Guide to Microsoft System Center Management Pack for Azure SQL Database

Microsoft®
System Center
Operations Manager

Published in October 2020 by Microsoft Corporation.

This guide is based on version 7.0.26.0 of the management pack.

The Operations Manager team encourages you to provide feedback on the management pack by sending it to sqlmpsfeedback@microsoft.com.

Copyright

This document is provided "as is". Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

Table of Contents

Changes History

October 2020 - 7.0.26.0 RTM

- **What's New**
  - Added filtering list for SQL Servers and Databases to "Add Monitoring Wizard" template
  - Removed deprecated workflows
- **Bug Fixes**
  - Fixed issue with Elastic Pool performance data on vCore-based pricing tiers

September 2020 - 7.0.25.0 CTP

- **What's New**
  - Added support of vCore-based pricing tier
  - Updated the token renewal algorithm to get rid of 401 responses
  - Updated Core Library MP and the “Summary” Dashboard
  - Updated display strings
- **Bug Fixes**
  - Fixed an issue with an unnecessary slash symbol in some requests to Azure REST API
  - Fixed monitoring issues for databases that are replicated by failover groups and elastic pools

September 2019 - 7.0.5.0 private drop

- **Bug Fixes**
  - Fixed issue: datediff used for Long Running Transactions monitoring results in overflow in some environments

April 2018 - 7.0.4.0 RTM

- **What's New**
  - Provided a few minor UI improvements to the Add Monitoring Wizard
- **Bug Fixes**
  - Fixed issue: The management pack may stop working due to a conflict of the Azure REST API libraries with the ones coming from the Microsoft Azure Management Pack

May 2017 - 6.7.28.0 RTM

- **What's New**
  - Due to performance problems, several monitors and performance rules were enabled for getting information via T-SQL queries only (the affected metrics are as follows: Failed Connections, Blocked Connections, Successful Connections, Deadlocks Count)
- **Bug Fixes**
  - Fixed Azure SQL DB: DB Transactions Locks Count rule and Transaction Locks Count monitor
  - Fixed Azure SQL Database Event Log Collection Target Management Service Discovery
  - Fixed Server Exclude list filter: servername could not contain whitespaces
  - Fixed the display strings, implemented appropriate Azure Portal naming style

March 2017 - 6.7.25.0 CTP2

- **What's New**
  - Implemented performance improvements
  - Improved error handling in Add Monitoring Wizard

- **Bug Fixes**
  - Fixed issue: "Collect Elastic Database Pool Number of Databases" rule does not collect performance data if REST monitoring is used
  - Fixed issue: "Operations Manager Expression Filter Module" error messages appear in the Operations Manager event log

December 2016 - 6.7.11.0

- **What's New**
  - Azure Resource Manager is now supported: the previous versions of the Management Pack used T-SQL queries to SQL Server system views to get information on the health and performance of the databases; now, the Management Pack can also get this information from Azure REST API (this is a preferred option)
  - Multiple subscriptions and multiple servers are now supported
  - Added support for Azure AD authentication
  - Added regular expression filtering capability for Azure SQL Database instances and Elastic Pools
  - Improved monitoring efficiency: monitoring target is now defined by monitoring pool; WatcherNode class is considered to be deprecated
  - Improved SCOM Add Monitoring Wizard to reflect the new features of the Management Pack
  - Added health monitoring for Database Geo-Replication
  - Added health monitoring for Elastic Pools
  - Added monitoring for “Average DTU utilization percentage” metric
  - Introduced performance improvements to the Management Pack
  - Optimized performance rules notation: all Object Names are standardized; Instance Names are not used anymore
  - Updated the guide to reflect all the changes
  - Updated the visualization library
- **Bug Fixes**
  - Fixed issue: some rules work only if more than 1% of Azure SQL Database space is used

June 2016 - 1.6.1.0

- **What's New**
  - Added Dashboards
  - Added a number of new monitors and rules, including the following:
    - CPU Usage (%)
    - Workers Usage (%)
    - Log write (%)
    - Data I/O (%)
    - Sessions (%)
    - Count Failed Connection
    - Count Successful Connection
    - Count Connection Blocked by Firewall
    - Count of Deadlock
    - Count Throttling long transaction

        - Count Connection Failed
        - XTP Storage (In-memory OLTP Storage, %)
    - Deprecated Collect Azure SQL Database Internal/External Network Egress/Ingress performance rules
    - Deprecated SQL Azure Federation and Federation member workflows
    - Implemented rebranding: the management pack and some workflow names have been changed

May 2013

- The original release of this management pack

Management Pack Scope and Supported Configurations

This management pack is designed to monitor Azure SQL Database by means of Azure REST API and T-SQL queries.

Azure SQL Database is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement.

This section explains what Azure SQL Database features are covered by this management pack, what configurations are supported, what monitoring features the management pack offers and what prerequisites should be met to begin with this management pack.

Azure SQL Database Features and Supported Purchase Models

Features

The Azure SQL Database Management Pack supports the following features and configurations of Azure SQL Database:

- SQL Server
    - DTU metrics
- SQL Database
    - CPU metrics
    - DTU metrics
    - Connections metrics
    - Transactions metrics
    - Space metrics
    - Sessions metrics
- SQL Elastic Pools
    - Storage metrics
    - DTU metrics
    - Sessions metrics
    - CPU metrics
    - I/O metrics
- Database Geo-Replication
    - Geo-Replication Link State

Purchase Models

The Azure SQL Database Management Pack supports monitoring of databases in any of the following purchase models:

- DTU-based SQL purchase models:
    - Basic
    - Standard
    - Premium
- vCore-based purchase models:
    - General Purpose
    - Hyperscale
    - Business Critical

vCore can also be *Provisioned* or *Serverless* within each purchase model. For more information, see Azure SQL Database pricing.

When using vCore-based purchase model, the following rules do not collect any data because no *DTULimit* metrics are available in this model:

- Azure SQL DB: DB DTU Used Count
- Azure SQL DB: DB DTU Limit Count
- Azure SQL DB: DB DTU Percentage

When using Hyperscale service tier, some of the space monitoring workflows may not collect data correctly. For more information see the related Known Issue.

SCOM Configurations

The Azure SQL Database Management Pack supports the following versions of System Center Operations Manager and operating systems:

- **System Center Operations Manager**
    - System Center Operations Manager 2012 R2
    - System Center Operations Manager 2016
    - System Center Operations Manager 1801
    - System Center Operations Manager 1807
    - System Center Operations Manager 2019
- **Operating Systems**
    - Windows Server 2012
    - Windows Server 2012 R2
    - Windows Server 2016
    - Windows Server 2019

Prerequisites

Installation of **.NET Framework 4.5.2** (at least) is required.

Management Pack Delivery

You can download the Azure SQL Database Management Pack from the Microsoft portal or find it in the System Center Operations Manager Online Catalog.

The package includes the following files:

- Microsoft.SqlServer.Azure.ManagementPack.msi
- AzureSQLDatabaseMPGuide.pdf

The Azure SQL Database Management Pack consists of the following files:

- Microsoft.SqlServer.Azure.mpb
- Microsoft.SqlServer.Azure.Presentation.mp
- Microsoft.SqlServer.UserMonitoring.mpb
- Microsoft.SQLServer.Core.Library.mpb
- Microsoft.SQLServer.Visualization.Library.mpb

The management pack supports monitoring of 2000 databases in a single management group.

SQL Azure Federation and Federation member workflows are considered to be deprecated in this management pack.

Importing Management Pack

The Azure SQL Database Management Pack can be imported, as described in How to Import an Operations Manager Management Pack.

If you have been using an agnostic version of the SQL Server Management Pack prior to the upgrade, you can remove both the *Microsoft.SQLServer.Generic.Dashboards.mp* and the *Microsoft.SQLServer.Generic.Presentation.mp* management packs after the upgrade.

For a non-agnostic version, removal of these management packs is not possible.

Monitoring Configuration

Azure SQL Database Add Monitoring Wizard

You can configure monitoring of Azure SQL Database by means of **Add Monitoring Wizard** using the Azure REST API and T-SQL queries.

Differences Between Azure REST API and T-SQL Monitoring

When using T-SQL monitoring, each of the existing monitoring workflows that come with this management pack is available. When only the Azure REST API is used, the following monitoring workflows do not work due to API limitations:

- Rules:
    - Azure SQL DB: DB Transactions Locks Count
    - Azure SQL DB: DB Sessions Count
    - Azure SQL DB: DB Sessions Average Memory Consumption (MB)
    - Azure SQL DB: DB Sessions Rows Returned

    - Azure SQL DB: DB Sessions Total CPU Time (ms)
    - Azure SQL DB: DB Sessions Total Read/Write Operations
    - Azure SQL DB: DB Sessions Total Memory Consumption (MB)
    - Azure SQL DB: DB Transactions Max Log Usage (MB)
    - Azure SQL DB: DB Transactions Max Running Time (minutes)
    - Azure SQL DB: DB Blocked by Firewall Count
    - Azure SQL DB: DB Failed Connections Count
    - Azure SQL DB: DB Successful Connections Count
    - Azure SQL DB: DB Deadlocks Count

- Monitors:

    - Transaction Locks Count
    - Sessions Count
    - Sessions Average Memory
    - Sessions Rows Returned
    - Sessions Total CPU Time
    - Sessions Total I/O
    - Sessions Total Memory
    - Transaction Log Space Used
    - Transaction Execution Time
    - Count of Failed Connection
    - Count of connections blocked by the Firewall

If you want to enable these monitoring workflows when using the Azure REST API, select the **Use T-SQL monitoring** checkbox and run required T-SQL scripts provided in Configuring Azure REST API Monitoring.

Configuring Azure REST API Monitoring

Azure REST API monitoring is intended for a wider range of monitoring targets.

When using this monitoring mode, the Azure SQL Database Management Pack utilizes an Azure AD application (or Service Principal Name) for authentication in Azure AD, which gives access to Azure Resource Management API. The account that you use must have either the *Owner* role (or higher), or any of the following roles:

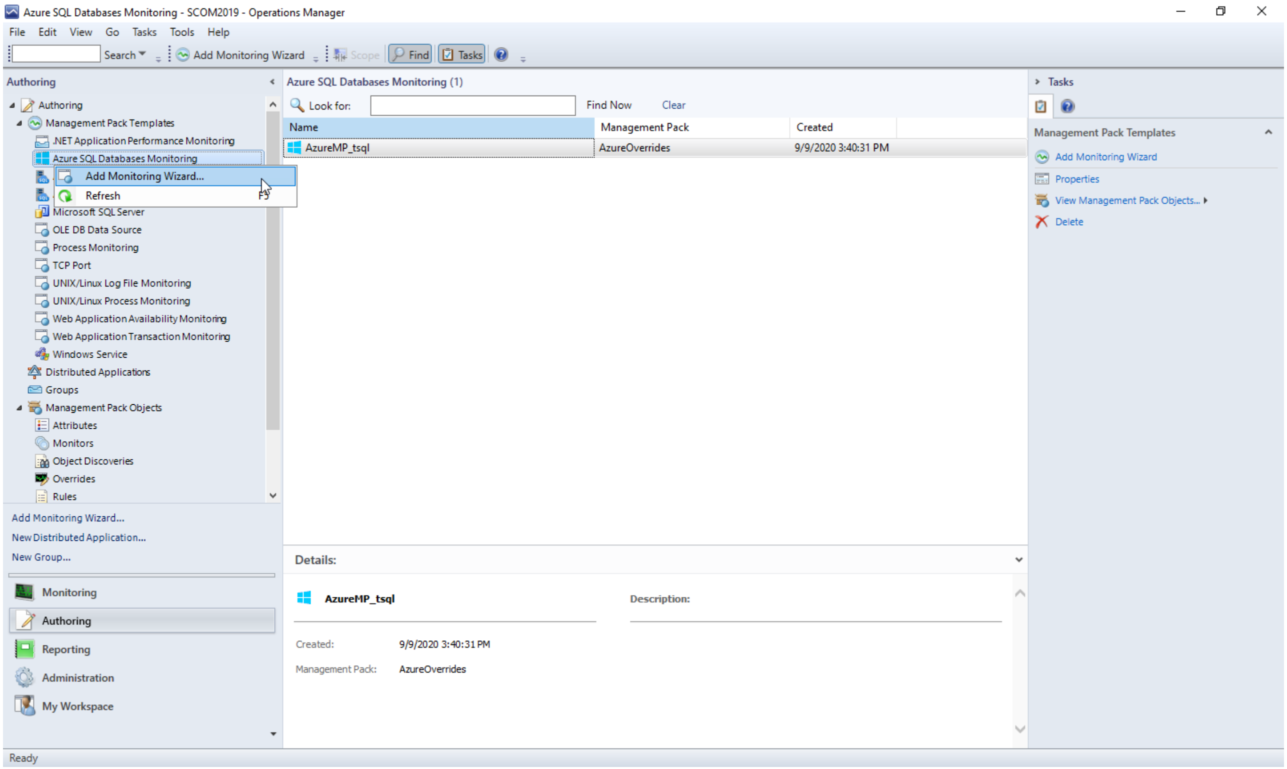
- Active Directory Administrator
- Service Administrator or Co-Administrator

For more information, see How to - Use the portal to create an Azure AD application and service principal that can access resources.

When configuring an Azure SQL Database Management Pack template, a new Run As account with Azure Service Principal Name credentials is created via Azure REST API. For more information, see Azure REST API Reference.

To begin monitoring of Azure SQL Database using the Azure REST API, perform the following steps:

1. In the System Center Operations Manager console, navigate to **Authoring | Management Pack Templates**, right-click **Azure SQL Databases Monitoring** and select **Add Monitoring Wizard**.


Azure SQL Databases Monitoring - SCOM2019 - Operations Manager
File Edit View Go Tasks Tools Help
Search
Add Monitoring Wizard
Scope
Find
Tasks
Authoring
Authoring
Management Pack Templates
.NET Application Performance Monitoring
Azure SQL Databases Monitoring
Add Monitoring Wizard...
Refresh
F5
Microsoft SQL Server
OLE DB Data Source
Process Monitoring
TCP Port
UNIX/Linux Log File Monitoring
UNIX/Linux Process Monitoring
Web Application Availability Monitoring
Web Application Transaction Monitoring
Windows Service
Distributed Applications
Groups
Management Pack Objects
Attributes
Monitors
Object Discoveries
Overrides
Rules
Add Monitoring Wizard...
New Distributed Application...
New Group...
Monitoring
Authoring
Reporting
Administration
My Workspace
Ready
Azure SQL Databases Monitoring (1)
Look for:
Find Now
Clear
Tasks
Management Pack Templates
Add Monitoring Wizard
Properties
View Management Pack Objects...
Delete

Name	Management Pack	Created
AzureMP_tsql	AzureOverrides	9/9/2020 3:40:31 PM

Details:

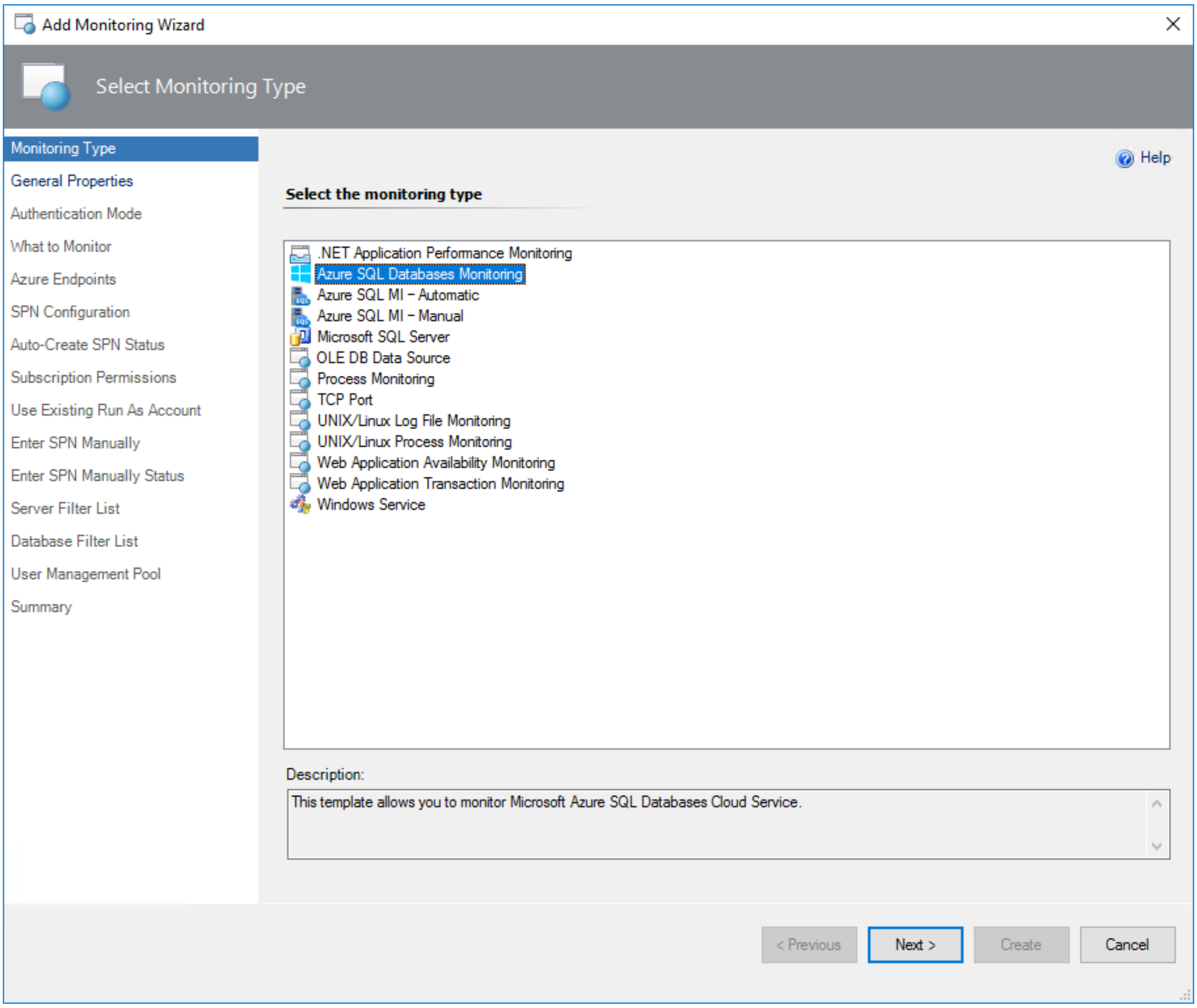
AzureMP_tsql

Description:

Created: 9/9/2020 3:40:31 PM

Management Pack: AzureOverrides

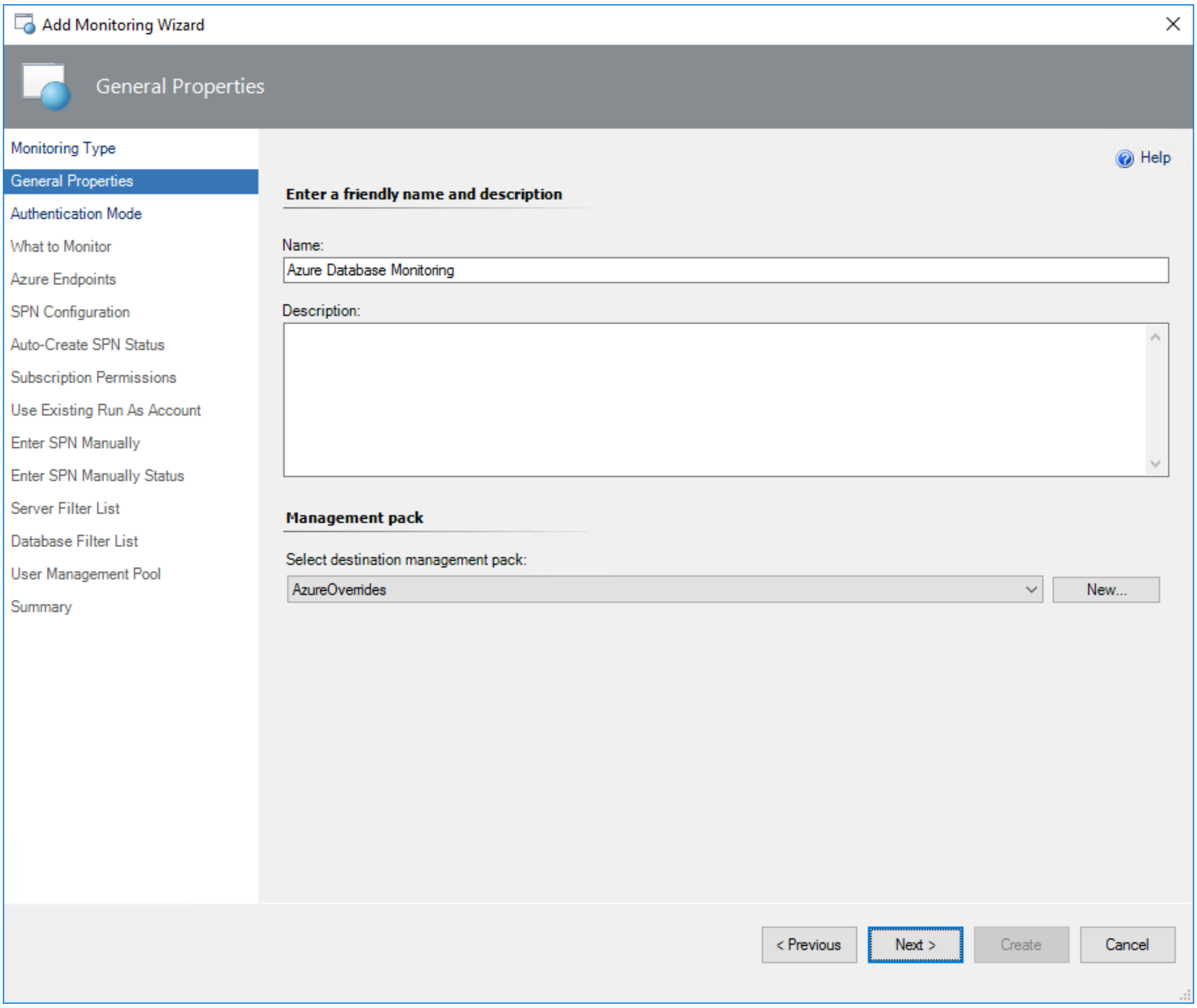
2. At the **Monitoring Type** step, select **Azure SQL Databases Monitoring** and click **Next**.



Description:

This template allows you to monitor Microsoft Azure SQL Databases Cloud Service.

3. At the **General Properties** step, enter a new name and description and from the **Select destination management pack** drop-down list, select a management pack that you want to use to store the template.



Enter a friendly name and description

Name: Azure Database Monitoring

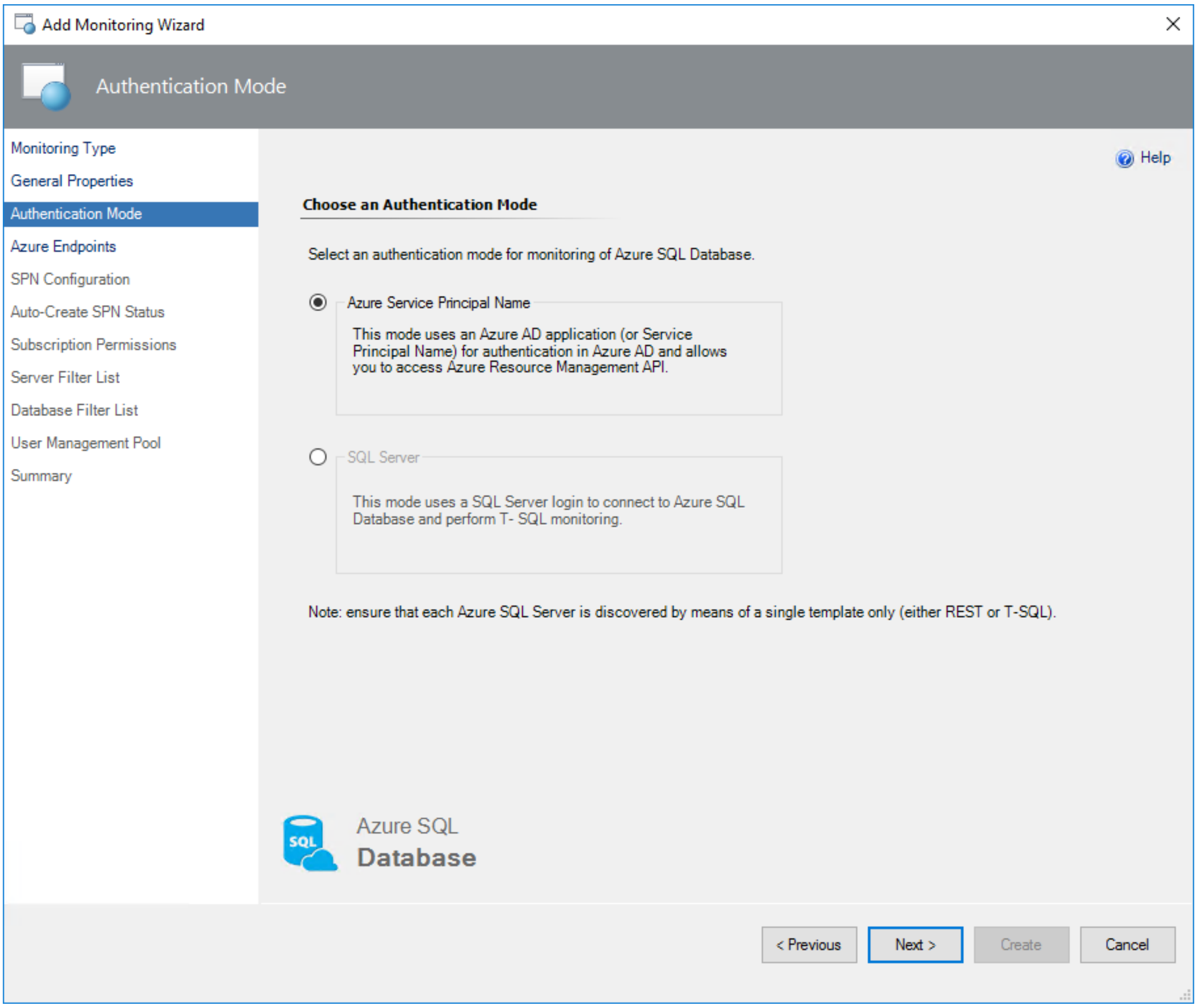
Description:

Management pack

Select destination management pack: AzureOverrides

If you do not have a management pack for this purpose, you can create a new one by clicking **New**.

