How to Setup the Azure SQL Managed Instance MP

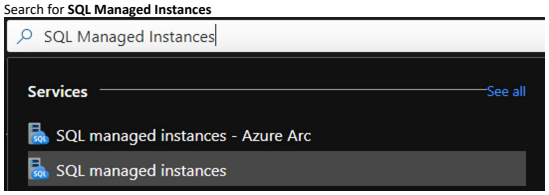
Friday, January 1, 2021 11:50 AM

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.

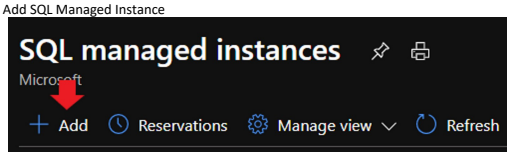
Setup Azure SQL Managed Instance (Azure Portal)

Create the SQL Managed Instance

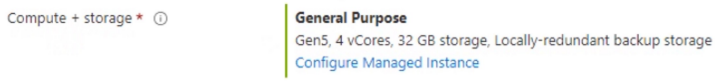
1. Navigate to https://ms.portal.azure.com
   a. Search for **SQL Managed Instances**



2. Click Add SQL Managed Instance



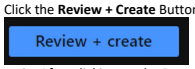
   a. Fill out the Form for Managed Instance Details (you can leave the Defaults for Networking, Additional Settings, and Tags)
      i. If you are creating this for a Test, make sure to modify the default ***vCores*** and ***Storage in GB*** by going to **Configure Managed Instance** (doing this will keep you bill as low as possible)



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

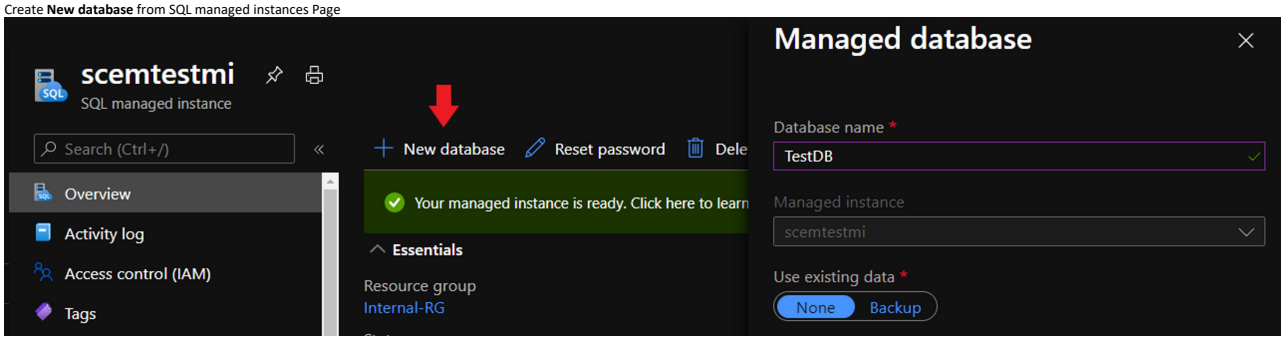
Managed Instance admin login	**mi_admin**
Password	**ThisIsTheManagedInstancePa$$word1**

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



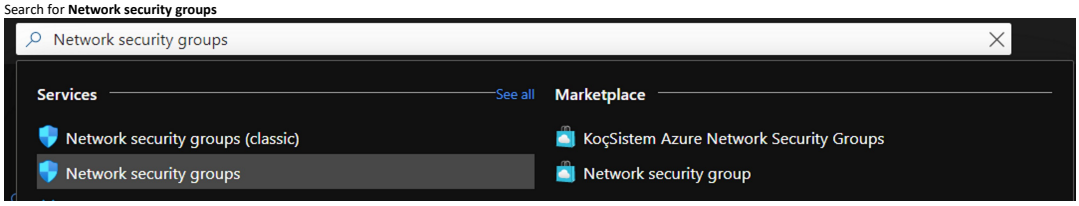
      i. After clicking on the Button, **please allow up to (2-10) Hours for the Managed Instance to be deployed**.

