

How to Setup the Azure SQL Managed Instance MP

Friday, January 1, 2021 11:50 AM

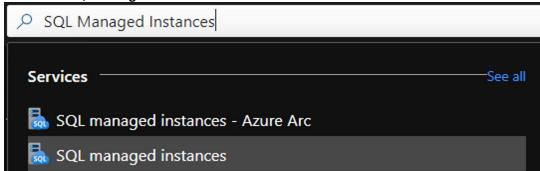
In this guide we will walk through setting up an Azure SQL Managed Instance and work on Configuring it to work with the SCOM MP for Azure SQL Managed Instance.

Setup Azure SQL Managed Instance (Azure Portal)

Create the SQL Managed Instance

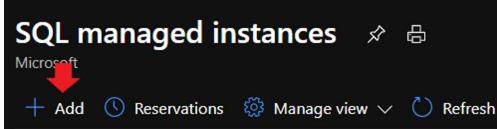
1. Navigate to <https://ms.portal.azure.com>

- a. Search for **SQL Managed Instances**



The screenshot shows the Azure portal search bar with the text 'SQL Managed Instances' entered. Below the search bar, a list of services is shown, with 'SQL managed instances - Azure Arc' and 'SQL managed instances' highlighted in grey.

2. Click **Add SQL Managed Instance**



The screenshot shows the 'SQL managed instances' blade. A red arrow points to the '+ Add' button. Below the button are 'Reservations', 'Manage view', and 'Refresh' buttons.

- a. Fill out the Form for Managed Instance Details (you can leave the Defaults for Networking, Additional Settings, and Tags)

- If you are creating this for a Test, make sure to modify the default **vCores** and **Storage in GB** by going to **Configure Managed Instance** (doing this will keep you bill as low as possible)

Compute + storage * ⓘ

General Purpose

Gen5, 4 vCores, 32 GB storage, Locally-redundant backup storage

[Configure Managed Instance](#)

- For this example, we will be using the following credentials for the Managed Instances admin login:

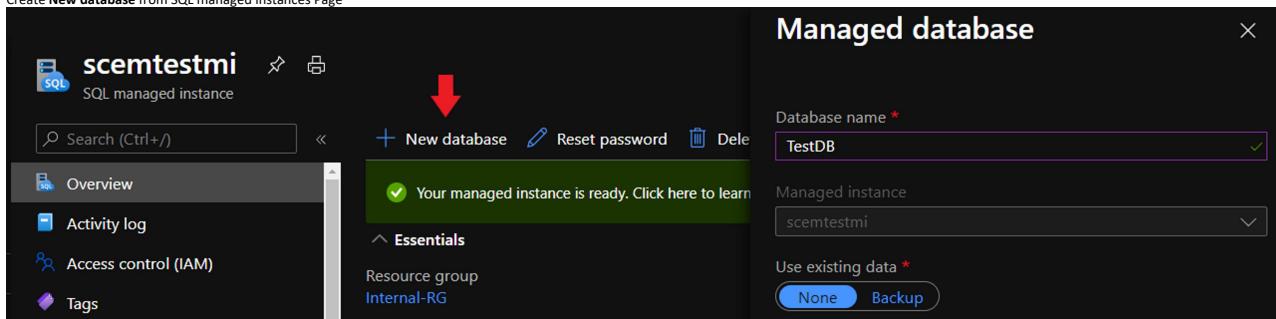
Managed Instance admin login	mi_admin
Password	ThisIsTheManagedInstancePa\$\$word1

- b. Click the **Review + Create** Button

Review + create

- After clicking on the Button, **please allow up to (2-10) Hours for the Managed Instance to be deployed**.

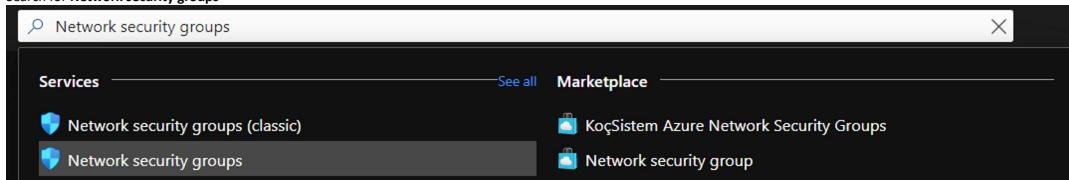
3. Create **New database** from SQL managed instances Page



The screenshot shows the 'Managed database' blade for the 'scemtestmi' managed instance. A red arrow points to the '+ New database' button. The 'TestDB' database is selected. The blade also shows the 'Your managed instance is ready. Click here to learn' message and the 'Essentials' section with 'Resource group: Internal-RG'.