4. At the **Authentication Mode** step, select **Azure Service Principal Name**.



Choose an Authentication Mode

Select an authentication mode for monitoring of Azure SQL Database.

● Azure Service Principal Name

This mode uses an Azure AD application (or Service Principal Name) for authentication in Azure AD and allows you to access Azure Resource Management API.

○ SQL Server

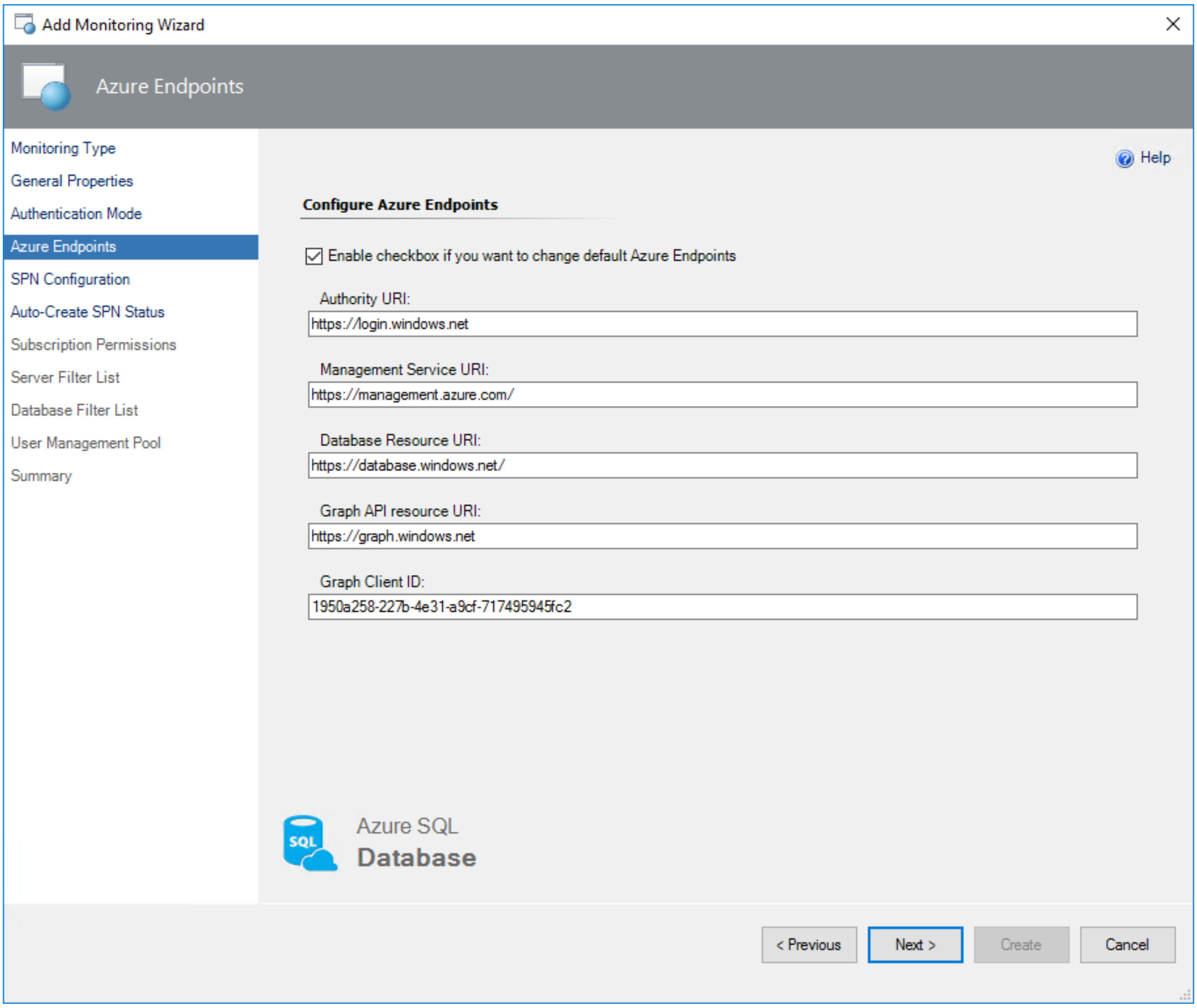
This mode uses a SQL Server login to connect to Azure SQL Database and perform T- SQL monitoring.

Note: ensure that each Azure SQL Server is discovered by means of a single template only (either REST or T-SQL).

5. At the **Azure Endpoints** step, select the **Enable checkbox if you want to change default Azure Endopints** checkbox and modify the default Azure endpoints if required. The default endpoints for creating Azure Service Principal Name are as follows:

   - Authority URI: https://login.windows.net
   - Management Service URI: https://management.azure.com

     This endpoint is also used for **Azure REST API**. In this case, the Firewall port 443 should be used. Nevertheless, according to the Ports beyond 1433 for ADO.NET 4.5 article, the Firewall port 1433 should be used.
   - Database Resource URI: https://database.windows.net
   - Graph API Resource URI: https://graph.windows.net



Configure Azure Endpoints

☑ Enable checkbox if you want to change default Azure Endpoints

Authority URI: https://login.windows.net

Management Service URI: https://management.azure.com/

Database Resource URI: https://database.windows.net/

Graph API resource URI: https://graph.windows.net

Graph Client ID: 1950a258-227b-4e31-a9cf-717495945fc2

6. At the **SPN Configuration** step, select any of the following options:

   - **Auto-Create SPN**

     Select this option If you want Azure Service Principal Name to be created automatically by the Azure SQL MP library using the Azure REST API. With this option selected, a new Run As Account is created with the specified Azure Service Principal Name.

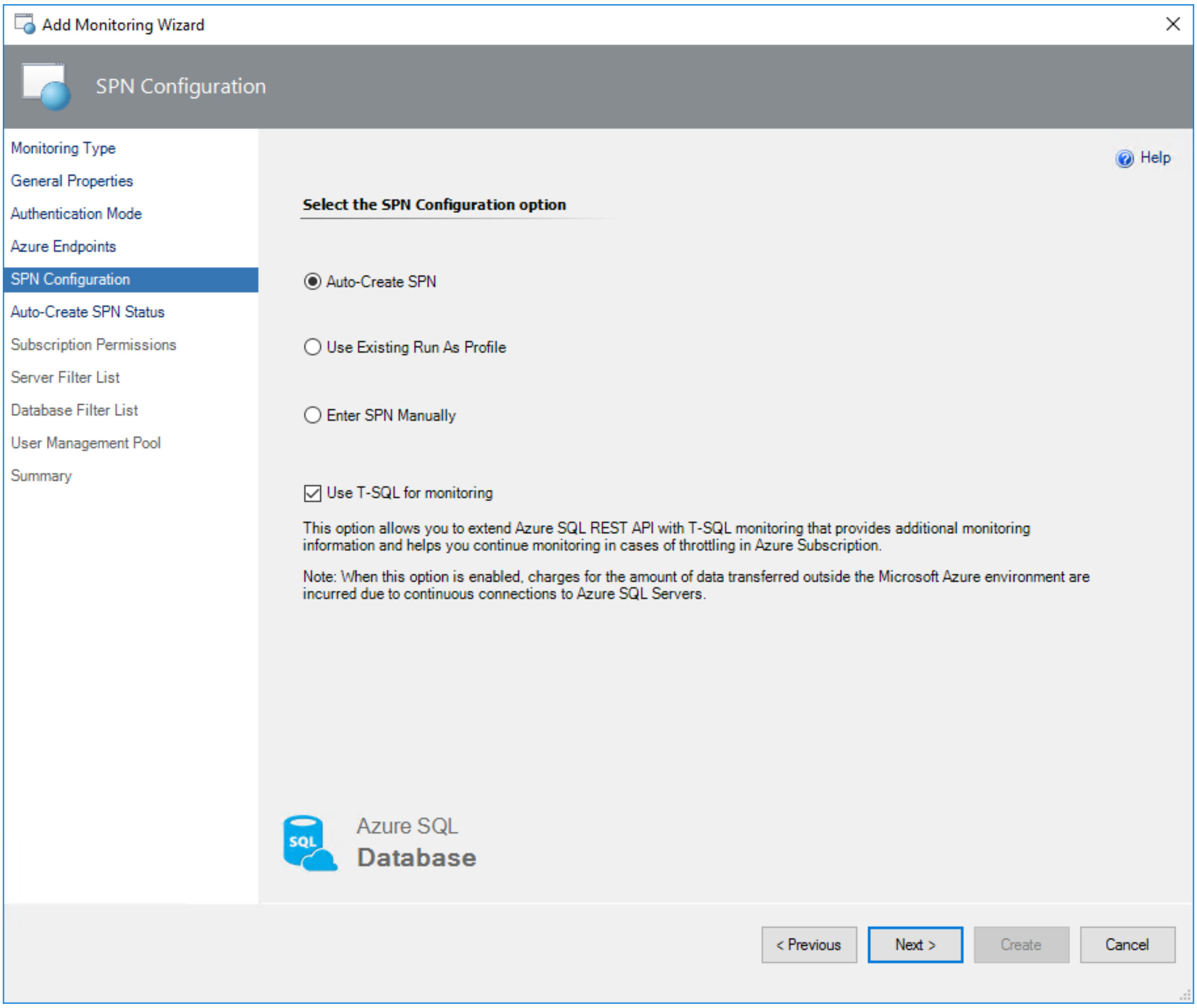
   - **Use Existing Run As Profile**

     Select this option if you want to use your own Azure Service Principal Name.

   - **Enter SPN Manually**

     Select this option if you have already configured a Run As Profile (e.g. by PowerShell) with appropriate Azure Service Principal Name credentials.

   For any of these options, you can select the **Use T-SQL for monitoring** checkbox if you want to receive additional monitoring information and neutralize Azure Subscription throttling effects. For more information, see Differences Between Azure REST API and T-SQL Monitoring.



Select the SPN Configuration option

● Auto-Create SPN

○ Use Existing Run As Profile

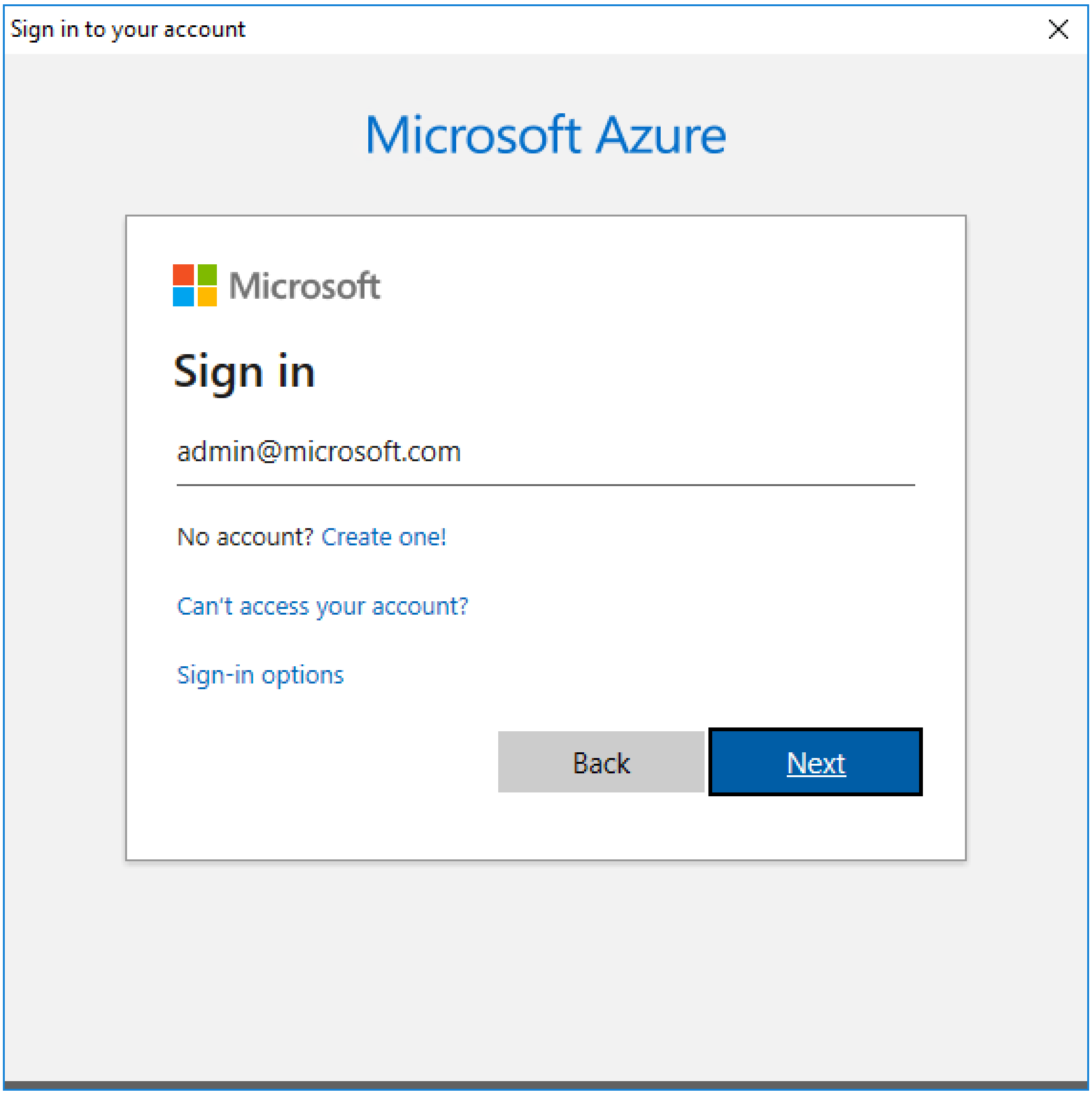
○ Enter SPN Manually

☑ Use T-SQL for monitoring

This option allows you to extend Azure SQL REST API with T-SQL monitoring that provides additional monitoring information and helps you continue monitoring in cases of throttling in Azure Subscription.

Note: When this option is enabled, charges for the amount of data transferred outside the Microsoft Azure environment are incurred due to continuous connections to Azure SQL Servers.

If you select the **Auto-Create SPN** option, the **Microsoft Azure sign-in** window appears. In this window, enter your work, school or personal Microsoft account credentials, click **Next** and complete the form.



Sign in to your account

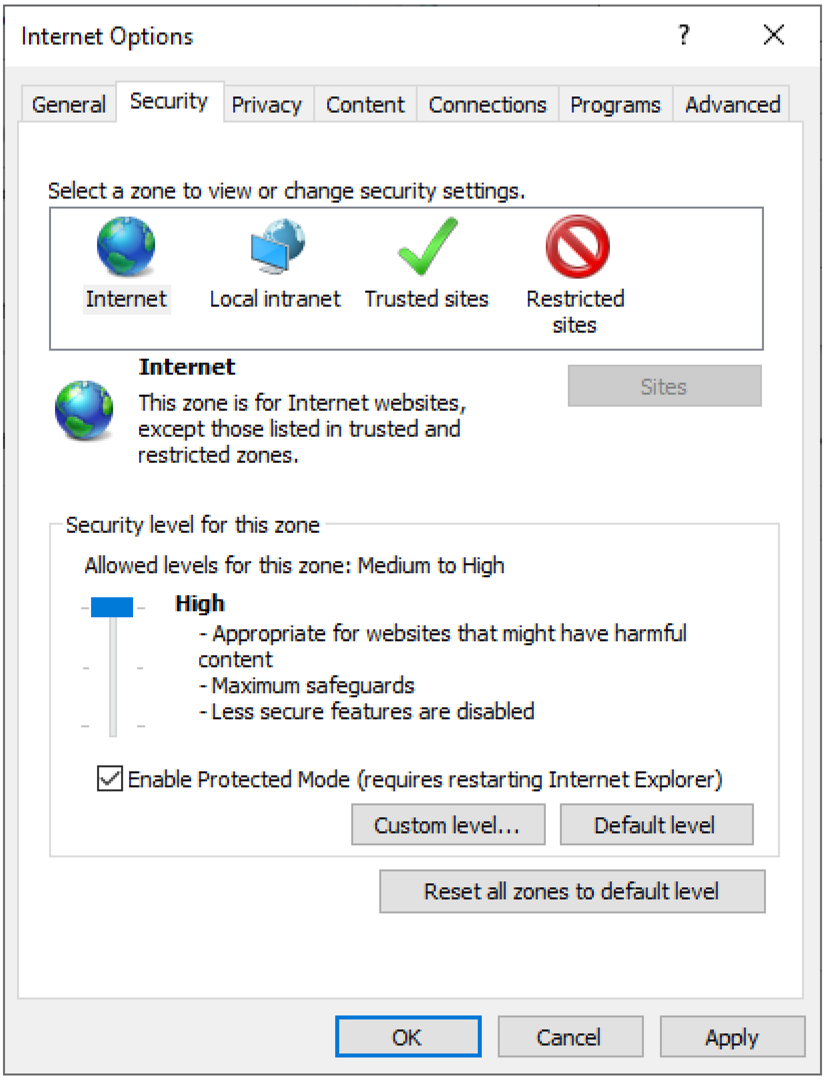
Microsoft Azure

Microsoft

Sign in

admin@microsoft.com

You may receive internet security alerts at this step. To solve this issue, go to the **Security** section of the **Internet Properties** and lower the security level for the internet zone.



Select a zone to view or change security settings.

Internet

This zone is for Internet websites, except those listed in trusted and restricted zones.

Security level for this zone

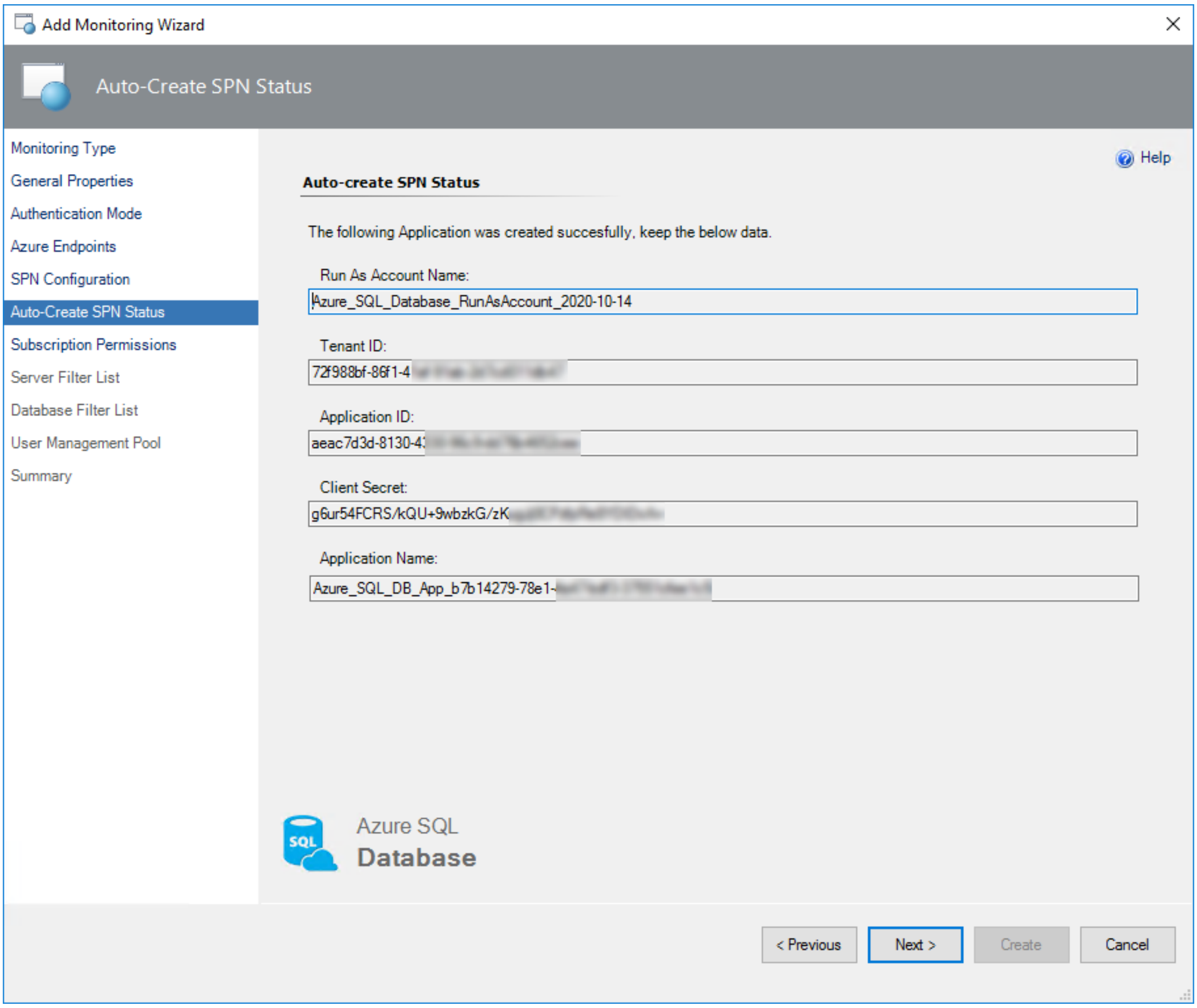
Allowed levels for this zone: Medium to High

High

- Appropriate for websites that might have harmful content
- Maximum safeguards
- Less secure features are disabled

☑ Enable Protected Mode (requires restarting Internet Explorer)

Upon successful creation of the Azure AD application, at the **Auto-Create SPN Status** step, authentication data will be displayed. Click **Next**.



Auto-create SPN Status

The following Application was created succesfully, keep the below data.

Run As Account Name: Azure_SQL_Database_RunAsAccount_2020-10-14

Tenant ID: 72f988bf-86f1-4[redacted]

Application ID: aeac7d3d-8130-4[redacted]

Client Secret: g6ur54FCRS/kQU+9wbzkG/zK[redacted]

Application Name: Azure_SQL_DB_App_b7b14279-78e1-[redacted]

⚠ We recommend to save this data for further usage.

To perform T-SQL monitoring when using an Azure service principal name, create a separate user for every monitored database and grant this user the '*dbmanager*' role according to the following queries.

