AutoMonX
The Monitoring Automation Company

AutoMonX Sensor Pack for AWS

Web : http://www.automonx.com
Email : support@automonx.com

Date	Change	Author
TBD	Initial Release 1.0	AutoMonX
17.11.2025	Added new ACM, ABV, SES, TGW, GuardDuty, WindowsBackup, FSx Sensors	
18.11.2025	Fixed existing sensors appearance and error messages	AutoMonX
23.11.2025	Enhanced day channel calculations and added threshold limits for all sensors and OVL status mappings for SES, TGW sensors	AutoMonX
11.12.2025	Added: APIGateway, Autoscaling,Elasticache,ElasticBeanstalk, Redshift, Route53, CloudFront, Bedrock, WAF, FSx new sensors and metrics for VMs and Volume Fixed: EMR sensors discovery, PRTG OVL lookup errors New guide sections: Open Central Account and configuring AWS Trusted Access Change ABV sensor name to Backup with more backup metrics, adding more certificate metrics to existing ACM sensors	AutoMonX
14.12.2025	Improved comments and logs for collector and API. Fixed SES, apigateway, WAF sensors errors and uploading to PRTG.	AutoMonX

Web : http://www.automonx.com
Email : support@automonx.com

AutoMonX
The Monitoring Automation Company

Table of Contents

Web : http://www.automonx.com
Email : support@automonx.com

Web : http://www.automonx.com
Email : support@automonx.com

Web : http://www.automonx.com
Email : support@automonx.com

1 Purpose

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

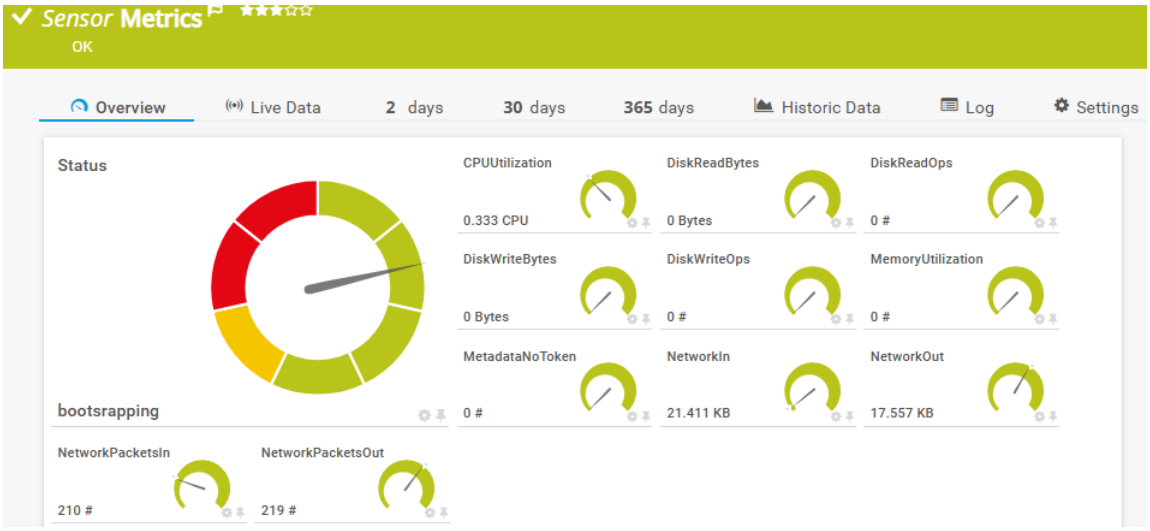
2 AWS Sensor Pack Overview

AutoMonX Ltd has developed the AWS Sensor pack aimed at monitoring AWS cloud environments for IT teams and service providers (MSPs and CSPs). The AWS Sensor pack can discover and monitor AWS resources located across multiple AWS accounts. These unique sensors are monitoring the various aspects of AWS's resources and services and have a tight integration with PRTG. The AWS Sensor pack currently supports auto-discovery and monitoring of several AWS resource types as seen below:

- Alarm
- ACM (Certificate Manager)
- Backup
- FSx
- API Gateway
- WAF (Web Application Firewall)
- SES (Simple Email Service)
- Guard Duty
- Autoscaling
- ElasticCache
- CloudFront
- Elastic Beanstalk
- Route53
- Redshift
- Windows Backup
- EC2
- S3
- ECS
- EBS (EC2 volumes)
- DynamoDB (including Aurora)
- RDS
- Glacier
- ELB (Load balancers)
- EMR
- Lambda
- KMS (Key management service)
- SNS (Topics)
- VPC (Virtual private network)
- EFS
- SQS (Queues)
- Cost Explorer

3 How Does It Work?

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



3.1 The AWS Sensor Pack Architecture

The AutoMonX AWS Sensor Pack application server sends multiple requests to the AWS 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 AWS Management API to provide reliable, swift, and lightweight performance. The AWS sensor pack is highly scalable and flexible and thus can monitor thousands of resources spread across multiple AWS Organizations, accounts and regions.

3.2 The AWS Sensor Pack – Integration with PRTG

The AutoMonX AWS Sensor Pack is tightly integrated with PRTG via the sensor type - the HTTP Advanced sensor. Such integration is automatically created by the Monitoring Automation feature of the AWS Sensor pack that pushes all the required configuration settings into PRTG.

3.3 The AWS Sensor Pack – Integration with Influx DB (future)

The AutoMonX AWS Sensor Pack is tightly integrated with InfluxDB via HTTPS, it creates a bucket for the information and automatically transfers sensor data to it. The AWS Sensor pack pushed all required data and configuration automatically to the InfluxDB server.

3.4 The AWS Sensor Pack – Essential Terminology

The AWS Sensor pack automatically adapts PRTG web interface for displaying the AWS. Below are some essential terms that are used through this deployment guide:

Group – PRTG group of devices. The AWS Sensor pack monitoring automation automatically organizes the AWS resources it discovers by creating groups of AWS resources (EC2, SQS, S3 etc) Read more about the automatically created hierarchy in section 6.4.

Device – Each AWS resource is represented in PRTG as a device (for example a Virtual Machine, an SQL database, or a Queue).

Sensor – Created under every PRTG device. The main sensors that are available for most AWS resource types are AWS App Metrics and AWS Service Status. Additional sensor types are available, read more about it in section 7.3.

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

4 Getting Started with AWS Sensor Pack

4.1 Supported Software versions

The AWS 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, AWS VM, Azure VM	
PRTG Core and Probe deployments	19.x, 20.x, 21.x, 22.x, 23.x	All On-Prem PRTG license types supported
PRTG Hosted (Cloud)	Any	Need to upload the custom OVLs

4.2 AWS Sensor Pack - Port requirements

The AutoMonX AWS sensor pack 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 correctly to allow the sensor pack to function correctly.

Port / URL/ IP	Purpose	Direction
List of IP addresses that are needed for full discovery of resources: https://ip-ranges.amazonaws.com/ip-ranges.json For each service, and region, the following url should be open: https://<service-name>.<region>.amazonaws.com i.e.: https://s3.us-east-1.amazonaws.com	AWS API connection	From PRTG Probe to AWS
TCP 443, 80	Connect to AWS, PRTG API	From PRTG Probe to AWS and PRTG Core
TCP 8148 TCP 8092 TCP 8075 TCP 8091 TCP 8095 TCP 8989	Internal service ports. Make sure these ports are not occupied by other programs on the server.	No need to open FW rules.

4.3 AWS Sensor Pack – Anti-Virus Requirements

The AutoMonX AWS sensor pack 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)\AutomonX

from on-access scanning. This would greatly improve the general performance of the AWS sensor pack.

4.4 Downloading the AWS 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	Download		a3a9aaa4ffff6034c9e95bf8d4e4da50
AWS Sensor Pack	1.2.2	25.11.2025	Download		80a97b3612a780cbea5cebceb355212c
Azure Sensor Pack	4.3.31.1 (Stable)	23.12.2024	Download	New License for versions prior to v4.0.26	952ce321d6a0b30a6e8eb982308889da

The AWS sensor pack is deployed via an installer exe file:

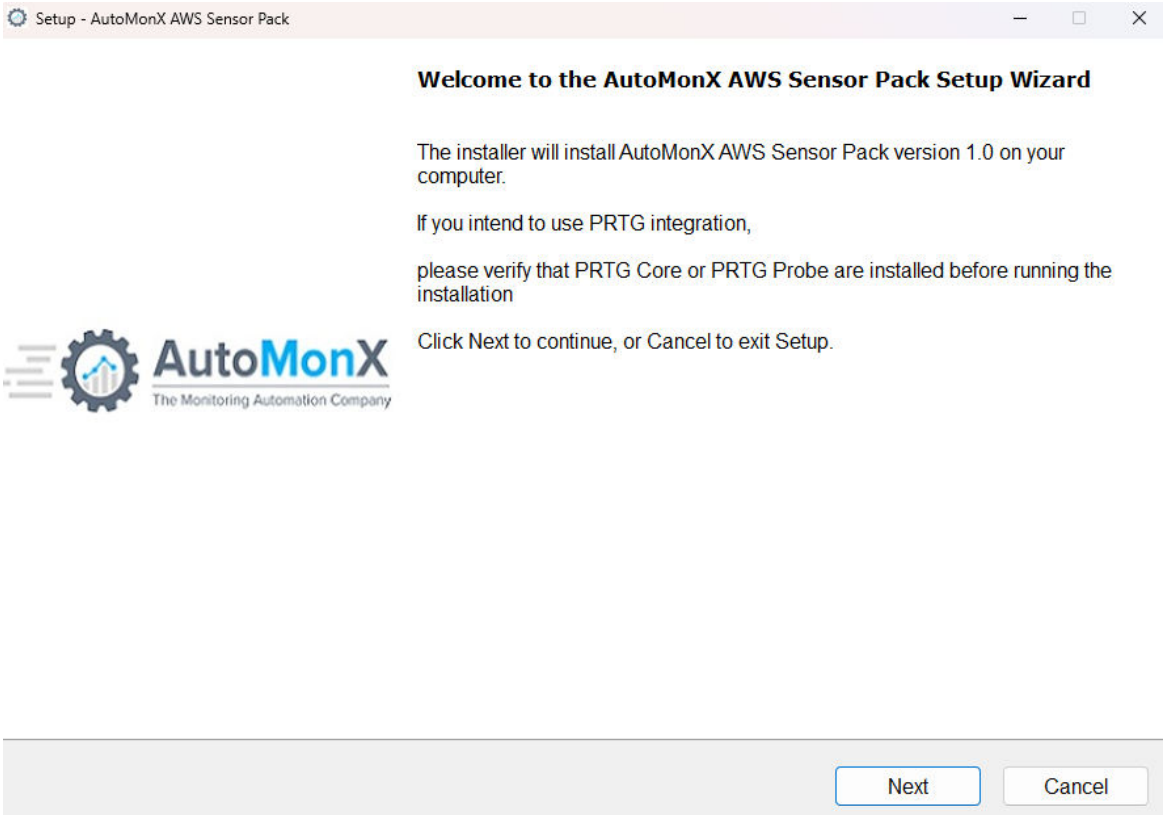
AutoMonX_AWS_Monitor_Pack_Installer_<version>.exe

Web : http://www.automonx.com
Email : support@automonx.com

4.5 Installing the AWS Sensor Pack files using the Installer

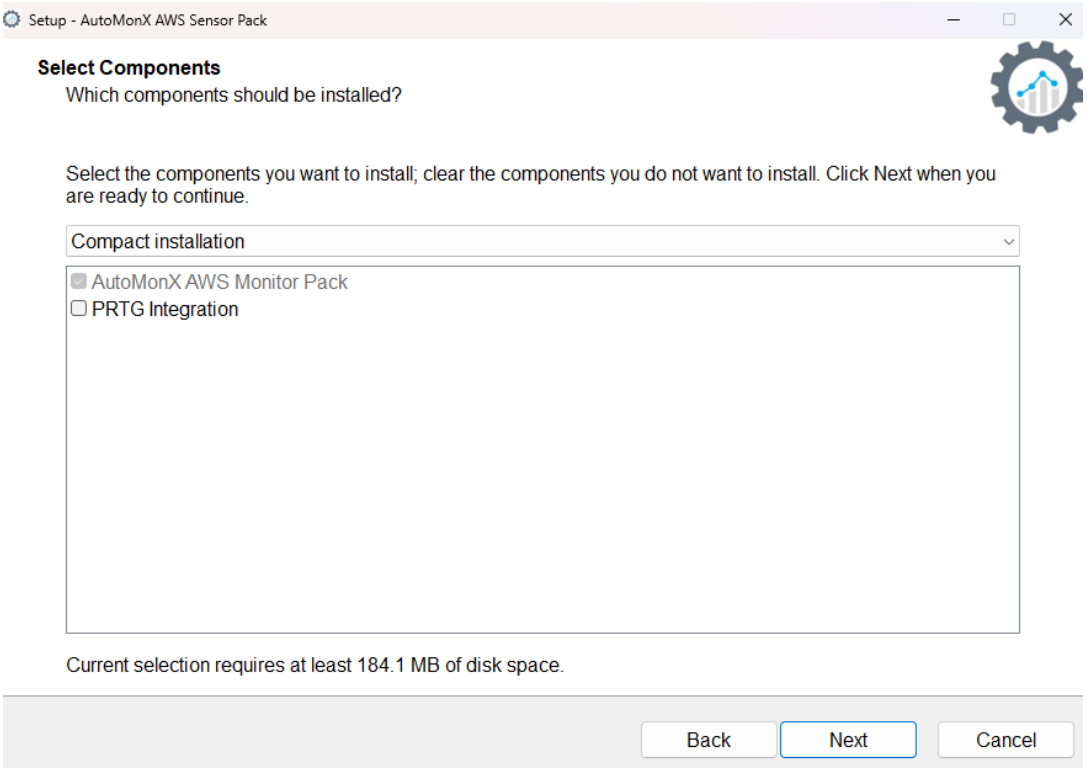
Download the latest AWS Sensor Installer from
https://www.automonx.com/downloads

Start the installer and follow the instructions:

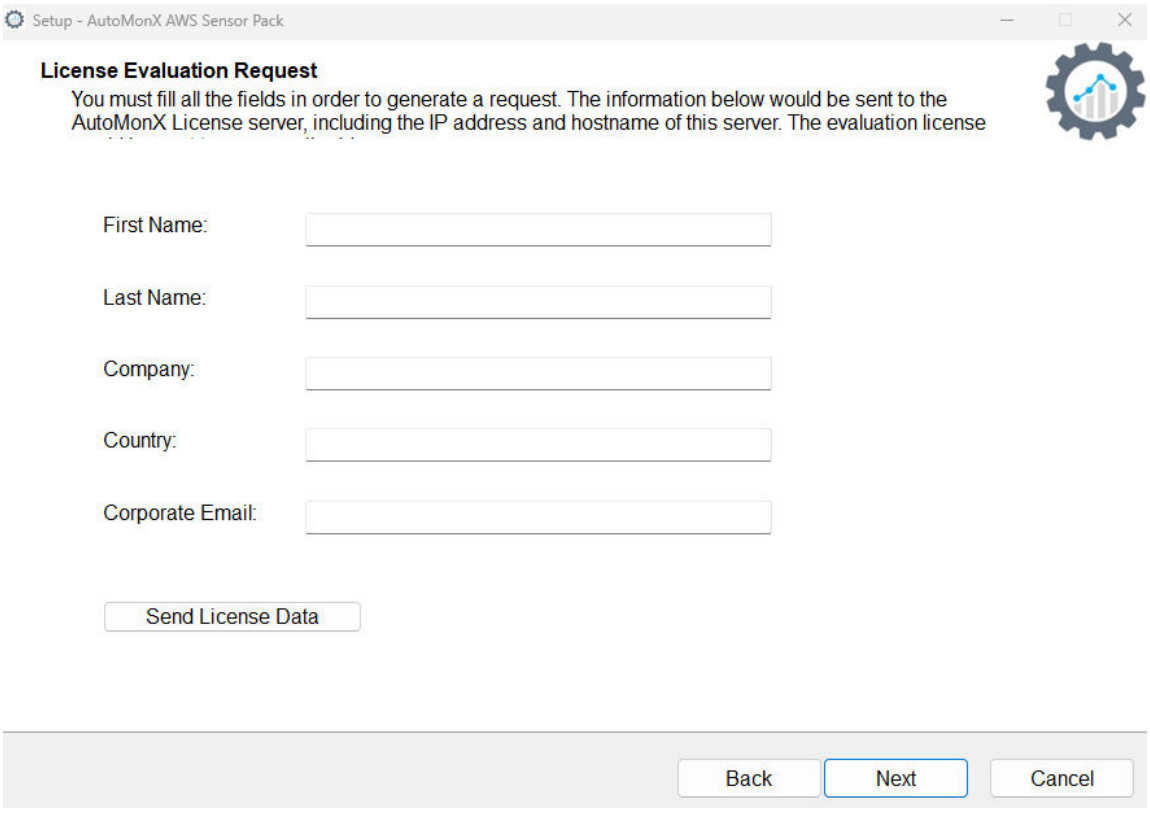


Web : http://www.automonx.com
Email : support@automonx.com

Select the components you wish to install:

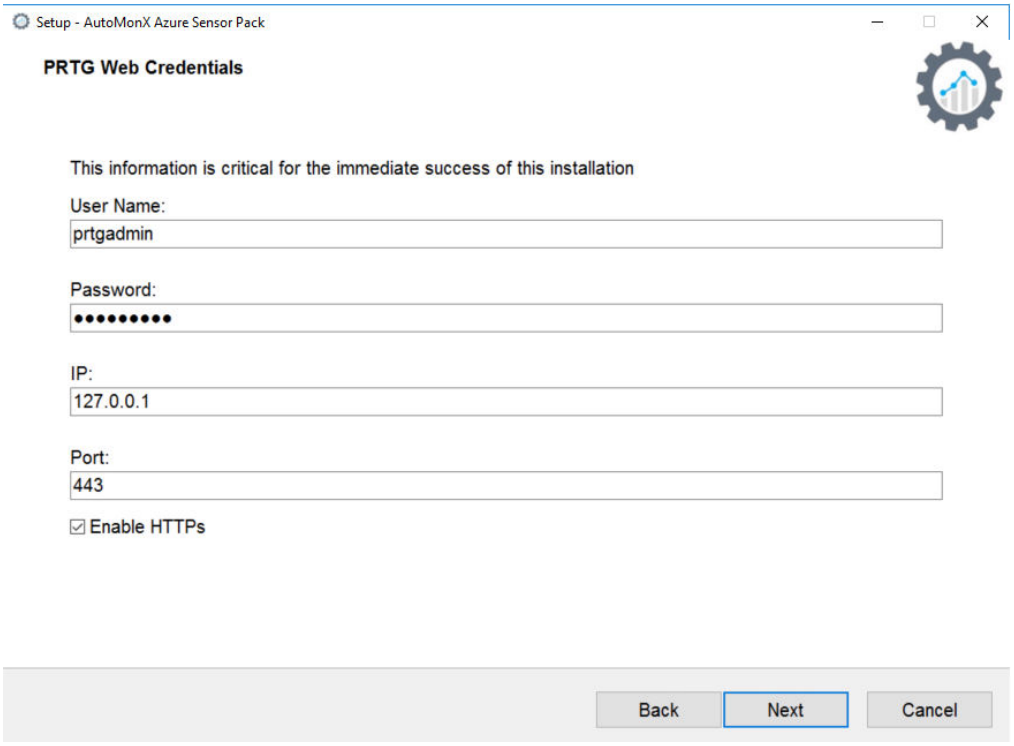


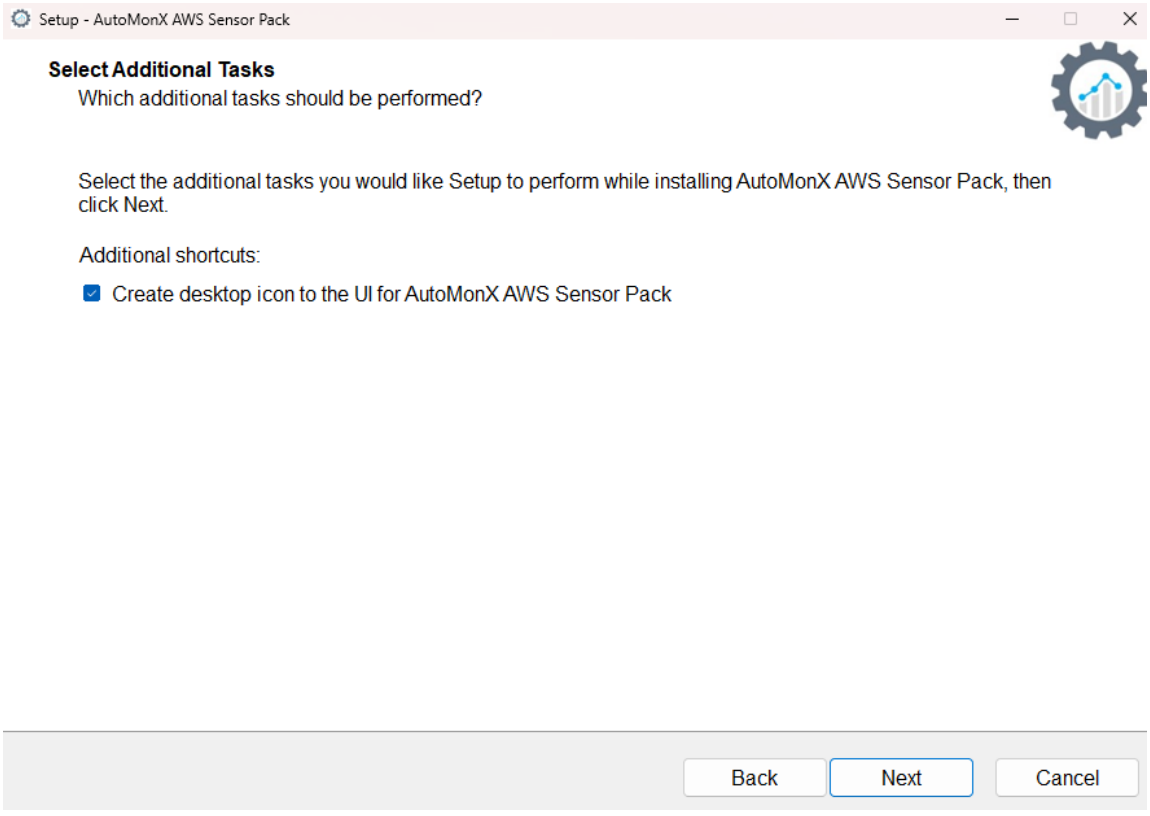
Fill in the details to get an evaluation license, and click on “Send License Data”:



Web : http://www.automonx.com
Email : support@automonx.com

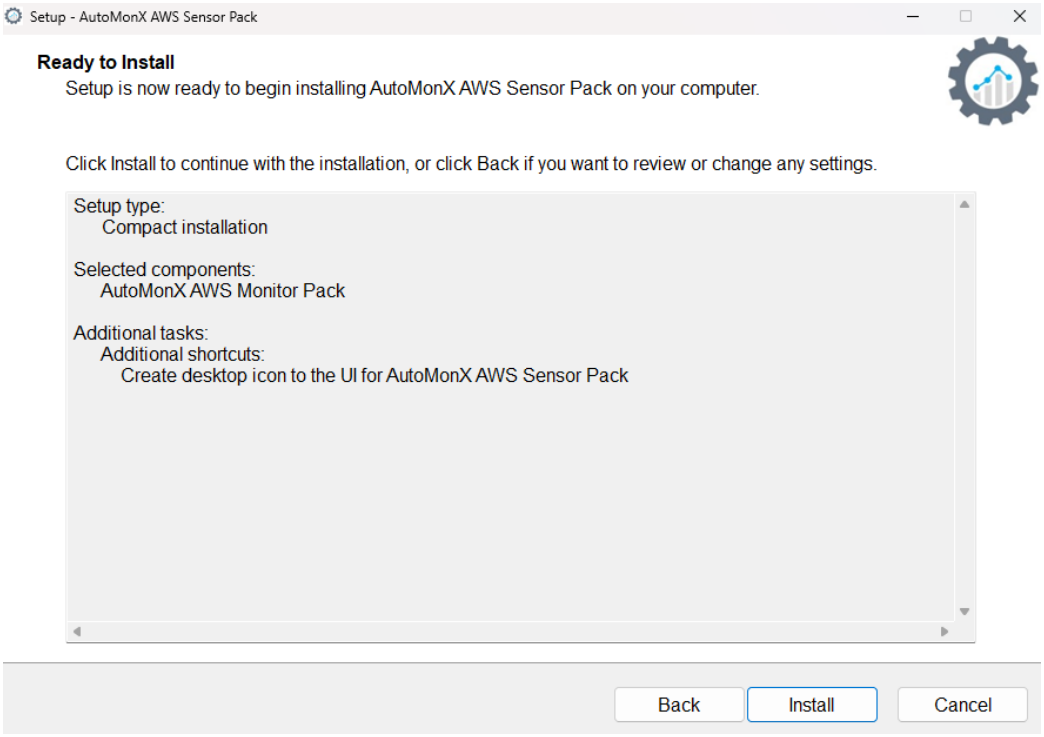
Configure the PRTG connection information. Make sure to mark “Enable HTTPS” if relevant.

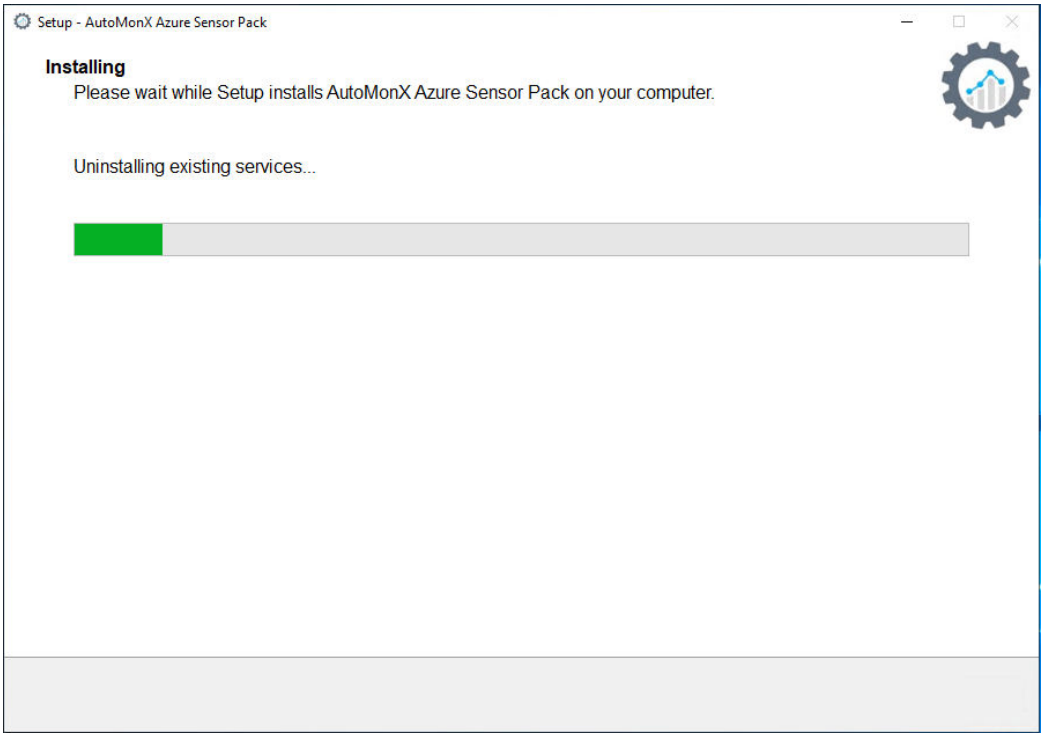




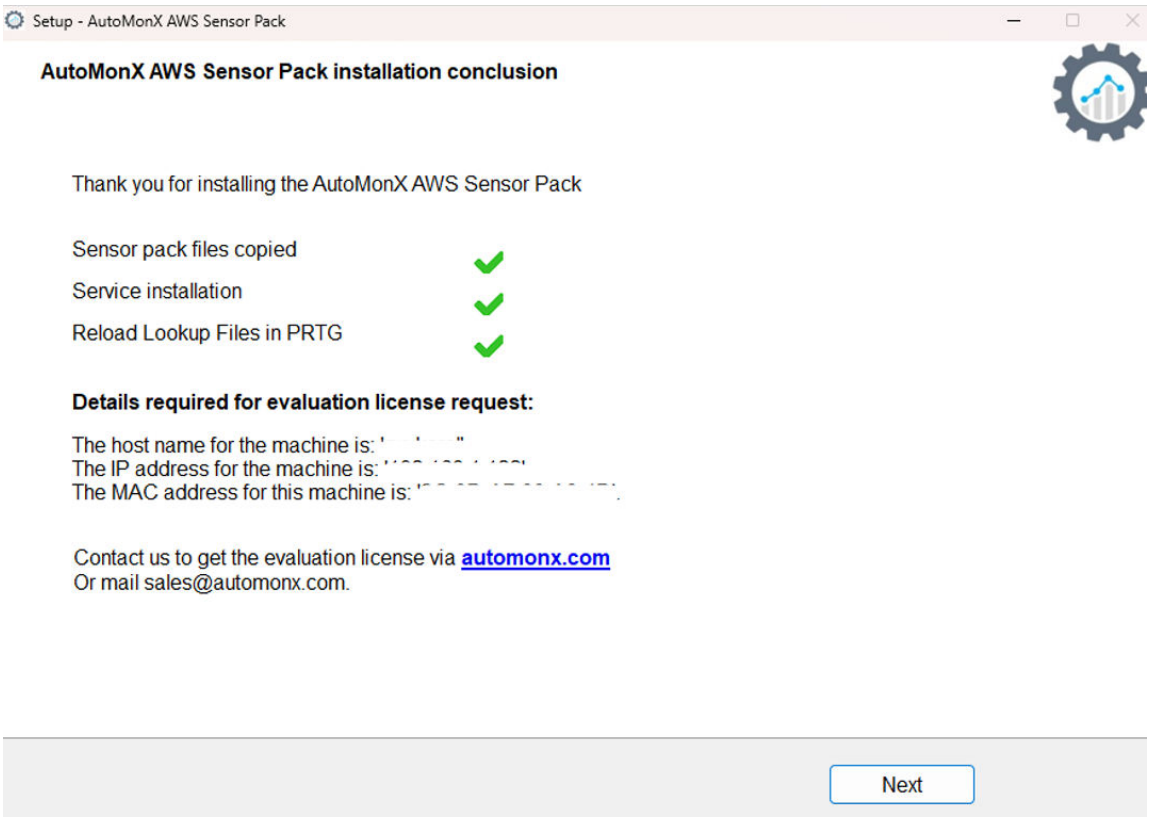
Web : http://www.automonx.com
Email : support@automonx.com

This is for example the window before the installation when selecting Compact installation:





Web : http://www.automonx.com
Email : support@automonx.com



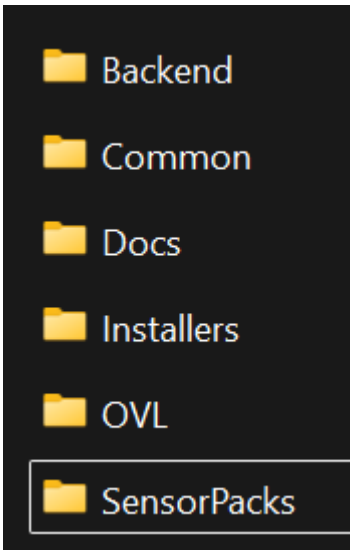
If an error occurred while updating the Lookup files, update them manually as explained in section 4.6.

4.6 AWS Sensor Pack files content

"<drive>:\Program Files (x86)\Automonx\"

The AWS sensor will not function anywhere else. The extracted files will create a directory structure as seen below.

AutoMonX directory content:



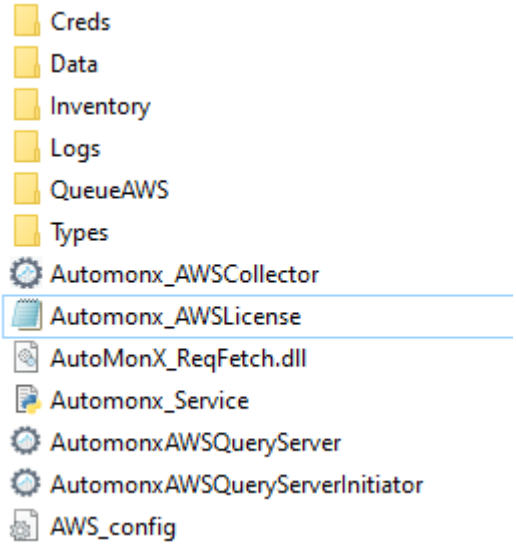
Common directory would include the following files:

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 machineDetails.txt AutoMonX_MonAutomationLicense.dat	Utility to gather the required details for license generation And software license
AutoMonX_PRTG_Automation.exe AutoMonX_PRTG_Automation.ini AMX_PRTG_senors_issues.exe	Monitoring Automation module files
exclude_mon - Example.csv exclude_mon.csv include_mon.csv down_sensors_filter.ini pause_sensors_filter.ini groups.txt group_list.ini	Filtering logic files for Monitoring Automation

sensors.txt	
AutoMonX_ReqFetch.dll libcrypto-1_1-x64.dll libgcc_s_seh-1.dll libssh2-1.dll libssl-1_1-x64.dll libstdc++-6.dll libwinpthread-1.dll zlib1.dll	DLLs required for Monitoring Automation

SensorPacks/AWS directory would include the following files:

Filename	Purpose
Data Inventory Logs Creds QueueAWS Types	Sub directories required for the AWS sensor operation
Automonx_AWSCollector.exe AutomonxAWSQueryServer.exe AutomonxAWSQueryServerinitiator.exe Getcreds.cmd	AWS sensor executables
AWS_config.ini	AWS sensor main configuration
AutoMonX_AWSLicense.dat	AWS sensor license file – Sensors
libcrypto-1_1-x64.dll libgcc_s_seh-1.dll libssl-1_1-x64.dll libstdc++-6.dll libwinpthread-1.dll zlib1.dll AutoMonX_ReqFetch.dll	AWS sensor DLL files



Backend directory would include the following files:

Filename	Purpose
Logs QueueBackend QueueScheduler	Sub directories required for the AWS sensor operation
AMX_Application_Server.exe Automonx_Backend_service.exe configWizardHelper.exe	Backend executables
Automonx_Backend_Service.ini	Main configuration
libcrypto-1_1-x64.dll libgcc_s_seh-1.dll libssl-1_1-x64.dll libstdc++-6.dll libwinpthread-1.dll zlib1.dll AutoMonX_ReqFetch.dll	DLL files
Automonx_BackendIndex.indx	Backend indexing

OVL directory content:

Filename	Purpose
automonx.aws.alarmstatus.ovl automonx.aws.dynamodbstatus.ovl automonx.aws.ec2status.ovl automonx.aws.ec2volumestatus.ovl automonx.aws.ecsstatus.ovl	PRTG custom lookup file for the AWS sensor pack

automonx.aws.acmcerttype.ovl	
automonx.aws.acmhealthstatus.ovl	
automonx.aws.acmstatus.ovl	
automonx.aws.acmvalidationmethod.ovl	
automonx.aws.apigatewaystatus.ovl	
automonx.aws.bedrockstatus.ovl	
automonx.aws.efsstatus.ovl	
automonx.aws.elbstatus.ovl	
automonx.aws.emrstatus.ovl	
automonx.aws.fsxstatus.ovl	
automonx.aws.fsxsvmstatus.ovl	
automonx.aws.fsxtype.ovl	
automonx.aws.fsxvolumestatus.ovl	
automonx.aws.fsxvolumetype.ovl	
automonx.aws.backupstatus.ovl	
automonx.aws.sesidentitystatus.ovl	
automonx.aws.sesstatus.ovl	
automonx.aws.transitgatewaystatus.ovl	
automonx.aws.wafstatus.ovl	
automonx.aws.elasticbeanstalkstatus.ovl	
automonx.aws.cloudfrontstatus.ovl	
automonx.aws.kmskeystatus.ovl	
automonx.aws.rdsstatus.ovl	
automonx.aws.vpcstatus.ovl	

4.7 Lookup File Handling (on-prem PRTG)

You need the AutoMonX AWS Sensor Lookup files to properly display the sensor output in PRTG. Copy the entire files from the OVL folder starting with automonx.aws, located in the zip file to the following folder on the PRTG Core server.

From:
"<drive>:\Program Files (x86)\AutoMonX\OVL\automonx.aws.<*>.ovl
To:
"<drive>:\Program Files (x86)\PRTG Network Monitor\lookups\custom"

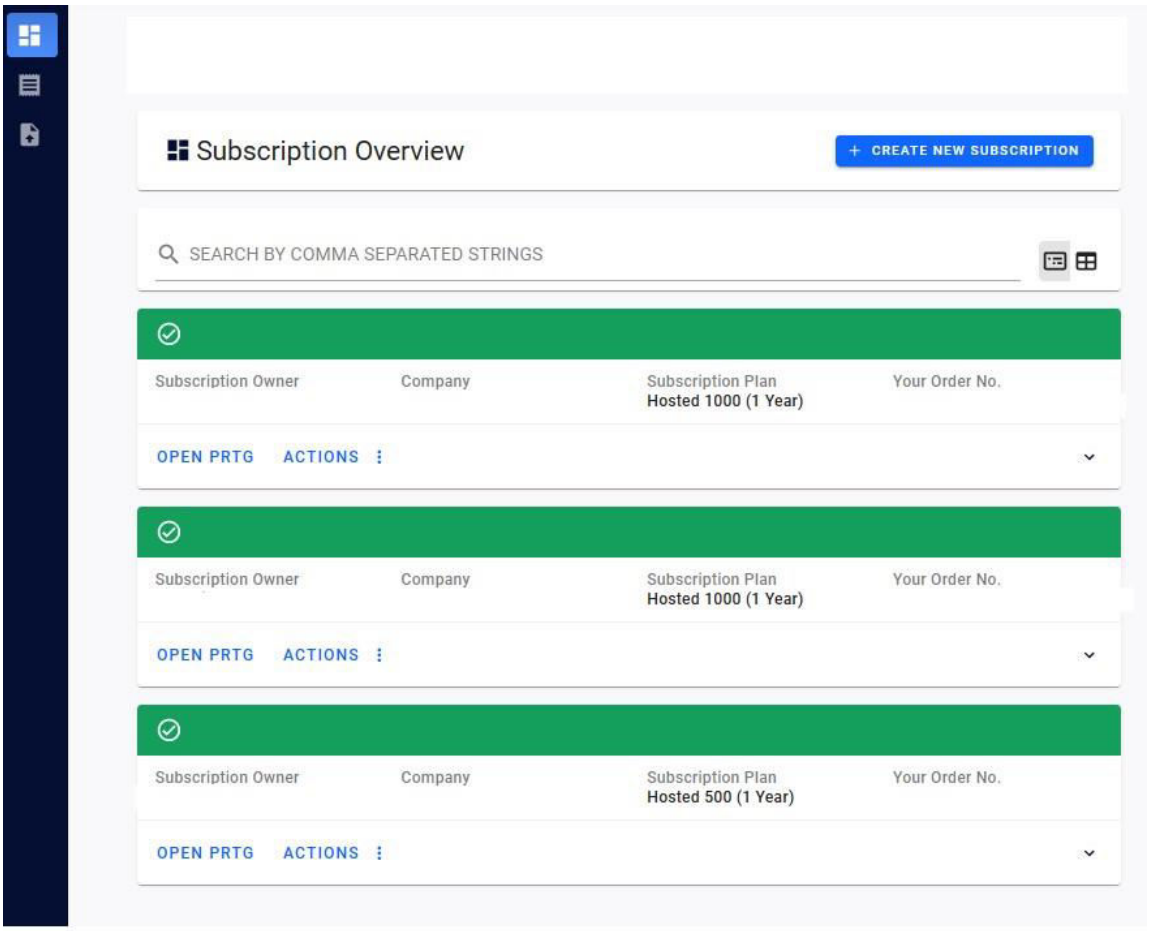
After copying the Lookup files to the PRTG Core Server, you would need to reload the PRTG Lookup database by the following action:

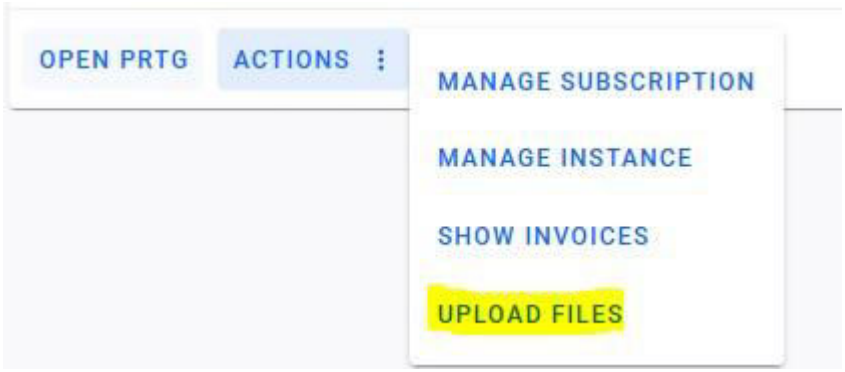
From the PRTG upper menu -> Setup -> System Administration -> Administrative Tools -> Reload Lookup Files

4.8 Lookup File Handling (PRTG Hosted)

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 or C:\Program Files (x86)\AutoMonX\OVL

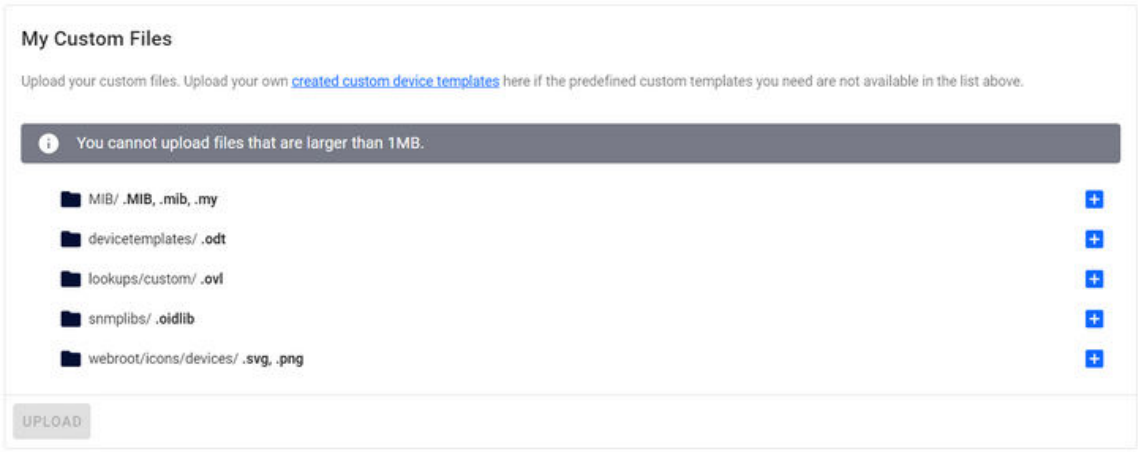
Web : http://www.automonx.com
Email : support@automonx.com





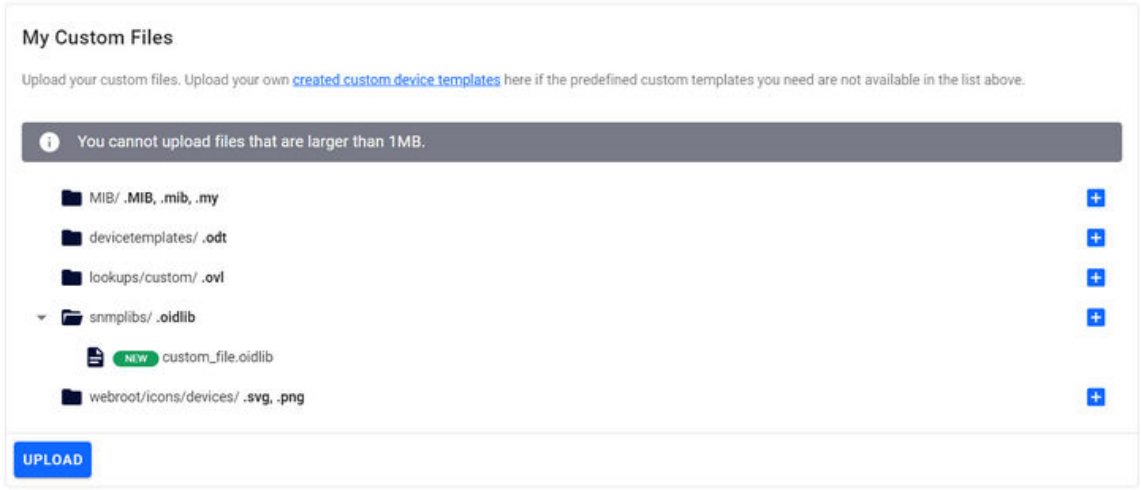
Upload Custom Files

Web : http://www.automonx.com
Email : support@automonx.com



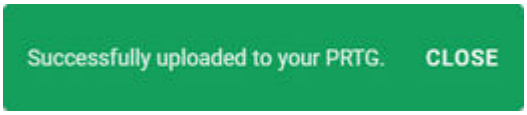
Custom Files

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**.



