



---

# Monitoring SQL Servers

Microsoft: SQL Server Enhanced PowerPack version 106

---

# Table of Contents

<b>Introduction</b> .....	<b>3</b>
What Does the Microsoft: SQL Server Enhanced PowerPack Monitor? .....	3
The Microsoft: SQL Server PowerPack vs the Microsoft: SQL Server Enhanced PowerPack .....	4
Installing the Microsoft: SQL Server Enhanced PowerPack .....	5
<b>Configuration and Discovery</b> .....	<b>7</b>
Prerequisites for Monitoring SQL Servers .....	8
Microsoft SQL Server Database Discovery Prerequisites .....	9
Creating a PowerShell SQL Server Credential .....	9
Creating a PowerShell SQL Server Credential in the SL1 Classic User Interface .....	11
SQL Cluster Monitoring .....	13
Monitoring SQL Clusters on SL1 8.12.1 or greater. ....	13
Monitoring SQL Clusters on SL1 8.8.1 to 8.12.0 .....	13
Discovering SQL Servers .....	14
Discovering SQL Servers in the SL1 Classic User Interface .....	17
Relationships Between Component Devices .....	18
SQL Cluster Node and SQL Cluster Instance Relationships .....	18
Viewing SQL Server Component Devices .....	20

---

# Chapter

# 1

## Introduction

---

### Overview

This manual describes how to monitor Microsoft SQL Servers in SL1 using the *Microsoft: SQL Server Enhanced PowerPack*.

The following sections provide an overview of SQL Servers and the *Microsoft: SQL Server Enhanced PowerPack*:

This chapter covers the following topics:

<a href="#">What Does the Microsoft: SQL Server Enhanced PowerPack Monitor?</a> .....	3
<a href="#">The Microsoft: SQL Server PowerPack vs the Microsoft: SQL Server Enhanced PowerPack</a> .....	4
<a href="#">Installing the Microsoft: SQL Server Enhanced PowerPack</a> .....	5

**NOTE:** ScienceLogic provides this documentation for the convenience of ScienceLogic customers. Some of the configuration information contained herein pertains to third-party vendor software that is subject to change without notice to ScienceLogic. ScienceLogic makes every attempt to maintain accurate technical information and cannot be held responsible for defects or changes in third-party vendor software. There is no written or implied guarantee that information contained herein will work for all third-party variants. See the End User License Agreement (EULA) for more information.

---

### What Does the Microsoft: SQL Server Enhanced PowerPack Monitor?

The *Microsoft: SQL Server Enhanced PowerPack* enables you to discover, model, and collect data about SQL 2012, 2014, 2016, 2017, and 2019 servers and their component devices.

The *Microsoft: SQL Server Enhanced PowerPack* includes:

- An example credential you can use to create PowerShell credentials to connect to SQL Servers
- Dynamic Applications to discover and monitor SQL Servers and their component devices
- Device Classes for each type of SQL Server component device monitored by SL1
- Event Policies and corresponding alerts that are triggered when SQL Servers and their component devices meet certain status criteria

## The Microsoft: SQL Server PowerPack vs the Microsoft: SQL Server Enhanced PowerPack

The following table describes the functions and differences between the *Microsoft: SQL Server PowerPack* and the *Microsoft: SQL Server Enhanced PowerPack*:

	<i>Microsoft: SQL Server</i>	<i>Microsoft: SQL Server Enhanced</i>
<b>Description</b>	This was the original PowerShell-based PowerPack to monitor SQL Servers.	This PowerPack was created to provide the ability to tune thresholds and suppress events on a per-database basis along with additional metrics being collected. It is a replacement to the other SQL PowerPack, not an add-on. Only <b>one</b> of these PowerPacks should be used to monitor an SQL Server.
<b>Dynamic Application Alignment</b>	All Dynamic Applications align to the Windows Server device.	The PowerPack builds a Device Component Map (DCM) tree below the Windows Server. The device components represent each SQL Instance on the server and each Database below the Instance.
<b>SL1 Agent</b>	The Dynamic Applications from this PowerPack can be aligned to SL1 Agents to monitor an SQL Server.	This PowerPack is not supported on SL1 Agents.
<b>Concurrent PowerShell</b>	This PowerPack can take advantage of the Concurrent PowerShell feature for improved collector scale.	This PowerPack does not leverage the Concurrent PowerShell feature.
<b>Merging Devices</b>	This PowerPack supports merging the Windows Server device with a virtual machine component in SL1.	This PowerPack does not support merging devices. Although the user interface does not prevent merging two DCM trees, it will break Dynamic Applications that depend on cached data from the root device.

<b>SQL Cluster</b>	This PowerPack does not include SQL Cluster support.	This PowerPack includes SQL Cluster support that is supplemented by the <i>Microsoft: Windows Server Cluster</i> PowerPack.
<b>Metrics Collected</b>	This PowerPack includes 8 Dynamic Applications that collect 79 metrics.	The PowerPack includes 21 Dynamic Applications that collect 157 metrics.
<b>Why do you have two SQL PowerPacks?</b>	Each of these PowerPacks have different strengths and weaknesses based on the Dynamic Application Type used (PowerShell vs. Snippet) and whether DCM is used.	
<b>Which PowerPack should I use?</b>	Use the <i>Microsoft: SQL Server</i> PowerPack if you intend to use the SL1 Agent, need increased collector scale, or plan to merge the SQL Server device with its matching virtual machine component. Use the <i>Microsoft: SQL Server Enhanced</i> PowerPack if you need the ability to control event thresholds or suppression on a per-database level, are monitoring an SQL Cluster, or want to sync the SQL Instance and Database components into a CMDB.	
<b>Is the Microsoft: SQL Server PowerPack still supported?</b>	Yes, it is still supported and maintained.	