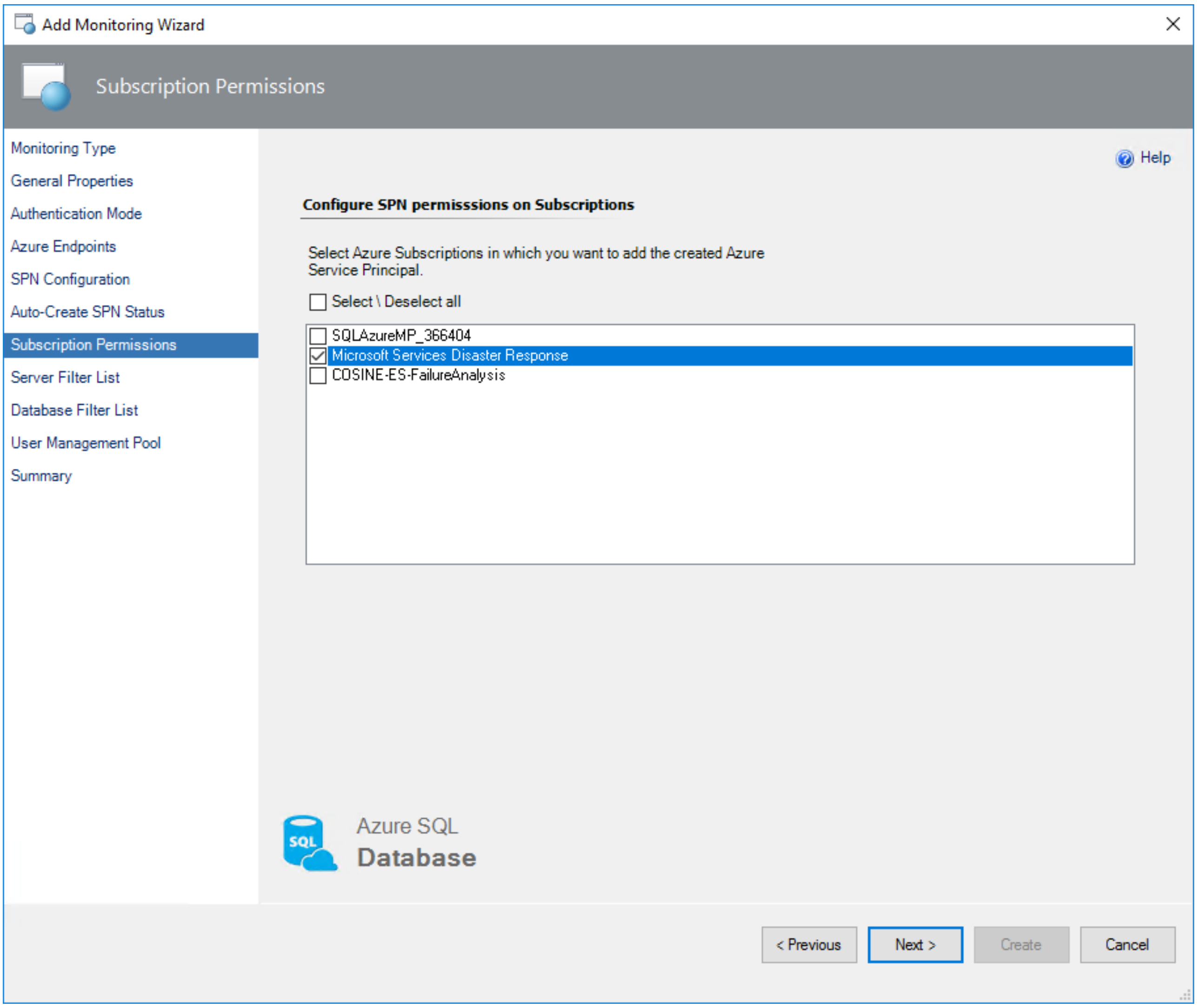
```
/*Run this on [master] database.
Replace the 'ApplicationName' parameter with that specified in the
Application Name field. See figure above.*/
CREATE USER [ApplicationName] FROM EXTERNAL PROVIDER;
exec sp_addrolemember 'dbmanager', 'ApplicationName';

/*Run this on all [user] databases.
Replace the 'ApplicationName' parameter with that specified in the
Application Name field. See figure above.*/
CREATE USER [ApplicationName] FROM EXTERNAL PROVIDER;
GRANT VIEW DATABASE STATE TO [ApplicationName];
```

To perform these queries via SSMS, connect to the Azure SQL server as **Active Directory Administrator**.

Upon assigning permissions to Azure Service Principal Name for every database, T-SQL monitoring should work properly in REST+T-SQL mode.

For proper monitoring of georeplicas by means of T-SQL, grant the *SQL Administrator* rights on each replica server.

At the **Subscription Permissions** step, select Azure subscriptions to which you want to add the created Azure Service Principal Name.



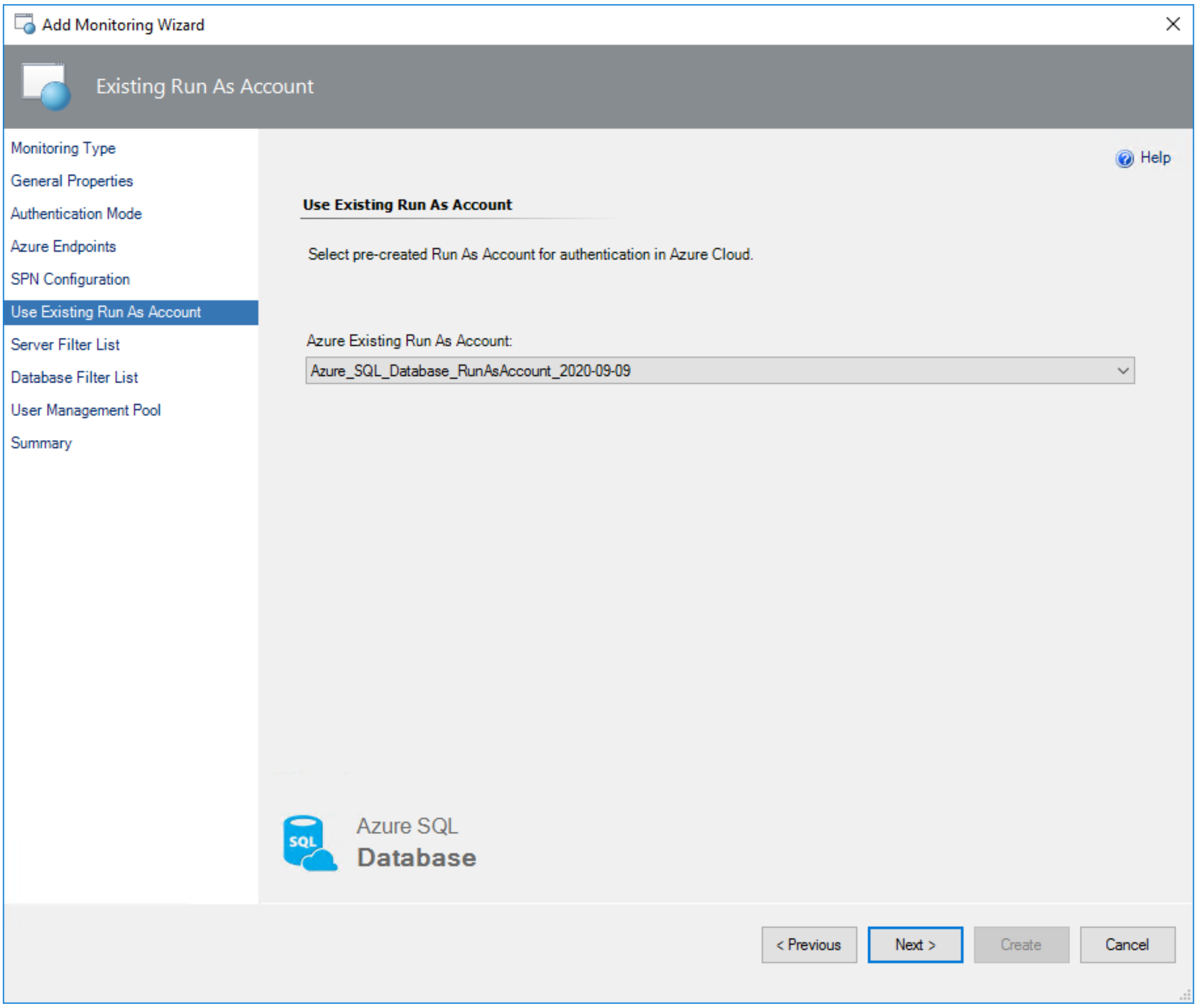
Configure SPN permisssions on Subscriptions

Select Azure Subscriptions in which you want to add the created Azure Service Principal.

□ Select \ Deselect all

□ SQLAzureMP_366404
☑ Microsoft Services Disaster Response
□ COSINE-ES-FailureAnalysis

If you want to use an existing Run As Profile, at the **SPN Configuration** step, select the **Use Existing Run As Profile** option, click **Next** and select an existing Run As Account associated with the Azure Service Principal Name. This Run As Account will be used for authentication in Azure Cloud.

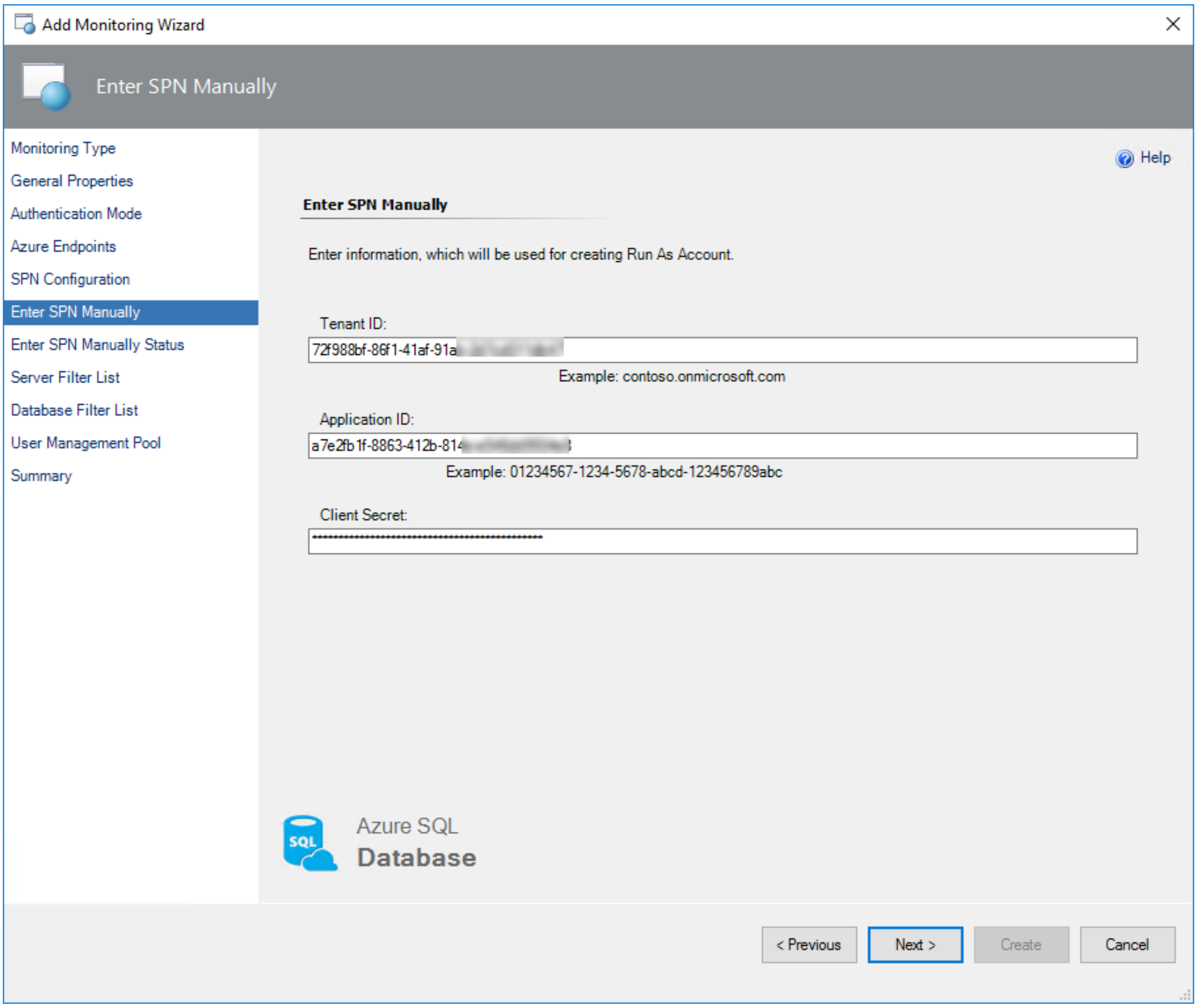


Use Existing Run As Account

Select pre-created Run As Account for authentication in Azure Cloud.

Azure Existing Run As Account: Azure_SQL_Database_RunAsAccount_2020-09-09

If you already have an Azure service principal name and want to use it to create a new Run As Account, at the **SPN Configuration** step, select the **Enter SPN Manually** option, click **Next** and provide required information about your Azure Service Principal Name. This information will be used to create a new Run As Account for authentication in Azure Cloud.



Enter SPN Manually

Enter information, which will be used for creating Run As Account.

Tenant ID: 72f988bf-86f1-41af-91a[redacted]

Example: contoso.onmicrosoft.com

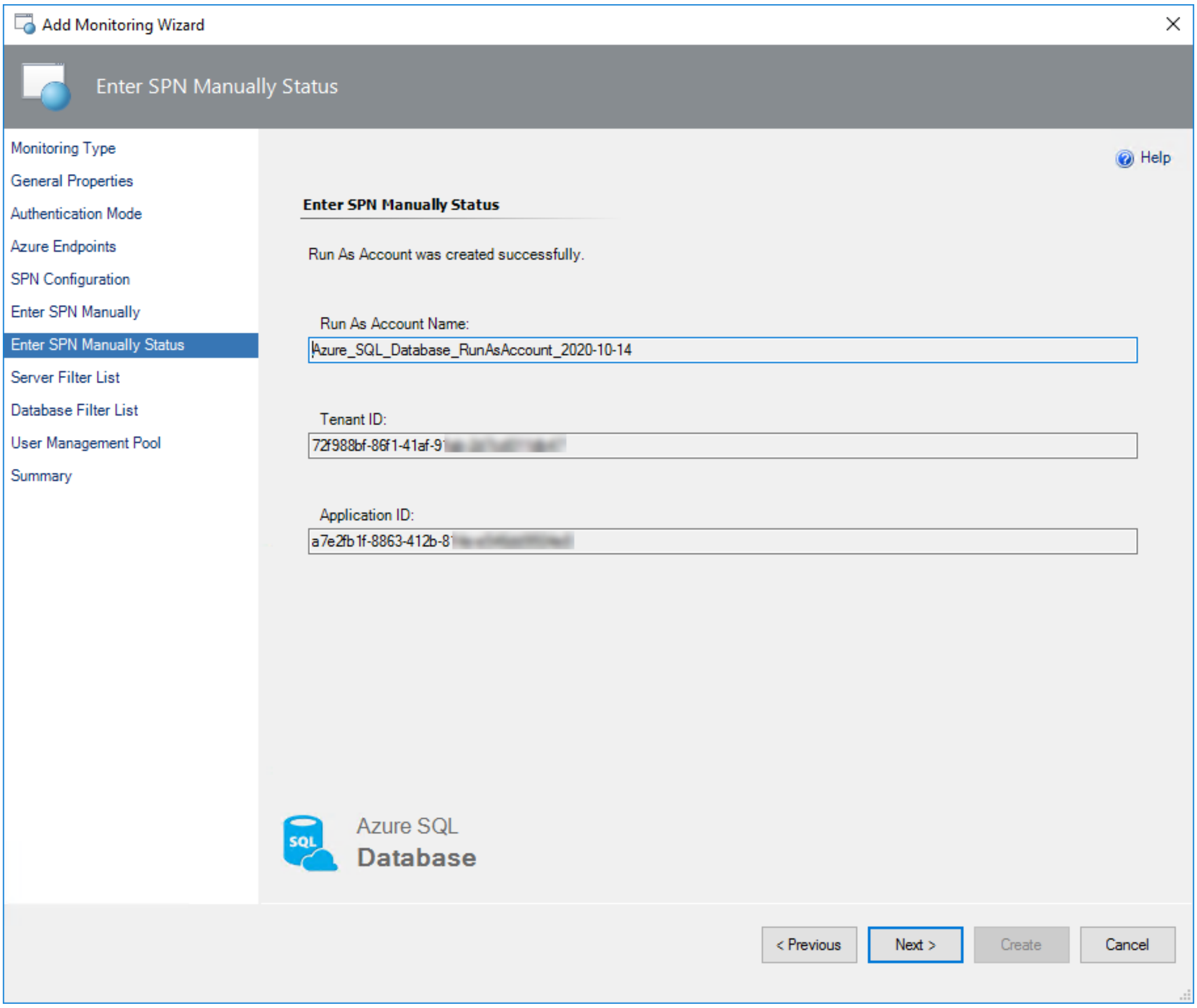
Application ID: a7e2fb1f-8863-412b-814[redacted]

Example: 01234567-1234-5678-abcd-123456789abc

Client Secret: ••

If necessary, you can create and configure a new Azure Active Directory application and Service Principal Name by using Azure PowerShell. For more information, see How to: Use Azure PowerShell to create a service principal with a certificate.

Once a new Run As Account is created, at the **Enter SPN Manually Status** step, review the status and click **Next**.



Enter SPN Manually Status

Run As Account was created successfully.

Run As Account Name: Azure_SQL_Database_RunAsAccount_2020-10-14

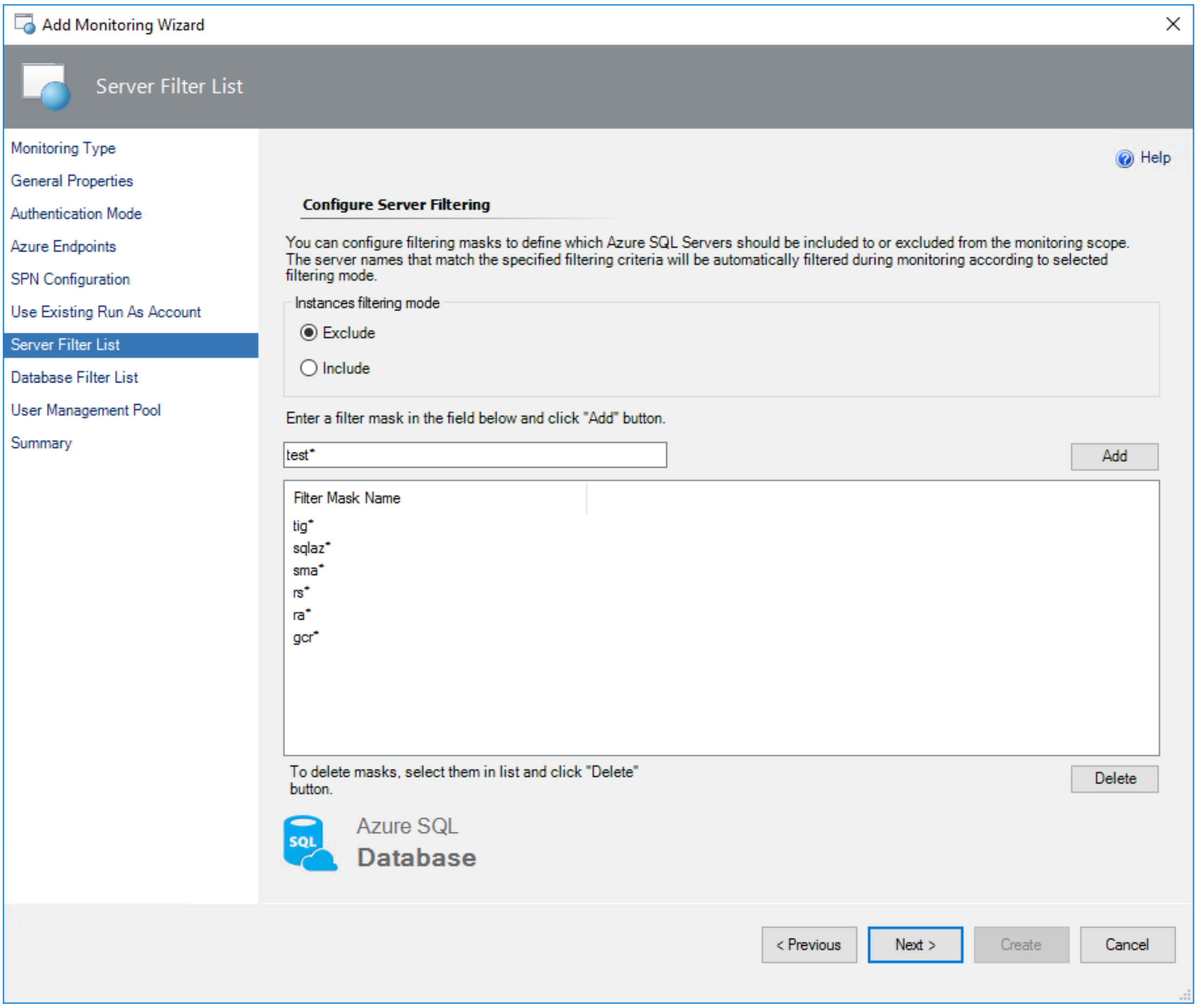
Tenant ID: 72f988bf-86f1-41af-91[redacted]

Application ID: a7e2fb1f-8863-412b-81[redacted]

7. [Optionally] At the **Server Filter List** step, select filtering mode, which can be either **Exclude** or **Include**, enter filtering masks that should match SQL Server names that you want to exclude from or include to the monitoring list, click **Add** and then click **Next**.

   A server name can contain only lowercase letters, numbers, and '-' character, but cannot start from or end with a -\ character or contain more than 63 characters. A server exclude list filter mask ignores whitespaces.

   If you want to remove an existing mask, select it and click **Delete**.



Configure Server Filtering

You can configure filtering masks to define which Azure SQL Servers should be included to or excluded from the monitoring scope. The server names that match the specified filtering criteria will be automatically filtered during monitoring according to selected filtering mode.

Instances filtering mode

● Exclude

○ Include

Enter a filter mask in the field below and click "Add" button.

test*

Filter Mask Name
tig*
sqlaz*
sma*
rs*
ra*
gcr*

To delete masks, select them in list and click "Delete" button.

8. [Optionally] At the **Database Filter List** step, select filtering mode, which can be either **Exclude** or **Include**, enter filtering masks that should match database names that you want to exclude from or include to the monitoring list, click **Add** and then click **Next**.

   A database name cannot end with '.' or ' ' characters, contain '<,>,*,%,&,:,,/,?' or control characters, and cannot have more than 128 characters.

   For example, if you select the **Exclude** option and set the following masks:

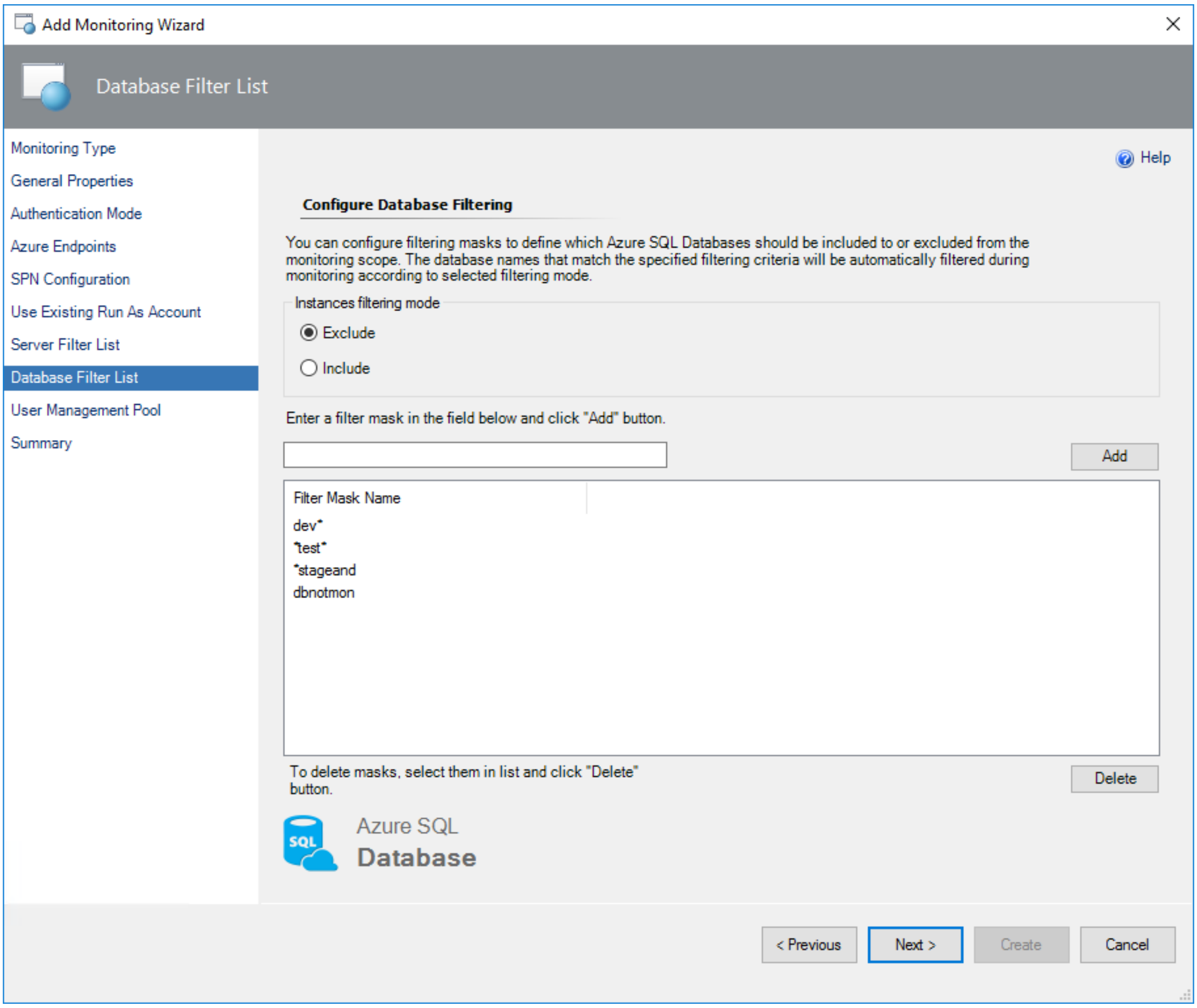
   - dev*
   - *test*
   - *stageand
   - dbnotmon

   the monitoring behavior would be as follows:

DB Name	Monitored/Not monitored
dev	Not monitored
dev_sales	Not monitored
sales_dev	Monitored

DB Name	Monitored/Not monitored
test	Not monitored
test_sales	Not monitored
sales_test	Not monitored
stage	Not monitored
stage_dev	Monitored
dev_stage	Not monitored
dbnotmon	Not monitored
dbnotmon_sales	Monitored
sales_dbnotmon	Monitored

If you want to remove an existing mask, select it and click **Delete**.



