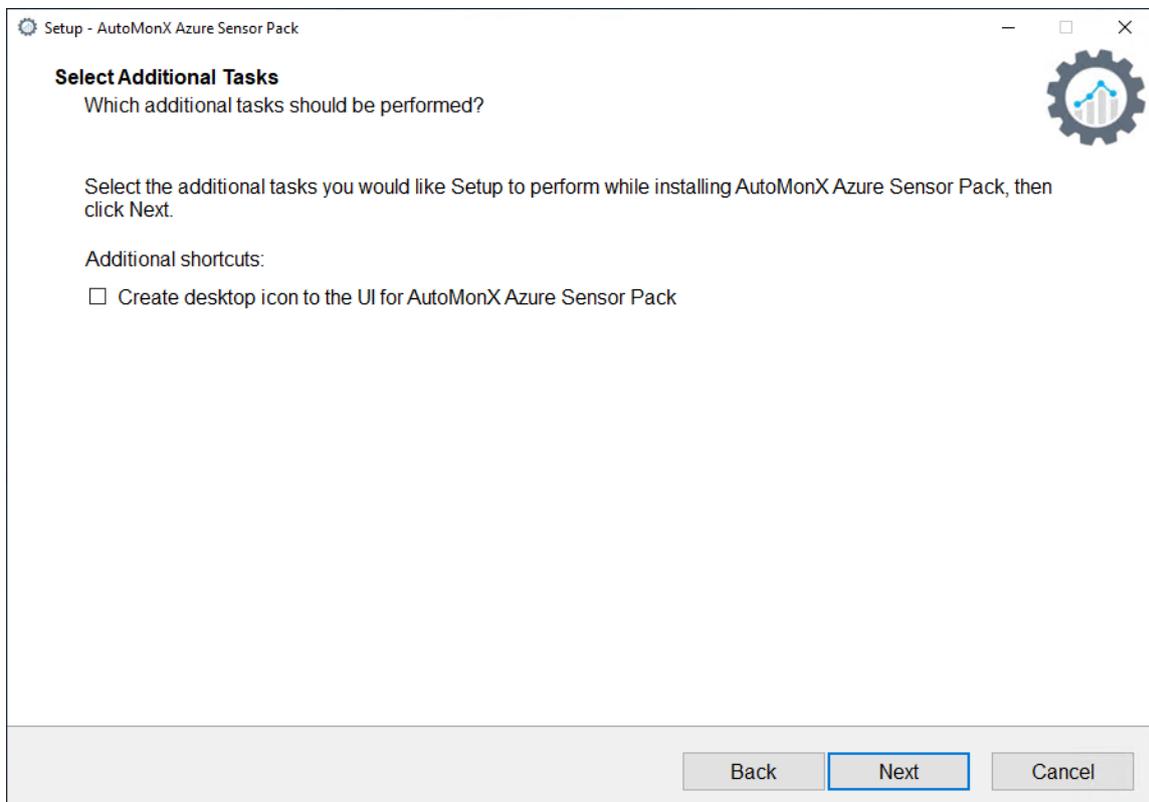
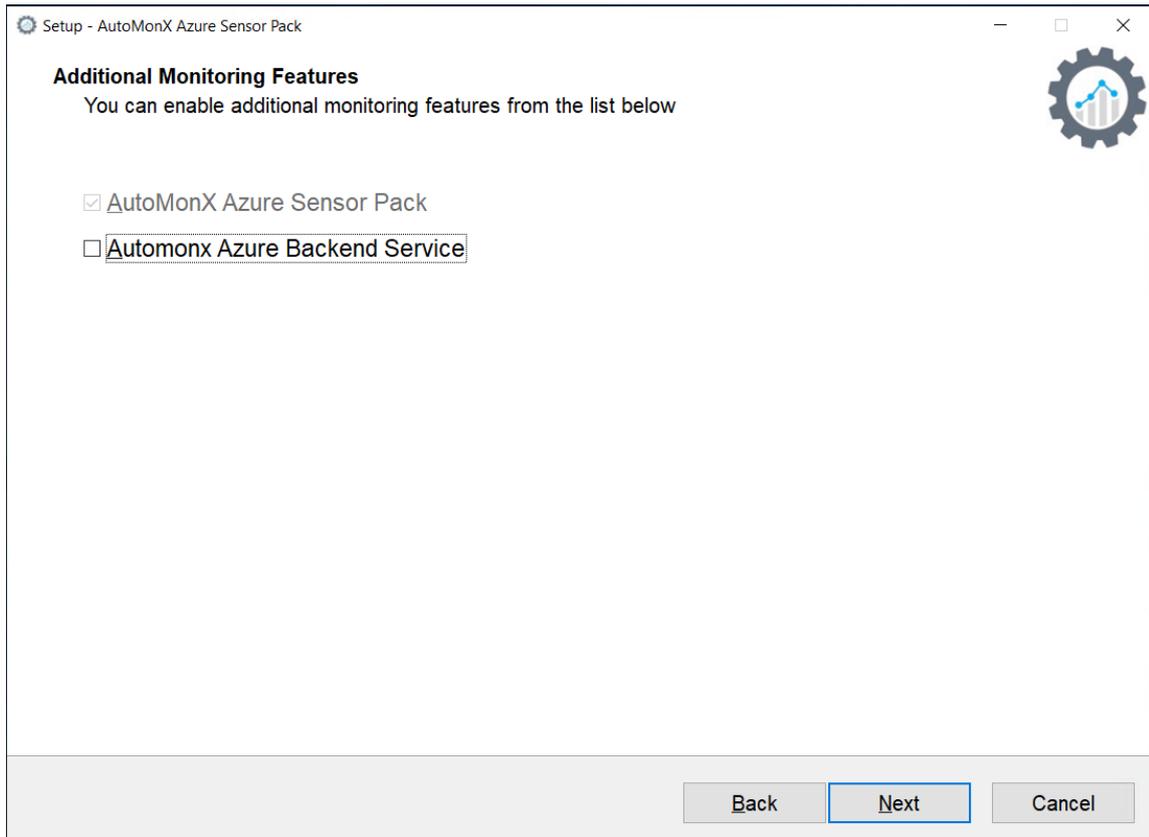
```
FIRST_CHECK_TIMEOUT=15  
SECOND_CHECK_TIMEOUT=5
```

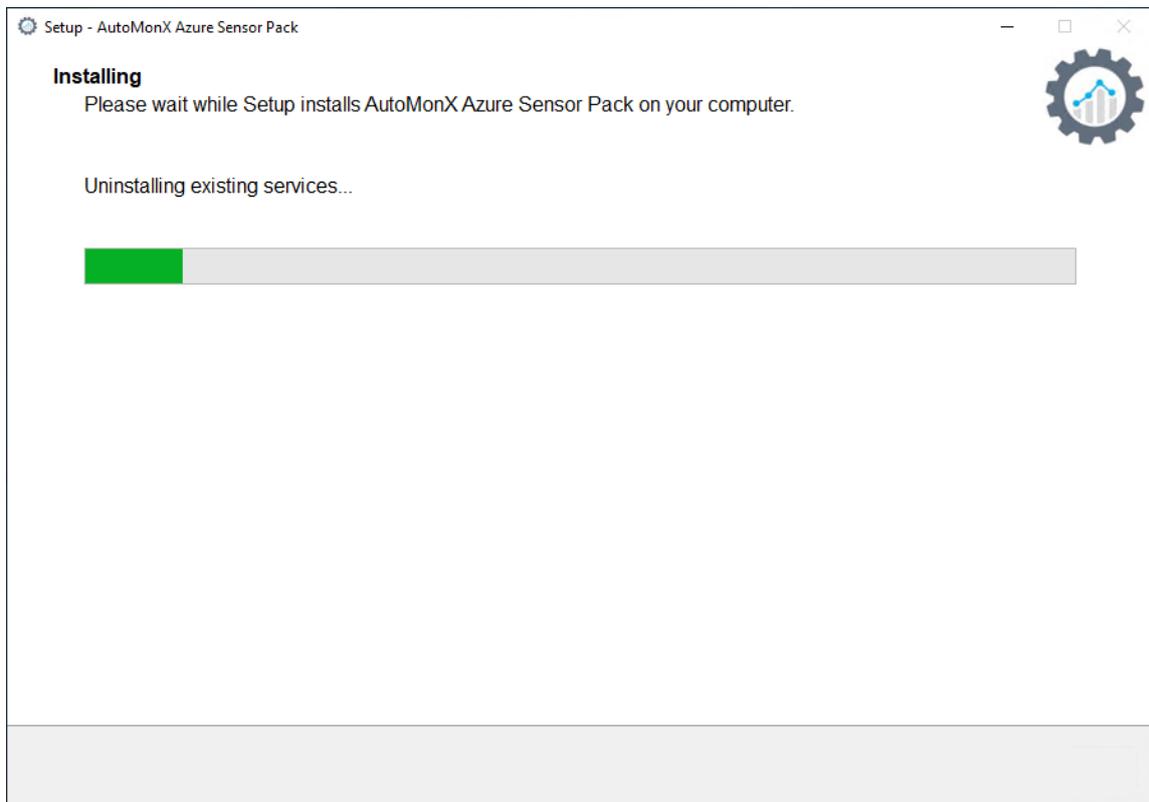
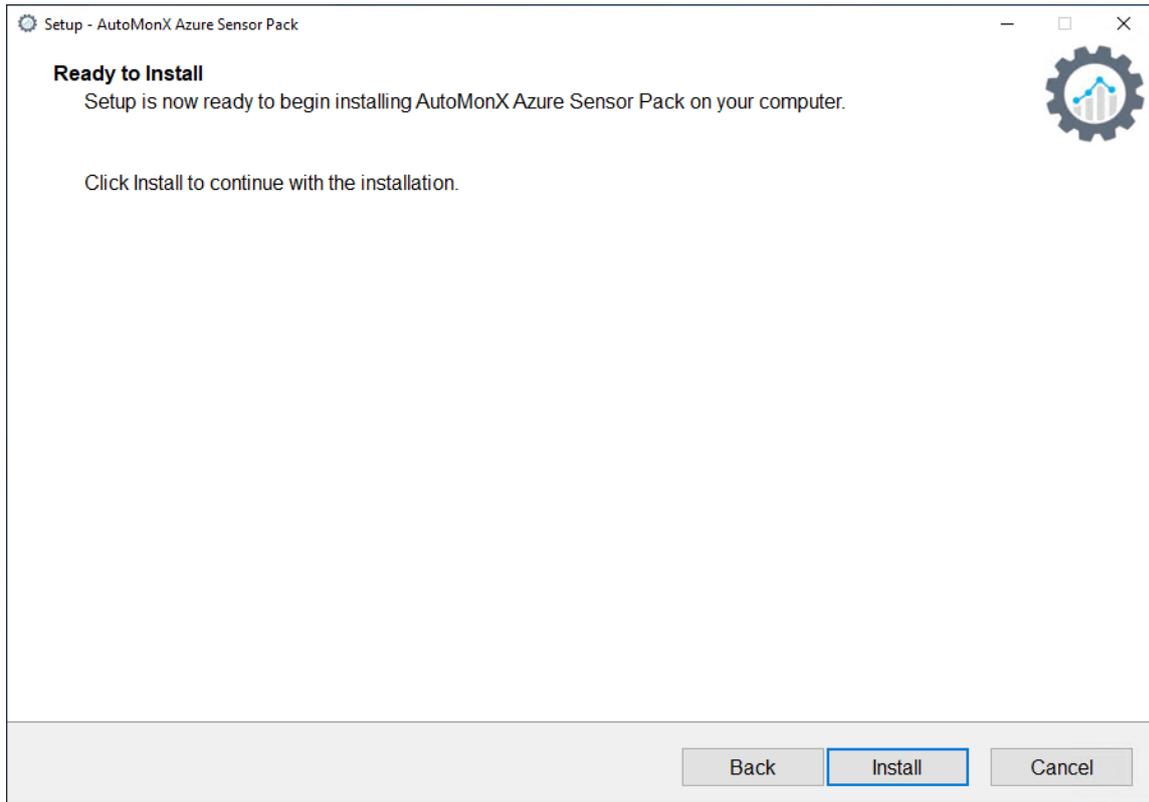
```
[Connections]  
PRTG_USER=prtgadmin  
PRTG_SERVER=127.0.0.1  
PRTG_PORT=443  
HTTPS_CONNECTION=1  
PRTG_PASSHASH=4224444444
```

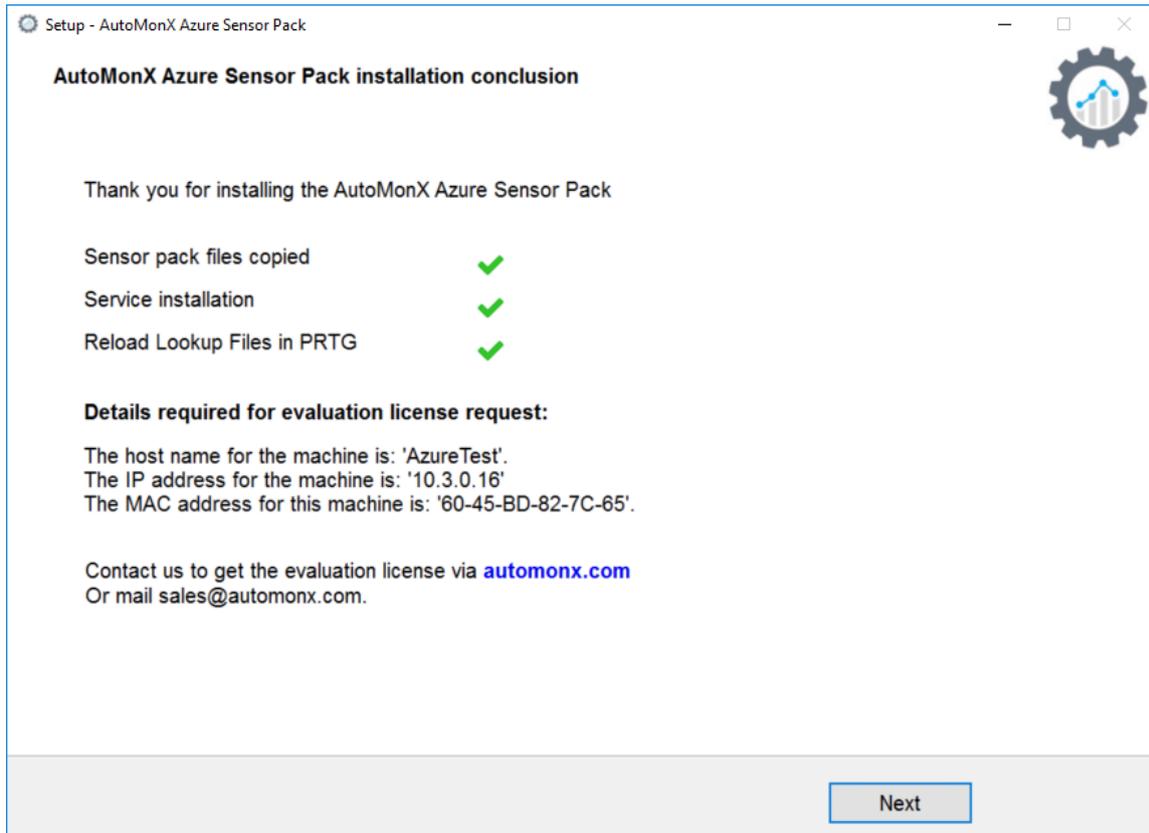
- Make a backup of the entire AutoMonX folder.
- Start the installer and follow the instructions.



When upgrading make sure to tick the AutoMonX Azure Backend Service every time if it is already installed:







- If an error occurred while updating the Lookup files, update them manually [as explained in section 4.6.](#)
- Resume the sensors in PRTG.

## 5.6 Azure Sensor pack Service – INI config

The following table shows the configurable settings in the AutoMonX\_AzureSensor.ini file.

Parameter	Default Value	Details
AZURE_AD_ID	Empty	This is the application connection ID. Can be retrieved using the following guide: <a href="https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal#get-application-id-and-authentication-key">https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal#get-application-id-and-authentication-key</a> Or refer to this section: <a href="#">Troubleshooting Retrieving Azure application ID and tenant ID</a>
AZURE_PASS	Empty	This is the application connection authentication key Can be retrieved using the same guide as above. Or refer to this section: <a href="#">Troubleshooting Retrieving Azure Application Pass</a>
AZURE_TENANT	Empty	This is the Tenant Id of the Active Directory Azure Application. Can be retrieved using the same guide as above. Or refer to this section: <a href="#">Troubleshooting Retrieving Azure application ID and tenant ID</a>
IGNORE_INVALID_DATA_RETRIES	2	How many times should the Sensor Pack ignore invalid data messages from Microsoft Azure
IGNORE_ERROR_RETRIES	2	How many times should the Sensor Pack ignore temporary errors conditions
SEND_ERROR_RETRIES	20	Retry count for getting data from an unresponsive resource.
CONNECTION_AND_TOKEN_TIMEOUT	60	Data fetch timeout from Azure Portal
REQUEST_TIMEOUT	80	This setting controls how long a sensor exe will stay alive until it reaches a timeout. Should be set higher as more sensors are added.
PREDATA_VALID_SEC	700	Sensor data validation intervals.
PROCESSING_TIMEOUT	200	Sensor data fetch timeout in seconds.

DATA_CACHING_SEC	<b>200</b>	Minimum time to wait between the same Azure API calls, to lower the number of fetches in case of excessive PRTG calls.
THREAD_NUMBER	<b>8</b>	The number of threads the Azure Sensor will use for running. More threads mean better Sensor efficiency but higher CPU. See <a href="#">Service Threads</a> section for more information.
SERVICE_RESTART_MIN	<b>1440</b>	This setting controls how long in minutes to keep the service running before a controlled restart.
AZURE_SELF_MON_SERVICE_CHECK	<b>60</b>	The interval in seconds between performing the service functionality check.
AZURE_SELF_MON_MEMORY_LIMIT_MB	<b>1000</b>	The limitation in MB of the amount of RAM taken up by the AutoMonX processes. Lowering this might cause unwanted frequent restarts to the service.
DEBUG_LOG_DIR	<b>LOG_DIR</b>	This setting tells the Sensor Pack where to store its debug log files. The default LOG_DIR points to the AutoMonX\Azure\Log folder. Leave blank to point to the PRTG log folder (Usually: C:\ProgramData\Paessler\PRTG Network Monitor\Logs (Sensors))
SERVICE_DEBUG	<b>0</b>	This setting runs the service in debug mode. Requires a service restart for every change of setting. The debug log is written to the AutoMonX_ServiceDebugLogger file. <b>This setting should be turned off when a debug mode is not necessary.</b>
HTTP_SENSOR_MODE	<b>1</b>	Discovery will generate sensors of type HTTP Data Advanced. To fall back to EXE/Script Advanced type sensors, turn this value to 0.
SET_SENSOR_NOT_FOUND_TO_WARN	<b>false</b>	When set to TRUE, will automatically mark removed resources as warning instead of error.
CLOUD_VLOOKUP	<b>false</b>	Support for non-custom lookup when using PRTG cloud. Configure this setting before adding to PRTG.
AZURE_DEPRECATED_METRICS	<b>OFF</b>	When set to TRUE, will allow the discovery of deprecated metrics from Azure Portal.
ENABLE_PREVIEW_METRICS	<b>FALSE</b>	When set to TRUE, will allow the discovery of preview metrics from Azure Portal.
DISCOVER_BACKUP_JOBS_SUMMARY	<b>FALSE</b>	Change to True if you wish the Backup jobs to be discovered in a summary sensor per backup vault.
AZURE_VM_SERVICE_HEALTH_SET_BY_USER	<b>CRITICAL</b>	When set to WARNING, will mark any virtual machines stopped by user as warning instead of error.

DISABLE_DEFAULT_THRESHOLDS	<b>FALSE</b>	When set to TRUE, will disable all default thresholds from channels (not including lookups)
AZURE_USED_REGIONS		Used for displaying the relevant channels in the quota sensors. List the used regions by your organization separated by commas to limit the channels to the regions you would like to see.
AZURE_QUOTA_LIMIT	<b>80</b>	The limit to alert quota usage.
<metric_name>_Aggregation_Type		Custom parameters to edit the aggregation type of specific metrics.
AZURE_GOV_ENABLED=TRUE AZURE_GOV_COUNTRY=USA	<b>FALSE</b>	Support for Azure Gov customers and their country (USA, Germany)
VM_NAME_FOR_SNMP	<b>TRUE</b>	Set to TRUE to activate SNMP discovery by PRTG for VMs in the SNMP section in the UI. Set to FALSE to remove the "Node" label from the VM name.
AZURE_VM_STATES_NOTIFY	<b>FALSE</b>	Controls the overall VM status sensor status. When set to FALSE the VM states are disregarded. When set to TRUE, users can set the status of the sensor to Warning or Critical based on specific VM states in Azure
AZURE_VM_STATES_WARN= AZURE_VM_STATES_CRITICAL=		These keys control the VM status sensor severity based on the VM status in Azure:  AZURE_VM_STATES_CRITICAL=Stopped,Deallocated  AZURE_VM_STATES_WARN=Stopping, Deallocating,Deallocated
[PROXY] PROXY_SERVER= PROXY_PORT=		Add these lines to configure the Sensor Pack for Azure to connect to Azure API via Proxy
azure_login_url azure_mgmt_core_url azure_mgmt_url azure_graph_url		Set different URLs to the default ones, for example for 21Vianet. This setting in the file AzureConnProfiles.ini will appear with <tenant_internal_id>_

## 5.7 Configuring Azure API Endpoints

Azure provides region-specific API endpoints to cater to different regulatory and operational requirements:

**Azure Global** – The default and basic support provided by the AutoMonX Sensor Pack, covering standard Azure deployments worldwide.

**Azure Government** - Designed for US government agencies and their partners, Azure Government provides a dedicated cloud that meets stringent government security and compliance requirements. It enables the transformation of mission-critical workloads to the cloud while accommodating data subject to US government regulations and requirements.

**21Vianet** - Microsoft Azure operated by 21Vianet (Azure in China) is a physically separated instance of cloud services located in China. Operated independently by Shanghai Blue Cloud Technology Co., Ltd., a wholly owned subsidiary of Beijing 21Vianet Broadband Data Center Co., Ltd., this service complies with China-specific regulations.

For users with Azure Government or 21Vianet instances, some adjustments must be made in the configuration of AutoMonX Sensor Pack.

For single tenant simply add this to the file “AutoMonX\_AzureSensor.ini”:

Azure Government:

```
AZURE_GOV_ENABLED=TRUE  
AZURE_GOV_COUNTRY=USA
```

21Vianet:

```
AZURE_GOV_ENABLED=TRUE  
AZURE_GOV_COUNTRY=CN
```

For multitenant users that have both Azure Global tenants and others, or any user with unique Azure endpoints, can configure the API endpoints for the tenant in the file “AzureConnProfiles.ini”.

For example, these are the required configurations for 21Vianet:

```
1_AZURE_LABEL=ChinaTenant  
1_AZURE_AD_ID=xxx  
1_AZURE_PASS=xxx  
1_AZURE_TENANT=xxx  
1_azure_mgmt_core_url=https://management.core.chinacloudapi.cn  
1_azure_login_url=https://login.chinacloudapi.cn  
1_azure_mgmt_url=https://management.chinacloudapi.cn  
1_azure_graph_url=https://microsoftgraph.chinacloudapi.cn
```

## 5.8 Enabling HTTPS sensors

For Secure connection between the PRTG and the AutoMonX Azure Sensor Pack, generate a certificate for the probe server and place it in:

“AutoMonX\Azure\Certs”

The files must be named “certificate.crt” and “private.key”.

Activate the ssl by adding this configuration to the file

“Automonx\_AzureSensor.ini”:

USE\_SSL=TRUE

Also ensure the correct server name is saved into

SSL\_SERVER\_NAME=<probe\_server\_dns>

The name must match the one in the certificate. The default value is 127.0.0.1.

Now newly discovered sensors will be created in PRTG with https URLs.

**Important** – the HTTPS port in use is 8492

## 6 Introducing Multi-Tenant Azure Monitoring

The AutoMonX Sensor Pack for Azure is capable to automatically discover and monitor multiple Azure Tenants from a single PRTG probe. This major improvement allows CSPs/MSPs and large enterprises to monitor their entire Azure estate without being limited by Tenant boundaries.

**Important License information:** To support the Multi-Tenant version, new license types are available as specified below. If you require to monitor multiple tenants, make sure to contact AutoMonX sales [sales@automonx.com](mailto:sales@automonx.com) to obtain the most suitable license type.

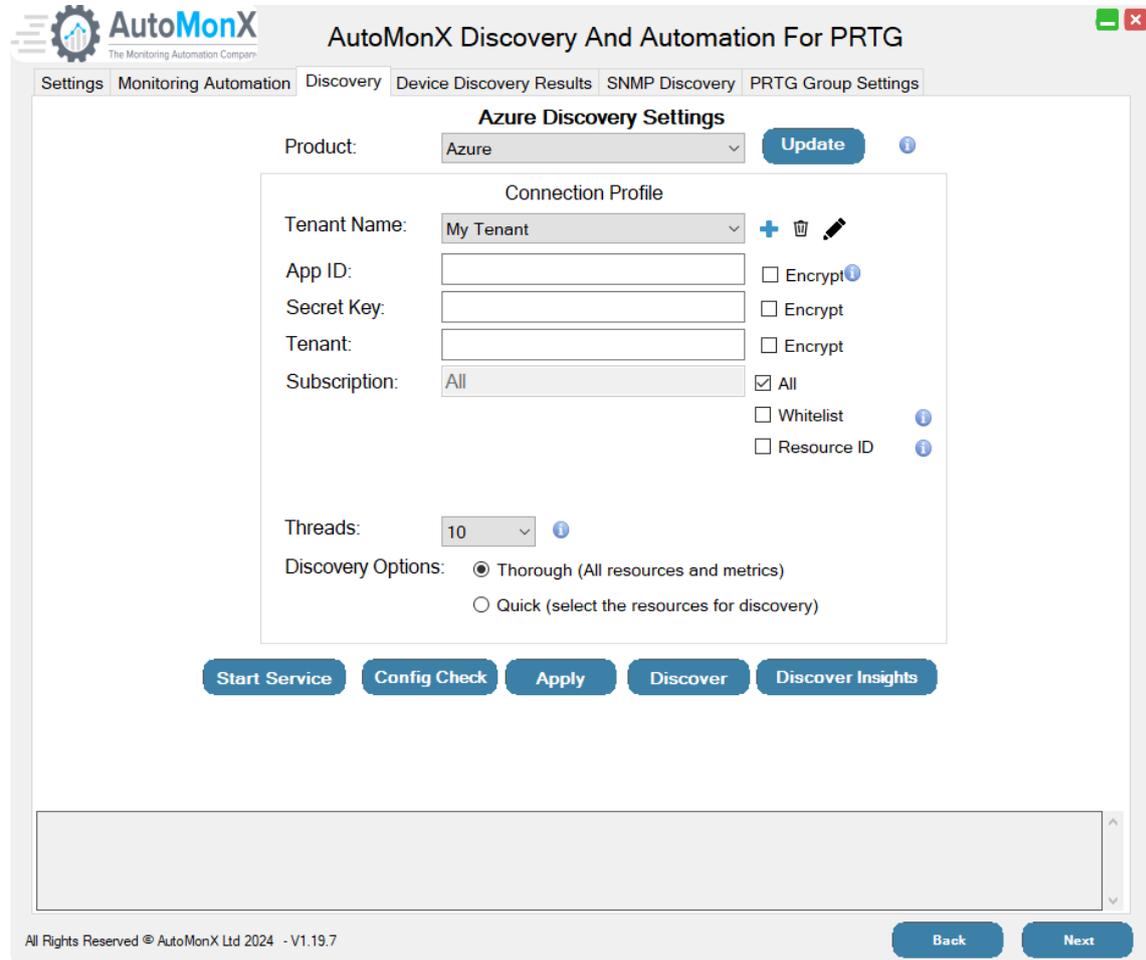
### 6.1 Multi-Tenant License Types Explained

Additional license types are available to facilitate the monitoring of resources in multiple Azure tenants.

- The license is bound to PRTG probe machine
- Single-tenant Azure Sensor pack licenses are still available
- Multi-Tenant licenses are sold in packs:
  - 5 Tenants
  - 10 Tenants
  - 15 Tenants
  - 20 Tenants
  - 25 Tenants
  - 50 Tenants
  
- Each Multi-tenant license pack allows you to monitor an unlimited number of subscriptions and unlimited number of sensors within the number of tenants you have purchased.
  
- **License boundaries/limitations:**
  - Your PRTG License
  - The number of Tenants in the Multi-Tenant license you have purchased (5,10,25,50)
  - The license is bound to a specific PRTG Probe
  - The PRTG Probe physical capabilities

## 6.2 Configuring multi-Tenant Discovery

You need to configure the details of each tenant you wish to add to Azure Sensor Pack auto-discovery by using our UI.



### Connection Profiles:

**Tenant Name:**  +  

Connection Profiles have been introduced for quick and easy identification of tenants in our UI and in PRTG. You can edit these labels by using the editing icons. You can move between different Connection Profiles by using the drop-box menu.

**Important:** Once a connection profile was added and its respective tenant subscriptions and their Azure resources have been added to PRTG, it is better to keep the same Connection Profile names and not modify them, as it may create duplicate entries in PRTG during additional rounds of auto-discovery and Monitoring automation activities.

**Connection Profile editing icons:**



- Plus sign: Add new connection profile details
- Trashcan: Delete a connection profile from the list
- Pencil: Modify tenant label

### 6.3 Encryption of Connection Profile details

Version 4.x introduces the ability to encrypt the Azure connection profile details. You can choose to encrypt any of the connection details: App ID, Secret Key or Tenant ID by ticking the check boxes on the right as seen below:

App ID:		<input type="checkbox"/> Encrypt <span style="font-size: 1.2em;">i</span>
Secret Key:		<input type="checkbox"/> Encrypt
Tenant:		<input type="checkbox"/> Encrypt

**Important:** Once the connection details of a profile are encrypted, there is no way to decrypt them via our software for you to see. This is by design and aimed at protecting your Azure connection details. It is recommended to store the connection details in a safe place or password management software in case you would need to enter them again.

Encryption via CLI is available with the command:

```
AutoMonX_AzureCollector.exe -enc_data <secret_key>
```

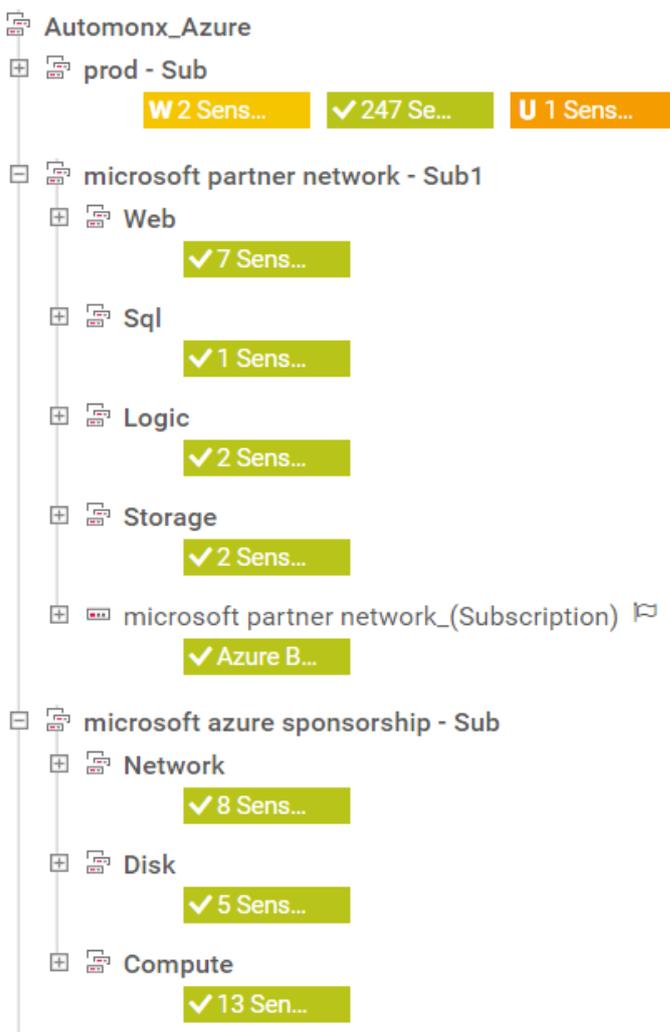
### 6.4 The Azure Sensor hierarchy in PRTG

Our Monitoring Automation creates a tree-structured hierarchy based on some initial configuration made in our UI. You need to provide the top-level group for all Azure assets (needs to be manually created in PRTG) and specify the labels for each Tenant in our UI. The rest of the hierarchy would be automatically created for you in PRTG as seen below:

**PRTG Probe** (where the Azure sensor is installed)

- **Azure** (The top-level PRTG group you need to manually create, could be any name)
  - o **Tenant label** (in a format of <Tenant Label>-<Last 4 digits of Tenant id> i.e. AutoMonX-3234. The group must be created manually.
    - **Subscription name** (in a format of <sub name>-sub. i.e. Prod – sub)
      - **Resource Group(s)** – Automatically generated by Monitoring Automation (i.e. SQL, LogicApp, Storage etc)

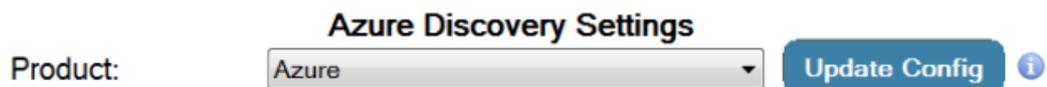
See below an example of the tree structure created automatically by our Monitoring automation:



## 6.5 Migration to Multi-Tenant Version

If you are planning to monitor multiple Azure tenants via the Multi-Tenant functionality and your existing Azure Sensor Pack version is prior to v4.x, follow the steps below:

1. **Upgrade the software version:** Complete the software upgrade steps as explained in [Upgrade instructions](#).
2. **Add new Azure tenant(s):** Start our Configuration UI and add the new Azure Tenants you want to auto-discover and monitor.
3. **Modify Tenant label(s):** Your existing Tenant will get a default label: MyTenant. You can modify it via the UI to a label of your choice.
4. **Update the Configuration:** Use the “Update Config button” to start the configuration update.



The following warning will be shown:

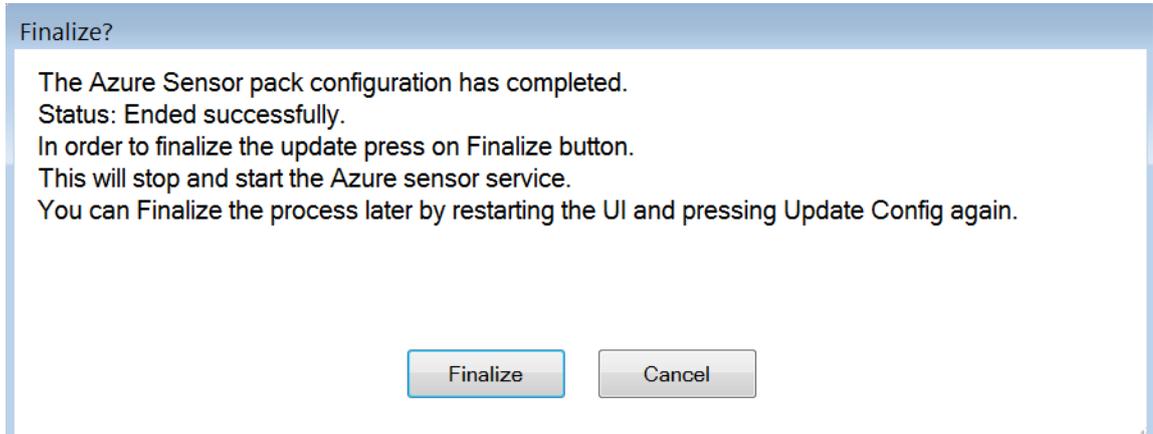
Update Configuration

You are about to update your Azure sensor settings to a new format.  
The process may take several hours but monitoring will continue normally.  
Make sure to update the Azure sensor pack files to the latest version before starting the process.  
See the deployment guide for details

Press on Update button to continue with the process or Cancel it.

5. **Upgrade Process:** The Azure Sensor Pack will start to re-discover the tenants in a sequential order. During the re-discovery, monitoring will continue without interruptions

6. **Finalizing the update:** At the end of the process that may take several hours, the following messages will be shown:



Press Finalize button to complete the update process. During this step the Azure Sensor pack service will stop and start automatically.

7. **Adding the new Tenant(s) to PRTG:** Use our UI to add the new Tenant(s) to PRTG as explained in the next section of this guide.

## 7 Auto Discovery and Monitoring Automation

### 7.1 Automatic Discovery of Azure Resources

The AutoMonX Sensor Pack for Microsoft Azure needs to scan the Azure environment for any resources it can monitor. In auto-discovery mode, the Sensor Pack will generate a list of all the Azure resources in your environment that it can monitor. It would also provide you with the required sensor configuration to monitor these resources.

**Important:** Starting version 4.2.x, the default sensor type used for integrating the Azure Sensor Pack with PRTG is HTTP Data Advanced as it has much lower performance hit on the PRTG Probe. If you wish to continue utilizing the EXE/Script Advanced type sensors, change the HTTP\_SENSOR\_MODE configuration in *the AutoMonX\_AzureSensor.ini* file prior to running discovery

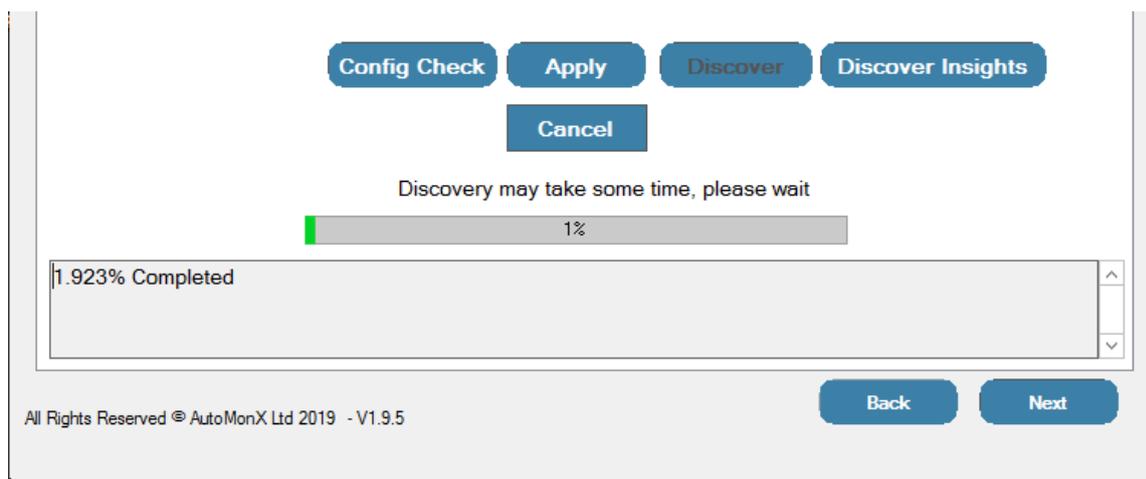
There are two discovery modes:

1. Thorough Discovery
2. Quick Discovery

The Quick discovery is currently only available via CLI, read more [here](#).

Press the “Discover” button to start the Azure resources discovery. At this stage, the auto discovery will take place.

**Note:** Depending on the network connection, the Azure API response time and taking into account the size of your Azure deployment, it can take between a few minutes to several hours to complete.



You can cancel the discovery process while it is running by pressing the “Cancel” button.