Configure the Network Security Group

4. Search for **Network security groups**



The screenshot shows the 'Network security groups' blade. A red arrow points to the '+ New database' button. The 'Network security groups' option is selected. The blade also shows 'KoçSistem Azure Network Security Groups' and 'Network security group'.

5. Locate the Network Security Group:



The screenshot shows the 'nsq-scemtestmi' blade. A red arrow points to the 'Add' button in the 'Inbound security rules' section. The blade also shows 'Internal-RG' and 'East US'.

- a. Select Inbound security rules, Click + Add
 - Click on **Destination**, Select **VirtualNetwork**
 - Click on **Destination port ranges**, Type: **3342**
 - Click on **Protocol**, Click **TCP**
 - Click on **Action**, Select **Allow**
 - Click on **Priority**, Type: **105**
 - Click on **Name**, Type: **ManagedInstance_3342**
 - Click on **Add** to complete

6. Go to Security - Networking, Turn on **Public endpoint** for Managed Instance



The screenshot shows the 'Networking' blade for the 'scemtestmi' managed instance. The 'Public endpoint' setting is turned on.

Install / Configure the Management Pack

There are 2 options when creating the Azure SQL Managed Instance Monitor, **Private endpoint** and **Public endpoint**; We will be using the public option with this example. We also are using the Automatic Monitor Template.

1. Download the MP:
[Microsoft System Center Management Pack for Microsoft Azure SQL Managed Instance](#)
 - a. Be sure to download the following MSI:
[Microsoft.Azure.ManagedInstance.ManagementPack.msi](#)

2. Import the Management Pack:

3. After successful import, open the SCOM Console and go to:
Authoring -> Management Pack Templates

- a. Click on the **Add Monitoring Wizard** button on the right side of console.

- b. Select **Azure SQL MI - Automatic** and click **Next >**

Select the monitoring type

NET Application Performance Monitoring
 Azure SQL MI - Automatic
 Azure SQL MI - Manual
 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

Description:
 This template allows you to configure automatic discovery and monitoring of Microsoft Azure SQL Managed Instances.

- c. **Name:** Azure SQL MI Monitor
 i. Create a New Management Pack, name it: **AzureSQL MI Custom**
 ii. Click **Next >**

d. Leave Defaults on Azure Endpoints Page, Click **Next >**

e. Verify that **Auto-Create SPN** is selected, Click **Next >**

Select the Azure SPN Configuration option

Azure Endpoints
SPN Configuration
 Auto-Create SPN Status
 Subscription Permissions
 SQL Connection Settings
 Configure Instances Filtering

SPN is used to discover Managed Instances using Azure REST API.

f. **Sign in with your Microsoft Azure Account**

g. When you successfully authenticate, you will see the SPN Status Page Populated, Click **Next >**

Add Monitoring Wizard

Auto-Create SPN Status

Monitoring Type

General Properties

Azure Endpoints

SPN Configuration

Auto-Create SPN Status

Subscription Permissions

SQL Connection Settings

Configure Instances Filtering

Summary

Help

Auto-create SPN Status

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

Run As Account Name: **Azure_SQL_ManagedInstance_RunAsAccount_2021-01-01**

Tenant ID: **7298bf-86f1-41af-91ab-2d7cd011db47**

Application ID: **f74bad89-8e16-4aa6-8462-41952da597c**

Client Secret: **X9m7TQPi+0QZUmBFR4lvkyYk.jCu73esjXRMoc3H98=**

Application Name: **Azure_SQL_ManagedInstance_App_448236ef-0f0a-4e42-aacc-44ad3c5f1e67**

Azure SQL Managed Instance

< Previous Next > Create Cancel

h. Select the Subscription for the SPN Configuration to Apply, Click **Next >**

i. Select SQL Credentials (SQL) and Click **New...**

i. Enter the Credentials for the Managed Instance admin login:

Account name	SQL Managed Instance Credentials
Login	mi_admin
Password	ThisIsTheManagedInstancePa\$\$word1

j. Verify your information is correct, Click **Next >**

SQL Connection Settings

Monitoring Type
General Properties
Azure Endpoints
SPN Configuration
Auto-Create SPN Status
Subscription Permissions
SQL Connection Settings
Configure Instances Filtering
Summary

SQL Connection Settings

Configure settings for SQL connections to discovered Managed Instances.