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



Configure the Network Security Group

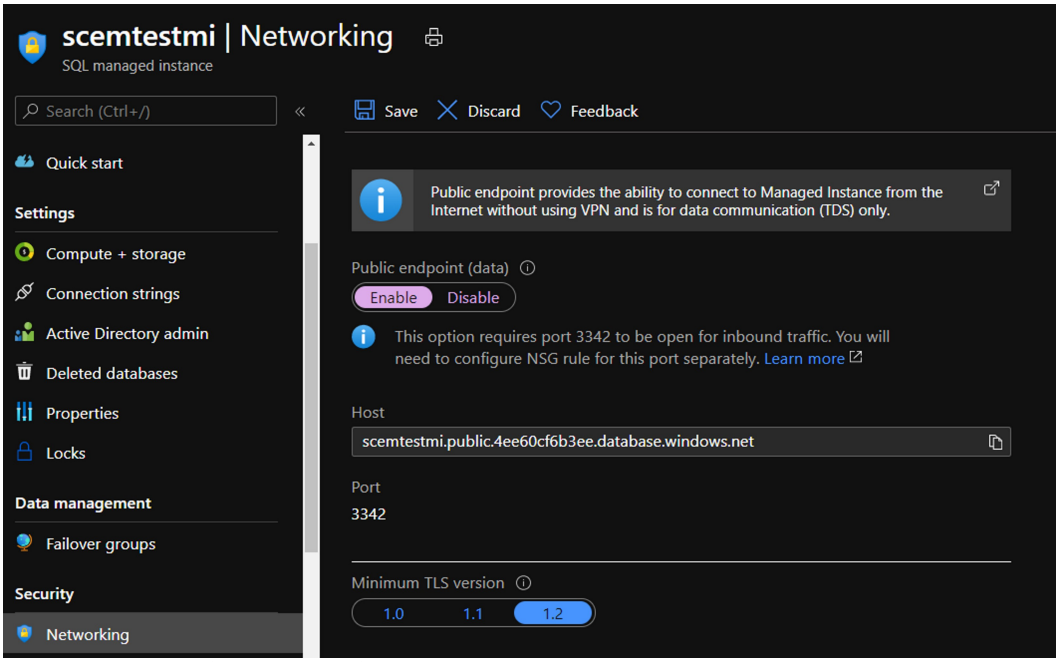
4. Search for **Network security groups**



5. Locate the Network Security Group:

   a. Select Inbound security rules, Click **+ Add**
      i. Click on **Destination**, Select **VirtualNetwork**
      ii. Click on **Destination port ranges**, Type: **3342**
      iii. Click on **Protocol**, Click **TCP**
      iv. Click on **Action**, Select **Allow**
      v. Click on **Priority**, Type: **105**
      vi. Click on **Name,** Type: **ManagedInstance_3342**
      vii. Click on **Add** to complete

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



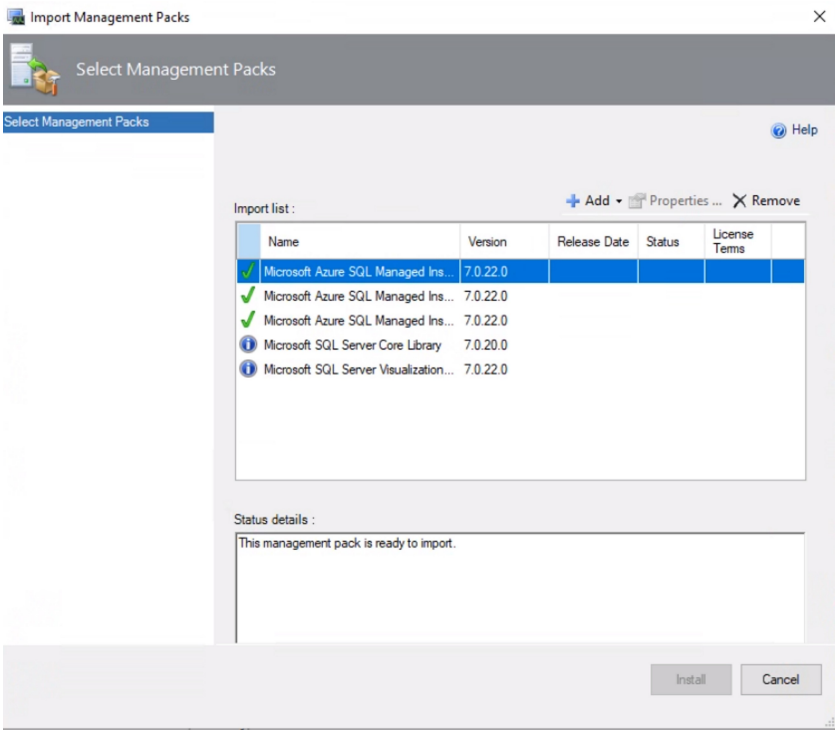
We will need this Hostname to configure Least Privilege Configuration.