## Installing the Microsoft: SQL Server Enhanced PowerPack

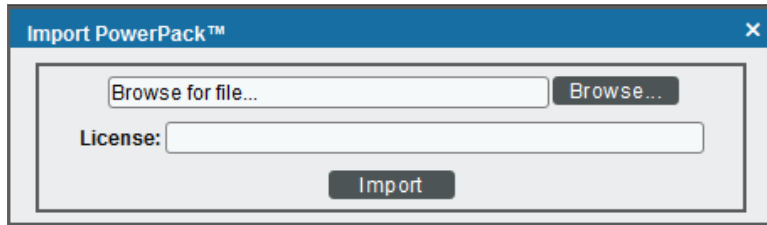
Before completing the steps in this manual, you must import and install the latest version of the *Microsoft: SQL Server Enhanced* PowerPack.

**TIP:** By default, installing a new version of a PowerPack overwrites all content from a previous version of that PowerPack that has already been installed on the target system. You can use the **Enable Selective PowerPack Field Protection** setting in the **Behavior Settings** page (System > Settings > Behavior) to prevent new PowerPacks from overwriting local changes for some commonly customized fields. (For more information, see the **System Administration** manual.)

To download and install a PowerPack:

1. Download the PowerPack from the ScienceLogic Support Site at <https://support.sciencelogic.com/s/powerpacks>.
2. Go to the **PowerPack Manager** page (System > Manage > PowerPacks).

3. In the **PowerPack Manager** page, click the **[Actions]** button, then select *Import PowerPack*. The **Import PowerPack** dialog box appears:



4. Click the **[Browse]** button and navigate to the PowerPack file.
5. When the **PowerPack Installer** modal appears, click the **[Install]** button to install the PowerPack.

**NOTE:** If you exit the **PowerPack Installer** modal without installing the imported PowerPack, the imported PowerPack will not appear in the **PowerPack Manager** page. However, the imported PowerPack will appear in the **Imported PowerPacks** modal. This page appears when you click the **[Actions]** menu and select *Install PowerPack*.

---

# Chapter

# 2

## Configuration and Discovery

---

### Overview

The following sections describe how to configure and discover Microsoft SQL Servers for monitoring in SL1 using the *Microsoft: SQL Server Enhanced PowerPack*:

This chapter covers the following topics:

<i>Prerequisites for Monitoring SQL Servers</i> .....	8
<i>Creating a PowerShell SQL Server Credential</i> .....	9
<i>SQL Cluster Monitoring</i> .....	13
<i>Discovering SQL Servers</i> .....	14
<i>Relationships Between Component Devices</i> .....	18
<i>Viewing SQL Server Component Devices</i> .....	20

**NOTE:** If you already have Windows Server discovered, you might not need to create a new SQL Server credential or run a separate discovery session for SQL Servers if the PowerShell credential information is the same as that used for the Windows Server credential. In this scenario, you need only to install the *Microsoft: SQL Server Enhanced PowerPack* and ensure that the Windows user account used in the credential has the appropriate permissions, as outlined in the [Prerequisites](#) section.

---

## Prerequisites for Monitoring SQL Servers

To configure the SL1 system to monitor SQL servers using the *Microsoft: SQL Server Enhanced PowerPack*, you must first have the following information about the SQL Servers that you want to monitor:

- IP addresses and ports for the SQL Servers
- Username and password for a Windows user account with access to the SQL Servers

The SQL Servers that you monitor must be running PowerShell version 3.0 or later and need to have the SQL Server PowerShell (SQLPS) module installed. This SQLPS module is installed by SQL Server Management Studio. You can also install the SqlServer PowerShell module found here:

<https://www.powershellgallery.com/packages/SqlServer/21.1.18218>

The `InvokeSqlCmd` cmdlet must be present on the server and is available in the SQLPS and SqlServer PowerShell modules mentioned above. To determine if the proper cmdlets are available for this PowerPack to collect, run `Get-Command Invoke-SqlCmd` to see if the `Invoke-SqlCmd` cmdlet is installed.

In addition, the *Microsoft: SQL Server Enhanced PowerPack* requires the following permissions for the user account used for monitoring:

- SQL 2014 and newer versions require one of the following configurations:
  - The user account has an enabled login on every instance and database to be monitored, with `CONNECT SQL`, `VIEW SERVER STATE`, and `CONNECT ANY DATABASE` permission granted to the login on each instance. The login should have `VIEW DATABASE STATE` permission and `DB_DATAREADER` role granted on the 'master' database, and the `DB_DATAREADER` role granted on the 'msdb' database.
  - The user account has an enabled login on every instance and has the `SYSADMIN` role.
- SQL 2012 requires one of the following configurations:
  - The user account has an enabled login on every instance and database to be monitored, with `CONNECT SQL` and `VIEW SERVER STATE` granted to the login on each instance. The login should also have `VIEW DATABASE STATE` permission and the `DB_DATAREADER` role granted on the 'master' database, and the `DB_DATAREADER` role granted on the 'msdb' database. In addition, every database in the instance should have `CONNECT` access granted to the login.
  - The user account has an enabled login on every instance and has the `SYSADMIN` role.

ScienceLogic provides a PowerShell script on that automates the permissions-granting that is required as stated above. The script is included with the *Microsoft: Windows Server PowerPack*.

