SQL Server Assessment: Prerequisites and Configuration

This document explains the required steps to configure the SQL Server Assessment included with your Azure Log Analytics Workspace and Microsoft Unified Support Solution Pack.

There are **two scenarios** available to configure the assessment. Determine which scenario fits best for your organization.

1. OMS Gateway and data collection machine
2. Data collection machine only

OMS Gateway and data collection machine
This scenario is the most secure and recommended option to help protect privileged account credentials which are used on the scheduled task configured on this machine needed to run the assessment. This scenario requires two computers. One will be designated as the data collection machine, and the second machine will be the OMS Gateway. In this scenario, the data collection machine has no Internet connection and connects to the OMS Gateway to upload the data to log analytics. The OMS Gateway must have Internet access. This scenario is recommended for environments where the Internet connection is restricted from the data collection machine or where security is a concern due to this schedule task requirement. For information about the OMS Gateway, go to https://docs.microsoft.com/en-us/azure/azure-monitor/agents/gateway.

The data collection machine must be a member of the domain containing the SQL Server environment being assessed. It will collect data from multiple servers or failover clusters running SQL Server. After the data is collected, the data collection machine will analyze the information, and for increased security, will forward the data to an OMS Gateway to upload it to log analytics.

The following path shows the relationship between your Windows computers and log analytics after you have installed and configured the OMS Gateway and data collection machine.

Data collection machine → Collects data from multiple servers or failover clusters running SQL Server → Forward collected data to the OMS Gateway → Submit data to the log analytics workspace

Data collection machine only
This scenario can be used when the data collection machine can contact log analytics directly. It requires one computer that will be designated as the data collection machine which has to be able to access the Internet to upload data to log analytics. This scenario can be used in environments where the Internet connection is not restricted.

The data collection machine must be a member of the domain containing the SQL Server environment being assessed. It will collect data from multiple servers or failover clusters running SQL Server. After the data is collected, the data collection machine will analyze the information and then upload the data to log analytics directly, which will require HTTPS connectivity to your log analytics workspace. The following path shows the relationship between your Windows computers and log analytics after you have installed and configured the data collection machine:

Data collection machine → Collects data from multiple servers or failover clusters running SQL Server → Submit data to the log analytics workspace.

Detailed information on these configurations and requirements is found later in this document. The Resource Center contains KB Articles, FAQs and Videos that will help you with the Assessment configuration and execution.

This document was last updated on April 12th, 2022. To ensure you have the latest version of this document, check here:
https://docs.microsoft.com/en-us/services-hub/health/assessment_prereq_docs/prereqssqlassessment.pdf

Table of Contents

System Requirements and Configuration at Glance **3**

Supported Versions 3

Common to Both Scenarios 3

Data Collection Machine 3

OMS Gateway (required in the **OMS Gateway and data collection machine** scenario) 4

PowerShell Remoting 4

User Profile Service 9

Setting up the SQL Assessment **10**

Appendix – A Data Collection Methods **13**

Appendix – B Ports requirements **15**

Appendix – C Special Requirements for Availability Group Cluster in Azure **16**

Appendix – D Requirements to run without sysadmin **17**

System Requirements and Configuration at Glance

According to the scenario you want to use, review the following details to ensure that you meet the necessary requirements.

Supported Versions

- Your SQL Server environment must run on SQL Server 2012, SQL Server 2014, SQL Server 2016, SQL Server 2017, SQL Server 2019. Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019 failover cluster or standalone server installations are supported.

Common to Both Scenarios

- You will need a log analytics workspace
- **User account rights:**
    - A domain account, Standalone Managed Service Account (sMSA) or Group Managed Service Account (gMSA) with the following rights:
        - Member of the local Administrators group on all servers in the environment
        - SysAdmin role on all Microsoft SQL Servers in the environment
    - For Standalone Managed Service Account (sMSA) or Group Managed Service Account (gMSA) you can learn more at https://docs.microsoft.com/en-us/services-hub/health/kb-running-assessments-with-msas

Data Collection Machine

- **Microsoft Monitoring Agent** (using OnDemand Assessments) requires computers running Windows Server 2012 or later (or Windows 10 or later). **Important**: The option of installing the Microsoft Monitoring Agent on client operating systems is strongly discouraged due to the risk of exposing privileged domain account credentials to lower trust workstations.
- Microsoft **.NET Framework 4.8 or newer** installed.
- The **data collection machine** must be a member server of the Active Directory domain which includes the SQL Server environment that needs to be assessed.
- **Data collection machine hardware:** Minimum 16 gigabytes (GB) of RAM, 2 gigahertz (GHz) dual-core processor, and minimum 10 GB of free disk space.
  Note: If the target environment is large or complex, our Support Engineers may request to upgrade the RAM on the data collection machine. Boosting the memory will have a direct impact on the data collection speed.
- The **data collection machine** is used to connect to server(s) running SQL Server environment and retrieve information from the environment. The machine communicates over Remote Procedure Call (RPC), Server Message Block (SMB), WMI, remote registry, SQL Server, Lightweight Directory Access Protocol (LDAP) and Distributed Component Object Model (DCOM).
- The **data collection machine** must be able to connect to the Internet using HTTPS to submit the collected data to your log analytics workspace. This connection can be direct, via a proxy.
- For the **Microsoft Monitoring Agent** to connect to and register with the log analytics service, it must have access to the Internet. If you use a proxy server for communication between the agent and the log analytics service, you will need to ensure that the appropriate resources are accessible. If you use a firewall to restrict access to the Internet, you need to configure your firewall to permit access to log analytics. To ensure data can be submitted follow the steps in *Configure Proxy and Firewall Settings in Log Analytics* at https://docs.microsoft.com/en-us/azure/azure-monitor/platform/log-analytics-agent#network-firewall-requirements

- Antivirus and any other type of Security software need to be configured to exclude Assessment related files, file types, working directory folders and process (Omsassessment.exe) to avoid process termination, blockage and alerts. Add an exclusion to Windows Security

OMS Gateway (required in the **OMS Gateway and data collection machine** scenario)

- The **OMS Gateway** can be a standalone or a member server. It requires Windows 10, Windows 11, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016 or Windows Server 2019.
- The **OMS Gateway** must be able to connect to the Internet using HTTPS to submit the collected data to your log analytics workspace. This connection can be direct or via a proxy.
- **OMS Gateway hardware:** Minimum 4 GB of RAM and 2 GHz processor.
- **OMS Gateway services:** When the Windows Firewall service is disabled the installation of the OMS Gateway fails.
- **OMS Gateway user account rights:** None required.