Configure Database Filtering

You can configure filtering masks to define which Azure SQL Databases should be included to or excluded from the monitoring scope. The database names that match the specified filtering criteria will be automatically filtered during monitoring according to selected filtering mode.

Instances filtering mode

● Exclude

○ Include

Enter a filter mask in the field below and click "Add" button.

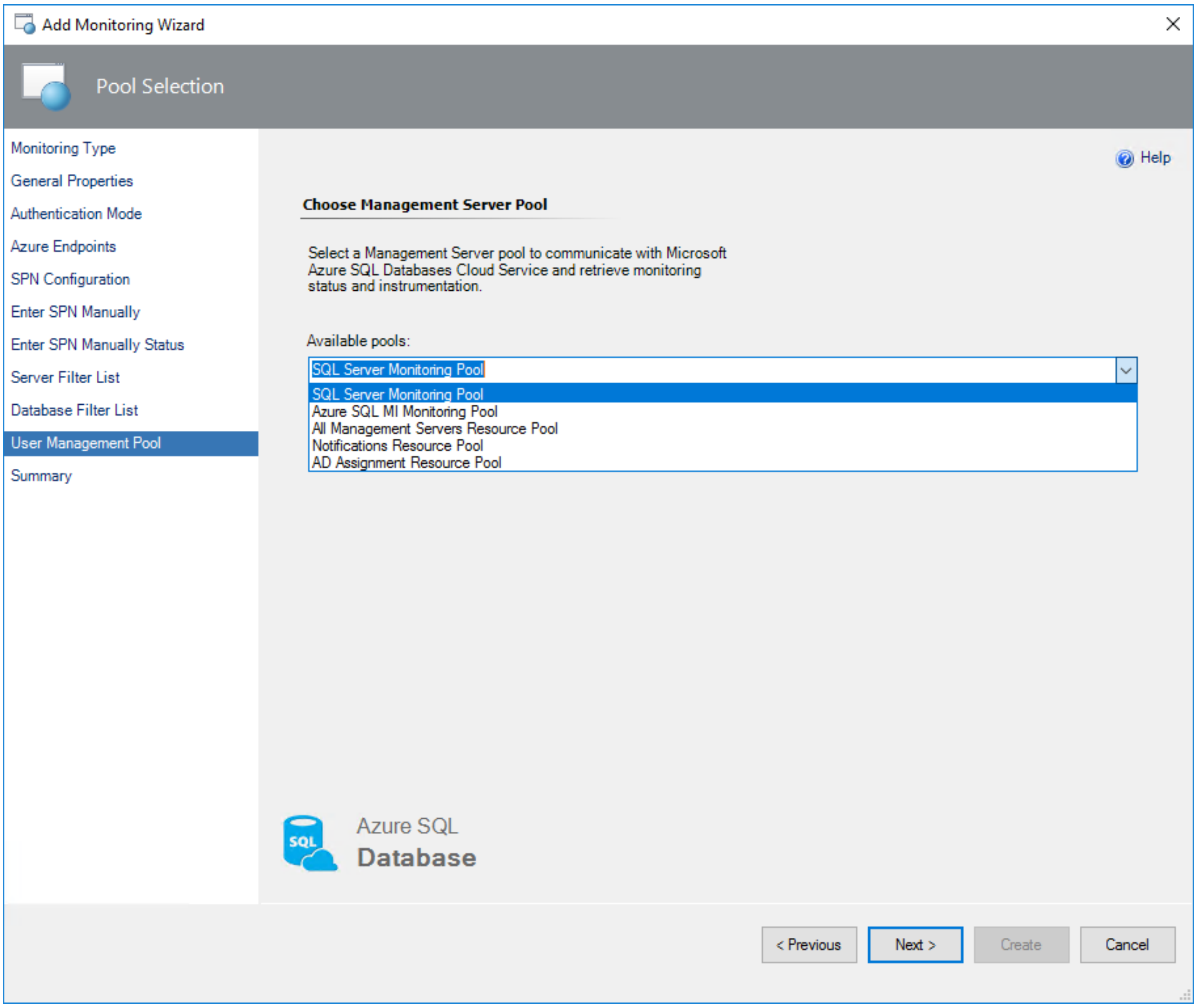
Add

Filter Mask Name
dev*
test
*stageand
dbnotmon

To delete masks, select them in list and click "Delete" button.

Delete

9. At the **User Management Pool** step, select a pool with management servers and click **Next**.



Choose Management Server Pool

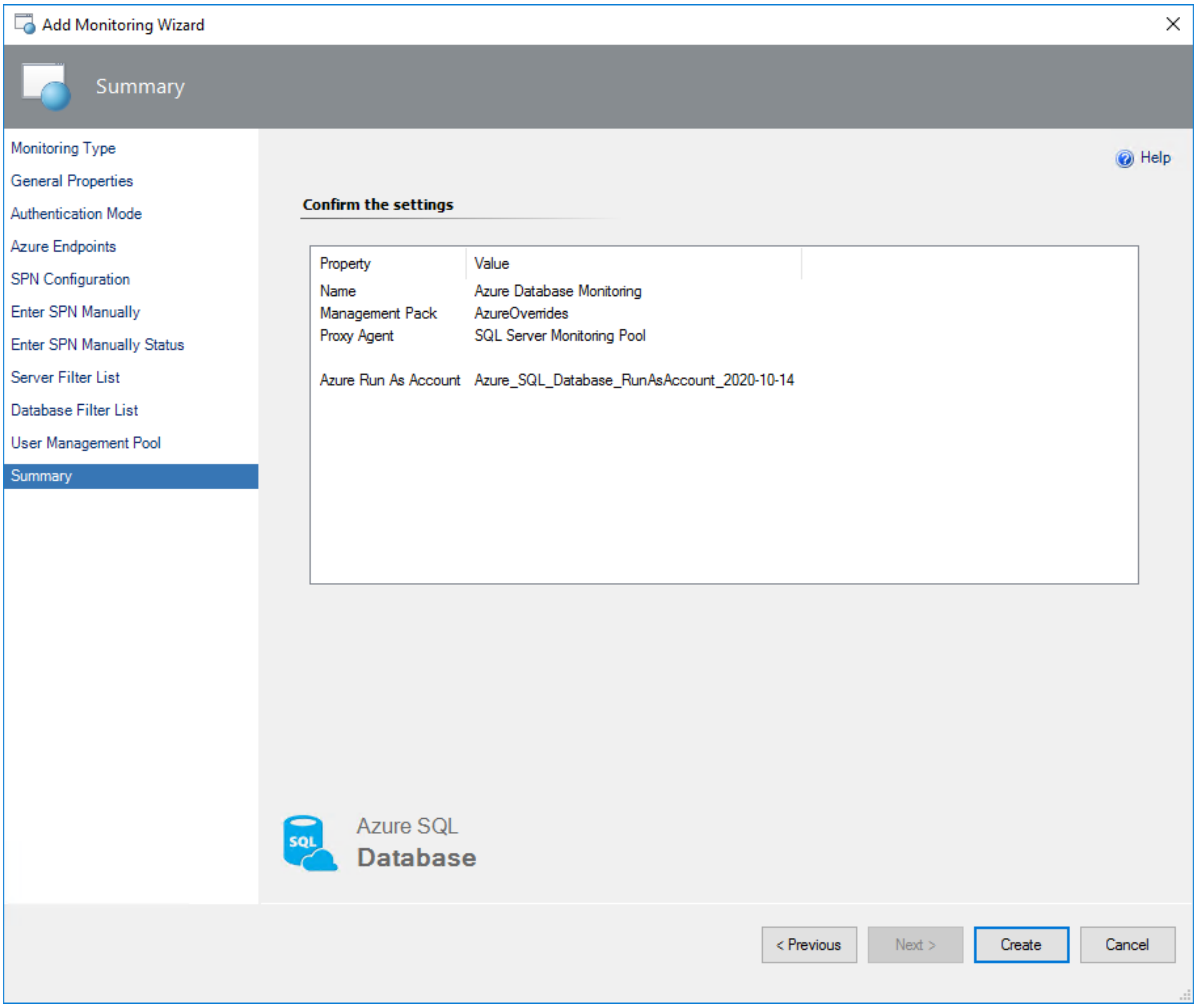
Select a Management Server pool to communicate with Microsoft Azure SQL Databases Cloud Service and retrieve monitoring status and instrumentation.

Available pools:

SQL Server Monitoring Pool

SQL Server Monitoring Pool
Azure SQL MI Monitoring Pool
All Management Servers Resource Pool
Notifications Resource Pool
AD Assignment Resource Pool

10. At the **Summary** step, review connection settings and click **Create**.



Confirm the settings

Property	Value
Name	Azure Database Monitoring
Management Pack	AzureOverrides
Proxy Agent	SQL Server Monitoring Pool
Azure Run As Account	Azure_SQL_Database_RunAsAccount_2020-10-14

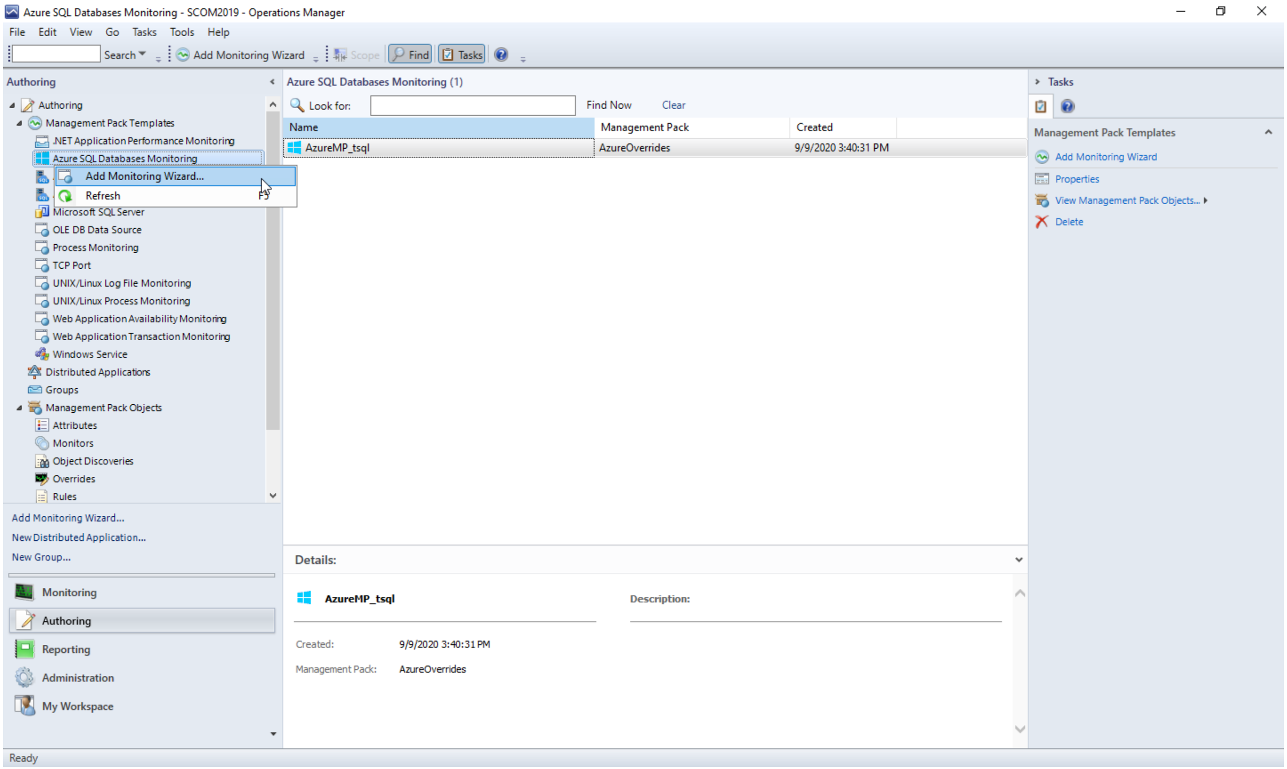
Configuring T-SQL Monitoring

T-SQL is intended for monitoring of specific Azure SQL Database Servers. When choosing this mode, the monitoring workflows such as discoveries, rules and monitors use T-SQL queries in datasources.

Each workflow datasource creates a new SQL connection for every pair of SQL Server credentials (login/password). SQL connections are counted for database transaction units and affect the bill. For more information, see Resource limits for Azure SQL Database and Azure Synapse Analytics servers.

To begin monitoring of Azure SQL Database using T-SQL queries, perform the following steps:

1. In the System Center Operations Manager console, navigate to **Authoring | Management Pack Templates**, right-click **Azure SQL Databases Monitoring** and select **Add Monitoring Wizard**.



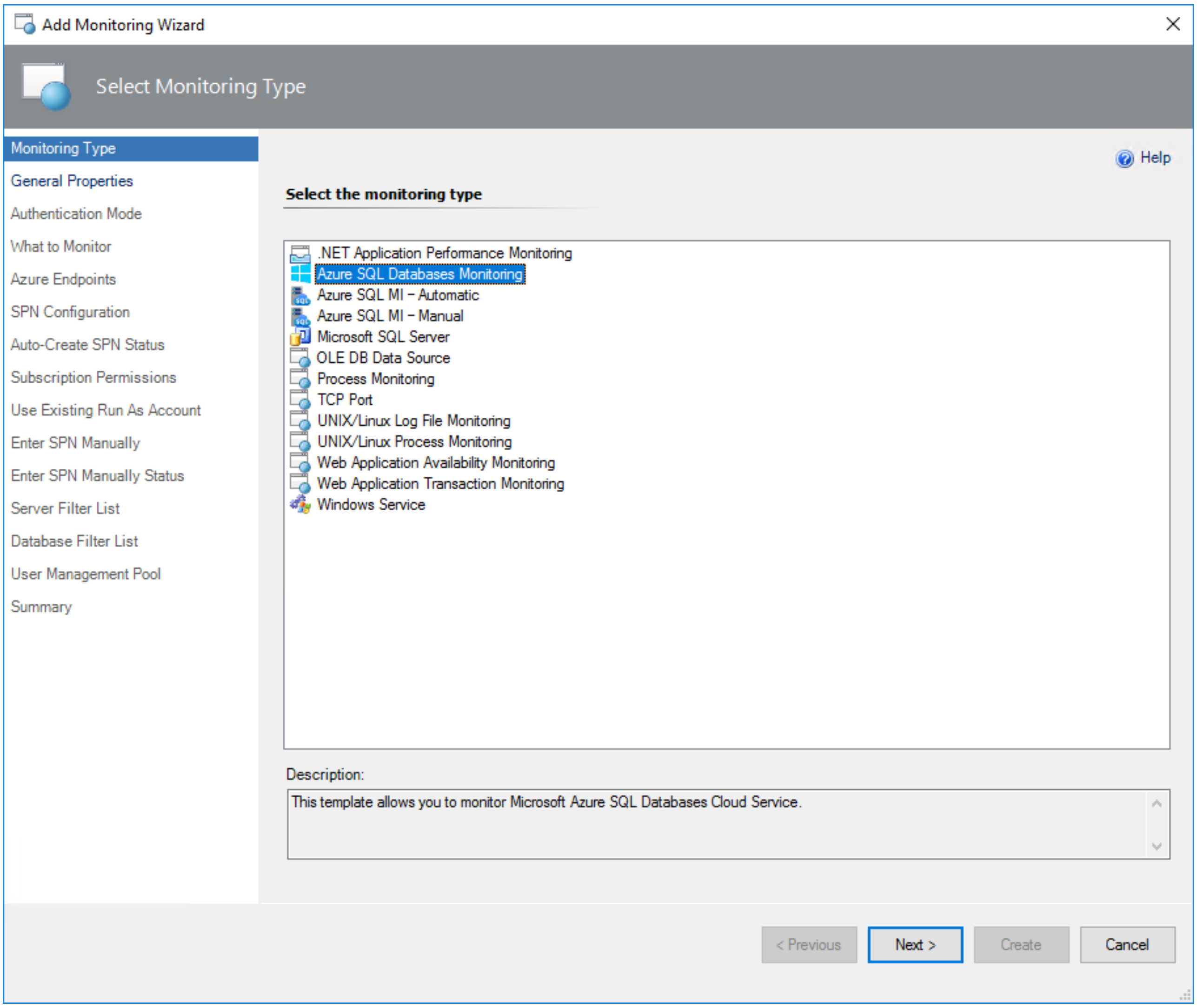
Name	Management Pack	Created
AzureMP_tsql	AzureOverrides	9/9/2020 3:40:31 PM

AzureMP_tsql

Created: 9/9/2020 3:40:31 PM

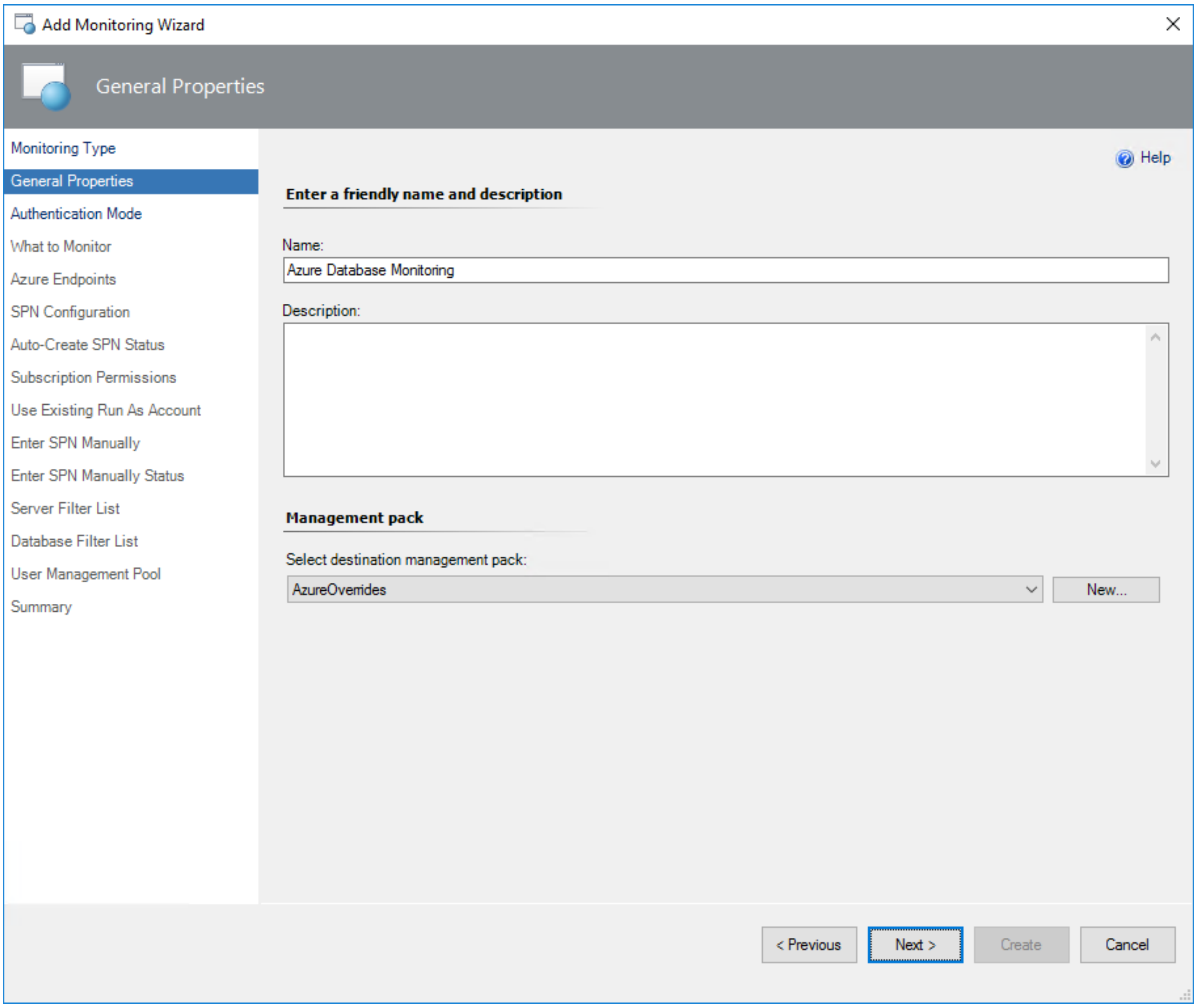
Management Pack: AzureOverrides

2. At the **Monitoring Type** step, select **Azure SQL Databases Monitoring** and click **Next**.



This template allows you to monitor Microsoft Azure SQL Databases Cloud Service.

3. At the **General Properties** step, enter a new name and description and from the **Select destination management pack** drop-down list, select a management pack that you want to use to store the template.



Enter a friendly name and description

Name: Azure Database Monitoring

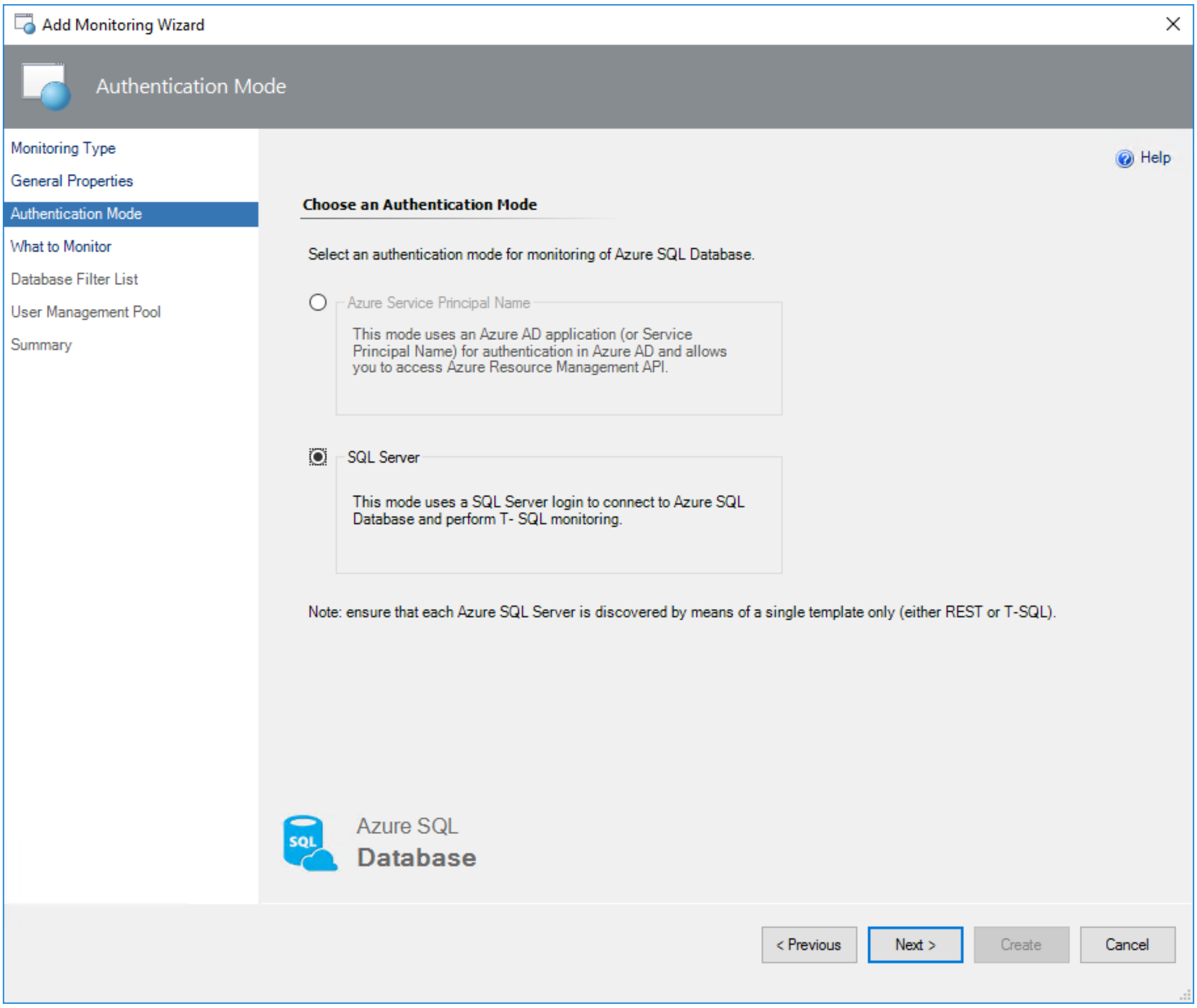
Description:

Management pack

Select destination management pack: AzureOverrides

If you do not have a management pack for this purpose, you can create a new one by clicking **New**.