The discovery process can take some time, follow-up the progress by checking the message area at the bottom of the screen.

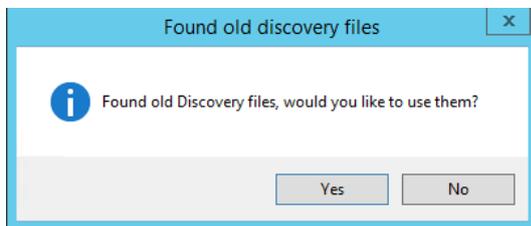
Timeout messages may appear sometimes during the discovery process, but you can safely ignore them if they last no longer than 10 minutes.

When auto-discovery has completed, the following window will pop-up. Now you can move to the next tab and examine the discovery results.

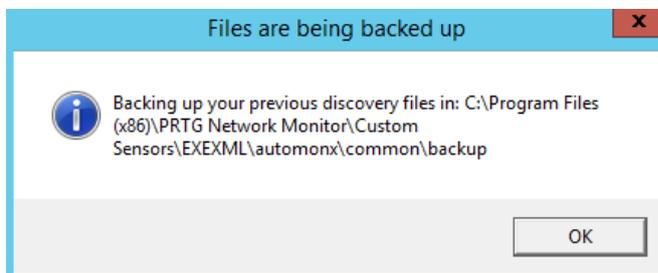


## 7.2 Previous Discovery Results handling

In cases there are previous auto-discovery results, the UI will offer to use them instead of re-discovering again the Azure resources, which can be time consuming.



Before starting auto-discovery, the UI will backup any previous discovery results and pop-up the following window:



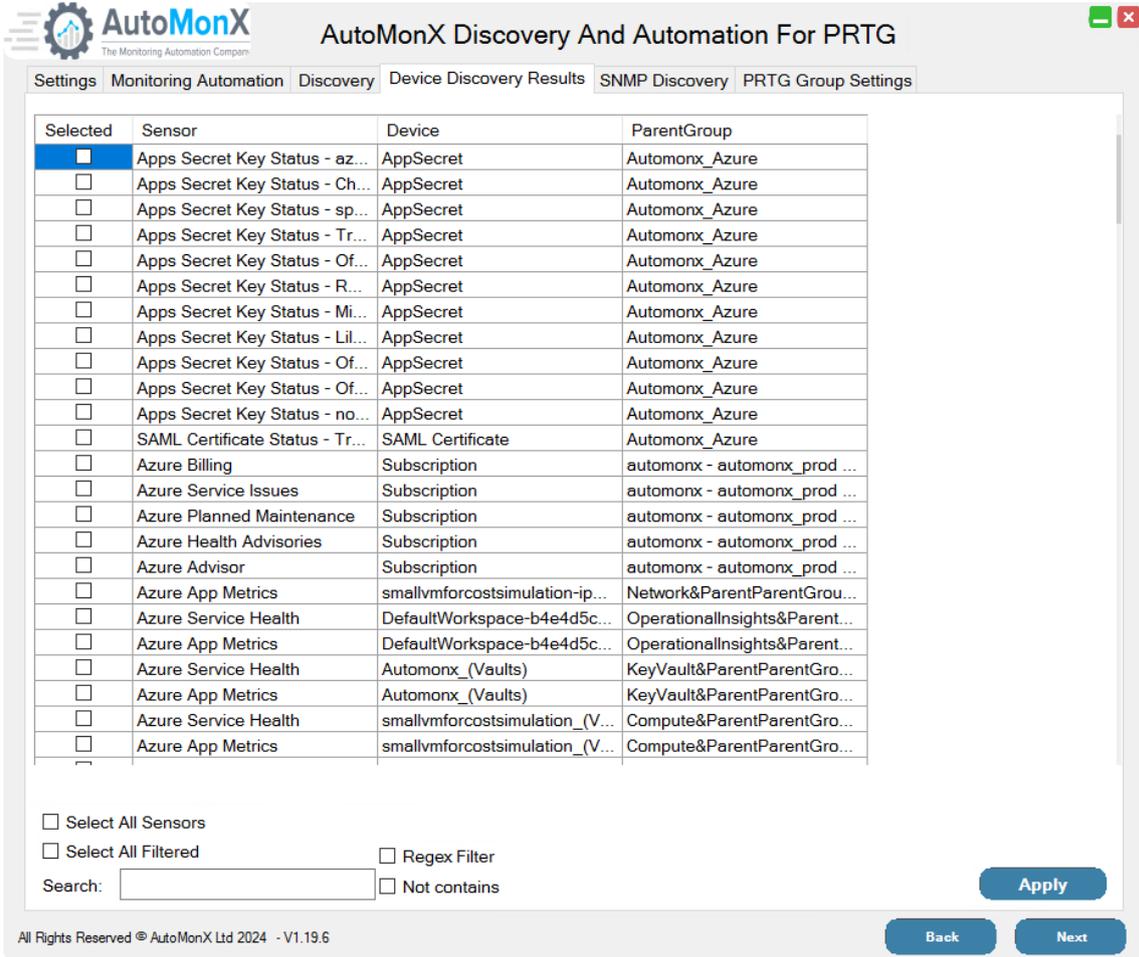
### 7.3 Sensor types created by the Azure Sensor pack

The Azure Sensor Pack auto-discovery will create several sensor types, most of them are custom sensors:

Sensor Type	Description	Actions in Azure Portal / PRTG
Azure App Metrics	Resource-specific performance metrics available via the Azure API, multiple channels in PRTG	Performance Metrics must be enabled per each resource via the Azure portal
Azure Service Health	Resource health as reported in Azure portal, two channels	Resource health needs to be configured via the Azure portal
Azure VM Multi-Disk	For Virtual Machines only, disk metrics polled with Log Analytics.	Log Analytics must be enabled on the virtual machine.
Azure LogicApp	The workflow status of a Logic App.	
Azure DataFactory DataSet	Data Factory Status	
Apps Secret Key Status	The expiry time of each secret key under available Apps.	
Azure Billing	Azure Billing data, several channels that cover supported and un-supported resources	
AutoMonX License	Self-monitoring sensor that shows license consumption and days left for maintenance and license to expire (if applicable)	N/A

## 7.4 Selecting Azure Sensors for Monitoring

Press “Next” to move to the next tab. All the discovered Azure resources would be presented:



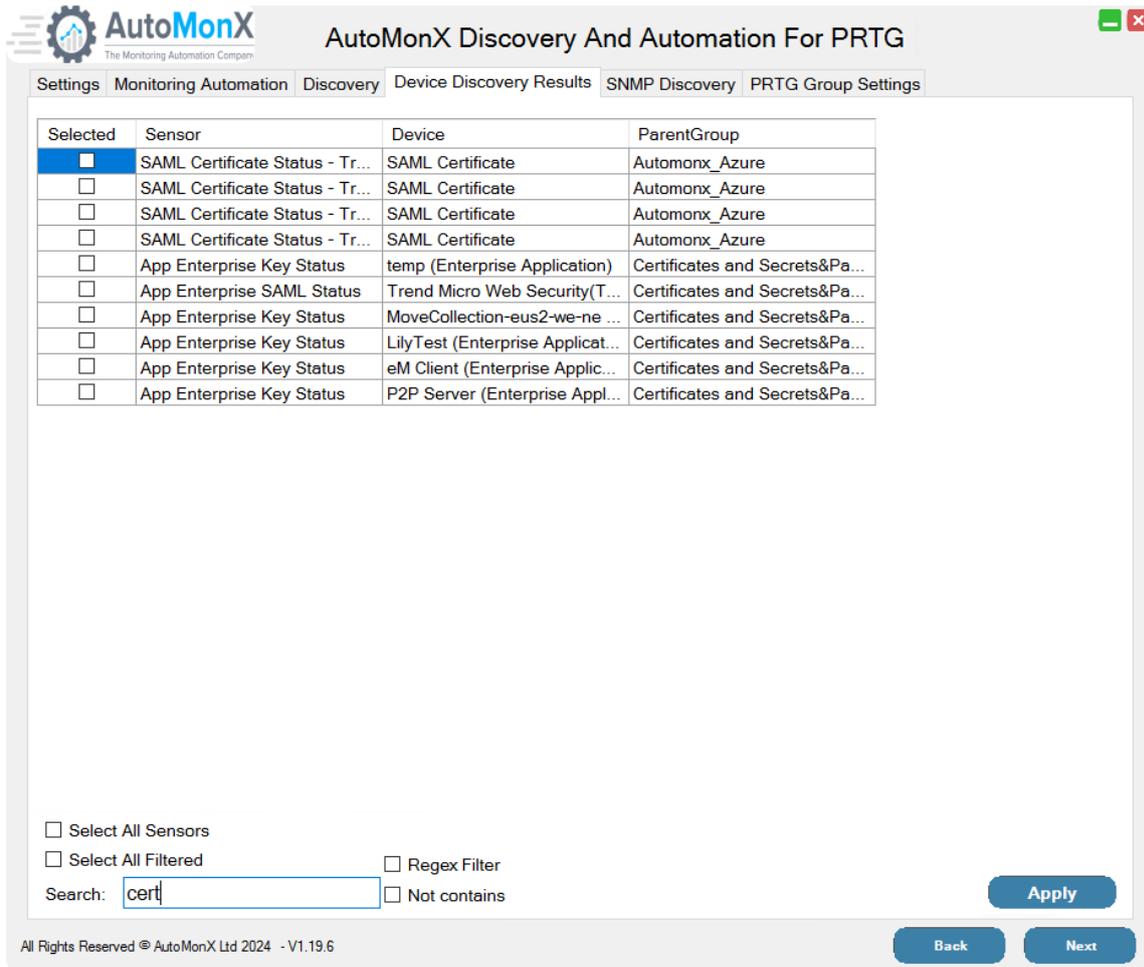
The screenshot shows the 'Discovery' tab in the AutoMonX interface. A table lists various sensors with checkboxes for selection. Below the table are options for 'Select All Sensors', 'Select All Filtered', a search box, and checkboxes for 'Regex Filter' and 'Not contains'. An 'Apply' button is also present.

Selected	Sensor	Device	ParentGroup
<input checked="" type="checkbox"/>	Apps Secret Key Status - az...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Ch...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - sp...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Tr...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Of...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - R...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Mi...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Lil...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Of...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - Of...	AppSecret	Automonx_Azure
<input type="checkbox"/>	Apps Secret Key Status - no...	AppSecret	Automonx_Azure
<input type="checkbox"/>	SAML Certificate Status - Tr...	SAML Certificate	Automonx_Azure
<input type="checkbox"/>	Azure Billing	Subscription	automonx - automonx_prod ...
<input type="checkbox"/>	Azure Service Issues	Subscription	automonx - automonx_prod ...
<input type="checkbox"/>	Azure Planned Maintenance	Subscription	automonx - automonx_prod ...
<input type="checkbox"/>	Azure Health Advisories	Subscription	automonx - automonx_prod ...
<input type="checkbox"/>	Azure Advisor	Subscription	automonx - automonx_prod ...
<input type="checkbox"/>	Azure App Metrics	smallvmforcostsimulation-ip...	Network&ParentParentGrou...
<input type="checkbox"/>	Azure Service Health	DefaultWorkspace-b4e4d5c...	OperationalInsights&Parent...
<input type="checkbox"/>	Azure App Metrics	DefaultWorkspace-b4e4d5c...	OperationalInsights&Parent...
<input type="checkbox"/>	Azure Service Health	Automonx_(Vaults)	KeyVault&ParentParentGro...
<input type="checkbox"/>	Azure App Metrics	Automonx_(Vaults)	KeyVault&ParentParentGro...
<input type="checkbox"/>	Azure Service Health	smallvmforcostsimulation_(V...	Compute&ParentParentGro...
<input type="checkbox"/>	Azure App Metrics	smallvmforcostsimulation_(V...	Compute&ParentParentGro...

Select All Sensors  
 Select All Filtered       Regex Filter  
 Search:        Not contains

All Rights Reserved © AutoMonX Ltd 2024 - V1.19.6           

Select the sensors you want to add to PRTG by clicking on the relevant checkbox on the left side of the table. You can also click on “Select All” to mark all the sensors. There is also an option to present only certain sensors by using the Search window as seen below:



AutoMonX Discovery And Automation For PRTG

Settings | Monitoring Automation | Discovery | Device Discovery Results | SNMP Discovery | PRTG Group Settings

Selected	Sensor	Device	ParentGroup
<input checked="" type="checkbox"/>	SAML Certificate Status - Tr...	SAML Certificate	Automonx_Azure
<input type="checkbox"/>	SAML Certificate Status - Tr...	SAML Certificate	Automonx_Azure
<input type="checkbox"/>	SAML Certificate Status - Tr...	SAML Certificate	Automonx_Azure
<input type="checkbox"/>	SAML Certificate Status - Tr...	SAML Certificate	Automonx_Azure
<input type="checkbox"/>	App Enterprise Key Status	temp (Enterprise Application)	Certificates and Secrets&Pa...
<input type="checkbox"/>	App Enterprise SAML Status	Trend Micro Web Security(T...	Certificates and Secrets&Pa...
<input type="checkbox"/>	App Enterprise Key Status	MoveCollection-eus2-we-ne ...	Certificates and Secrets&Pa...
<input type="checkbox"/>	App Enterprise Key Status	LilyTest (Enterprise Applicat...	Certificates and Secrets&Pa...
<input type="checkbox"/>	App Enterprise Key Status	eM Client (Enterprise Applic...	Certificates and Secrets&Pa...
<input type="checkbox"/>	App Enterprise Key Status	P2P Server (Enterprise Appl...	Certificates and Secrets&Pa...

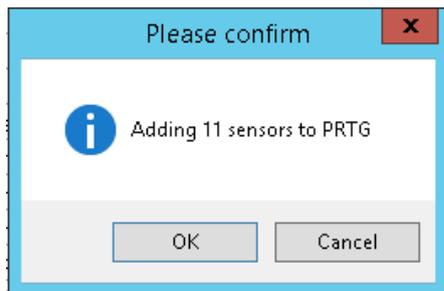
Select All Sensors  
 Select All Filtered  Regex Filter  
 Search:   Not contains

Apply

All Rights Reserved © AutoMonX Ltd 2024 - V1.19.6

Back Next

Click “Apply” to save your settings. A confirmation window will pop-up. Click “OK” to confirm or “Cancel”.



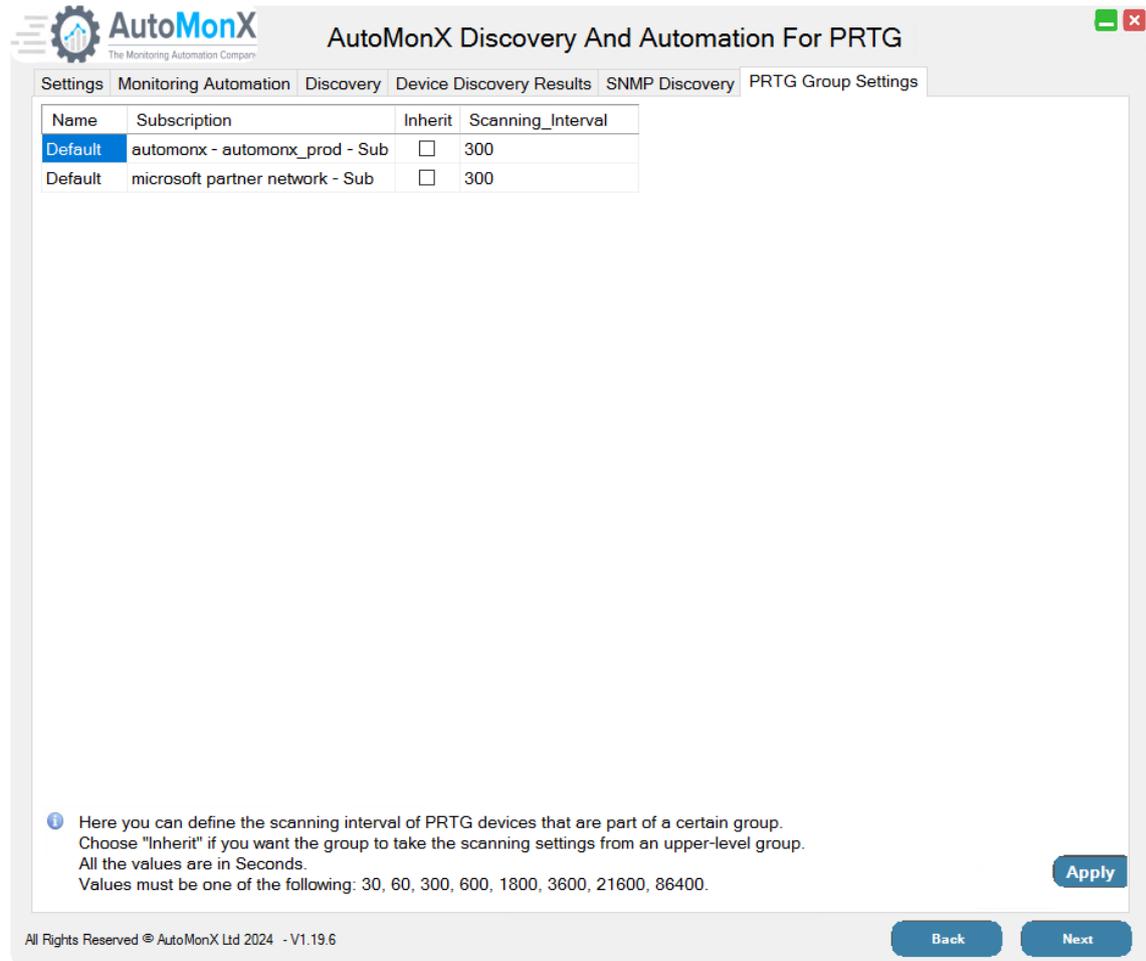
Press “Next” to proceed to the “Add sensors to PRTG” tab.

See section [6.10](#) for the location of the Discovery results report.

## 7.5 Configuring PRTG Group Settings

The PRTG Group Settings window provides you with the option to change the scanning intervals of the sensors under the Azure PRTG group. By default, the sensors are added with a 300 second interval (5 minutes). You can tick on the “inherit” checkbox to add the sensors in the Azure groups with the PRTG interval inherit system. More information can be seen it the bottom of the window.

This is an optional setting. Use the “Next” button to move to the next tab



Name	Subscription	Inherit	Scanning_Interval
Default	automonx - automonx_prod - Sub	<input type="checkbox"/>	300
Default	microsoft partner network - Sub	<input type="checkbox"/>	300

ⓘ Here you can define the scanning interval of PRTG devices that are part of a certain group.  
 Choose "Inherit" if you want the group to take the scanning settings from an upper-level group.  
 All the values are in Seconds.  
 Values must be one of the following: 30, 60, 300, 600, 1800, 3600, 21600, 86400.

All Rights Reserved © AutoMonX Ltd 2024 - V1.19.6

## 7.6 Auto-Discovery of Azure Virtual Machines SNMP Sensors

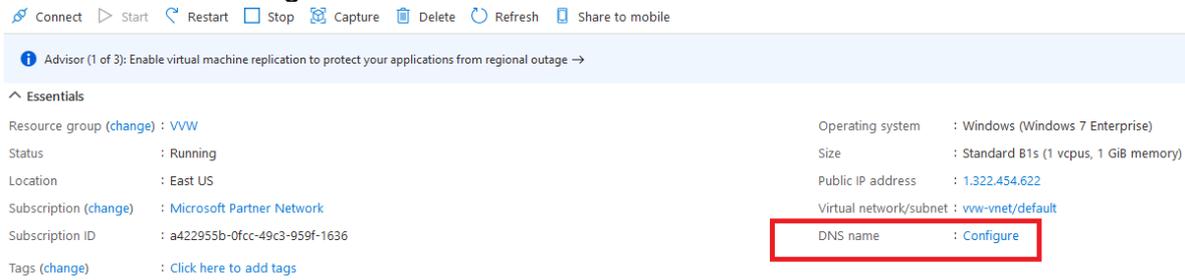
Starting with version v3.32 of the Azure Sensor Pack, it is possible to automatically discover and monitor the SNMP performance metrics and Ping (ICMP) of Virtual Machines (Windows or Linux) hosted in Azure, along with the Azure-based metrics, already available via the Azure Sensor Pack.

Such hybrid approach will allow you to enjoy the best of two worlds – the rich functionality of Azure metrics monitoring coupled with OS-based SNMP performance metrics under a single device in PRTG.

## 7.7 Preparing for Auto-Discovery of Azure Virtual Machines

You need to perform the following tasks before initiating Auto-Discovery of SNMP metrics:

- **DNS** – Configure a DNS record for each VM in the Azure Portal



Connect Start Restart Stop Capture Delete Refresh Share to mobile

Advisor (1 of 3): Enable virtual machine replication to protect your applications from regional outage →

Essentials

Resource group (change) : VVV

Status : Running

Location : East US

Subscription (change) : Microsoft Partner Network

Subscription ID : a422955b-0fcc-49c3-959f-1636

Tags (change) : Click here to add tags

Operating system : Windows (Windows 7 Enterprise)

Size : Standard B1s (1 vcpu, 1 GiB memory)

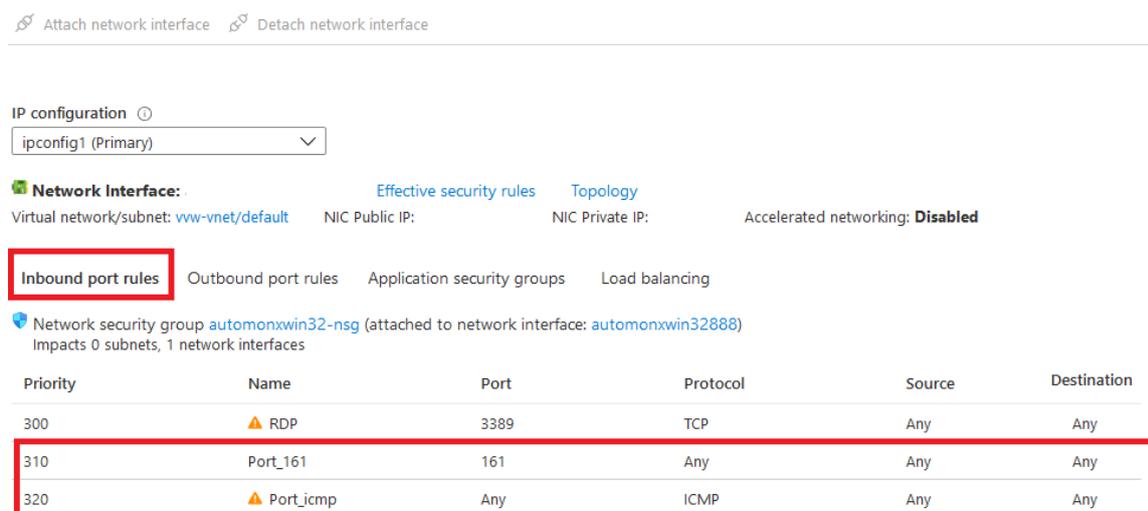
Public IP address : 1.322.454.622

Virtual network/subnet : vvv-vnet/default

DNS name : **Configure**

- **Networking Ports** – Configure and open networking ports for SNMP (UDP 161) and ICMP

### Networking



Attach network interface Detach network interface

IP configuration

Network Interface: Effective security rules Topology

Virtual network/subnet: vvv-vnet/default NIC Public IP: NIC Private IP: Accelerated networking: Disabled

**Inbound port rules** Outbound port rules Application security groups Load balancing

Network security group automonxwin32-nsg (attached to network interface: automonxwin32888)  
Impacts 0 subnets, 1 network interfaces

Priority	Name	Port	Protocol	Source	Destination
300	⚠ RDP	3389	TCP	Any	Any
310	Port_161	161	Any	Any	Any
320	⚠ Port_icmp	Any	ICMP	Any	Any

- **SNMP Service** - install the SNMP service on each Virtual Machine you wish to monitor via the SNMP Discovery.
- **Community String in PRTG:** You must configure the SNMP community string in the PRTG group where plan to create the Azure Virtual Machines

### Credentials for SNMP Devices

 inherit from  Linux Monitor SSH (SNMP  
Version: V2, SNMP Port: 161, SNMP Timeou...)

SNMP Version   v1  
 v2c (recommended)  
 v3

Community String  public

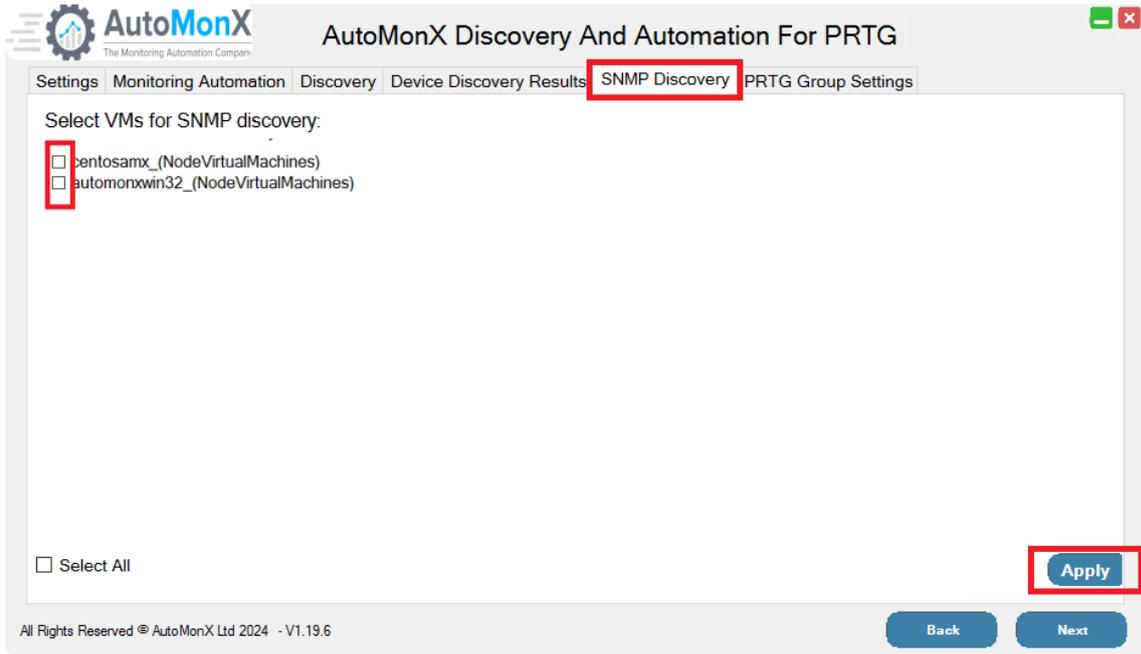
SNMP Port  161

SNMP Timeout (Sec.)  5

## 7.8 Running Auto-Discovery of Azure Virtual Machines

Make sure you have completed the preparations for monitoring the Virtual Machines. Initiate a new Azure resources discovery using our UI.

In the Discovery Results tab, select the Virtual Machines resources it identified and press “Next”. In the SNMP Discovery tab choose the Virtual Machines that have been prepared for monitoring via SNMP and press “Apply”

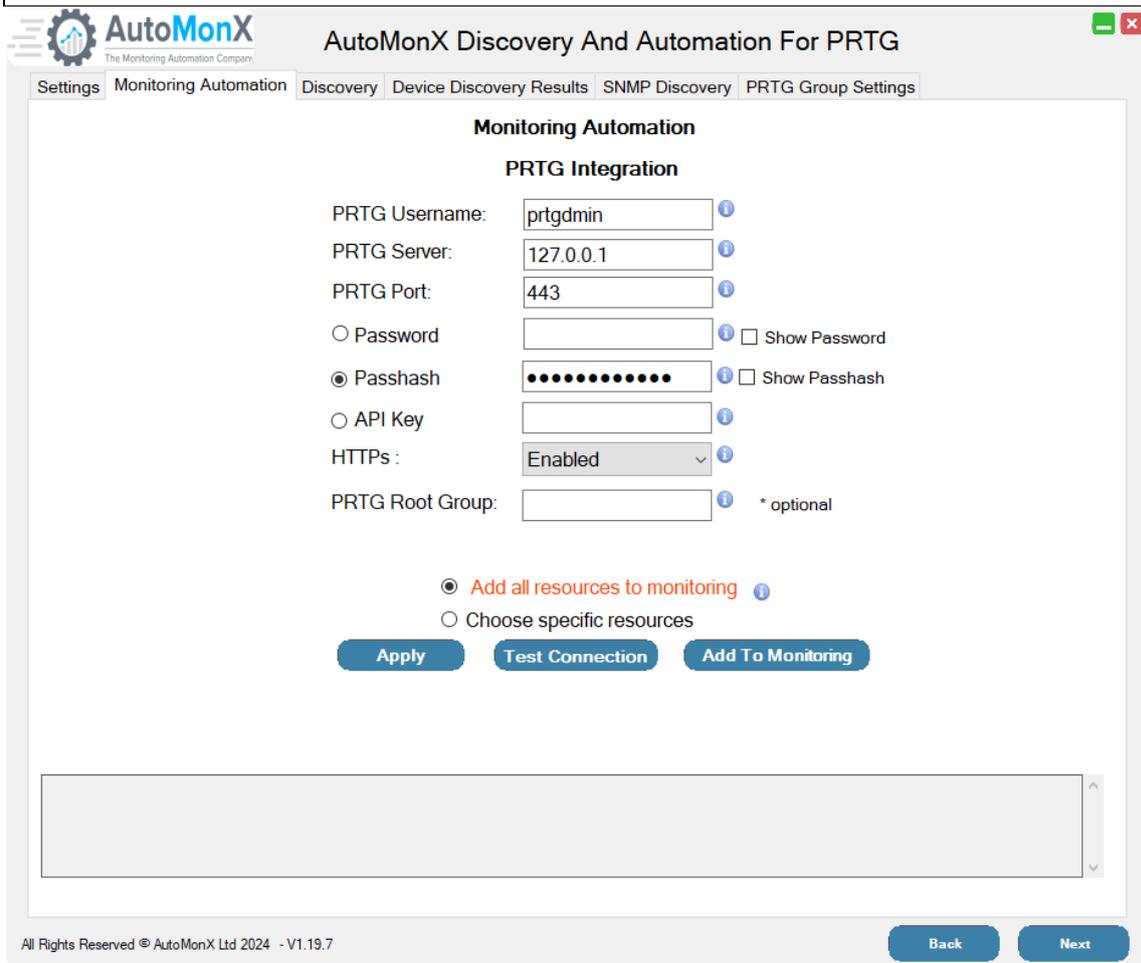


Pos	Sensor	Status	Message	Graph
1.	✓ Azure Service Health	Up	OK	Service Health Available
2.	✓ Azure App Metrics	Up	OK	Percentage C 4 %
3.	✓ Ping	Up	OK	Ping Time 150 msec
4.	✓ Uptime	Up	OK	System Uptime: 5 d 2 h 41 m
5.	✓ (002) Local Area Connection* Traffic	Up	OK	Traffic Total 0 kbit/s
6.	✓ CPU Load	Up	OK	Total 0 %
7.	✓ Disk Free: C:\ Label:Windows Serial N...	Up	OK	Free Space 82 %
8.	✓ Disk Free: D:\ Label:Temporary Storang...	Up	OK	Free Space 74 %
9.	✓ Memory: Physical Memory	Up	OK	Percent Availi 60 %
10.	✓ Memory: Virtual Memory	Up	OK	Percent Availi 60 %
11.	✓ Service Task Scheduler	Up	OK	Operating Sta Active

## 7.9 Automatically Adding Azure Sensors to PRTG

### Important:

- Fill-in your PRTG credentials and make sure that the PRTG Web interface connection details (username, password, server IP, port and if HTTPs was enabled) for this step to succeed
- When using Hosted PRTG password is not supported, only passhash.
- You need to **manually** create a target group in PRTG that will contain the Azure resources sensors before running “Add sensors to PRTG”. The default group that our Monitoring Automation is configured to use is AutoMonX\_Azure. You can create a group with a name of your choice and indicate it in the “PRTG Group” field.
- In case of **Multiple Tenants**, make sure to create groups with different names in PRTG. Discover each tenant from its respective PRTG Probe and make sure to point our UI to the relevant group (per the Azure tenant you have discovered)



The screenshot shows the 'AutoMonX Discovery And Automation For PRTG' window. The 'Monitoring Automation' section is active, and the 'PRTG Integration' sub-section is expanded. The following fields are visible:

- PRTG Username:
- PRTG Server:
- PRTG Port:
- Authentication:  Password,  Passhash,  API Key
- HTTPs:
- PRTG Root Group:

Below the fields, there are two radio button options:

- Add all resources to monitoring
- Choose specific resources

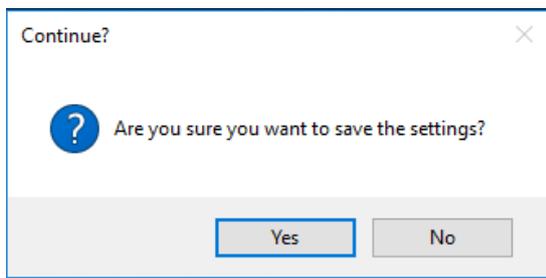
At the bottom of the form, there are three buttons: 'Apply', 'Test Connection', and 'Add To Monitoring'. The footer of the window contains the text 'All Rights Reserved © AutoMonX Ltd 2024 - V1.19.7' and 'Back' and 'Next' buttons.

Press “Apply” to save your settings.

You can also test the connection to PRTG to make sure everything is correct.



Press “Add to PRTG” to add the device and the sensors to PRTG. Confirm the group in PRTG that the Azure resources would be added to.



Allow the AutoMonX Monitoring Automation to add the resources and their sensors to PRTG. This could take several minutes depending on the size of the PRTG installation and the number of sensors to be added. When the process has successfully completed you can close the UI.

## 7.10 Resuming Adding Sensors in case of Timeouts

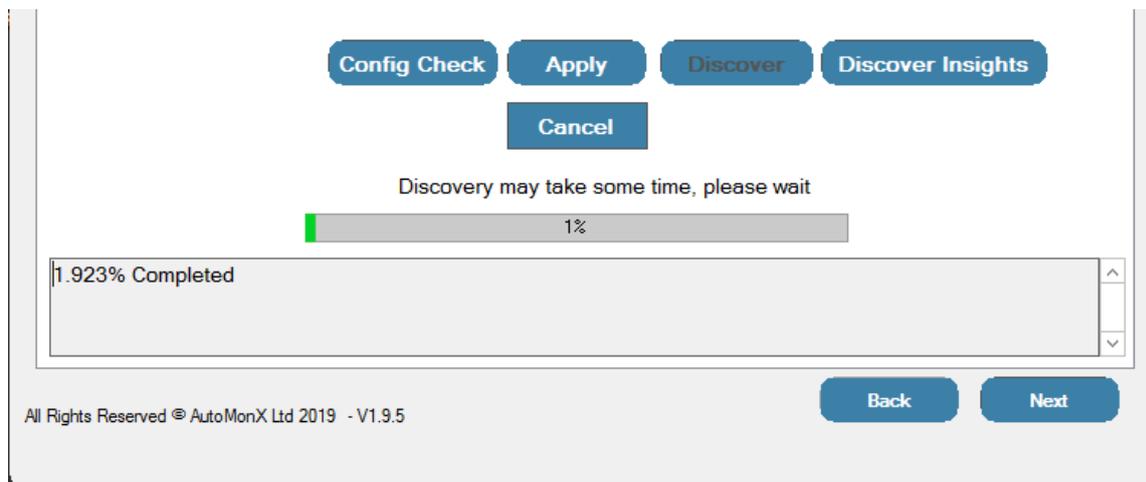
**Important:** The PRTG API sometimes fails to timely respond to the Monitoring Automation API calls. It may cause a timeout and the process of adding sensors may fail. In such cases, wait a few minutes and resume the addition of devices and sensors by pressing the “Add to PRTG” button. The Monitoring Automation will continue from the point it has stopped.

## 7.11 Automatic Discovery of Azure Application Insights

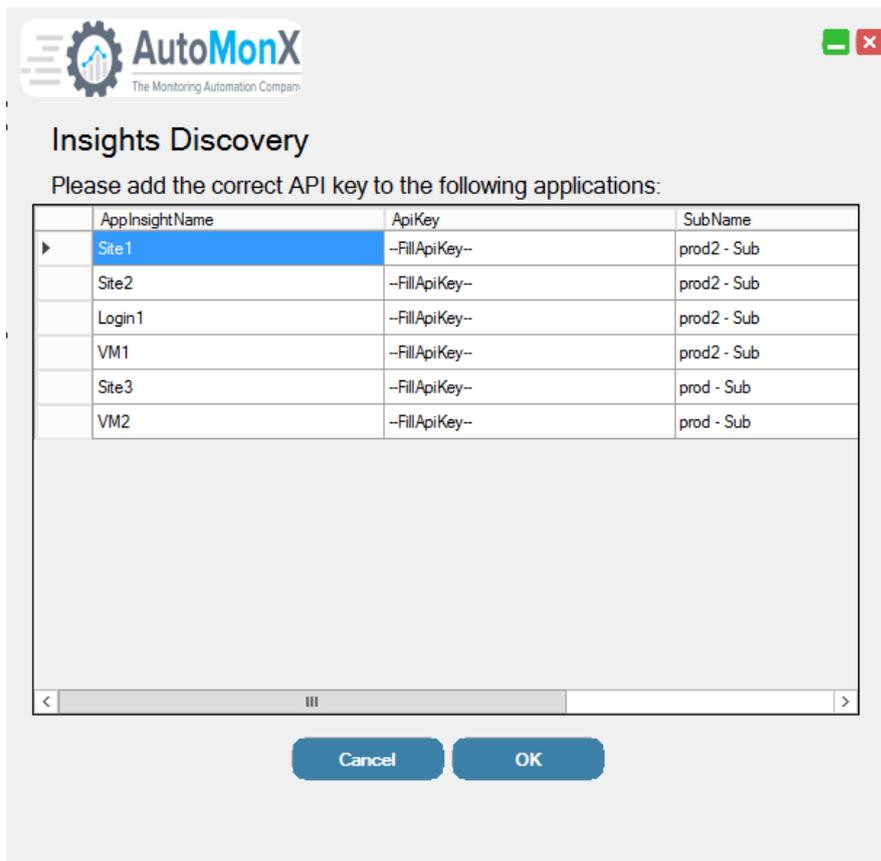
The AutoMonX Sensor Pack for Azure has the capability to monitor the Azure Application Insights in your environment. The Application Insights sensor is used to monitor Azure resources that perform requests. Request information is saved in Azure Insights. The Sensor uses the insight identification and key to monitor the Insights.

The process is very similar to adding other Azure resources to monitoring through PRTG.

After you discover the resources in your environment using the Auto Discovery, another button would appear in the buttons line “Discover Insights” as seen below:



Clicking the “Discover Insights” button will open a window showing all the discovered insights in your environment.



**Insights Discovery**

Please add the correct API key to the following applications:

AppInsightName	ApiKey	SubName
Site 1	--FillApiKey--	prod2 - Sub
Site2	--FillApiKey--	prod2 - Sub
Login1	--FillApiKey--	prod2 - Sub
VM1	--FillApiKey--	prod2 - Sub
Site3	--FillApiKey--	prod - Sub
VM2	--FillApiKey--	prod - Sub

Cancel OK

You will need to fill the API keys according to each Application Insight you want to monitor. Create an API key for Application Insights via the Application Insight page in your Azure Portal environment as seen below.