To use the PowerShell script, perform the following steps:

1. When you download the *Microsoft: Windows Server PowerPack* from the [ScienceLogic Support](#) site, a .zip file for the **WinRM Configuration Wizard Script(winrm\_configuration\_wizard.ps1)** will be in the folder with the PowerPack's EM7PP file.



2. Copy the **WinRM Configuration Wizard Script** .zip file to the Windows server where Microsoft SQL Server is installed and from which you will be collecting data. Unzip the file
3. Using the credentials for an account that is a member of the Administrator's group, log in to the Windows server you want to monitor. You can log in directly or use Remote Desktop to log in.
4. Copy the PowerShell script named **winrm\_configuration\_wizard** to the Windows server that you want to monitor with SL1.
5. Right-click on the PowerShell icon and select **Run As Administrator**.
6. At the PowerShell prompt, navigate to the directory where you copied the PowerShell script named **winrm\_configuration\_wizard**.
7. At the PowerShell prompt, enter the following to enable execution of the script:

```
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope Process  
-Force
```

**NOTE:** The execution policy setting persists only during the current PowerShell session.

8. After the warning text, select Y.

**NOTE:** If your Windows configuration requires further steps to allow execution of the script, PowerShell will display prompts. Follow the prompts.

9. To set the required, least-privileged permissions for the user account SL1 will use to monitor all SQL Server instances and databases on the server, run the following script:

```
.\winrm_configuration_wizard.ps1 -user <domain>\<username> -sql_only
```

## Microsoft SQL Server Database Discovery Prerequisites

To discover SQL databases, users must have grant permissions to connect to the "master" and "ms\_db" database, and permissions to create a new login on each SQL instance monitored.

You can use the following script to configure these permissions for monitoring SQL servers:

[https://code.eng.sciencelogic.com/projects/TOOL/repos/powershell/browse/winrm\\_configuration\\_wizard.ps1](https://code.eng.sciencelogic.com/projects/TOOL/repos/powershell/browse/winrm_configuration_wizard.ps1)

---

## Creating a PowerShell SQL Server Credential

To configure SL1 to monitor SQL Servers, you must first create a PowerShell credential. This credential allows the Dynamic Applications in the *Microsoft: SQL Server Enhanced PowerPack* to connect with an SQL Server. An example PowerShell credential that you can edit for your own use is included in the PowerPack.

**NOTE:** If you are using an SL1 system prior to version 11.1.0, the new user interface does not include the **Duplicate** option for sample credential(s). ScienceLogic recommends that you use [the classic user interface and the Save As button](#) to create new credentials from sample credentials. This will prevent you from overwriting the sample credential(s).

To create a PowerShell credential for an SQL Server:

1. Go to the **Credentials** page (Manage > Credentials).
2. Locate the **SQL PowerShell - Example** credential, click its **[Actions]** icon (⋮) and select **Duplicate**. A copy of the credential, called **SQL PowerShell - Example copy** appears.
3. Click the **[Actions]** icon (⋮) for the **SQL PowerShell - Example copy** credential and select **Edit**. The **Edit Credential** modal page appears:

The screenshot shows the 'Edit Credential' modal page. The main form has the following fields and controls:

- Name:** SQL PowerShell - Example copy
- All Organizations:** Toggle is on (blue). Below it is a dropdown menu labeled 'Select the organizations the credential belongs to \*'.
- Timeout (ms):** 180000
- Account Type:** Active Directory (dropdown)
- Encrypted:** Toggle is off (gray)
- Username:** USER\_NAME\_GOES\_HERE
- Password:** \*\*\*\*\*
- Hostname/IP:** %D
- Port:** 5986
- PowerShell Proxy Hostname/IP:** (empty field)
- Active Directory Host/IP:** AD\_HOSTNAME\_GOES\_HERE
- Active Directory Domain:** DOMAIN\_GOES\_HERE

At the bottom right of the main form is a blue button labeled 'Save & Test'. At the bottom right of the modal is a 'Save & Close' button.

On the right side, there is a 'Credential Tester' panel with the following controls:

- Select Credential Test:** (dropdown menu)
- Select Collector:** (dropdown menu)
- IP or Hostname to test \*:** (text input field)
- Test Credential:** (button)

4. Supply values in the following fields:
  - **Name.** Type a new name for your SQL Server credential.
  - **All Organizations.** Toggle on (blue) to align the credential to all organizations, or toggle off (gray) and then select one or more specific organizations from the **What organization manages this service?** drop-down field to align the credential with those specific organizations.
  - **Timeout (ms).** Type "18000".
  - **Account Type.** Select *Active Directory* from the dropdown.
  - **Encrypted.** Toggle this setting off.
  - **Username.** Type the username for a Windows user with access to the SQL Server.
  - **Password.** Type the password for the Windows account username.

**NOTE:** The user account whose username and password are provided in the credential must have certain permissions in all SQL Server instances that SL1 will monitor. For a list of these permissions, see the [Prerequisites](#) section.

- **Hostname/IP.** Type "%D".
- **Port.** Type "5985".
- **PowerShell Proxy Hostname/IP.** Leave this field blank.
- **Active Directory Host/IP.** Specify the hostname or IP address of the Active Directory server that will authenticate the credential.
- **Active Directory Domain.** Specify the domain where the monitored SQL Server resides.