Authentication

Select authentication method for SQL connections

Azure Active Directory (AAD)

SQL Credentials (SQL)

Existing SQL Run As Account:

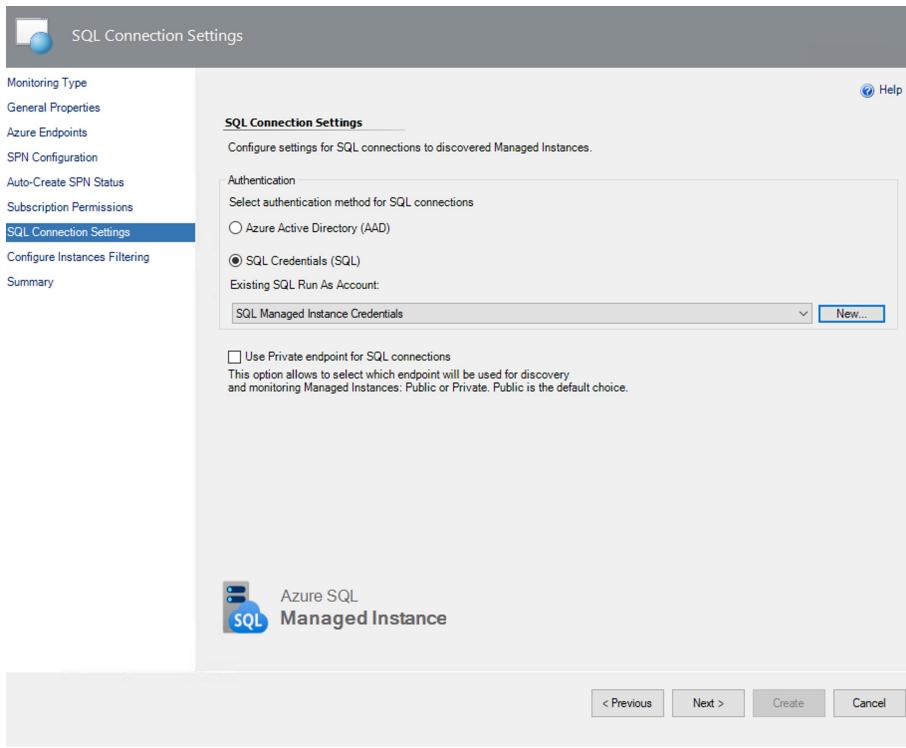
SQL Managed Instance Credentials

Use Private endpoint for SQL connections

This option allows to select which endpoint will be used for discovery and monitoring Managed Instances: Public or Private. Public is the default choice.

Azure SQL
Managed Instance

< Previous



- k. Click on **Include** and in the **Enter a filter mask box** type: * and click **Add**, Click **Next >**

Configure Instances Filtering

Monitoring Type
General Properties
Azure Endpoints
SPN Configuration
Auto-Create SPN Status
Subscription Permissions
Configure Instances Filtering
Summary

Configure Instances Filtering

This page allow to configure a list of filter masks, which will be used for filtering Azure SQL Managed Instance Names. Those Instances, which names match the filter masks will be included or excluded during the monitoring depending on the selected filtering mode.