**Important:** Make sure to grant the API Key permissions to Read Telemetry information.

Microsoft Azure Search resources, services, and docs (G+)

Home > - API Access

**- API Access**  
Application Insights

Search (Ctrl+/) << 2 + Create API key Delete API key Help

- Funnels
- User Flows
- Retention
- Impact
- Cohorts
- More

**Configure**

- Properties
- Smart Detection settings
- Usage and estimated costs
- Continuous export
- Performance Testing
- API Access 1**
- Work Items

Application ID

API key description	Last Used	Created On
You haven't set up any API keys. Click 'Create API key' to get started.		

Microsoft Azure Search resources, services, and docs

Home > faf3aga - API Access > Create API key

**Create API key**

Create an API key to read Application Insights data.

API keys are used by applications outside the browser to access this resource. Your API keys should be managed like passwords. Keep them secret.

Provide a description to help you identify this API key in the future.

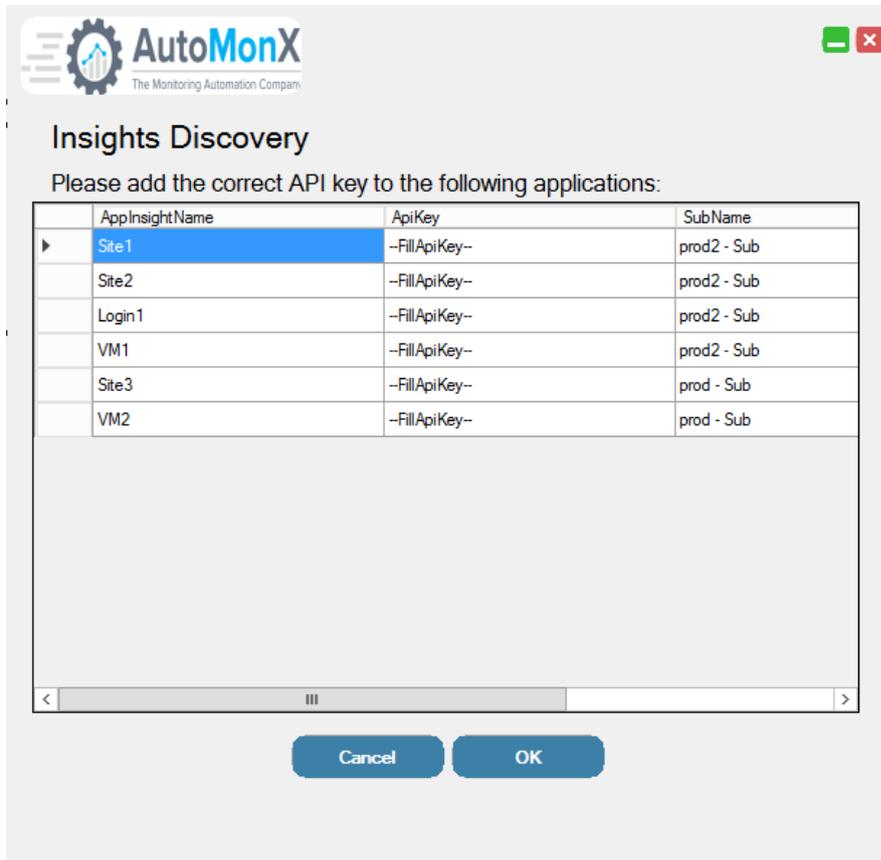
AutoMon Insgihts Access ✓

Choose what this API key will allow apps to do:

- Read telemetry
- Write annotations
- Authenticate SDK control channel

**Generate key**

Place the newly generated key in the API Key cell for each Application Insight you want to monitor.



When you finish, click the OK button and let the Application Insights discovery to run. The next steps are similar to the addition of any other sensor to PRTG as described in section [6.3 Selecting the Azure sensors for monitoring](#)

## 7.12 Using CLI to Discover Azure Resources

The AutoMonX Sensor Pack for Microsoft Azure needs to scan the Azure environment for any resources that it can monitor. In the discovery mode, the Sensor Pack will generate a list of all the Azure resources in your environment that it can monitor and provide you with the required sensor configuration to monitor these resources. There are several discovery modes of Azure resources:

1. Thorough Discovery (Resources and Metrics)
2. Quick Discovery of Azure Resources only
3. Quick Discovery of Azure Resource Metrics
4. Quick Discovery with Inclusion/Exclusion

### 7.13 Azure Sensor Pack Discovery Modes explained

Since there are multiple discovery options, below is a table that compares the discovery functionality and their benefits:

Discovery Mode	Functionality	Pros and Cons
Thorough	Default discovery mode available in our UI. Discovers all the subscriptions, resources and metrics	<p>Easy to use in a UI-interactive mode.</p> <p>Longer time to see full results.</p> <p>Great for the initial discovery of your Azure estate and all their metrics. Push all the metrics in two simple steps to monitoring</p>
Quick – Resources only	Quick discovery of Resources only (no metrics), available in CLI only	Great for initial discovery of your Azure estate without spending time on metrics discovery
Quick – Metrics only	Quick discovery of the metrics of Resources	<p>Great to focus on specific resources only and get their metrics ready to be pushed to monitoring.</p> <p>Uses the Azure resources discovered in a previous step and discovers only their metrics.</p>
Quick – New resources only	Quick discovery of new resources and their metrics	Great for when you have a large environment and want to discover and add only new resources to monitor
Quick – Discovery with Inclusion or exclusion of specific resource types	Quick discovery using the exclude and include configuration files. Can be used with any of the above options.	<p>Great for even quicker discovery when you have a good grasp of what you want to monitor.</p> <p>Might exclude resources you do wish to monitor.</p>

## 7.14 Azure Resources Discovery – CLI Options for thorough discovery

Below are some examples for running Auto discovery of Azure resources using the CLI.

- Discover all subscriptions and resources (single Tenant)

```
AutoMonX_AzureCollector.exe -discovery -sub -All
```

- Discover resources of a specific subscription (single Tenant)

```
AutoMonX_AzureCollector.exe -discovery -sub <Subscription Name>
```

- Discover a specific Azure Tenant in a Multi-tenant environment. The tenant number is configured in the file “AzureConnProfiles.ini”. For the first tenant configured in the file “AutoMonX\_AzureSensor.ini” use 0 or run discovery without the tenant option as in the previous examples.

```
AutoMonX_AzureCollector.exe -discovery -sub -All -tenant  
<tenant_number>
```

## 7.15 Quick Discovery CLI Options

Depending on the size of your Azure environment, the full discovery process can take a long time. For shorter discovery (50% or more in time savings) use the following CLI options.

- **Azure Resources-only discovery** (no metrics, quicker results):

```
AutoMonX_AzureCollector.exe -discovery -sub -All -resources
```

- **Azure Metrics-only discovery** on supported resources.

**Important:** For first time discovery, use this option only after running the resource discovery or after a thorough discovery was completed

```
AutoMonX_AzureCollector.exe -discovery -sub -All -metrics
```

Running a discovery without the options resources and metrics will result in a thorough discovery.

- **Azure new resources only discovery** (only resources that weren't discovered in a previous discovery):

**Important:** For first time discovery, use this option only after running the metrics discovery or after a thorough discovery was completed

*AutoMonX\_AzureCollector.exe -discovery -sub -All -new\_resources*

- **Include/Exclude discovery**

Using the option below would substantially narrow down your discovery time only for specific resource types or resources with a predefined Azure Tag. It instructs the discovery process to consult two configuration files: include\_mon.csv and exclude\_mon.csv. See [this section](#) for more details.

*AutoMonX\_AzureCollector.exe -discovery -sub -All -whitelist*

## 7.16 Azure Resources Discovery Report

Running the discovery commands generates a report that contains the following information:

- Subscription names.
- List of Microsoft Azure resources for each subscription.
- Command line parameters for the AutoMonX Azure PRTG Sensor pack.
- List of unsupported resource types.
- List of Azure resources that don't have their monitoring metrics enabled.

The commands to monitor the supported resources will be found in **<Sub Name>\_AutoMonX\_Azure\_Report.html** in the PRTG EXEXML program folder under the Azure folder.

**Note:** The PRTG EXEXML program folder is usually located in the path:  
<Drive>: \Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML  
Or:

<Drive>: \Program Files\PRTG Network Monitor\Custom Sensors\EXEXML

If the -sub -all parameters were used, the report name will be set to ALL\_AutoMonX\_Azure\_Report.html.

Below is a sample report:

**PRTG Sensor pack for Microsoft Azure  
by Automonx - Discovery Results**

**Subscription microsoft azure sponsorship**

**Discovered Resources:**

Resource Name	Sensor Azure Type	Configuration Line
microsoft azure sponsorship_(Subscription)	Azure Billing	-sub "microsoft azure sponsorship" -cons
WindowsSRV2_(VirtualMachines)	Azure Service Health	-type Microsoft.Compute/virtualMachines_health -resgrp MS_Sponsored -res "WindowsSRV2" -sub "microsoft azure sponsorship"
WindowsSRV2_(VirtualMachines)	Azure App Metrics	-type Microsoft.Compute/virtualMachines -resgrp MS_Sponsored -res "WindowsSRV2" -sub "microsoft azure sponsorship"
linuxsrv167_(NetworkInterfaces)	Azure App Metrics	-type Microsoft.Network/networkInterfaces -resgrp MS_Sponsored -res "linuxsrv167" -sub "microsoft azure sponsorship"
WindowsSRV2_disk1_d5a2d7a8cc3a4c3bb09c732a6175847e_(Disk)	Azure App Metrics	-type Microsoft.Compute/disks -resgrp MS_SPONSORED -res "WindowsSRV2_disk1_d5a2d7a8cc3a4c3bb09c732a6175847e" -sul

## 7.17 Azure Delta Discovery Report

For customers with a constantly expanding Azure environment, who want to track the newly added resources, a Delta discovery report was added. The delta reports shows only the new resources added during last discovery.

<Sub Name>\_AutoMonX\_Azure\_Discovery\_Delta.html

### Azure Auto discovery - New Results

**Tenant:** My Tenant (5d3)

**Discovery started at:** 22 January 2023 16:12

Subscription	New resources	Total resources
microsoft partner network	1	21

**Discovery Type:** Quick, Metrics

**Discovery completed at:** 22 January 2023 16:33

**Total Running time:** 0 hours, 20 min

Resource Name
---------------

## 7.18 Azure Channel Limits Report

This report is created automatically based on your sensors and shows the out of the box limits configured for different types of resources. The purpose of this report is to give you a clear picture of the thresholds the AutoMonX sensors have configured.

*AutoMonX\Azure\Logs\Channellimits.csv*

## 7.19 Monitoring Automation Files

The Monitoring Automation files are created in the Azure Sensor pack settings folder and contain the commands that allows adding the discovered Azure resources to PRTG.

After a successful discovery, a file will be generated in the Data folder for each Azure subscription. The files are named according to the following format:  
<tenant\_internal\_id>\_<subscription name>DiscoveryData.csv

## 7.20 Using the Monitoring Automation CLI

The Azure Sensor Pack contains a command line interface that automates the addition of Azure resources as sensors to the PRTG system.

To use the automation CLI, first you must edit the file below that is in the Azure\Common folder:

AutoMonX\_PRTG\_Automation.INI

**PRTG\_USER=<prtg\_administrative\_user>**

### **PRTG\_SERVER=<prtg\_server\_name>**

The PRTG user must have read and write permissions to operate the program. A target group must be created in PRTG that will contain the Azure resources sensors.

Below is an example of how to use the Monitoring Automation CLI:

```
AutoMonX_PRTG_Automation.exe -file <DiscoveryData>.csv -p <passhash> -group <target_group>
```

Instead of passing the passhash in the command line, it can be stored inside the file AutoMonX\_PRTG\_Automation.INI by editing the value of PRTG\_PASSHASH=<passhash>. Another option is to use the prtg api key and store it in the file, PRTG\_API\_KEY=<api\_key>.

## **7.21 Adding the Azure Resource URL to PRTG Comments**

Adds the full URL to access the Azure resource in the Azure Portal. The URL is added automatically to the device comments tab in PRTG. This feature can help monitoring teams to rapidly access the relevant resource in the Azure Portal.

To activate this feature, add the option resource\_id to the discovery via CLI. Any newly created sensors will show the full resource URL in the device comments:

```
AutoMonX_AzureCollector.exe -discovery -sub -All -resource_id
```



## **7.22 Multi-tenant Discovery Scheduler**

For automatic discovery scheduling especially in large multi-tenant environments, see the following scheduling function.

Configuration is done in the file “AutoMonx\Azure\Scheduler\Scheduler.csv”.

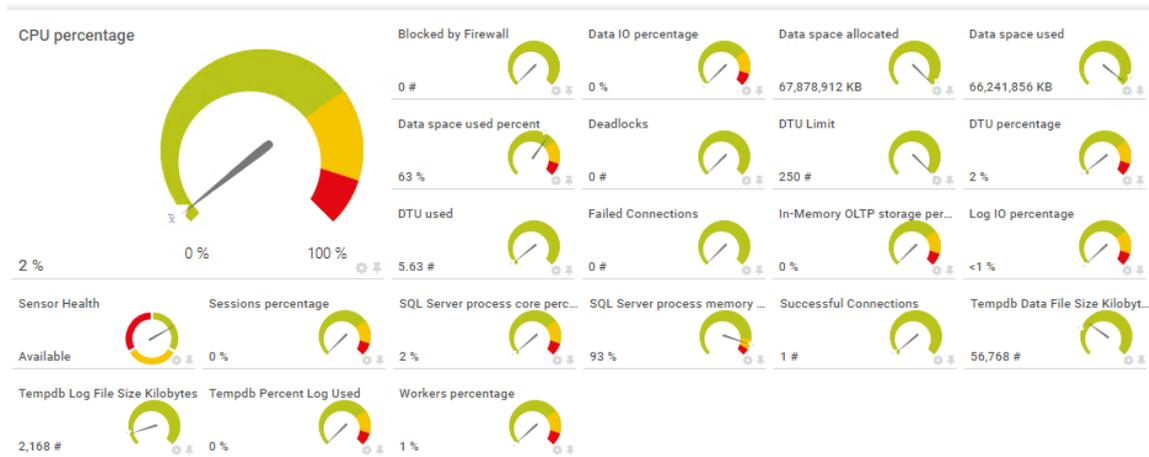
You can also find an example file in the same directory. Adjust the “Time To Run” (Must be in the format Mon HH:mm), and the tenant id you want to run the discovery on. Please refer to the [API guide](#) for explanation about the API Params.



## 8 Supported sensor types

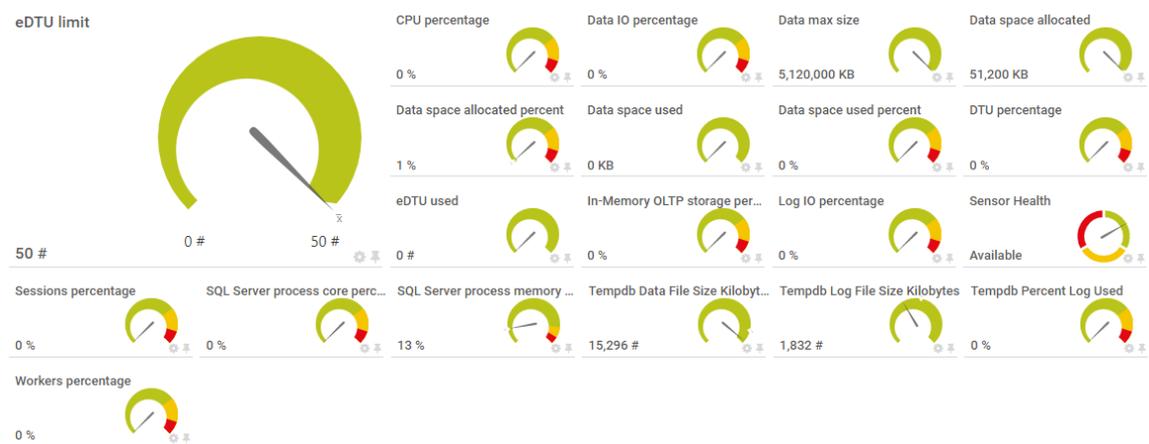
### 8.1 SQL Database

The Azure SQL Database resource is a fully managed relational cloud database. Its Metrics measure the database's health and performance.



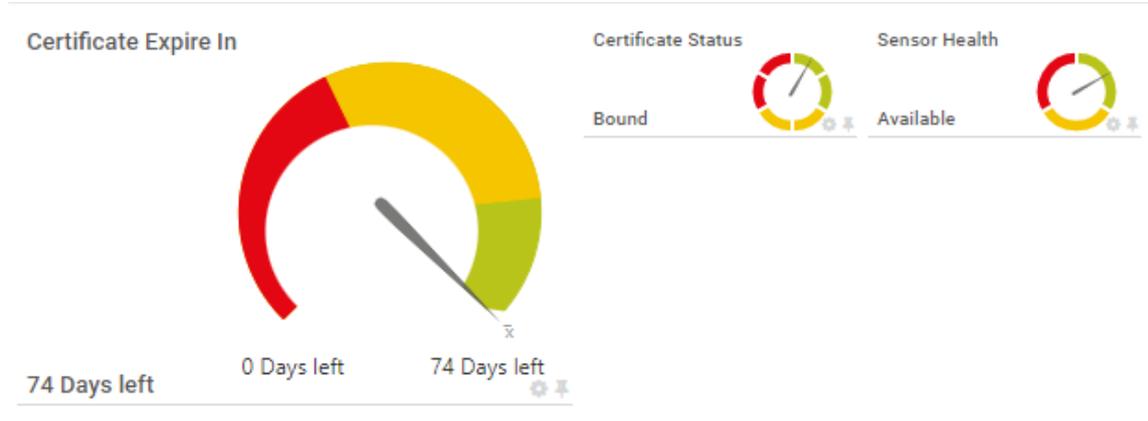
### 8.2 SQL Elastic Pools

The SQL Elastic Pool is a simple, cost-effective solution for managing and scaling multiple databases that have varying and unpredictable usage demands. The databases in an elastic pool are on a single server and share a set number of resources at a set price. The SQL Elastic pool sensor is used to monitor such pools.



### 8.3 Certificate

The Azure Certificate resource verifies a secure connection to an internet resource. Its Metric measure its expiration date and host name binding status.



### 8.4 App Service Plan (Server Farms)

An App Service Plan consists of the underlying virtual machines that will host the Azure App Services. The App Service Plan defines the region of the physical server where your app will be hosted on and the amount of storage, RAM, and CPU the physical servers will have. Below is an example of such service (i.e. Server Farm)



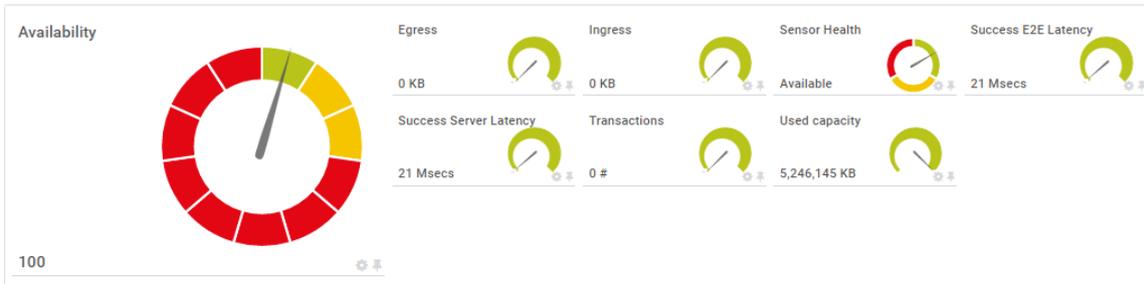
## 8.5 Web Sites

The Azure Web Apps resource is a scalable web application. Its Metrics measure the Web-App's health and performance.



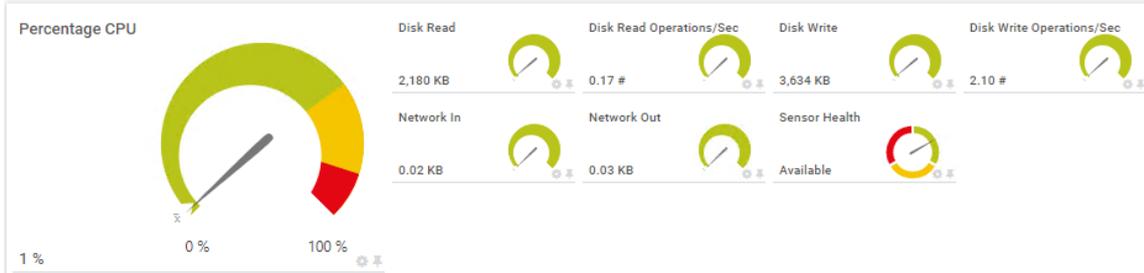
## 8.6 Storage Accounts

The Azure Storage Account resource is a massively scalable object storage for unstructured data. Its Metrics measure the storage's health and performance.



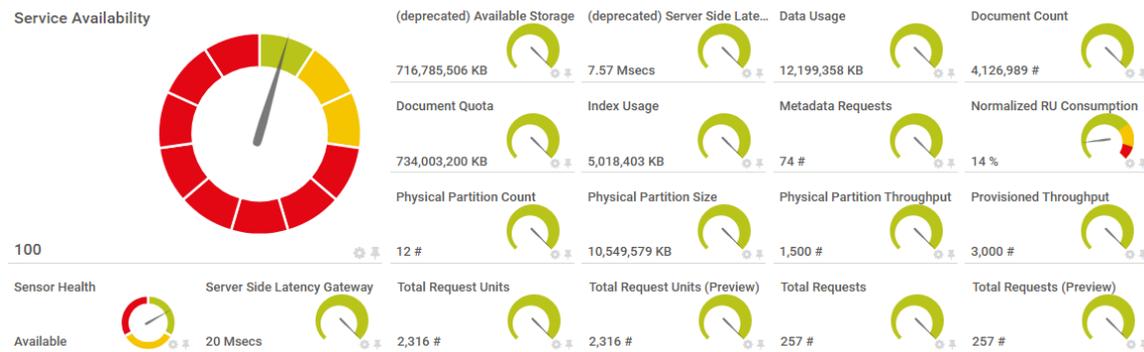
## 8.7 Cloud Services

Azure Cloud Services is an example of a platform as a service (PaaS). This technology is designed to support applications that are scalable, reliable, and inexpensive to operate.



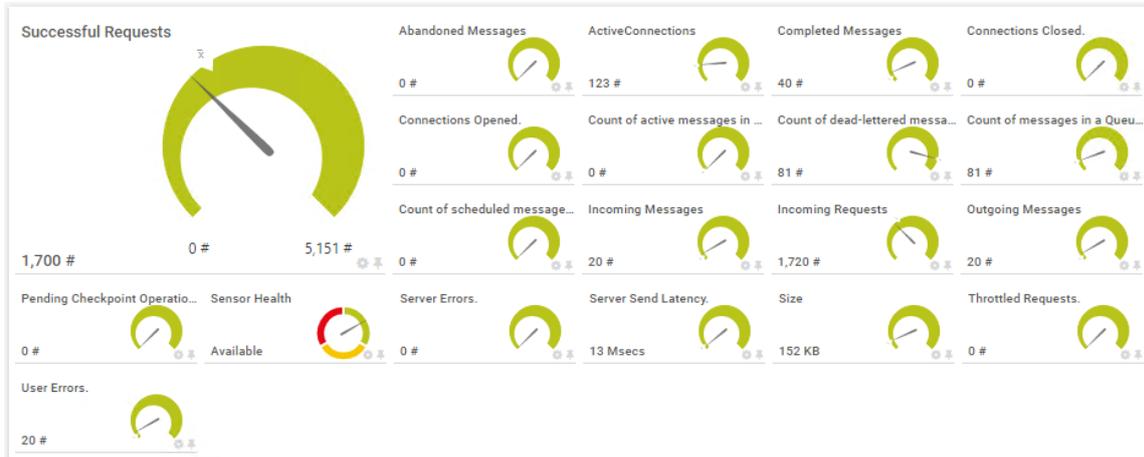
## 8.8 Database Accounts

The Azure Database account resource manages scalability of databases. Its Metrics measure the account's health and the databases requests.



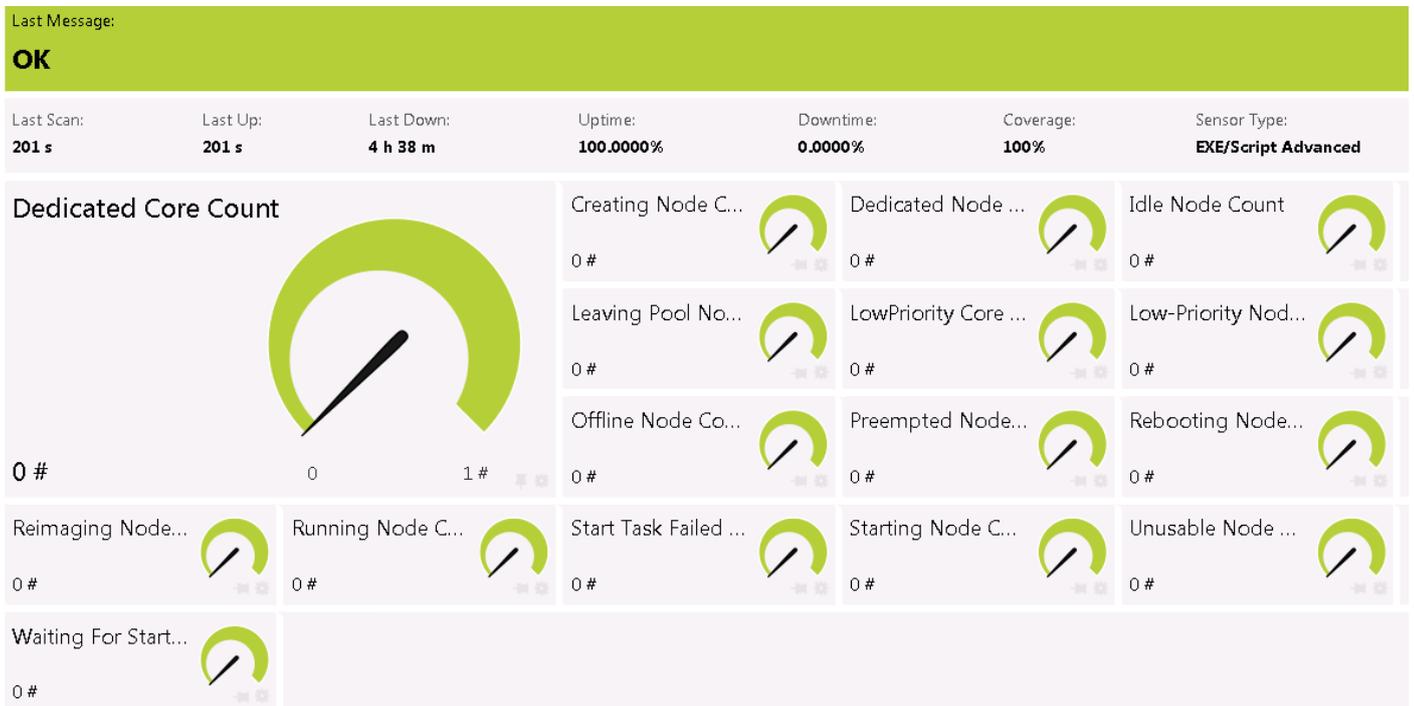
## 8.9 Service Bus Namespace

The Azure Service Namespace resource manages messaging communication between applications using a bus. Its Metrics measure the Namespace's health and performance.



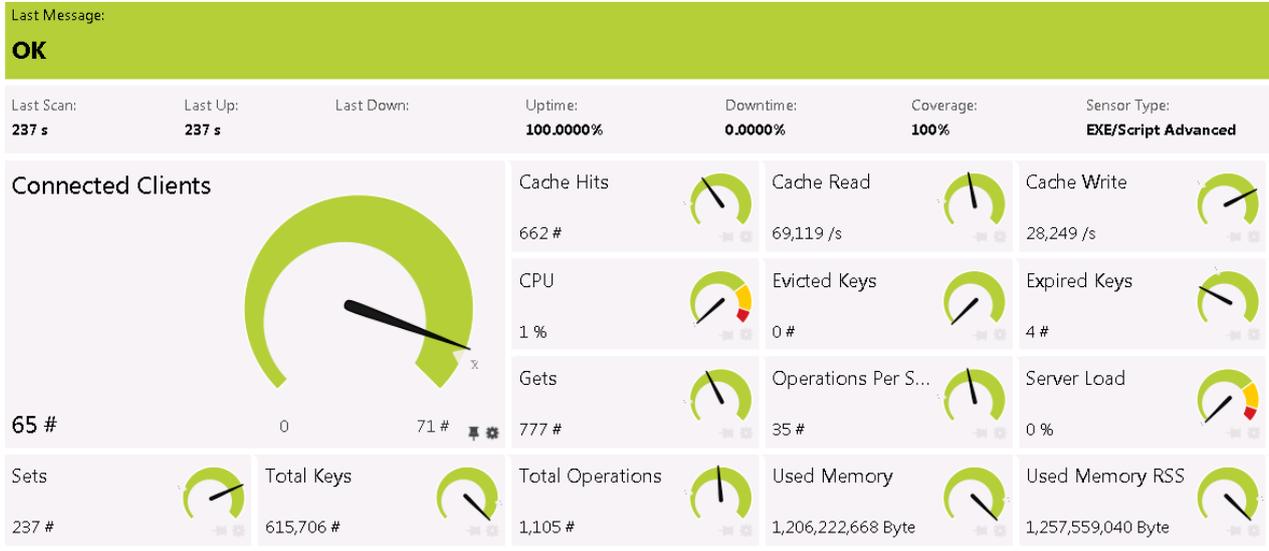
## 8.10 Batch Accounts

The Azure Batch Account resource manages job tasks. Its Metrics measure the Batch Manager's health and performance.



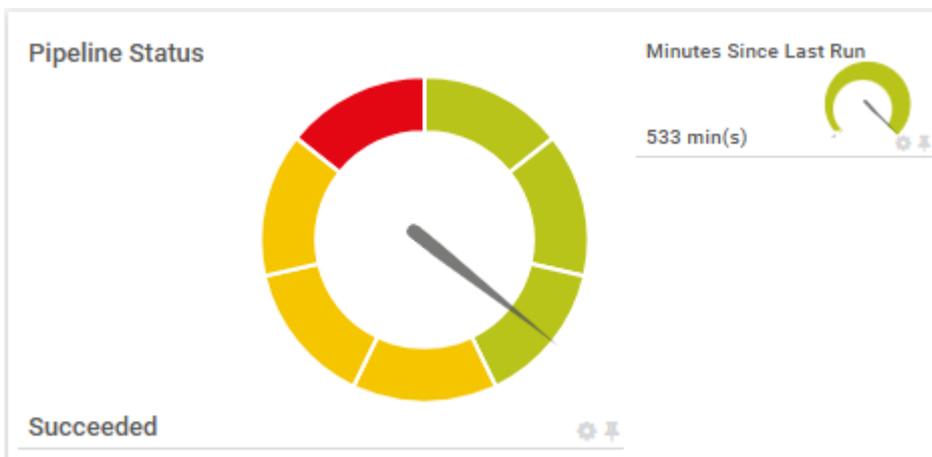
### 8.11 Redis

The Azure Redis Database resource is a managed key/value database. Its Metrics measure the database's health and performance.



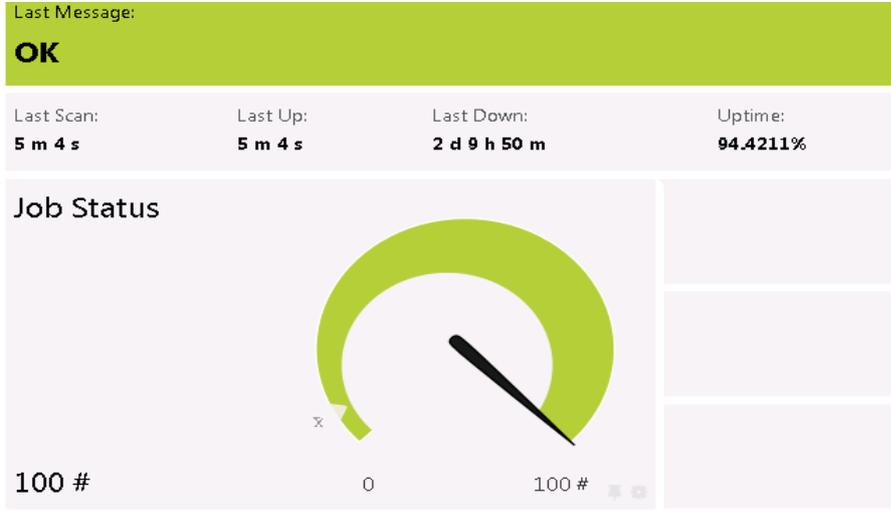
### 8.12 Data Factory

The Azure Data Factory resource performs analytical operations on data from various data sources using pipelines. Its Metric measure the pipeline's last run status and how many minutes passes since the last run.



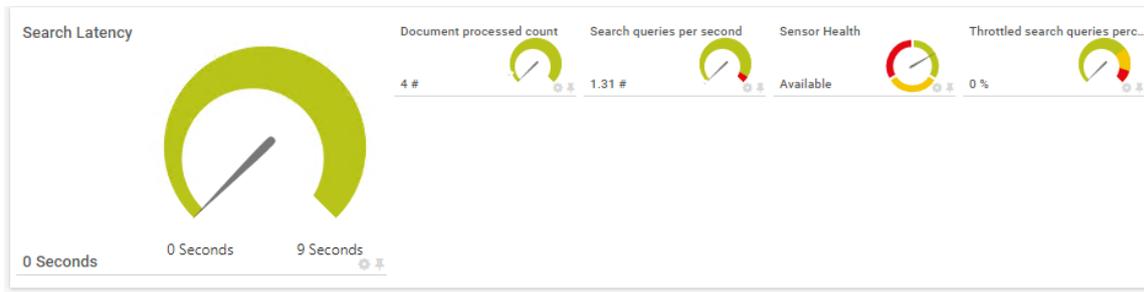
### 8.13 Scheduler (deprecated)

The Azure Scheduler resource manages scheduling of job tasks. Its Metrics measure the task's Health Status



### 8.14 Search Services

The Azure Search Services resource manages search services performed on applications. Its Metrics measure the search service's health and operations



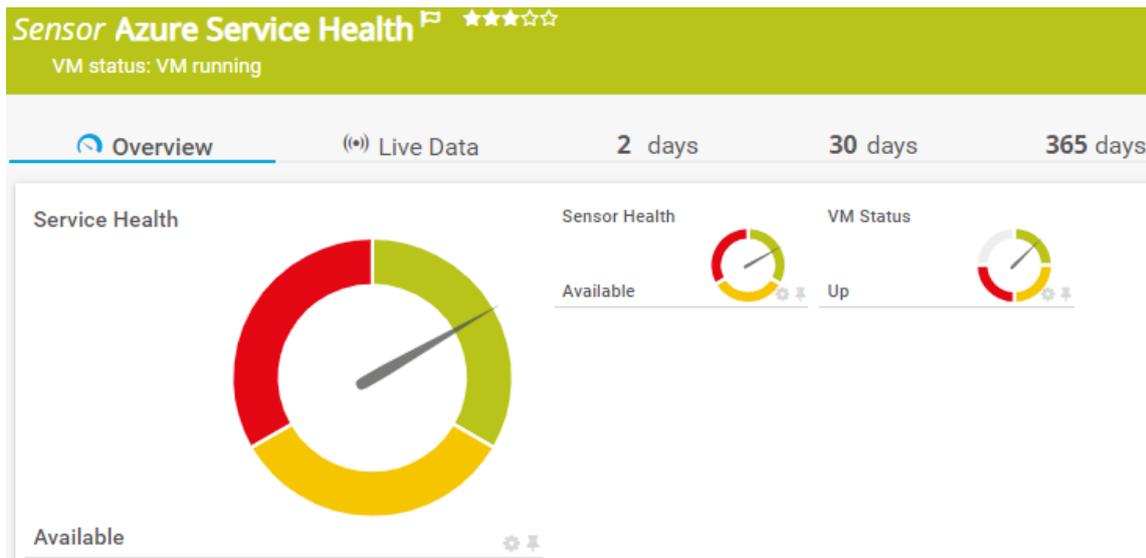
### 8.15 Service Bus Queues

Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics (in a namespace). Service Bus is used to decouple applications and services from each other. Azure Service Bus supports reliable message queuing and durable publish/subscribe messaging.



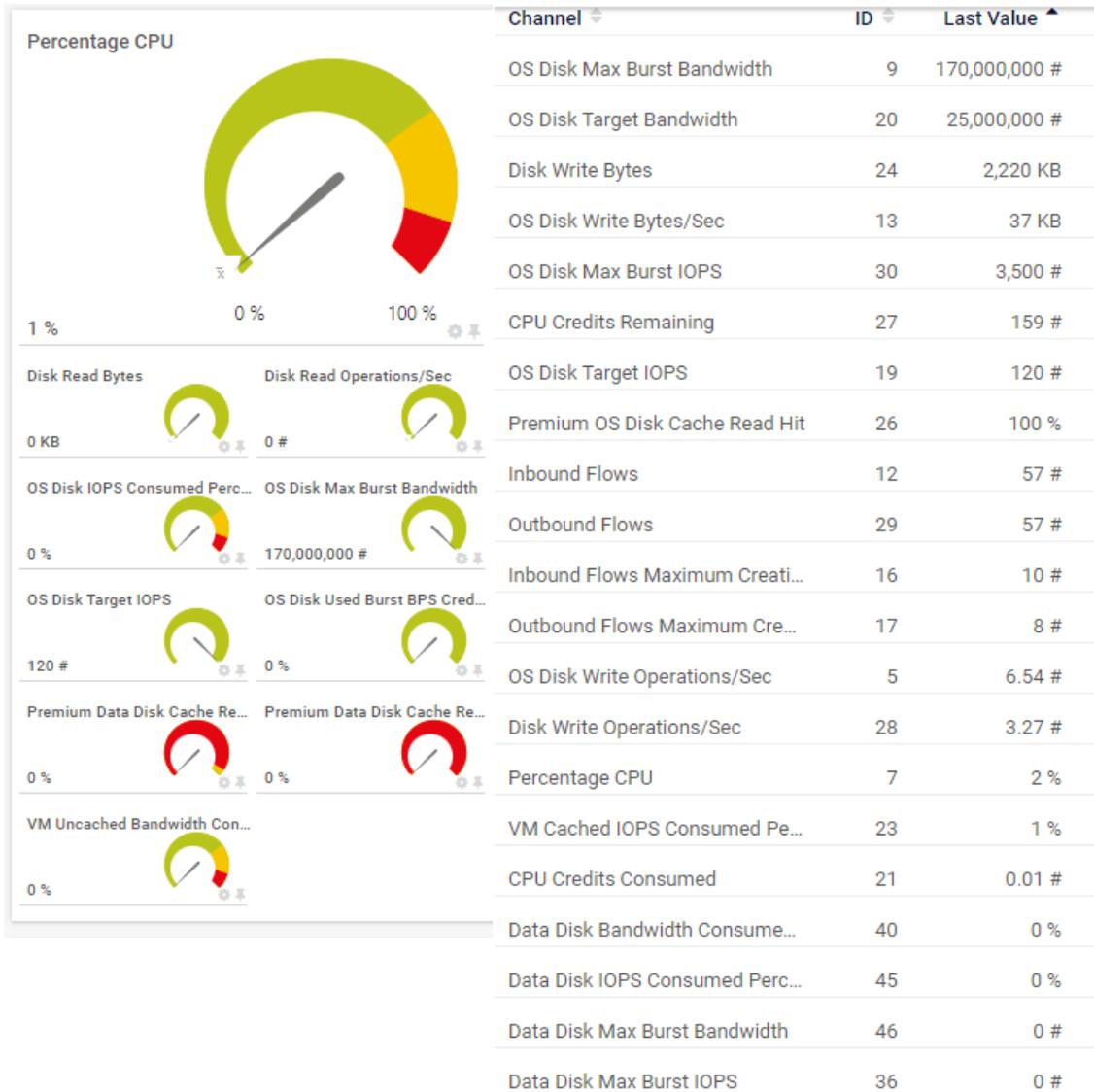
### 8.16 Service Health

The Azure Service Health sensor checks the current service health status of a resource. An error status will show as red with the provided error. Unknown statuses are shown as Warning. For Virtual Machines only, an additional channel VM Status will show with a text description.