5. Click **[Save & Close]**.

**NOTE:** The PowerShell credential test is not supported by the *Microsoft: SQL Server Enhanced PowerPack*.

## Creating a PowerShell SQL Server Credential in the SL1 Classic User Interface

To configure SL1 to monitor SQL Servers, you must first create a PowerShell credential. This credential allows the Dynamic Applications in the *Microsoft: SQL Server Enhanced PowerPack* to connect with an SQL Server. An example PowerShell credential that you can edit for your own use is included in the PowerPack.

To create a PowerShell credential for an SQL Server:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. Locate the **SQL PowerShell - Example** credential, and then click its wrench icon (). The **Edit PowerShell Credential** modal page appears.

3. Complete the following fields:

The screenshot shows the 'Credential Editor [72]' window. The subtitle is 'Edit PowerShell Credential #72'. There are 'New' and 'Reset' buttons in the top right. The 'Basic Settings' section contains the following fields: Profile Name (SQL PowerShell - Example), Account Type ([ Active Directory ]), Hostname/IP (%D), Timeout(ms) (180000), Username (USER\_NAME\_GOES\_HERE), Password (masked with dots), Encrypted (no), Port (5985), and PowerShell Proxy Hostname/IP. The 'Active Directory Settings' section contains Active Directory Hostname/IP (AD\_HOSTNAME\_GOES\_HERE) and Domain (DOMAIN\_GOES\_HERE). At the bottom are 'Save' and 'Save As' buttons.

- **Profile Name.** Type a new name for your SQL Server credential.
- **Account Type.** Select *Active Directory*.
- **Hostname/IP.** Type "%D".
- **Timeout.** Type "18000".
- **Username.** Type the username for a Windows user with access to the SQL Server.
- **Password.** Type the password for the Windows account username.

**NOTE:** The user account whose username and password are provided in the credential must have certain permissions in all SQL Server instances that SL1 will monitor. For a list of these permissions, see the [Prerequisites](#) section.

- **Encrypted.** Select *no*.
- **Port.** Type "5985".
- **PowerShell Proxy Hostname/IP.** Leave this field blank.
- **Active Directory Hostname/IP.** Specify the hostname or IP address of the Active Directory server that will authenticate the credential.
- **Domain.** Specify the domain where the monitored SQL Server resides.

4. Click the **[Save As]** button.
5. When the confirmation message appears, click **[OK]**.

---



# SQL Cluster Monitoring

For SQL Clusters that only include SQL Instances in an Active/Active configuration, follow the steps in the [Discovering SQL Servers](#) section.



For SQL Clusters that include an SQL Instance in an Active/Passive configuration, additional discovery steps are required and listed below.

**NOTE:** SL1's Active/Passive SQL Instance monitoring leverages the SL1 GUID Component Identifier to allow the SQL Instance component and its child database components to move between SQL Servers during a failover. Adding this GUID Component Identifier on SL1 versions prior to 8.12.1 will create a duplicate SQL Instance component on any already discovered SQL Servers. To prevent this, the GUID Component Identifier is not used by default. The "Enable Active Passive Cluster Failover" threshold in the "Microsoft: SQL Server Cache and Discovery" Dynamic Application provides the option to use the GUID Component Identifier when enabled. A value of "0" in the **Threshold Value** disables Active/Passive cluster failover; a value of "1" enables it.

## Monitoring SQL Clusters on SL1 8.12.1 or greater.

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon () for the "Microsoft: SQL Server Cache and Discovery" Dynamic Application to open the **Dynamic Applications Properties Editor** page.
3. In the **[Thresholds]** tab, click the wrench icon () for the "Enable Active Passive Cluster Failover" threshold and change the **Threshold Value** to 1.
4. Click **[Save]**.
5. Follow the steps in the [Discovering SQL Servers](#) section on each Windows Server in the cluster.

## Monitoring SQL Clusters on SL1 8.8.1 to 8.12.0

1. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
2. Click the wrench icon () for the "Microsoft: SQL Server Cache and Discovery" Dynamic Application to open the **Dynamic Applications Properties Editor** page.
3. In the **[Properties]** tab, change the **Operational State** field to *Disabled*.
4. Click **[Save]**.
5. Follow the steps in the [Discovering SQL Servers](#) section on each Windows Server in the cluster.
6. Go to the **Device Components** page (Registry > Devices > Device Components).
7. Click the wrench icon () for one of the Windows Servers that make up the SQL Cluster to open its **Device Properties** page.

8. In the **[Thresholds]** tab, under **Dynamic App Thresholds | Microsoft: SQL Server Cache and Discovery**, change **Enable Active Passive Cluster** to **1**.
9. Repeat steps 7 and 8 for each of the Windows Servers that make up the SQL Cluster.
10. Go to the **Dynamic Applications Manager** page (System > Manage > Applications).
11. Click the wrench icon (🔧) for the "Microsoft: SQL Server Cache and Discovery" Dynamic Application to open the **Dynamic Applications Properties Editor** page.
12. In the **[Properties]** tab, change the **Operational State** field to **Enabled**.
13. Click **[Save]**.