Custom File

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



Custom File Upload Successful

4.9 Requesting a License for the AutoMonX AWS Sensor pack

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

A license can be automatically obtained by the installer on initial setup!

If you have chosen to skip the installer license obtaining, or that an error occurred, to successfully activate the AWS sensor pack, you must contact AutoMonX Ltd either by filling the license request form at http://www.automonx.com/AWS

Or by sending an email to 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, or copy the details at the end of the installation process

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

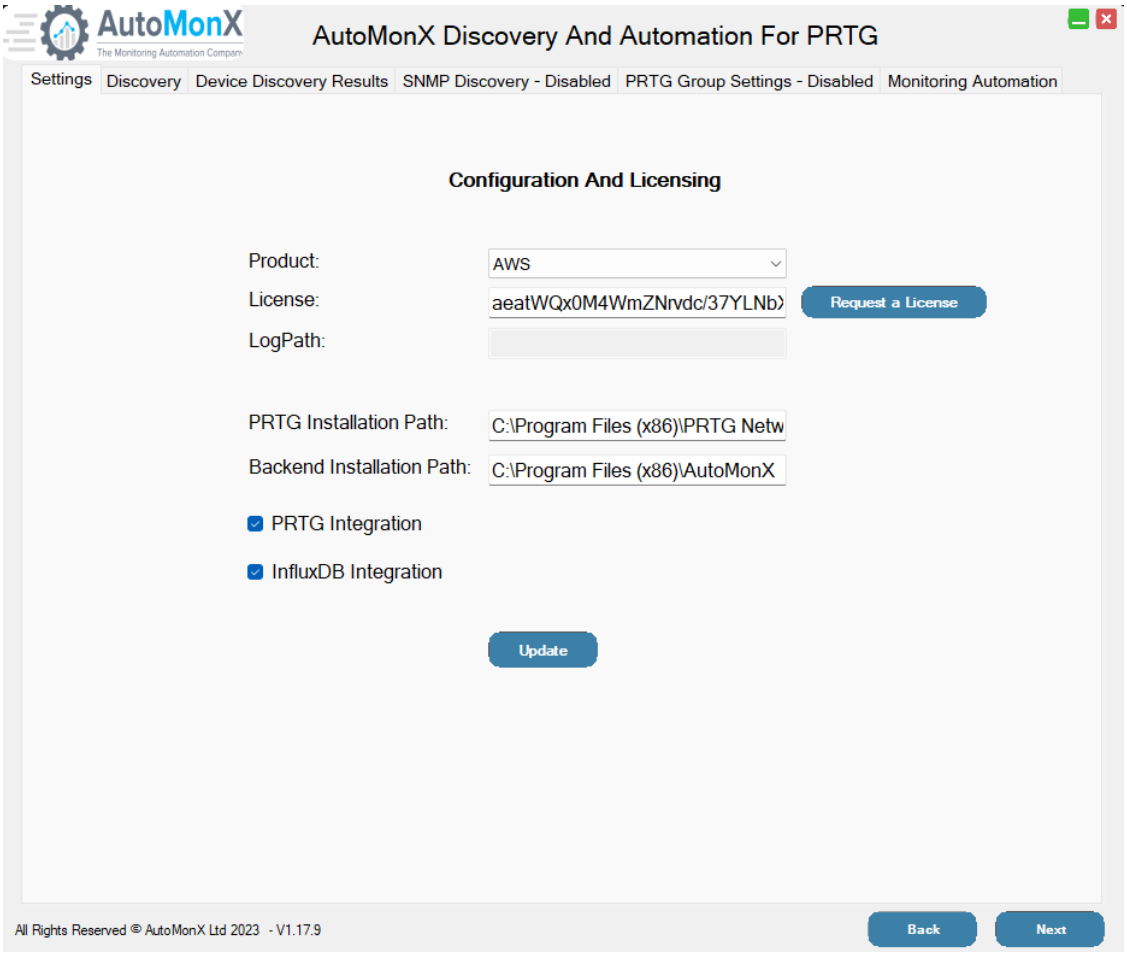
At the end of the evaluation period, you would need to purchase a license to continue monitoring your AWS infrastructure.

4.10 Activating the AWS Sensor pack License

You can activate the licenses of the AWS sensor pack by opening our UI and selecting the Settings Tab. Select “AWS” from the Product drop-down list (if not selected) and paste the license string you have received via email.

You can also activate the AWS sensor pack by editing the following files via Notepad, pasting the relevant license string you have received via email and saving the files.

AutoMonX_AWSLicense.dat – For AWS Sensor pack resources monitoring.



Web : http://www.automonx.com
Email : support@automonx.com

5 AWS Sensor Pack Configuration

5.1 Preparing for Configuring the AutoMonX AWS Sensor pack

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

- Access Key ID
- Secret Access Key

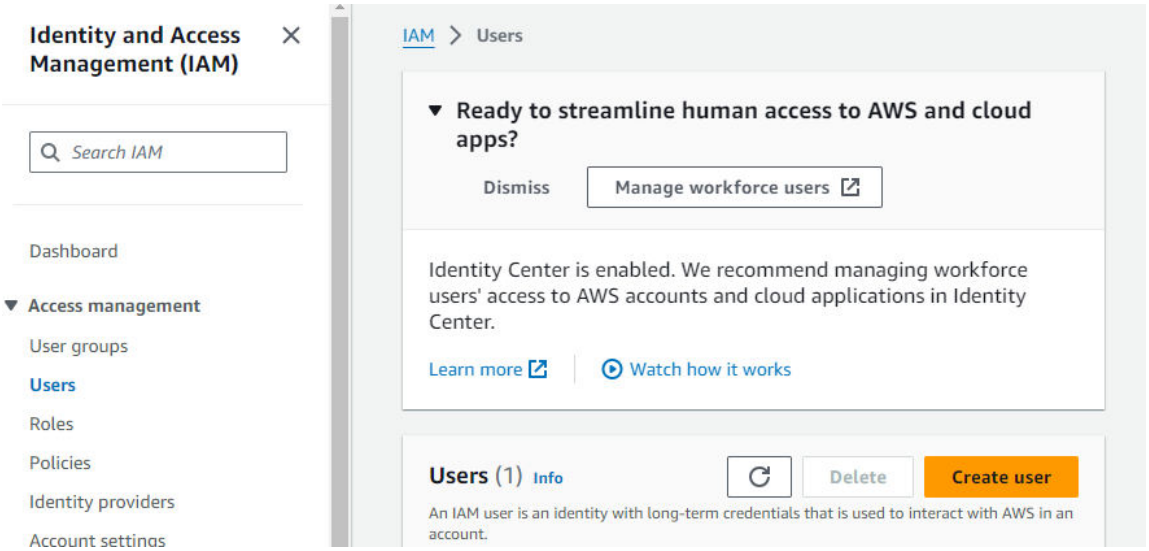
The connection information of the first AWS account added to monitoring is always stored in:
%userprofile%/.aws/credentials
C:\Windows\System32\config\systemprofile\.aws\credentials
Or, if encrypted, in the Creds directory inside AutoMonX AWS directory

The AWS connection settings are modified by using the AutoMonX UI. Check the Configuring the AWS sensor pack section for more information.
Use the following guide to create a service principal to get a Secret access key with the correct permissions:

5.1.1 AWS Management Console

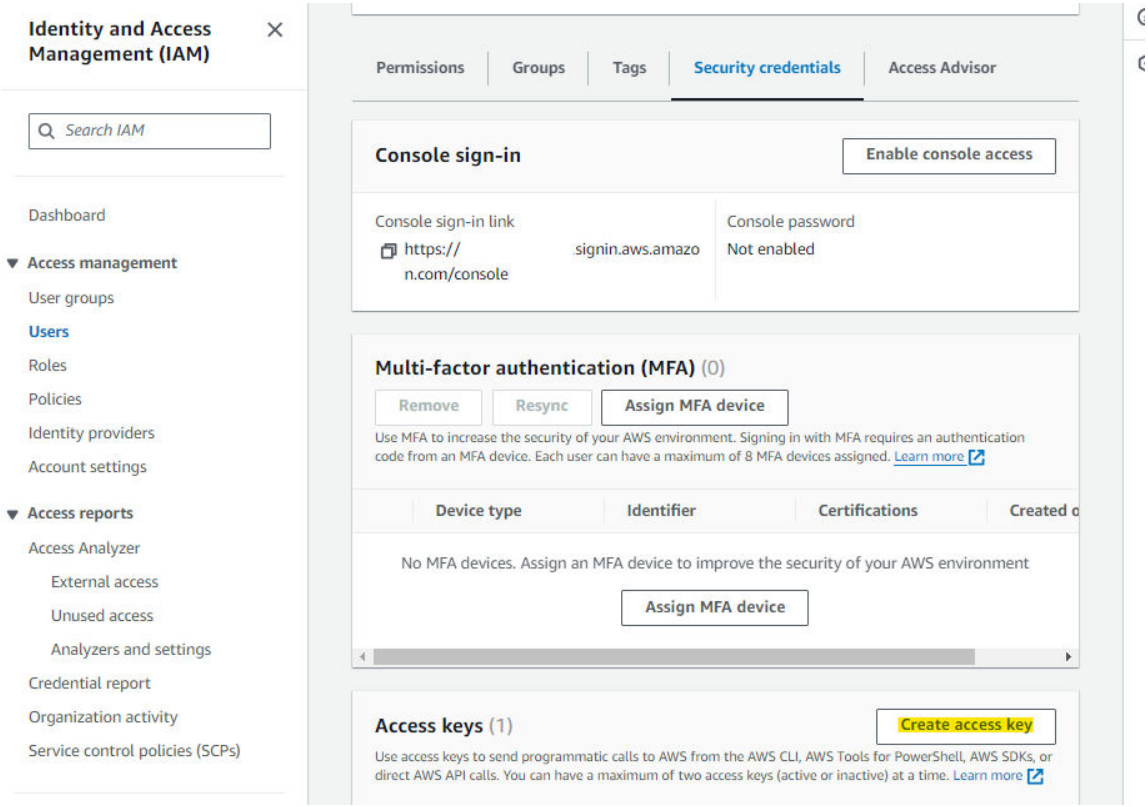
5.1.1.1 Read Only User Creation

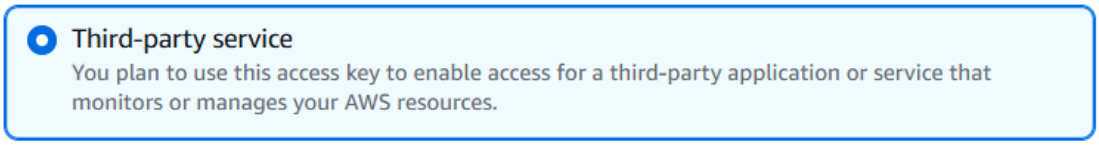
1. In the search window search “IAM”
2. On the left you will find “Users”
3. Select Create User
4. Add the user with your name selection



5.1.1.2 Generate Access Key, and Secret Key

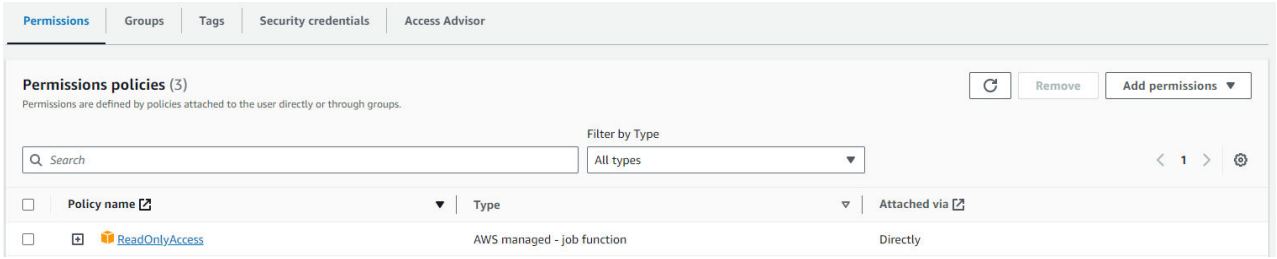
1. Go to the AWS Management Console.
2. Open the IAM service.
3. Select "Users" from the left sidebar and choose your IAM user or create a new one.
4. Navigate to the "Security credentials" tab.
5. Under the "Access keys" section, click on "Create access key" and choose “Third Party Service” access key type.
6. Note down the generated access key ID and secret access key – This will be used by the AutoMonX program.
7. It is possible to download the keys in csv format.





5.1.1.3 Set Up permissions to retrieve information from AWS.

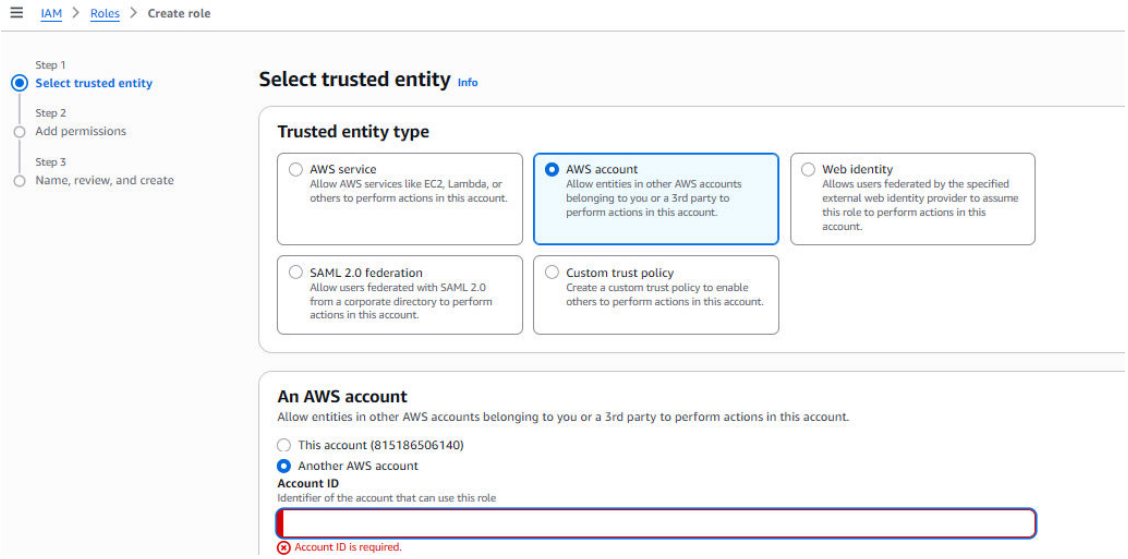
1. Go to the AWS Management Console and open the IAM service.
2. Select "Users" from the left sidebar and choose the user to which you want to assign the policy.
3. Navigate to the "Permissions" tab.
4. Select “Add Permissions” -> “add permissions”
5. Select “Attach Policy Directly”
6. Search for “ReadOnlyAccess”
7. After adding you should be able to see the permissions for the user as follows:



It is recommended to follow the entire guide in order to prepare all the relevant settings for the AutoMonX AWS sensor pack.

5.1.1.4 Creating credentials for monitoring multi-Account.

1. Follow previous steps to configure Central Account credentials
2. For each member account Navigate to IAM > Roles > Create role
3. Choose 'AWS Account'



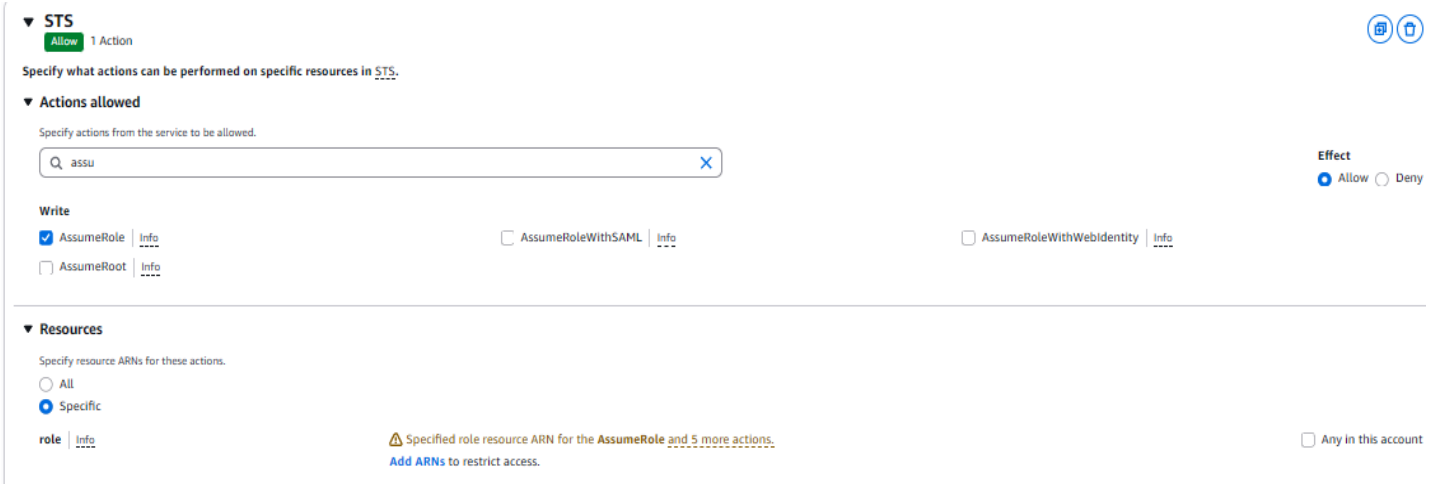
4. Select Another AWS Account

4. Enter Central Account's ID

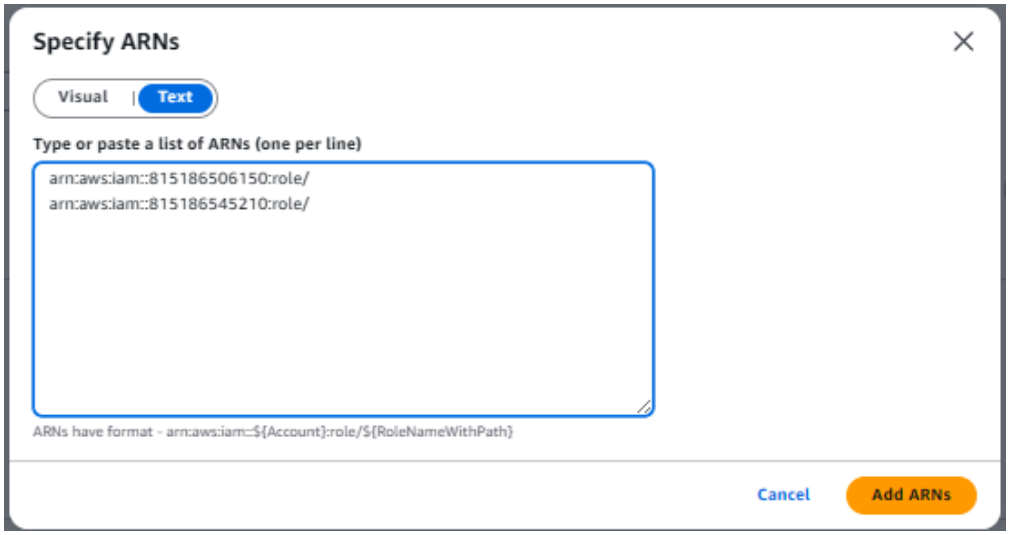
5. On the Permission step search for ReadOnlyAccess

6. After attaching and clicking next, give a name and description to the new rule

7. Click on create role, a green notification will appear on top. Click View Role.
8. Copy it's Role ARN and go to IAM > Users > Central Account > Permissions
9. On **Permissions policies** choose Add Permissions > Create inline policy
10. Select **Service**: STS, **Actions**: AssumeRole, **Resources**: Specific



11. Click on Add ARNs, choose Text and add ARNs line by line



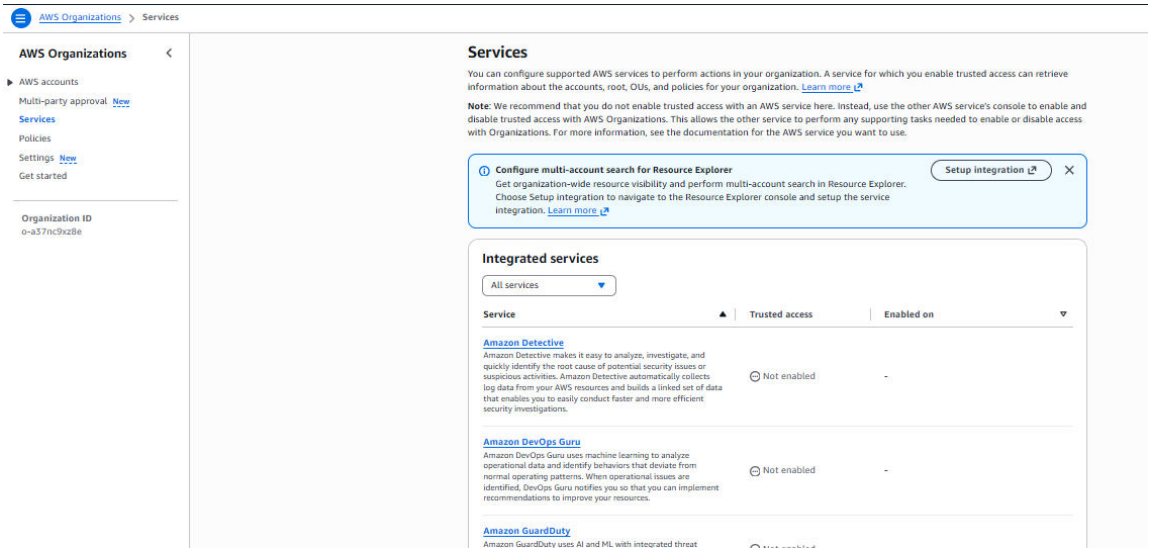
Use the Central Account credentials withing the AutoMonX Sensor Pack for AWS.