## 8.17 Virtual Machines

The Azure Virtual Machine sensor provides performance information regarding virtual machines hosted in Azure. An example is seen below. Additionally, our monitoring automation can invoke PRTG-native SNMP sensors that can also be automatically discovered and added under the same Virtual Machine device.



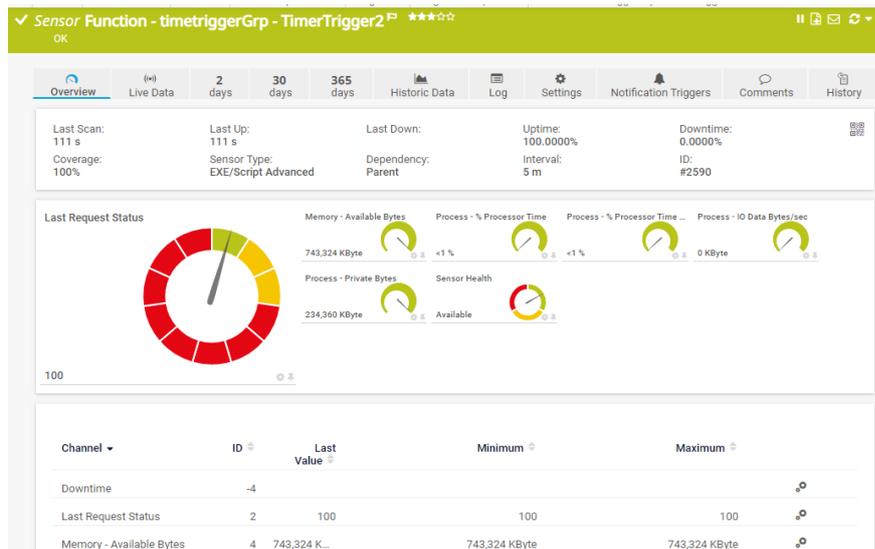
## 8.18 Azure Billing

The Azure Billing sensor provides information on the daily cost of a subscription, and the accumulated cost from the beginning of the month. Billing will be displayed in the currency relevant to your subscription and covers only the resources that the Azure API provides billing information about them.



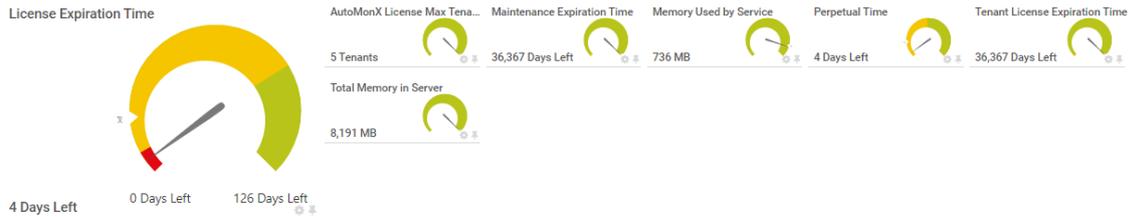
## 8.19 Azure Application Insights

Application Insights, a feature of [Azure Monitor](#), is an extensible Application Performance Management (APM) service for developers and DevOps professionals. Use it to monitor your live applications. The AutoMonX Sensor Pack for Azure can help you to Auto-discover and monitor Azure Insights.



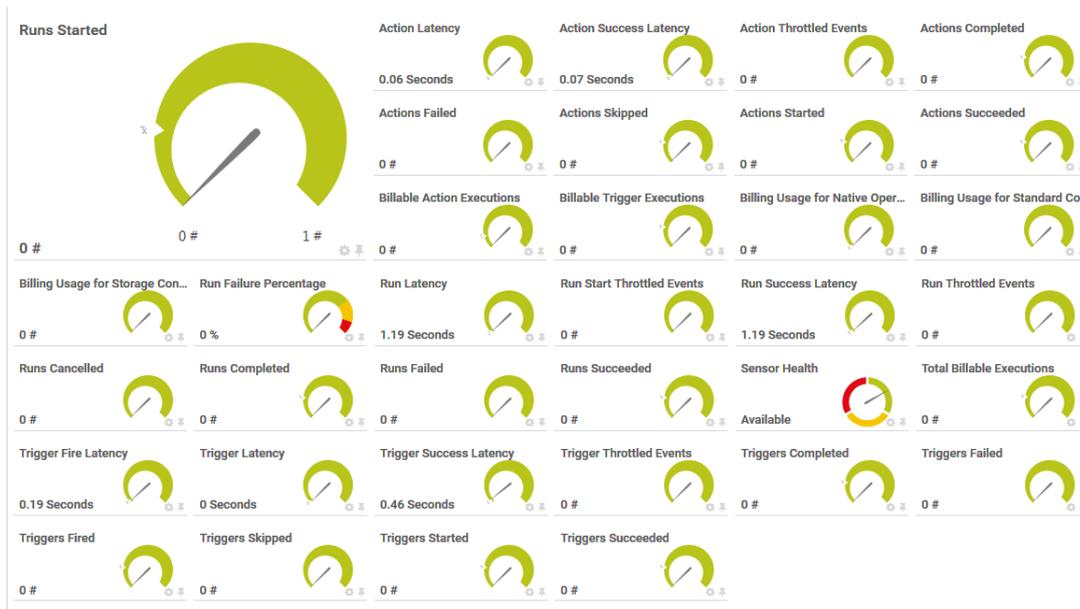
## 8.20 AutoMonX License

The License sensor is an additional sensor type aimed at monitoring the AutoMonX licensing and maintenance status. This sensor will let you know if your license or maintenance period is about to expire and helps you to renew it on time. This will help you to enjoy continues monitoring and prepare for renewal, without the license ending unexpectedly. It also includes the memory usage of AutoMonX.



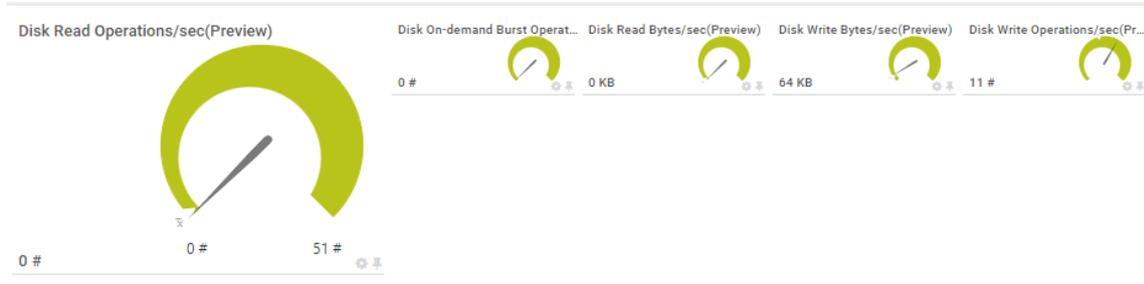
## 8.21 Azure Logic App

Azure Logic App is a cloud service that helps you schedule, automate, and orchestrate tasks, business processes, and workflows when you need to integrate apps, data, systems, and services across enterprises or organizations. Below is a screenshot of typical metrics you will get in PRTG:



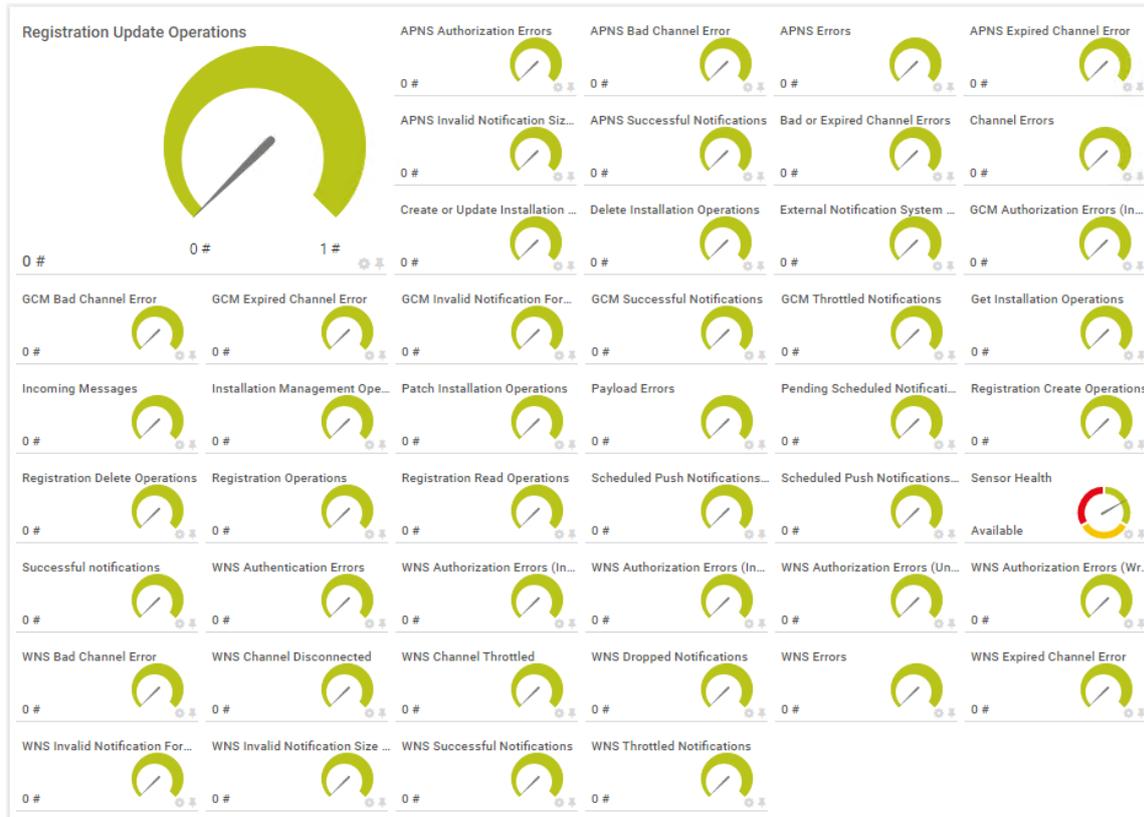
## 8.22 Azure Disks

Azure makes available several disk types for attaching them to virtual machines for usage such as OS or Data disks. The sensor below provides the metrics available for such disks:



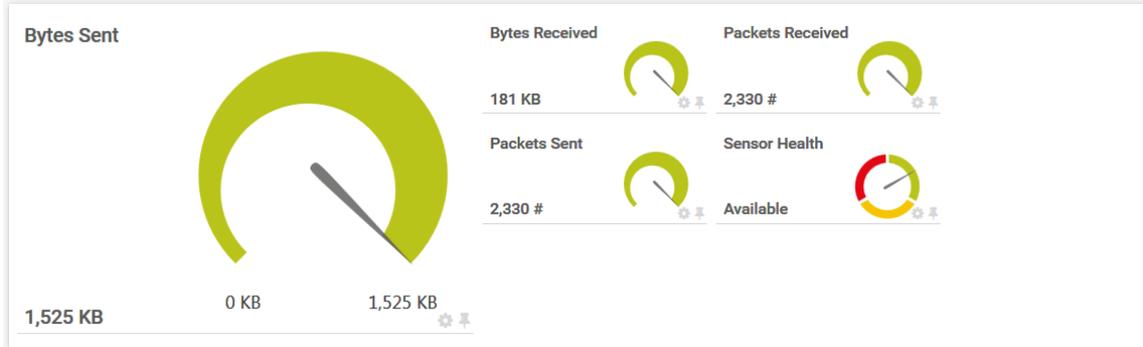
## 8.23 Azure Notification Hubs

Azure Notification Hubs is a massively scalable mobile push notification engine for quickly sending millions of notifications to iOS, Android, Windows, or Kindle devices, working with APNs (Apple Push Notification service), GCM (Google Cloud Messaging), WNS (Windows Push Notification Service), MPNS (Microsoft Push Notification Service), and more.



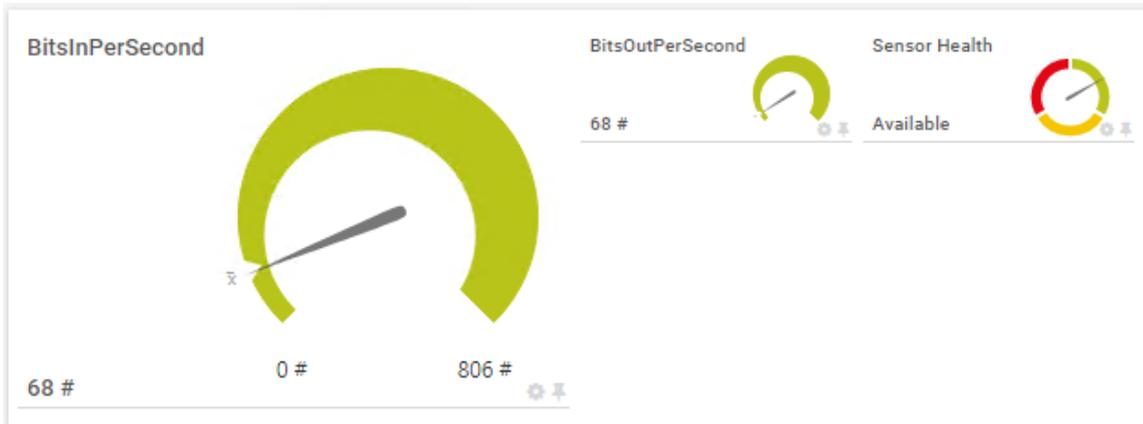
### 8.24 Azure Network Interfaces

A network interface enables an Azure Virtual Machine to communicate with internet, Azure, and on-premises resources.



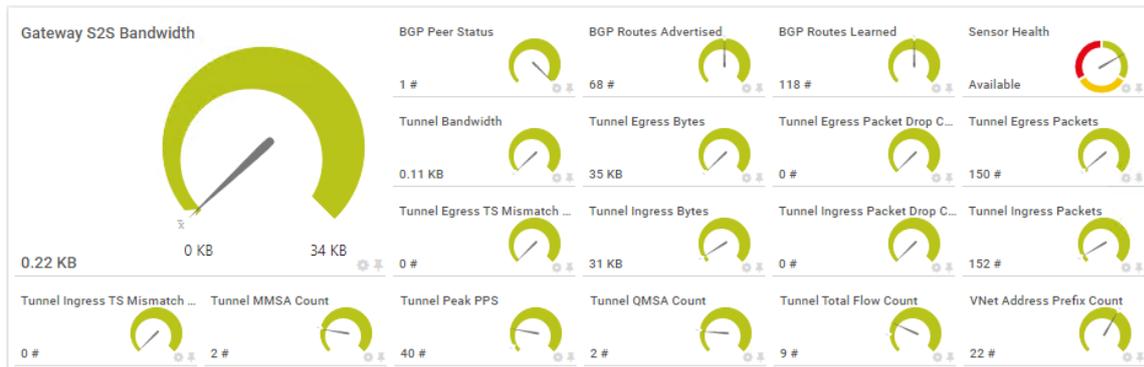
### 8.25 Azure ExpressRoute Connections

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider.



## 8.26 Azure VPN Gateways Metrics

A VPN gateway is a specific type of virtual network gateway that is used to send encrypted traffic between an Azure virtual network and an on-premises location over the public Internet. You can also use a VPN gateway to send encrypted traffic between Azure virtual networks over the Microsoft network. Each virtual network can have only one VPN gateway. However, you can create multiple connections to the same VPN gateway. When you create multiple connections to the same VPN gateway, all VPN tunnels share the available gateway bandwidth. The Azure Sensor Pack can monitor various VPN Connection types such as: S2S (Site to Site), P2S (Point to Site) and M2S (Multi-site) connections.

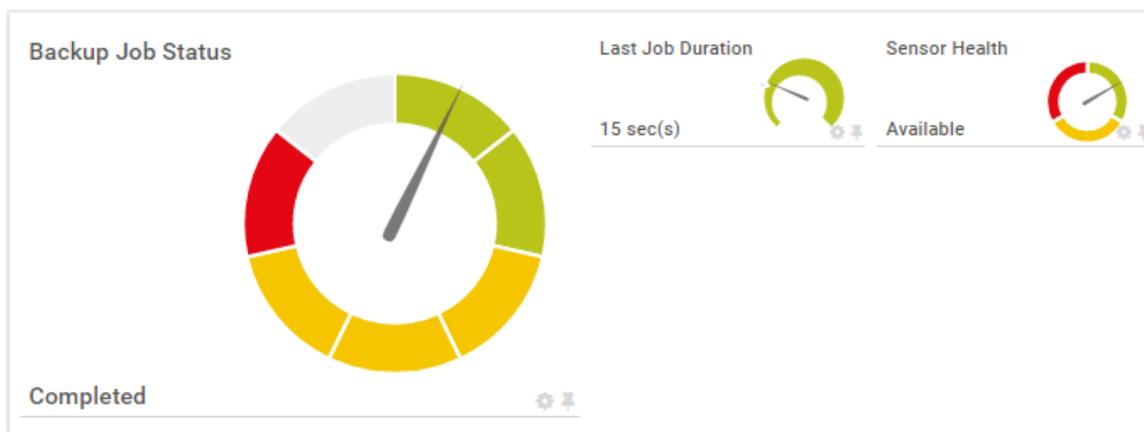


## 8.27 Azure Backup Jobs

The Backup Job sensor monitors the status of the backup jobs under every Recovery Service Vault. By default, the AutoMonX Sensor Pack for Azure supports the daily backup. If you have a weekly backup configured, please add the following configuration in the file "AutoMonX\_AzureSensor.ini":

BACKUP\_JOB\_VAULT\_<vault\_name>\_HOUR=168

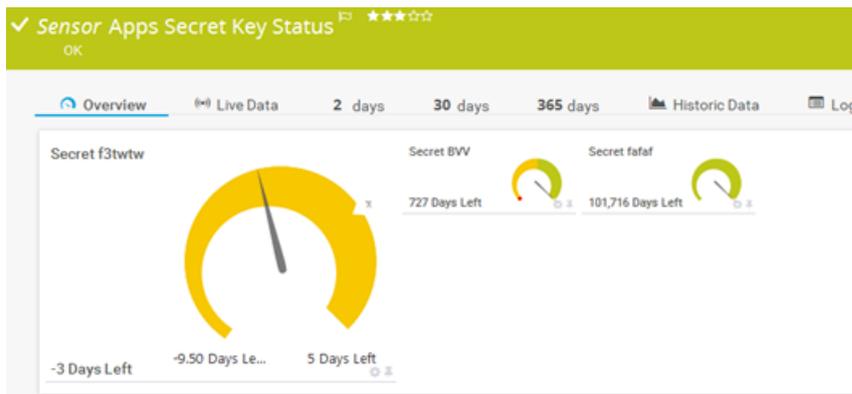
And restart the service for the changes to take effect.



## 8.28 Azure App Secret Key and/or Certificate Status

The Azure Apps Secret Key is used for authenticating with the Azure API for a specific Azure App (App Registrations). Each secret key has an expiry date. After the secret key has expired it can no longer be used. The Azure App Secret Key sensor displays the remaining time until its expiry date. The Certificates for the App will also appear as a channel in this sensor

To make the Azure Secret Keys discoverable and available for monitoring, you need to follow the procedure in [Chapter 14.1](#).



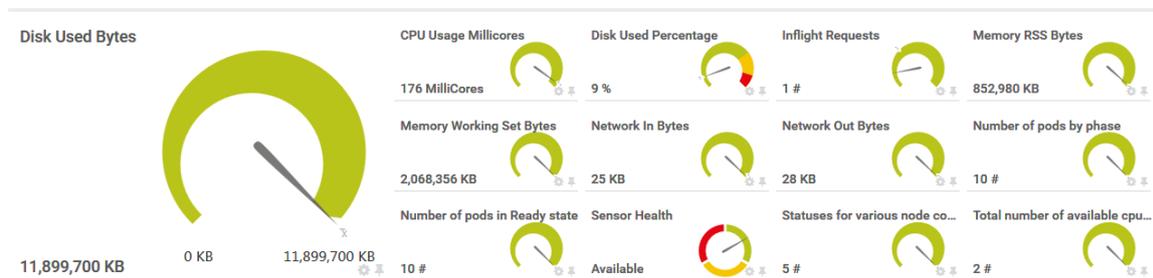
### 8.29 Azure Kubernetes Services (AKS)

Starting with release 4.0.15, The Azure Sensor Pack can discover and monitor the Azure Kubernetes Services (AKS). If the sensors are not discovered, enable the insights monitors (follow [this guide](#)). Four new sensor types were developed to support AKS:

- Azure Kubernetes Cluster Metrics
- Azure Kubernetes Deployments
- Azure Kubernetes Internal Deployments

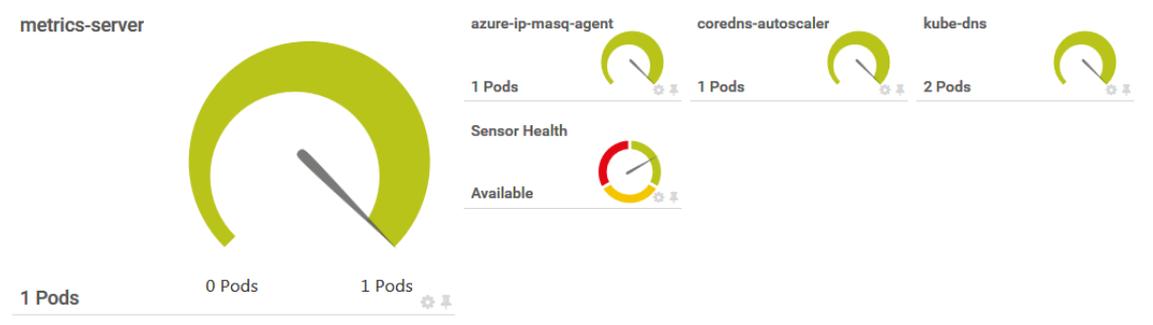
### 8.30 AKS Cluster Metrics

The Azure Kubernetes Metrics sensor shows the general resource consumption of the entire AKS cluster (average usage figures)



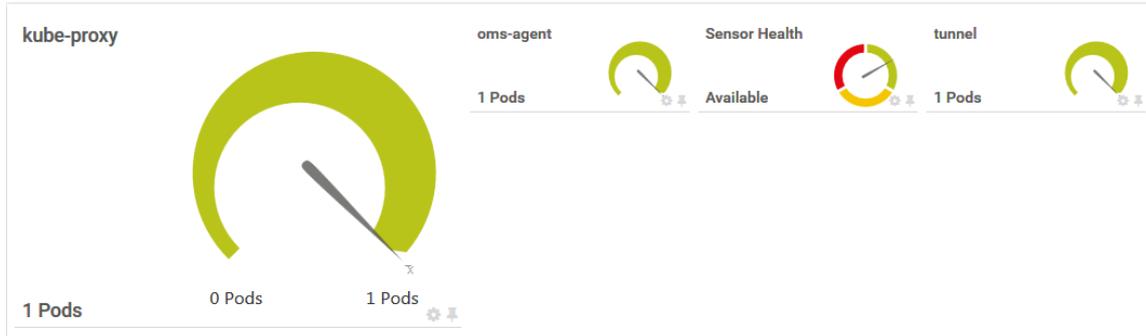
### 8.31 AKS Deployments

The Azure Kubernetes Deployments sensor shows per each deployment the number of running pods (pod is a core building block of a deployment managed by Kubernetes) in the AKS cluster.



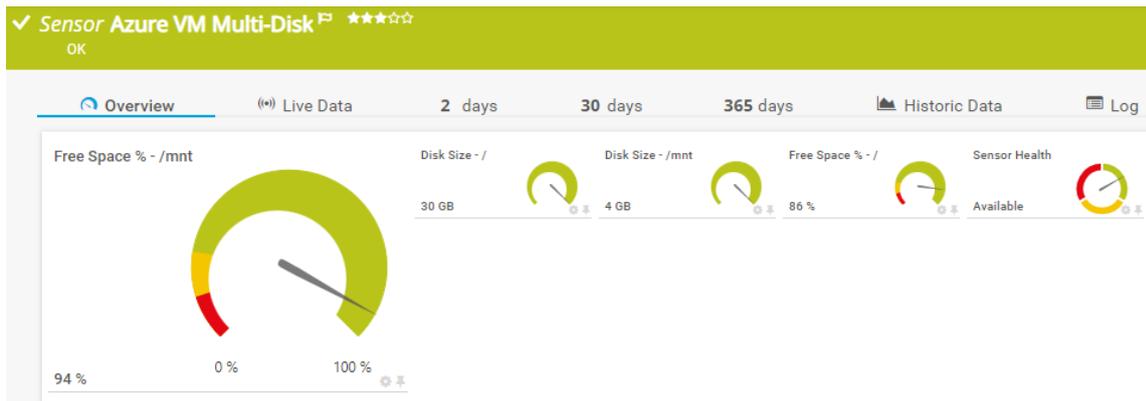
### 8.32 AKS Internal Deployments

The Azure Kubernetes Internal Deployments sensor shows the Azure-specific (internal) Kubernetes deployments and their pod count.



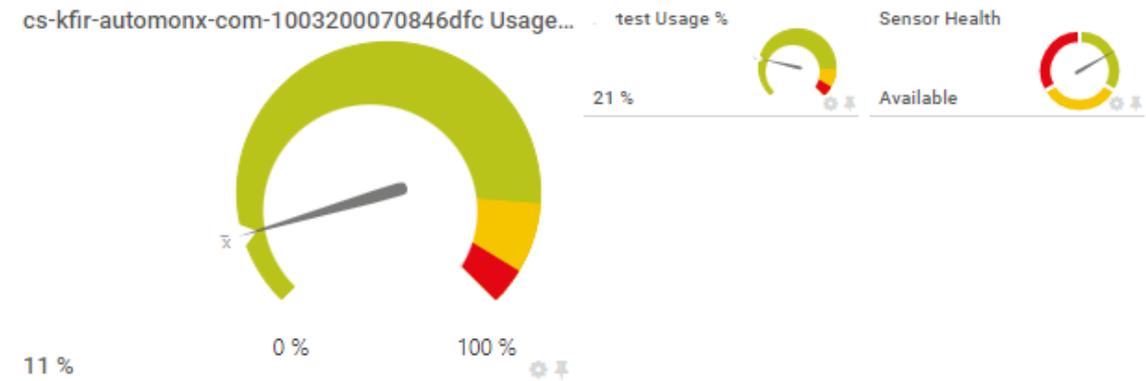
### 8.33 Azure VM Multi-Disk

After activating Insights for a VM, available disk space and size can be monitored through the Log Analytics API. Please follow [this guide](#) to activate this in the Azure portal.



### 8.34 Azure Storage File Share

This sensor shows the usage of your file shares.



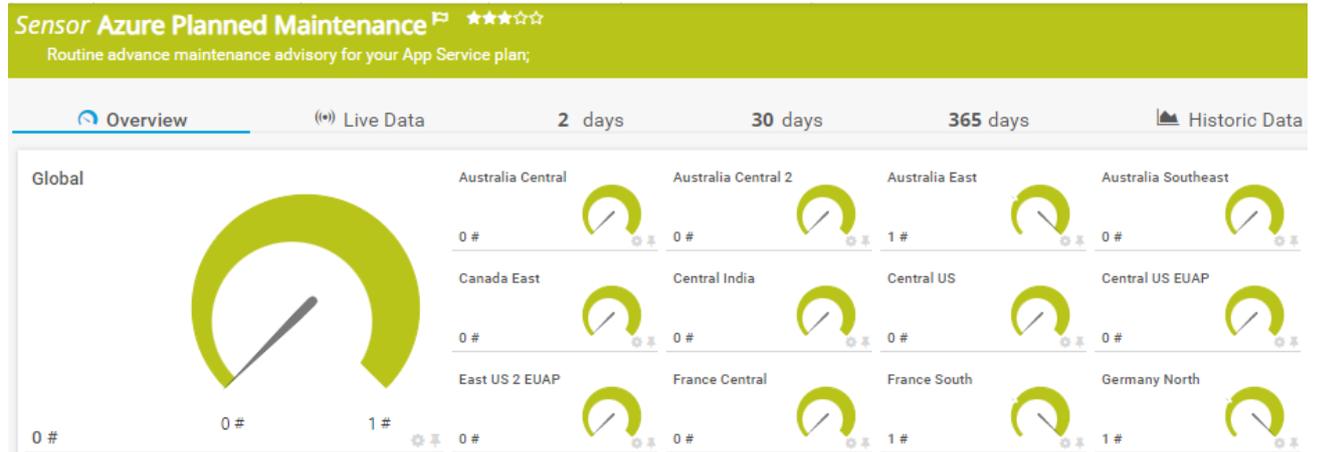
### 8.35 Azure Service Issues

Azure Service Issues sensor monitors the active issues and events per Azure subscription per region as seen in Service Health in the Azure portal. We recommend setting your own limits.



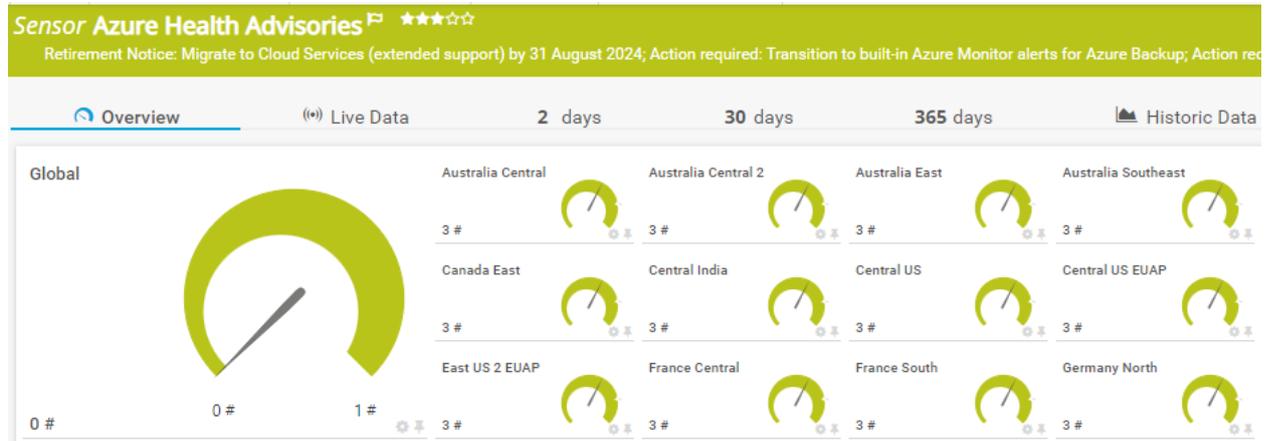
### 8.36 Azure Planned Maintenance

The Azure Planned Maintenance sensor monitors the Azure-planned maintenance that may impact your resources. It shows the Azure regions and the number of planned maintenances in each region.



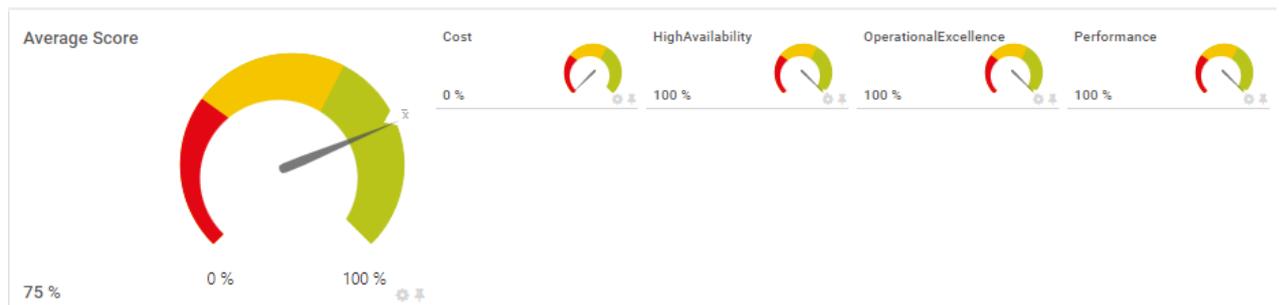
### 8.37 Azure Health Advisories

Azure Health Advisories monitors the number of Azure advisories per the geographic locations where your Azure resources are located.



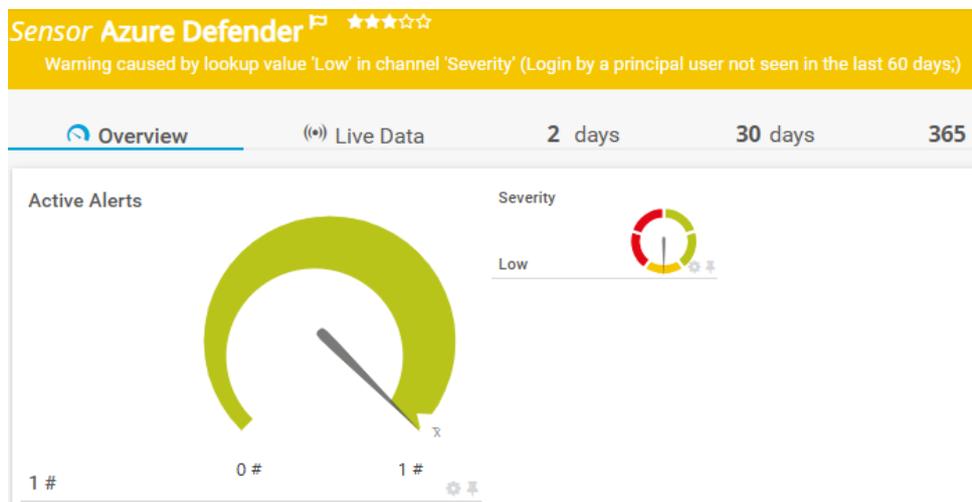
### 8.38 Azure Advisor

Per subscription Advisor score as seen in Azure portal, showing each category score to assess how well-architected your workloads are.



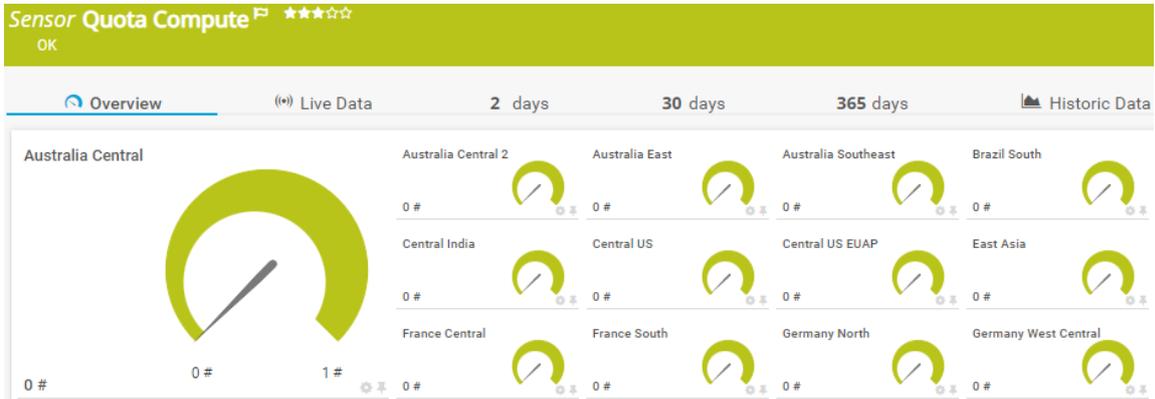
### 8.39 Azure Defender

Azure Defender for cloud lists the security alerts that currently impact your resources.



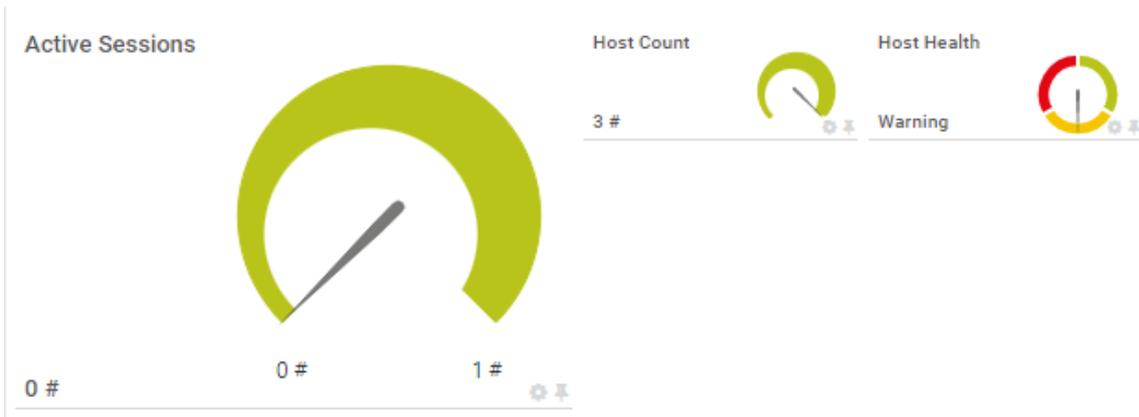
### 8.40 Azure Quotas

The Azure Quotas sensor monitors the number of resources that have reached their quota per resource type (i.e Compute, Storage, Vmware, Networking etc). The quota limit is configured in an INI parameter AZURE\_QUOTA\_LIMIT in AutoMonX\_AzureSensor.ini. Regions with no usage will not be displayed.



### 8.41 Azure Host Pool

Monitors the Virtual Desktop Hosts under Host Pools.

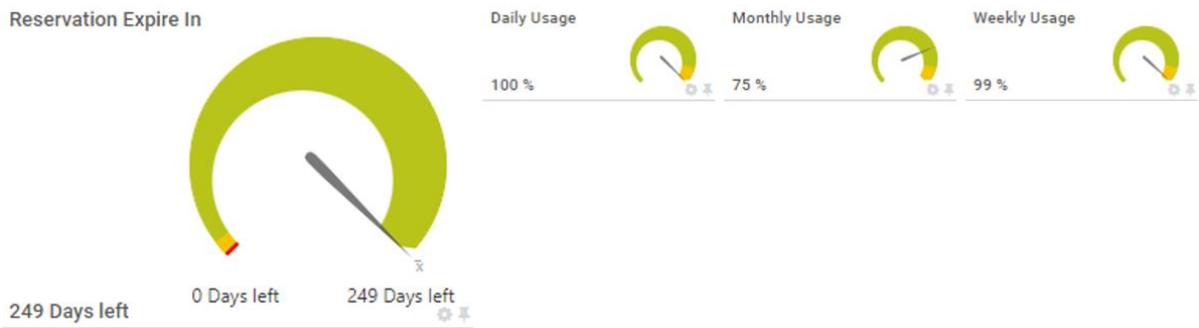


### 8.42 Azure Reservations

This sensor is created per Azure subscription. It measures the percentage of the subscriptions' usage of the reservations you have set and the reservation expiry time.