For detailed information about the Microsoft Monitoring Agent including system requirements, network firewall configuration requirements, download, and installation instructions, see the following:
https://docs.microsoft.com/en-us/azure/azure-monitor/platform/agent-windows
https://docs.microsoft.com/en-us/azure/azure-monitor/agents/gateway

PowerShell Remoting

To complete the assessment with accurate results, you will need to configure all in-scope target machines for PowerShell remoting.

PowerShell on the tools machine is used to scan the servers for installed security patches as well as audit policy configuration.

- Windows Update Agent must be running on all SQL Servers for the security update scan

Additional requirements for Windows Server Target Machines:

The following three items must be configured on target SQL Servers to support data collection: PowerShell Remoting, WinRM service and Listener, and Inbound Allow Firewall Rules.

Note: *Starting Windows Server 2012 R2 WinRM and PowerShell remoting are enabled by default. One of the following configuration options will need to be implemented support PowerShell Remoting if WinRM and PowerShell remoting are disabled on any target SQL servers:*

Option 1: Execute **Enable-PSRemoting** Powershell cmdlet on each target machine within the scope of the assessment. This one command will configure PS-Remoting, WinRM service and listener, and enable required Inbound FW rules. A detailed description of everything Enable-PSRemoting does is documented here.

OR

Option 2: Implement all three steps below.

1. Configure **WinRM / PowerShell remoting** via Group Policy (Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service)
2. Configure **WinRM service for automatic start** via Group Policy (Computer Configuration\Policies\Windows Settings\Security Settings\SystemServices)
    a. Define **Windows Remote Management** (WS-Management) service for **Automatic startup mode**

3. Configure **Inbound allow Firewall Rules:** This can be done individually in the local firewall policy of every in-scope target SQL Servers or via a group policy which allow communication from the tools machine.

Detailed step by step instructions to implement the above 3 steps.

Two steps are involved to configure a group policy to enable both WinRM listener and the required inbound allow firewall rules:

A) Identify the IP address of the source computer where data collection will occur from.

B) Create a new GPO linked to the SQL Server organizational unit, and define an inbound rule for the tools machine

A.) Log into the chosen data collection machine to identify its current IP address using IPConfig.exe from the command prompt.

An example output is as follows

C:\>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :

Link-local IPv6 Address . . . . . : fe80::X:X:X:X%13

IPv4 Address. . . . . : **X.X.X.X**

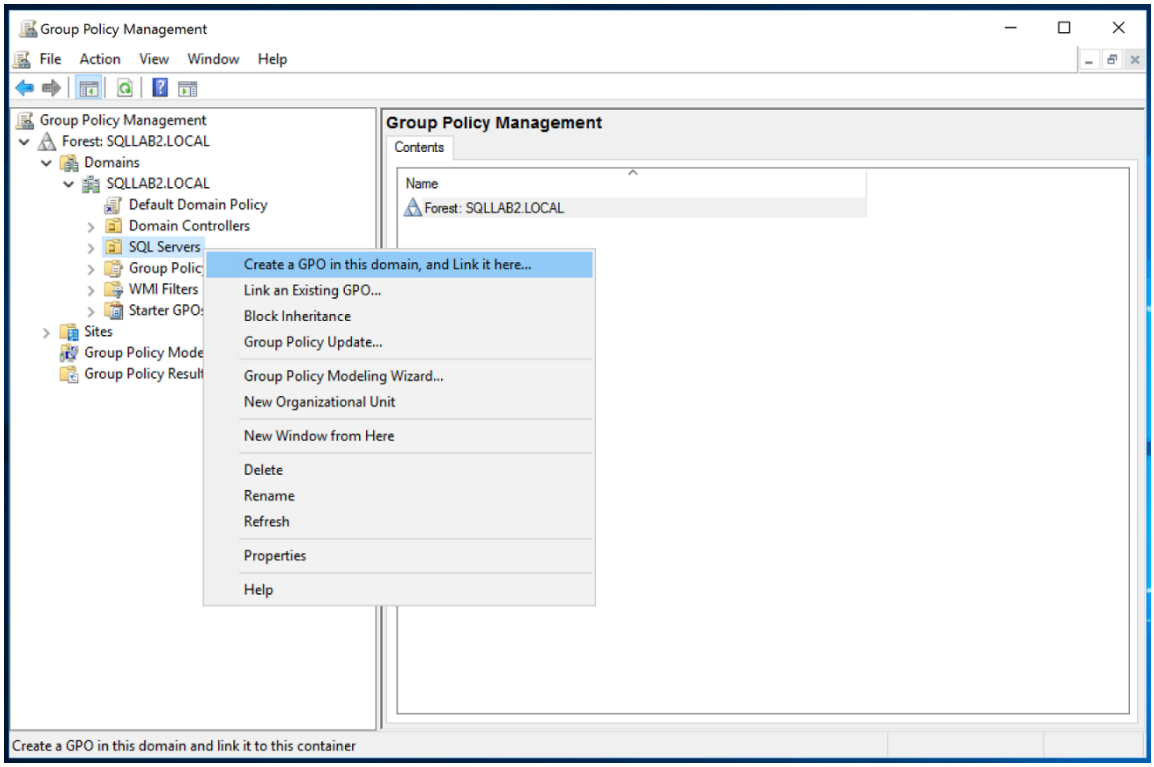
Subnet Mask . . . . . : X.X.X.X

Default Gateway . . . . . : X.X.X.X

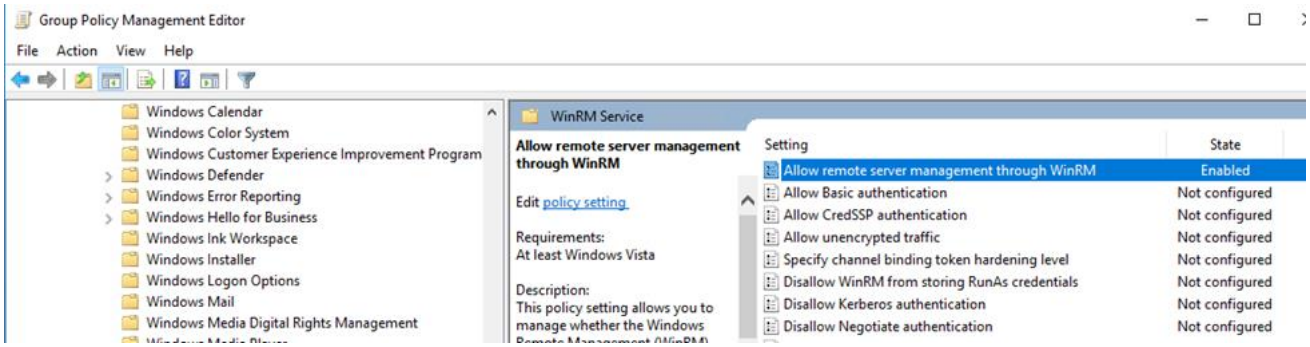
Make a note of the IPv4 address of your machine. The final step in the configuration will use this address to ensure only the data collection machine can communicate with the Windows Update Agent on the SQL Servers.