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.

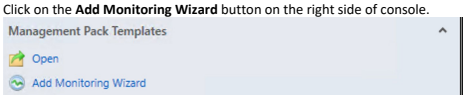
1. Download the MP:
   [Microsoft System Center Management Pack for Microsoft Azure SQL Managed Instance]
   a. Be sure to download the following MSI:
   Microsoft.Azure.ManagedInstance.ManagementPack.msi

2. Import the Management Pack:

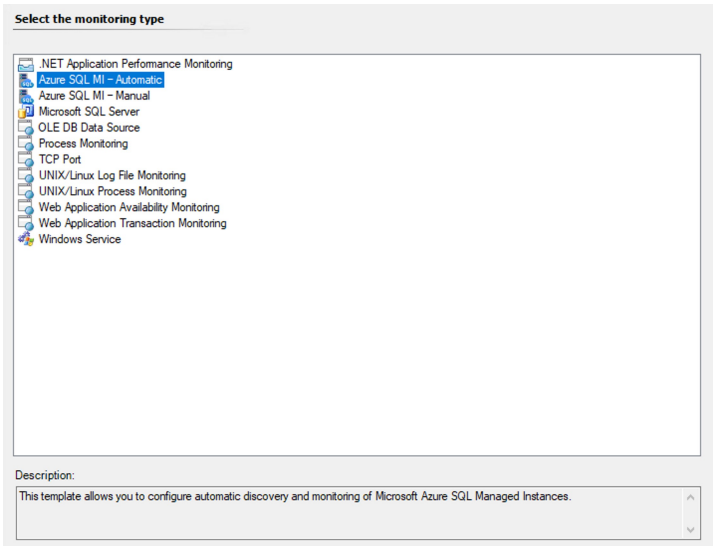


3. After successful import, open the SCOM Console and go to:
   Authoring -> Management Pack Templates

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



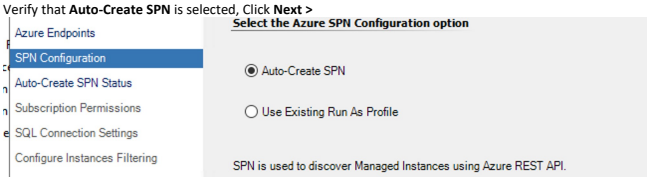
   b. Select **Azure SQL MI - Automatic** and click **Next >**



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

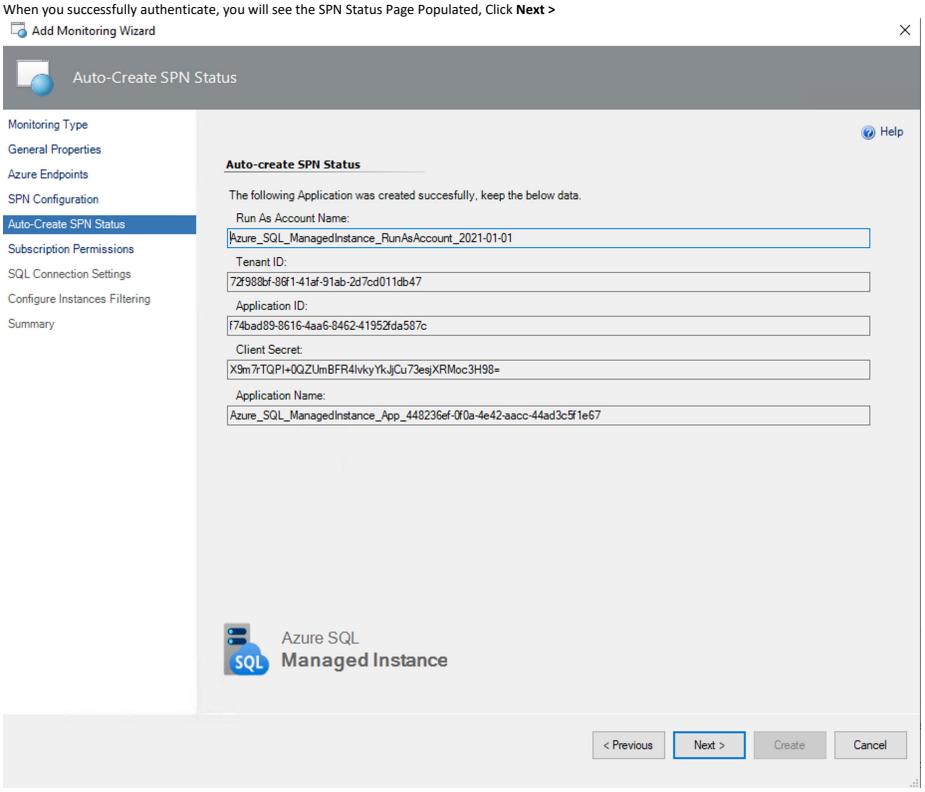
d. Leave Defaults on Azure Endpoints Page, Click **Next >**