4. At the **Authentication Mode** step, select **SQL Server**.



Choose an Authentication Mode

Select an authentication mode for monitoring of Azure SQL Database.

○ Azure Service Principal Name

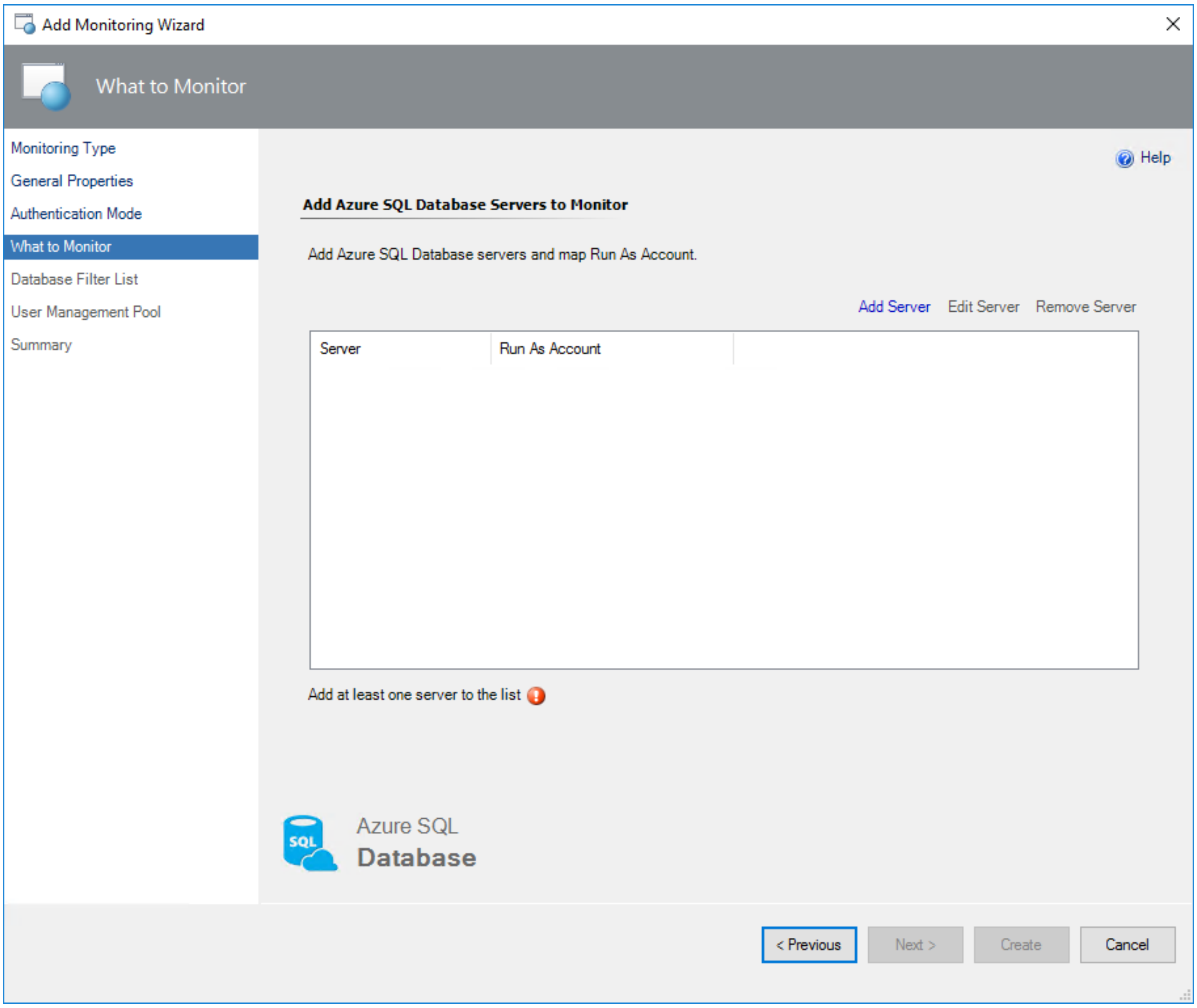
This mode uses an Azure AD application (or Service Principal Name) for authentication in Azure AD and allows you to access Azure Resource Management API.

● SQL Server

This mode uses a SQL Server login to connect to Azure SQL Database and perform T- SQL monitoring.

Note: ensure that each Azure SQL Server is discovered by means of a single template only (either REST or T-SQL).

5. At the **What to Monitor** step, click **Add Server**.



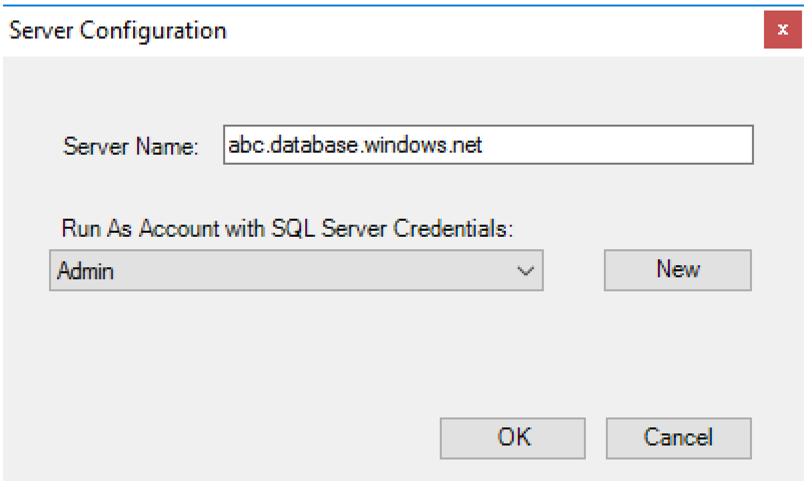
Add Azure SQL Database Servers to Monitor

Add Azure SQL Database servers and map Run As Account.

Server	Run As Account

Add at least one server to the list

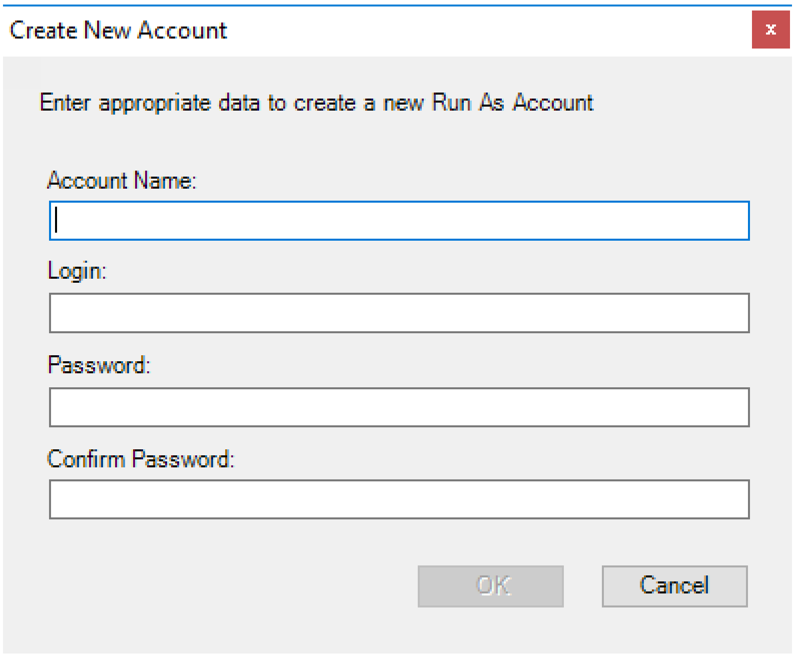
6. In the **Server Name** field, enter a name of the Azure SQL Database server that you want to monitor, select a Run As Account associated with SQL Server credentials and click **OK**. The provided SQL credentials must authorize the *System Administrator* rights.



Server Name: abc.database.windows.net

Run As Account with SQL Server Credentials: Admin

If you want to create a new Run As Account, click **New** and enter a new Run As Account name and credentials for the SQL server that you want to monitor.



Create New Account

Enter appropriate data to create a new Run As Account

Account Name:

Login:

Password:

Confirm Password:

For more information on how to create a new SQL Server authentication login, see Authorize database access to SQL Database, SQL Managed Instance, and Azure Synapse Analytics.

7. Click **Next**.

8. [Optionally] At the **Database Filter List** step, select filtering mode, which can be either **Exclude** or **Include**, enter filtering masks that should match database names that you want to exclude from or include to the monitoring list, click **Add** and then click **Next**.

   A database name cannot end with '.' or ' ' characters, contain '<,>,*,%,&,:,,/,?' or control characters, and cannot have more than 128 characters.

   For example, if you select the **Exclude** option and set the following masks:

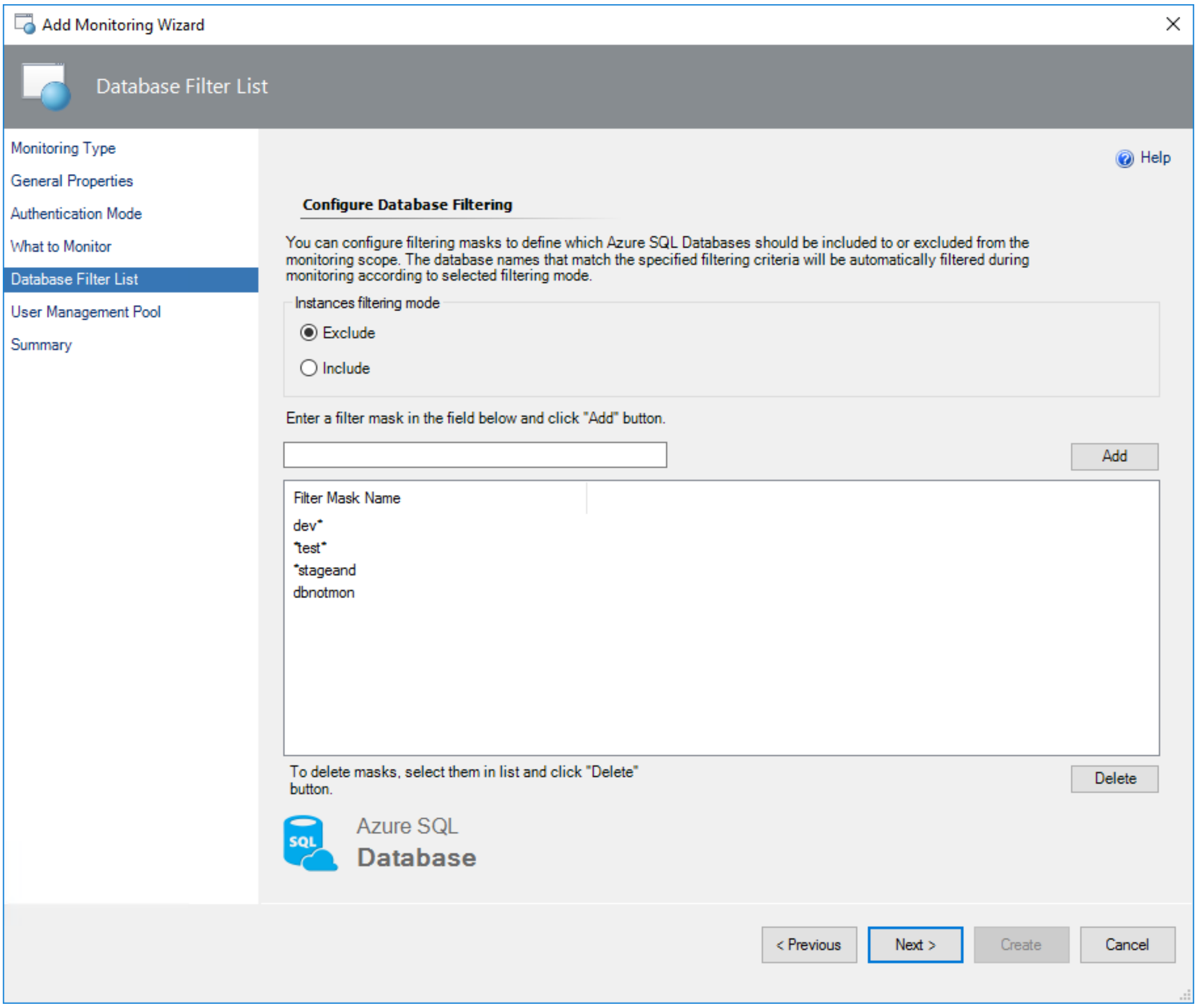
   - dev*
   - *test*
   - *stageand
   - dbnotmon

   the monitoring behavior would be as follows:

DB Name	Monitored/Not monitored
dev	Not monitored
dev_sales	Not monitored
sales_dev	Monitored
test	Not monitored
test_sales	Not monitored
sales_test	Not monitored
stage	Not monitored

DB Name	Monitored/Not monitored
stage_dev	Monitored
dev_stage	Not monitored
dbnotmon	Not monitored
dbnotmon_sales	Monitored
sales_dbnotmon	Monitored

If you want to remove an existing mask, select it and click **Delete**.



Configure Database Filtering

You can configure filtering masks to define which Azure SQL Databases should be included to or excluded from the monitoring scope. The database names that match the specified filtering criteria will be automatically filtered during monitoring according to selected filtering mode.

Instances filtering mode

● Exclude

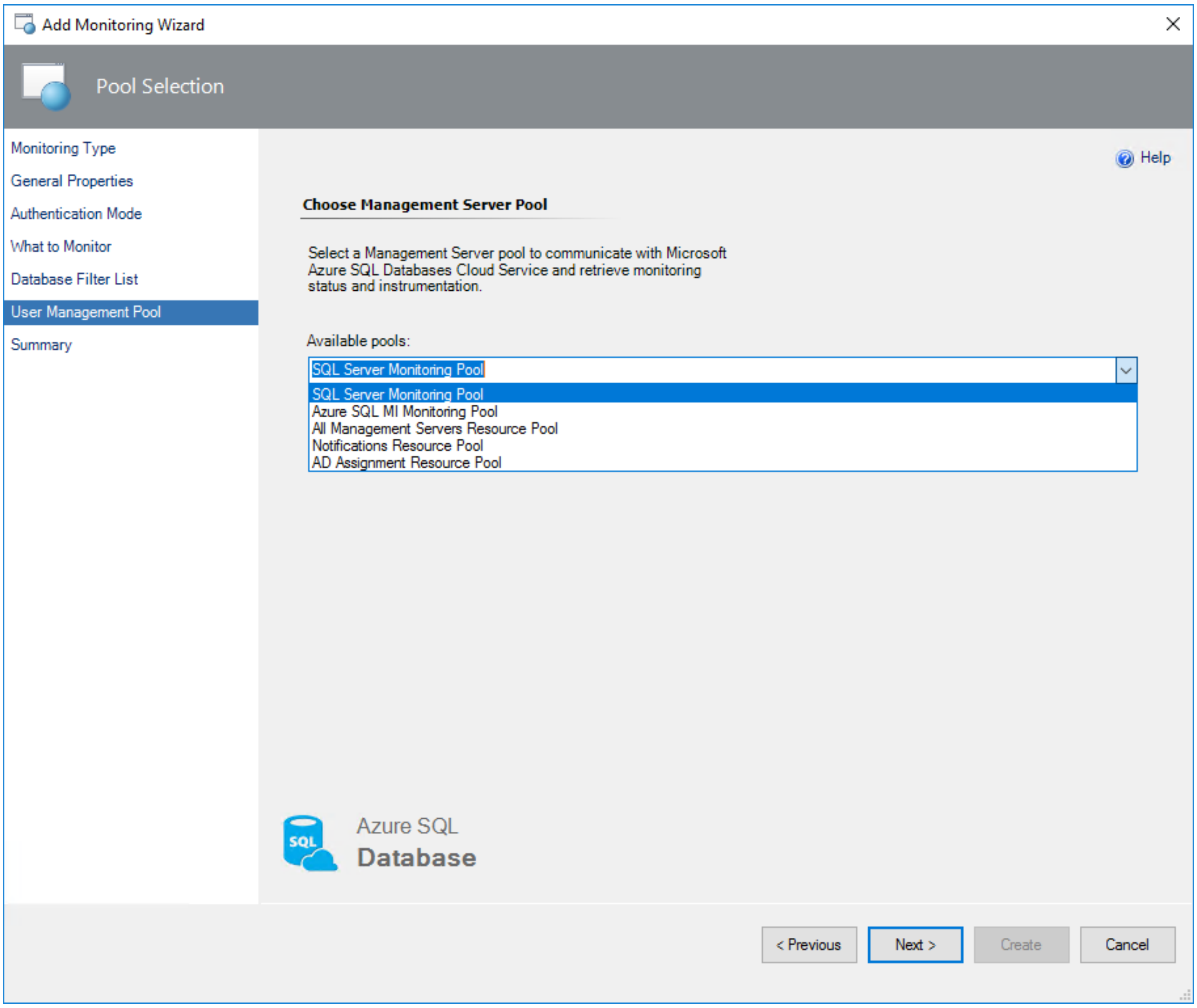
○ Include

Enter a filter mask in the field below and click "Add" button.

Filter Mask Name
dev*
test
*stageand
dbnotmon

To delete masks, select them in list and click "Delete" button.

9. At the **User Management Pool** step, select a pool with management servers and click **Next**.



Choose Management Server Pool

Select a Management Server pool to communicate with Microsoft Azure SQL Databases Cloud Service and retrieve monitoring status and instrumentation.

Available pools: SQL Server Monitoring Pool

SQL Server Monitoring Pool
Azure SQL MI Monitoring Pool
All Management Servers Resource Pool
Notifications Resource Pool
AD Assignment Resource Pool
Azure SQL Database
< Previous
Next >
Create
Cancel

10. At the **Summary** step, review connection settings and click **Create**.

Key Monitoring Scenarios

The Azure SQL Database Management Pack includes a number of key monitoring scenarios that can be configured as described below. This list, however, is not intended to be a complete manifest of the management pack functionality.

Service Availability Monitoring

The **State changes of the master database** monitor tracks availability of discovered Azure SQL Database. This monitor is not considered to be noisy and does not require any special configuration.

Service Performance Monitoring

Currently, there is a single server performance monitor that tracks situations when the number of databases per server exceeds the specified threshold.

By default, this monitor goes into the warning state when 120 or more databases are created per server and goes into the critical state when 135 or more databases are created per server.

In some situations, these default values are not appropriate. For example, an application may be designed to use all 150 databases for Azure SQL Database. When the default values would create noise, the monitor should be disabled or the thresholds should be overridden, depending on the situation.

Note that database performance monitors roll up to service performance monitoring, which can affect the health state of the service.

Service Performance Collection

Currently, there is a single rule that collects the number of databases hosted in each discovered Azure SQL Database.

Database Availability Monitoring

The **State changes of the database** monitor tracks availability of the discovered databases. This monitor is not considered to be noisy and does not require any special configuration.

Database Performance Monitoring

There are several monitors that detect when resource consumption has exceeded a predefined limit. Almost all of these monitors are disabled by default with the exception of the database free space monitor.

To use these disabled monitors create an override that adjusts the thresholds of the monitor to values appropriate for the database applications and then enable the monitor.

The database performance monitors detect:

- Excessive storage space consumed by each database.
- Excessive resources consumed by database sessions.
- Excessive resources consumed by database transactions.

Database Performance Collection

There are several rules that collect performance information about each discovered database. These rules collect information about:

- Network usage
- The amount of resources consumed by database sessions
- The amount of resources consumed by database transactions
- Disk space consumed by each database

Active Geo-Replication Monitoring

This Management Pack has the ability to monitor databases that participate in failover groups.

Active geo-replication is designed as a business continuity solution that allows the application to perform quick disaster recovery of individual databases in case of a regional disaster or large scale outage.

If geo-replication is enabled, the application can initiate failover to a secondary database in a different Azure region. For more information, see the Creating and using active geo-replication - Azure SQL Database article.

Elastic Pools Monitoring

This Management Pack has the ability to monitor databases that are part of SQL elastic pools.

Elastic pools provide a simple resource allocation mechanism for managing and scaling multiple databases that have varying and unpredictable usage demands. For more information, see the Elastic pools help you manage and scale multiple databases in Azure SQL Database article.

Custom User Query Monitoring

In addition to standard health and performance monitoring of Azure SQL Database, you can define custom T-SQL queries that allow you to monitor the application-specific health state.

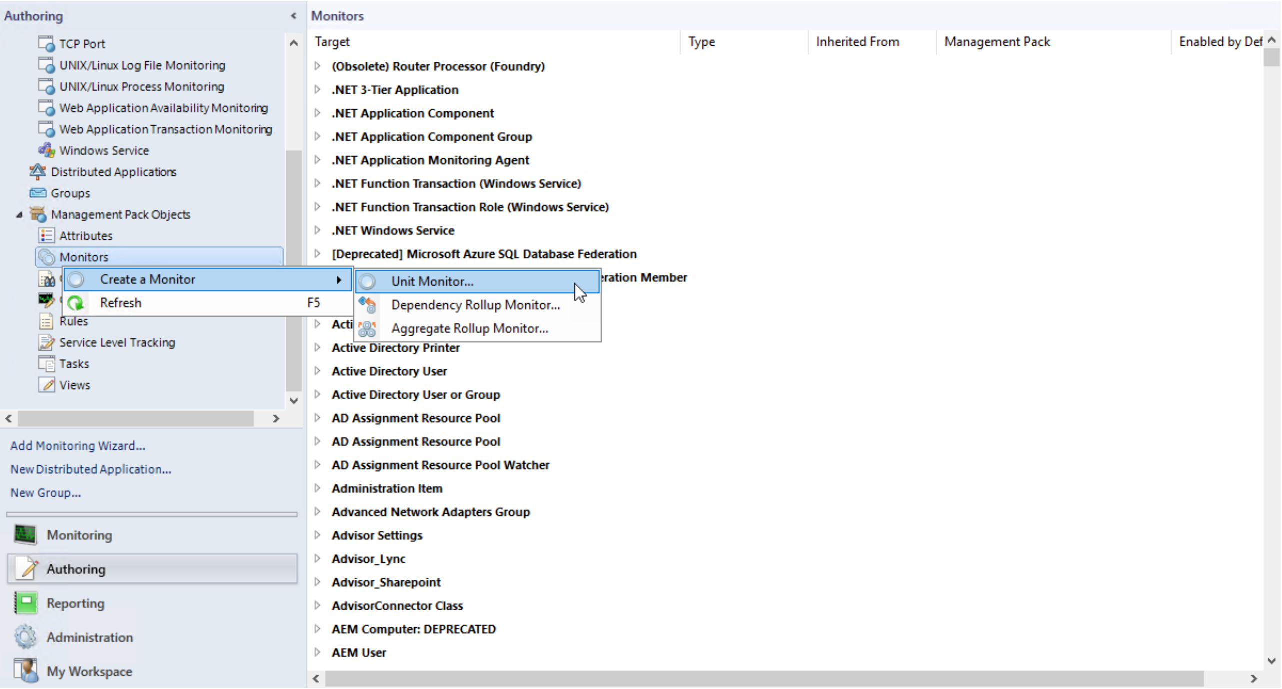
This Management pack supports two-state and three-state query-based monitors.

Before using custom query monitors, grant required permissions to accounts used for monitoring. For more information, see Configuring Azure SQL Database Run As Accounts.