B.) Create, configure, and link a group policy object to the SQL Servers OU which has target SQL Servers.

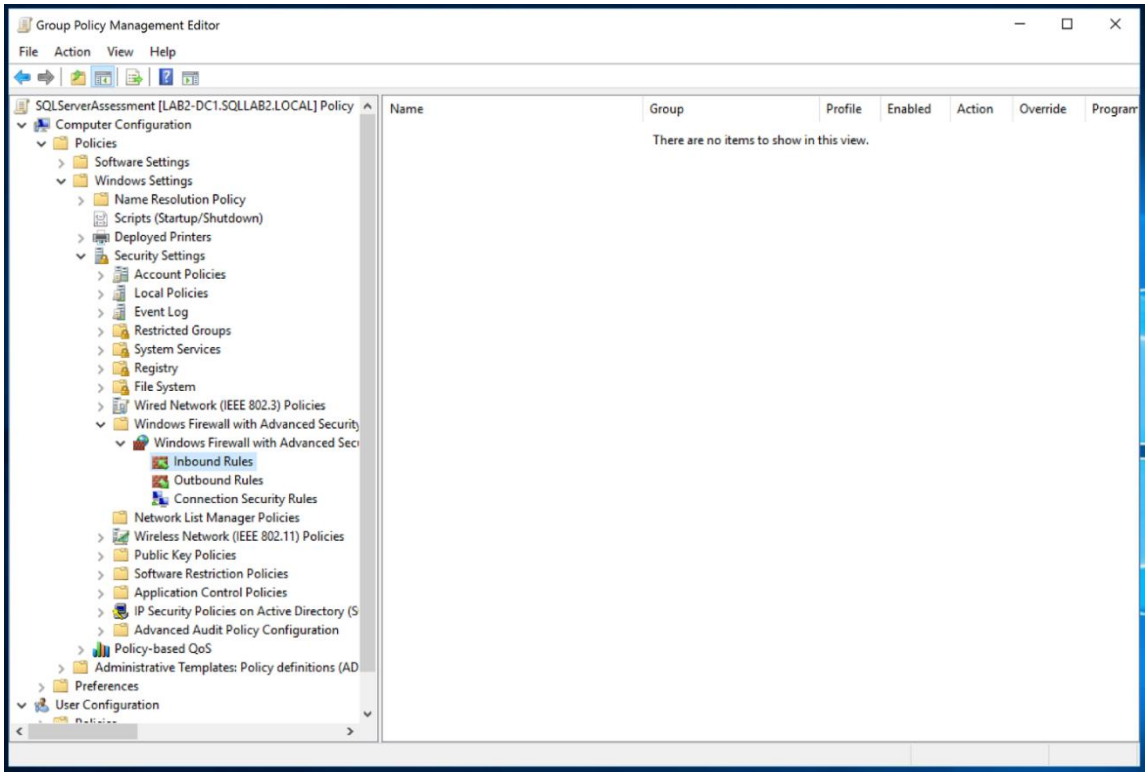
1. Create a new GPO. Make sure the GPO applies to the SQL Servers organizational unit. Give the new group policy a name based on your group policy naming convention or something that identifies its purpose similar to "SQLServerAssessment"



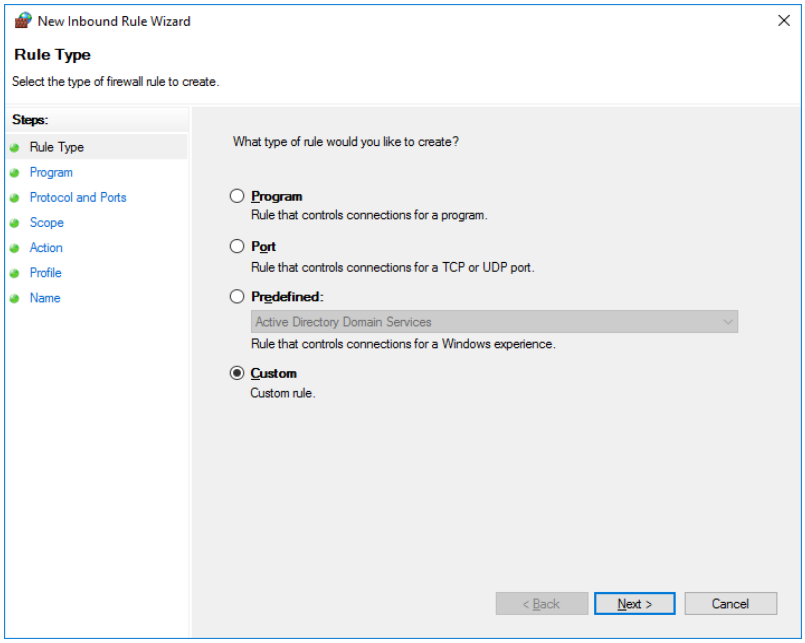
2. Within the GPO open, right click on the new GPO and select Edit and go to "Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service". Enable "**Allow remote server management through WinRM**" or "**Allow automatic configuration of listeners**" depending on your OS.