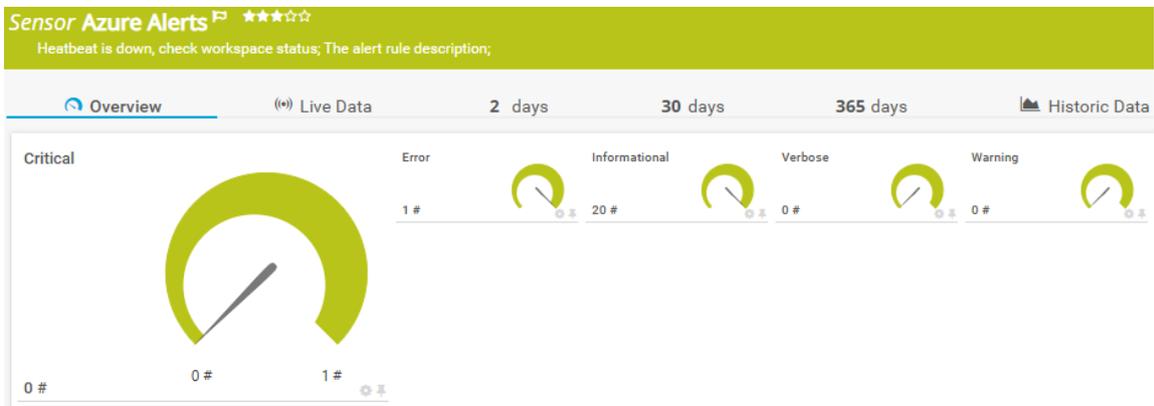
To discover this resource, you must add a permission to the service principle as described in [appendix B 2](#).

**Note** – Archived reservations are not discovered.



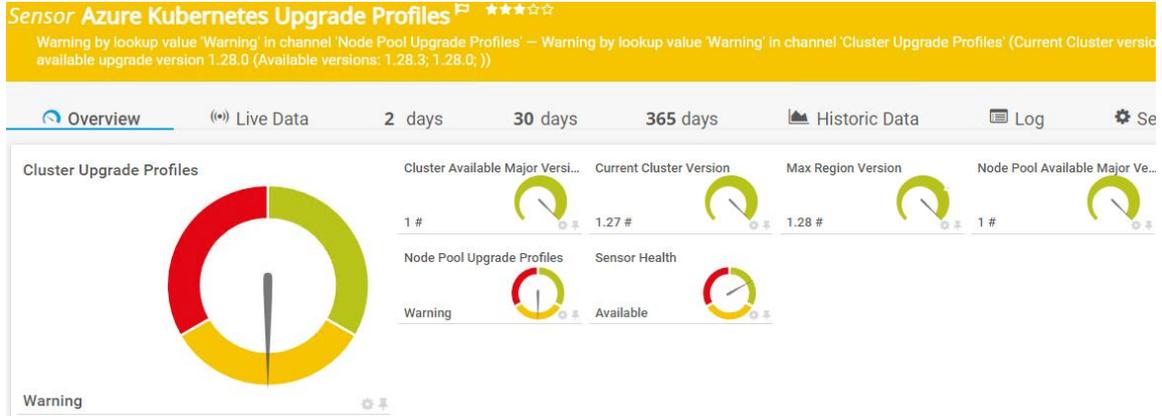
### 8.43 Azure Alerts

Allows a general overview of the number of active alerts by their severity and their short description. It allows you to quickly get notified when something major happens in your Azure estate. Set your own relevant thresholds.



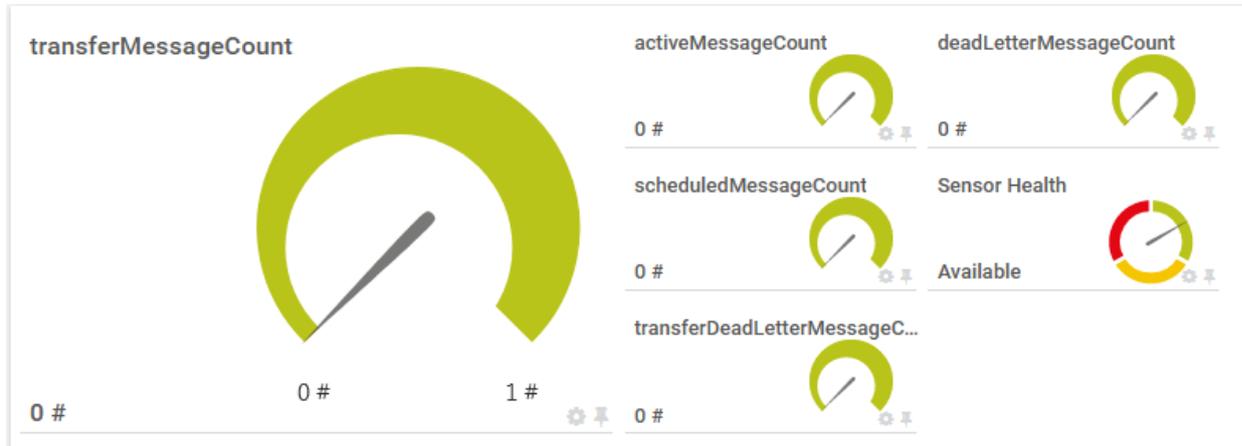
### 8.44 AKS Upgrade Profiles

Allows to monitor the version of your AKS deployment and generate an alert when a new software upgrade is available. Preview versions will not trigger an alert.



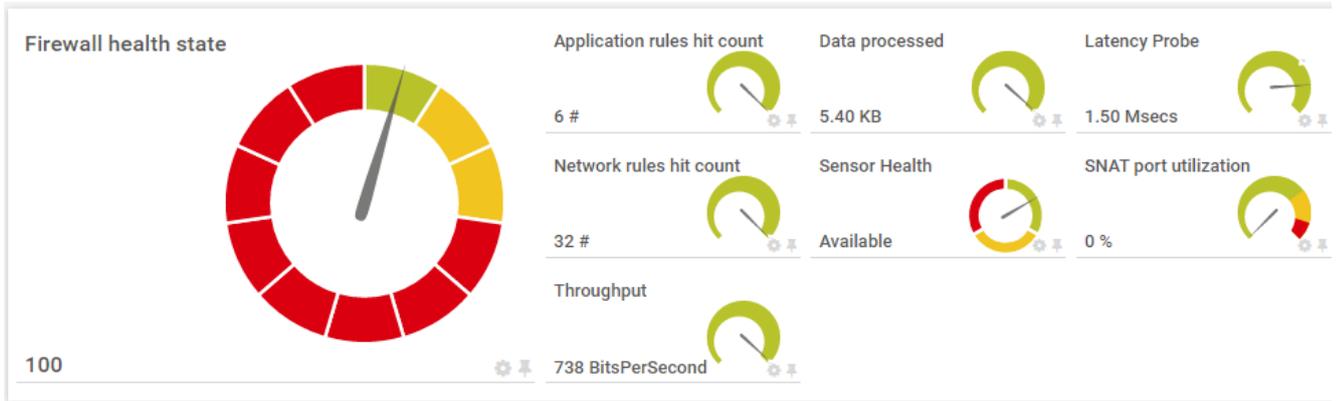
### 8.45 Azure Service Bus Topic Subscriptions

This sensor extends our Azure Service Bus sensor. It adds the ability to monitor the messages counts of each subscription to a Service Bus topic.



### 8.46 Azure Firewalls

Azure Firewall is a cloud-native and intelligent network firewall security service that provides the best of breed threat protection for your cloud workloads running in Azure. This sensor collects the metrics available for Azure firewalls and its health state.



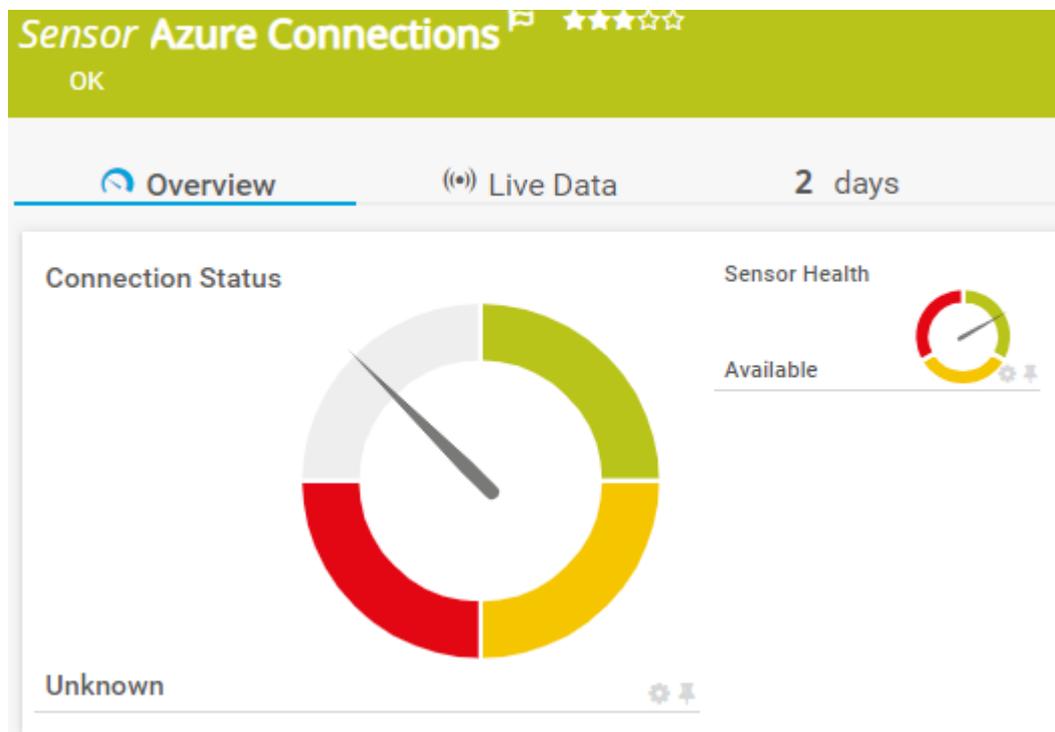
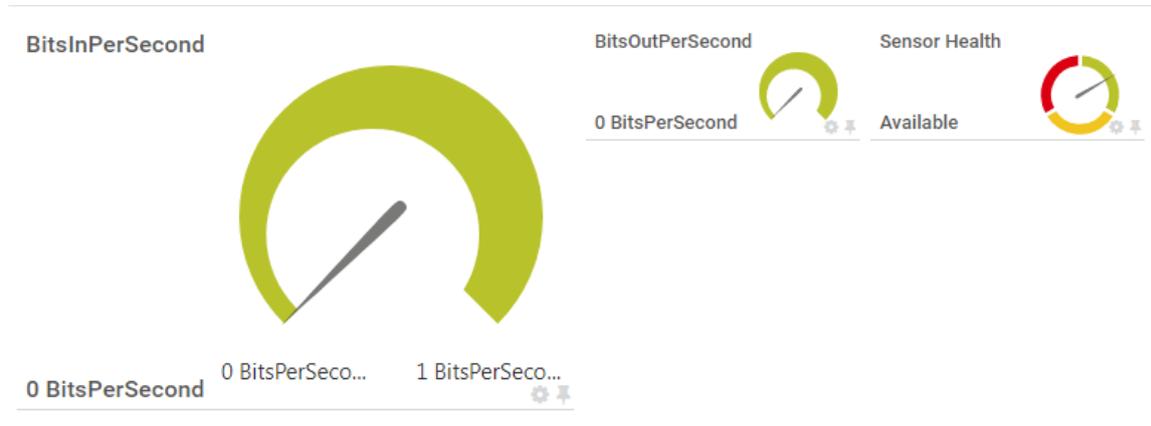
### 8.47 Azure Application Gateway (WAF)

Azure Web Application Firewall is a cloud-native service that protects web apps from common web-hacking techniques such as SQL injection and security vulnerabilities such as cross-site scripting. This sensor provides the vital metrics of the Azure WAF



## 8.48 Azure (VPN) Connections

Site to Site and VNet to VNet VPN Gateway Connection metrics & status.



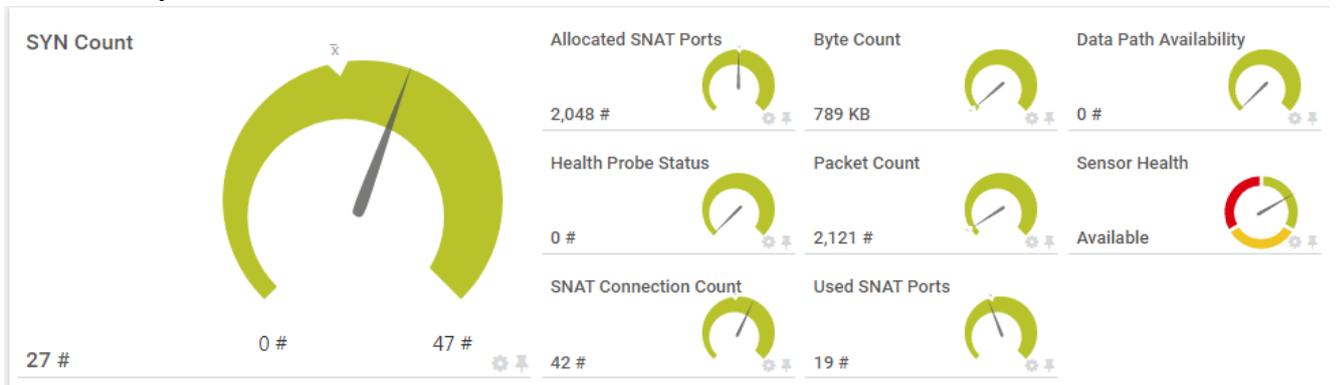
### 8.49 IP Address

The Azure IP Address sensor provides an in-depth view of the public Azure IP Address metrics.



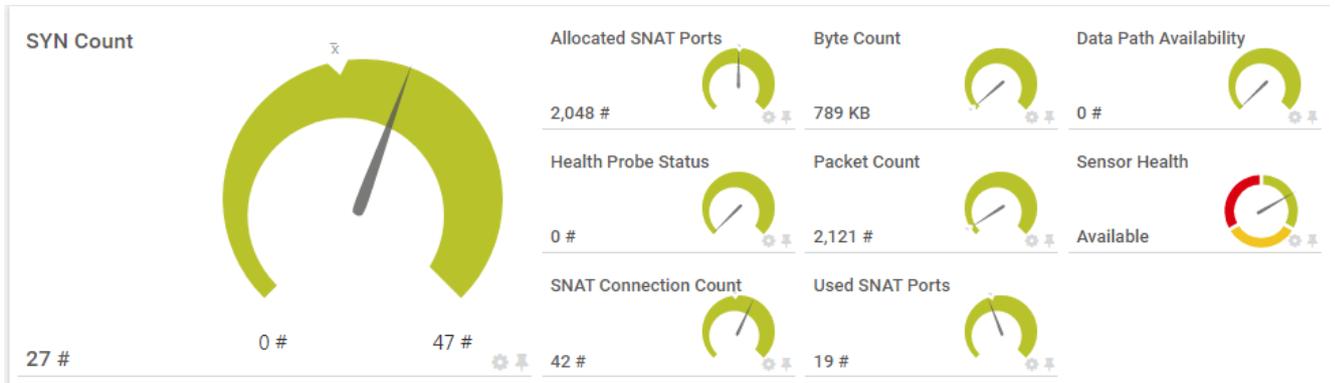
### 8.50 Azure Load Balancer

Load balancer distributes inbound flows that arrive at the load balancer's front end to backend pool instances. The backend pool instances can be Azure Virtual Machines or instances in a Virtual Machine Scale Set. The Azure Load Balancer sensors provides some interesting performance metrics to monitor its functionality.



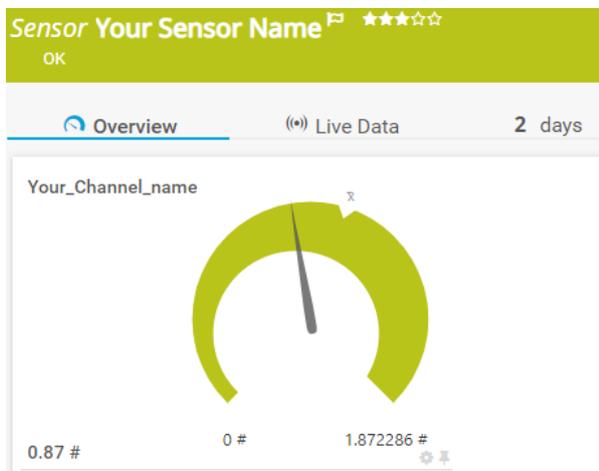
### 8.51 Azure Network Watcher (deprecated)

Azure Network Watcher provides a suite of tools to monitor, diagnose, view metrics, and enable or disable logs for Azure IaaS (Infrastructure-as-a-Service) resources.



### 8.52 Azure Log Analytics Custom Sensor

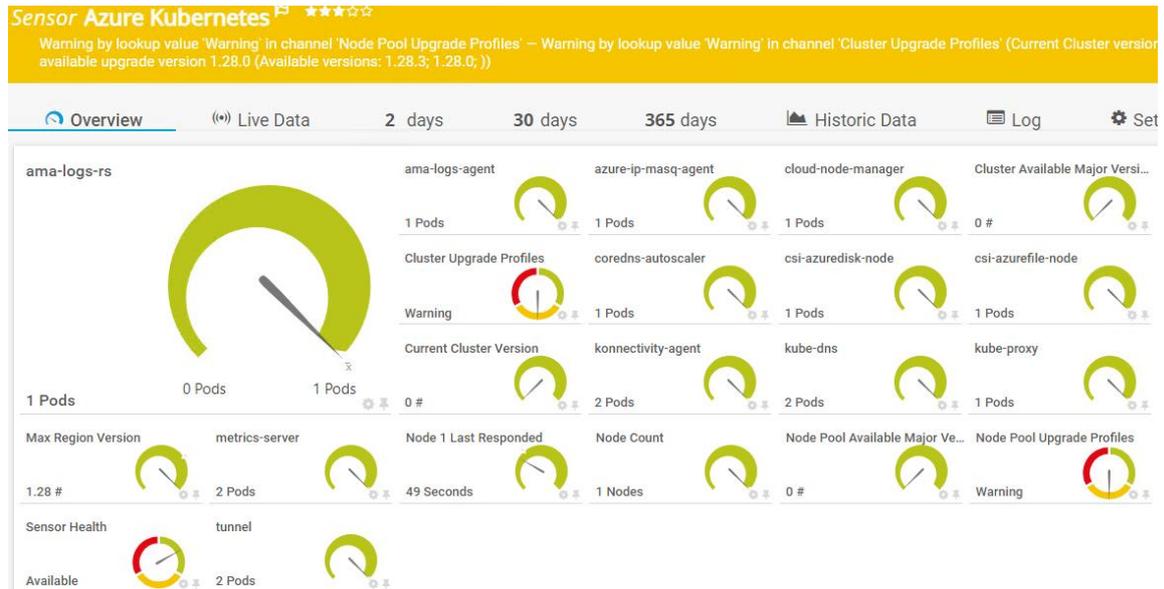
Custom sensor created by your own KQL query in Azure Log Analytics. The sensor configuration is explained in [Appendix D](#).



### 8.53 Azure Kubernetes combined sensor

For Clients that need to optimize sensor use, this sensor combines all AKS specific sensors listed above. To activate this sensor discovery the following configuration must be added to the file “AutoMonX\_AzureSensor.ini”

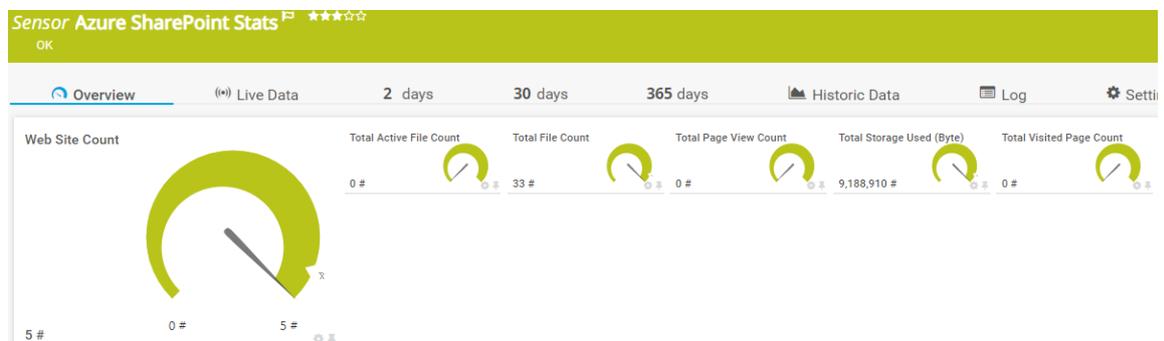
COMBINE\_AKS\_SENSORS=TRUE



### 8.54 Azure SharePoint Statistics

Displays the usage statistics for your active sites as seen in the SharePoint Admin center.

To allow the Application to discover the SharePoint metrics, please add the Reports.Read.All API Permission for the Application (Similarly to the process explained in [Chapter 14.1](#))



### 8.55 Azure Enterprise Secrets and Certificates

The Azure App Secret Keys and Certificates are used for authenticating with the Azure API for a specific Azure App (Enterprise Application that do not appear in App Registrations). Each key has an expiry date. After the secret key has expired it can no longer be used. The Azure App Enterprise Key sensor displays the remaining time until its expiry date.

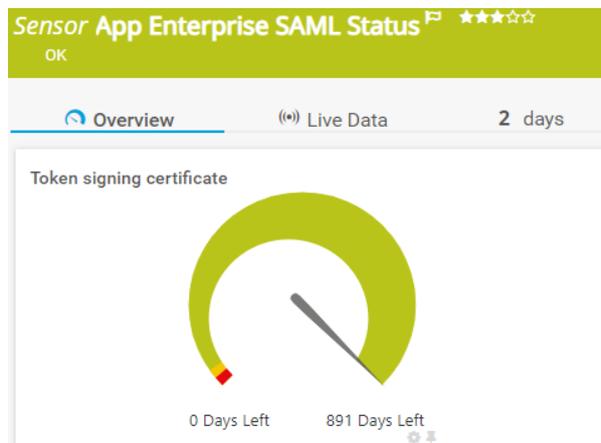
To make the Azure Secret Keys discoverable and available for monitoring, you need to follow the procedure in [Chapter 14.1](#).



### 8.56 Azure SAML Enterprise Certificates

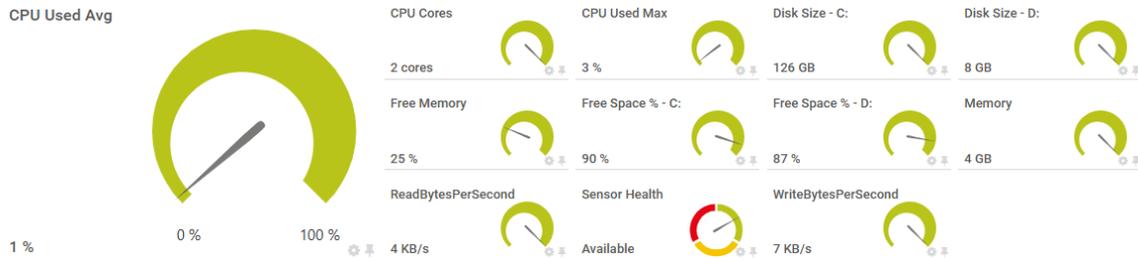
The Azure SAML (Security Assertion Markup) certificates are used for authenticating with the Azure Entra to establish federated single sign-on (SSO) to your software as a service (SaaS) Enterprise Applications. Each Azure SAML Certificate sensor can monitor a single application.

Each SAML certificate has an expiry date. After the SAML certificate has expired it can no longer be used. The Azure SAML Certificates sensor displays the status of the certificate and the remaining time until its expiry date.



### 8.57 Azure Arc Machines

The Azure Portal does not automatically show metrics for machines connected with Arc. Activate the Insights Log Analytics (explained in [Appendix C](#)), and this sensor will fetch the CPU, Memory, Disk and Network metrics for your Arc Machine.



### 8.58 Azure DevOps Pipelines

Azure Pipelines provides a quick, easy, and safe way to automate building your projects with consistent and quality code that's readily available to users. Add this sensor to show the status of the last pipeline run. See [chapter 14.4](#) on how to add permissions for the Azure DevOps Pipelines. Projects with more than 45 pipelines will be broken into several sensors.

By default, the status shows the last run result of the pipeline. The channels are configurable to check for success in the last X hours, by adding this line to the file "AutoMonX\_AzureSensor.ini":

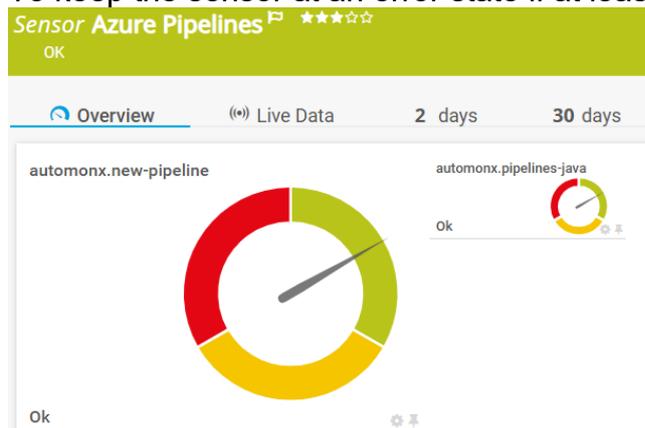
PIPELINE\_<name>\_CHECK\_HOURS=24

And restart the service for the changes to take effect.

To monitor fails of the pipeline, also add the line

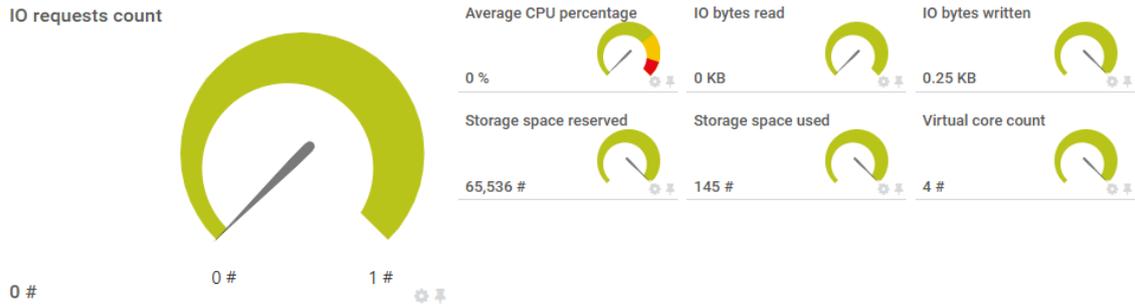
PIPELINE\_<name>\_CHECK\_FAILS=TRUE

To keep the sensor at an error state if at least one run failed in the last 24 hours.



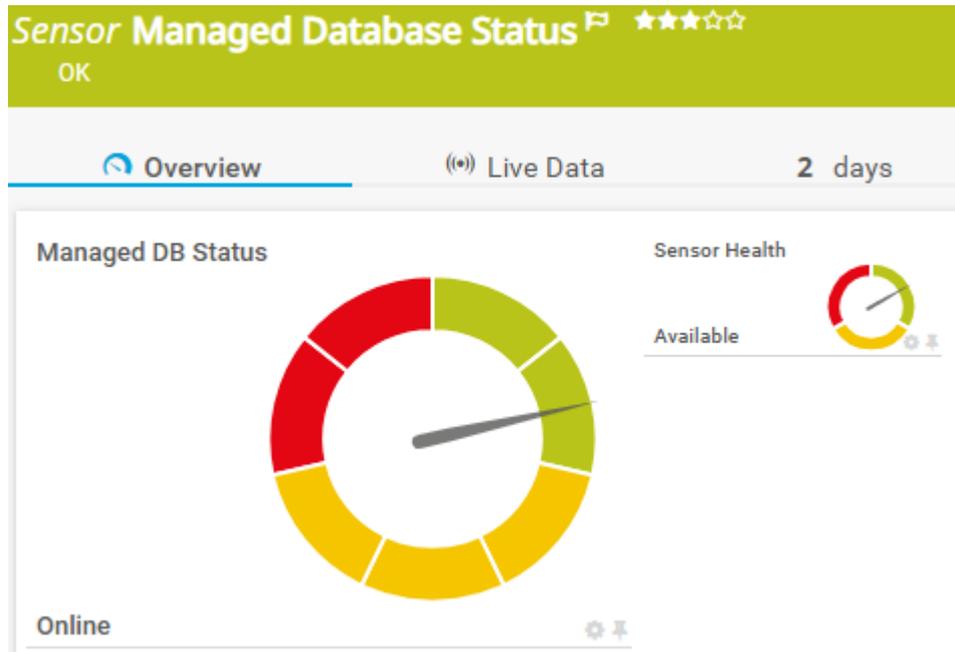
### 8.59 Azure SQL Managed Instances

SQL Managed Instance is an intelligent cloud database service combining the broadest SQL Server engine compatibility with the benefits of a fully managed, up-to-date platform as a service (PaaS) database engine that handles most database management functions such as upgrading, patching, backups, and monitoring without user involvement.



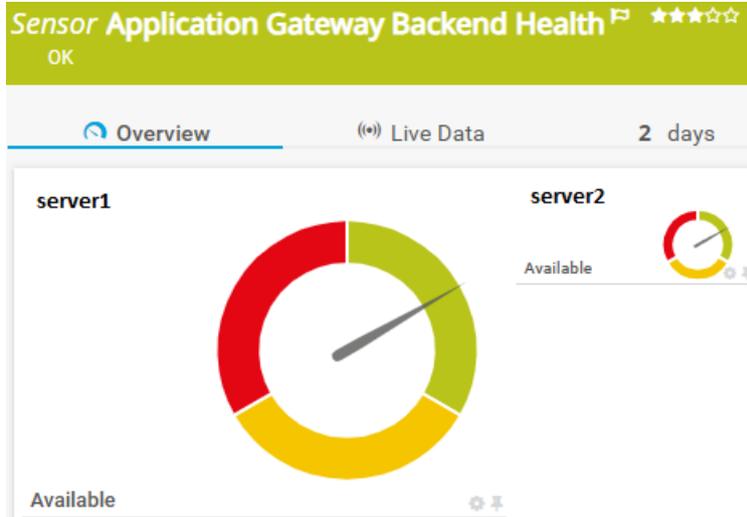
### 8.60 Azure Managed databases

Status of the Databases managed by the SQL Managed Instance. Located under the same device as the SQL Managed Instance.



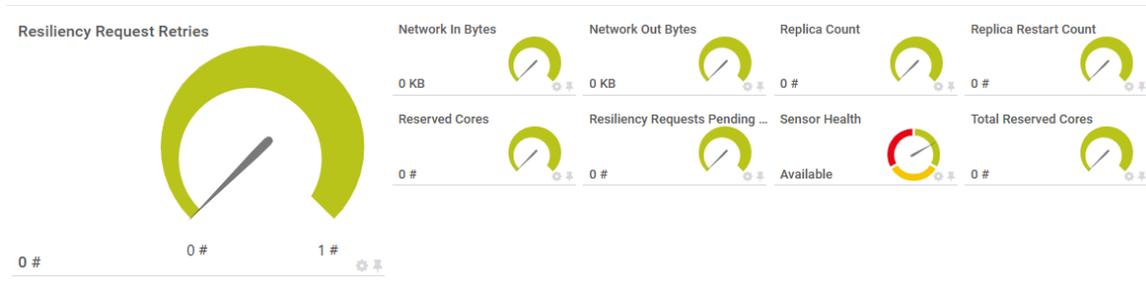
### 8.61 Azure Application Gateway backend servers

Application Gateway health probes continuously monitor all the backend servers in a pool to ensure the incoming traffic is sent only to the servers that are up and running.



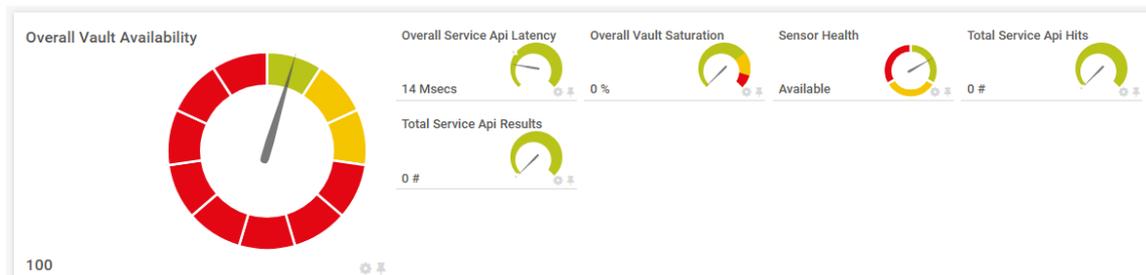
### 8.62 Container Apps

Azure Container Apps enables you to run microservices and containerized applications on a serverless platform.

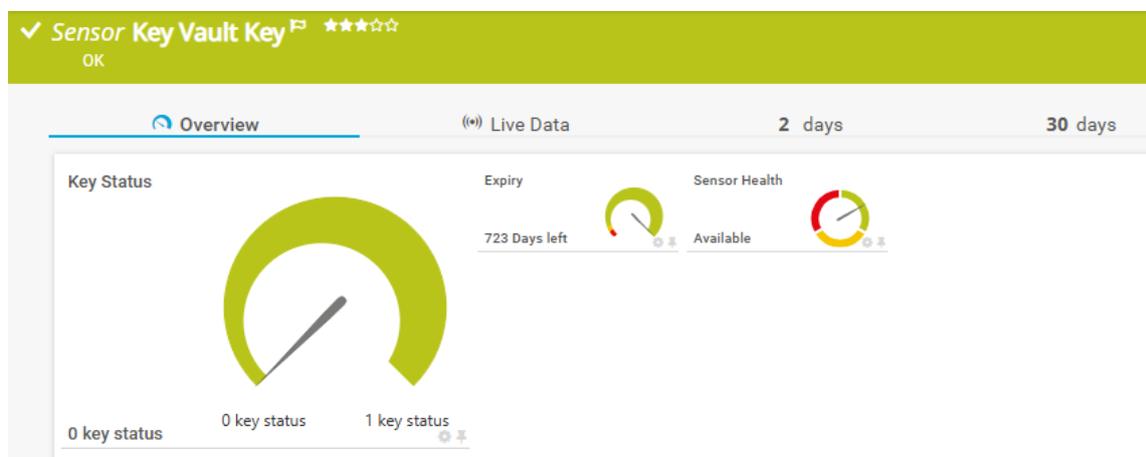
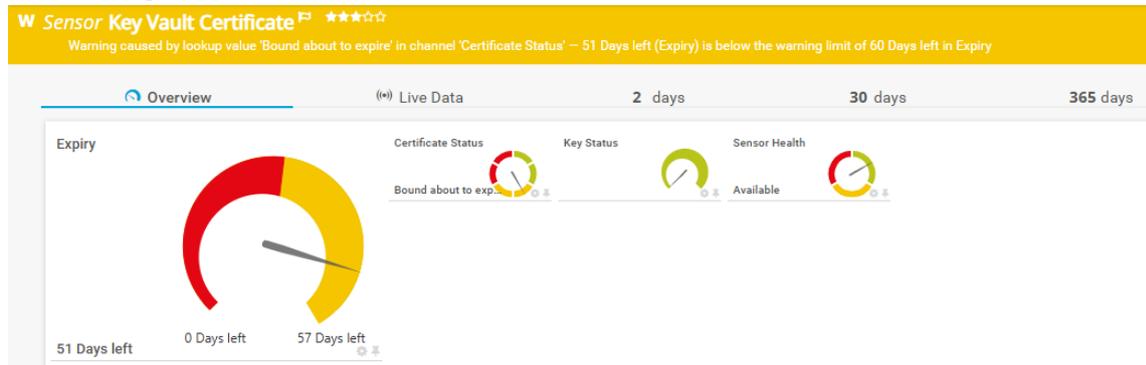


### 8.63 Key Vaults

Azure Key Vault is a secure secrets store, providing management for secrets, keys, and certificates, all backed by Hardware Security.



The supported resources are Key Vault Keys and Key Vault Certificates monitoring:

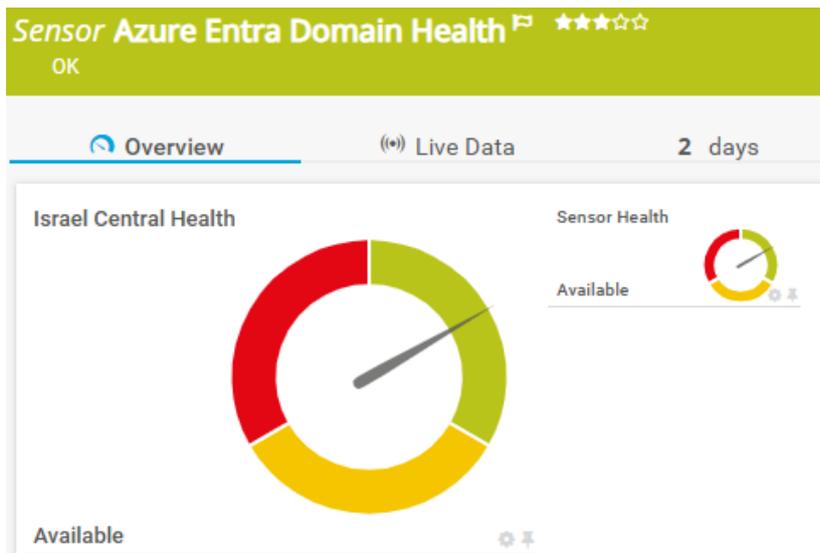


To use Key Vault Certificates\Keys monitoring you need to perform the following:

- If you use **RBAC roles** in the key vault(s) make sure that the service principal that you use for monitoring has "**Key Vault Certificates User**" and "**Key Vault Crypto User**" roles.
- If you use the **classic permissions mode**, make sure that the Service Principal you used for monitoring is added with **Read\List** permissions (Access policies) in the Key Vault.  
If the permissions are correct, check our [troubleshooting appendix](#) for more troubleshooting steps.

## 8.64 Azure Entra Domain Health

Microsoft Entra Domain Services runs some background tasks to keep the managed domain healthy and up-to-date. This sensor shows a summary of alerts in each replica set in your Entra Domain Service.