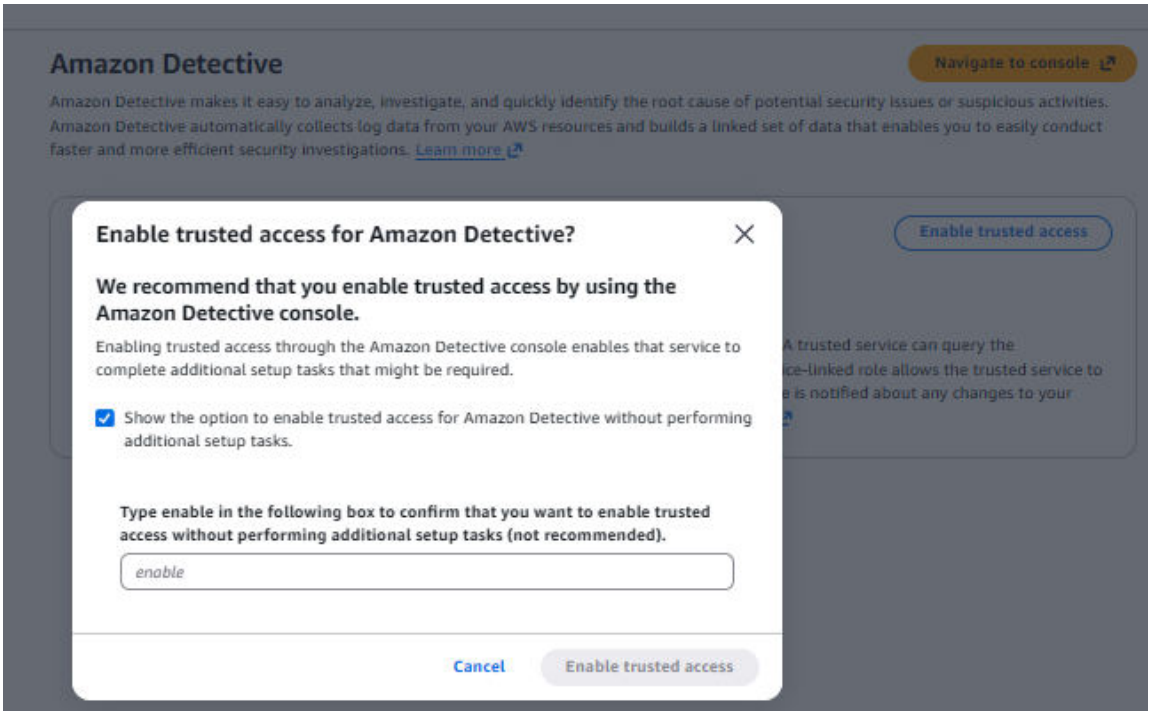
5.1.1.5 Enabling Trusted Access in AWS Organizations.

Trusted access lets a specific AWS service (for example **AWS Account Management**, CloudFormation StackSets, Config, etc.) act across **all accounts in your Organization** using service-linked roles that AWS creates

To enable it:

1. Sign in to the management account and open AWS Organizations from the AWS console
2. In the left navigation panel choose Services
3. Choose service to Enable trusted access
4. Click Enable trusted access
5. Type 'enable' in the confirmation dialog and select Enable trusted access

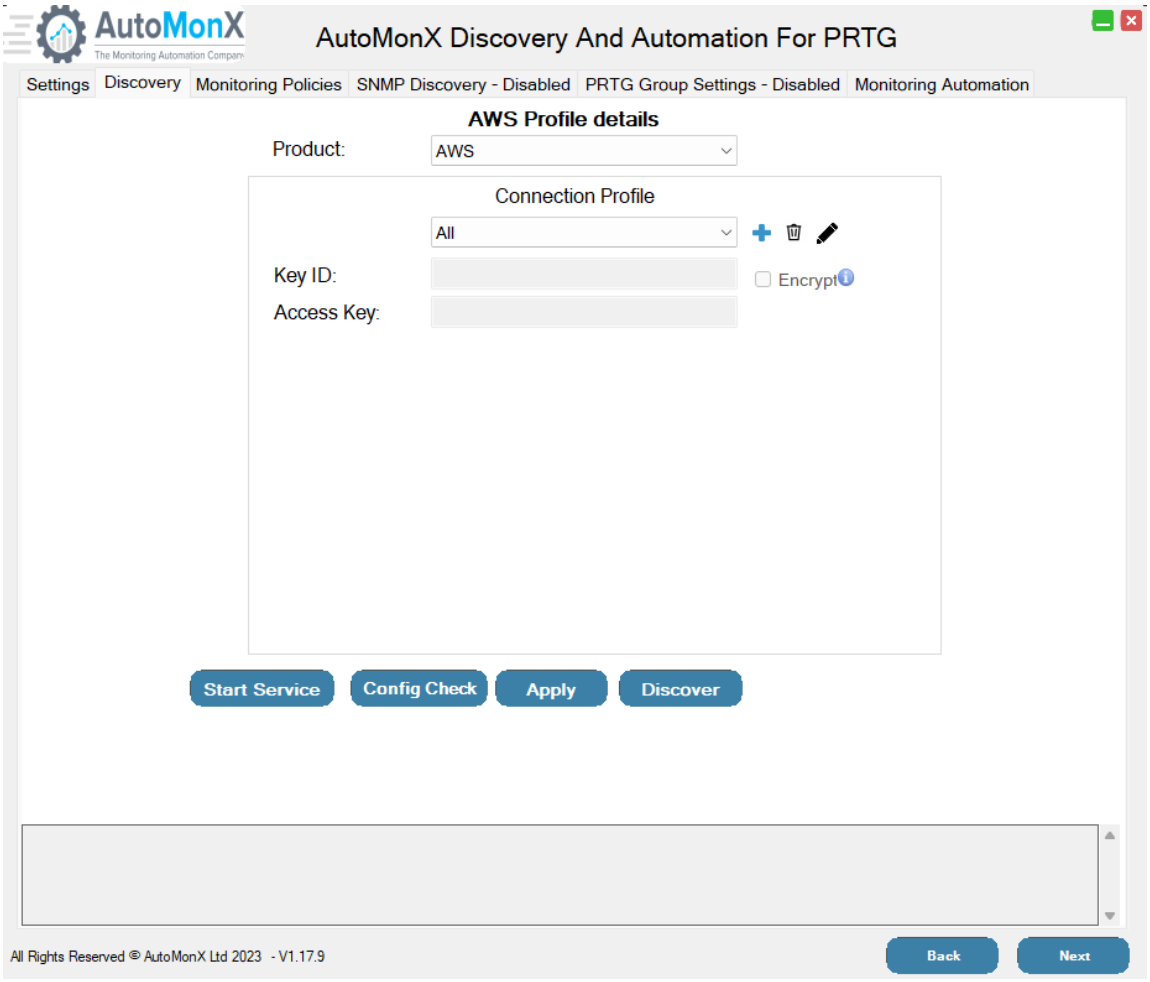




Configuring the AutoMonX Sensor pack for AWS

You need to start the AWS 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 the required details for the AWS sensor pack to connect to AWS API as there is no data at the beginning.

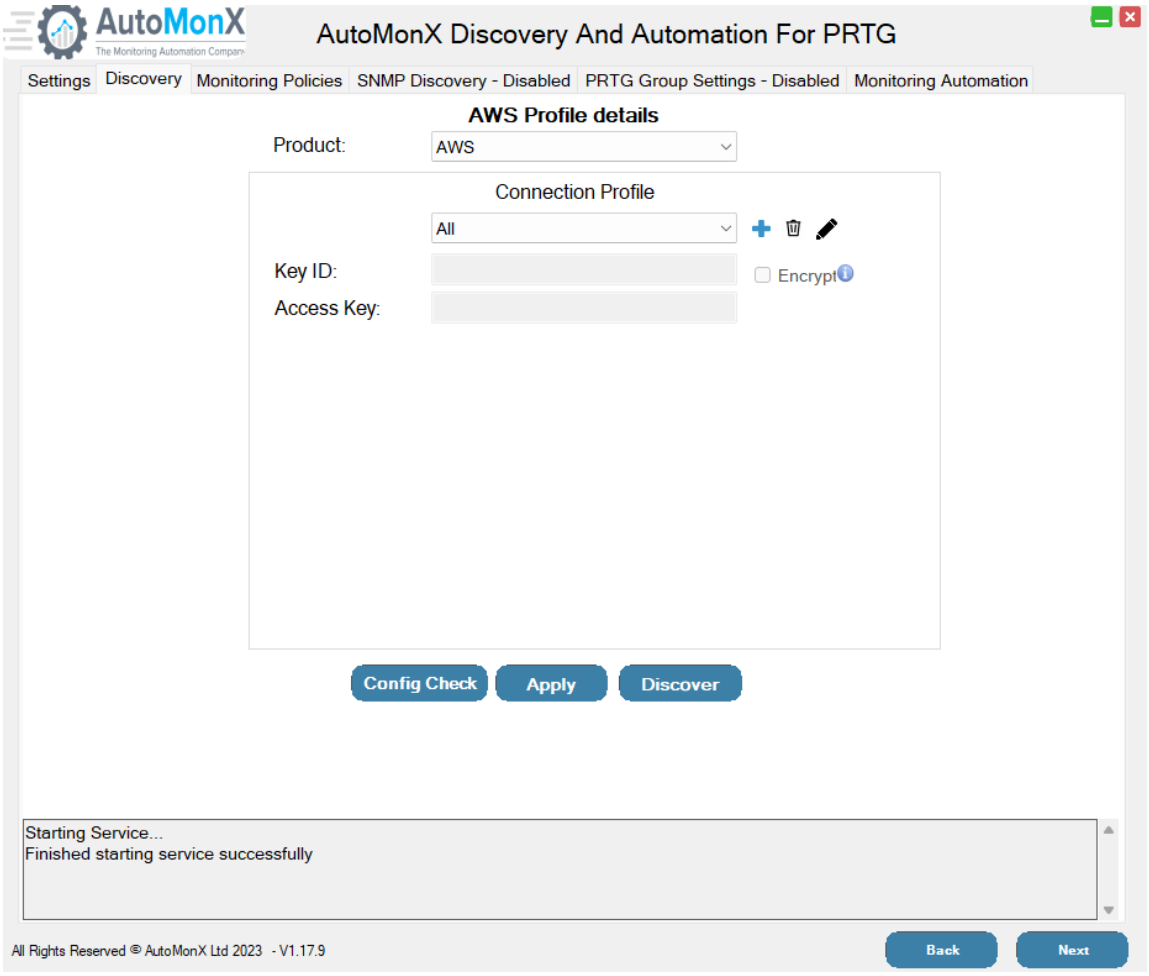


Important: Press the Apply button to save your changes and press the “Install Service”/”Start Service” button to install (or start) the AWS sensor pack service.

AWS Billing Information: The AWS billing information will be automatically discovered.

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

Web : http://www.automonx.com
Email : support@automonx.com

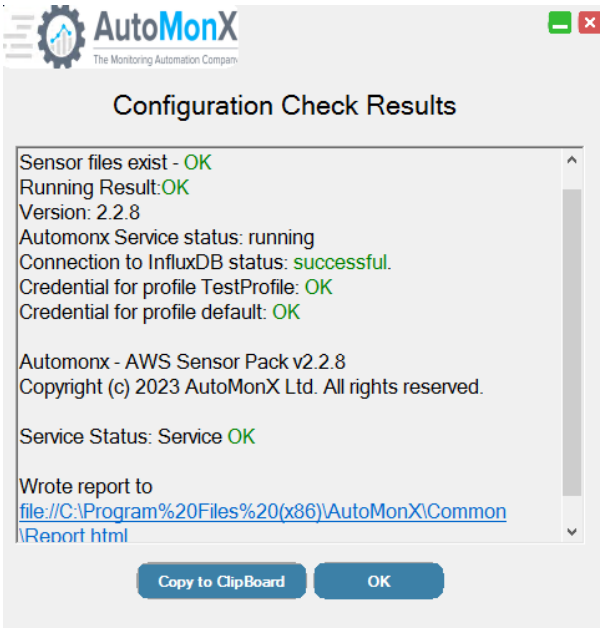


In some cases, where the AWS Sensor Pack service is not started, the UI would make visible the “Start Service” button so you can start the sensor pack service from our UI

5.2 AWS Sensor pack - Configuration Check

To verify your AWS sensor pack installation, start the UI, go to AutoMonX \Common directory and run as Administrator the SensorAutoDisco_UI.exe file.

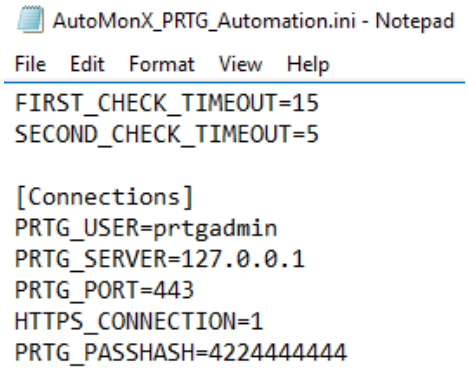
Fill in all the required information. Press the Config Check button to initiate a self-check to make sure everything was configured correctly. Successful test will look like the screen below:



5.2.1 Automatic Upgrade Using Installer (Recommended)

Using the AWS 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 1.2.0 of the AWS Sensor pack.

- Download the latest AWS Sensor Installer from https://www.automonx.com/downloads
- Make sure to pause the AWS 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:



- Make a backup of the entire AutoMonX folder.
- Start the installer and follow the instructions as in Installing the AutoMonX Sensor Pack for AWS files using the Installer
- If an error occurred while updating the Lookup files, update them manually as explained in section 4.6.

- Resume the sensors in PRTG.

5.3 AWS Sensor pack Service - Manual Installation

Skip this section if you have configured the sensor via our UI. Before installing the service, you must fill the INI file with correct parameters. It is strongly suggested to use our UI for this purpose. The following table shows the configurable settings in the AWS_config.ini file.

Parameter	Default Value	Details
data_lib	Data	The library where the data of the existing sensors found in the discovery will be saved
Inventory_lib	Inventory/AWS	The library where the aggregated data about services used in the AWS will be saved to
log_path	Logs/Automonx_aws.log	The log file location for the service
discovery_log_path	Logs/Automonx_discovery_out.log	Path for the log during the discovery process
verify_ssl	True	Using HTTPS with AWS
regions	Contains the full list of regions	This is a default value of available regions in case we can't get the available regions for a given account
evaluate_costs	True	If to use the cost explorer to get the costs of the AWS services usage
sensors_port	8091	An internal service port

listening_port	8989	An internal service port
sensors_interval	900	A ms interval for sensors polling
sensor_timeout	600	A ms interval for sensors polling
APIC_POLLING_INTETRVAL	300	Internally used value
SENSORS_POLLING_INTERVAL	600	A ms interval for sensors polling
INFLUX_SCHEDULER_DIR	A directory in the backend service directory	For internal use with InfluxDB
THREAD_NUMBER	8	Number of threads to run
RESULTS_SERVER	The querying service address	For internal use

After filling the required information, start cmd as Administrator and run the following command to install the service:

Automonx/Backend/Automonx_Backend_Service.exe -install

Note: This command must run with elevated permissions – this will pop up a User Access Control (UAC) message.

When the service installation was successful, the output would be as follows:

"AutoMonX AWS Monitoring Service installation successful!"

And the following command as well:

Sc.exe create Automonx_AWS_Query_Service binpath= <drive>:/program files (x86)/Automonx/SensorPacks/AWS/ AutomonxAWSQueryServerInitiator.exe

Sc start Automonx_AWS_Query_Service

6 Introducing Multi-Account AWS Monitoring

The AutoMonX AWS sensor pack is capable to automatically discover and monitor multiple AWS Accounts from a single PRTG probe. This major improvement allows CSPs/MSPs and large enterprises to monitor their entire AWS estate without being limited by Accounts boundaries.

Important License information: To support the Multi-Account version, new license types are available as specified below. If you require to monitor multiple accounts, make sure to contact AutoMonX sales sales@automonx.com to obtain the most suitable license type.

6.1 Multi-Account License Types Explained

Additional license types are available to facilitate the monitoring of resources in multiple AWS accounts.

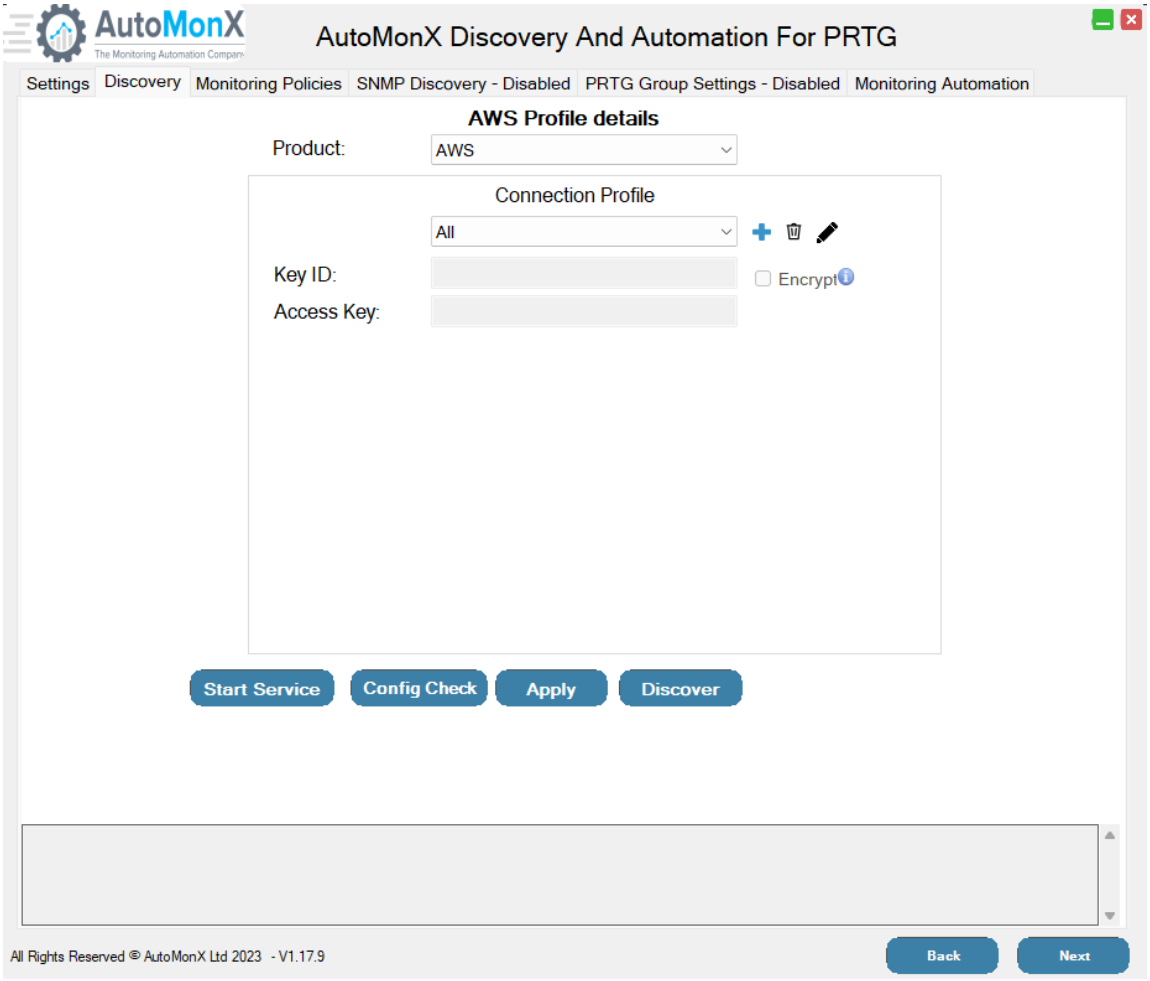
- The license is bound to PRTG probe machine
- Single-Account AWS Sensor pack licenses are still available
- Multi-Account licenses are sold in packs:
  - 5 Accounts
  - 10 Accounts
  - 15 Accounts
  - 20 Accounts
  - 25 Accounts
  - 50 Accounts

- Each Multi-Account license pack allows you to monitor an unlimited number of sensors within the number of Accounts you have purchased.