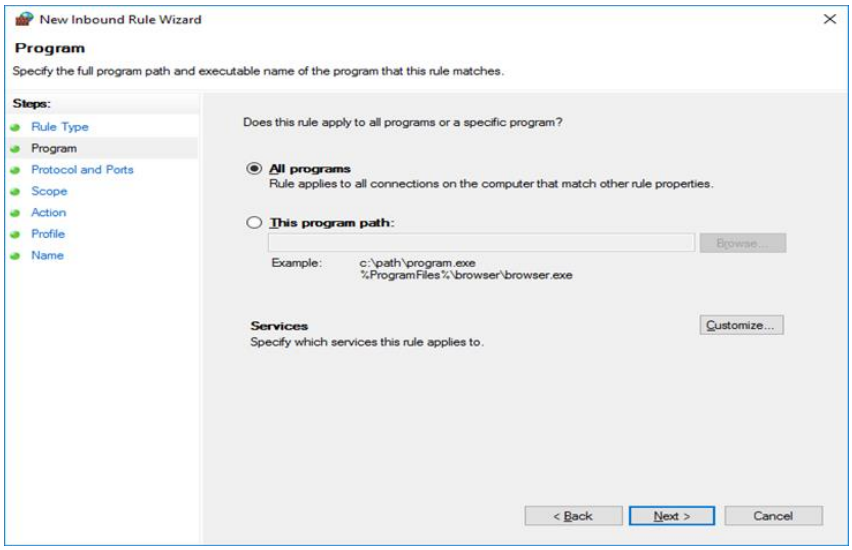
3. Create an advanced Inbound Firewall Rule to allow all network traffic from the tools machine to the SQL Servers. This can be the applied to the same GPO that was used in step 1 above. (Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Windows Firewall with Advanced Security –LDAP:/xxx\Inbound Rules)



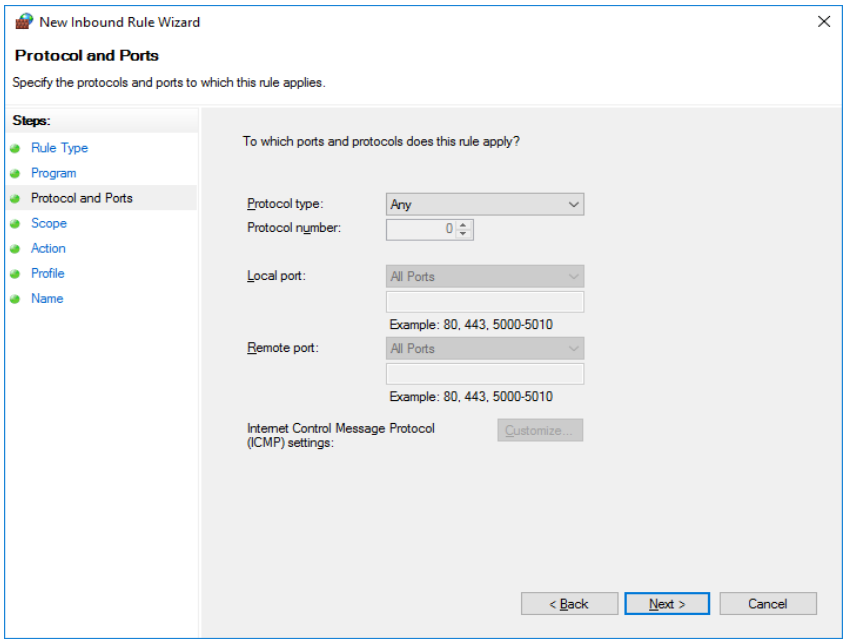
4. To create the new rule, Right Click on “Inbound Rules” and select “New”
5. Create a custom rule and choose “Next”



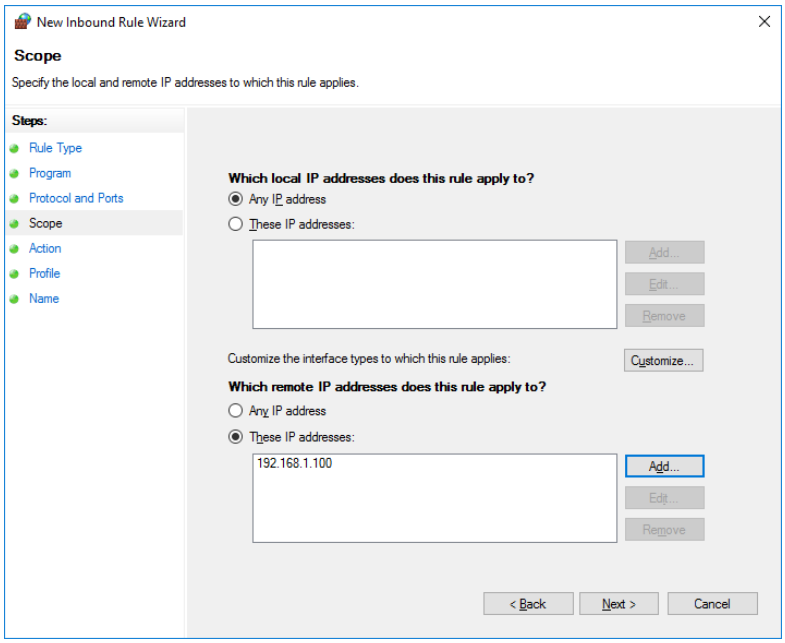
6. Allow “All programs” from the tools machine and click “Next”.



7. Allow all protocols and ports, then click “Next”.



8. Specify the IP address of the tools machine and click “Next”.



9. Choose “Allow the connection” and click Next

10. Choose to select network profile “Domain” and click “Next”

11. Choose a name for the rule (Example: SQLAssessmentToolsMachine) and save.

12. Navigate to Computer Configuration\Policies\Windows Settings\Security Settings\System Services. Select and Define Windows Remote Management (WS-Management) service for automatic startup.

13. Save the GPO and ensure the target SQL servers apply it.

User Profile Service

It is necessary to modify the default behavior of the User Profile Service as it relates to user logoff. Windows, by default, forcibly unloads user registry hive on logoff even if there are applications with open handles to the user registry hive. This default behavior interferes with remote Powershell initialization routines during execution of the on-demand assessment via scheduled task and can prevent successful collection and submission of assessment data to the log analytics portal.

On the data collection machine, change the following setting in the group policy editor (gpedit.msc) from "not configured" to "enabled":

Computer Configuration->Administrative Templates->System-> User Profiles

'Do not forcefully unload the user registry at user logoff'

After you have finished the installation of the Microsoft Monitoring Agent/OMS Gateway, and configured Security Updates Prerequisites on the Data Collection machine and target machines, continue with the next section to set up the assessment. Additional details can be found at Configure Microsoft On-Demand Assessment

Do not allow storage password policy

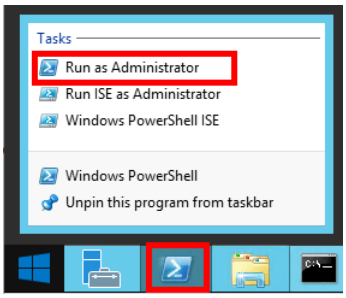
For the collection machine you might need to explicitly disable the policy “Network Access: Do not allow storage of passwords and credentials for network authentication”. Additional details can be found at Configure Microsoft On-Demand Assessment

Setting up the SQL Assessment

When you have finished the installation of the Microsoft Monitoring Agent/OMS Gateway, you are ready to setup the SQL Assessment.