Instances filtering mode

Exclude

Include

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

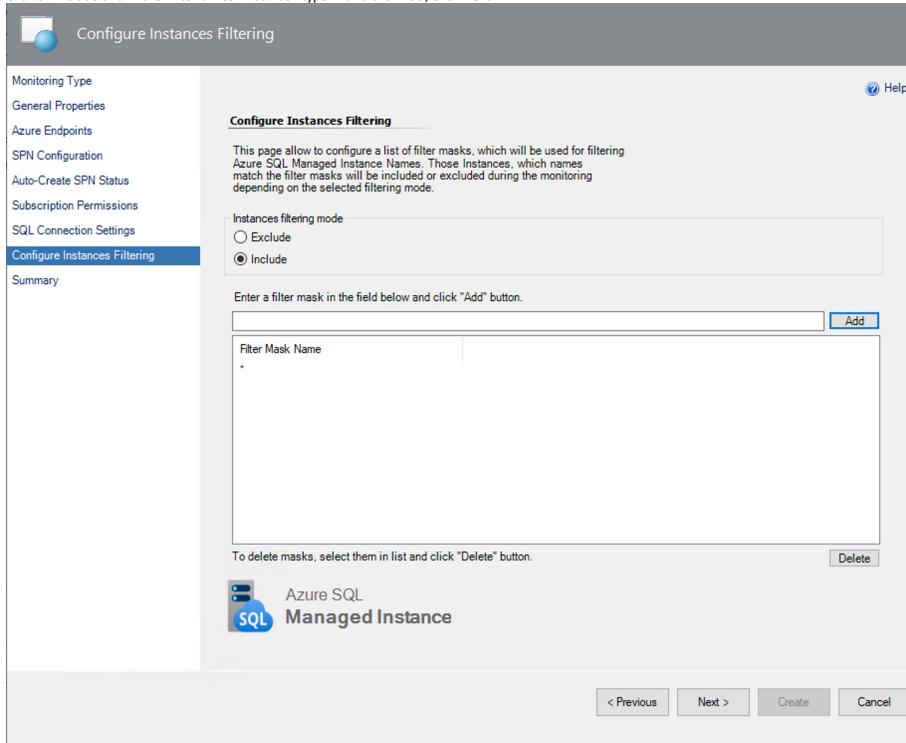
Filter Mask Name

*

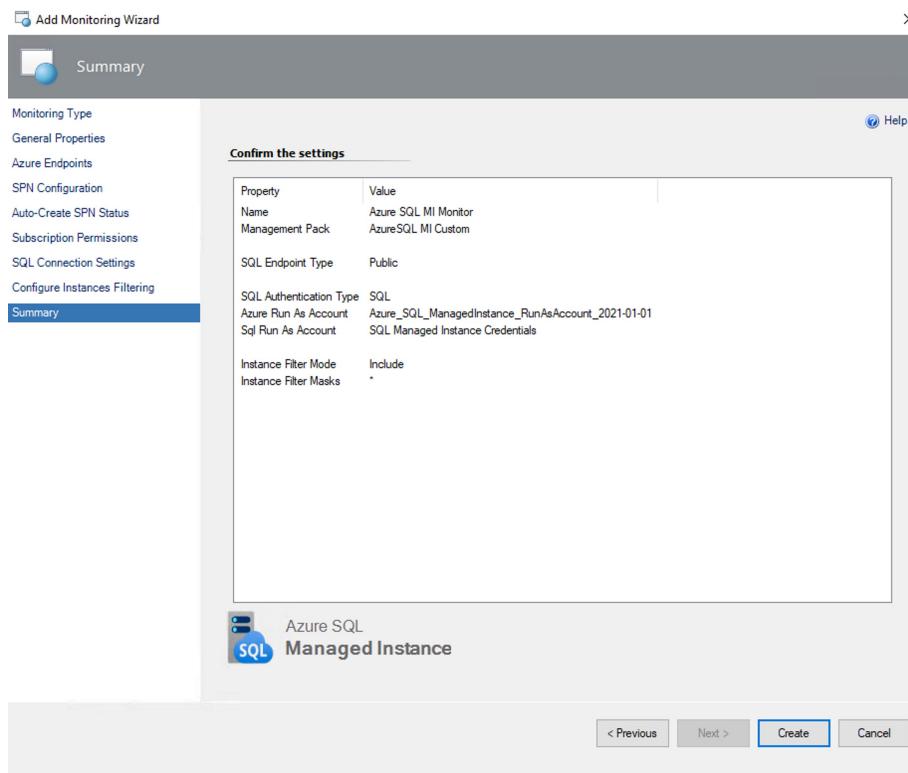
To delete masks, select them in list and click "Delete" button.