- **License boundaries/limitations**:
  - Your PRTG License
  - The number of Accounts in the Multi-Account 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-Account Discovery

You need to configure the details of each Account you wish to add to AWS sensor pack auto-discovery by using our UI.



Connection Profiles:

Connection Profiles have been introduced for quick and easy identification of Accounts 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 Account subscriptions and their AWS 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 Account label

6.3 Encryption of Connection Profile details

Version 4.x introduces the ability to encrypt the AWS connection profile details. You can choose to encrypt the connection details by ticking the check boxes on the right as seen below:

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 AWS 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_AWSCollector.exe -profile <profile name> -keyID <key id> -accessKey <secret_key>

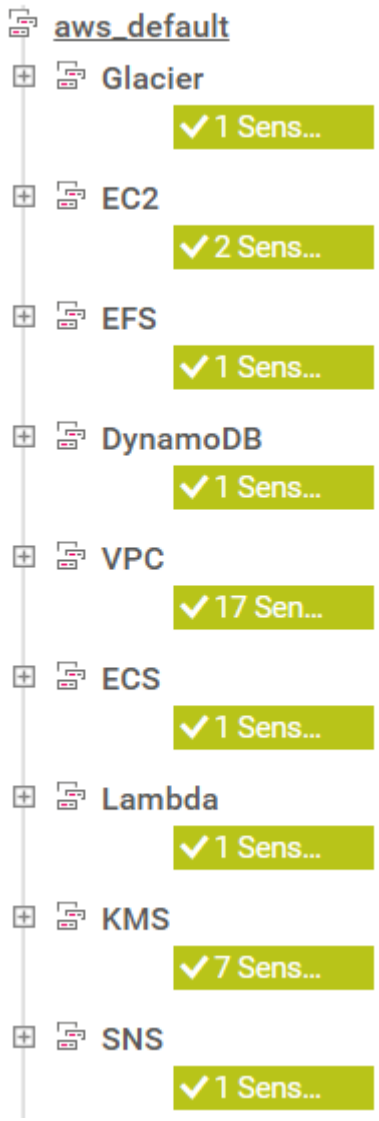
6.4 The AWS Sensor hierarchy in PRTG

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 AWS assets (needs to be manually created in PRTG) and specify the labels for each Account in our UI. The rest of the hierarchy would be automagically created for you in PRTG as seen below:

PRTG Probe (where the AWS sensor is installed)

- **AWS** (The top-level PRTG group you need to manually create, could be any name)
    - **Account label** (in a format of <Account Label>-<Last 4 digits of Account id) i.e. AutoMonX-3234. The group must be created manually.
        - **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:



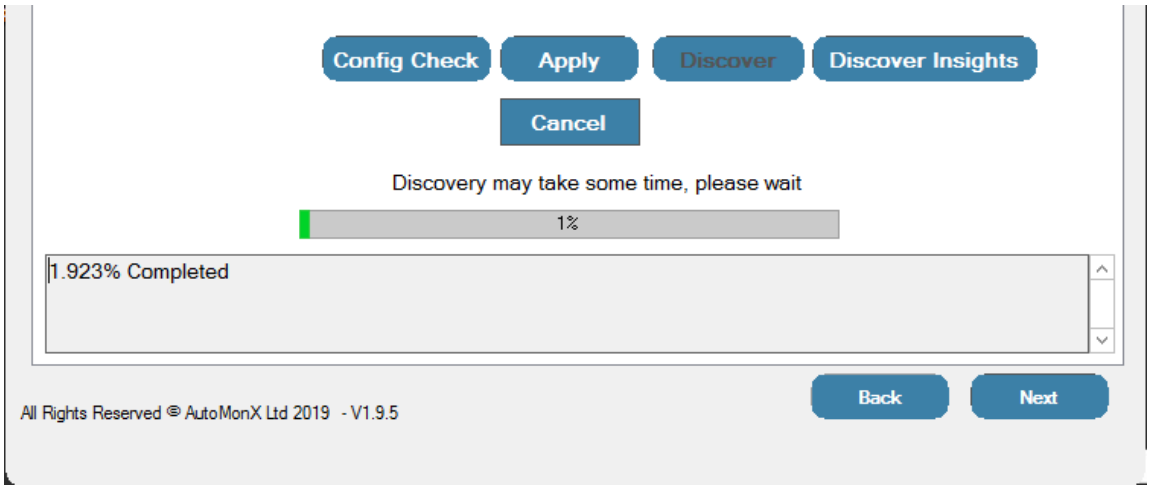
7 Auto Discovery and Monitoring Automation

7.1 Automatic Discovery of AWS Resources

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

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

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

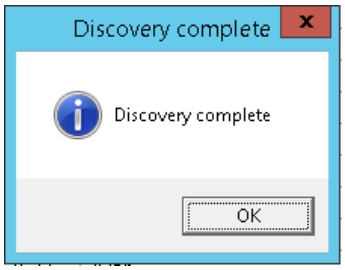


You can cancel the discovery process while it is running by pressing the “Cancel” button.

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

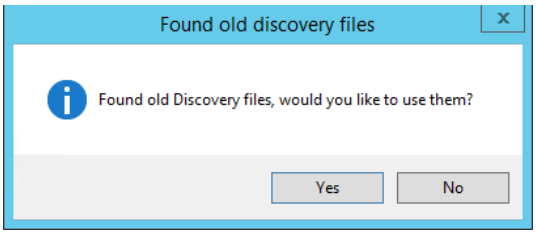
Timeout messages may appear sometimes during the discovery process, but you can safely ignore them if they last no longer than 10 minutes.

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

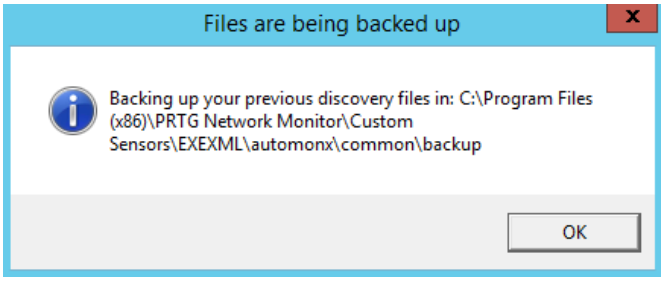


7.2 Previous Discovery Results handling

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



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



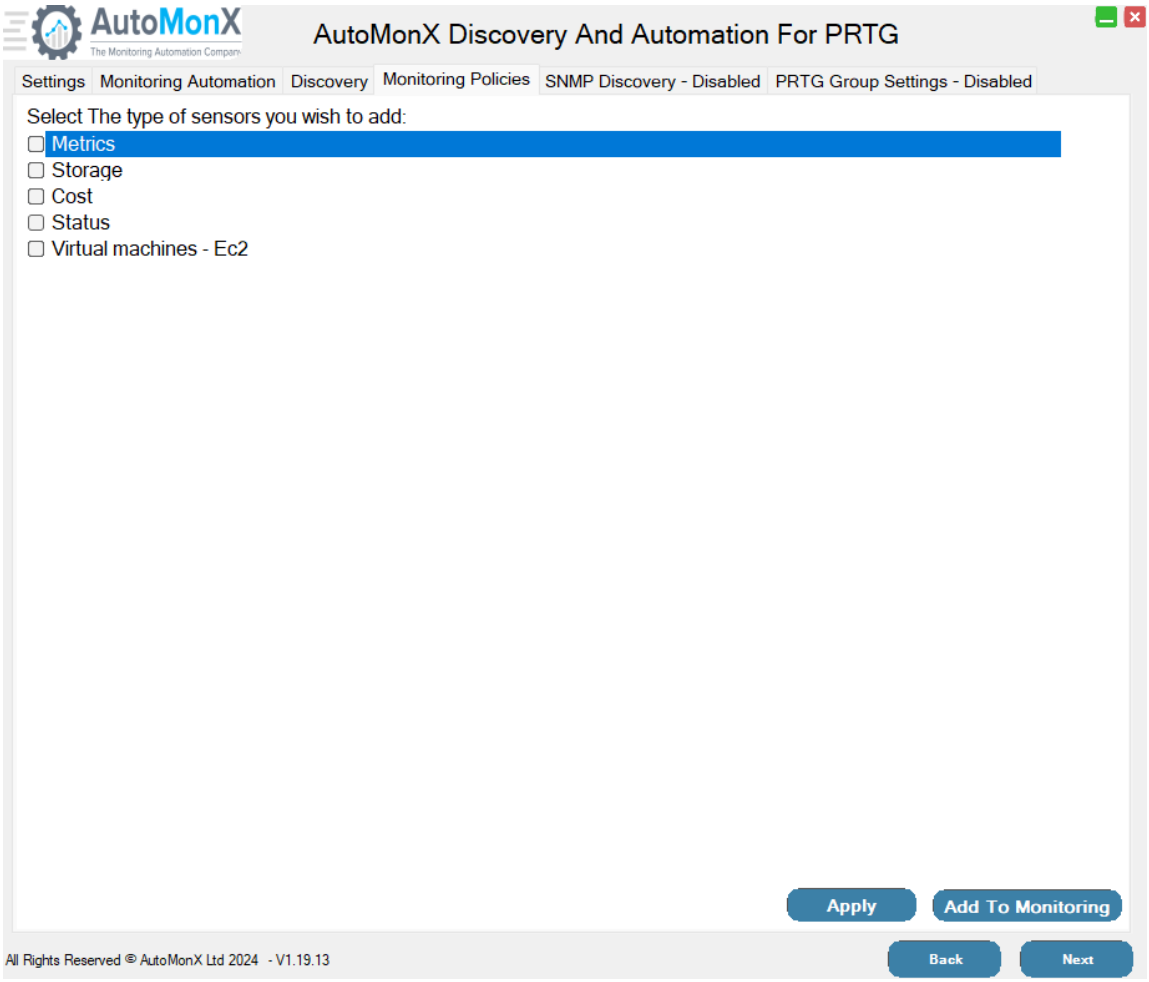
7.3 Sensor types created by the AWS Sensor pack

The AWS sensor pack auto-discovery will create several sensor types:

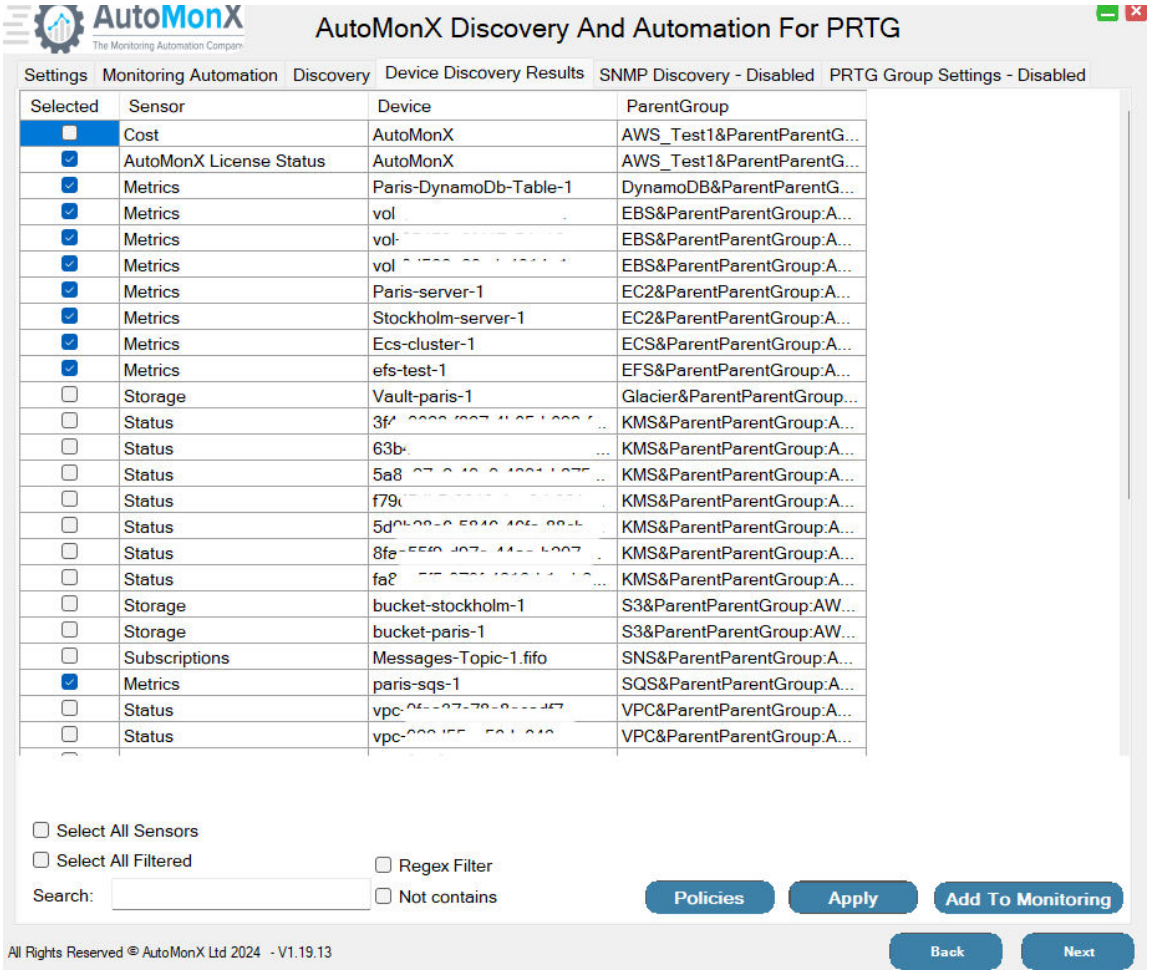
Sensor Type	Description	Actions in AWS Portal / PRTG
AWS Metrics	Resource-specific performance metrics available via the AWS API, multiple channels in PRTG (CloudWatch)	This feature uses CloudWatch metrics
AWS Cost	AWS Billing data, several channels that cover supported and un-supported resources	This feature uses the Cost Explorer of AWS
AWS Status	Resource-specific status	
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 AWS Sensors for Monitoring

Press “Next” to move to the next tab. All the discovered AWS resources would be presented, first you can select categories to be automatically selected for you:

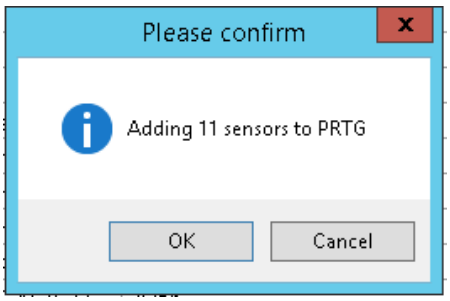


Then after pressing apply, you can select manually the sensors:



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.

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

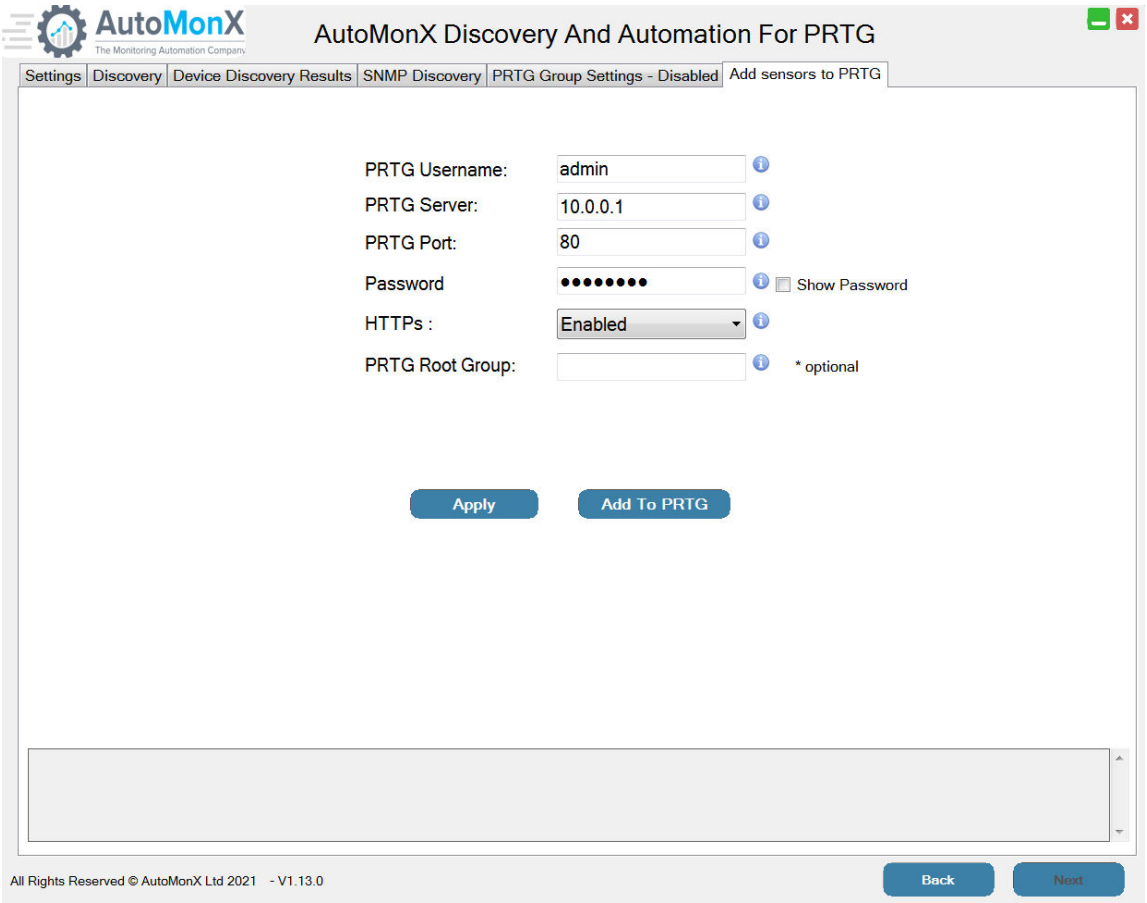


Press “Add To Monitoring” to add the sensors to PRTG/InfluxDB

7.5 Automatically Adding AWS Sensors to PRTG

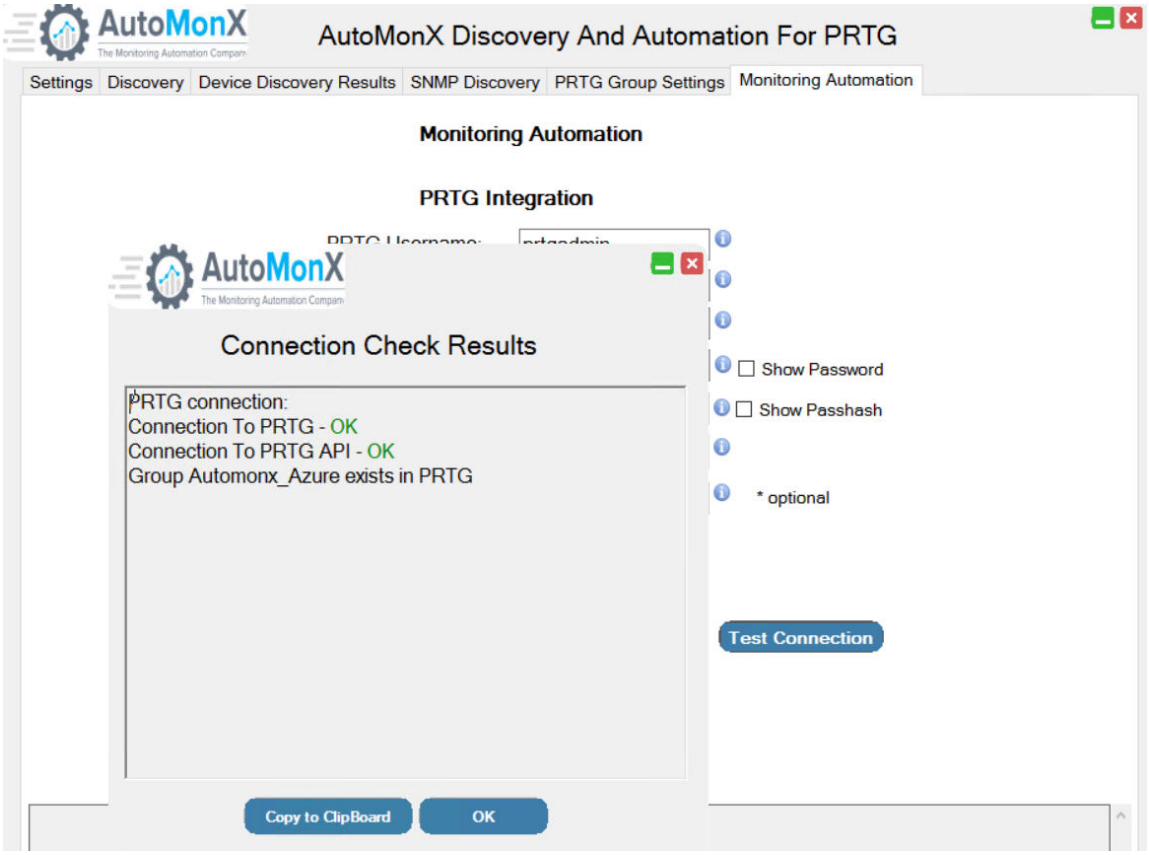
Important:

- 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
- Using Passhash is no longer required!
- You need to **manually** create a target group in PRTG that will contain the AWS resources sensors before running “Add sensors to PRTG”. The default group that our Monitoring Automation is configured to use is Automonx_AWS. You can create a group with a name of your choice and indicate it in the “PRTG Group” field.
- In case of **Multiple Accounts**, make sure to create groups with different names in PRTG. Discover each Account from its respective PRTG Probe and make sure to point our UI to the relevant group (per the AWS Account you have discovered)

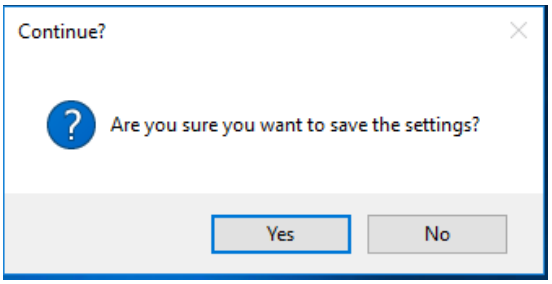


Press “Apply” to save your settings.