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



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

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

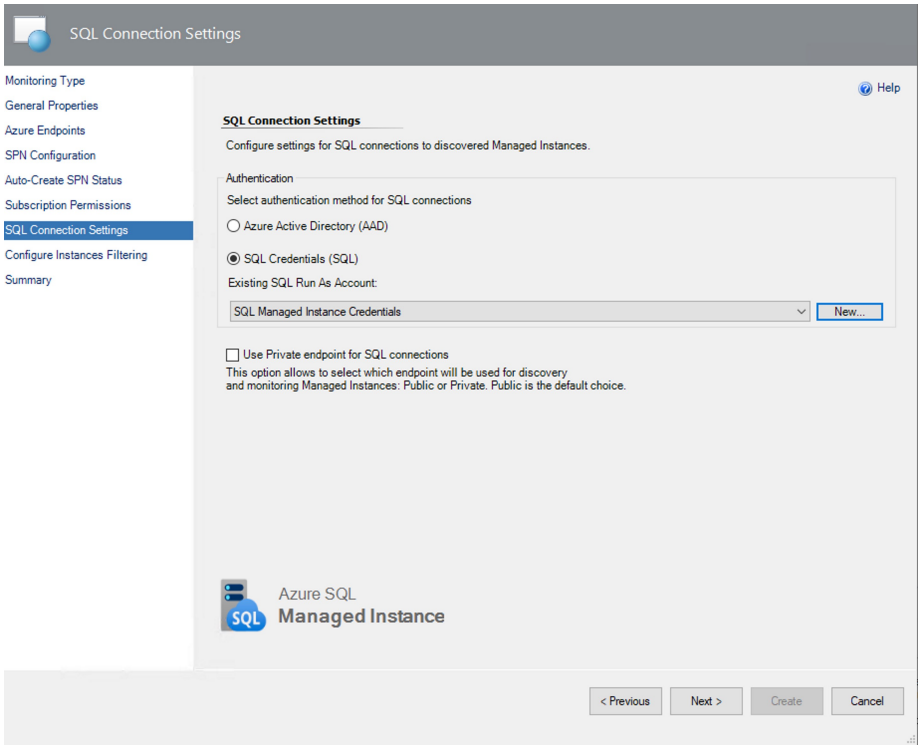


h. Select the Subscription for the SPN Configuration to Apply, Click Next >