Azure SQL
Managed Instance

< Previous



- l. Confirm information is correct, Click **Create**



File Edit View Go Tasks Tools Help

Search Scope Find Tasks

Authoring

- Management Pack Templates
 - .NET Application Performance Monitoring
 - Azure SQL MI – Automatic
 - Azure SQL MI – Manual

Add Monitoring Wizard...

New Distributed Application...

New Group...

Monitoring

Authoring

Reporting

Administration

My Workspace

Azure SQL MI – Automatic (1)

Name	Management Pack	Created
Azure SQL MI Monitor	AzureSQL MI Custom	1/1/2021 1:20:52 PM

Details:

Administration -> Run As Configuration -> Accounts -> SQL Managed Instance Credentials

File Edit View Go Tasks Tools Help

Search Scope Find Tasks

Administration

- Product Connectors
 - Internal Connectors
 - Resource Pools
- Run As Configuration
 - Accounts
 - Profiles
 - UNIX/Linux Accounts

Discovery Wizard...

Monitoring

Authoring

Reporting

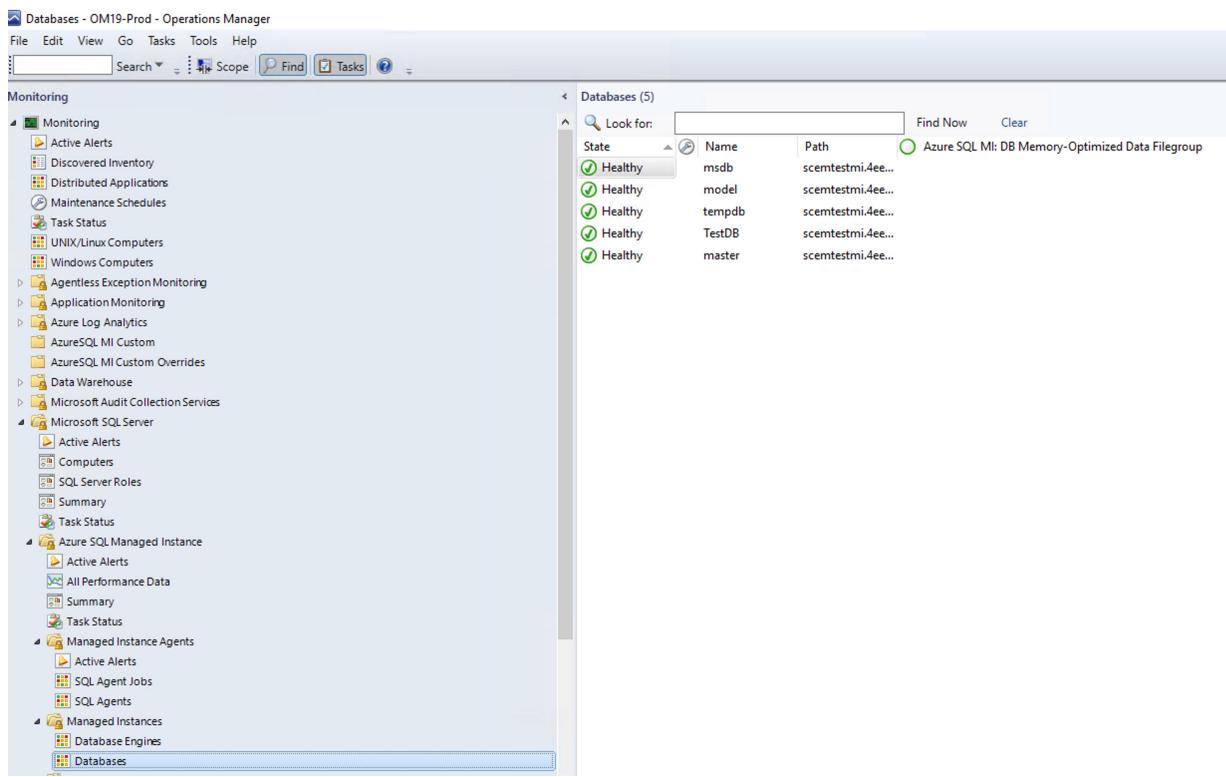
Administration

My Workspace

Accounts (8)