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 AWS 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.

Web : http://www.automonx.com
Email : support@automonx.com

7.6 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.

7.7 AWS Resources Discovery – CLI Options

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

- Discover all subscriptions and resources

  Automonx_AWSCollector.exe -discovery

- Discover resources of a specific Account

  Automonx_AWSCollector.exe -discovery -profile <Account Name>

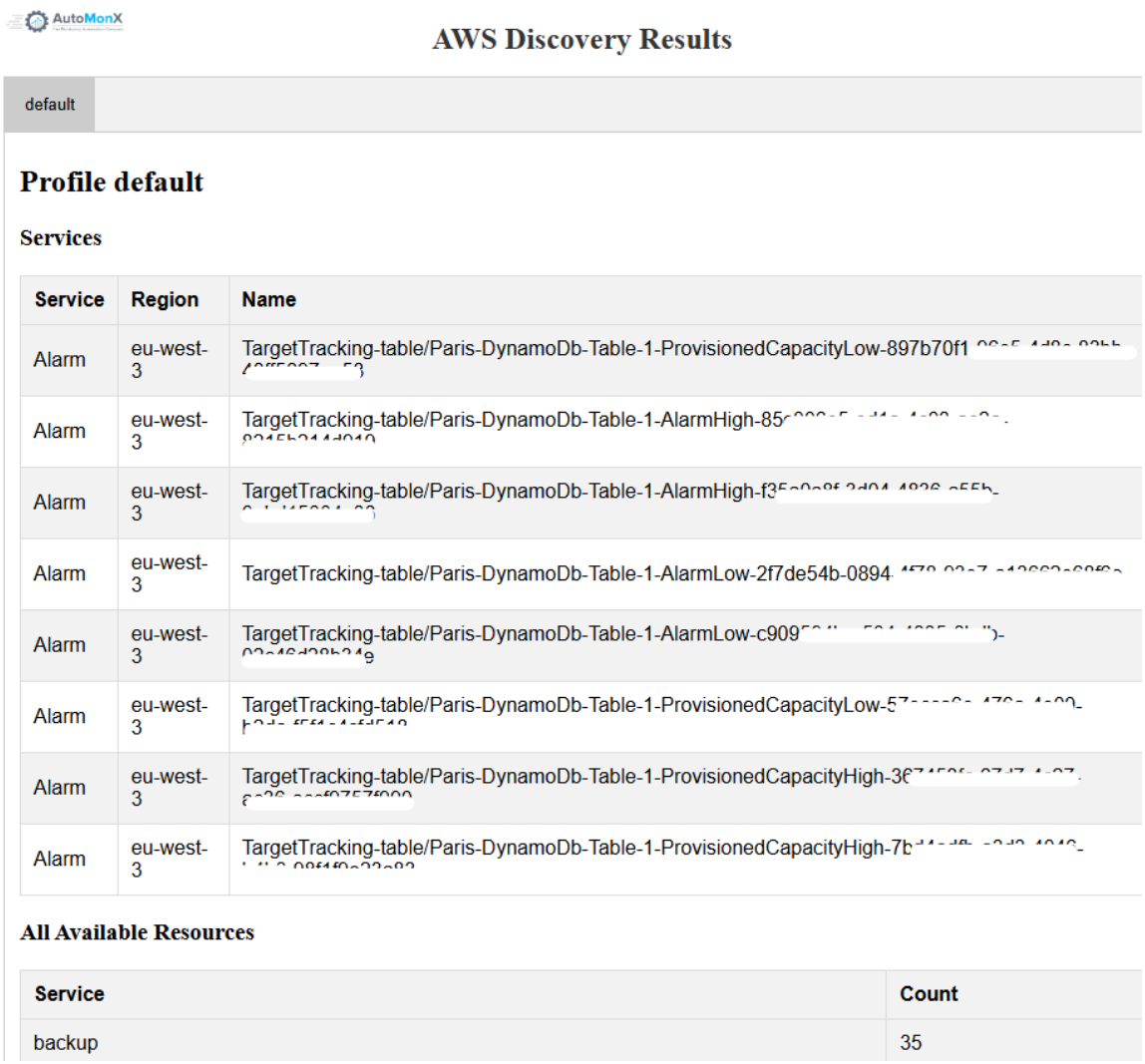
7.8 AWS Resources Discovery Report

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

- List of AWS resources per each Account.
- Command line parameters for the AutoMonX AWS PRTG Sensor pack that can be directly configured to monitor that resource.
- List of AWS resources that exists in the AWS Account by quantity

The report will be found in C:/Program Files(x86)/Automonx/SensorPacks/AWS/Logs/AWSDiscovery.html

Below is a sample report:



AutoMonX

AWS Discovery Results

default

Profile default

Services

Service	Region	Name
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-ProvisionedCapacityLow-897b70f1
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-AlarmHigh-85
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-AlarmHigh-f3
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-AlarmLow-2f7de54b-0894
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-AlarmLow-c909
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-ProvisionedCapacityLow-5
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-ProvisionedCapacityHigh-3
Alarm	eu-west-3	TargetTracking-table/Paris-DynamoDb-Table-1-ProvisionedCapacityHigh-7b

All Available Resources

Service	Count
backup	35

7.9 Monitoring Automation Files

The Monitoring Automation files are created in the AWS Sensor pack settings folder and contain the commands that allow to add the discovered AWS resources to PRTG.

After a successful discovery, a file will be generated in the AWS folder per each AWS subscription. The files are named according to the following format: <profile>Discovery.csv

7.10 Using the Monitoring Automation CLI

The AWS sensor pack contains a command line interface that automates the addition of AWS resources as sensors to the PRTG system.

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

AutoMonX_PRTG_Automation.INI

PRTG_USER=<prtg_administrative_user>

PRTG_SERVER=<prtg_server_name>

- You need a PRTG user with read and write permissions to operate the program.
- You will need to create a target group in PRTG that will contain the AWS resources sensors.

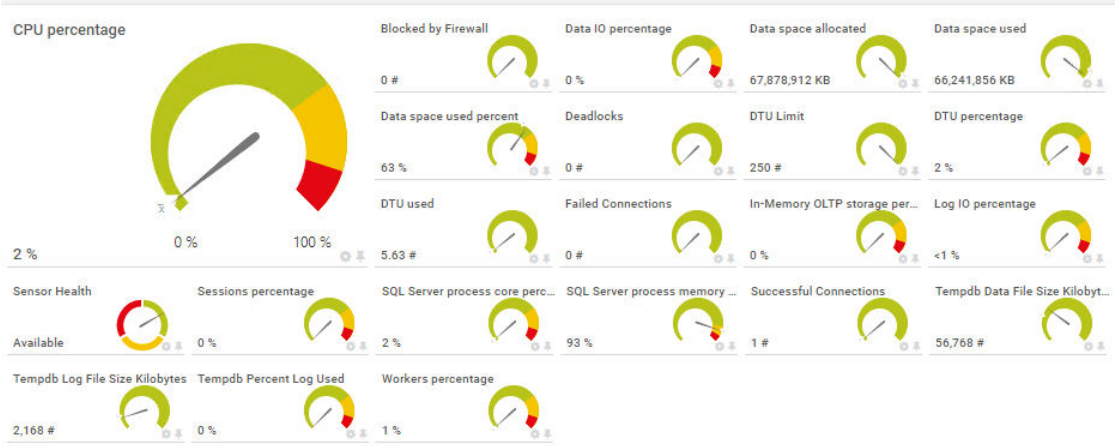
Below is an example how to use the Monitoring Automation CLI:

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

8 Supported sensor types

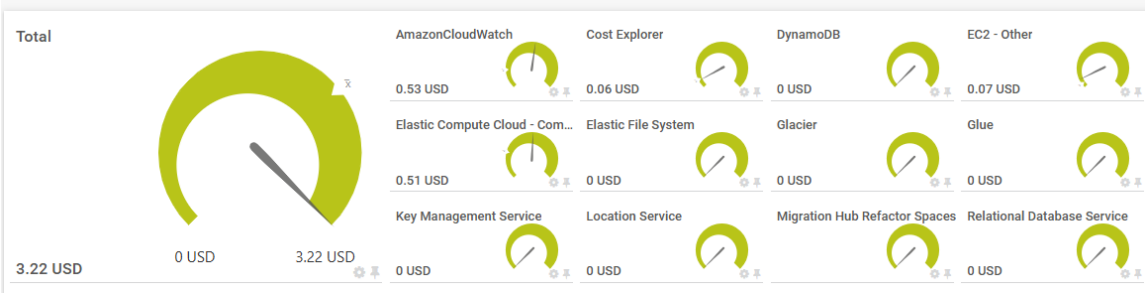
8.1 Alarm

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



8.2 Cost Explorer

Visualize, understand, and manage your AWS costs and usage over time

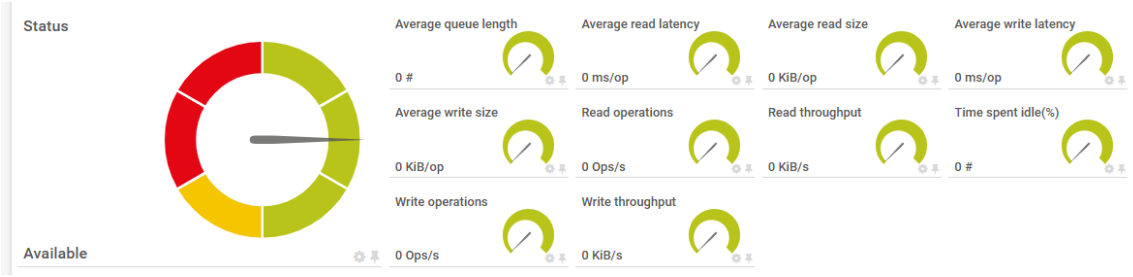


8.3 Dynamo DB

A managed NoSQL database service that provides fast and predictable performance with seamless scalability.

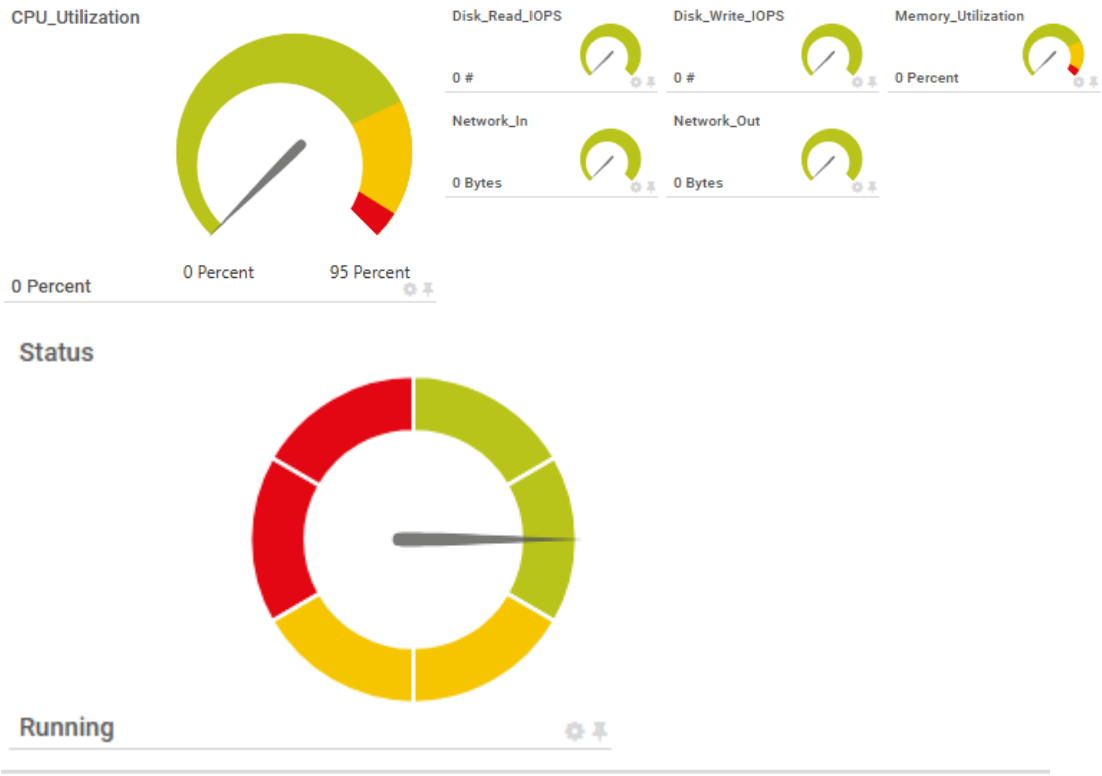
8.4 EBS - Elastic Block Store

Amazon Elastic Block Store (Amazon EBS) is an easy-to-use, scalable, high-performance block-storage service designed for Amazon Elastic Compute Cloud (Amazon EC2).



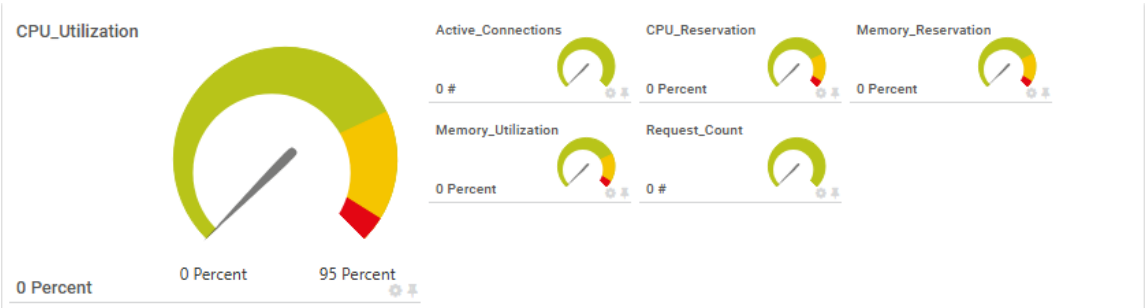
8.5 EC2 – Elastic Compute Cloud

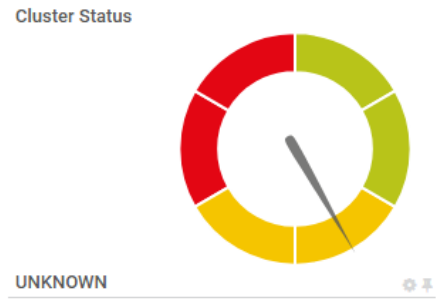
Provides resizable compute capacity in the cloud, commonly used for hosting applications, websites, and virtual servers.



8.6 ECS – Elastic Container Service

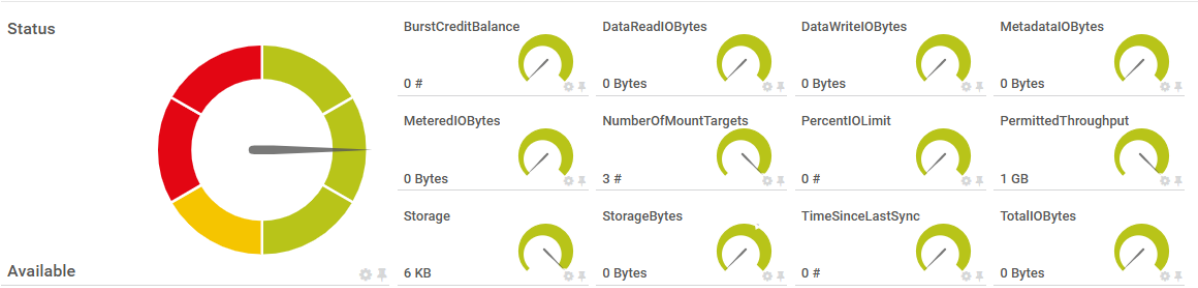
Enables you to run, stop, and manage Docker containers on a cluster of Amazon EC2 instances.





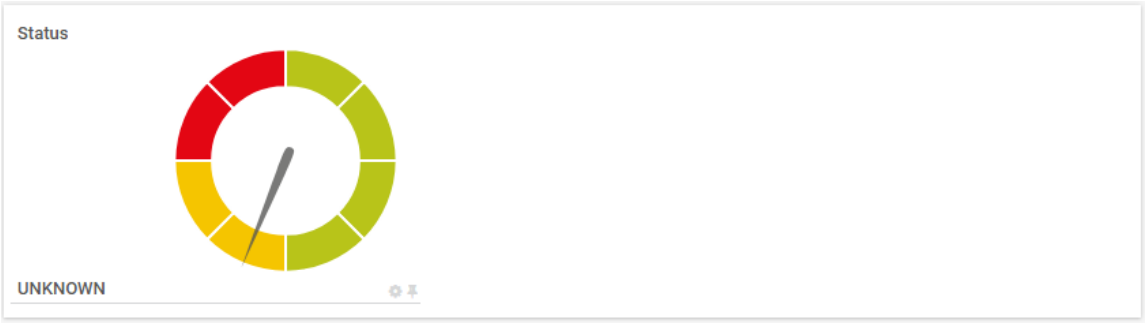
8.7 EFS – Elastic File System

Amazon Elastic File System (Amazon EFS) automatically grows and shrinks as you add and remove files with no need for management or provisioning.



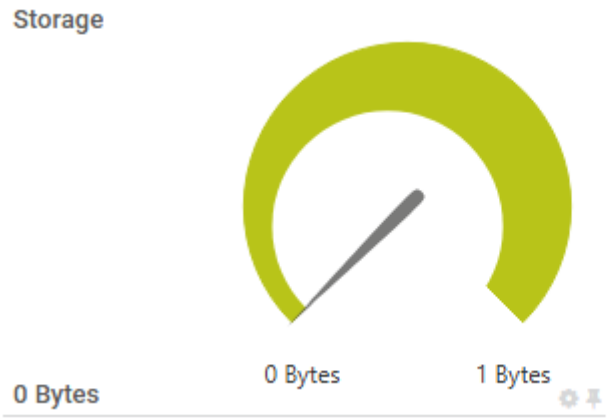
8.8 EMR – Elastic Map reduce

Provides managed Hadoop and Spark clusters for big data processing.



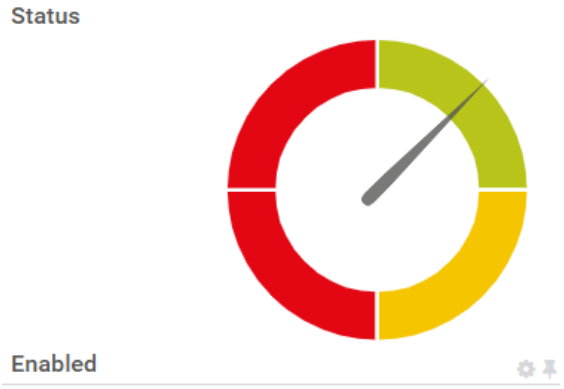
8.9 Glacier

Offers long-term archival storage with retrieval times ranging from minutes to hours.



8.10 KMS – Key Management System

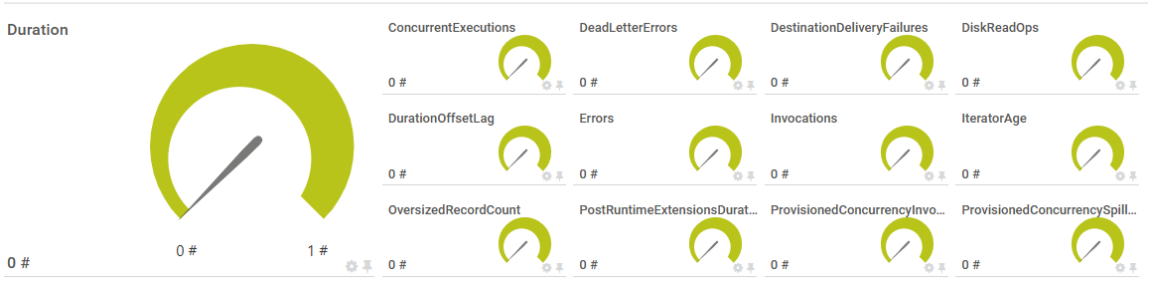
Create and control keys used to encrypt or digitally sign your data



Web : http://www.automonx.com
Email : support@automonx.com

8.11 Lambda

Enables serverless computing, allowing you to run code without provisioning or managing servers.

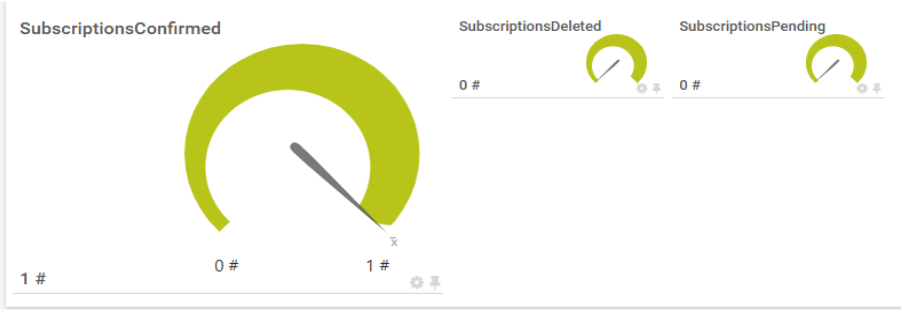


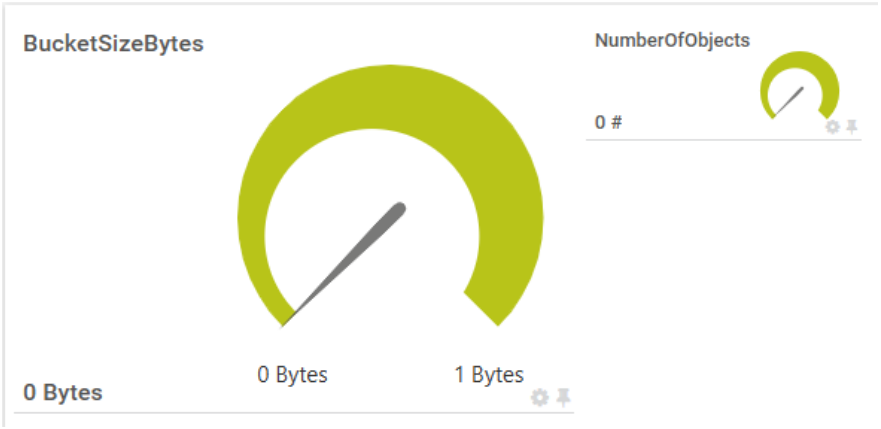
8.12 RDS – Relational Database Service

Provides managed database services for MySQL, PostgreSQL, SQL Server, and others.