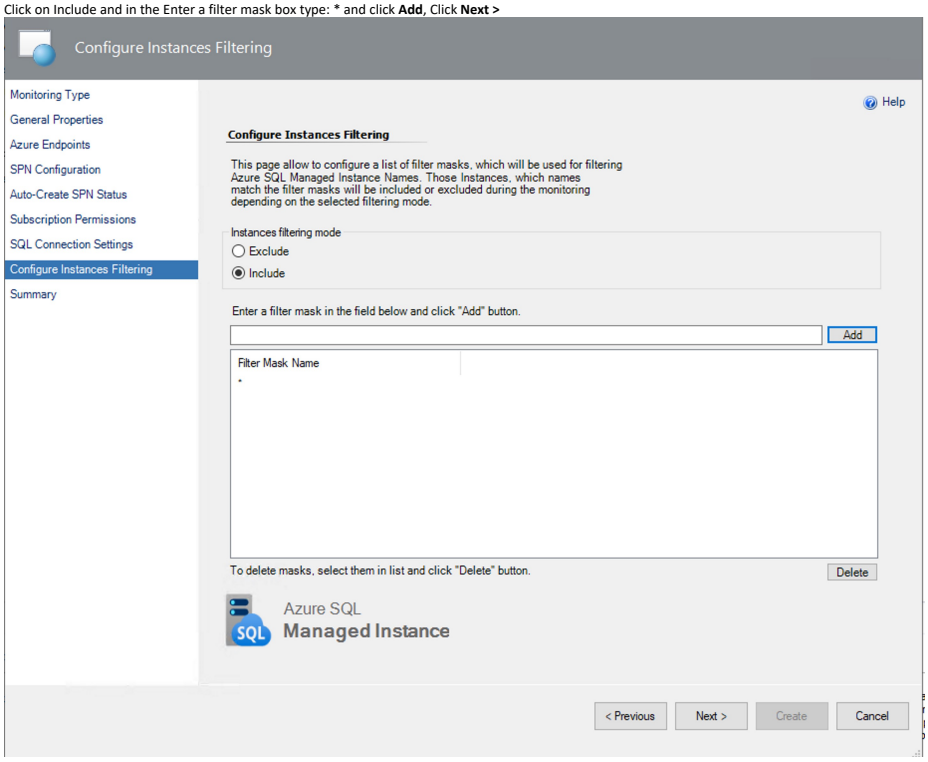
i. Select SQL Credentials (SQL) and Click **New…**
   i. Enter the Credentials for the Managed Instance admin login:

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

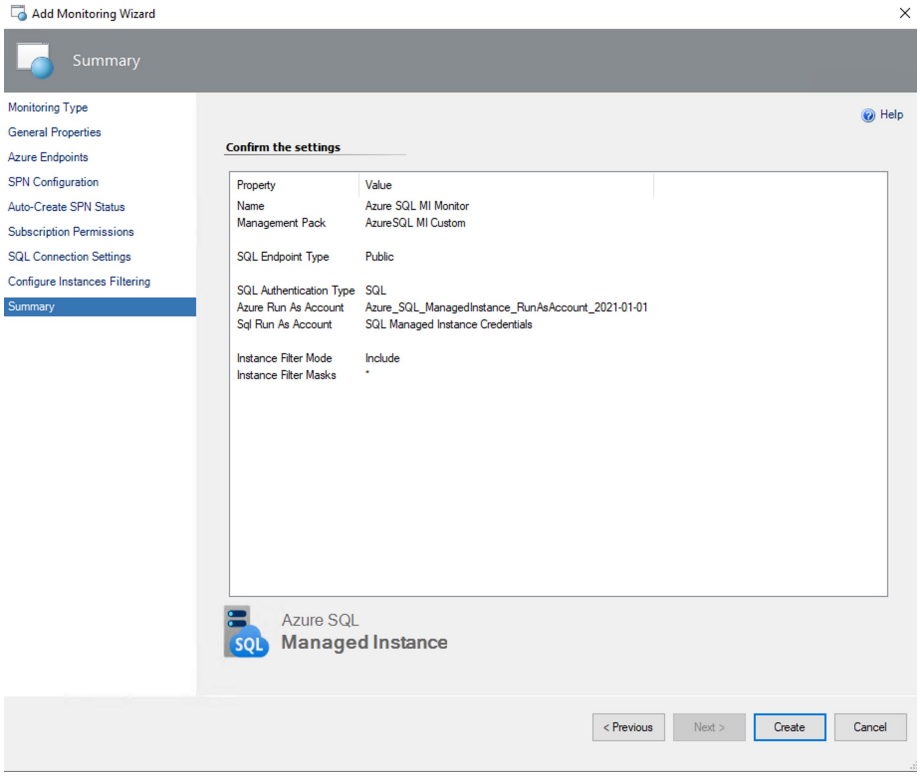
j. Verify your information is correct, Click **Next >**