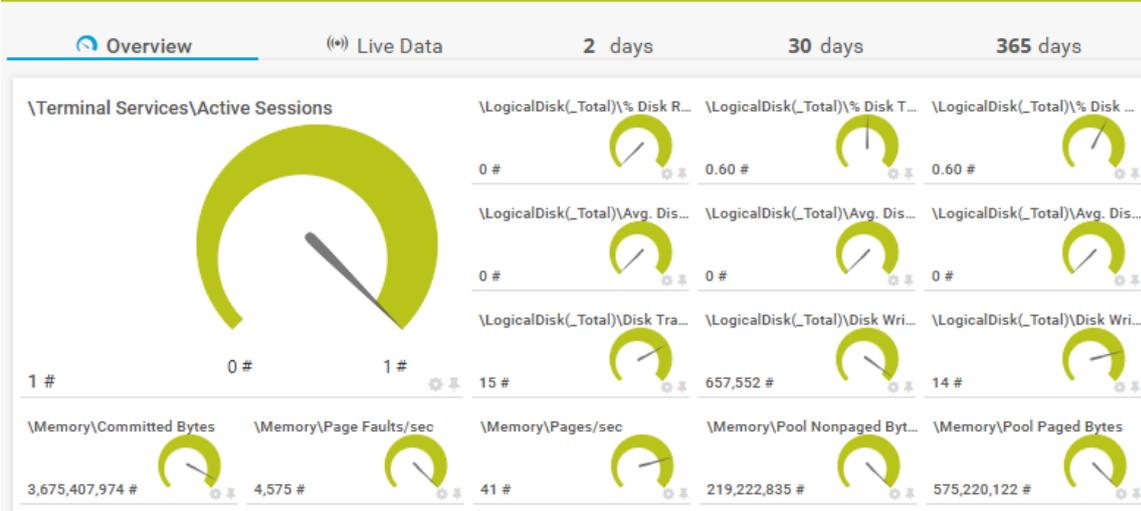


## 8.65 Azure Windows Guest Metrics

Metrics are collected from the guest operating system of a virtual machine. You can enable guest OS metrics for Windows virtual machines by using the Azure Monitor Agent, as explained in [Appendix E](#)

Sensor Azure Windows Guest Metrics ☆☆☆☆

OK



## 9 Troubleshooting

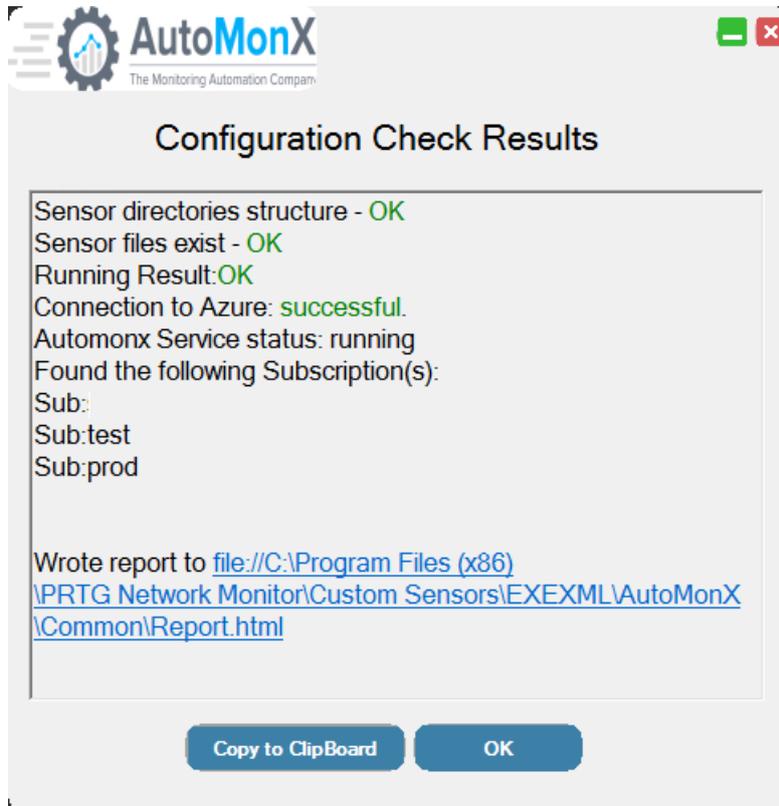
### 9.1 Troubleshooting the Azure Sensor pack Installation

Problem Description	Troubleshooting Steps
Azure Sensor pack Service is not starting	<ol style="list-style-type: none"> <li>1. Run Check Config via the UI (<b>as administrator</b>), check the results and fix any problems. Refer to <a href="#">Troubleshooting the Azure Configuration</a></li> <li>2. Make sure your Azure User is set up OK. Refer to <a href="#">Troubleshooting the Azure Connection Error</a></li> <li>3. Make sure the PRTG probe is open to the Internet and can access Microsoft Azure</li> <li>4. Make sure that the Product Key is valid</li> <li>5. Use the service debug mode to check service errors. Refer to <a href="#">Debug Using Service Debug Mode</a></li> </ol>
Discovery is not providing any results	<ol style="list-style-type: none"> <li>1. Make sure that the Azure Sensor Pack service is running</li> <li>2. Make sure your Azure User has enough permissions to the desired subscription. Refer to <a href="#">Troubleshooting the Azure Connection Permission</a></li> <li>3. Submit a support request via <a href="mailto:support@automonx.com">support@automonx.com</a> and send the following log files: Refer to <a href="#">Sending the Discovery Files to the support team.</a></li> </ol>
Discovery provides partial results	<ol style="list-style-type: none"> <li>1. Make sure your Azure User has enough permissions to the desired subscription. Refer to <a href="#">Troubleshooting the Azure Connection Permission</a></li> <li>2. Make sure Diagnostics is enabled for the desired resource. Refer to <a href="#">Troubleshooting Missing Azure Resource Metrics</a></li> <li>3. Submit a support request via <a href="mailto:support@automonx.com">support@automonx.com</a> and send the following log files: Refer to <a href="#">Sending the Discovery Files to the support team.</a></li> </ol>
Discovery is not able to discover the Azure Billing resources	Make sure your Azure User has enough permissions to the desired subscription. Refer to <a href="#">Troubleshooting the Azure Connection Permission</a>
Discovery is not able to discover Azure App metrics sensors	Make sure Diagnostics is enabled for the desired resource.

	Refer to <a href="#">Troubleshooting Missing Azure Resource Metrics</a>
Discovery is not able to discover Azure Service Health metrics sensors	Make sure service health is enabled for all the Azure resources Refer to <a href="#">Troubleshooting Missing Service Health</a>
Azure Sensors are Down with error message: The AutoMonX Azure service is down, cannot connect	Make sure that the Azure connection parameters are correct (use the AutoMonX UI and run Config Check)  Make sure that Azure is not blocked by a proxy server or Firewall of your organization
Warning: lookup that is not available / undefined lookup	Follow the steps to reload the lookup files depending on your PRTG core ( <a href="#">on-prem</a> or <a href="#">hosted</a> )



Below is an example of a successful configuration check:



```
Connection to Azure: successful.  
Automonx Service status: running  
Found the following Subscription(s):  
Sub:| _m1  
Sub:test  
Sub:prod
```

The Azure sensor was able to connect to Azure using the supplied information, the service is up and running and subscriptions were found in the Azure account.

### 9.3 Troubleshooting Azure Discovery Connection Errors

In case of a failed connection to Azure, an error will be displayed in the messages area of the Configuration UI.

Possible causes can be that Azure is unreachable due to limitations of network access, incorrect connection information such as App ID, Secret Key, or Tenant ID.

Check the Azure sensor settings using the instructions in [Section 6.3](#).

### 9.4 Troubleshooting Azure Discovery - Permissions

When no subscriptions are found during discovery, it means that the current credentials set doesn't have permissions to view any subscriptions and their resources under your Azure account.

Please assign at least a **Reader** role for the connection ID on the subscriptions you want to monitor. Please refer to [Microsoft Azure - assign-the-application-to-a-role](#)

### 9.5 Collecting the Discovery Files for AutoMonX Support

In case of any other problems encountered during Azure discovery, open a case with our support team at [support@automonx.com](mailto:support@automonx.com). You would need to provide the following information:

- Discovery log file - **AutoMonX\_Discovery\_out.log**
- Discovery results in a form of CSV files with names such as:  
**AutoMonX\_Azure\_automation\_<Sub> .csv**
- The Azure types file: **AutoMonX\_Azure\_types.dat**

These files are located in the following directory:

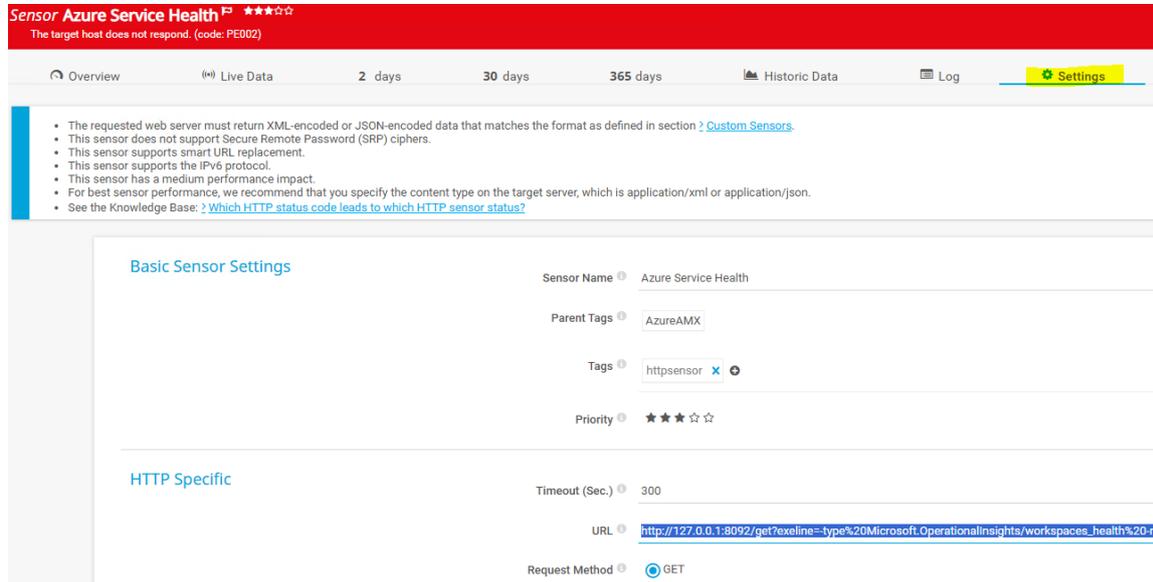
**Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure**

- The output of the following command:

**AutoMonX\_AzureCollector.exe -discovery -sub -All**

## 9.6 Troubleshooting Sensors Locally

If after first discovery all sensors are down, open the setting of one of the sensors and copy the URL.



**Sensor Azure Service Health** ★★★☆☆  
The target host does not respond. (code: PE002)

Overview Live Data 2 days 30 days 365 days Historic Data Log Settings

- The requested web server must return XML-encoded or JSON-encoded data that matches the format as defined in section > [Custom Sensors](#).
- This sensor does not support Secure Remote Password (SRP) ciphers.
- This sensor supports smart URL replacement.
- This sensor supports the IPv6 protocol.
- This sensor has a medium performance impact.
- For best sensor performance, we recommend that you specify the content type on the target server, which is application/xml or application/json.
- See the Knowledge Base: > [Which HTTP status code leads to which HTTP sensor status?](#)

**Basic Sensor Settings**

Sensor Name: Azure Service Health

Parent Tags: AzureAMX

Tags: httpsensor

Priority: ★★★☆☆

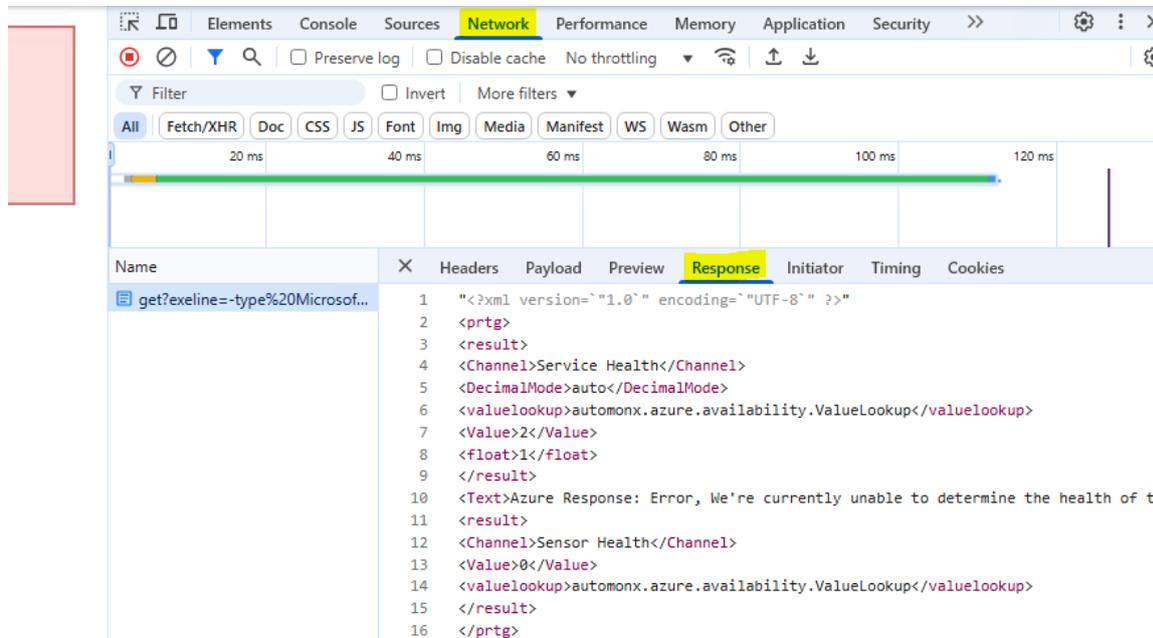
**HTTP Specific**

Timeout (Sec.): 300

URL: [http://127.0.0.1:8092/get?exeline=-type%20Microsoft.OperationalInsights/workspaces\\_health%20-r](http://127.0.0.1:8092/get?exeline=-type%20Microsoft.OperationalInsights/workspaces_health%20-r)

Request Method: GET

Paste the URL to your local Probe browser. Open the developer tools (F12). Press Network. Now refresh the page to generate a polling request. Press on the URL and on Response. If you see an XML like in the example below, the local service is working well. If not, there is an issue with the service. Run Configuration Check to see the problem.



Elements Console Sources Network Performance Memory Application Security

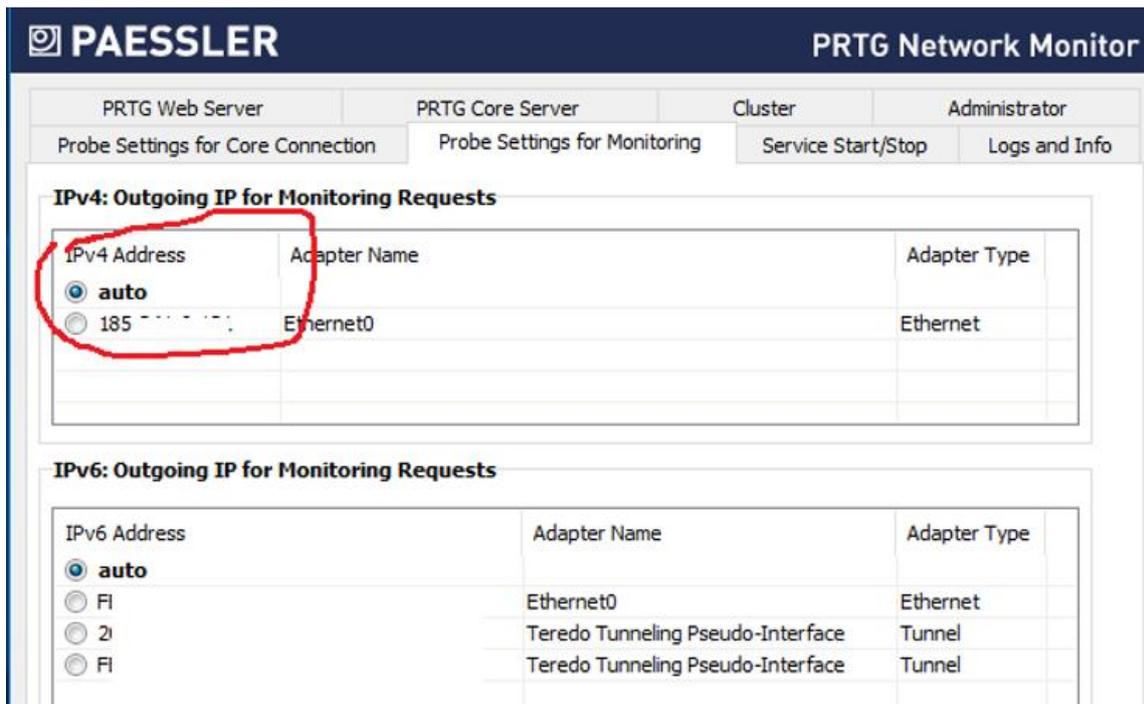
Filter: Invert More filters

All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

20 ms 40 ms 60 ms 80 ms 100 ms 120 ms

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
get?exeline=-type%20Microsof...				<pre> 1  "&lt;?xml version="1.0" encoding="UTF-8" ?&gt;" 2  &lt;prtg&gt; 3  &lt;result&gt; 4  &lt;Channel&gt;Service Health&lt;/Channel&gt; 5  &lt;DecimalMode&gt;auto&lt;/DecimalMode&gt; 6  &lt;valuelookup&gt;automonx.azure.availability.ValueLookup&lt;/valuelookup&gt; 7  &lt;Value&gt;2&lt;/Value&gt; 8  &lt;float&gt;1&lt;/float&gt; 9  &lt;/result&gt; 10 &lt;Text&gt;Azure Response: Error, We're currently unable to determine the health of t 11 &lt;result&gt; 12 &lt;Channel&gt;Sensor Health&lt;/Channel&gt; 13 &lt;Value&gt;0&lt;/Value&gt; 14 &lt;valuelookup&gt;automonx.azure.availability.ValueLookup&lt;/valuelookup&gt; 15 &lt;/result&gt; 16 &lt;/prtg&gt; </pre>			

Make sure the following configuration is set on your PRTG Probe:

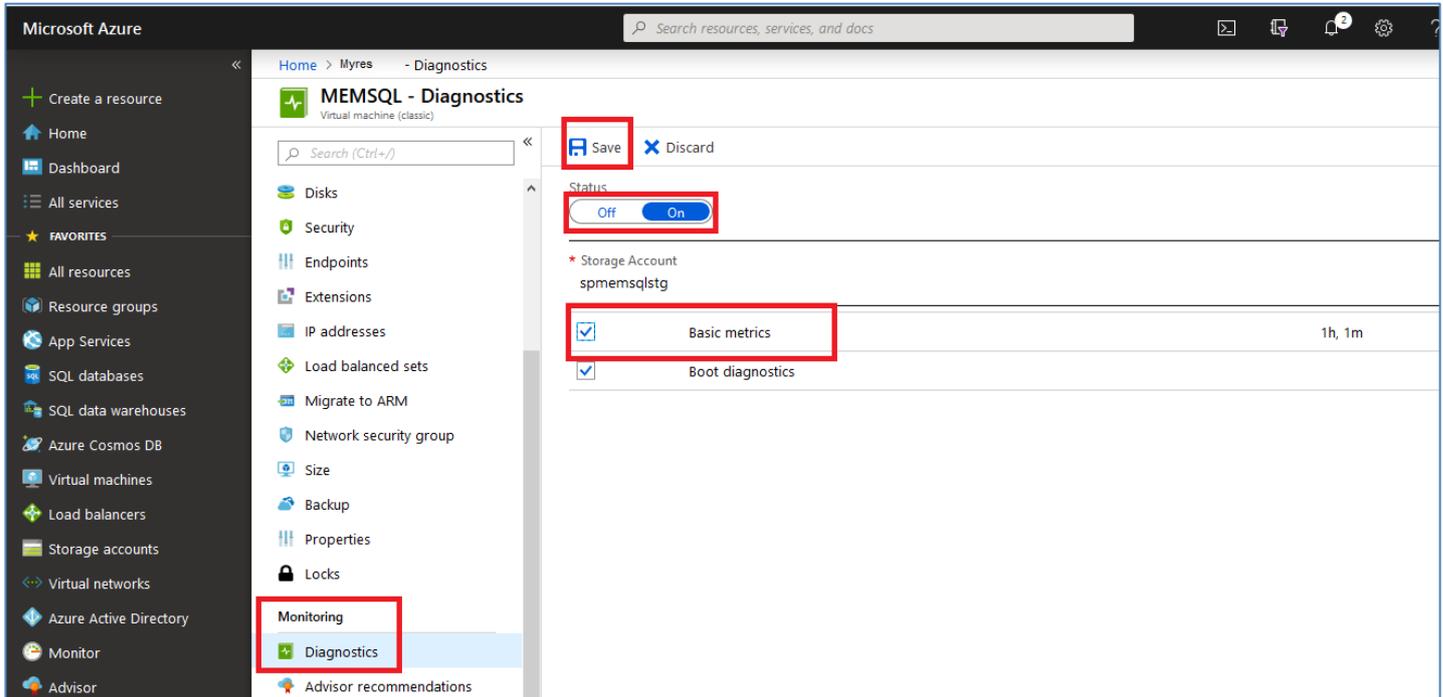


## 9.7 Troubleshooting the Discovery of Azure Metrics

When the discovery process can't discover one or more Azure resource's App Metrics data, make sure that the Azure diagnostics was enabled for this resource.

To enable the resource diagnostics, navigate to the Azure resource settings page in the Azure admin portal.

On the left side of the resource page, click on Diagnostics. Set up the metrics collections settings and make sure they are turned on (enabled)



## 9.8 Troubleshooting the Discovery of Azure Service Health

When the discovery process can't discover the Azure Service Health sensors, make sure service health was enabled in the Azure portal for these resources in the link below:

[https://portal.azure.com/#blade/Microsoft\\_Azure\\_Health/AzureHealthBrowseBlade/resourceHealth](https://portal.azure.com/#blade/Microsoft_Azure_Health/AzureHealthBrowseBlade/resourceHealth)

## 9.9 Collecting Azure Service Debug information

To activate the service debugger, you would need to set the SERVICE\_DEBUG variable to 1 (default is 0) in the AutoMonX\_AzureSensor.ini file. This setting will activate the service debug mode upon the next start of the service.

During debug mode, a special log file is created. This file tracks all the Azure sensor service operations. This file needs to be examined by the AutoMonX support team to detect any issues. Open a case with our support team at [support@automonx.com](mailto:support@automonx.com). You would need to provide the following file:

- Service Debugger file - AutoMonX\_ServiceDebugLogger

The files are in the following directory:

C:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure\Logs

## 9.10 Collecting Azure Sensor Debug Information

To activate the debug logs of the Azure sensors, add the -debug argument to the parameters of the AutoMonx\_AzureSensor.cmd in the PRTG sensor settings as seen below:

### Sensor Settings

The EXE file has to run on the computer where the parent probe is installed, not on the parent directory for EXE files is the probe directory. .vbs files, .ps1 files, or other script files may use other directories.

EXE/Script  Automonx\_AzureSensor.cmd

Parameters  -type Microsoft.Logic/workflows -resgrp mylogicapps -res "firstlogicapp" -sub "mpn" **-debug**

Environment   Default Environment  
 Set placeholders as environment values

**Please note** – when the sensor is of type HTTP, spaces will cause an error – you must add %20 instead:

### Basic Sensor Settings

Sensor Name ⓘ Azure Billing

---

Parent Tags ⓘ

Tags ⓘ  x +

---

Priority ⓘ ★★☆☆☆

### HTTP Specific

Timeout (Sec.) ⓘ 900

---

URL ⓘ <http://127.0.0.1:8092/get?exeline=-sub%20microsoft%20partner%20network%20cons%20-debug>

This setting will activate the sensor debug mode upon the next run of the sensor. The Azure sensor debug logs are created in the default PRTG sensors log directory. You can change the location of these logs by modifying the following variable in the AutoMonX\_AzureSensor.ini file:

```
DEBUG_LOG_DIR=C:\Temp
```

Typically, the log file name will look like the example below:

*AutoMonX\_Azure\_Microsoft.DocumentDb\_databaseAccounts\_res\_prod.log*

**Please note** – make sure to keep track of the sensors you activated debug for to revert after completion. Otherwise, the log file will keep growing and take up space on your probe machine.

## 9.11 Collecting In-Depth Azure Sensor Debug Information

To activate the in-depth debug logs of the Azure sensors, add the following parameters to the AutoMonx\_AzureSensor.cmd in the PRTG sensor settings.

```
-debug verbose
```

This setting will activate the sensor in-depth debug mode upon the next run of the sensor. The Azure sensor debug logs are created in the default PRTG sensors log directory. You can change the location of these logs by modifying the following variable in the AutoMonX\_AzureSensor.ini file:

```
DEBUG_LOG_DIR=C:\Temp
```

### HTTP Specific

Timeout (Sec.) ⓘ 60

---

URL ⓘ <icrosoft.Compute/virtualMachines/AzureTest%22%20-sub%20%22microsoft%20azure%20sponsorship%22%20-debug%20verbose>

## 10 Command Line Options (CLI)

### 10.1 The Azure Sensor Pack Command Line Options Reference

Option	Details
-install	Installs the AutoMonX Azure Sensor Service. The service communicates with the Azure environments, retrieves and stores the Azure communication token.
-config_check [-tenant <tenant_number>]	Checks the service communication to the Microsoft Azure environment.
-discovery -sub {<Subscription_Name>  -All}  -discovery -sub SubName [-tenant <tenant_number>] [-outmode] [-whitelist] [-resource_id] [-verbose]  -discovery -sub SubName -resources  -discovery -sub SubName -metrics  -discovery -sub SubName -new_resources	Discovers all the resources in an AZURE environment and creates a report in a HTML format. The report is AutoMonX_Azure_Report-<Sub Name>.html in the PRTG EXEXML program folder.  Use the resources flag or the metrics for a quicker discovery, that doesn't get all metrics. Running the discovery without these two last flags will perform the thorough discovery.
-create_conn_profile -tenant_label <display_label> -azure_appid <app_id> -azure_secretkey <secret_key> -azure_tenantid <tenant_id> [-encrypt_secret]	For multi-tenant users. Creates a new credentials set in the file "AzureConnProfiles.ini".
-remove_tenant -tenant <tenant_number>	For multi-tenant users. Deletes all information related to a specific tenant. <b>Use with caution!</b>
-install_backend	Installs a complementary service for REST API access to configuration and discovery.
-get_tenant_id -tenant_label <display_label> / -azure_tenantid <tenant_id>	Given a tenant label or full tenant id returns the configured internal tenant id.
-status_discovery	Activate with the same parameters as the discovery to get a progress status for the discovery process.
-update_key -tenant_label <display_label> -azure_secretkey <secret_key> [-encrypt_secret]	Updates existing App Key. To be used before the key expires. Will accept tenant_label, azure_tenantid or tenant (internal id)

-type <res_type> -resgrp <res_group> -res <res_name> -sub <res_subscription> [-debug]  -type Azure/Type -resgrp ResGroup -res ResName -sub SubName	These are the arguments required for the Sensor Pack to monitor a resource in the Microsoft Azure environment. This is only used with the AutoMonX_AzureSensorRun executable.
-cons -sub SubName	These are the arguments required for the Sensor Pack to monitor the billing of a subscription. This is only used with the AutoMonX_AzureSensorRun executable.
-licstatus	These are the arguments required for the Sensor Pack to monitor the AutoMonX License status. This is only used with the AutoMonX_AzureSensorRun executable.
-enc_data <secret_key>	Encrypts a given string. You can use it to encrypt the Azure secret key to safely store in the credentials file.
-ovl_map	Create a report of OVL values mapping
-validate_filter_file	Checks validity of the files "include_mon.csv" and "exclude_mon.csv" and saves the result to file "Automation_Syntax_Check.log"
-version	Displays the program's version.
-help	Displays the command option list.

## 10.2 Fully Automated Azure Monitoring

Starting with version 4.1.7, fully automated monitoring of your Azure estate is available as part of the Azure Sensor pack. To deploy it, you need to adapt a simple batch file, that can run on a scheduled and non-interactive fashion and cover the entire cycle of automatically discovering, adding to monitoring and even deleting dead/orphan resources from monitoring.

### 10.2.1 Automated Discovery and Monitoring

Depending on the structure of your Azure estate (Tenants, subs etc) you can easily adapt the contributed batch file below and create a Windows scheduled task for periodic discovery and automatic addition of discovery results for monitoring.

An example is available in:

“AutoMonX/Azure/Contrib/Amx\_Discovery\_Addition.cmd”. Please provide the pass hash and group name as parameters to the script.

*Amx\_Discovery\_Addition.cmd <passhash> <PRTG\_group>*

You may update the script to run on only specific subscription.

Please make sure the scheduled task is run with the highest privileges:

- Run only when user is logged on
- Run whether user is logged on or not
  - Do not store password. The task will only have access to local computer resources.
- Run with highest privileges

We recommend you add the `-metrics` flag to the discovery line after the initial run to shorten the discovery time.

## 10.2.2 Automated Clean-Up of Un-Needed Resources from Monitoring

**USE WITH EXTRA CAUTION:** Please note that deleting sensors is irreversible and will delete the sensors and all their historic data. This feature is bound to our EULA agreement. AutoMonX LTD will not be responsible for any direct or collateral damage due to usage of any of our products or their features. The deletion of sensors also deletes devices without any sensors and empty sub-groups to avoid phantom resources

This feature allows you to automatically delete any sensors in PRTG that no longer provide any useful monitoring data and seen in Down state. Typically, such state is seen because the Azure resource was deleted or shut down via the Azure Portal.

Make sure that the file "down\_sensors\_filter.ini" exists in the AutoMonX/Common folder. Otherwise copy it from the latest Azure Sensor Pack version zip archive and insert the message text of the sensors you wish to delete.

You can set a periodic sensor deletion of removed resources with a Windows scheduled task running a batch file. An example is available in "AutoMonX/Azure/Contrib/ Amx\_Sensor\_Deletion.cmd". Please provide the pass hash and PRTG group(s) name(s) as parameters to the script.

```
Amx_Sensor_Deletion.cmd <passhash> <PRTG_group(s)>
```

If the group name contains spaces, make sure to add quotation marks enclosing the "group name". It can also be run on several groups, separated by commas (for example: Web,Logic,"Azure test" ).

By default, it will delete sensors in a Down state that display an error message that contains the text "Resource Not Found".

To adjust this functionality to delete sensors with other error types, based on their last message text, you need to edit the *down\_sensors\_filter.ini* file.

Add a new line of text per the relevant sensor error message(s) as seen in PRTG, one per each line. Make sure to be as specific as possible.

The auto-deletion option can be used to clean-up any PRTG sensor types.

### 10.2.3 Automatically Pausing Un-Needed Resources

This feature allows you to automatically pause any sensor in PRTG that no longer provides any useful monitoring data and seen in Down state. Typically, this state is seen because the Azure resource has been deleted or shut down via the Azure Portal. It is useful for NOC teams that wish to investigate the source of the problem rather than automatically deleting the sensor(s).

Make sure that the file "pause\_sensors\_filter.ini" exists in the AutoMonX/Common folder. Otherwise copy it from the latest Azure Sensor Pack version zip archive and insert the relevant messages of the sensors you wish to pause.

You can set a periodic sensor pausing of removed resources with a Windows scheduled task running a batch file. Example:

```
AutoMonX_PRTG_Automation.exe -pause_sensors -p 11111111 -grouplist AutoMonX_Azure
```

If the group name contains spaces, make sure to add quotation marks enclosing the "group name". It can also be run on several groups, separated by commas (for example: Web,Logic,"Azure test" ).

By default, it will pause sensors in a Down state that display an error message that contains the text that caused the down message, in the following format: "AutoPaused:AutoMonX: Sensor Full Error: "<original error message">" if no message was found the following message would appear after the sensor pause: "AutoPaused:AutoMonX".

To adjust this functionality to pause sensors with other error types, based on their last message text, you need to edit the *pause\_sensors\_filter.ini* file.

Add a new line of text with the relevant sensor error message(s) as seen in PRTG, one per each line. Make sure to be as specific as possible.

The auto-pause feature can be used to pause any PRTG sensor types.

#### 10.2.3.1 Automatically Setting Removed Resources to Warning State

As an alternative to pausing the sensors, you may activate automatic setting of the sensor to Warning in PRTG (instead of Down) for Azure resources that have been deleted.

Update the following configuration value in *AutoMonX\_AzureSensor.ini* file and restart the service Azure Sensor Pack service:

```
SET_SENSOR_NOT_FOUND_TO_WARN=true
```

This will not require setting up any additional scheduled tasks but will only work for removed resources and not for user-specified phrases.

#### 10.2.4 Automated Inclusion and Exclusion of Sensors and Channels

Most of the Azure resources contain a multitude of monitoring information which yields lots of channels in PRTG, some of them are less useful for certain IT teams. Another frequently seen scenario is when the Azure monitoring teams wish to add only specific resource types to PRTG (i.e. only the production Databases). The purpose of this feature is to allow fine granularity of which Azure Resources and metrics you want to add to PRTG.

This section explains how to utilize the Exclude and Include Functions of AutoMonX Sensor Pack for Azure to control the addition of Azure Resource Groups, Azure Resources (devices), Azure sensors and Azure Metrics (channels) to PRTG. This functionality can replace the selection of sensors in the AutoMonX UI and allow full automation of the discovery and monitoring automation.

#### **Important:**

1. Make sure to run full discovery before applying any configuration to the filter files
2. The discovery process creates a csv file for each Azure Subscription with all discovered Resources, their sensors and channels in a format of the *include* and *exclude* files. Use this file to create your include or exclude filters. Typically, the name of the file would like “*AutoMonX\Azure\Logs\AutoMonX\_Channel\_Report-<sub>.csv*”.
3. Monitoring Automation will firstly process the *Include* file and then the *Exclude* file.
4. Edit the *exclude\_mon.csv* and *include\_mon.csv* files only with a simple text editor such as Notepad or Notepad++. Don't save these files in a non-textual format such as XLX or XLXS.

#### 10.2.4.1 Excluding an Azure Resource Group from Monitoring

Under the PRTG Group column, enter the Azure Resource group you wish to exclude (i.e. Compute, Network, Web etc.).

The configuration below will exclude the group named Batch and any Azure Resources (devices) and their sensors below that:

PRTG Group	DeviceType_Category	SensorName	ChannelsBlackList	TagName	TagValue
Batch	any	any	any	any	any

#### 10.2.4.2 Excluding an Azure Resource

Under the DeviceType\_Category column enter the type of the Azure Resource (device) you wish to exclude. Use the resource type located in parentheses (i.e. *VirtualMachines* if the Azure Resource name is *Windowstest\_(VirtualMachines)*), or part of the device name.

The configuration below will exclude Azure Resources of type Vaults under the group RecoveryServices

PRTG Group	DeviceType_Category	SensorName	ChannelsBlackList	TagName	TagValue
RecoveryServices	Vaults	any	any	any	any

#### 10.2.4.3 Excluding an Azure Sensor Type

In order to exclude specific Azure sensor types, specify the exact sensor type under the SensorName column. Entering “any” in this column will result in exclusion of all sensor types under this Azure Resource type (device). Which would effectively not add this Azure resource to PRTG.

The configuration below will not add sensors of type “Azure Service Health” for Azure resources (devices) of the type ServerFarms under the group Web.

PRTG Group	DeviceType_Category	SensorName	ChannelsBlackList	TagName	TagValue
Web	ServerFarms	Azure Service Health	any	any	any

#### 10.2.4.4 Excluding Azure Metrics (Channels)

In order to exclude specific Azure metrics (channels), add the full channel name under the Channels column separated by the sign “~”. Another option is to specify a single Azure Metric per line.

The configuration below excludes the “OS Disk Max Burst IOPS” and “Disk Read Bytes” metrics for all VirtualMachines.

PRTG Group	DeviceType_Category	SensorName	ChannelsBlackList	TagName	TagValue
Compute	VirtualMachines	Azure App Metrics	OS Disk Max Burst IOPS~Disk Read Bytes	any	any

Azure Tags Include/Exclude

#### 10.2.4.5 Excluding by Azure Tags

If you utilize Azure Tags in your Azure estate, you can leverage them by adding the Azure Tag under the column TagName and its value under TagValue in order to exclude any resources marked with this tag. Otherwise leave empty.

**Important** – Only Azure Tags that are assigned directly to Azure resources in the Azure Portal can be used for filtering. Azure Tags set on a Resource group level will be ignored.

The configuration below excludes all resources with Azure Tag and Value pair MonitorWithPRTG:FALSE

PRTG Group	DeviceType_Category	SensorName	ChannelsBlackList	TagName	TagValue
any	any	any	any	MonitorWithPRTG	FALSE

#### 10.2.4.6 Further Information and Troubleshooting of Exclusion

You may find that the **include** functionality is more suitable for your needs. Make sure to configure the file *include\_mon.csv*, in ways like explained in the paragraphs above. You would be able to add specific Groups of Azure Resources, Azure Resources and Azure Metrics to PRTG. Both exclude and include functions can be used in conjunction to tailor the automation for your needs.

#### **Important:**

1. Partial names may not be filtered correctly, except where stated.
2. The “Sensor Health” channel cannot be excluded
3. Make sure not to list the same sensors in both files – this might result in an unwanted behavior.
4. Excluding channels may result in gray channels, 0 value channels or sensors shown in error, this by design in PRTG. Please delete and re-add the sensors via the AutoMonX UI or the [PRTG monitoring automation CLI](#).

#### **Debugging the discovery process:**

For Quicker discovery you can apply the Include/Exclude policies to the discovery itself, as described in [section 7.15](#). Refer to the file “AutoMonX\_Discovery\_out.log” under the folder Azure/Logs. If a resource was filtered by either policy, you will see the lines:

Device: MyServer\_(SqlDatabases) filtered by blacklist

Device: MyServer\_(SqlDatabases) filtered by whitelist

### **Debugging addition to PRTG:**

Refer to the file “AutoMonX\_Automation\_Progress\_out.old.log” under the Common folder. In it you will see the complete PRTG Addition logs, where at the top of the file you will see the number of filters applied, and if the sensors passed or failed the include\_mon policy. Example:

Notice: 3 Filters are applied via C:/Program Files (x86)/PRTG Network Monitor/Custom Sensors/EXEXML/AutoMonX/Common/include\_mon.csv file

Found device: Dev:Name:AutoMonX\_LicStatus

dev:name:automonx\_licstatus - Removed by Whitelist

For each Group, Device and Sensor that pass the Whitelist, you will see if it was successfully added to PRTG or excluded. Example:

GroupName:Web - DeviceName:MyTest(Sites) - SensorName:Azure App Metrics - This Sensor would be skipped by the **exclude\_mon.csv policy at line:2**

Skipping Sensor Azure App Metrics by Blacklist exclusion

The file “PRTG\_Automation\_Results.csv” lists all the groups, devices and sensors that were not added to PRTG and the reason why.

### **Debugging Channel Exclusion:**

When debugging a sensor, you will see in the generated debug file the excluded channels. Read more about sensor debugging in [section 9.9](#).

In the folder “Common” you can find the file “exclude\_mon – Example.csv” with the examples provided previously.

#### ***10.2.4.7 Inclusion \ Exclusion validation check***

Checks the syntax and resolution logic of your **include/exclude** filter files and reports errors, warnings, and the interpreted (“human-readable”) rules.

Run the command:

```
AutoMonX_AzureCollector.exe -validate_filter_file
```

The file “Automation\_Syntax\_Check.log” contains a summary and a line-by-line interpretation of your rules.

## Typical issues

- Missing/extra columns, or header typos (e.g., “PRTG Group” must include the space).
- Non-CSV separators or unquoted commas inside fields.
- Using values other than Any to indicate wildcard.

Automation\_Syntax\_Check.log - Notepad

File Edit Format View Help

```
Line 2: Exclude resources where tag "Monitoring" = "Exclude".
Line 3: Include resources with device type/category "AutoMonX_LicStatus".
Line 4: Include resources in PRTG group "Reservations".
Line 5: Include resources in PRTG group "Disk".
Line 6: Include resources in PRTG group "Web", and with device type/category "StaticSites".
Line 7: Include resources in PRTG group "Synapse", and with device type/category "Synapse".
```

### 10.2.5 Automated Scan-Now Functionality

In some PRTG versions, we have noticed certain functionality issues for some types of custom sensors. The most frequently observed are the "Undefined lookup value" situations and sometimes “out of range” values (i.e. 10000% values). The solution we found for these issues was running a manual rescan of the sensor. Obviously, it is not a scalable solution for large environments, and this is the primary reason why this feature was introduced.

#### 10.2.5.1 Automatic scan-now functionality

To allow automatic sensor scanning, add the desired groups names to check in the "AutoMonX\_PRTG\_Automation.ini" file:

```
[RESCAN]
PRTG_GROUPS=AutoMonX_Azure
```

The feature is activated automatically by the Azure Sensor pack every 20 minutes and checks if new sensors were added. It will go over the PRTG groups configured in the INI file and look for sensors in Down or Warning states that have specific last message text values. Then, the feature will perform an automatic re-scan of these sensors in batches of 5 sensors per minute to avoid lags in the PRTG core.