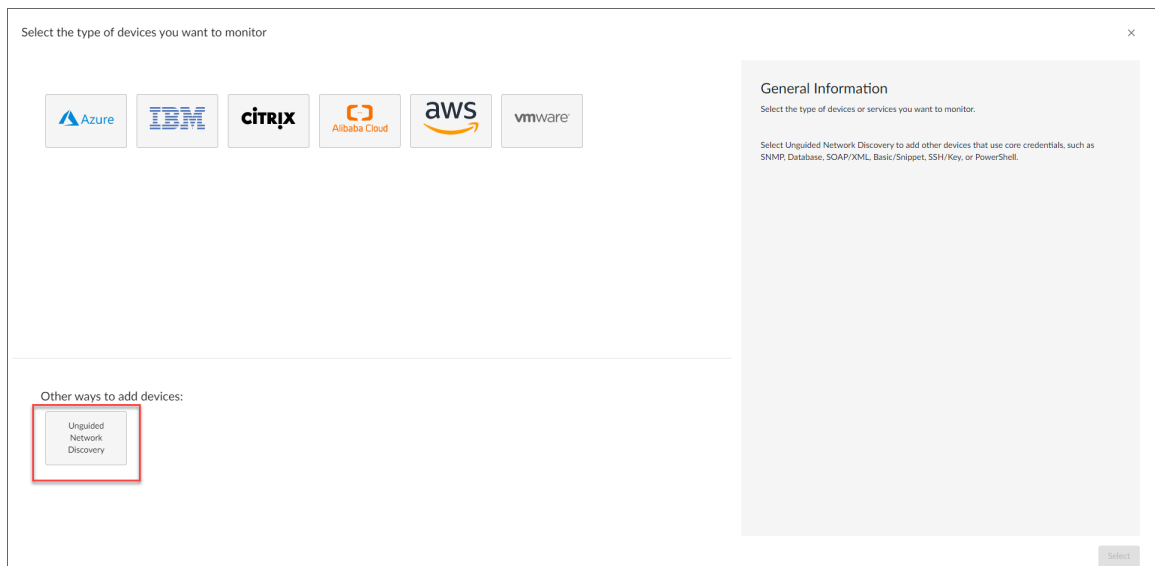
## Discovering SQL Servers

When you discover SQL Servers in SL1, SL1 auto-aligns a series of Dynamic Applications to discover, configure, and monitor the following SQL Server component devices:

- SQL Servers
  - SQL Server instances
  - SQL Server databases

To discover SQL Servers and their component devices, perform the following steps:

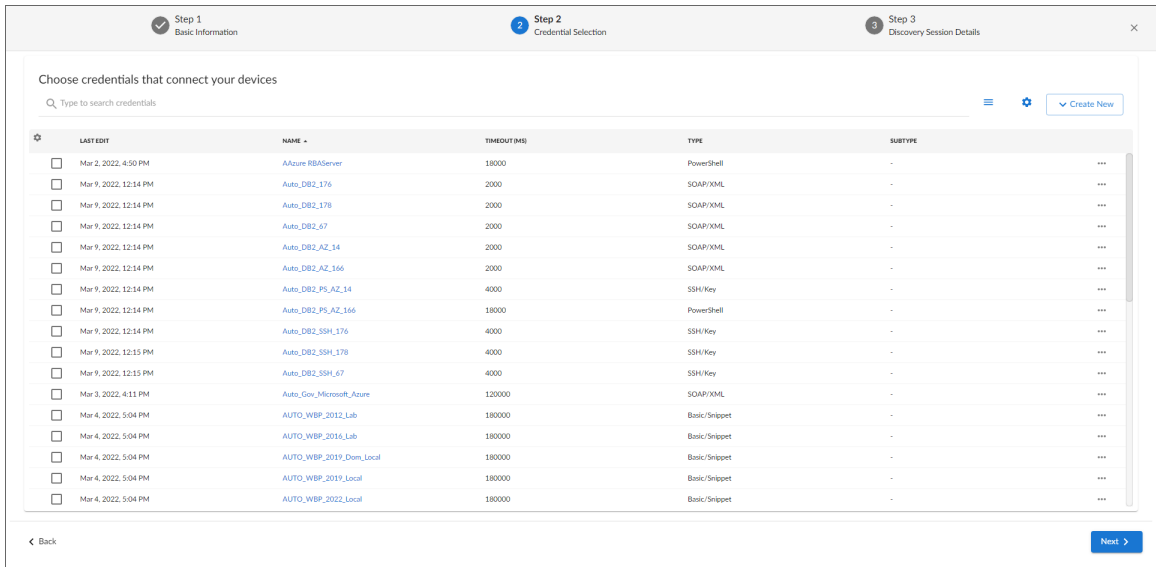
1. On the **Devices** page (🖨️) or the **Discovery Sessions** page (Devices > Discovery Sessions), click the **[Add Devices]** button. The **Select** page appears:



2. Click the **[Unguided Network Discovery]** button. Additional information about the requirements for discovery appears in the **General Information** pane to the right.