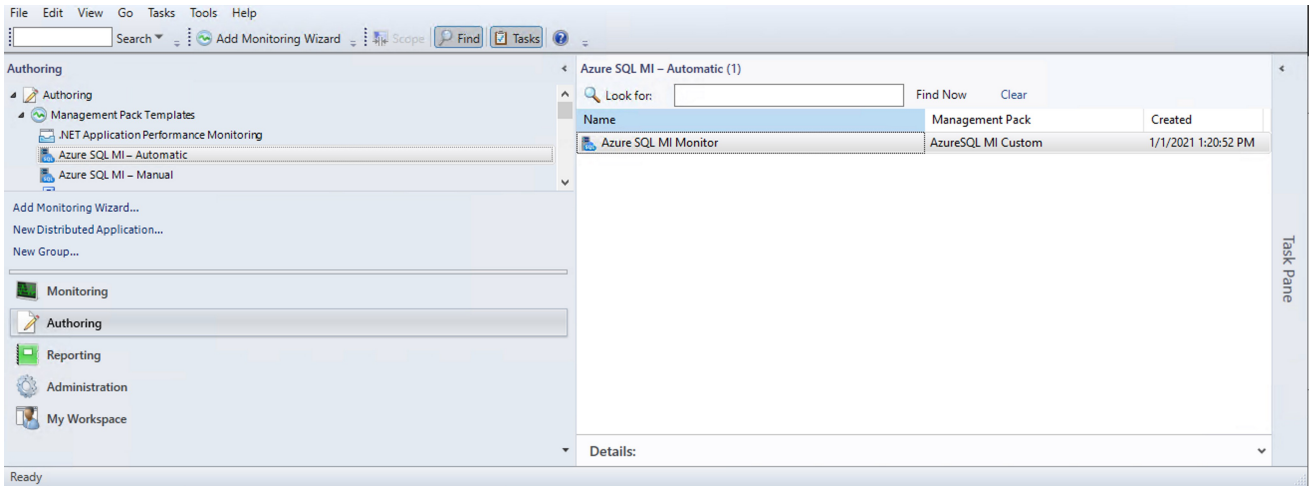


k. Click on Include and in the Enter a filter mask box type: * and click **Add**, Click **Next >**