8.13 S3 – Simple Storage Service

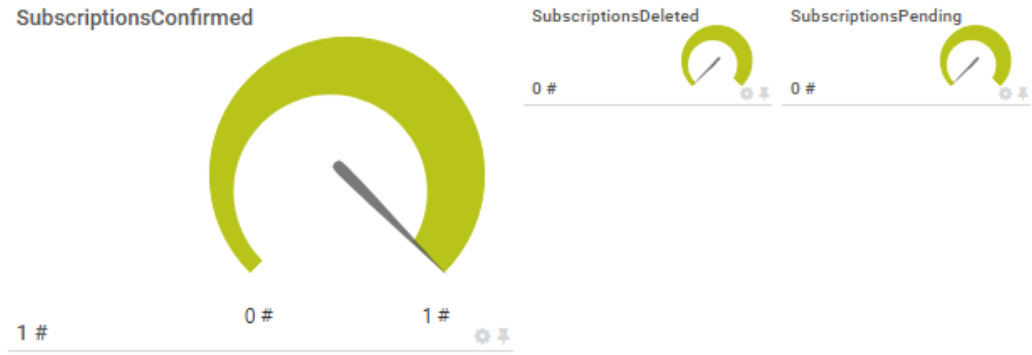
Offers scalable object storage for data backup, archiving, and serving static assets like images and videos.





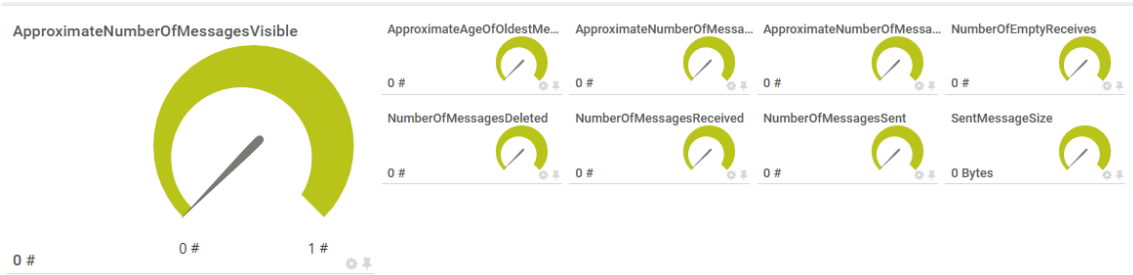
8.14 SNS – Simple Notification Service

Offers pub/sub messaging for building event-driven architectures.



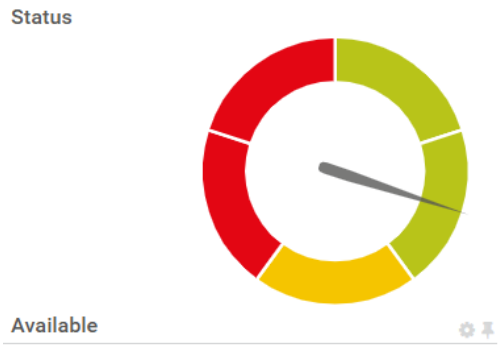
8.15 SQS – Simple Queue Service

Provides a fully managed message queue service for decoupling and scaling microservices, distributed systems, and serverless



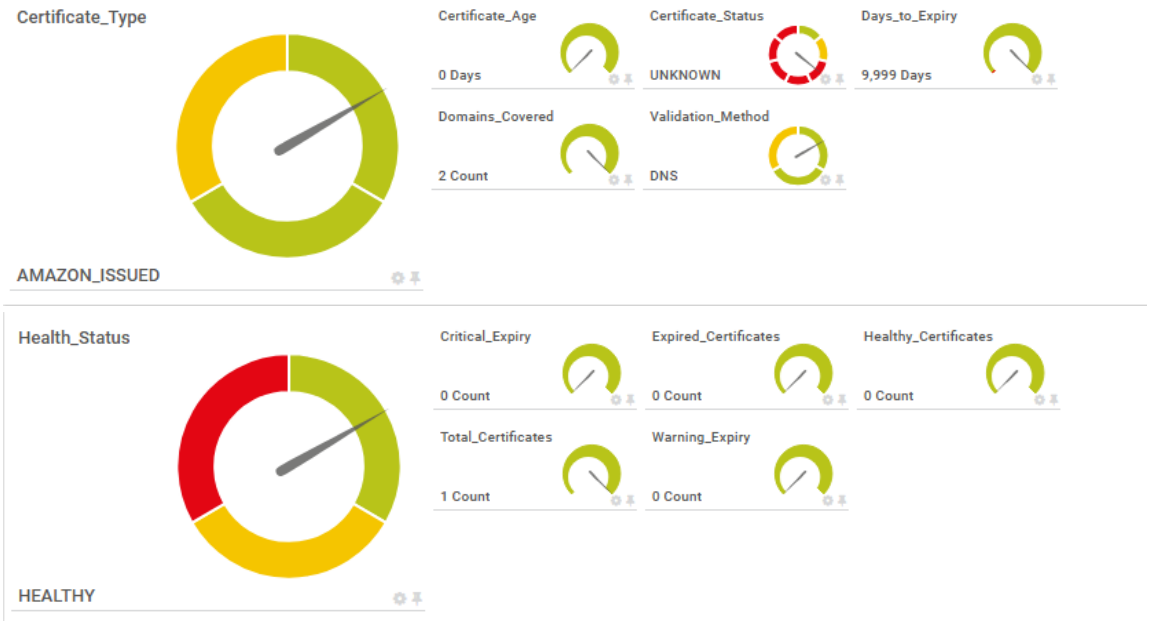
8.16 VPC – Virtual Private Cloud

Allows you to provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network.



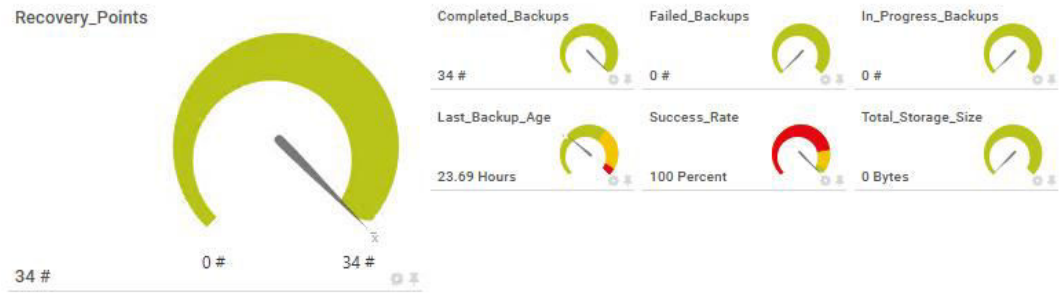
8.17 ACM – AWS Certificate Management

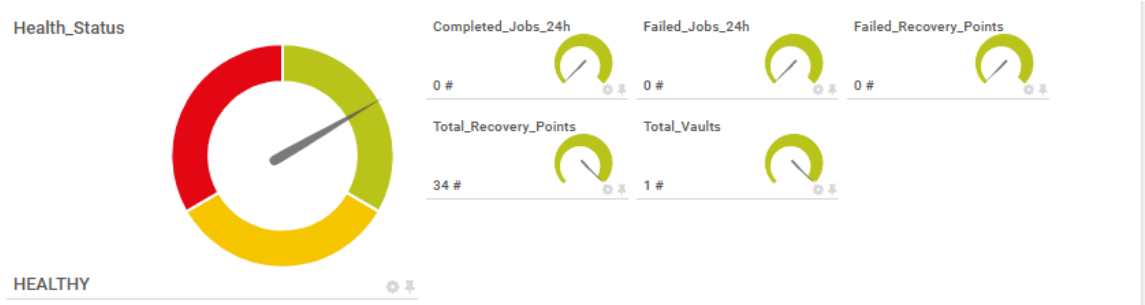
Monitor SSL/TLS certificates managed by ACM



8.18 Backup

Monitor AWS Backup vaults and backup operations



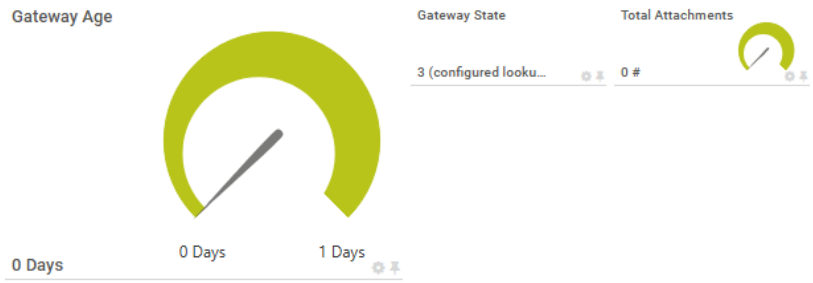


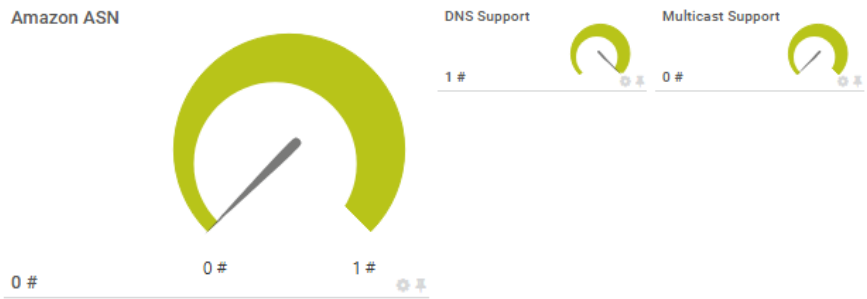
8.19 SES – Simple Email Service

Monitor SES sending configuration and identity / metrics for sending/complaints/bounces.

8.20 Transit Gateway

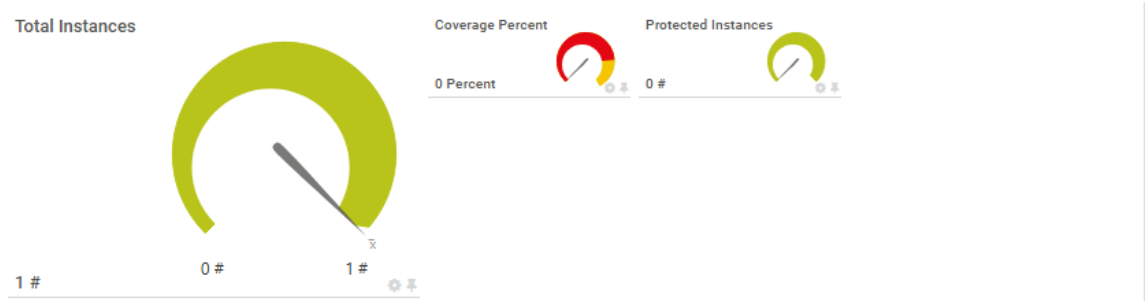
Monitor Transit Gateways and attachments (VPCs/VPNs/..) health and state.

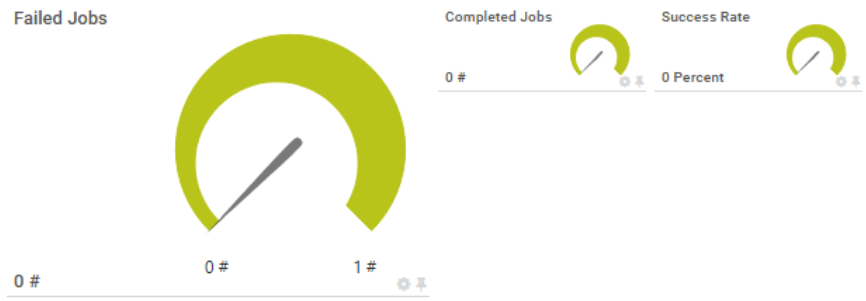




8.21 Windows Backup

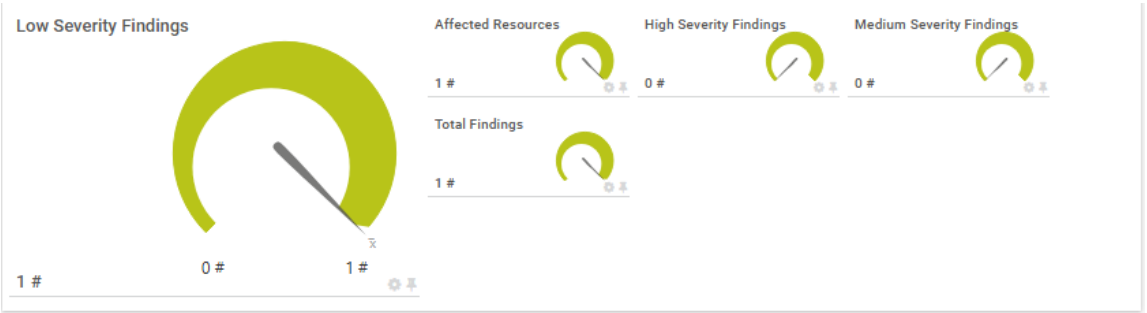
Provide Windows-server-like backup sensor using AWS Backup (coverage, recent job failures, protected instance counts).

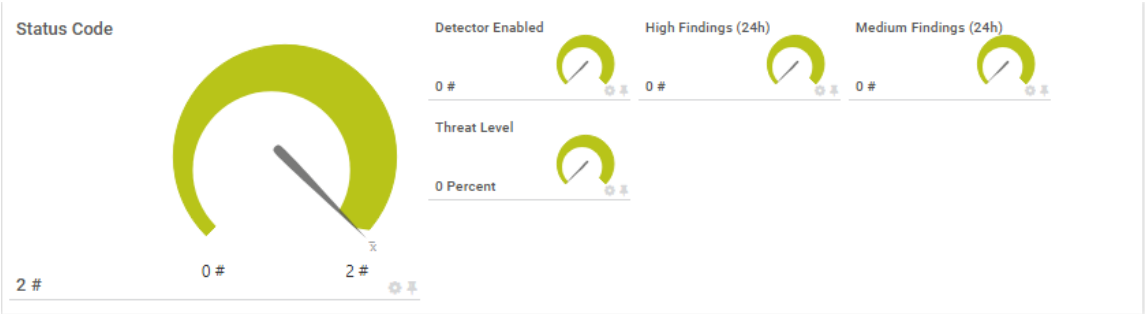




8.22 GuardDuty

Threat detection sensor — check detector status and count/highlight severity findings.





8.23 FSx

Comprehensive monitoring for FSx file systems (Windows, Lustre, ONTAP, OpenZFS), volumes, and storage virtual machines

8.24 APIGateway

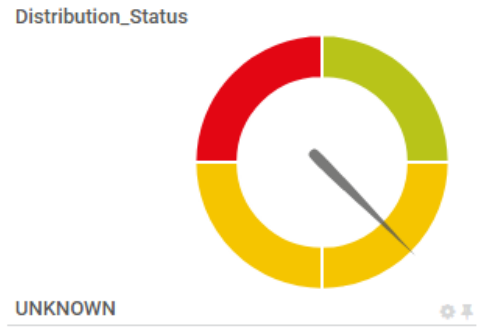
Monitors REST and HTTP APIs, stages, and deployments. Tracks endpoint health latency, throttling. Useful for availability of SLAs and catching misconfigurations after deployments.

8.25 Autoscaling

Observes Auto Scaling Groups, desired/min/max capacity, scaling activities, and health of instances. Highlights policy-triggered scale events, pending instance counts, and failures to launch/terminate.

8.26 CloudFront

Monitors distributions, enabled status.

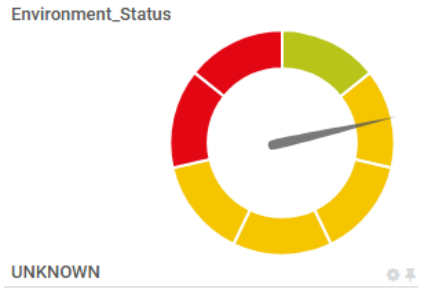


8.27 ElastiCache

Tracks Redis/Memcached clusters: node states, CPU/memory, eviction, replication lag, failover events, and parameter group compliance. Alerts on degraded nodes and insufficient reserved memory.

8.28 Elastic Beanstalk

Monitors application environments and health. Surfaces deployment state, instance health, and key EB metrics (latency, 5xx, CPU). Helps catch application misconfig or unhealthy rollouts.



8.29 Route53

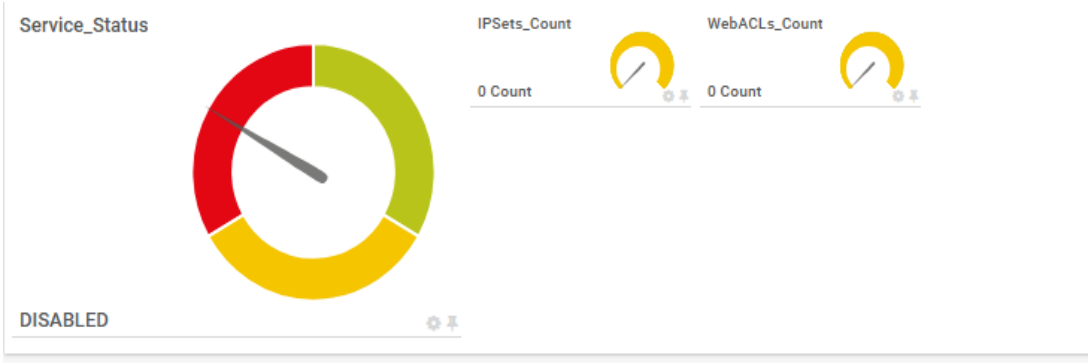
Watches hosted zones and record sets. Validates health checks and routing policies (weighted, latency, failover). Reports DNS change status and flags stale or failing health checks

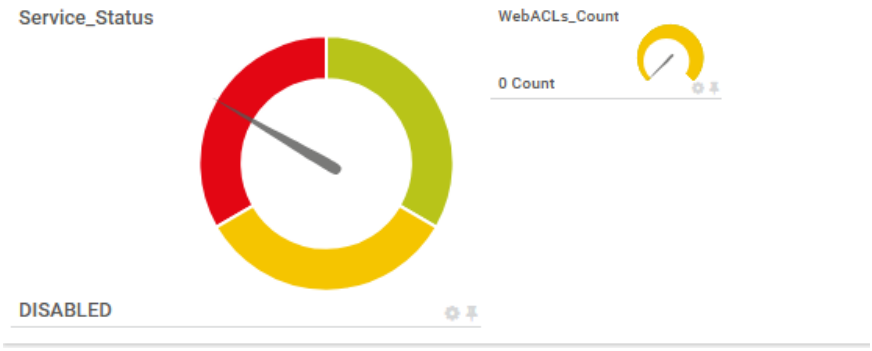
8.30 Redshift

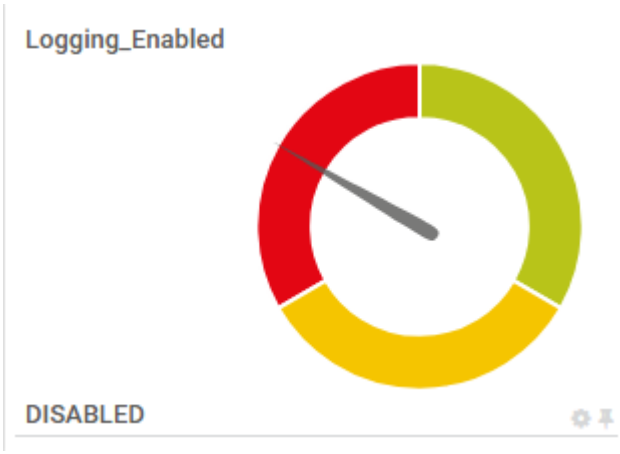
Monitors clusters: status, node health, storage, WLM queue pressure, query performance, and maintenance windows. Highlights connection/availability issues and unusual load patterns.