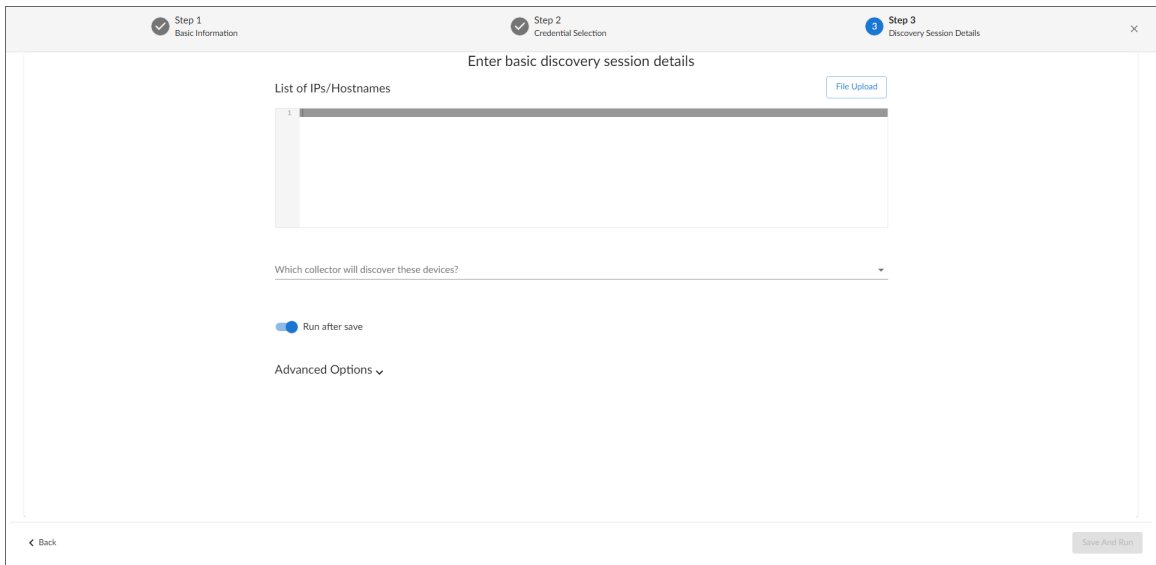
3. Click **[Select]**. The **Add Devices** page appears.
4. Complete the following fields:
  - **Name**. Type a unique name for this discovery session. This name is displayed in the list of discovery sessions on the **[Discovery Sessions]** tab.
  - **Description**. Optional. Type a short description of the discovery session. You can use the text in this description to search for the discovery session on the **[Discovery Sessions]** tab.
  - **Select the organization to add discovered devices to**. Select the name of the organization to which you want to add the discovered devices

5. Click **[Next]**. The **Credentials** page of the **Add Devices** wizard appears:



6. On the **Credentials** page, locate and select the **PowerShell credential** you created.

7. Click **[Next]**. The **Discovery Session Details** page of the **Add Devices** wizard appears:



8. Complete the following fields:

- **List of IPs/Hostnames.** Type the IP addresses or the range of IP addresses for the SQL Servers you want to discover.
- **Which collector will monitor these devices?.** Required. Select an existing collector to monitor the discovered devices.
- **Run after save.** Select this option to run this discovery session as soon as you save the session.



In the **Advanced options** section, click the down arrow icon (▼) to complete the following fields:

- **Discover Non-SNMP**. Enable this setting.
9. Click **[Save and Run]** if you enabled the Run after save setting, or **[Save and Close]** to save the discovery session. The **Discovery Sessions** page (Devices > Discovery Sessions) displays the new discovery session.
  10. If you selected the **Run after save** option on this page, the discovery session runs, and the **Discovery Logs** page displays any relevant log messages. If the discovery session locates and adds any devices, the **Discovery Logs** page includes a link to the **Device Investigator** page for the discovered device.

## Discovering SQL Servers in the SL1 Classic User Interface

When you discover SQL Servers in SL1, SL1 auto-aligns a series of Dynamic Applications to discover, configure, and monitor the following SQL Server component devices:

- SQL Servers
  - SQL Server instances
  - SQL Server databases

To discover SQL Servers and their component devices, perform the following steps:

1. Go to the **Discovery Control Panel** page (System > Manage > Classic Discovery).
2. Click the **[Create]** button. The **Discovery Session Editor** page appears:

The screenshot shows the 'Discovery Session Editor - Create New' interface. It features a blue header with 'New' and 'Reset' buttons. The main content area is divided into several sections:

- Identification Information:** Name: MySQL Example, Description: (empty).
- IP and Credentials:** IP Address/Hostname Discovery List: 10.40.3.4. Includes an 'Upload File' section with a 'Browse...' button.
- Detection and Scanning:** Initial Scan Level: System Default (recommended). Scan Throttle: System Default (recommended). Port Scan All IPs: System Default (recommended). Port Scan Timeout: System Default (recommended). Detection Method & Port: [ Default Method ]. A list of methods is shown, including UDP: 161 SNMP, TCP: 1 - logmux, TCP: 2 - compressnet, TCP: 3 - compressnet, TCP: 5 - rje, TCP: 7 - echo, TCP: 9 - discard, TCP: 11 - systat, TCP: 13 - daytime, TCP: 15 - netstat, TCP: 17 - qotd, TCP: 18 - msp, TCP: 19 - chargen, and TCP: 20 - ftp-data. Interface Inventory Timeout (ms): 600000. Maximum Allowed Interfaces: 10000. Bypass Interface Inventory: (unchecked).
- Basic Settings:** Discover Non-SNMP: (checked). Model Devices: (checked). DHCP: (unchecked). Device Model Cache TTL (h): 2. Collection Server PID: 50C-MUD-DCU-19. Organization: [ System ]. Add Devices to Device Group(s): None, LayerX Appliances, Servers. Apply Device Template: [ Choose a Template ].
- Other Credentials:** A list of credential types including EM7 Central Database, EM7 Collector Database, EM7 DB, LDAP/AD, QA-Silo AD, Power Shell, Lync 2010 Credentials - Example, SQL PowerShell - Example, Windows PowerShell - Example, and SOAP/XML Host.

At the bottom, there is a 'Save' button and a 'Log All' checkbox.