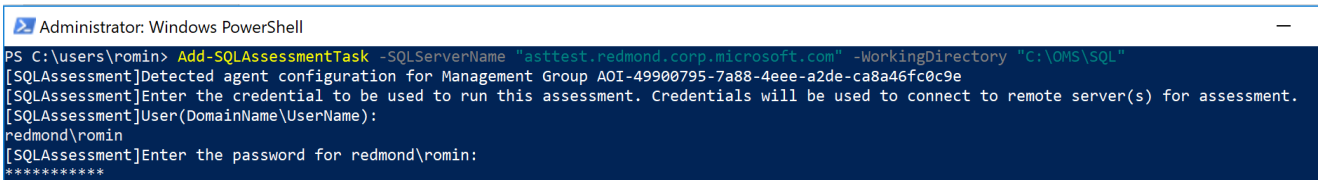
On the designated data collection machine, complete the following:

1. Open the Windows PowerShell command prompt as an Administrator



2. Run the **Add-SQLAssessmentTask -SQLServerName <YourServerName> – WorkingDirectory <Directory>** command where <*YourServerName*> is the fully qualified domain name (FQDN) or the NetBIOS name of single server or failover cluster running SQL Server environment. If more than one environment is assessed, ";" is used between the environments. For failover cluster, check failover cluster virtual network name and <*Directory*> is the path to an existing directory used to store the files created while collecting and analyzing the data from the environment(s).
   NOTE: If the directory does not exist, it must be created before you continue with the execution.

3. Provide the required user account credentials. These credentials are used to run the SQL Assessment.
   For sMSA or gMSA the parameter **RunWithManagedServiceAccount** must be supplied and set it to **$True**, when the cmdlet prompts the user for a password, leave it blank and press <Enter>. For more information review the article *"Running Assessments with Managed Service Accounts"*



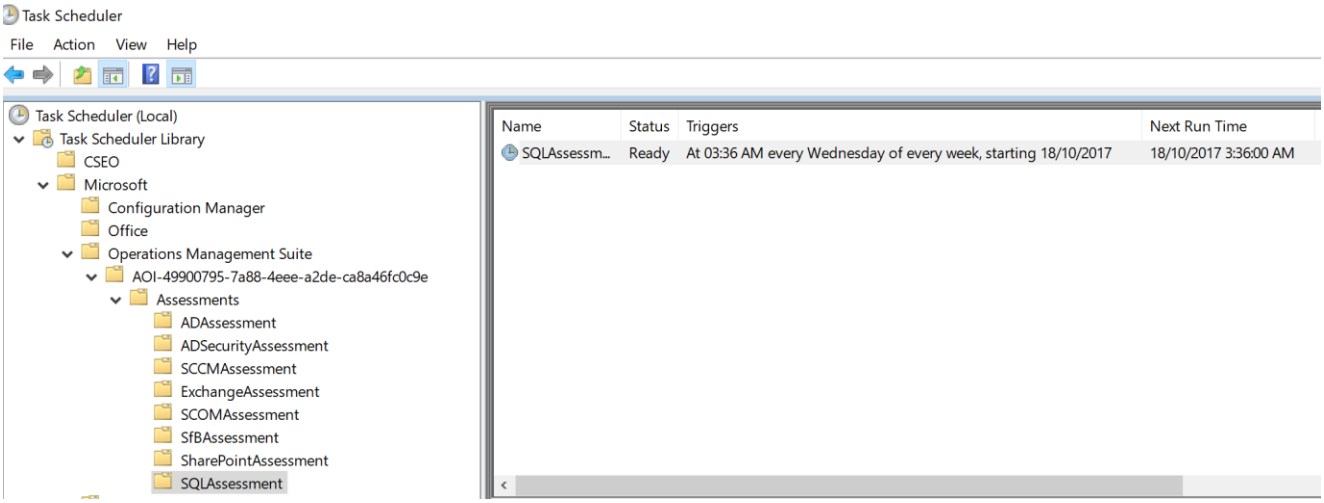
   NOTE: This domain account must have all the following rights:

   - Member of the local Administrators group on all servers in the environment
   - SysAdmin role on all Microsoft SQL Servers in the environment

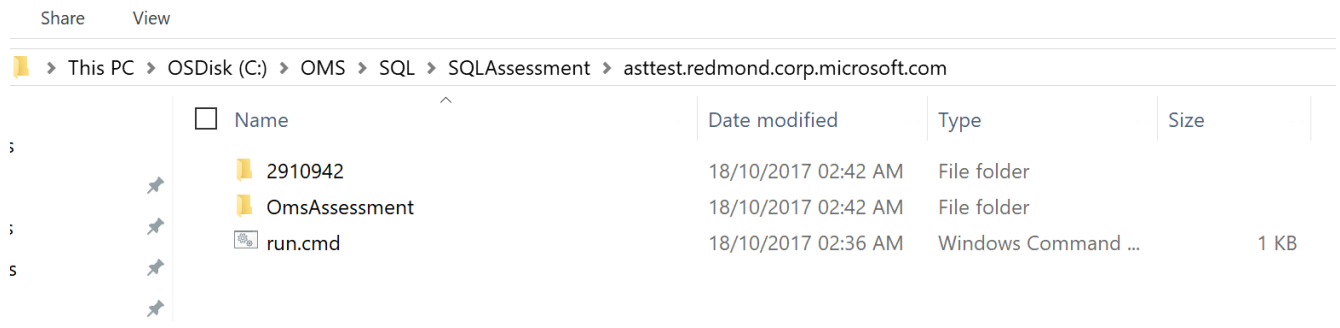
4. The script will continue with the necessary configuration. It will create a scheduled task that will trigger the data collection.

```
Administrator: Windows PowerShell
PS C:\users\romin> Add-SQLAssessmentTask -SQLServerName "asttest.redmond.corp.microsoft.com" -WorkingDirectory "C:\OMS\SQL"
[SQLAssessment]Detected agent configuration for Management Group AOI-49900795-7a88-4eee-a2de-ca8a46fc0c9e
[SQLAssessment]Enter the credential to be used to run this assessment. Credentials will be used to connect to remote server(s) for assessment.
[SQLAssessment]User(DomainName\UserName):
redmond\romin
[SQLAssessment]Enter the password for redmond\romin:
**********
[SQLAssessment]Creating Windows Schedule task to run assessment...
[SQLAssessment]SQLAssessment setup successful.
[SQLAssessment]Detailed log is at: C:\Users\romin\AppData\Local\Temp\Assessments_Configuration_20171018_093612.log
PS C:\users\romin>
```

5. Data collection is triggered by the **scheduled task** named **SQLAssessment -ServerName <YourServerName>** within an hour of running the previous script and then every 7 days. The task can be modified to run on a different date/time or even forced to run immediately.