#### 10.2.5.2 Manually activating the scan-now functionality:

Delete the file "sensors\_not\_to\_scan.txt" if exists.

```
AMX_PRTG_sensors_issues.exe -grouplist AutoMonX_Azure,"Azure test"
```

### 10.2.6 Automated Addition and Removal of Tenants

For customers who manage large number of tenants such as MSPs or CSPs, we have added a fully automated CLI support for adding and removing tenants.

#### 10.2.6.1 Adding a new tenant:

The command below creates a new Azure Tenant connection profile in the file "AzureConnProfiles.ini".

```
AutoMonX_AzureCollector.exe -create_conn_profile -tenant_label <display_label> -azure_appid <app_id> -azure_secretkey <secret_key> -azure_tenantid <tenant_id>
```

After adding a new Azure Tenant, run a configuration check so that the Azure Sensor Pack service will immediately pick-up the Azure API token for the new tenant, after which you can start the discovery process:

```
AutoMonX_AzureCollector.exe -config_check
```

#### 10.2.6.2 Deleting a tenant

The command below will delete the Tenant connection profile, log files and historical data related to the specified tenant. It does not delete the PRTG sensors and groups – This should be done prior, because this action will not delete the tenant if there are active sensors in it. **Use with utmost care!**

```
AutoMonX_AzureCollector.exe -remove_tenant -tenant 1
```

## 11 REST API Service

To fully automate the AutoMonX Sensor Pack for Azure, in version 4.2.14 and later, a new API service was introduced. The main functions of the REST API service are to initiate auto-discovery and activate the monitoring automation to add newly discovered Azure resources to PRTG. Additionally, adding and removing Tenants actions are also available.

The service can be installed via the Installer under additional services. For manual installation of the API Service use the following command:

```
AutoMonX_AzureCollector.exe -install_backend
```

The API service is listening on TCP Port 8075 only on the localhost IP address. The service will not accept any attempts to access it from outside the machine it was installed on.

To list all supported API options, use any web browser and access the following URL: <http://127.0.0.1:8075/api>

When running the PRTG Probe on an Azure Virtual Machine, you can activate these options remotely through the Azure Portal run-command feature. Example:

```
az vm run-command create --name "AutoMonX_Discovery" --vm-name "PRTG_Probe"
--resource-group "Test" --script "Invoke-RestMethod -Uri
'http://127.0.0.1:8075/api/discovery/start?tenant_id=1&outmode'"
```

### 11.1 Function get\_tenant\_id

Provides the Azure tenant internal id as written in the AzureProfiles.ini file. You need to provide either the tenant label or the full tenant id. Example: [http://127.0.0.1:8075/api/get\\_tenant\\_id?tenant\\_label=MyTenant](http://127.0.0.1:8075/api/get_tenant_id?tenant_label=MyTenant)

Param	Required	Description
<b>tenant_label</b>		Name of the tenant as saved in the file "AzureConnProfiles.ini".
<b>azure_tenantid</b>		Full ID of the tenant as appears in Azure Portal and saved in the file "AzureConnProfiles.ini".

#### Response on success:

The internal id of the tenant (i.e. 2)

#### Response on failure:

- No tenant with the name MyTenant
- Failed getting tenant id

- Missing tenant\_label or azure\_tenantid parameter

## 11.2 Function Initiate Discovery via API

Starts a discovery process for a tenant. Example:

[http://127.0.0.1:8075/api/discovery/start?tenant\\_id=1&outmode](http://127.0.0.1:8075/api/discovery/start?tenant_id=1&outmode)

Param	Required	Description
<b>tenant_id</b>	✓	The tenant id as returned in the previous API.
sub=subscription		The subscription name.
thorough		Run a thorough discovery, default behavior
outmode		When passed will generate a log file.
resources		Quick discovery of resources.
metrics		Quick discovery of metrics.
new_resources		Discovers only new resources
filtered		Filtered discovery according to the configured include_mon.csv and exclude_mon.csv files.
resource_id		Add the resource URL to PRTG comments
add_comments		Add the resource URL to PRTG comments for existing sensors

### Response on success:

Initiated: "C:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure\AutoMonX\_AzureCollector.exe" -discovery -tenant 1 -sub "-All" -outmode

### Response on failure:

Missing tenant\_id param

## 11.3 Function add\_tenant

This function adds a new tenant. Example:

[http://127.0.0.1:8075/api/add\\_tenant?tenant\\_label=NewTenant&azure\\_tenantid=xxxx&azure\\_appid=xxxx&azure\\_secretkey=xxxx](http://127.0.0.1:8075/api/add_tenant?tenant_label=NewTenant&azure_tenantid=xxxx&azure_appid=xxxx&azure_secretkey=xxxx)

Param	Required	Description
<b>azure_tenant_id</b>	✓	The full tenant id as appears in Azure Portal.
<b>tenant_label</b>	✓	The name of the tenant to be saved.
<b>azure_appid</b>	✓	The App ID.
<b>azure_secretkey</b>	✓	The Secret Key.
encrypt_secret		When passed will encrypt the secret key in the file.

### Response on success:

Initiated: "C:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure\AutoMonX\_AzureCollector.exe" - create\_conn\_profile -azure\_tenantid xxxx -tenant\_label "NewTenant" - azure\_appid xxxx -azure\_secretkey xxxx

**Response on failure:**

- Missing azure\_tenantid parameter
- Missing tenant\_label parameter
- Missing azure\_appid parameter
- Missing azure\_secretkey parameter

**11.4 Function remove\_tenant**

Removes all data related to a tenant. **Use with care!**

Example:

[http://127.0.0.1:8075/api/remove\\_tenant?tenant\\_id=2](http://127.0.0.1:8075/api/remove_tenant?tenant_id=2)

Param	Required	Description
<b>tenant_id</b>	✓	The internal tenant id

**Response on success:**

Tenant 4 was removed successfully.

**Response on failure:**

- Missing tenant\_id parameter
- Failed getting command status
- Error: Deletion Failed. Active sensors use this tenant

**11.5 Function to check Auto discovery status**

Gets the status of a specific discovery process. Example:

[http://127.0.0.1:8075/api/discovery/status?tenant\\_id=1](http://127.0.0.1:8075/api/discovery/status?tenant_id=1)

Param	Required	Description
<b>tenant_id</b>	✓	The internal tenant id
sub=subscription		The subscription name.

**Response on success:**

- Discovery is not running
- Discovery is still running

**Response on failure:**

- Missing tenant\_id parameter
- Failed getting discovery status

### 11.6 Function add\_to\_prtg

Initiates the monitoring automation that automatically adds groups, devices, and sensors in PRTG for a specific Azure tenant. Example:

[http://127.0.0.1:8075/api/add\\_to\\_prtg?tenant\\_id=1&passhash=xxxx](http://127.0.0.1:8075/api/add_to_prtg?tenant_id=1&passhash=xxxx)

Param	Required	Description
<b>tenant_id</b>	✓	The internal tenant id.
group_name		The PRTG group name to add the sensors to
passhash		The passhash of the saved user to connect to PRTG. You can also store it on the server.
file=file_name		The exact file name you want to add to PRTG, from the file under the Data directory.

#### Response on success:

Initiated add to PRTG automation for tenant 1

#### Response on failure:

Missing tenant\_id parameter

### 11.7 Function discover\_and\_add

Starts a discovery process for a tenant and after it is completed, initiates the automatic groups, devices, and sensor creation in PRTG for the tenant.

Example:

[http://127.0.0.1:8075/api/discover\\_and\\_add?tenant\\_id=1&passhash=xxxx&group\\_name=Azure\\_Sensors](http://127.0.0.1:8075/api/discover_and_add?tenant_id=1&passhash=xxxx&group_name=Azure_Sensors)

Param	Required	Description
<b>tenant_id</b>	✓	The internal tenant id.
sub=subscription		The subscription name.
outmode		When passed will generate a log file.
thorough		Run a thorough discovery, default behavior
resources		Quick discovery of resources.
metrics		Quick discovery of metrics.
quick		Quick discovery of both resources & metrics.
new_resources		Discovers only new resources
filtered		Filtered discovery according to the configured include_mon.csv and exclude_mon.csv files.
resource_id		Add the resource URL to PRTG comments
add_comments		Add the resource URL to PRTG comments for existing sensors
group_name		The PRTG group name to add the sensors to
passhash		The passhash of the saved user to connect to PRTG. You can also store it on the server.

Note – the options thorough, resources, metrics, quick and new\_resources are mutually exclusive.

**Response on success:**

Initiated discover and add to PRTG for tenant 1

**Response on failure:**

Missing tenant\_id parameter

**11.8 Function update\_key**

Updates the secret key for a specific tenant. To be used before the key is expired. Example:

[http://127.0.0.1:8075/api/update\\_key?tenant\\_label=OldTenant&azure\\_secretkey=XXXX](http://127.0.0.1:8075/api/update_key?tenant_label=OldTenant&azure_secretkey=XXXX)

Param	Required	Description
<b>tenant_label</b>	✓	The name of the tenant to be updated.
<b>azure_secretkey</b>	✓	The new secret key.
encrypt_secret		When passed will encrypt the secret key in the file.
<b>azure_tenantid</b>		Full ID of the tenant as appears in Azure Portal and saved in the file "AzureConnProfiles.ini".
<b>tenant_id</b>		The internal tenant id.

**Response on success:**

Initiated: "C:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure\AutoMonX\_AzureCollector.exe" - update\_key -tenant\_label "NewTenant" -azure\_secretkey xxxx

**Response on failure:**

- Missing tenant\_label, internal tenant id, or azure\_tenantid parameter
- Missing azure\_secretkey parameter

**11.9 Allowing remote connection to API server**

The API Server can receive requests from addresses configured in the file "AzureSensor\_ACL.ini":

```
UPDATE_WIN_FW=TRUE
ALLOWED_IP=8.8.8.8,8.8.4.4
```

Add all clients into the value of ALLOWED\_IP. A relevant Windows Firewall rule will be created to allow the connection and will be updated with every service restart if you add clients to the list.

If you wish to manually update the Windows Firewall, change the value of UPDATE\_WIN\_FW to FALSE.

### 11.10 Enabling HTTPS connection to the API Service

For Secure connection between the user endpoint and the AutoMonX Azure Backend Service, generate a certificate for the probe server and place it in: "AutoMonX\Azure\Certs"

The files must be named "certificate.crt" and "private.key".

Activate the ssl by adding this configuration to the file

"Automonx\_AzureSensor.ini":

API\_USE\_SSL=TRUE

The port 8075 will now only answer to https calls instead of http

Important – the HTTPS port in use is 8075

## 12 Monitoring the AutoMonX Sensor Pack for Azure Processes

To provide our customers with an effortless experience using the AutoMonX Sensor Pack, we added an additional service that with no additional overhead will monitor all the processes and restart the service when needed.

The service can be installed via the Installer under additional services. For manual installation of the API Service use the following command:

```
AutoMonX_AzureCollector.exe -install_backend
```

Configure the service using the following INI parameters:

**AZURE\_SELF\_MON\_SERVICE\_CHECK** – How often to run the full check in seconds. Default is 1 minute. If you wish to turn off this feature without stopping the API server, change this value to 0.

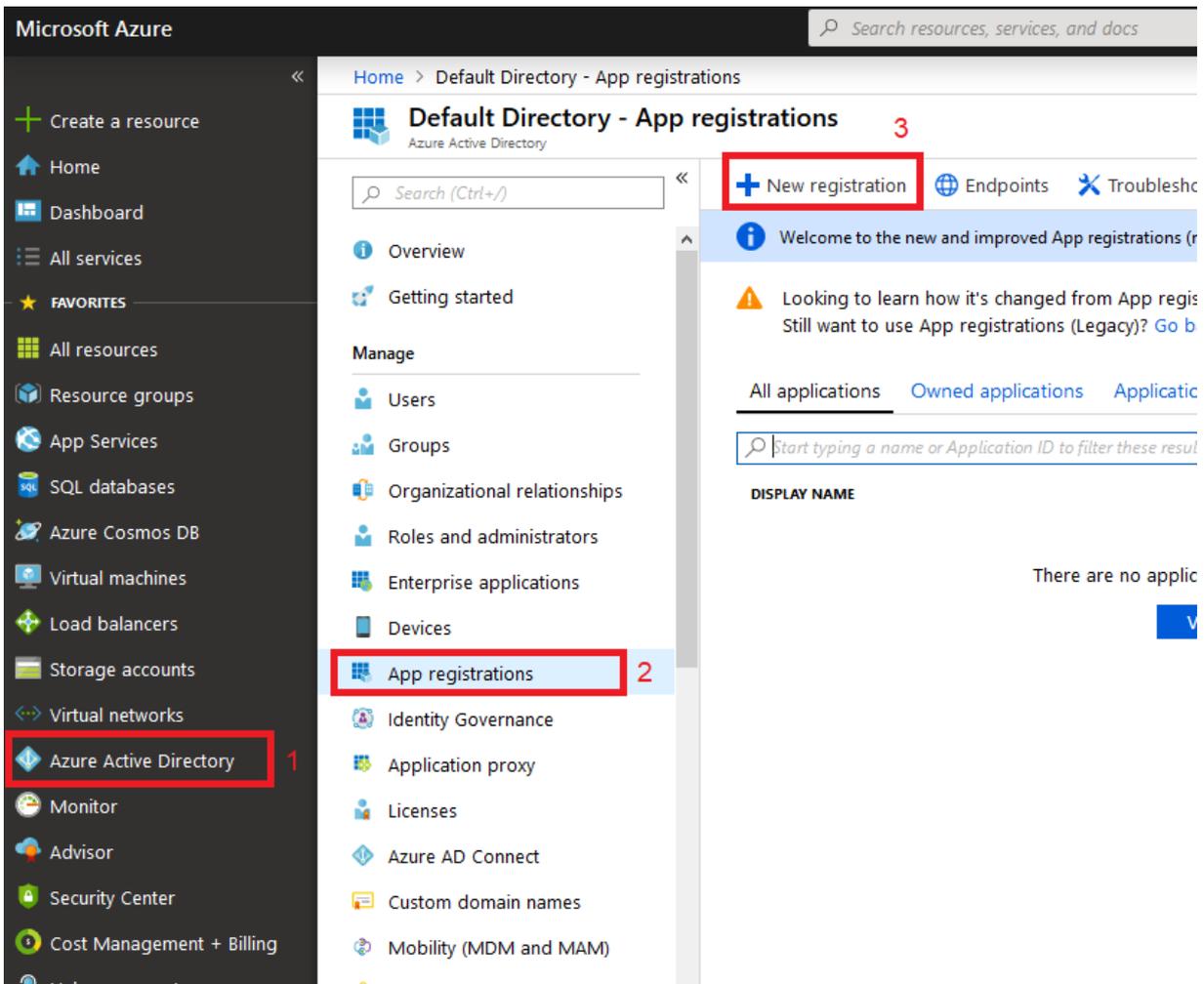
**AZURE\_SELF\_MON\_MEMORY\_LIMIT\_MB** – Limiting the AutoMonX Processes RAM usage. The default is 1000 MB. Please note that lowering this might cause frequent unwanted restarts to the service.

The Service will log all errors and restart times into the file “Backend\_Error\_Log.log”.

## 13 Appendix A - Obtaining the Azure Application and Tenant IDs

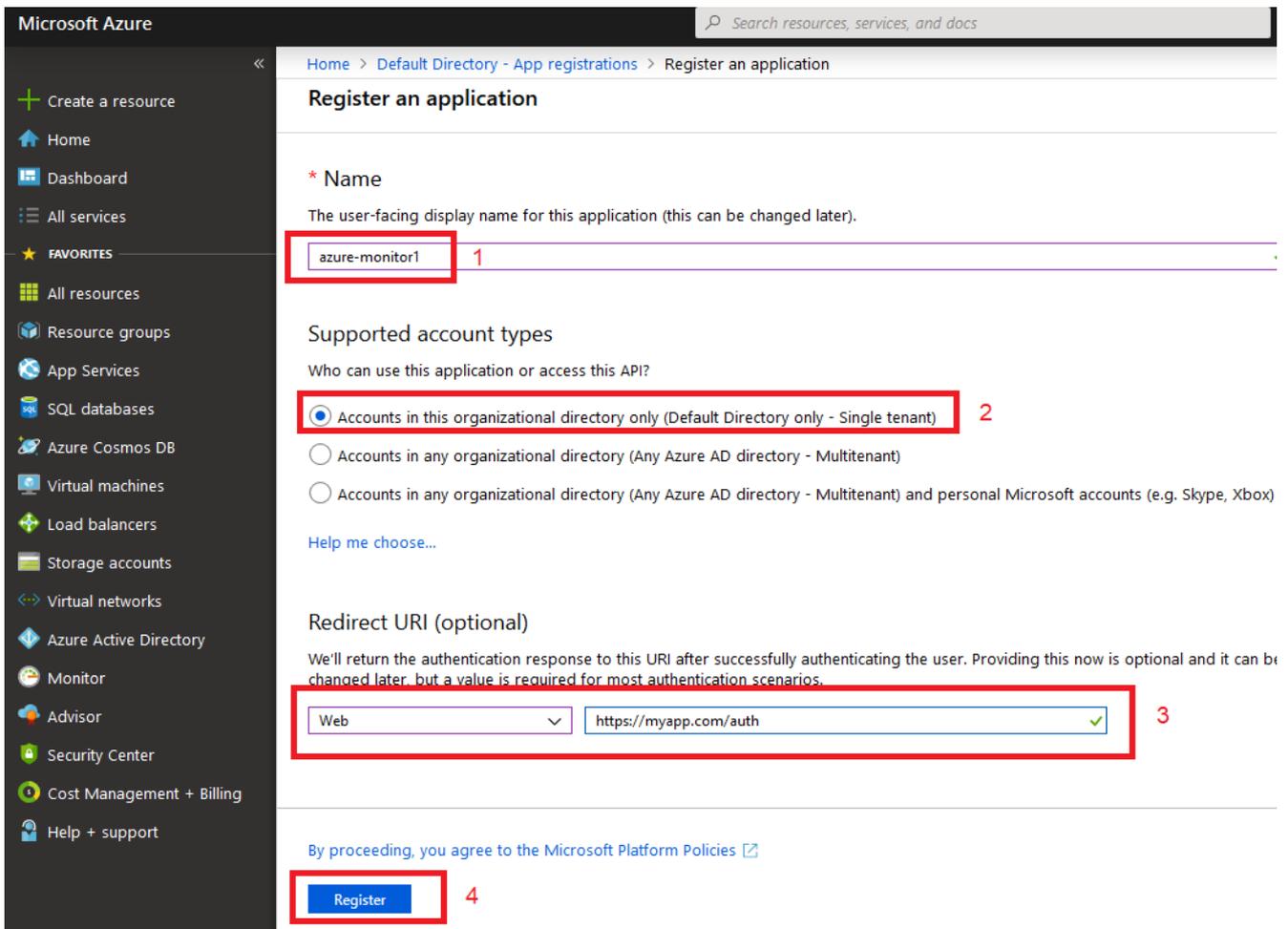
This section describes how to retrieve an Azure application ID required to monitor your Azure environment.

Select Azure Active Directory -> App registrations -> New registration



In the next windows:

- Choose an application name (for example PRTG\_app).
  - Select Accounts in this organizational directory only (Single tenant)
  - Select Web and fill and <https://myapp.com/auth>
- And Click Register



Microsoft Azure

Home > Default Directory - App registrations > Register an application

### Register an application

**\* Name**  
The user-facing display name for this application (this can be changed later).

azure-monitor1 1

**Supported account types**  
Who can use this application or access this API?

Accounts in this organizational directory only (Default Directory only - Single tenant) 2

Accounts in any organizational directory (Any Azure AD directory - Multitenant)

Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)

[Help me choose...](#)

**Redirect URI (optional)**  
We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

Web | https://myapp.com/auth 3

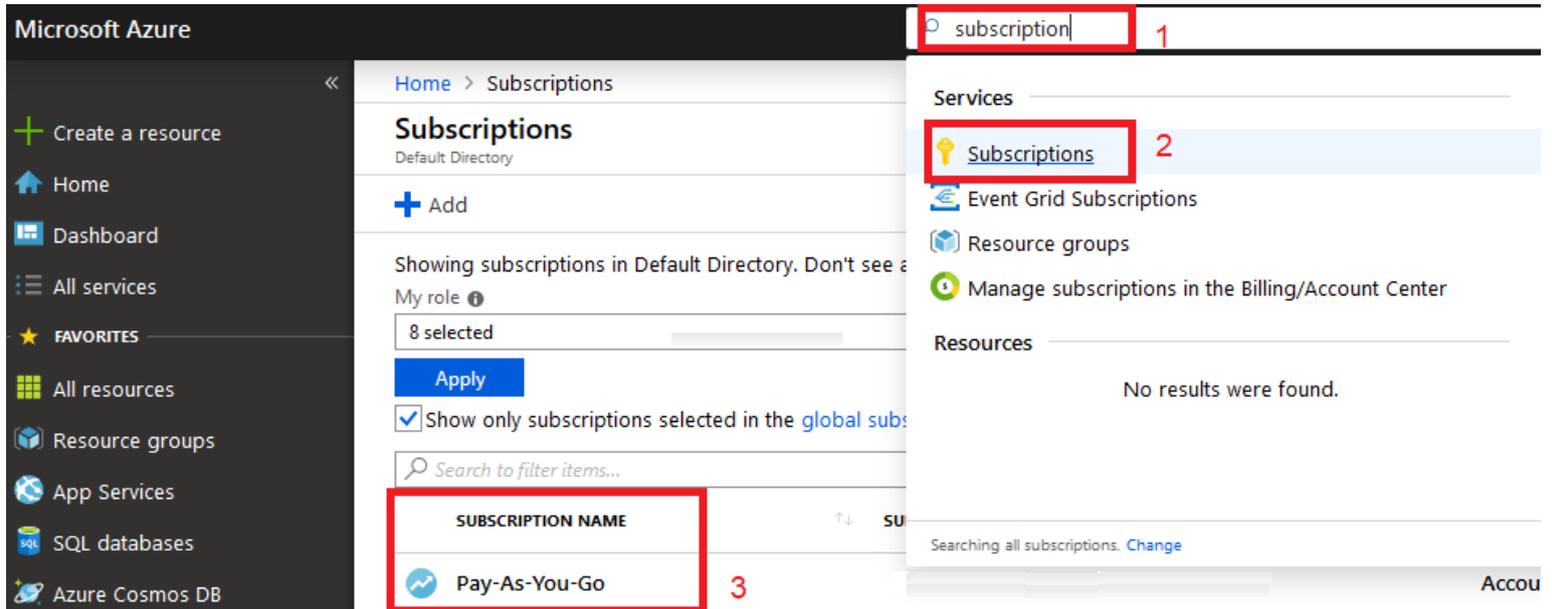
By proceeding, you agree to the [Microsoft Platform Policies](#)

Register 4

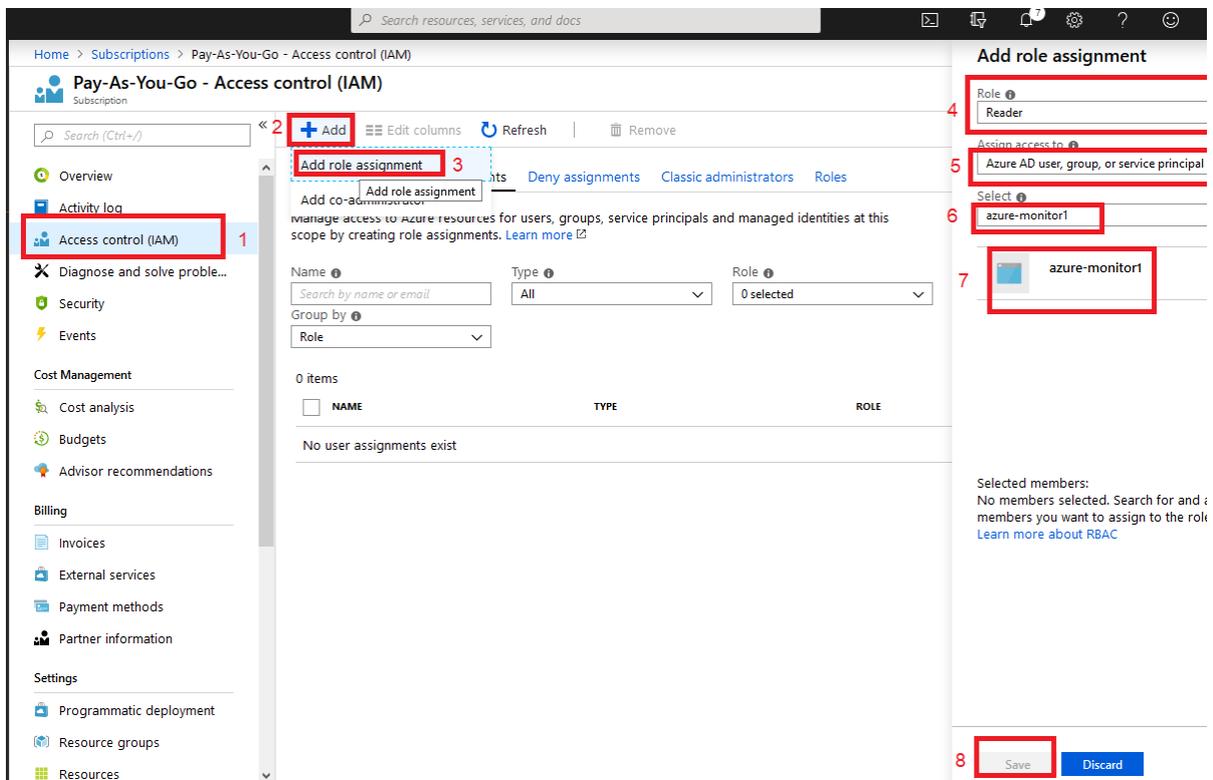
Next step: Assign subscription permissions to the newly created application ID.

Repeat this procedure for all the subscriptions you wish to discover and monitor:

- Write Subscriptions in the search box
- Click the Subscriptions Item
- Perform the grant permissions process for each subscription



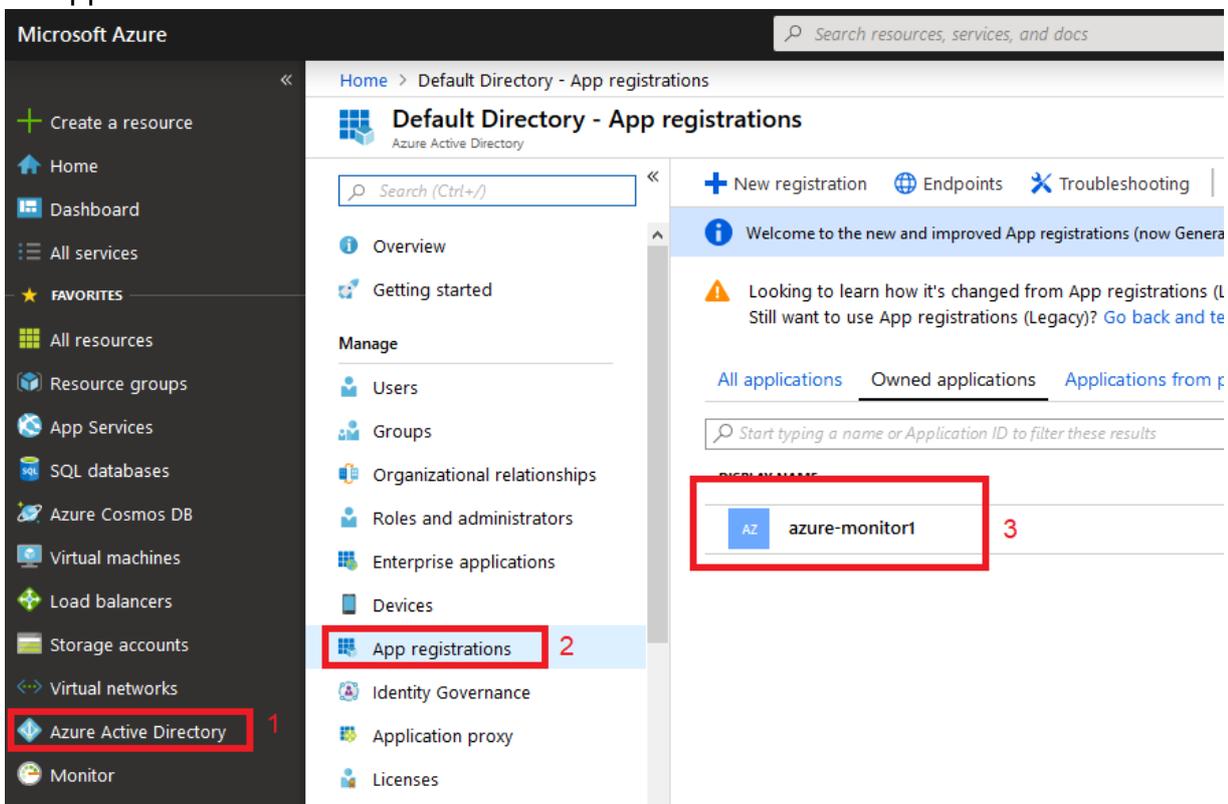
The process of granting permissions for accessing each subscription is shown below:



- Select Access control (IAM)
- Select Add button and select Add role assignment
- Select in the Add role assignment box the role of Reader.
- In the Assign Access to box the service principal option.
- Fill in the select box the name of Application ID connection
- Select the application id you created earlier
- Click the save button.

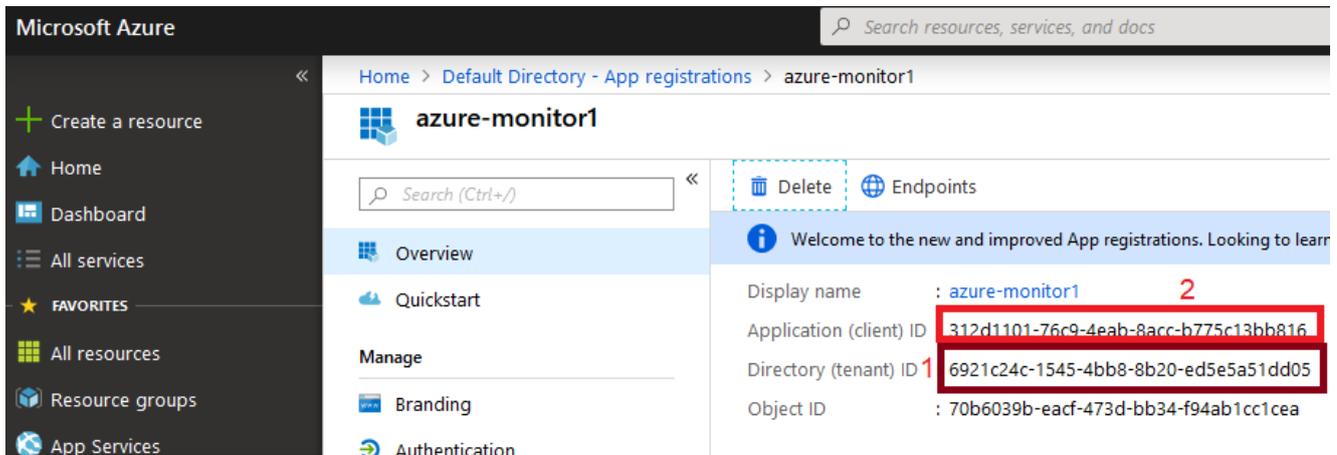
Now we can retrieve the Application ID and Tenant ID:

We go to Azure Active Directory -> App registrations -> Select the monitor application connection.



The screenshot shows the Microsoft Azure portal interface. On the left sidebar, 'Azure Active Directory' is highlighted with a red box and a '1'. In the main navigation pane, 'App registrations' is highlighted with a red box and a '2'. The main content area shows the 'Default Directory - App registrations' page. A search bar contains the text 'azure-monitor1'. Below the search bar, a table of application registrations is displayed, with the first entry 'AZ azure-monitor1' highlighted by a red box and a '3'.

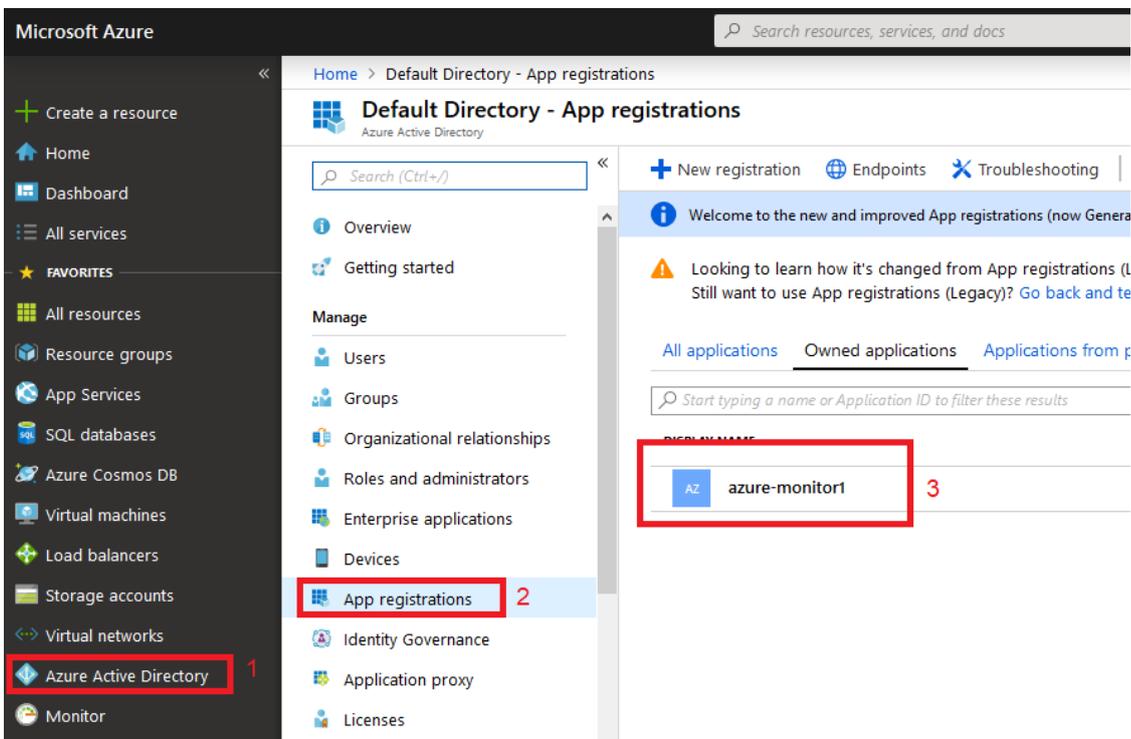
In the next page copy the Application ID and Tenant ID



### 13.1 Retrieving the Azure Secret Key

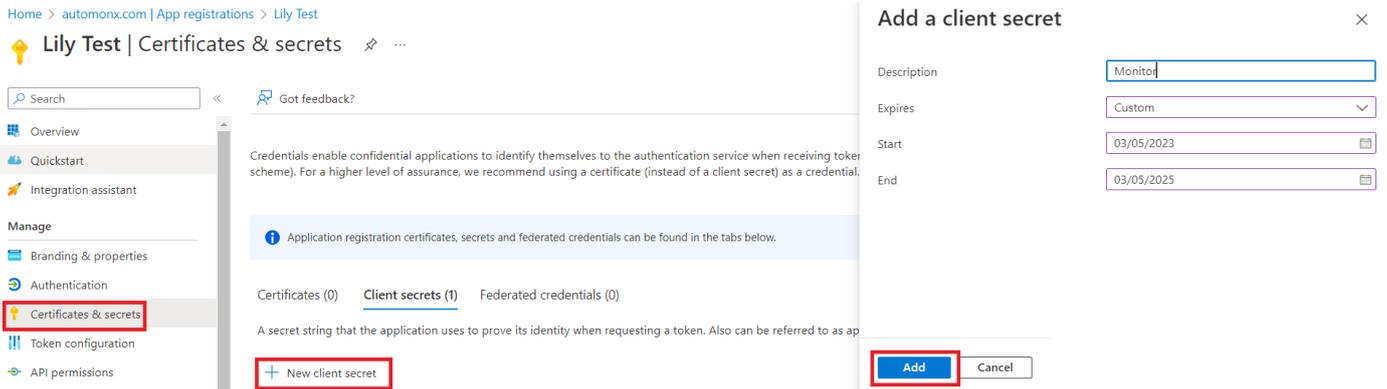
The next step is obtaining the Azure secret key:

Navigate to Azure Active Directory -> App registrations -> Select the monitor application connection.



In the next page perform the following:

- Click the Certificates & secrets menu
- Select under Client secrets the New client secret button
- Fill a in the Description
- Select when the token will expire. The longest expiry time is two years, therefore, monitoring this is important (explained in section 13).
- Select Add



In the next page copy the secret key because you won't be able to retrieve it afterwards.



**Client secrets**

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

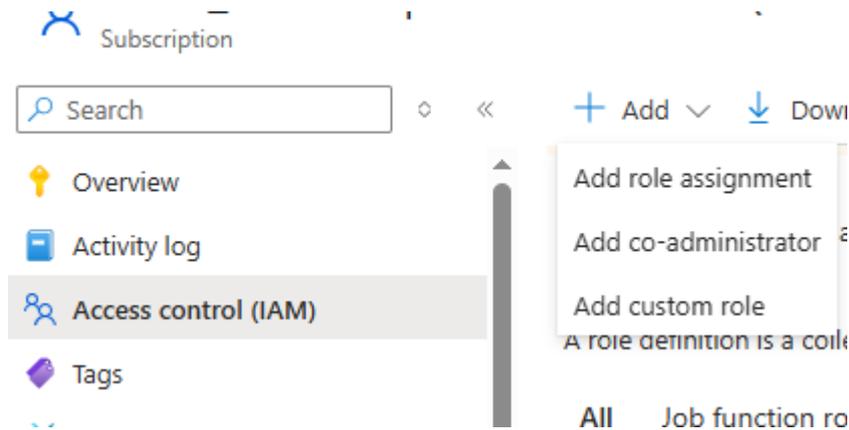
[+ New client secret](#)

DESCRIPTION	EXPIRES	VALUE
demo secret	5/14/2020	nWu9HVZ7Rnj.2y7XSkVyUngZ][x9Z:e

### 13.2 Creating Limited Monitoring Role

The Reader role permission has some security vulnerabilities. Follow these steps to create a more limited role to assign to your application.