3. Supply values in the following fields:
  - **IP Address/Hostname Discovery List.** Type the IP addresses or the range of IP addresses for the SQL Servers you want to discover.
  - **Other Credentials.** Select the [PowerShell credential you created](#).
  - **Discover Non-SNMP.** Because the discovery session is not using an SNMP credential, select this checkbox.
4. Optionally, supply values in the other fields in this page. For a description of the fields in this page, see the **Discovery & Credentials** manual.
5. Click the **[Save]** button.
6. The **Discovery Control Panel** page will refresh. Click the lightning bolt icon (⚡) for the discovery session you created.
7. In the pop-up window that appears, click the **[OK]** button. The **Discovery Session** page displays the progress of the discovery session.

## Relationships Between Component Devices

SL1 can automatically build relationships between SQL servers and other associated devices:

- If you discover Windows server clusters using the Dynamic Applications in the *Microsoft: Windows Server Cluster PowerPack* version 100 or later, SL1 will automatically create relationships between SQL servers and Windows server clusters.



## SQL Cluster Node and SQL Cluster Instance Relationships

When discovering clustered SQL Server instances you will see multiple component device trees that represent the clustered SQL server. In these component device trees, the SQL servers are described as Nodes and the instances described as Roles.

Device Name	IP Address	Device Category	Device Class / Sub-class	DB	Organization	Current State	Collection Group	Collection State
Microsoft SQL Server	10.2.10.70	Application	Microsoft   SQL Server	3	System	Healthy	CLUG	Active
CLUS_INST01	--	Instance	Microsoft   SQL Server Instance	5	System	Major	CLUG	Active
CLUS_INST02	--	Instance	Microsoft   SQL Server Instance	4	System	Major	CLUG	Active
MSSQLSERVER	--	Instance	Microsoft   SQL Server Instance	6	System	Major	CLUG	Active
Microsoft SQL Server	10.2.10.71	Application	Microsoft   SQL Server	25	System	Healthy	CLUG	Active
CLUS_INST01	--	Instance	Microsoft   SQL Server Instance	27	System	Major	CLUG	Active
CLUS_INST02	--	Instance	Microsoft   SQL Server Instance	28	System	Major	CLUG	Active
MSSQLSERVER	--	Instance	Microsoft   SQL Server Instance	26	System	Major	CLUG	Active

**NOTE:** Discovering clustered SQL instances is the same process as discovering standalone SQL servers.

The following SQL cluster configurations are supported by the PowerPack. The relationship between the SQL cluster nodes and instances are described for each:

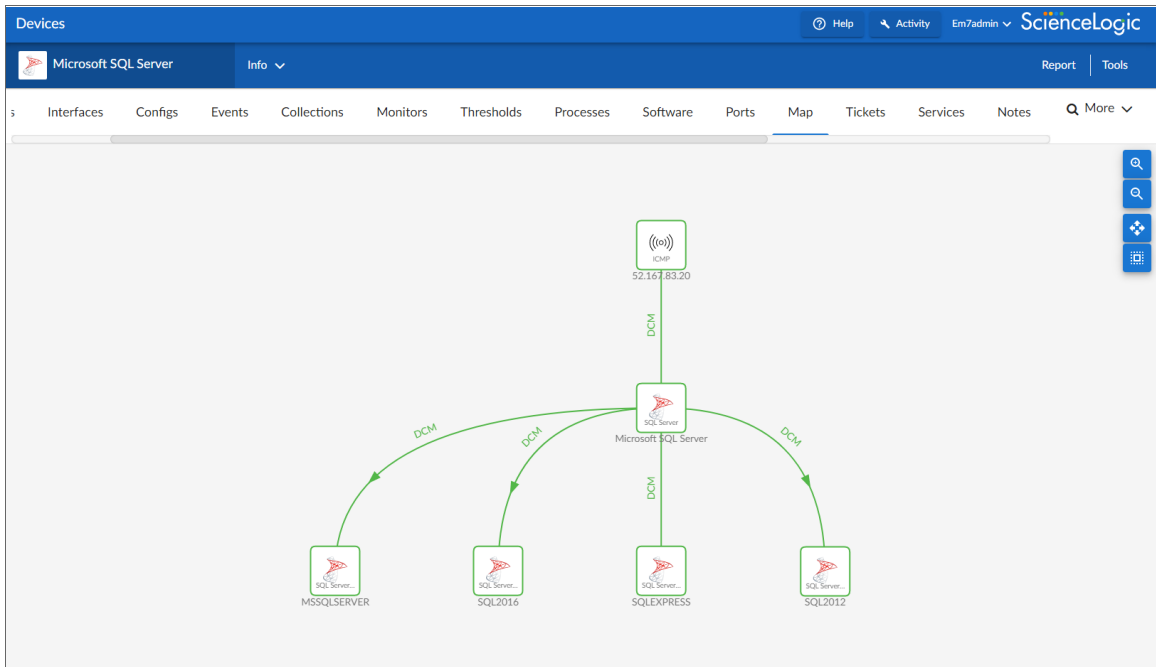
- **Active/Active.** The default configuration. After discovery, the roles (instances) will be modeled in both nodes (servers). To determine which instances belong to a specific node, go to the **Devices** page and click on the node device. Click the **[Configs]** tab from the **Device Investigator** and click "Microsoft: SQL Server Instance Discovery" in the pane on the left. There you will see which roles (instances) belong to the node.
- **Active/Passive.** To enable this configuration, you must find the "Microsoft: SQL Server Cache and Discovery" Dynamic Application in the **Dynamic Applications Manager** page (System > Manage > Dynamic Applications). Click its wrench icon () and then click the **[Thresholds]** tab. Click the wrench icon () for the **Enable Active Passive Cluster Failover** threshold object. In the **Override Threshold Value** dropdown, select *Enabled* and then click **[Save]**. This configuration is more intuitive, as the roles (instances) will be modeled only under the node (server) that they belong to, and in the event of a failover the instances will move between nodes.