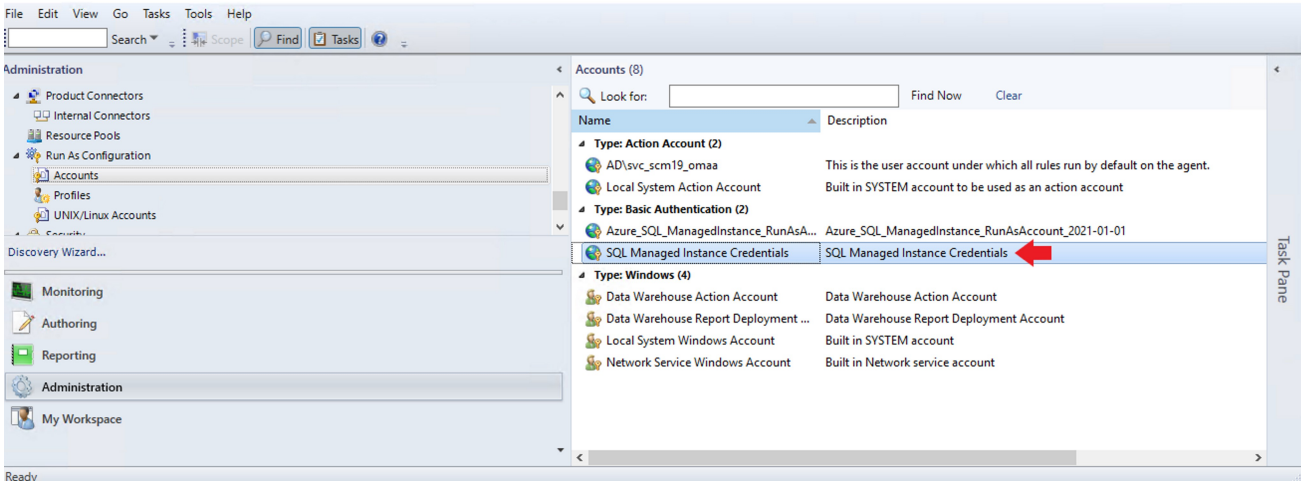


l. Confirm information is correct, Click **Create**

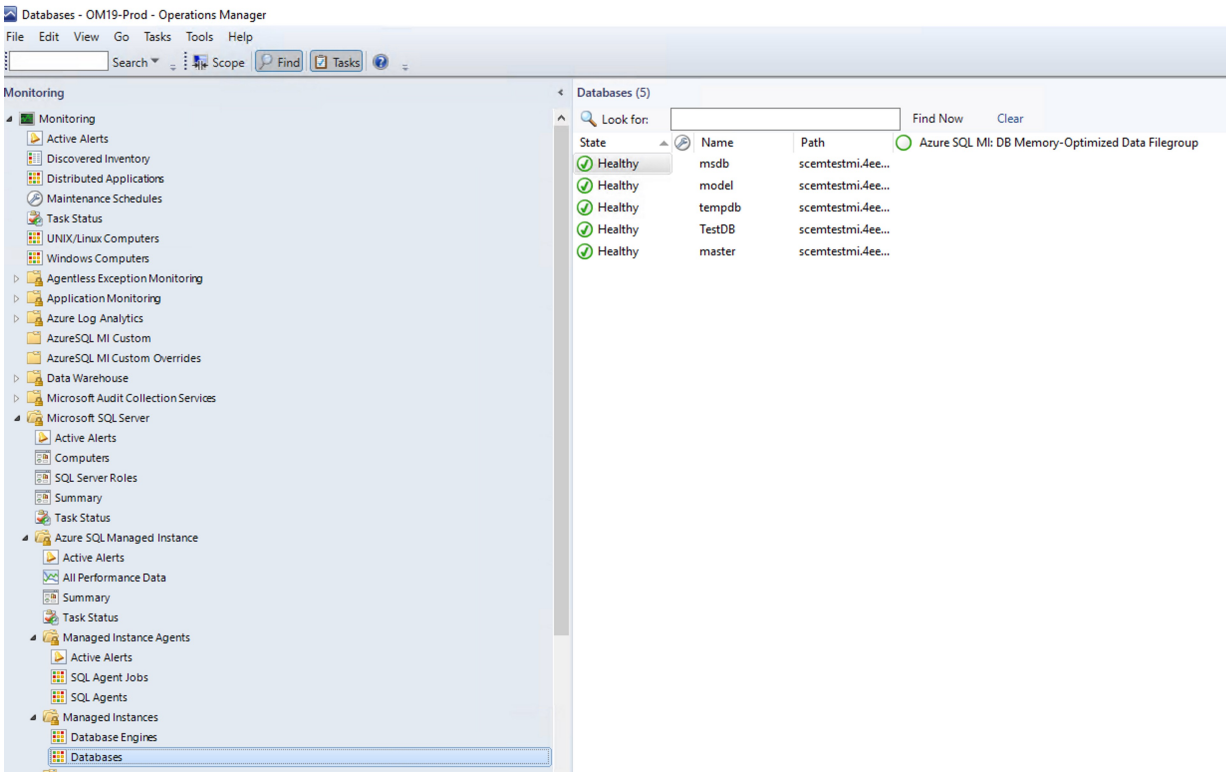




Administration -> Run As Configuration -> Accounts -> SQL Managed Instance Credentials



Monitoring -> Microsoft SQL Server -> Azure SQL Managed Instance

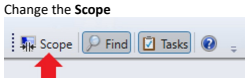


Azure SQL Managed Instance Discoveries

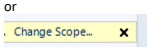
To See all the Object Discoveries Associated with Azure SQL MI you will need to perform the following steps.

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.