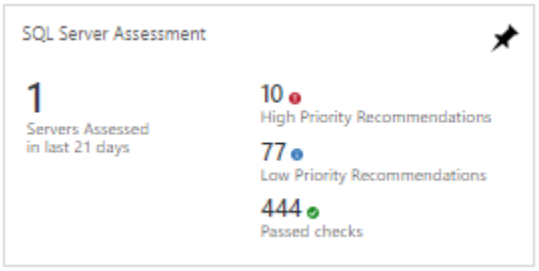


6. During collection and analysis, data is temporarily stored under the **WorkingDirectory** folder that was configured during setup, using the following structure:

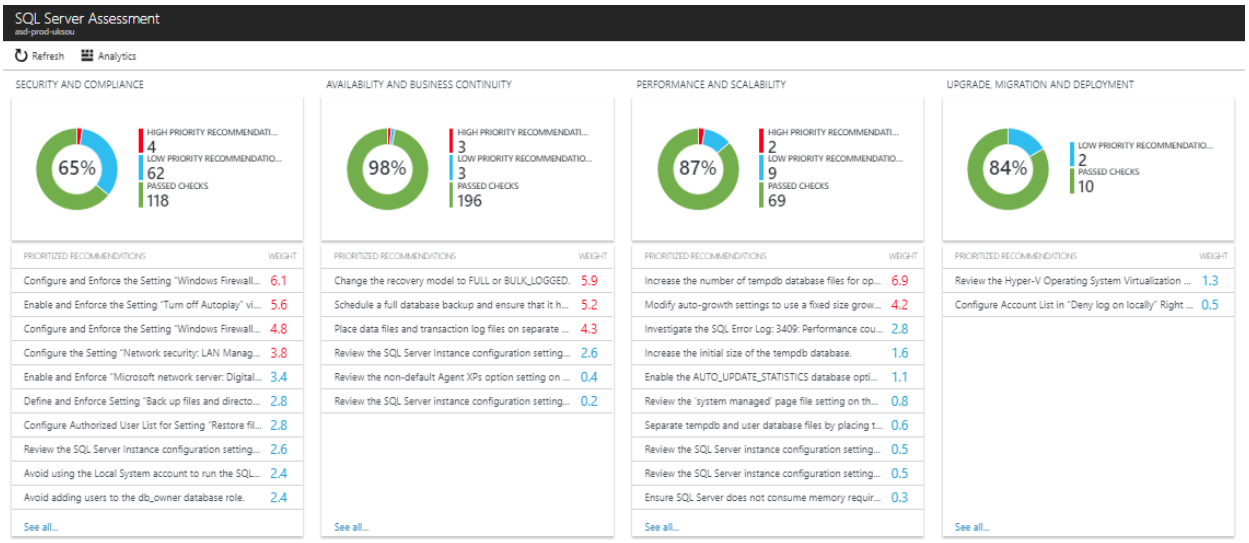


7. After data collection and analysis is completed on the tools machine, it will be submitted to your log analytics workspace depending on the scenario you have chosen:
   - **Directly** if the Data Collection Machine is connected to the Internet.
   - **Through the OMS Gateway** if the Data Collection Machine is not connected to the Internet.

8. After a few hours, your assessment results will be available on your log analytics dashboard. Click the **SQL Assessment** tile to review:



9. You will be presented with findings grouped by the focus area.



Appendix – A Data Collection Methods

The **SQL Assessment in the log analytics workspace and Microsoft Unified Support Solution Pack** uses multiple data collection methods to collect information from your environment. This section describes the methods used to collect data from your environment. No Microsoft Visual Basic (VB) scripts are used to collect data.

1. Event Log collectors
2. File Data collectors
3. Registry data collectors
4. User rights collectors
5. Mount points data collectors
6. SQL data collectors
7. SQL error log collectors
8. WMI data collectors
9. Windows PowerShell data collectors

1. Event Log collectors

Collects last 7 days of application and system event logs including Warnings and Errors from SQL Servers.

2. File data collectors

Enumerates files in a folder on a remote machine, and optionally retrieves those files.

3. Registry data collectors

Registry keys and values are collected from the SQL Server environment. They include items such as:

- Power options information from "HKLM\SYSTEM\CurrentControlSet\Control\Power\User\PowerSchemes"
- This allows you to understand how you set power plan on SQL Server.

4. User rights collectors

Collect local security policies.

5. Mount points data collectors

Collect mount points if they exist.

6. **SQL data collectors**

   T-SQL queries are used to collect information such as:

   - Always On configuration, database names, health status
   - TempDB files, file sizes, auto growth sizes
   - Backup histories
   - Duplicate, redundant, disabled, hypothetical indexes

7. **SQL error log collectors**

   Collects last 15 days SQL Server error logs. If any log file size is more than 6MB, it is not analyzed.

8. **WMI data collectors**

   WMI is used to collect various information such as:

   - WIN32_Volume
     Collects information on Volume Settings for each server in the environment. For example, the information is used to determine the system volume and drive letter, which allows a client to collect information on the files located on the system drive.
   - Win32_Process
     Collect information on the processes running on each server in the environment. The information provides insight in processes that consume a large amount of threads, memory, or have a large page file usage.
   - Win32_LogicalDisk
     Used to collect information on the logical disks. Microsoft use the information to determine the amount of free space on the disk where the database or log files are located.

9. **Windows PowerShell data collectors**

   PowerShell is used to collect various information such as:

   - Collect resource dependencies
   - Auditing configuration

Appendix – B Ports requirements

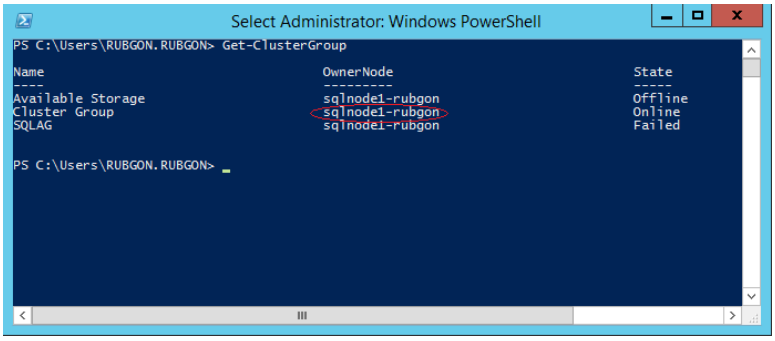
If opening all ports between tools machine and target servers is not possible, here is the list of specific ports that need to be open for the toolset to work correctly. These ports need to be open in each target server.