8.31 WAF (Web Application Firewall)

Monitor WAF configurations and protection status

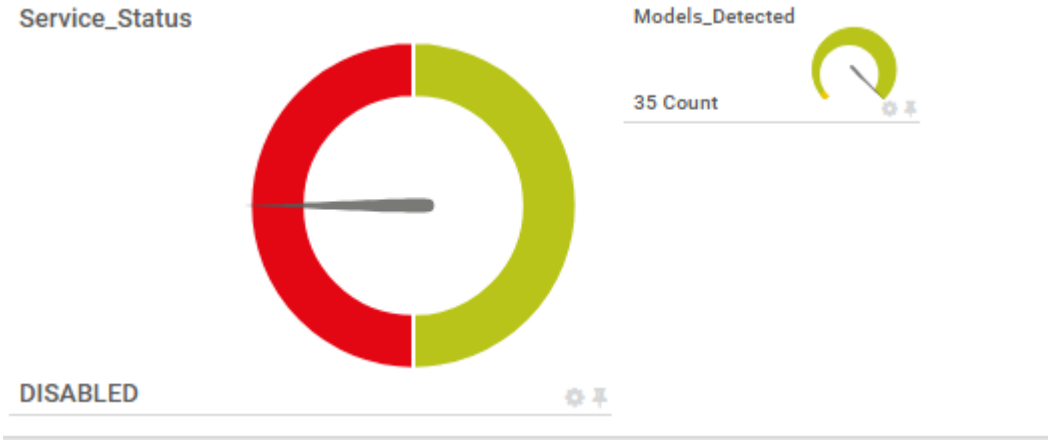






8.32 Bedrock

Monitor AWS Bedrock AI/ML foundation model service availability and performance



Web : http://www.automonx.com
Email : support@automonx.com

9 Troubleshooting

9.1 Troubleshooting the AWS Sensor pack Installation

Problem Description	Troubleshooting Steps
AWS Sensor pack Service is not starting	1. Run Check Config via the UI (**as administrator**), check the results and fix any problems. Refer to Troubleshooting the AWS Configuration 2. Make sure your AWS User is set up OK. Refer to Troubleshooting the AWS Connection Error 3. Make sure the PRTG probe is open to the Internet and can access AWS 4. Make sure that the Product Key is valid 5. Use the service debug mode to check service errors. Refer to Debug Using Service Debug Mode
Discovery is not providing any results	1. Make sure that the AWS sensor pack service is running 2. Make sure your AWS User has enough permissions to the desired subscription. Refer to Troubleshooting the AWS Connection Permission 3. Submit a support request via support@automonx.com and send the following log files: Refer to Sending the Discovery Files to the support team.
Discovery provides partial results	1. Make sure your AWS User has enough permissions to the desired subscription. Refer to Troubleshooting the AWS Connection Permission 2. Submit a support request via support@automonx.com and send the following log files: Refer to Sending the Discovery Files to the support team.
Discovery is not able to discover the AWS Cost resources	Make sure your AWS User has enough permissions to the desired Account. Refer to Troubleshooting the AWS Connection Permission
Discovery is not able to discover AWS App metrics sensors	Make sure your AWS User has enough permissions to the desired resource. Refer to Troubleshooting Missing AWS Resource Metrics

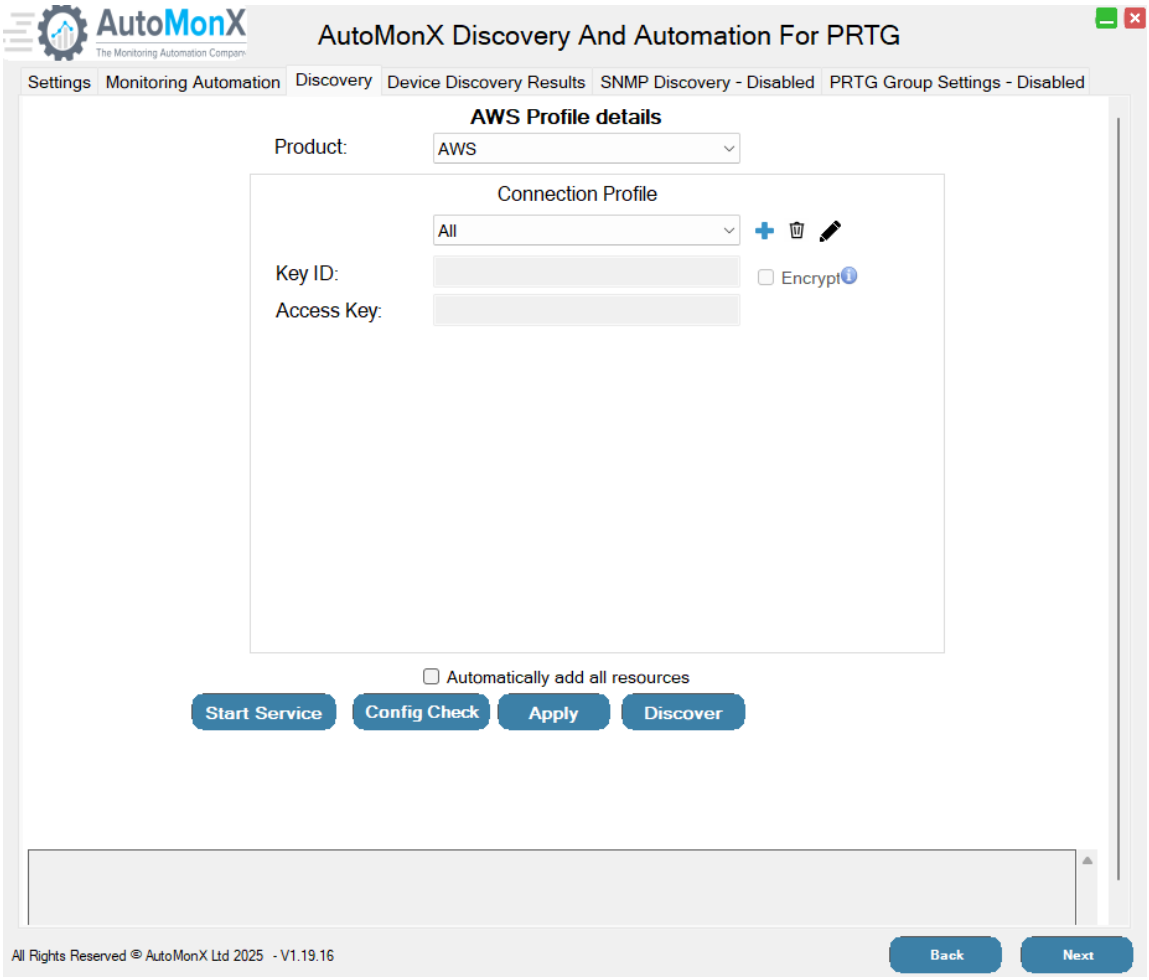
Discovery is not able to discover AWS Service Health metrics sensors	Make sure your AWS User has enough permissions to the desired resource Refer to Troubleshooting Missing Service Health
AWS Sensors are Down with error message: The AutoMonX AWS service is down, cannot connect	Make sure that the AWS connection parameters are correct (use the AutoMonX UI and run Config Check Make sure that AWS is not blocked by a proxy server or Firewall of your organization

9.2 Troubleshooting the AWS Sensor Configuration

In order to analyze the status of the connection to AWS and the AWS Sensor pack configuration, use one of the following options:

- AutoMonX Configuration UI
- Config Check command line utility

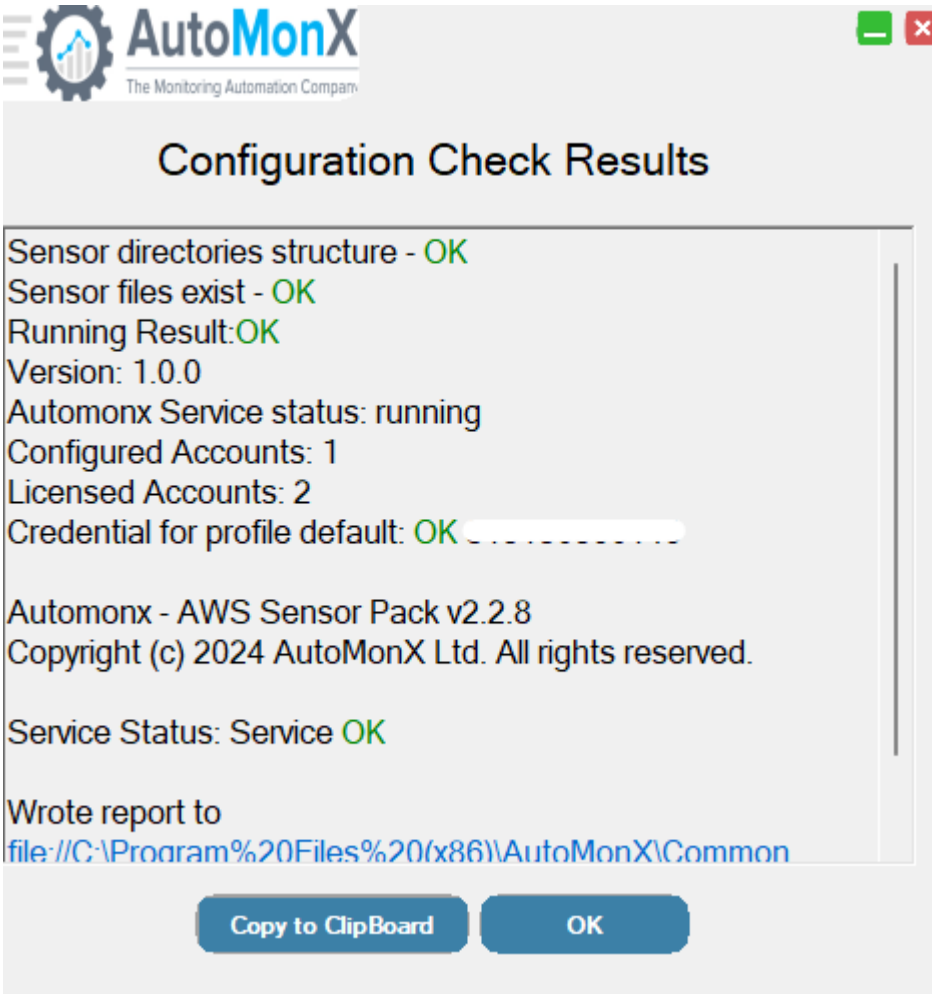
These tools perform various checks of the AWS sensor pack configuration, its service and the connection to AWS and displays vital information that may assist in checking for issues. Through the Automation UI “Config Check” Button:



Through command line: **Automonx_AWSCollector.exe -config_check**

Below is an example of a successful configuration check:

Web : http://www.automonx.com
Email : support@automonx.com



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

9.3 Troubleshooting AWS Discovery Connection Errors

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

Possible causes can be that AWS is unreachable due to limitations of network access, incorrect connection information such as Key Id or Access Key

Check the AWS sensor settings using the instructions in Section 6.3.

9.4 Troubleshooting AWS Discovery - Permissions

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

Please assign at least a **ReadOnly** role for the connection ID on the subscriptions you want to monitor. Please refer to Read Only User Creation

9.5 Collecting the Discovery Files for AutoMonX Support

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

- Discovery log file –
  - **Logs/Automonx_Discovery_out.log**
  - **Logs/Automonx_AWS.log**
- Discovery results in a form of CSV files with names such as:
  - **Data/<Account>Discovery.csv**
- The AWS types file:
  - **Types/Automonx_AWS_types.dat**

These files are located in the following directory**:**

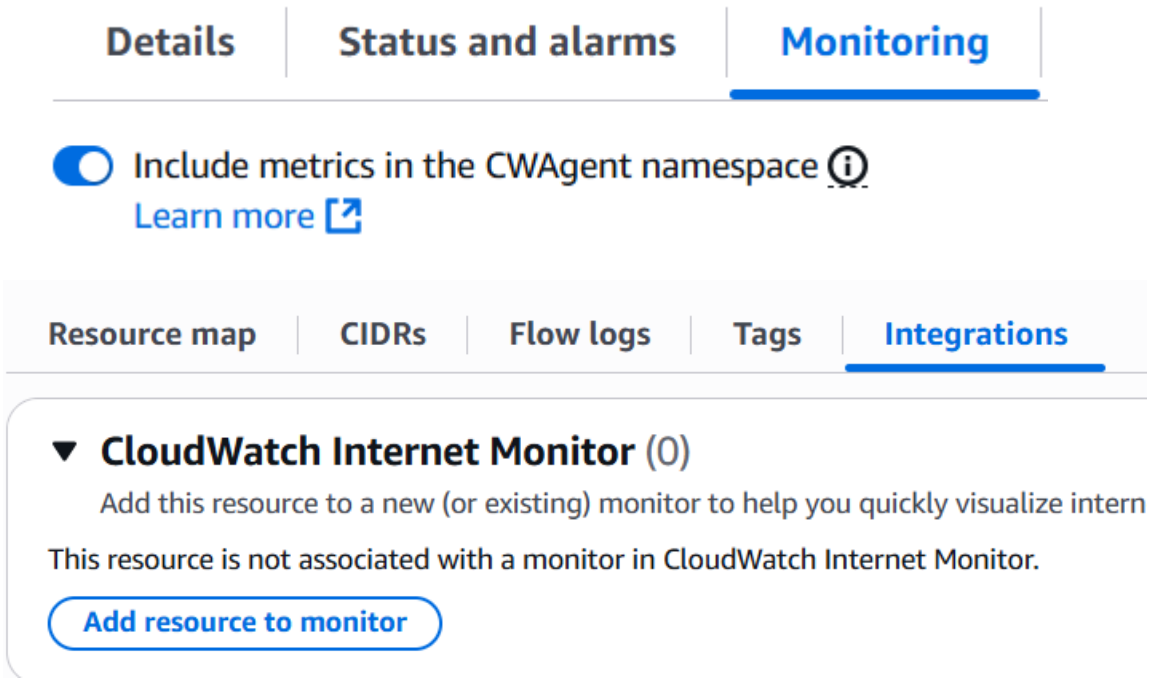
C:\Program Files (x86) \AutoMonX\SensorPacks\AWS

- The output of the following command:

  Automonx_AWSCollector.exe -discovery

9.6 Troubleshooting the Discovery of AWS Metrics

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



Some resource types do not support metrics.

9.7 Troubleshooting the Discovery of AWS Status

When the discovery process can't discover the AWS Status sensors, make sure status was enabled in the AWS portal for these resources.
Some resource types do not support the Status sensor.

9.8 Collecting AWS 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_AWSSensor.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 AWS 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. You would need to provide the following file:

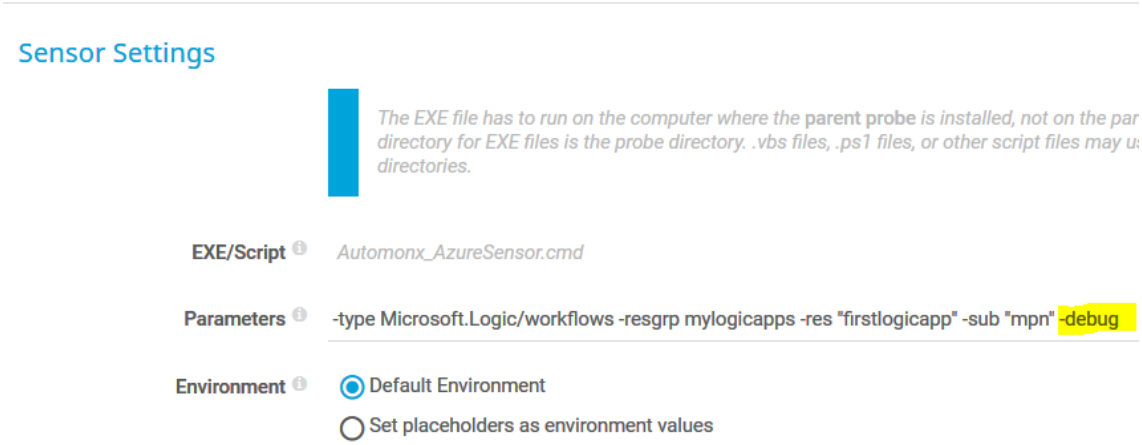
- Service Debugger file - **Automonx_AWSDebugLogger.log**

The files are in the following directory**:**

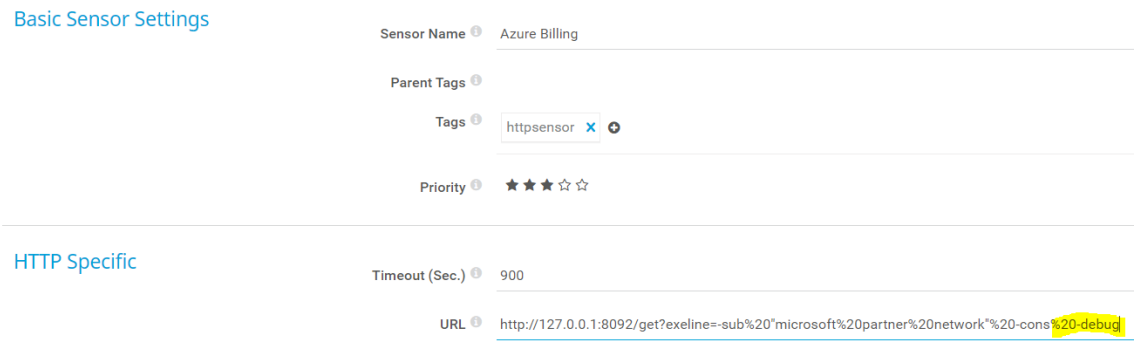
C:\Program Files (x86) \AutoMonX\SensorPacks\AWS

9.9 Collecting AWS Sensor Debug Information (future)

To activate the debug logs of the AWS sensors, add the -debug argument to the parameters of the AutoMonx_AWSSensor.cmd in the PRTG sensor settings as seen below:



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



This setting will activate the sensor debug mode upon the next run of the sensor. The AWS 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_AWSSensor.ini file:

DEBUG_LOG_DIR=C:\Temp

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

9.10 Collecting In-Depth AWS Sensor Debug Information

To activate the in-depth debug logs of the AWS sensors, add the following parameters to the AutoMonx_AWSSensor.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 AWS 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_AWSSensor.ini file:

DEBUG_LOG_DIR=C:\Temp

10 Command Line Options (CLI)

10.1 The AWS Sensor Pack Command Line Options Reference

Option	Details
-install	Installs the AutoMonX AWS Sensor Service. The service communicates with the AWS environments, retrieves and stores the AWS communication token.
-config_check	Checks the service communication to the AWS environment. Validates the license information
-discovery -discovery <profile name>	Discovers all the resources in an AWS environment and creates a report in a HTML format. The report is **AWS/Logs/AWSDiscovery.html**
-profile <profile name> -keyId <key ID> -accessKey <Secret Access key>	For multi-Account users. Creates a new credentials set as encrypted in the file: C:\Windows\System32\config\systemprofile\.aws\credentials Or %Userprofile%\.aws\credentials And the encrypted content is saved into AWS/creds folder
-Service	Runs as a service, for internal use.
-version	Displays the program's version.
-help	Displays the command option list.

10.2 Fully Automated AWS Monitoring (future)

Starting with the next major version , fully automated monitoring of your AWS estate is available as part of the AWS 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.

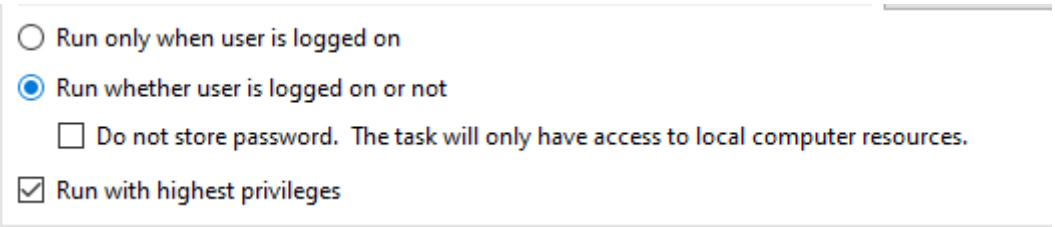
10.2.1 Automated Discovery and Monitoring

Depending on the structure of your AWS estate (Accounts, 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/AWS/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 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 damages direct or collateral 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 AWS resource was deleted or shut down via the AWS Portal.

Make sure that the file "down_sensors_filter.ini" exists in the Automonx/Common folder. Otherwise copy it from the latest AWS 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/AWS/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,”AWS 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 errors types, based on their last message text, you need to edit the *down_sensors_filter.ini* file.

Add 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, such state is seen because the AWS resource has been deleted or shut down via the AWS Portal. It is useful for NOC teams that wish to investigate the source of the problem rather than automatically delete the sensor(s).

Make sure that the file "pause_sensors_filter.ini" exists in the Automonx/Common folder. Otherwise copy it from the latest AWS 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_AWS

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,"AWS test").

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

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

Add 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-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 AWS resources that have been deleted.

Update the following configuration value in *AutoMonX_AWSSensor.ini* file and restart the service AWS 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/Exclusion of Sensors and Channels

Most of the AWS resources contain 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 AWS 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 AWS Resources and metrics you want to add to PRTG.

This section explains how to utilize the Exclude and Include Functions of AutoMonX AWS Sensor Pack to control the addition of AWS Resource Groups, AWS Resources (devices), AWS sensors and AWS 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 AWS 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\AWS\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 AWS Resource Group from Monitoring

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

The configuration below will exclude the group named Batch and any AWS 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 AWS Resource

Under the DeviceType_Category column enter the type of the AWS Resource (device) you wish to exclude. Use the resource type located in parentheses (i.e. *VirtualMachines* if the AWS Resource name *is Windowstest_(VirtualMachines)*), or part of the device name.

The configuration below will exclude AWS 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 AWS Sensor Type

In order to exclude specific AWS 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 AWS Resource type (device). Which would effectively not add this AWS resource to PRTG.

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

10.2.4.4 Excluding AWS Metrics (Channels)

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

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

10.2.4.5 Excluding by AWS Tags

If you utilize AWS Tags in your AWS estate, you can leverage them by adding the AWS 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 AWS Tags that are assigned directly to AWS resources in the AWS Portal can be used for filtering. AWS Tags set on a Resource group level will be ignored.

The configuration below excludes the all resources with AWS 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 AWS Resources, AWS Resources and AWS 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 AWS/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 in the top of the file you will see the number of filters applied, and if the sesnors passed of 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:AWS App Metrics - This Sensor would be skipped by the **exclude_mon.csv policy at line:2**

Skipping Sensor AWS App Metrics by Blacklist exclusion

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.

10.2.5 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_AWS

The feature is activated automatically by the AWS 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_senors_issues.exe -grouplist Automonx_AWS,"AWS test"

10.2.6 Automated Addition and Removal of Accounts

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

10.2.6.1 Adding a new Account:

The command below creates a new AWS Account connection profile in the file "AWSConnProfiles.ini".

Automonx_AWSCollector.exe -create_conn_profile -Account_label <display_label> -AWS_appid <app_id> -AWS_secretkey <secret_key> -AWS_Accountid <Account_id>

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

Automonx_AWSCollector.exe -config_check

10.2.6.2 Deleting a Account

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

Automonx_AWSCollector.exe -remove_Account -Account 1