Go to any subscription -> Access control (IAM) -> Add -> Add custom role



Save the following JSON into a file “Minimal Monitoring Reader.json”

```
{
  "properties": {
    "roleName": "Minimal Monitoring Reader",
    "description": "Grants least-privilege access for monitoring Azure resources, excluding VPN and sensitive infrastructure.",
    "assignableScopes": [
      "/subscriptions/<your-sub>"
    ],
    "permissions": [
      {
        "actions": [
          "Microsoft.Insights/metrics/read",
          "Microsoft.Insights/metricDefinitions/read",
          "Microsoft.ResourceHealth/availabilityStatuses/read",
          "Microsoft.Insights/logs/read",
          "Microsoft.OperationalInsights/workspaces/read",
          "Microsoft.OperationalInsights/workspaces/query/read",
          "Microsoft.Compute/virtualMachines/read",
          "Microsoft.Compute/virtualMachines/instanceView/read",
          "Microsoft.Network/networkInterfaces/read",
          "Microsoft.Resources/subscriptions/resourceGroups/read",
          "Microsoft.Resources/subscriptions/resources/read",
          "Microsoft.Insights/diagnosticSettings/read",
          "Microsoft.Insights/dataCollectionRules/read",
          "Microsoft.Authorization/policyAssignments/read",
          "Microsoft.ContainerService/managedClusters/read",
          "Microsoft.ContainerService/managedClusters/upgradeProfiles/read",
          "Microsoft.AAD/domainServices/read",

```



```
"Microsoft.Network/publicIPAddresses/read",
"Microsoft.Authorization/roleAssignments/read",
"Microsoft.CostManagement/query/action",
"Microsoft.Compute/disks/read",
"Microsoft.Compute/availabilitySets/read",
"Microsoft.Network/networkSecurityGroups/read",
"Microsoft.Network/virtualHubs/routeTables/read",
"Microsoft.Network/loadBalancers/read",
"Microsoft.Network/frontDoorWebApplicationFirewallPolicies/read",
"Microsoft.Network/trafficManagerProfiles/read",
"Microsoft.Network/applicationGateways/backendhealth/action",
"Microsoft.Web/serverfarms/read",
"Microsoft.Web/certificates/read",
"Microsoft.Web/sites/read",
"Microsoft.Web/sites/hostnameBindings/read",
"Microsoft.ResourceGraph/resources/read",
"Microsoft.Logic/workflows/runs/read",
"Microsoft.RecoveryServices/vaults/backupJobs/read",
"Microsoft.Sql/managedInstances/databases/read",
"Microsoft.DataFactory/dataFactories/datasets/activitywindows/read",
"Microsoft.DataFactory/factories/queryPipelineRuns/action",
"Microsoft.DesktopVirtualization/hostpools/sessionHosts/read",
"Microsoft.ContainerService/managedClusters/agentPools/read",
"Microsoft.ContainerService/locations/kubernetesVersions/read",
"Microsoft.ServiceBus/namespaces/queues/read",
"Microsoft.ServiceBus/namespaces/topics/subscriptions/read",
"Microsoft.Storage/storageAccounts/fileServices/shares/read",
"Microsoft.ResourceHealth/emergingIssues/read",
"Microsoft.Advisor/advisorScore/read",
"Microsoft.Security/alerts/read"
],
"notActions": [
  "Microsoft.Network/vpnGateways/*",
  "Microsoft.Network/connections/*",
  "Microsoft.Network/localNetworkGateways/*",
  "Microsoft.Network/virtualNetworkGateways/*",
  "Microsoft.KeyVault/vaults/*"
],
"dataActions": [],
"notDataActions": []
}
```

```
]
}
}
```

Select “Start from JSON” and upload the file “Minimal Monitoring Reader.json”

### Create a custom role ...

[Basics](#) [Permissions](#) [Assignable scopes](#) [JSON](#) [Review + create](#)

To create a custom role for Azure resources, fill out some basic information. [Learn more](#) 

Custom role name \*

Description

Baseline permissions  Clone a role  Start from scratch  Start from JSON

File \*  

The rest of the fields will be updated automatically.

### Create a custom role ...

[Basics](#) [Permissions](#) [Assignable scopes](#) [JSON](#) [Review + create](#)

To create a custom role for Azure resources, fill out some basic information. [Learn more](#) 

Custom role name \*  

Description

Baseline permissions  Clone a role  Start from scratch  Start from JSON

File \*  

In Assignable scopes, make sure to delete <your-sub> and instead Add assignable scopes of all the subscriptions you want you monitor with AutoMonX.

### Create a custom role ...

[Basics](#) [Permissions](#) [Assignable scopes](#) [JSON](#) [Review + create](#)

[+ Add assignable scopes](#)

Click Add assignable scopes to select the scopes (management groups, subscriptions, or resource groups) where this role will be available for assignment. Your role must have at least one assignable scope. [Learn more](#) 

Assignable scope	↑↓ Type	↑↓
/subscriptions/<your-sub>	Subscription	

Press Review + Create -> Create

Now follow the Add Role Assignment process to add the new role to the Application.

## Add role assignment ...

**Role** Members Conditions Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own cu

**Job function roles** Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

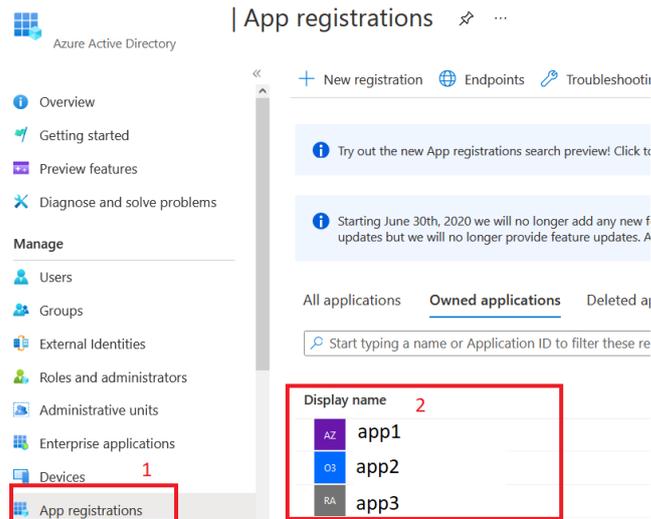
<input type="text" value="minimal monitoring"/>	×	Type : <b>All</b>	Category : <b>All</b>
Name ↑↓		Description ↑↓	
Minimal Monitoring Reader		Grants least-privilege access for monitor	

## 14 Appendix B - Adding permissions

### 14.1 Enabling App Secret Keys for monitoring

The App Secret monitoring requires additional permissions to run properly. The addition of permissions is done through the Azure Active Directory (Entra) panel.

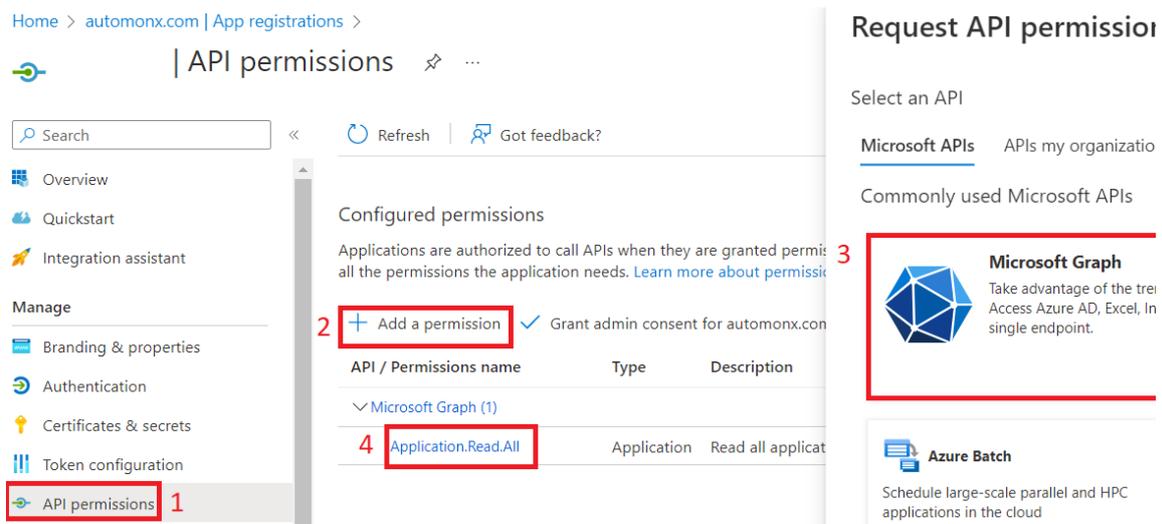
From the App registrations panel select the app that is used for the AutoMonX Azure access and perform the following:



Click API Permissions and click the Add a permission button. From there go to Microsoft Graph and add a permission from type Application named:

Application.Read.All

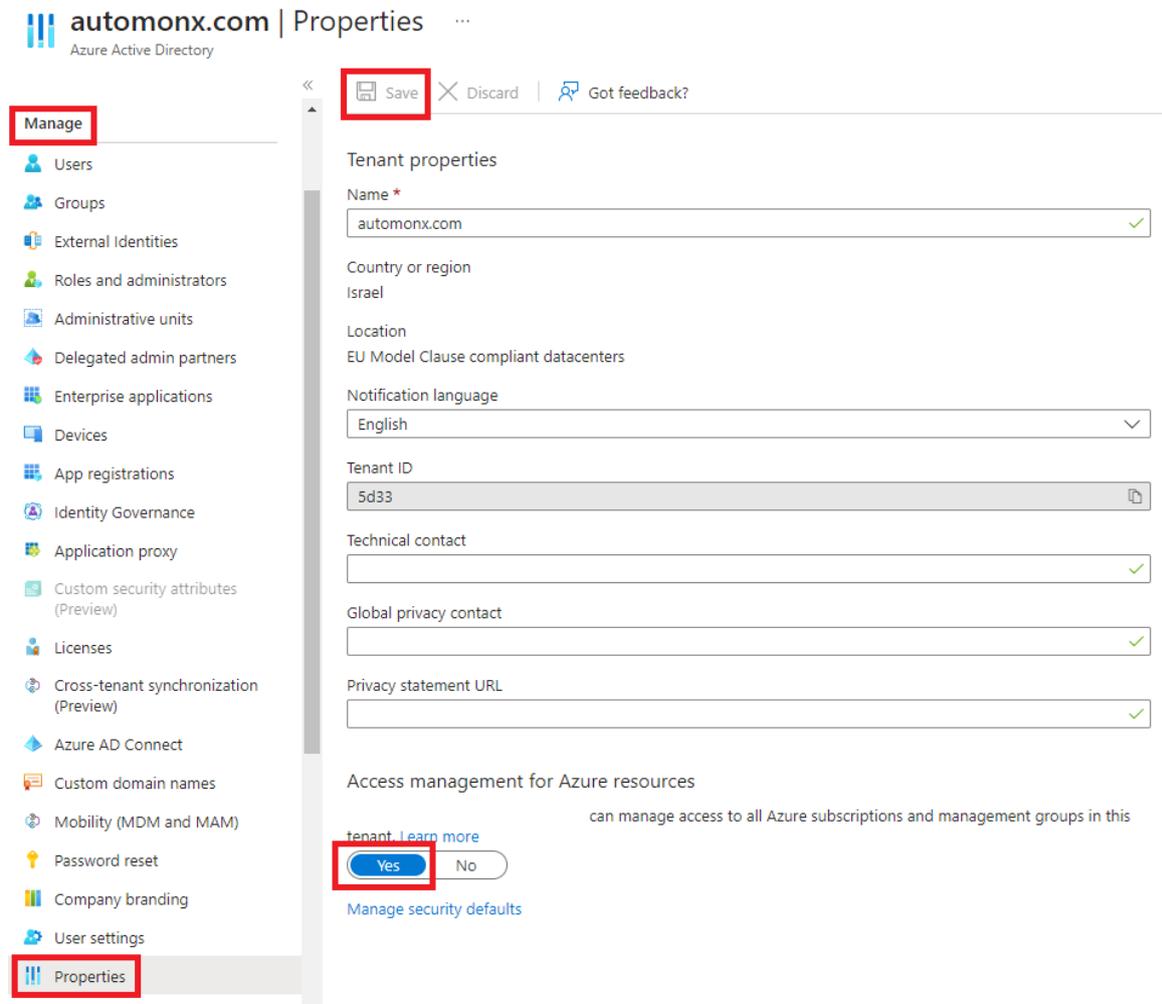
Make sure to give it Admin consent.



## 14.2 Enabling Reservations for monitoring

Make sure your user has role assignment permission:

Go to Azure Active Directory (Entra) and under Manage select Properties. Toggle Yes under Access management for Azure resources. Press Save.



The screenshot shows the 'Properties' page for 'automonx.com' in Azure Active Directory. The 'Manage' menu on the left is open, and 'Properties' is selected. The 'Save' button at the top is highlighted with a red box. The 'Access management for Azure resources' section is visible, with the 'Yes' button highlighted by a red box. The 'Name' field contains 'automonx.com', 'Country or region' is 'Israel', 'Location' is 'EU Model Clause compliant datacenters', 'Notification language' is 'English', and 'Tenant ID' is '5d33'. Other fields like 'Technical contact', 'Global privacy contact', and 'Privacy statement URL' have green checkmarks.

Now open Azure Cloud Shell  , then [Switch to PowerShell](#)

Find the App id using this command: `Get-AzADServicePrincipal -SearchString <app name>` (The name of the App used for Automonx monitoring)

```
PS /home/ > Get-AzADServicePrincipal -SearchString "Microsoft Azure PRTG"

DisplayName      Id                                     AppId
-----
Microsoft Azure PRTG 3e529dad-... 4aced4fa...
```

Copy the **Id** value (This is different from the App Object ID)

Run the following commands:

Import-Module Az.Accounts

Import-Module Az.Resources

New-AzRoleAssignment -Scope "/providers/Microsoft.Capacity" -PrincipalId <id>  
-RoleDefinitionName "Reservations Reader"

```
PS /home/lily> Import-Module Az.Accounts
PS /home/lily> Import-Module Az.Resources
PS /home/lily> New-AzRoleAssignment -Scope "/providers/Microsoft.Capacity" -PrincipalId 3e529dad-... -RoleDefinitionName "Reservations Reader"

RoleAssignmentName : 95aa1e54-
RoleAssignmentId    : /providers/Microsoft.Capacity/providers/Microsoft.Authorization/roleAssignments/95aa1e54-
Scope               : /providers/Microsoft.Capacity
DisplayName         : Microsoft Azure PRTG
SignInName          :
RoleDefinitionName  : Reservations Reader
RoleDefinitionId    : 582fc458-
ObjectId            : 3e529dad-
ObjectType          : ServicePrincipal
CanDelegate         : False
Description         :
ConditionVersion    :
Condition           :
```

Now the Azure Sensor Pack will be able to discover the Reservations.

**Note** – Archived reservations are not discovered.

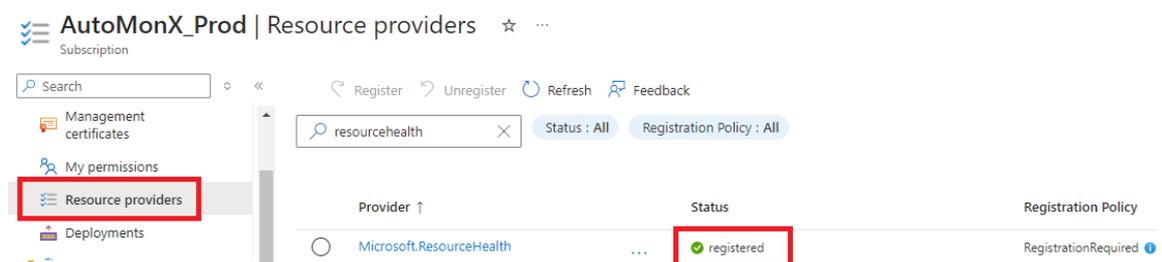
### 14.3 Registering Resource Providers

If you encounter the following error:

```
409: {"code":"AuthError","message":"An error ocured in authentication or authorization. Make sure the resource provider has been registered with this subscription"}
```

*The resource Automonx encountered a permissions issue.*

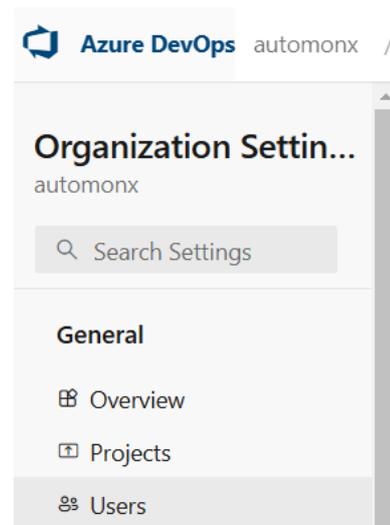
The following resource provider must be registered in each subscription:  
Microsoft.ResourceHealth



## 14.4 Enabling DevOps for monitoring

Go to: <https://dev.azure.com/>

Press on Organization Settings -> Users



Add users and add **the same** Service Principle you are using to monitor the rest of Azure with AutoMonX. Give the Service Principle Basic Access level and add it as a Project Reader to your projects:

### Add new users ×

Required fields are marked with an asterisk

Users or Service Principals \*

AT Automonx Test × +

Access level

Basic ▼

Add to projects

Automonx ▼

Azure DevOps Groups

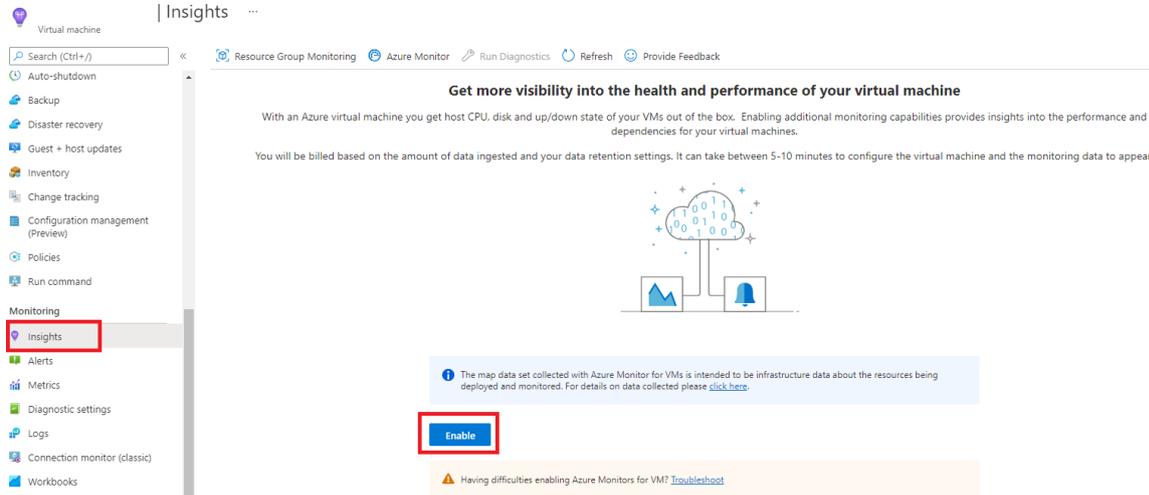
Project Readers ▼

Press Add.

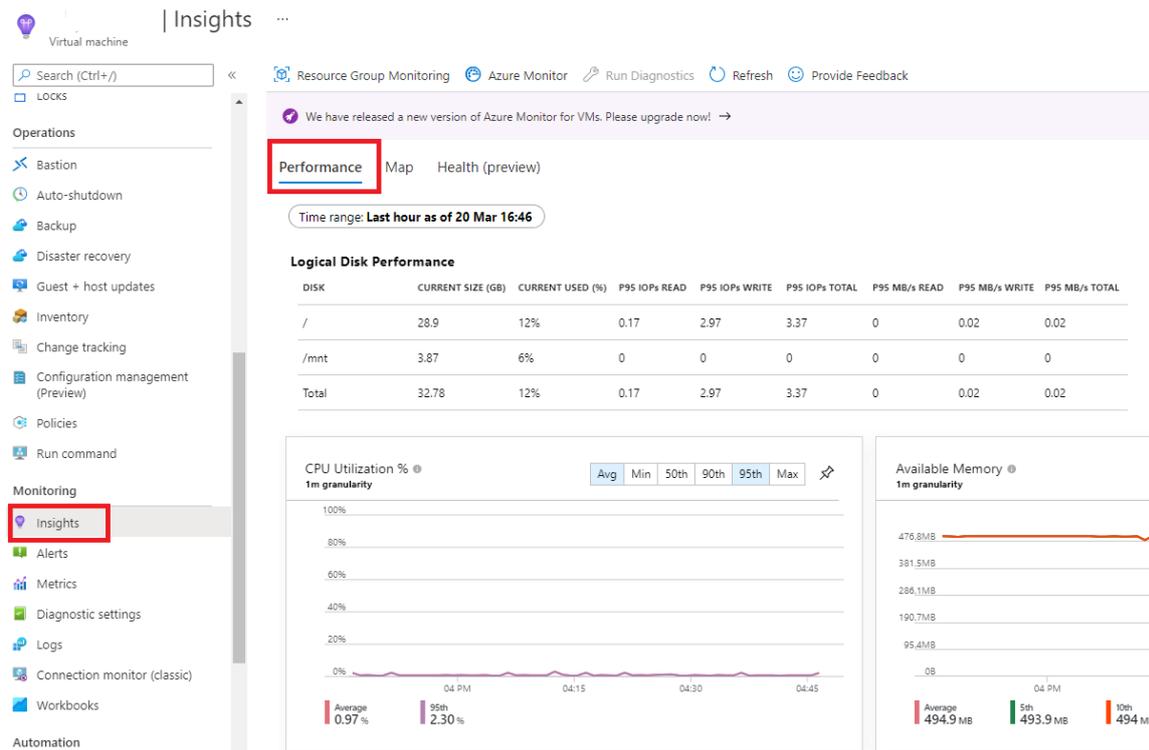
**Known limitation:** If you added the Service Principle a while ago but it is still not discovering your pipelines, we recommend you remove it from the organization (on the same screen as explained above) and re-add it. Run discovery again to see the sensor.

## 15 Appendix C - Activating Log Analytics on a VM

Go to the desired VM in the Azure portal and navigate to the Insights tab under the Monitoring section. Click “Enable” to activate this monitoring feature.



Give it a few minutes to collect data. Then it will look like this:



DISK	CURRENT SIZE (GB)	CURRENT USED (%)	P95 IOPs READ	P95 IOPs WRITE	P95 IOPs TOTAL	P95 MB/s READ	P95 MB/s WRITE	P95 MB/s TOTAL
/	28.9	12%	0.17	2.97	3.37	0	0.02	0.02
/mnt	3.87	6%	0	0	0	0	0	0
Total	32.78	12%	0.17	2.97	3.37	0	0.02	0.02

Now you can re-run the discovery process and pick the “Azure VM Multi-Disk” sensor when it is complete.

## 16 Appendix D – Creating Custom KQL Log Analytics Sensors

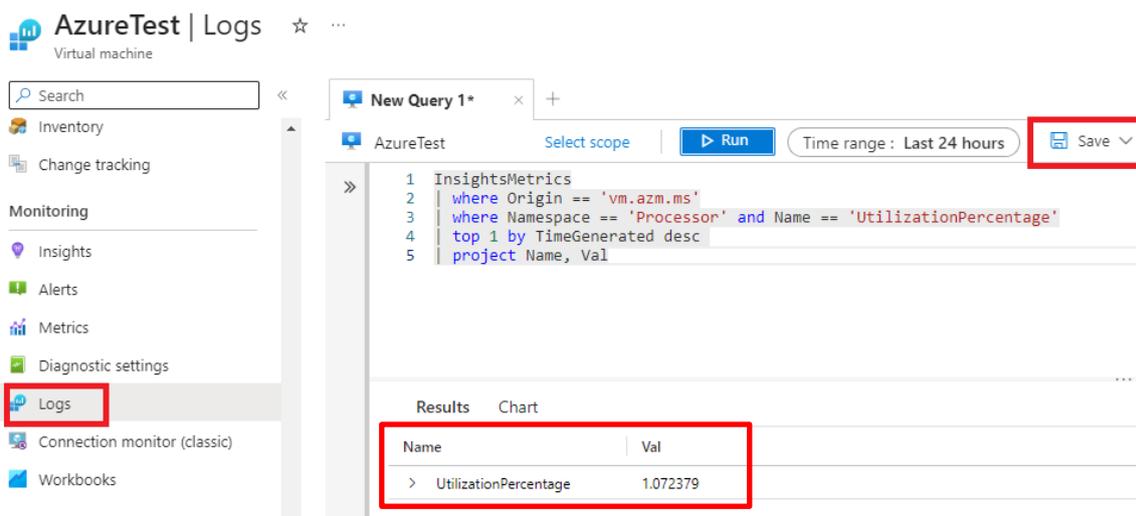
The Azure Sensor pack allows customers to create custom KQL Query sensors based on Azure KQL log analytics query mechanism. Below is a quick summary of how to create such sensors:

1. Create the query in Azure portal and make sure it returns values in a specific format.
2. Save the query into a QueryPack
3. Add the Azure resource to perform the query on, in the Query description field.
4. Run the AutoMonX auto-discovery to discover the KQL sensors and make them available to be added to monitoring.

### Creating the query

Go to the resource (for example, Virtual Machine). Under Monitoring select Logs and write your own query. Query must include UTF-8 characters only. Adjust it so that the returned values will be one of the following formats:

1. Single value – Relevant for state or count queries. The channel name will be “Value”. The value must be numeric.
2. Channel name and value – Make sure that the first column will always return the same name for the channel name. Limited to 40 results. The value must be numeric.
3. Channel name, value, and unit – the third column can represent the custom unit to show in PRTG. Limitations as listed above.



AzureTest | Logs ☆ ...  
Virtual machine

Search

Inventory

Change tracking

Monitoring

Insights

Alerts

Metrics

Diagnostic settings

Logs

Connection monitor (classic)

Workbooks

New Query 1\* x +

AzureTest Select scope Run Time range: Last 24 hours Save

```

1 InsightsMetrics
2 | where Origin == 'vm.azm.ms'
3 | where Namespace == 'Processor' and Name == 'UtilizationPercentage'
4 | top 1 by TimeGenerated desc
5 | project Name, Val

```

Results Chart

Name	Val
> UtilizationPercentage	1.072379

In this example the query is:

InsightsMetrics | where Origin == 'vm.azm.ms' | where Namespace == 'Processor' and Name == 'UtilizationPercentage' | top 1 by TimeGenerated desc | project Name, Val

## Saving the Query

After you reached the desired output, press Save to Save as query. In the popped-out window, name the query – this will be the name of the sensor. The most important part is the **Description** field. Here you must specify the specific resource to perform this query on. This must include the full Resource ID, you can find in under the Configuration -> Properties tab of the resource.



The screenshot shows the Azure portal interface for a virtual machine named 'AzureTest'. The 'Properties' tab is selected and highlighted with a red box. The left-hand navigation pane lists various categories: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Connect, Networking, Settings, Disks, Extensions + applications, Configuration, and Advisor recommendations. The main content area displays the following properties:

- Hibernation: Disabled
- Ephemeral OS disk: N/A
- Azure Spot eviction policy: N/A
- Azure Spot eviction type: N/A
- Host group: None
- Host: -
- Proximity placement group: -
- Resource ID: /subscriptions/[redacted]/resourceGroups/DefaultResourceGroup-EUS2/providers/Microsoft.Compute/virtualMachines/AzureTest
- Location: East US 2
- Availability zone: 3

## Save as query ✕

Query name \*

CPUutilization ✓

Description

7dcde0db3272/resourceGroups/prod\_rg/  
providers/Microsoft.Compute/virtualMac  
hines/AzureTestVM

Path

Save to the default query pack ⓘ

Tags

Resource type

Virtual machines ∨

Category

0 selected ∨

Label

0 selected ∨

[Create new label](#)

Save

Cancel

If you want to save the query pack under a non-default name (DefaultQueryPack), make sure to create the query pack first:

[Home](#) > [Log Analytics query packs](#) >

## Create Log Analytics query pack ...

**Basics** Tags Review + create

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="Azure subscription 1"/>
Resource group *	<input type="text" value="AutoMonX_Test_Servers"/> <a href="#">Create new</a>

### Instance details

Name *	<input type="text" value="QueriesForAutomonX"/>
Region *	<input type="text" value="East US"/>

And on the Save as query tab:

### Path

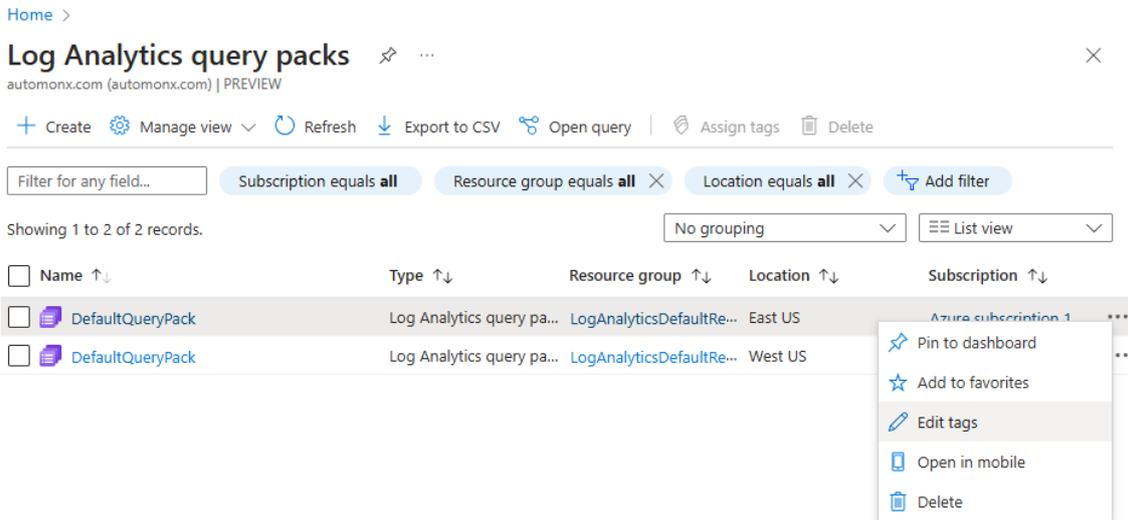
Save to the default query pack

Subscription *	<input type="text" value="Azure subscription 1"/>
Resource group *	<input type="text" value="AutoMonX_Test_Servers"/>
Log Analytics query pack *	<input type="text" value="QueriesForAutomonX"/>

## Discovering the KQL Custom sensor

The KQL custom sensors will be automatically discovered like any other AutoMonX sensor during the auto-discovery process. It will be placed under the resource's device (given the resource id was provided in the description).

**Important:** If you are using the Azure Sensor pack whitelist mechanism to filter out resources, make sure that the relevant resource (i.e. VM) that the KQL Sensor query refers to is not filtered out. Also make sure to add relevant Azure tags to the Query Pack in case they are used by our whitelist mechanism.



### Resource ID – Must be added to the Description Field

If you have forgotten to add a description in one of the previous steps, you must add it to the KQL query using the following format (the comma at the end is important): **"description": "<your\_resource\_id>,<your\_resource\_id>"**

**Note:** There is a limit of 148 characters in the Description field. You can bypass the limit by editing the Description field in the KQL query itself (not via the Azure form) as explained in the next paragraph

### Modifying the KQL query

To adjust or add new resources to the Query description, open the QueryPack you saved the query into, press on the Export templates tab, and press Deploy.

DefaultQueryPack | Export template ☆ ...

microsoft.operationalinsights/querypacks

Search << Download Add to library **Deploy** Visualize template

Overview  
Activity log  
Access control (IAM)  
Settings  
Locks  
Automation  
CLI / PS  
Tasks (preview)  
**Export template**  
Help  
Support + Troubleshooting

To export related resources, select the resources from the Resource Group view then select the "Export template" option from the tool bar.

Include parameters

Template Parameters Scripts

```

14 {
15   "type": "microsoft.operationalinsights/querypacks/queries",
16   "apiVersion": "2019-09-01",
17   "name": "DefaultQueryPack/41f9a30e-3158-4e9f-ba93-992accdec8d0",
18   "dependsOn": [
19     "[resourceId('microsoft.operationalinsights/querypacks', 'DefaultQueryPack')]"
20   ],
21   "properties": {
22     "displayName": "CPUUtilization",
23     "body": "InsightsMetrics| where Origin == 'vm.azm.ms'| where Namespace == 'Proc
desc | project Name, Val",
24     "related": {
25       "categories": [],
26       "resourceTypes": [
27         "microsoft.compute/virtualmachines"
28       ]
29     }

```

Now press Edit template:

## Custom deployment

Deploy from a custom template

**New!** Deployment Stacks let you manage the lifecycle of your deployments. Try it now →

Basics **Review + create**

### Template



Custom template [↗](#)  
3 resources



### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Resource group \* ⓘ

[Create new](#)

### Instance details

Region \* ⓘ

Querypacks\_Default Query Pack\_name  ✓

Virtual Machines\_Azure Test\_externalid  ✓

Virtual Machines\_Azure Test VM\_externalid  ✓

To add several resources, separate them with a comma.

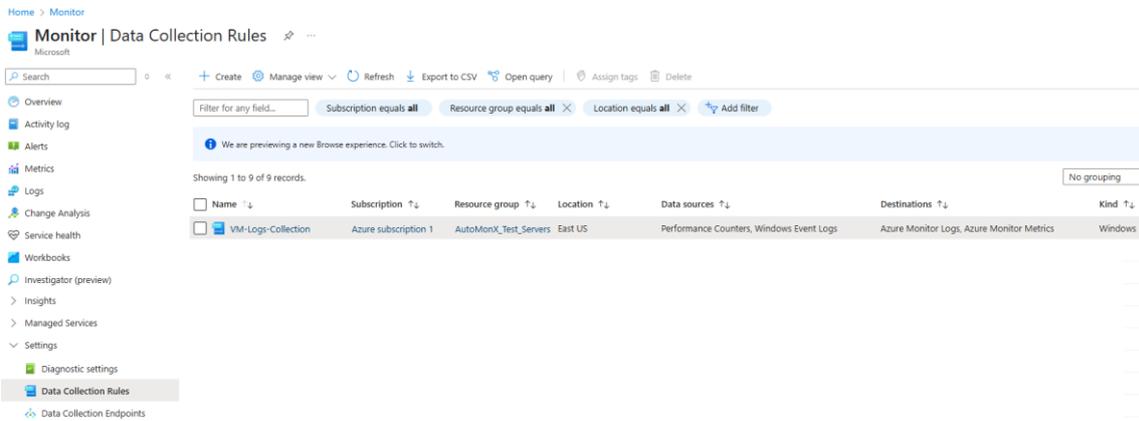
```
    "properties": {  
      "displayName": "CPU util",  
      "description": "/subscriptions/xxx/resourceGroups/CONTROL_GROUP/providers/Microsoft.Compute/virtualMachines/ContoLVMH/subscriptions/xxx/resourceGroups/prod_rg/providers/Microsoft.  
Compute/virtualMachines/AzureTestVM",  
      "body": "InsightsMetrics | where Origin == 'vm.azure.ms' | where Namespace == 'Processor' and \\r\\nName == 'UtilizationPercentage' | top 1 by TimeGenerated desc | project Name, Val ",  
    }
```

After you finish editing the query body or description, press **Save** and continue to **Review+Create** the template. The KQL query of the sensor will be updated automatically, but to create the sensors for new resources the discovery must be run again.

## 17 Appendix E – Monitoring VM Active Sessions

Note – this only works for Azure hosted VMs

Create a Data Collection Rule to start collecting your vm data.



Edit the default Performance Counters, and add the custom counter:

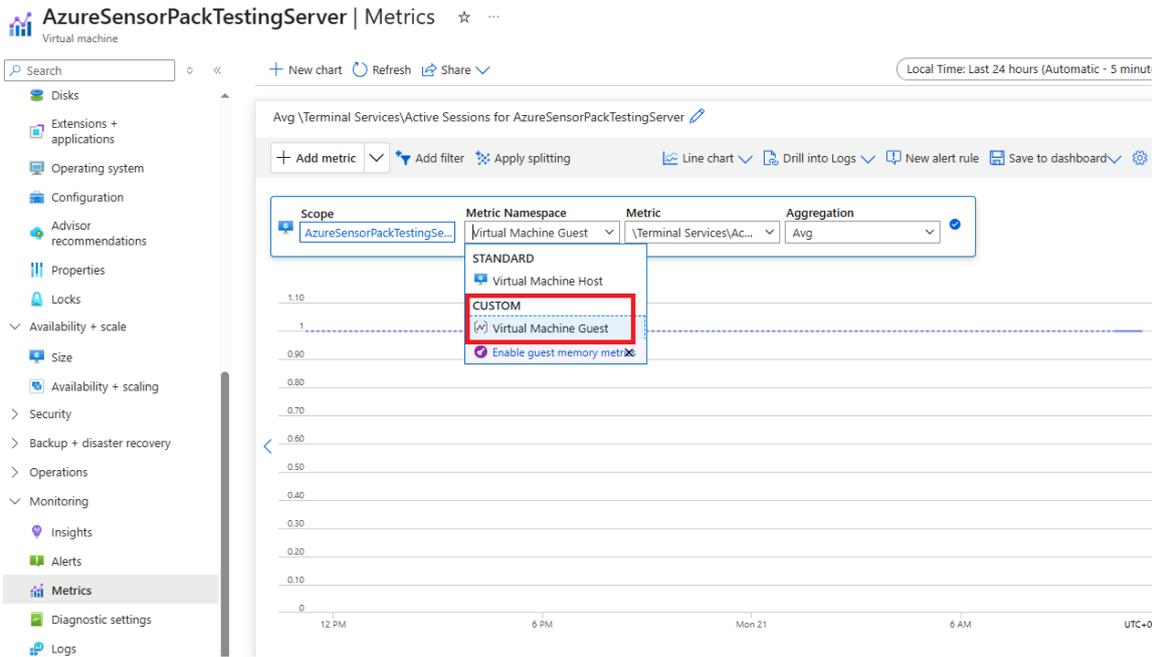
`\Terminal Services\Active Sessions`



Press **Add** and **Save**

Make sure you added a Log Analytics Workspace as a destination, and added the VMs you want to monitor in the Resources tab.

After a few minutes check if the data is now collecting (make sure to switch the metric namespace):



Now run discovery and a new sensor type will appear in the VM device.

## 18 Appendix E – Troubleshooting Key Vault Issues

If you experience error 403 when performing discovery via the log at "C:\Program Files (x86)\PRTG Network Monitor\Custom Sensors\EXEXML\AutoMonX\Azure\Logs\ Automonx\_Discovery\_out.log" Usually this is the network configuration of the Key Vault (most of the time the Azure API would also return the exact reason):

When integrating the AutoMonX Azure Sensor Pack with Azure Key Vault, network access is just as important as permissions. If the Key Vault is restricted at the network level, the collector must be able to reach it accordingly.

### 1. Check the Key Vault networking configuration

1. In the **Azure Portal**, open **Key Vaults**
2. Select the relevant **Key Vault**
3. Go to **Networking**

Review the following settings:

- **Allow access from**
  - **Allow public access from all networks**  
→ The Key Vault is reachable over the public Azure endpoint
  - **Disable public access**  
→ The Key Vault is private-endpoint-only
- **Private endpoint connections**
  - If one or more private endpoints enabled, the Key Vault can only be accessed from within the connected VNets (or peered networks) and thus if the Azure Sensor Pack is not located inside the VNET it would not have access to the Key Vault.

### How to enable public access and restrict by IP for Azure Sensor Pack installations outside the VNets

If Private Endpoints are not mandatory, the Key Vault can be accessed via the public endpoint while limiting access to specific IP addresses. This allows the collector to connect without VNet placement.

Steps:

1. Under **Allow access from**, select **Allow public access from specific virtual networks and IP addresses**
2. Under **Firewall**, add the **public egress IP address** of the AutoMonX Azure Sensor Pack (or collector host)
3. Keep the required **Key Vault RBAC roles** assigned to the service principal

**Note:** The IP address must be the public IP of the server used to reach Azure. Internal or private IP addresses are not sufficient.

If the Key Vault is configured as private-endpoint-only, this IP-based approach is not supported and the collector must have network connectivity to the VNet hosting the private endpoint.

If you are still experiencing issues, make sure that you assigned the RBAC permissions correctly using the guide below:

How to set the correct RBAC Roles to the monitoring Service Principal (for key vaults) :

1. Go to key vault
2. Click on Access control (IAM)
3. Under Grant access to this resource click on "Add role assignment"
4. You would need to apply these permissions (one by one as the Azure Portal does not let you pick them all at once):
  - a. Key Vault Certificate User
  - b. Key Vault Crypto User
5. Click on "User, group, or service principal" (the top radio button)
6. Then click "+ Select members"
7. Search in the list for the Service Principal you are using
8. Select the Service Principal you are using
9. Click Review + assign and you should have now permissions to monitor certificates and keys from the Key Vault.