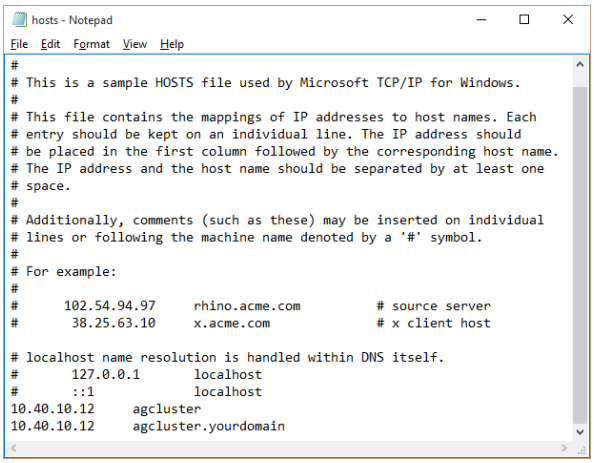
- SQL Instances ports, 1433 or as configured by Customer.
- SQL Browser port – UDP 1434
- RPC Dynamic Ports Range
    - By default TCP ports from 49152 to 65535
    - They can change this range but it could have some other implications that they need to evaluate as this range of ports is used by different components and you need to be careful of not running out of ports. https://support.microsoft.com/en-us/help/929851/the-default-dynamic-port-range-for-tcp-ip-has-changed-in-windows-vista
- Default PS Remoting ports are TCP 5985 and TCP 5986, if customer changed, it could be 80 and 443 or any other ports they specified.
- Port TCP 135
- Port UDP 137 (maybe not necessary)
- Port TCP 139
- Port TCP 445

Appendix – C Special Requirements for Availability Group Cluster in Azure

Virtual networks in Azure put some connectivity restrictions to clusters and availability groups that affect the toolset. In summary:

1. If you are using external load balancer you cannot use the listener name for discovery. You can use the cluster name or a node name.
2. If you are not using an Standard Load Balancer with HA ports for the Cluster Name IP, then you need to modify the hosts file to make the Cluster Collectors work. You need to identify the IP of the active node and modify the hosts files as shown below. The hosts file is located at "C:\Windows\System32\drivers\etc".





Note: The hosts configuration change needs to be reviewed every time you run the toolset in case a failover has happened since the last time the toolset was executed.

Appendix – D Requirements to run without sysadmin

If granting sysadmin privileges is not possible you can use the script provided here to grant only necessary privileges to the account running the assessment. Consider that a few rules can only collect the necessary data when running with sysadmin, these rules will be skipped when not running with sysadmin privileges.

One special case is when you have read only databases (other than availability group databases). In that case the necessary user won't be created and several rules will be skipped on these read only databases. To collect necessary information for these databases you need to change to read-write only to run the script provided here to create the necessary user and permissions or run the toolset with an account that is sysadmin (the toolset only supports Windows Authentication).

The script below is long because it grants permissions that usually are already gr anted to public role by default. This means the script will work even when all permissions have been revoked from public.