**NOTE:** If a node has been stopped it won't be discovered.

# Viewing SQL Server Component Devices

In addition to the **Devices** page, you can view the SQL Server and all associated component devices in the following places in the user interface:

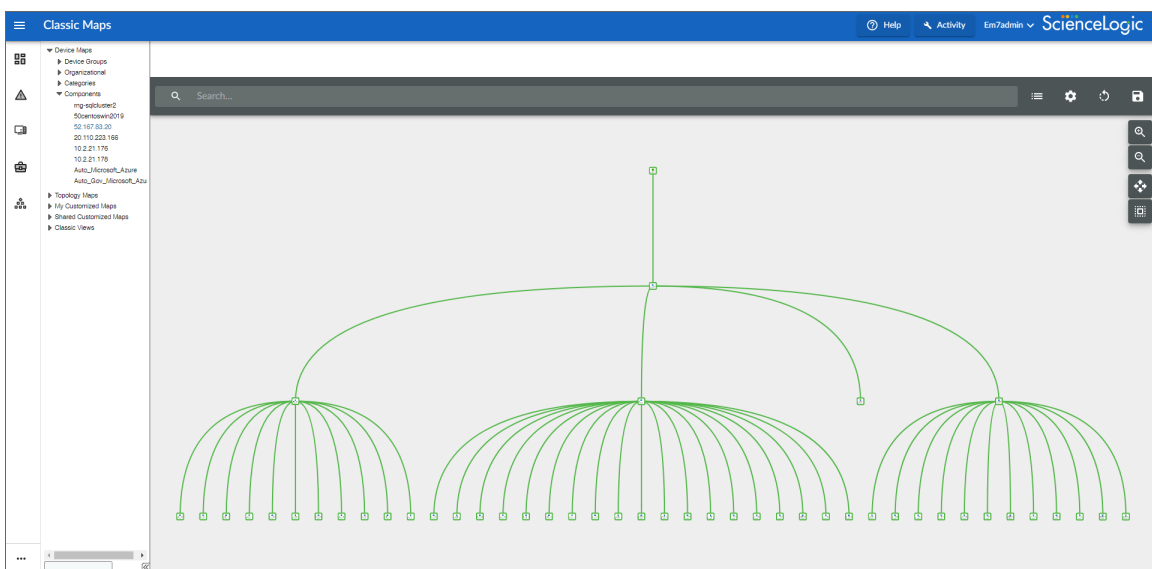
- The **Device Investigator** Map page (click **Map** in the **Device Investigator** page) displays a map of a particular device and all of the devices with which it has parent-child relationships. Double-clicking any of the listed devices reloads the page to make the selected device the primary device.



- The **Device Components** page (Devices > Device Components) displays a list of all root devices and component devices discovered by SL1 in an indented view, so you can easily view the hierarchy and relationships between child devices, parent devices, and root devices. To view the component devices associated with a SQL Server, find the server and click its plus icon (+):

Device Name	IP Address	Device Category	Device Class   Sub-class	DID	Organization	Current State	Collection Group	Collection State
Office 365 Root	--	Account	Microsoft   Office 365 Account	1	Office 365 Root	Minor	CUG1	Active
10.40.2.6	--	Servers	Microsoft   Windows Server 2008 R2	1165	PS	Notice	CUG1	Active
Microsoft SQL Server	--	Servers	Microsoft   SQL Server	1176	PS	Healthy	CUG1	Active
MSSQLSERVER	--	Controller	Microsoft   SQL Server Instance	1179	PS	Healthy	CUG1	Active
SHAREPOINT	--	Controller	Microsoft   SQL Server Instance	1180	PS	Healthy	CUG1	Active
master	--	Volume	Microsoft   SQL Server Database	1187	PS	Healthy	CUG1	Active
model	--	Volume	Microsoft   SQL Server Database	1184	PS	Healthy	CUG1	Active
msdb	--	Volume	Microsoft   SQL Server Database	1189	PS	Healthy	CUG1	Active
tempdb	--	Volume	Microsoft   SQL Server Database	1186	PS	Healthy	CUG1	Active
10.40.2-01	10.40.2.4	Servers	Microsoft   Windows Server 2008 R2	1163	PS	Healthy	CUG1	Active
10.40.3-01	10.40.3.7	Servers	Microsoft   Windows Server 2012 R2	1173	PS	Healthy	CUG1	Active
10.40.4-01	10.40.3.4	Servers	Microsoft   Windows Server 2012 R2	1171	PS	Major	CUG1	Active

- The **Component Map** page (Classic Maps > Device Maps > Components) allows you to view devices by root node and view the relationships between root nodes, parent components, and child components in a map. This page makes it easy to visualize and manage root nodes and their components. SL1 automatically updates the **Component Map** as new component devices are discovered. The platform also updates each map with the latest status and event information. To view the map for a SQL Server, go to the **Component Map** page and select the map from the list in the left NavBar. To learn more about the **Component Map** page, see the **Maps** manual.



© 2003 - 2023, ScienceLogic, Inc.

All rights reserved.

#### LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and / or changes in the products or services described in this Site at any time without notice.

#### Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic's trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

#### Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic's legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: [legal@sciencelogic.com](mailto:legal@sciencelogic.com). For more information, see <https://sciencelogic.com/company/legal>.

ScienceLogic

800-SCI-LOGIC (1-800-724-5644)

International: +1-703-354-1010