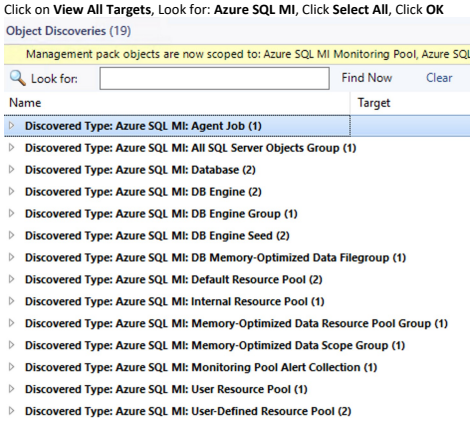
1. Go to **Authoring -> Management Pack Objects -> Object Discoveries**
   a. Change the **Scope**



or



   b. Click on **View All Targets**, Look for: **Azure SQL MI**, Click **Select All**, Click **OK**



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 } }, TimeStarted, TimeScheduled, TimeFinished, Output) | Out-String -Width 4096
    $output
    }
}
catch
{
    Write-Warning $_
    Write-Host "Unable to trigger the discovery" -ForegroundColor Red
}
```

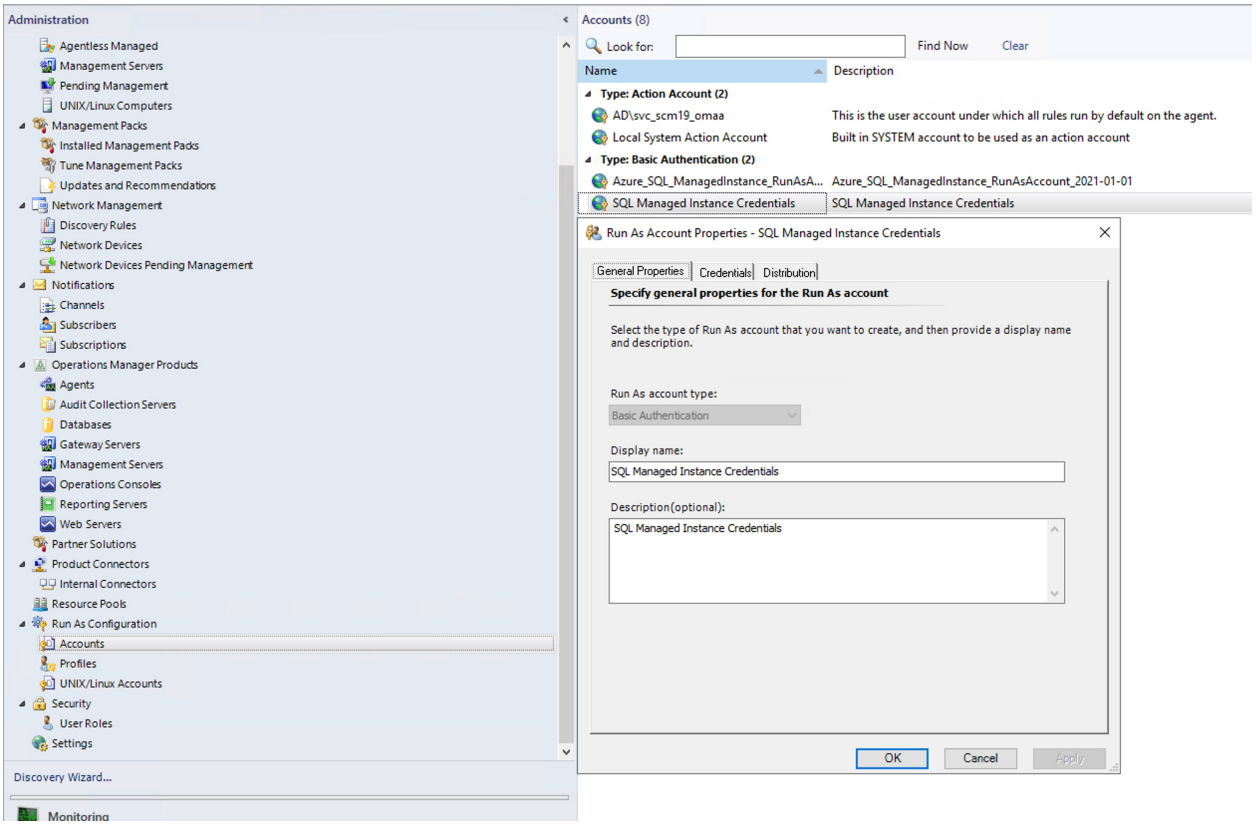
Least Privilege Configuration for Monitoring

1. Open - SQL Server Management Studio
   a. Server name: **scemtestmi.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 @loginame = @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_readerrorlog 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



a. Go to Credentials Tab, set the following:

Account Name	**MILowPriv**
Password	**LowPrivPassword1234!**

b. Click **OK**