Two-State Query Monitor

To add a new two-state custom query monitor, perform the following steps:

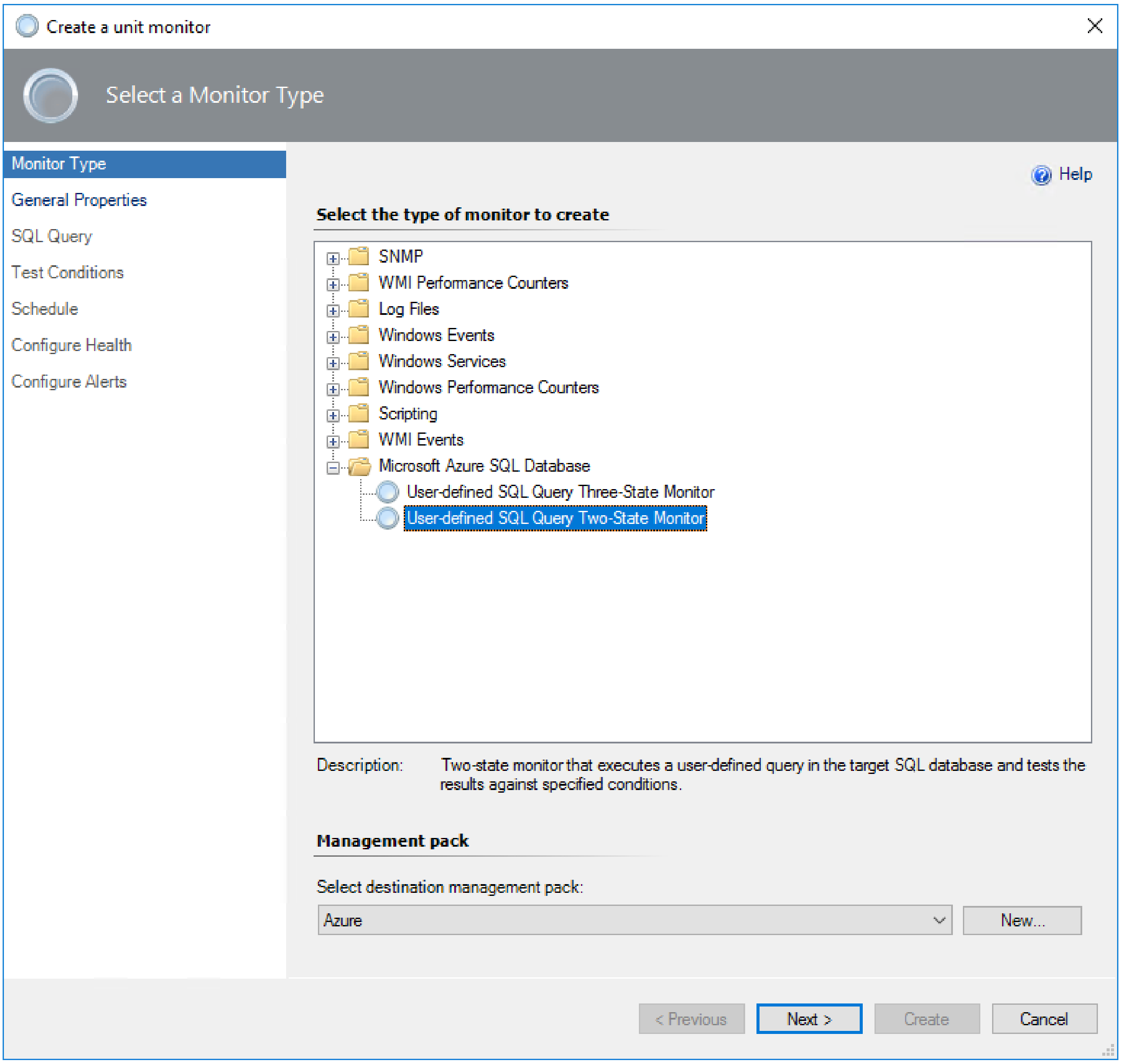
1. In the System Center Operations Manager console, navigate to **Authoring | Management Pack Objects**, right-click **Monitors** and select **Create a Monitor | Unit Monitor**.



Target	Type	Inherited From	Management Pack	Enabled by Def
(Obsolete) Router Processor (Foundry)				
.NET 3-Tier Application				
.NET Application Component				
.NET Application Component Group				
.NET Application Monitoring Agent				
.NET Function Transaction (Windows Service)				
.NET Function Transaction Role (Windows Service)				
.NET Windows Service				
[Deprecated] Microsoft Azure SQL Database Federation				
eration Member				
Acti				
Active Directory Printer				
Active Directory User				
Active Directory User or Group				
AD Assignment Resource Pool				
AD Assignment Resource Pool				
AD Assignment Resource Pool Watcher				
Administration Item				
Advanced Network Adapters Group				
Advisor Settings				
Advisor_Lync				
Advisor_Sharepoint				
AdvisorConnector Class				
AEM Computer: DEPRECATED				
AEM User				

2. At the **Monitor Type** step, select **Microsoft Azure SQL Database | User-defined SQL Query Two State Monitor**. Select destination management pack and click **Next**.

   If you want to create a custom query monitor for specific Azure SQL Database, select a management pack with the template used to monitor this service. If you want to add a query to all Azure SQL Database services, you can store the monitor in any management pack.

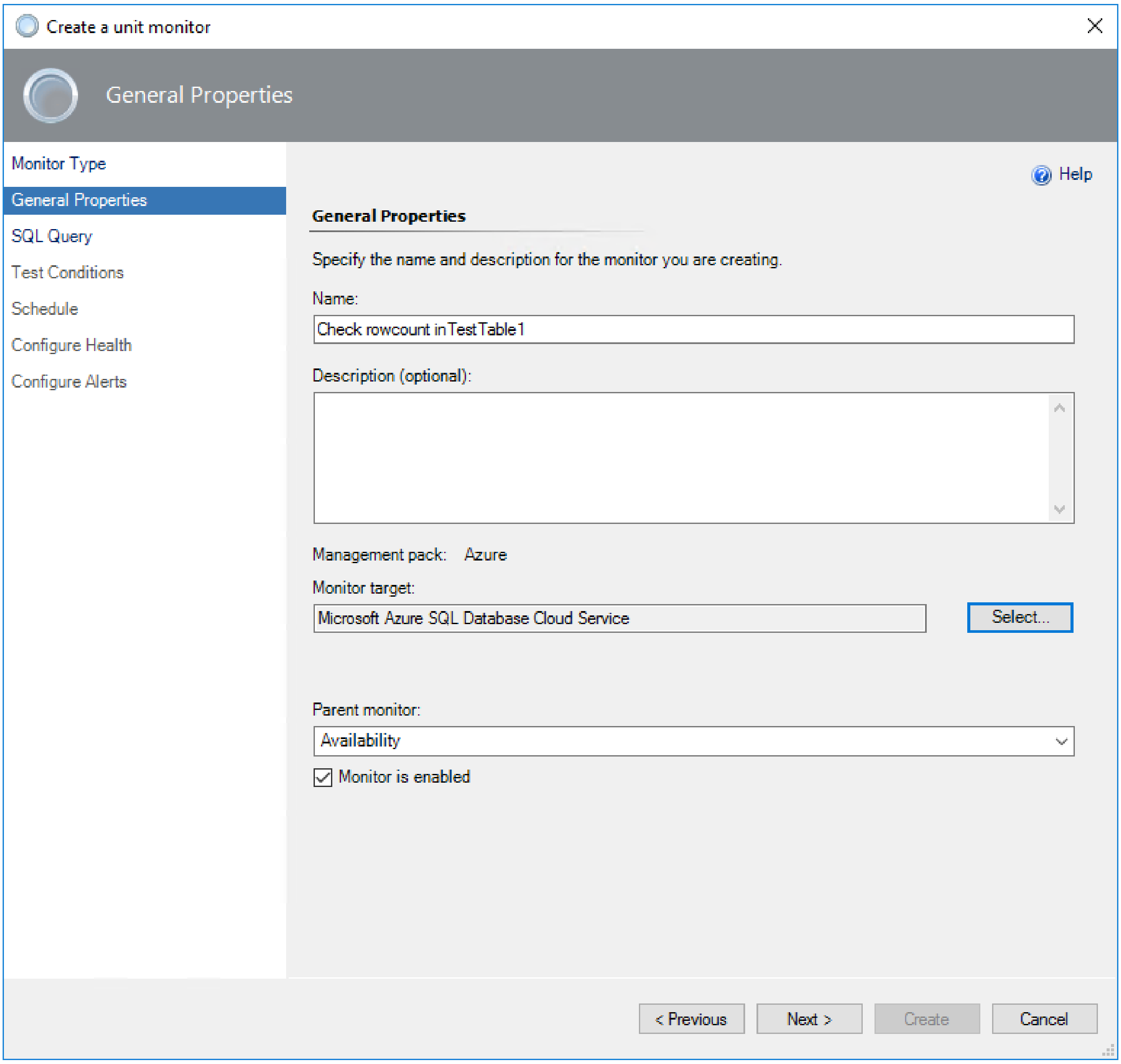


Description: Two-state monitor that executes a user-defined query in the target SQL database and tests the results against specified conditions.

Select destination management pack: Azure

3. At the **General** step, enter a monitor name and optional description, select **Monitor target** and **Parent monitor**. Click **Next**.

   If you have selected to save a new monitor to the management pack that contains one or more Azure SQL Database templates, you will be able to pick one of the Azure SQL Database services monitored by the templates. Otherwise, only base **Microsoft Azure SQL Database** will be available as a target. Selecting **Microsoft Azure SQL Database Cloud Server** means all cloud services that you monitor will be executing your query.



General Properties

Specify the name and description for the monitor you are creating.

Name: Check rowcount in TestTable1

Description (optional):

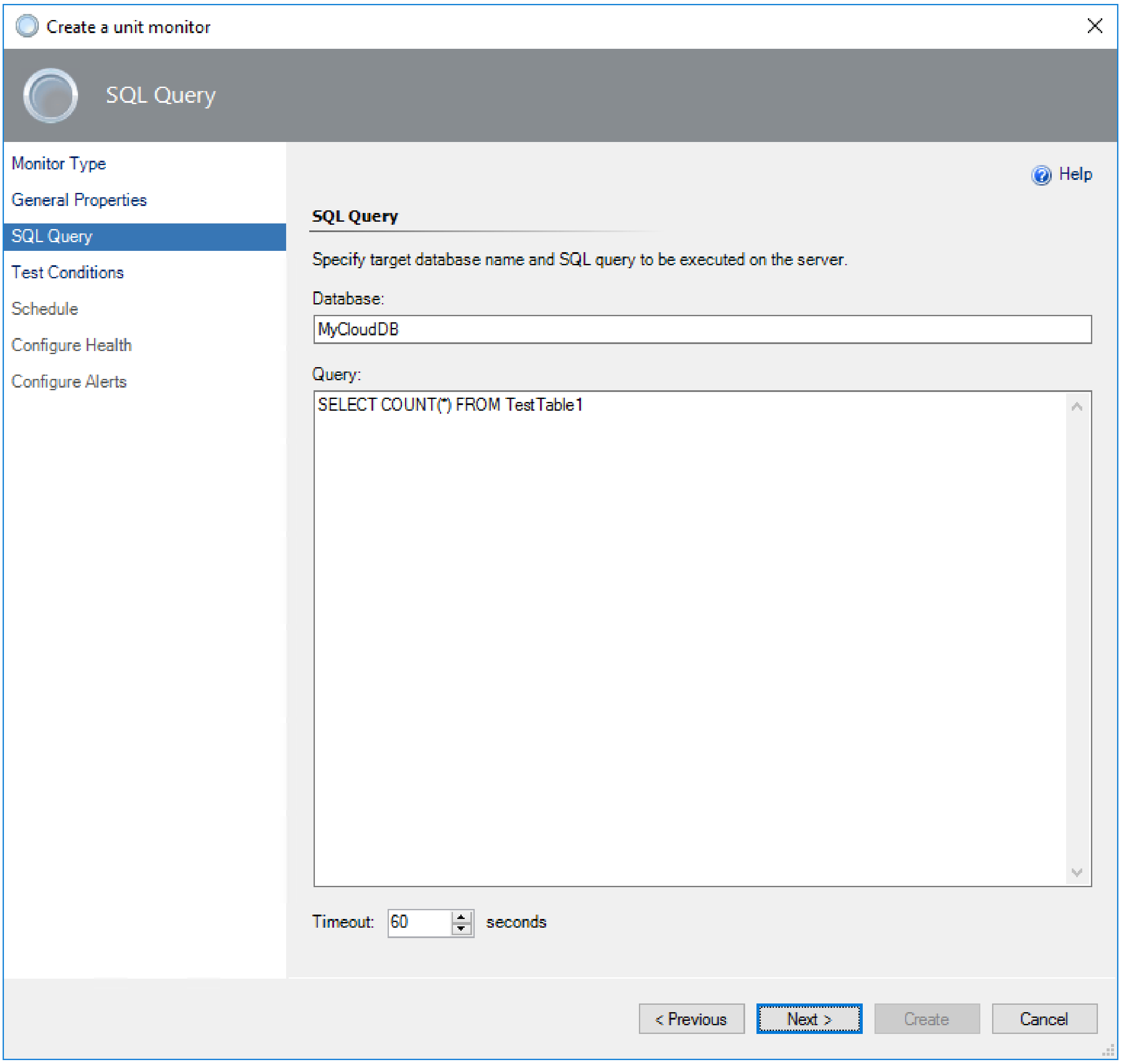
Management pack: Azure

Monitor target: Microsoft Azure SQL Database Cloud Service

Parent monitor: Availability

☑ Monitor is enabled

4. At the **SQL Query** step, enter the database name, query text, and timeout (in seconds).



SQL Query

Specify target database name and SQL query to be executed on the server.

Database: MyCloudDB

Query:

```
SELECT COUNT(*) FROM TestTable1
```

Timeout: 60 seconds

5. At the **Test Conditions** step, add one or more **Test conditions** to verify query results.

   To add a new condition, click **Add** and pick one of the available conditions from the drop-down list:

   - **Empty Result Set**

     Checks if the specified result set returned by the query is empty.

   - **Not Empty Result Set**

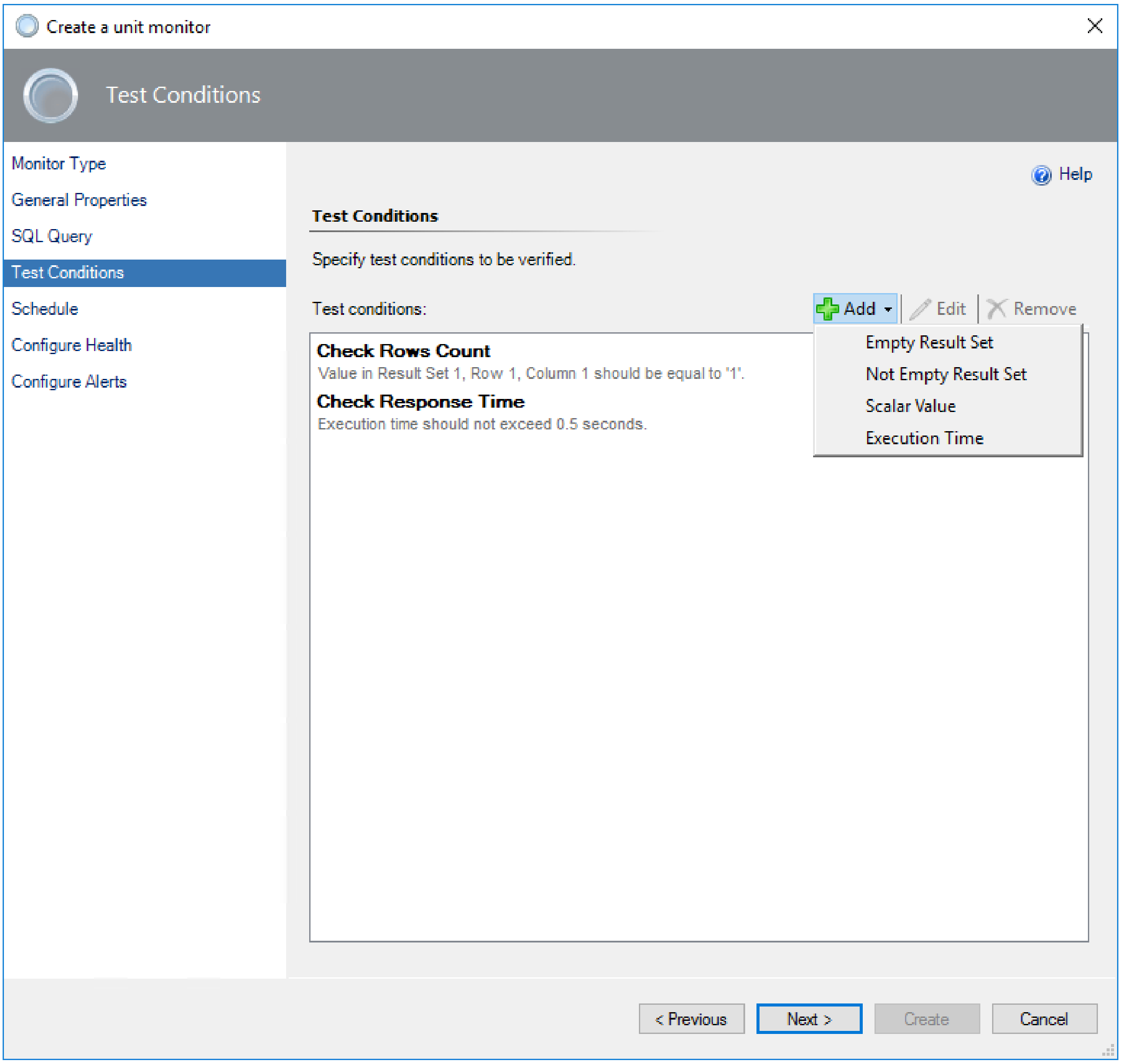
     Checks if the specified result set returned by the query is not empty.

   - **Scalar Value**

     Checks the scalar value in the specified cell of the result set. Only equal comparison is available at this moment; if you need complex logic, you must cover that by the query.

   - **Execution Time**

     Checks execution duration of the query.



Test Conditions

Specify test conditions to be verified.

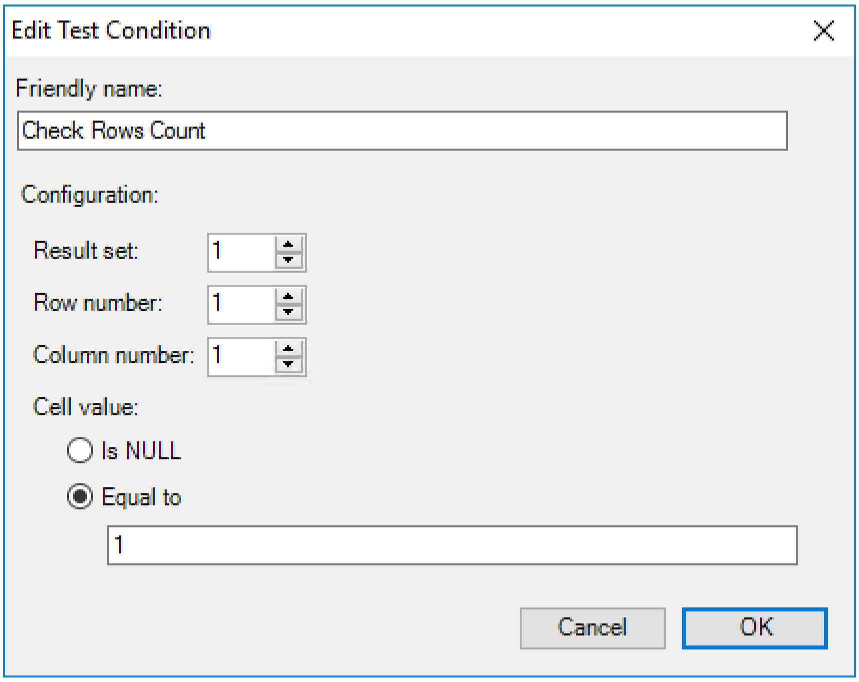
Test conditions:

Check Rows Count
Value in Result Set 1, Row 1, Column 1 should be equal to '1'.

Check Response Time
Execution time should not exceed 0.5 seconds.

When you add a condition, you must specify the **Friendly name** and the entire **configuration** required for a specific check to be performed.

We will be using the **Scalar Value** condition to verify the rows count in *TestTable1*.


Edit Test Condition

Friendly name: Check Rows Count

Configuration:

Result set: 1

Row number: 1

Column number: 1

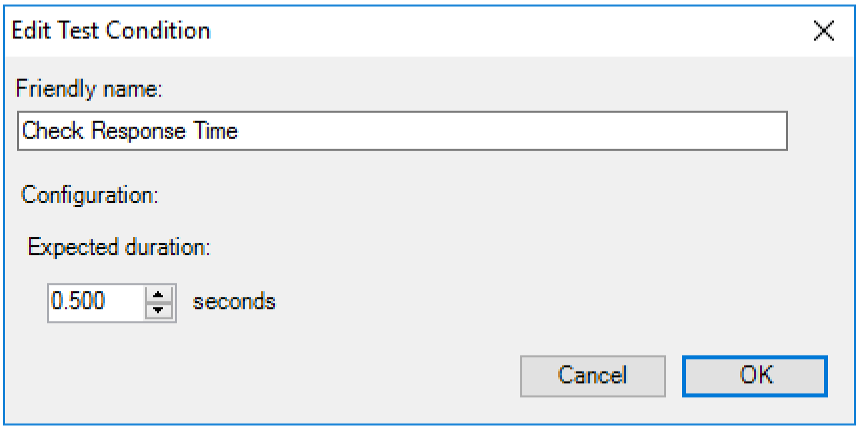
Cell value:

○ Is NULL

● Equal to

1

You can have more than one condition. It is useful to add the **Execution Time** condition to all tests to check how your Azure SQL Database service is performing.


Edit Test Condition

Friendly name: Check Response Time

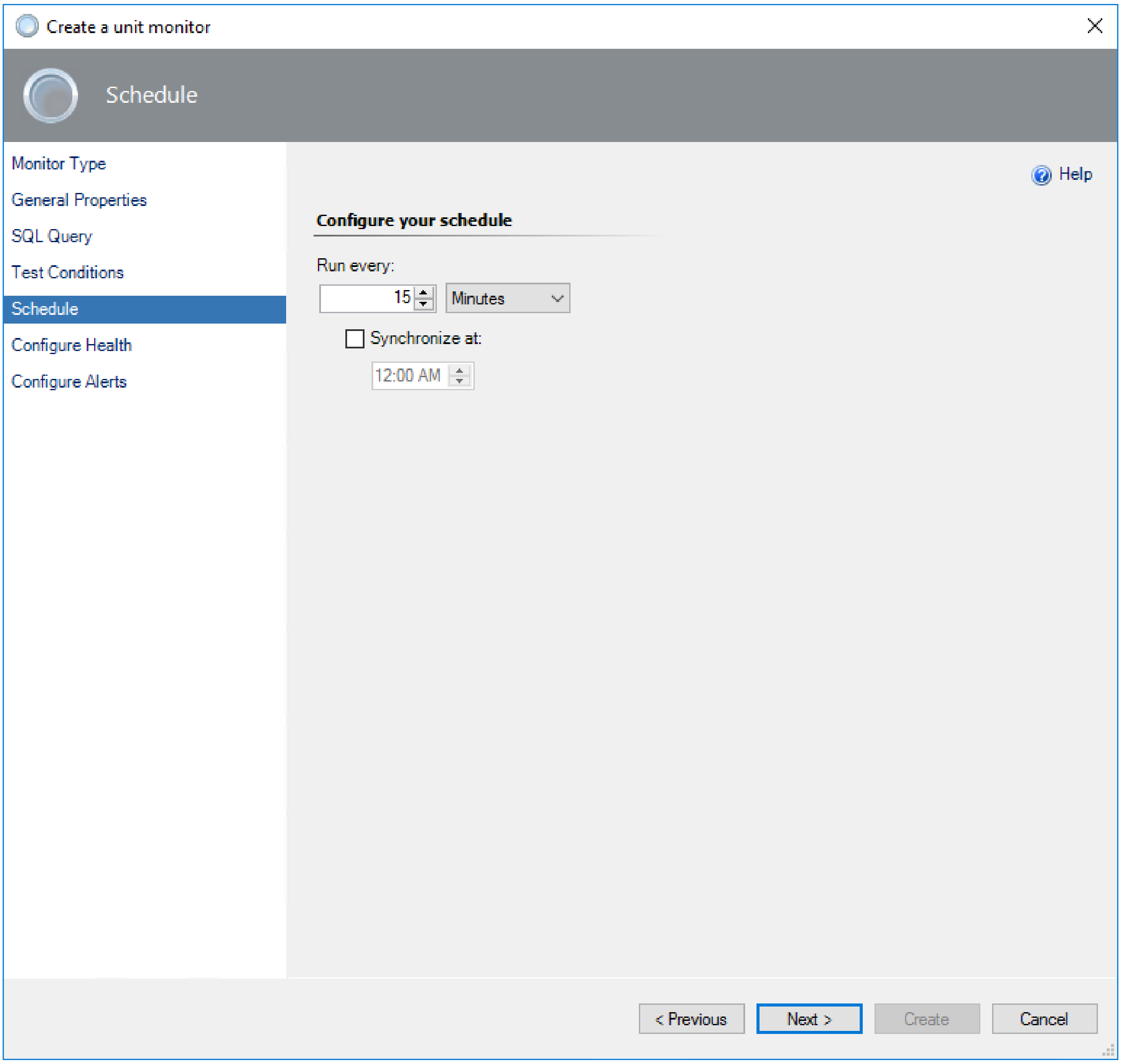
Configuration:

Expected duration:

0.500 seconds

After all required conditions are configured, click **Next**.