Name	Description
ADsvc_scm19_oma	This is the user account under which all rules run by default on the agent.
Local System Action Account	Built in SYSTEM account to be used as an action account
Azure_SQL_ManagedInstance_RunAsA...	Azure_SQL_ManagedInstance_RunAsAccount_2021-01-01
SQL Managed Instance Credentials	SQL Managed Instance Credentials
Type: Action Account (2)	
Type: Basic Authentication (2)	
Type: Windows (4)	
Data Warehouse Action Account	Data Warehouse Action Account
Data Warehouse Report Deployment ...	Data Warehouse Report Deployment Account
Local System Windows Account	Built in SYSTEM account
Network Service Windows Account	Built in Network service account

Ready



Azure SQL Managed Instance Discoveries

To See all the Object Discoveries Associated with Azure SQL MI you will need to perform the following steps.

BE MINDFUL OF THE FOLLOWING:

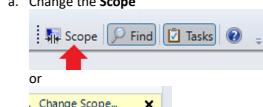
Any changes you make to intervals may result in inconsistent data.

The Intervals should remain untouched in a Production environment to ensure data integrity.

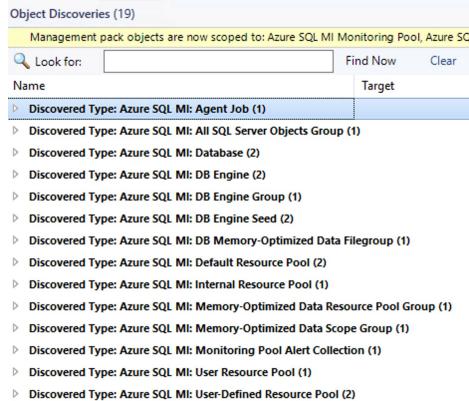
You may kick start the Discoveries by changing the intervals, but after the discovery happens,

Revert the values back to the defaults.

1. Go to Authoring -> Management Pack Objects -> Object Discoveries



- a. Change the Scope



Quick Discovery Script

Run the below Powershell from a Management Server to quickly discover Azure SQL MI objects

```

try
{
    $ManagementServer = $env:COMPUTERNAME
    $Task = Get-SCOMTask -Name Microsoft.SystemCenter.TriggerOnDemandDiscovery
    $Discoveries = Get-SCOMDiscovery -DisplayName 'Azure SQL MI:*
    'Starting Discoveries (Count: ' + $Discoveries.Count + ')' | Write-Host
    $i = 0
    foreach ($Discovery in $Discoveries)
    {
        $i = $i
        $i++
        ('+' + $i + '/' + $Discoveries.Count + ')' | Write-Host
        $output = @()
        $Override = @{}{DiscoveryId = $Discovery.Id.ToString() || TargetInstanceId = $Discovery.Target.Id.ToString() || }
        $Instance = Get-SCOMClass -Name Microsoft.SystemCenter.ManagementServer | Get-SCOMClassInstance | where { $_.Displayname -like "$ManagementServer*" }
    }
}

```

```

    $output += (Start-SCOMTask -Task $Task -Instance $Instance -Override $Override | Select-Object Status, @{Name = "Discovery" Expression = { $Discovery } }, StartTime, TimeScheduled, TimeFinished, Output) | Out-String -Width 4096
    $output
}
}
catch
{
    Write-Warning $_
    Write-Host "Unable to trigger the discovery" -ForegroundColor Red
}