6. At the **Schedule** page, specify how frequently your query will be executed.



Configure your schedule

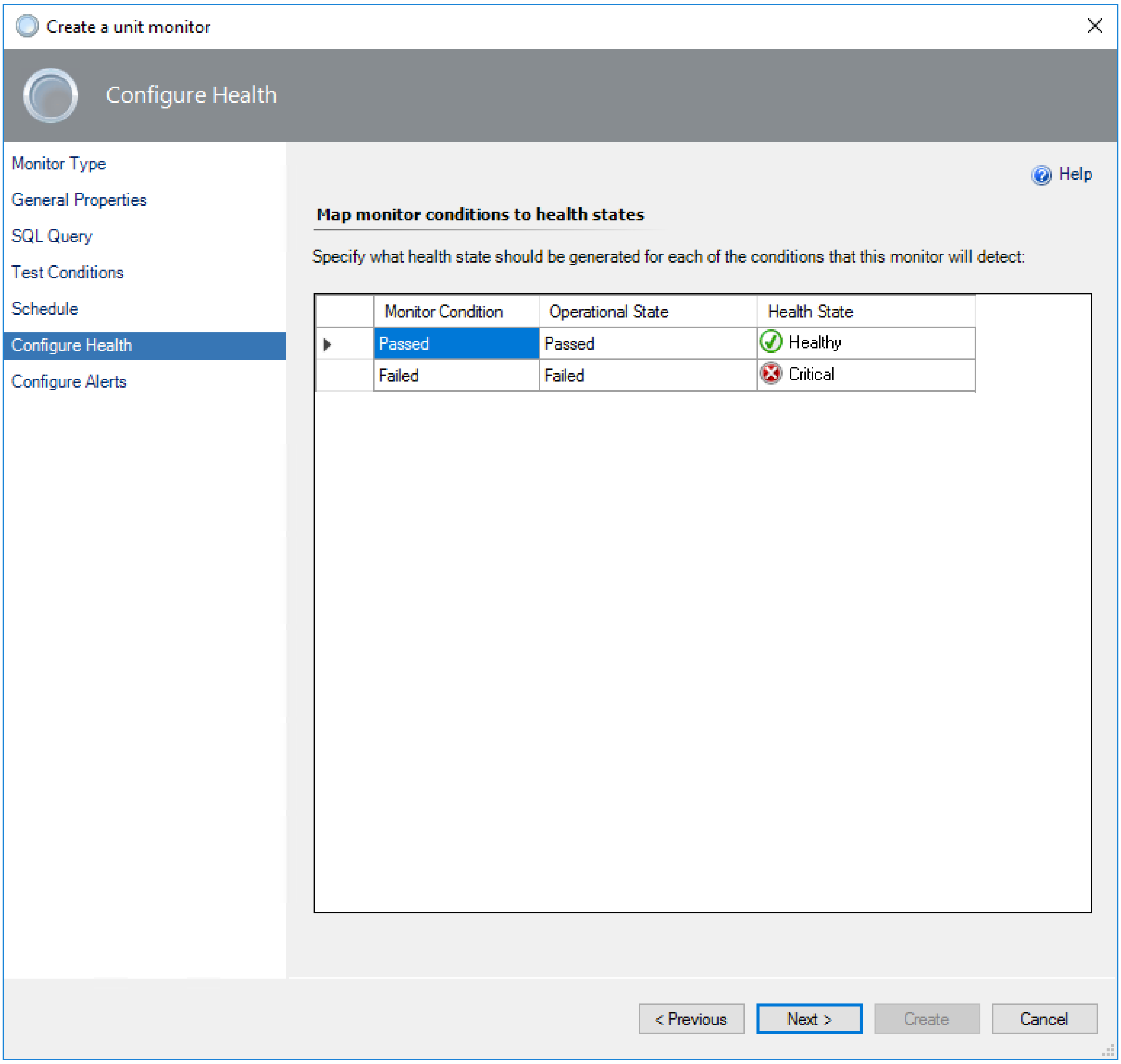
Run every:

15 Minutes

□ Synchronize at:

12:00 AM

7. At the **Configure Health** step, select what health state should be generated by the monitor.



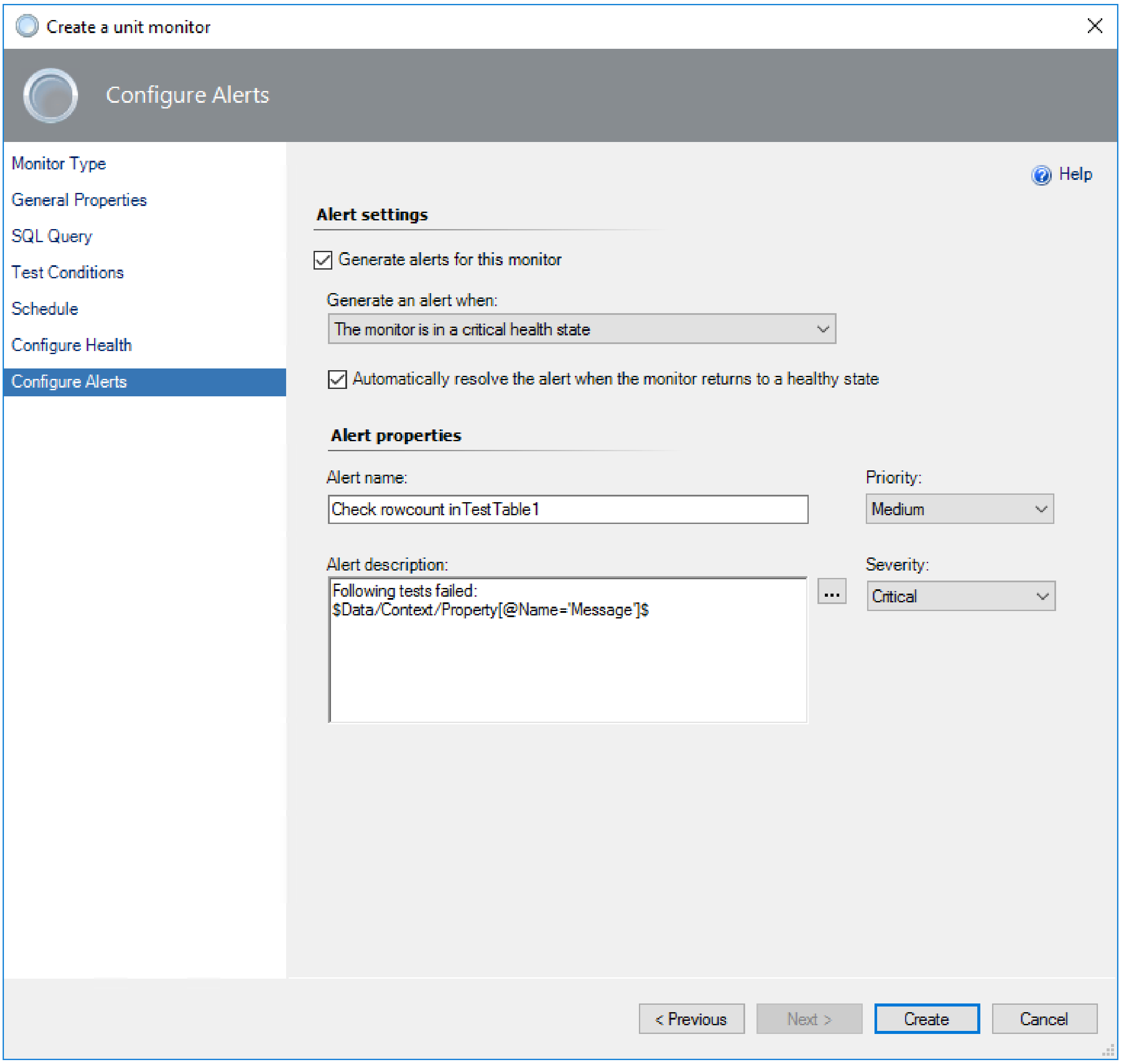
Map monitor conditions to health states

Specify what health state should be generated for each of the conditions that this monitor will detect:

Monitor Condition	Operational State	Health State
Passed	Passed	✓ Healthy
Failed	Failed	☒ Critical

8. At the **Configure Alerts** step, set up an alert name and description to be shown in cases if one or more test conditions fails.

> Use $Data/Context/Property[@Name='Message']$ placeholder to show a list of failed tests in the alert description.



Alert settings

☑ Generate alerts for this monitor

Generate an alert when: The monitor is in a critical health state

☑ Automatically resolve the alert when the monitor returns to a healthy state

Alert properties

Alert name: Check rowcount in TestTable1

Priority: Medium

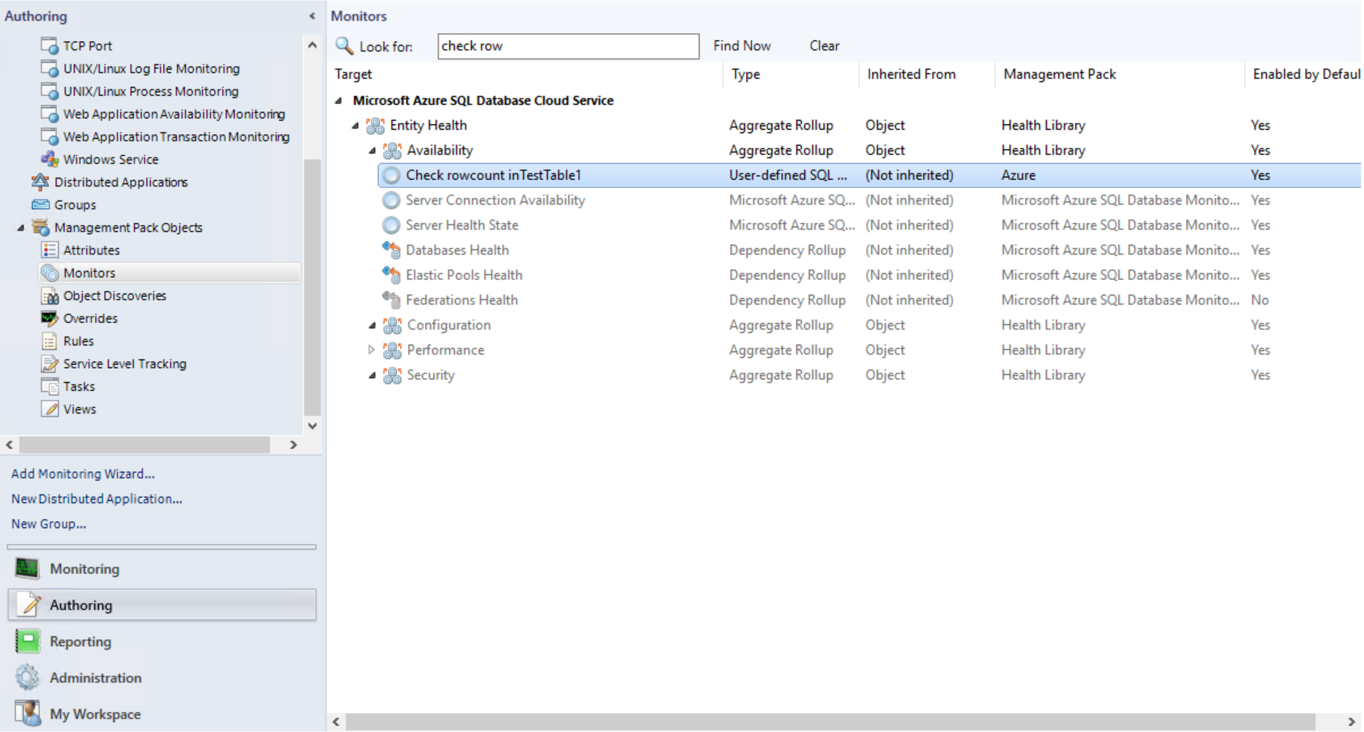
Alert description:

```
Following tests failed:
$Data/Context/Property[@Name='Message']$
```

Severity: Critical

9. Click **Create**.

Once completed, a new monitor becomes available.

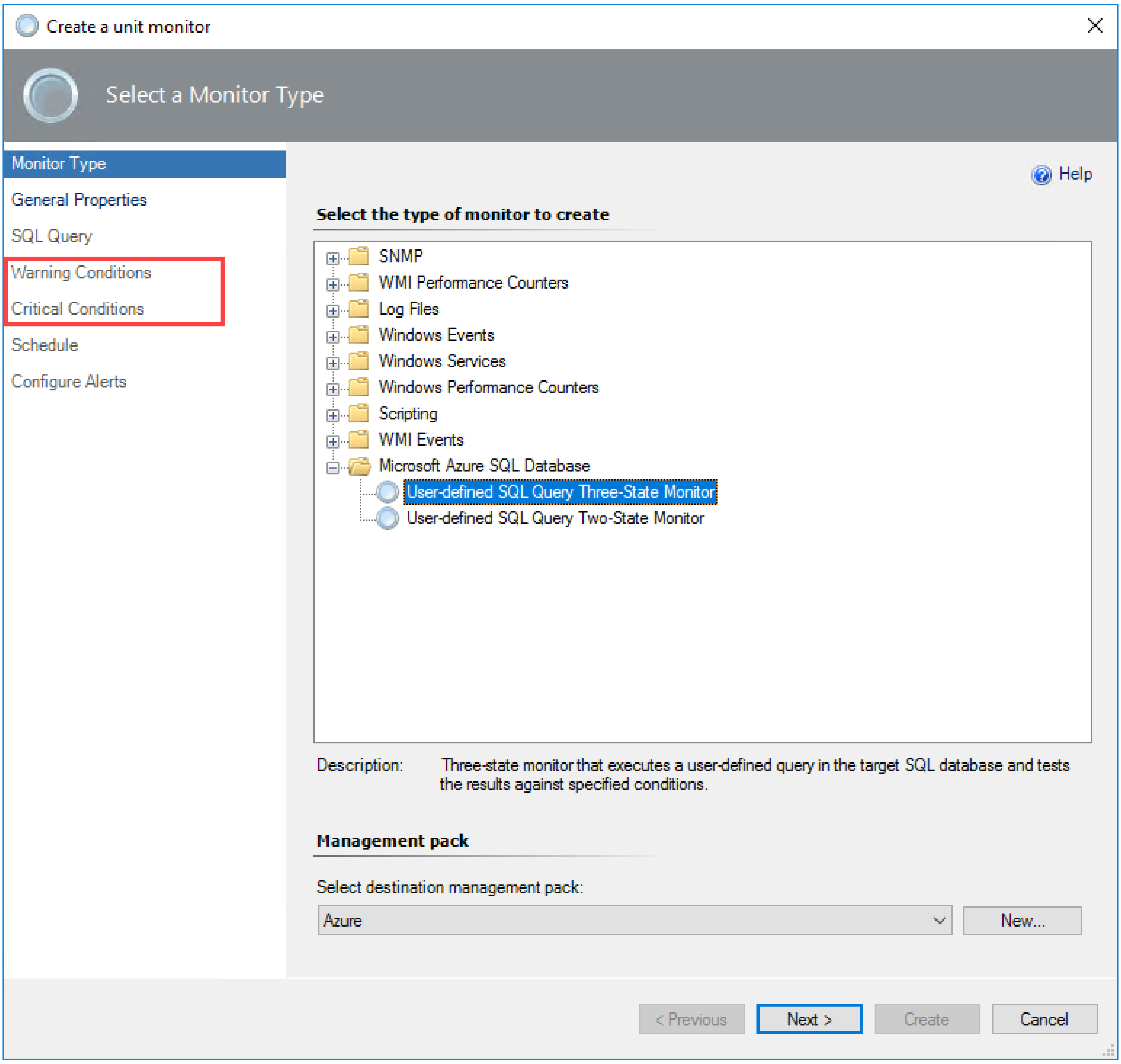


Look for: check row

Target	Type	Inherited From	Management Pack	Enabled by Defaul
Microsoft Azure SQL Database Cloud Service				
Entity Health	Aggregate Rollup	Object	Health Library	Yes
Availability	Aggregate Rollup	Object	Health Library	Yes
Check rowcount inTestTable1	User-defined SQL ...	(Not inherited)	Azure	Yes
Server Connection Availability	Microsoft Azure SQ...	(Not inherited)	Microsoft Azure SQL Database Monito...	Yes
Server Health State	Microsoft Azure SQ...	(Not inherited)	Microsoft Azure SQL Database Monito...	Yes
Databases Health	Dependency Rollup	(Not inherited)	Microsoft Azure SQL Database Monito...	Yes
Elastic Pools Health	Dependency Rollup	(Not inherited)	Microsoft Azure SQL Database Monito...	Yes
Federations Health	Dependency Rollup	(Not inherited)	Microsoft Azure SQL Database Monito...	No
Configuration	Aggregate Rollup	Object	Health Library	Yes
Performance	Aggregate Rollup	Object	Health Library	Yes
Security	Aggregate Rollup	Object	Health Library	Yes

Three-State Query Monitor

Adding a new three-state query monitor is similar to a two-state monitor. The main difference is that you must specify *Warning* and *Critical* conditions.



Description: Three-state monitor that executes a user-defined query in the target SQL database and tests the results against specified conditions.

Select destination management pack: Azure

Critical conditions are verified first. If one or more critical conditions fail, the monitor will switch to the critical state and warning conditions will not be verified.

For more information on how to setup query and conditions, see the Two-State Query Monitor section above.

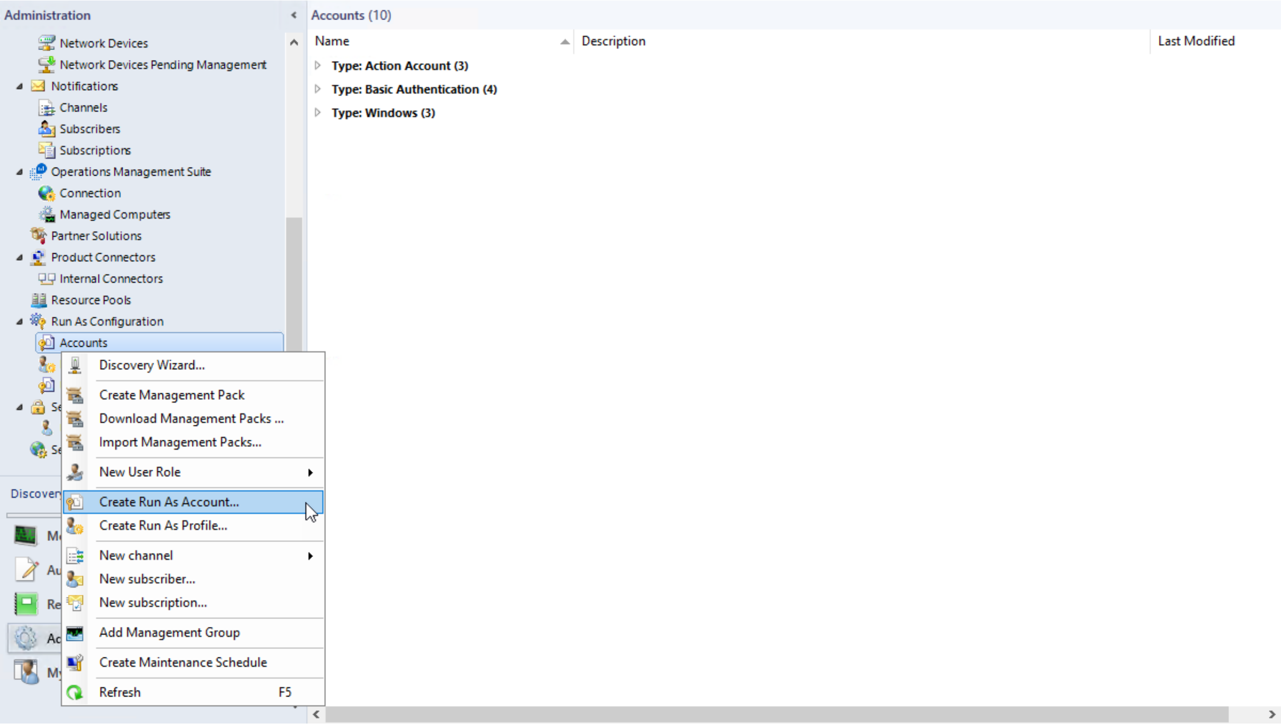
Security Configuration

Configuring Azure SQL Database Run As Accounts

To monitor Azure SQL Database servers, create one or more **Simple** or **Basic** authentication Run As accounts.

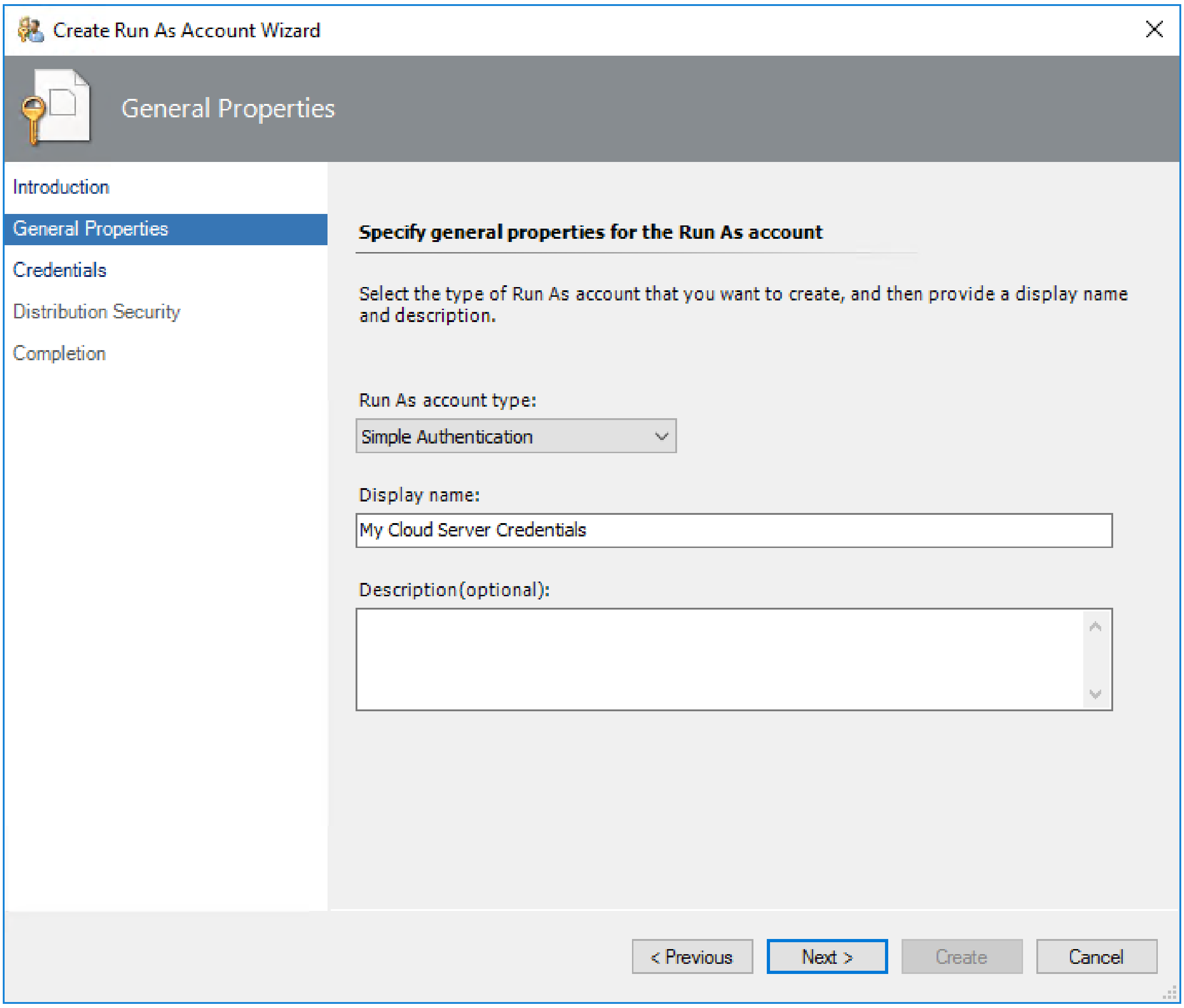
To create Run As accounts, perform the following steps:

1. In the System Center Operations Manager console, right-click the **Administration | Run As Configuration | Accounts** node and select **Create Run As Account**.



2. At the **Introduction** step, click **Next**.

3. At the **General Properties** step, from the **Run As account type** drop-down list, select *Simple Authentication*, enter a display name and optional description and click **Next**.



Specify general properties for the Run As account

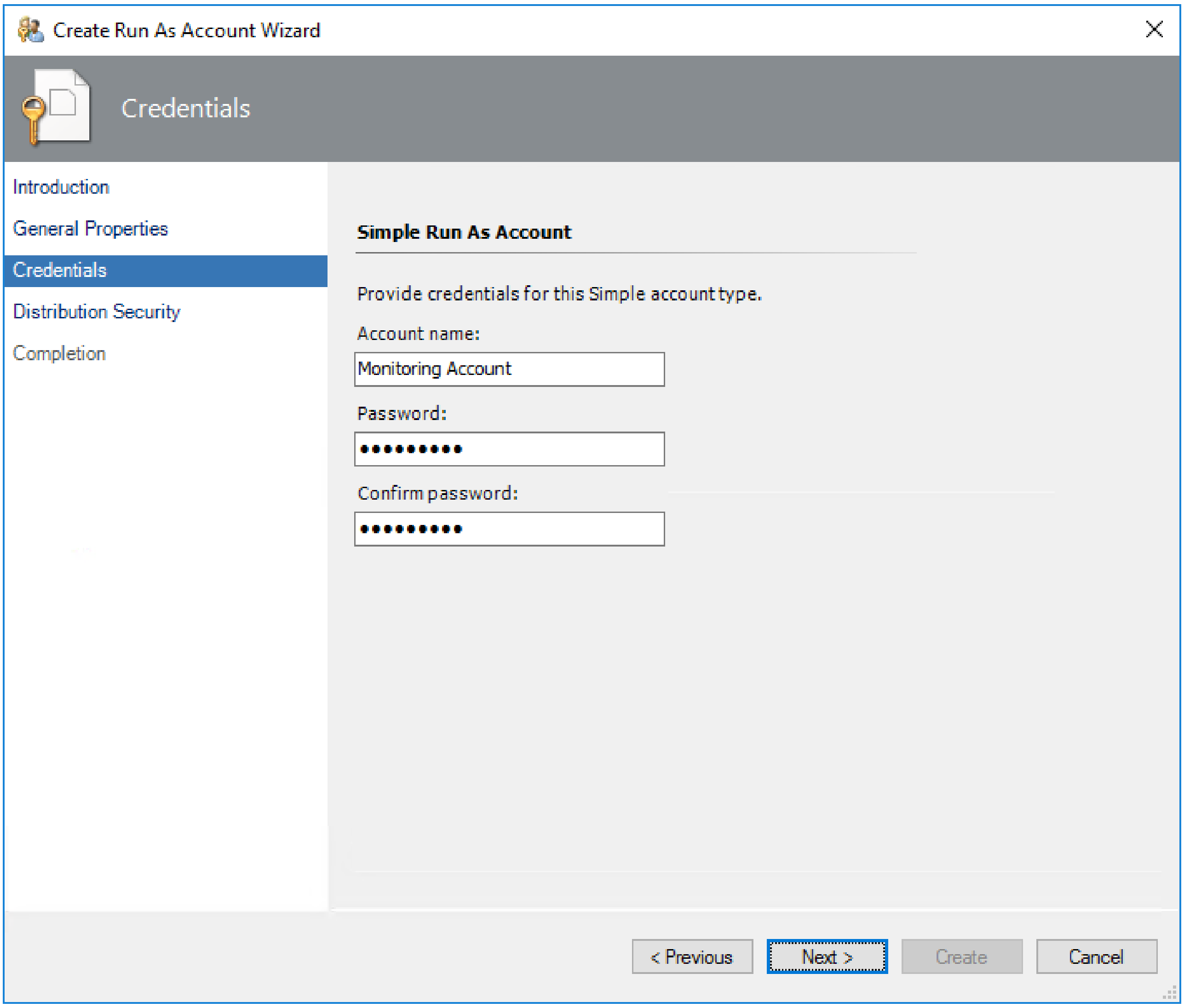
Select the type of Run As account that you want to create, and then provide a display name and description.

Run As account type: Simple Authentication

Display name: My Cloud Server Credentials

Description(optional):

4. At the **Credentials** step, specify credentials that you want to use to connect to Azure SQL Database and click **Next**. For more information, see the Low-Privilege Configuration section.



Simple Run As Account

Provide credentials for this Simple account type.

Account name: Monitoring Account

Password: •••••••••

Confirm password: •••••••••

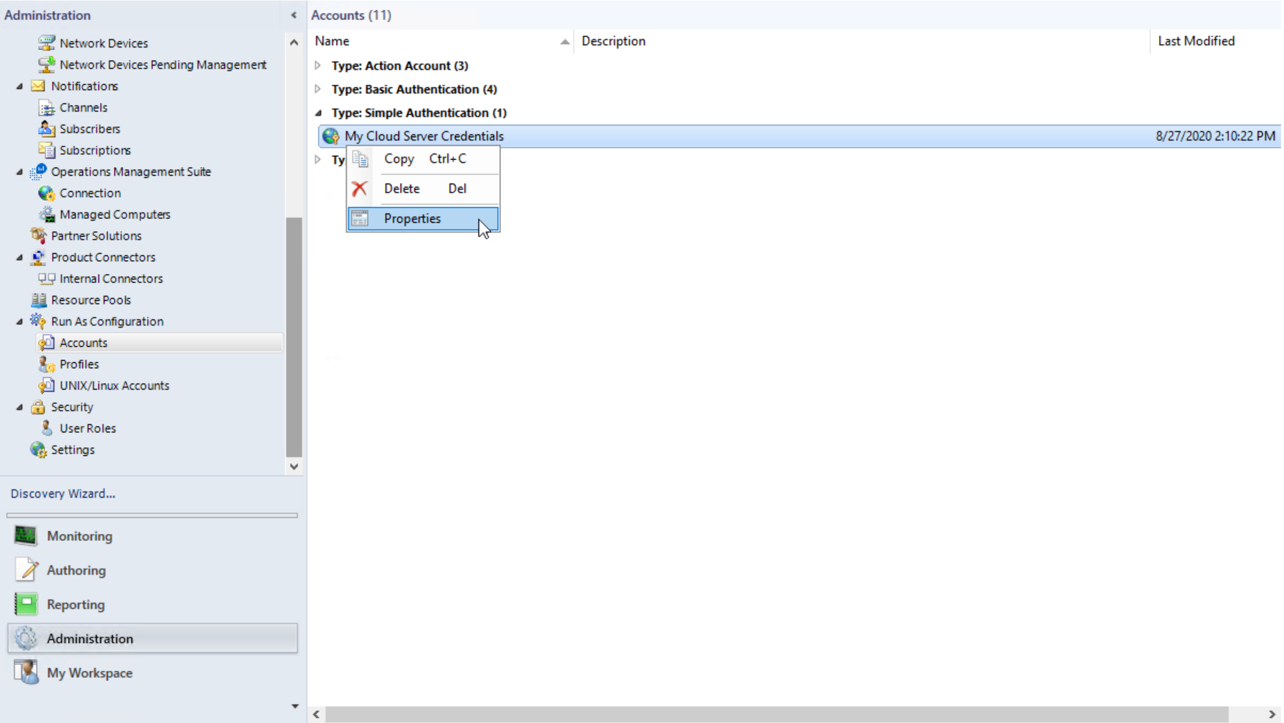
5. At the **Distribution Security** step, select the **More secure** option and click **Create**.

   You can use the **Less secure** option and skip steps 7 – 8 if your environment is secure.

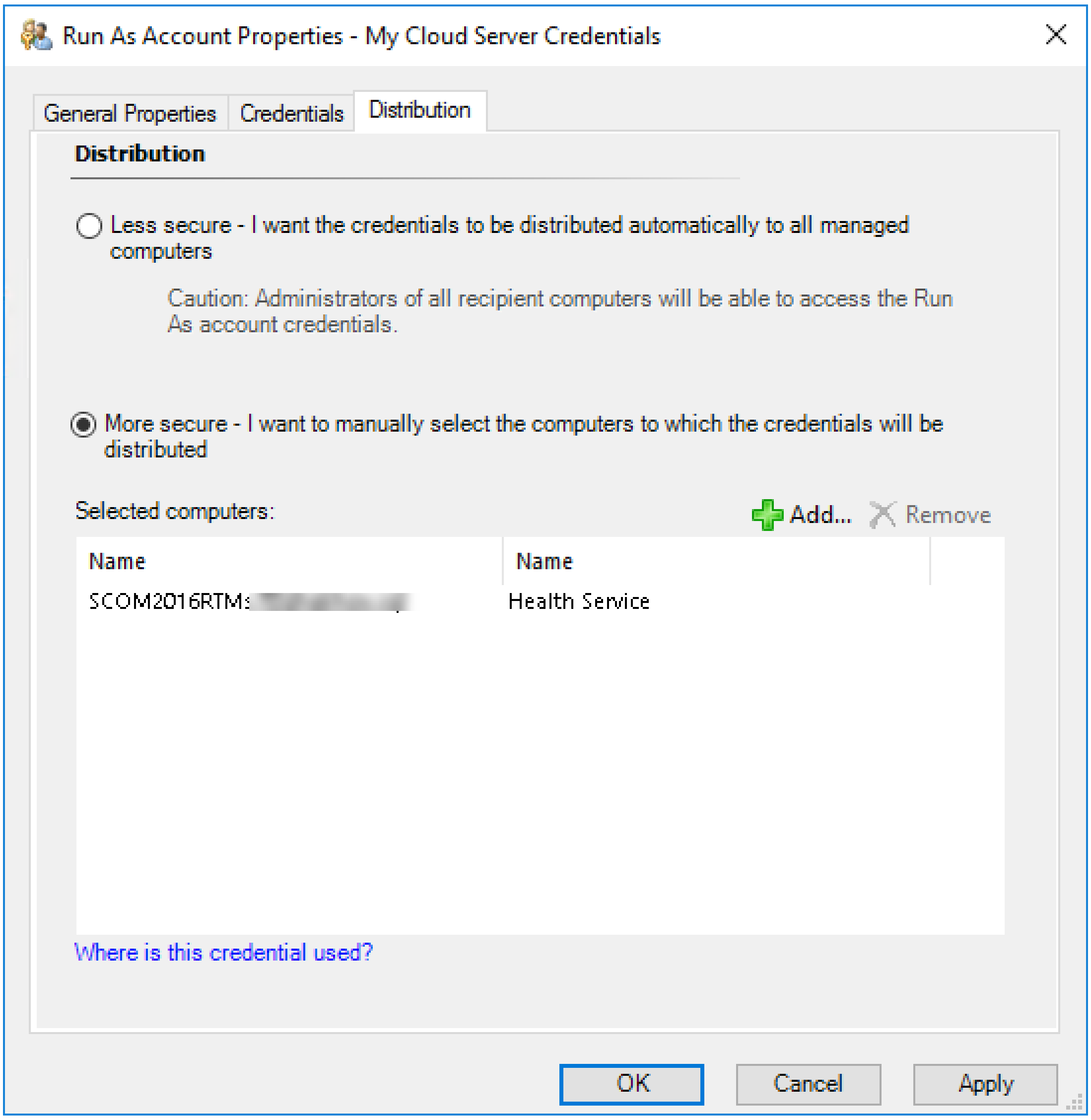
6. Click **Close** to close the window.

   If you select the **Less secure** option on step 5, you can skip the next steps.

7. Right-click the newly created account and select **Properties**.



8. Navigate to the **Distribution** tab and add a System Center Operations Manager agent that you want to use as a watcher node to monitor Azure SQL Database.


Run As Account Properties - My Cloud Server Credentials
General Properties
Credentials
Distribution
Add...
Remove
Where is this credential used?
OK
Cancel
Apply

Distribution

○ Less secure - I want the credentials to be distributed automatically to all managed computers

Caution: Administrators of all recipient computers will be able to access the Run As account credentials.

● More secure - I want to manually select the computers to which the credentials will be distributed

Selected computers:

Name	Name
SCOM2016RTMs[redacted]	Health Service

For more information about Run As accounts, see the Managing Run As Accounts and Profiles article.

Low-Privilege Configuration

Since Azure SQL Database service evolving very fast, some of the permissions required for monitoring may change over time; use an Administrator account.

The following steps will allow you to set up a low-privilege account to monitor the service:

1. Connect to the master database and create server-level credentials for low-privilege monitoring user by means of the following query:

   ```
   CREATE LOGIN [MonitoringUser] WITH PASSWORD = <'YourPassword'>
   ```

2. Connect to the *master* database and map server-level login to the database user by executing the following query:

   ```
   CREATE USER [MonitoringUser] FOR LOGIN [MonitoringUser] WITH DEFAULT_SCHEMA
   = sys
   ```

3. In every user database (excluding *master* members), map server-level login to the database user and grant it the *VIEW DATABASE STATE* permission by executing the following command:

   ```
   CREATE USER [MonitoringUser] FOR LOGIN [MonitoringUser] WITH DEFAULT_SCHEMA
   = sys
   GO
   GRANT VIEW DATABASE STATE TO [MonitoringUser]
   ```

Use the *MonitoringUser* value when Configuring Azure SQL Database Run As Accounts.

⚠ Important! If you are using Custom User Query Monitoring, you must grant all required permissions to this account as well. Custom query monitors use these credentials to execute all queries.

Performance and Billing Considerations

[Applicable to T-SQL monitoring only]

Since this management pack retrieves data from Azure SQL Database, you will be charged for the amount of data transferred outside the Microsoft Azure environment. Although management pack queries are designed to be executed fast and retrieve a small amount of data, keep monitoring and discovery intervals as high as possible to reduce both the load and the amount of transferred data.

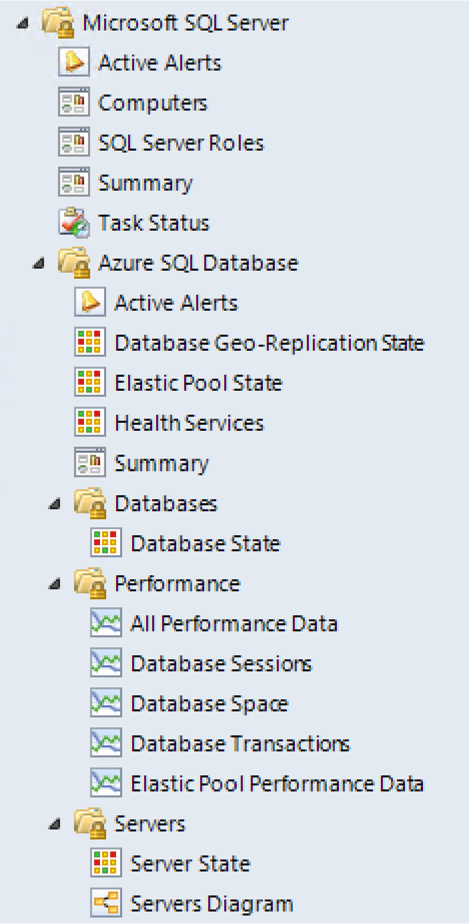
If you are not interested in certain metrics collected by this management pack, disable them.

Viewing Information in the Operations Console

You can observe a high-level view of object types in your Azure SQL Database service.

A view can contain a lengthy list of objects. To find a specific object or group of objects, you can use the **Scope**, **Search**, and **Find** buttons on the Operations Manager toolbar. For more information, see Finding Data and Objects in the Operations Manager Consoles.

The following views are provided by the Azure SQL Database management pack and available under the **Azure SQL Database** node in the **Monitoring** pane of the Operations Manager console:



- Microsoft SQL Server
  - Active Alerts
  - Computers
  - SQL Server Roles
  - Summary
  - Task Status
  - Azure SQL Database
    - Active Alerts
    - Database Geo-Replication State
    - Elastic Pool State
    - Health Services
    - Summary
    - Databases
      - Database State
    - Performance
      - All Performance Data
      - Database Sessions
      - Database Space
      - Database Transactions
      - Elastic Pool Performance Data
    - Servers
      - Server State
      - Servers Diagram

Database Views

The following table describes views that show a databases health state.

View Path	Description
Databases\Database State	Displays a list of monitored databases and their current states. Double-click the health state icon for a database to launch a Health Explorer window for that databases to locate monitors that affect the health state of the server and investigate any issue. The **Detail View** pane displays properties of the database selected above.
Performance\Database Sessions	The **Legend** pane displays a list of database sessions related counters for every monitored database. The chart illustrates information selected in the **Legend** pane.

View Path	Description
Performance\Database Space	The **Legend** pane displays a list of disk space related counters for every monitored database. The chart illustrates information selected in the **Legend** pane.
Performance\Database Transactions	The **Legend** pane displays a list of database transactions related counters for every monitored database. The chart illustrates information selected in the **Legend** pane.

Server Views

The following table describes views that show a cloud services health state.

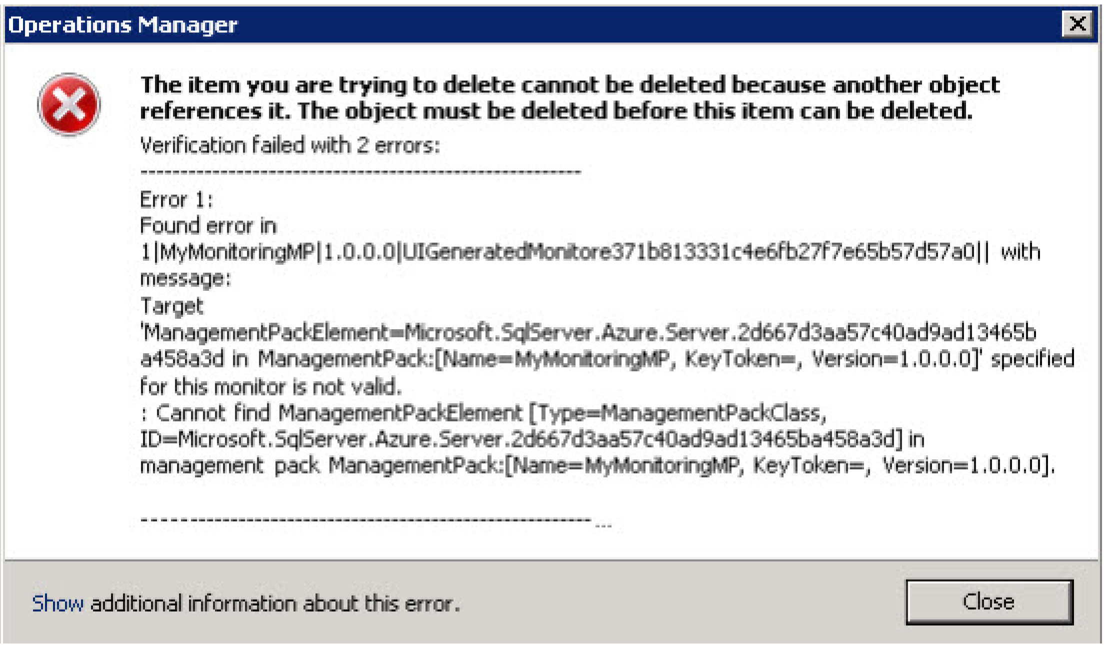
View Path	Description
Servers\Server State	Displays a list of monitored cloud services and their current states. Double-click the health state icon for a database to launch a Health Explorer window for that databases to locate monitors that affect the health state of the server and investigate any issue. The **Detail View** pane displays properties of the database selected above.
Servers\Servers Diagrams	Displays a structured picture of all monitored cloud services with hosted databases. Expand the required cloud service node to drill down into hosted objects.

Appendix: Known Issues and Troubleshooting

“The item you are trying to delete cannot be deleted because another object references it” error appears when trying to remove the template

Issue: When you are trying to remove a monitoring template, the following message is displayed:

“The item you are trying to delete cannot be deleted because another object references it…”



The item you are trying to delete cannot be deleted because another object references it. The object must be deleted before this item can be deleted.

```
Verification failed with 2 errors:
-------------------------------------------------------
Error 1:
Found error in
1|MyMonitoringMP|1.0.0.0|UIGeneratedMonitore371b813331c4e6fb27f7e65b57d57a0|| with
message:
Target
'ManagementPackElement=Microsoft.SqlServer.Azure.Server.2d667d3aa57c40ad9ad13465b
a458a3d in ManagementPack:[Name=MyMonitoringMP, KeyToken=, Version=1.0.0.0]' specified
for this monitor is not valid.
: Cannot find ManagementPackElement [Type=ManagementPackClass,
ID=Microsoft.SqlServer.Azure.Server.2d667d3aa57c40ad9ad13465ba458a3d] in
management pack ManagementPack:[Name=MyMonitoringMP, KeyToken=, Version=1.0.0.0].

-------------------------------------------------------...
```

Show additional information about this error.

This is a known SCOM issue. Since SCOM does not support cascade delete for templates, you must manually remove all monitors targeting the server defined by the template, before you will be able to remove the template itself.

Resolution: In SCOM Console, navigate to Authoring | Management Pack Objects | Monitors, scope the list to the server, defined by the template you want to delete and remove all custom monitors.

Some Elastic Pools may not be discovered

Issue: Elastic Pools that do not contain any databases are not discovered.

Resolution: No resolution.

Error messages are received when Azure SQL Server is discovered by means of several templates simultaneously

Issue: If several Azure SQL Database templates with different user rights are used simultaneously to discover same Azure SQL Servers, error events (ID 6302) appear in the Operations Manager Event Viewer.

Resolution: Each Azure SQL Server must be discovered by means of a single template only.

Rules and monitors may provide incorrect data if default interval override values are changed

Issue: If the value of Interval (seconds) overridable parameter is set lower than the default value, rules and monitors may provide incorrect data.

Resolution: The Interval (seconds) overridable parameter must be set to no lower than the default value.

Server exclude list option may work incorrectly

Issue: Server exclude list may behave incorrectly: the set masks may disappear from the list, and some performance may be received from the servers that should have been excluded.

Resolution: No resolution.

Some performance collection rules fail to collect data when REST+T-SQL is enabled

Issue: Some performance collection rules may not work due to lack of required T-SQL permissions.

Resolution: Run T-SQL queries specified in the Configuring Azure REST API Monitoring section.

"Use T-SQL for monitoring" checkbox configuration cannot be saved

Issue: After creating Azure SQL Database Monitoring template using "Azure Service Principal" Authentication Mode and "Use Existing Run As Profile" SPN Configuration, "Use T-SQL for monitoring" checkbox remains enabled regardless of the user choice.

Resolution: No resolution.

The monitored objects become unavailable if management server is changed in the resource pool

Issue: The monitored objects become unavailable (turn grey) in the Operations Manager if management server is changed in the resource pool. An alert with the following description is displayed in the SCOM log: "The pool member no longer owns any managed objects assigned to the pool because half or fewer members of the pool have acknowledged the most recent lease request. The pool member has unloaded the workflows for managed objects it previously owned."

Resolution: Wait until the objects are processed on the new management server.

Azure Portal may stop retrieving results in responses to Azure REST API requests from some performance rules

Issue: In case of a great number of databases (about 1000 databases), Azure Portal may stop retrieving results in responses to Azure REST API requests from some performance rules. The errors in such responses are as follows:

HTTP/1.1 504 Gateway Timeout

.....

Connection: close

Content-Length: 141

{"error":{"code":"GatewayTimeout","message":"The gateway did not receive a response from 'Microsoft.Sql' within the specified time period."}}

Resolution: No resolution.

SQL connection to the Azure SQL Databases may fail if the number of databases is great

Issue: If the number of databases is great (over 2000 databases), SQL connection to the Azure SQL Databases may fail with the following exceptions:

- A connection was successfully established with the server, but then an error occurred during the pre-login handshake.
- Connection Timeout Expired. The timeout period elapsed while attempting to consume the pre-login handshake acknowledgment. This could be because the pre-login handshake failed or the server was unable to respond back in time.
- A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible. Verify that the instance name is correct and the SQL Server is configured to allow remote connections.

As a result, Database Connection Availability monitor changes its state from "Healthy" to "Warning". It may also affect workflows with T-SQL query datasources due to connection loss.

Resolution: No resolution.

Limitations of space monitoring in Hyperscale service tier

Issue: Some space monitoring workflows may not collect data correctly for databases of the Hyperscale service tier:

- Rules:
    - Free Space (MB)
    - Free Space Percentage
    - Used Space Percentage
    - Total Space Quota (MB)
- Unit monitors:
    - Database Free Space

Resolution: As a temporary solution, you can turn off these performance rules and monitors. This issue will be fixed in one of the next updates of the management pack.

Appendix: Disabled Monitors

Most of the database performance monitors are disabled by default because the appropriate thresholds need to be determined based on the database applications being monitored. If this functionality is required for proper monitoring of the database applications, perform the following:

1. Determine the correct threshold values based on the expected usage patterns or observed resource consumption.
2. Override one or more of these monitors to adjust the thresholds and enable them.

Disabled monitors are as follows:

- Connections
    - Count of Failed Connection
    - Count of connections blocked by the Firewall
- Sessions
    - Sessions Count
    - Sessions Average Memory
    - Sessions Rows Returned
    - Sessions Total CPU Time
    - Sessions Total I/O
    - Sessions Total Memory
- Transactions
    - Transaction Locks Count
    - Transaction Log Space Used
    - Transaction Execution Time
- Geo-Replication Link State