```

Least Privilege Configuration for Monitoring

1. Open - SQL Server Management Studio
 - a. Server name: `scmtestmi.public.4ee60cf6b3ee.database.windows.net,3342`
 - Username: `mi_admin`
 - Password: `ThisIsTheManagedInstancePa$$word1`

```

--First script that:
-- Grants server-level permissions to the monitoring account.
-- Creates a user and role in the master, msdb and model databases and grants the required permissions to it.
-- Creates a user and role in all user databases.
--Don't forget to replace "LowPrivPassword1234!" with your value in @createLoginCommand variable.
USE [master];
SET NOCOUNT ON
DECLARE @accountname SYSNAME = 'MILowPriv';
CREATE SERVER ROLE [MILowPriv_role];
GRANT VIEW ANY DEFINITION TO [MILowPriv_role];
GRANT VIEW ANY DATABASE TO [MILowPriv_role];
GRANT ALTER ANY DATABASE TO [MILowPriv_role];
GRANT VIEW SERVER STATE TO [MILowPriv_role];
DECLARE @createLoginCommand nvarchar(200);
SET @createLoginCommand = 'CREATE LOGIN ' + QUOTENAME(@accountname) + ' WITH PASSWORD=N''LowPrivPassword1234!', DEFAULT_DATABASE=[master],';
EXEC (@createLoginCommand);
EXEC sp_addsrvrolemember @loginname = @accountname, @rolename = 'MILowPriv_role';
DECLARE @createDatabaseUserAndRole nvarchar(max);
SET @createDatabaseUserAndRole = '';
SELECT @createDatabaseUserAndRole = @createDatabaseUserAndRole + 'USE ' +
QUOTENAME(db.name) + '';
CREATE USER ' + QUOTENAME(@accountname) + ' FOR LOGIN ' + QUOTENAME(@accountname) + '';
CREATE ROLE [MILowPriv_role];
EXEC sp_addrolemember @rolename = "MILowPriv_role", @membername = ' + QUOTENAME(@accountname) + "';
FROM sys.databases db
WHERE db.database_id <> 2
AND db.user_access = 0
AND db.state = 0
AND db.is_read_only = 0;
EXEC (@createDatabaseUserAndRole);
GO
USE [master];
GRANT EXECUTE ON xp_readererrorlog TO [MILowPriv_role];
GRANT EXECUTE ON xp_instance_regread TO [MILowPriv_role];
GRANT EXECUTE ON xp_sqlagent_enum_jobs TO [MILowPriv_role];
GRANT EXECUTE ON sp_enumerrorlogs TO [MILowPriv_role];
USE [msdb];
GRANT EXECUTE ON sp_help_job TO [MILowPriv_role];
GRANT EXECUTE ON sp_help_jobactivity TO [MILowPriv_role];
GRANT SELECT ON sysjobschedules TO [MILowPriv_role];
GRANT SELECT ON backupset TO [MILowPriv_role];
EXEC sp_addrolemember @rolename='db_datareader', @membername='MILowPriv_role';
EXEC sp_addrolemember @rolename='db_owner', @membername='MILowPriv_role';
EXEC sp_addrolemember @rolename='SQLAgentReaderRole',
@membername='MILowPriv_role';

--Second script that adds MILowPriv user to db_owner role for master, msdb, model, and all user databases.
--This is only to enable running DBCC checks right on SCOM (Check Catalog, Check Database, Check Disk).
--If you don't need this functionality, don't run this script.
DECLARE @accountname sysname = 'MILowPriv';
DECLARE @createDatabaseUserAndRole nvarchar(max);
SET @createDatabaseUserAndRole = '';
SELECT @createDatabaseUserAndRole = @createDatabaseUserAndRole + 'USE ' +
QUOTENAME(db.name) + '';
EXEC sp_addrolemember @rolename = "db_owner", @membername = ' +
QUOTENAME(@accountname) + "';
FROM sys.databases db
WHERE db.database_id <> 2
AND db.user_access = 0
AND db.state = 0
AND db.is_read_only = 0;
EXEC (@createDatabaseUserAndRole);
GO

```

2. Go to SCOM Console, Administration -> Run As Configuration -> SQL Managed Instance Credentials, Right Click go to Properties

The screenshot shows the 'Administration' section of the Operations Manager interface. The left pane displays a tree view of management components, including Agentless Managed, Management Servers, Pending Management, UNIX/Linux Computers, Management Packs, Network Management, Notifications, Operations Manager Products, Run As Configuration, Security, and Discovery Wizard. The 'Run As Configuration' node is expanded, showing 'Accounts', 'Profiles', and 'UNIX/Linux Accounts'. The 'Accounts' node is selected, and its properties dialog is open on the right. The dialog title is 'Run As Account Properties - SQL Managed Instance Credentials'. It has tabs for 'General Properties', 'Credentials', and 'Distribution'. The 'General Properties' tab is selected, showing the 'Specify general properties for the Run As account' section. It includes fields for 'Run As account type' (set to 'Basic Authentication'), 'Display name' (set to 'SQL Managed Instance Credentials'), and 'Description (optional)' (set to 'SQL Managed Instance Credentials'). At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

a. Go to Credentials Tab, set the following:

Account Name	MLowPriv
Password	LowPrivPassword1234!

b. Click OK