The script to create the login users and grant permissions is provide here. Remember that the script needs to be run in each instance that will be assessed. Given the limitations of a document to include long scripts, be careful when copying the script as it spans several pages.

```
DECLARE @UserName nvarchar(500) = 'NORTHAMERICA\RaaSUser', --replace with your domain and username, the user needs to exist in the domain
@Command nvarchar(max)
SET @Command = 'USE master;
IF NOT EXISTS(SELECT name FROM sys.server_principals WHERE name LIKE ''' + @UserName + ''')
BEGIN
        CREATE LOGIN [' + @UserName + '] FROM WINDOWS WITH DEFAULT_DATABASE=[master], DEFAULT_LANGUAGE=[us_english];
END'
EXEC sp_executesql @Command
--Create user on each database
SET @Command = 'USE [?];
IF EXISTS (SELECT 1
FROM sys.databases d LEFT JOIn sys.dm_hadr_database_replica_states r ON d.database_id = r.database_id
WHERE d.is_read_only = 0
AND (r.is_primary_replica = 1 OR r.is_primary_replica IS NULL)
AND d.name = DB_NAME()
)
BEGIN
        IF NOT EXISTS(SELECT name FROM sys.database_Principals WHERE name LIKE ''' + @UserName + ''')
        BEGIN
                CREATE USER [' + @UserName + '] FOR LOGIN [' + @UserName + '];
        END
END
'
EXECUTE master.sys.sp_MSforeachdb @Command

--master permissions
SET @Command = '
USE master;
GRANT VIEW SERVER STATE TO [' + @UserName + ']
GRANT VIEW ANY DEFINITION TO [' + @UserName + ']
GRANT SELECT ON sys.master_files TO [' + @UserName + ']
GRANT SELECT ON sys.databases TO [' + @UserName + ']
GRANT SELECT ON sys.configurations TO [' + @UserName + ']
GRANT SELECT ON sys.sql_logins TO [' + @UserName + ']
GRANT SELECT ON sys.server_principals TO [' + @UserName + ']
GRANT SELECT ON sys.server_role_members TO [' + @UserName + ']
GRANT SELECT ON sys.endpoints TO [' + @UserName + ']
GRANT SELECT ON sys.database_mirroring_endpoints TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_loaded_modules TO [' + @UserName + ']
GRANT SELECT ON sys.servers TO [' + @UserName + ']
GRANT SELECT ON sys.server_audits TO [' + @UserName + ']
GRANT SELECT ON sys.server_event_sessions TO [' + @UserName + ']
GRANT SELECT ON sys.tcp_endpoints TO [' + @UserName + ']
GRANT SELECT ON sys.database_mirroring TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_index_usage_stats TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_performance_counters TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_sys_info TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_nodes TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_schedulers TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_partition_stats TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_persisted_sku_features TO [' + @UserName + ']
```

```
GRANT SELECT ON sys.dm_db_missing_index_details TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_missing_index_groups TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_missing_index_group_stats TO [' + @UserName + ']
GRANT SELECT ON sys.dm_xe_sessions TO [' + @UserName + ']
GRANT SELECT ON sys.dm_exec_query_stats TO [' + @UserName + ']
GRANT SELECT ON sys.dm_exec_text_query_plan TO [' + @UserName + ']
GRANT SELECT ON sys.dm_exec_sql_text TO [' + @UserName + ']
GRANT SELECT ON sys.dm_os_wait_stats TO [' + @UserName + ']
GRANT SELECT ON sys.dm_exec_connections TO [' + @UserName + ']
'
EXEC sp_executesql @Command


SET @Command = 'GRANT EXEC ON sys.xp_enumerrorlogs TO [' + @UserName + ']
GRANT EXEC ON sys.sp_executesql TO [' + @UserName + ']
GRANT EXEC ON sys.sp_validatelogins TO [' + @UserName + ']
--For SQL Server 2012 or later
IF CONVERT(int,SUBSTRING(CONVERT(varchar,SERVERPROPERTY(''ProductVersion'')), 1, 2)) >= 11
BEGIN
GRANT SELECT ON sys.availability_groups TO [' + @UserName + ']
GRANT SELECT ON sys.availability_replicas TO [' + @UserName + ']
GRANT SELECT ON sys.availability_group_listener_ip_addresses TO [' + @UserName + ']
GRANT SELECT ON sys.availability_group_listeners TO [' + @UserName + ']
GRANT SELECT ON sys.dm_hadr_availability_replica_states TO [' + @UserName + ']
GRANT SELECT ON sys.dm_db_stats_properties TO [' + @UserName + ']
GRANT SELECT ON sys.dm_hadr_availability_group_states TO [' + @UserName + ']
GRANT SELECT ON sys.dm_hadr_database_replica_states TO [' + @UserName + ']
END
--For SQL 2017 or later
IF CONVERT(int,SUBSTRING(CONVERT(varchar,SERVERPROPERTY(''ProductVersion'')), 1, 2)) >= 14
BEGIN
GRANT SELECT ON sys.dm_db_log_info TO [' + @UserName + ']
END'

EXEC sp_executesql @Command

--msdb permissions
SET @Command = '
USE msdb
GRANT SELECT ON dbo.backupmediafamily TO [' + @UserName + ']
GRANT SELECT ON dbo.backupset TO [' + @UserName + ']
GRANT SELECT ON dbo.backupfile TO [' + @UserName + ']
GRANT SELECT ON dbo.backupmediaset TO [' + @UserName + ']
GRANT SELECT ON dbo.restorefile TO [' + @UserName + ']
GRANT SELECT ON dbo.restorefilegroup TO [' + @UserName + ']
GRANT SELECT ON dbo.restorehistory TO [' + @UserName + ']
GRANT SELECT ON dbo.sysdbmaintplans TO [' + @UserName + ']
GRANT SELECT ON dbo.log_shipping_monitor_secondary TO [' + @UserName + ']
GRANT SELECT ON dbo.log_shipping_secondary_databases TO [' + @UserName + ']
GRANT SELECT ON dbo.log_shipping_secondary TO [' + @UserName + ']
GRANT SELECT ON dbo.log_shipping_monitor_primary TO [' + @UserName + ']
GRANT SELECT ON dbo.log_shipping_primary_databases TO [' + @UserName + ']
GRANT SELECT ON dbo.sysjobs TO [' + @UserName + ']
GRANT SELECT ON dbo.sysjobhistory TO [' + @UserName + ']
GRANT SELECT ON dbo.suspect_pages TO [' + @UserName + ']
IF EXISTS(SELECT 1 FROM sys.objects WHERE name = ''MSdistributiondbs'')
GRANT SELECT ON dbo.MSdistributiondbs TO [' + @UserName + ']
'
EXEC sp_executesql @Command

--user databases permissions
SET @Command = '
USE [?];
IF EXISTS (SELECT 1
FROM sys.databases d LEFT JOIn sys.dm_hadr_database_replica_states r ON d.database_id = r.database_id
WHERE d.is_read_only = 0
AND (r.is_primary_replica = 1 OR r.is_primary_replica IS NULL)
AND d.name = DB_NAME()
)
BEGIN
GRANT SELECT ON sys.foreign_keys TO [' + @UserName + ']
GRANT SELECT ON sys.database_files TO [' + @UserName + ']
GRANT SELECT ON sys.allocation_units TO [' + @UserName + ']
GRANT SELECT ON sys.extended_properties TO [' + @UserName + ']
GRANT SELECT ON sys.objects TO [' + @UserName + ']
GRANT SELECT ON sys.partitions TO [' + @UserName + ']
GRANT SELECT ON sys.schemas TO [' + @UserName + ']
GRANT SELECT ON sys.indexes TO [' + @UserName + ']
GRANT SELECT ON sys.internal_tables TO [' + @UserName + ']
GRANT SELECT ON sys.database_principals TO [' + @UserName + ']
GRANT SELECT ON sys.all_objects TO [' + @UserName + ']
```

```
GRANT SELECT ON sys.database_permissions TO [' + @UserName + ']
GRANT SELECT ON sys.database_role_members TO [' + @UserName + ']
END
'

EXECUTE master.sys.sp_MSforeachdb @Command
SET @Command = '
USE [?];
IF EXISTS (SELECT 1
FROM sys.databases d LEFT JOIn sys.dm_hadr_database_replica_states r ON d.database_id = r.database_id
WHERE d.is_read_only = 0
AND (r.is_primary_replica = 1 OR r.is_primary_replica IS NULL)
AND d.name = DB_NAME()
)
BEGIN
GRANT SELECT ON sys.symmetric_keys TO [' + @UserName + ']
GRANT SELECT ON sys.asymmetric_keys TO [' + @UserName + ']
GRANT SELECT ON sys.assembly_modules TO [' + @UserName + ']
GRANT SELECT ON sys.assemblies TO [' + @UserName + ']
GRANT SELECT ON sys.assembly_types TO [' + @UserName + ']
GRANT SELECT ON sys.xml_indexes TO [' + @UserName + ']
GRANT SELECT ON sys.columns TO [' + @UserName + ']
GRANT SELECT ON sys.index_columns TO [' + @UserName + ']
GRANT SELECT ON sys.foreign_key_columns TO [' + @UserName + ']
GRANT SELECT ON sys.tables TO [' + @UserName + ']
GRANT SELECT ON sys.numbered_procedures TO [' + @UserName + ']
GRANT SELECT ON sys.database_audit_specifications TO [' + @UserName + ']
GRANT SELECT ON sys.filegroups TO [' + @UserName + ']
GRANT SELECT ON sys.stats TO [' + @UserName + ']
GRANT SELECT ON sys.sysindexes TO [' + @UserName + ']
GRANT SELECT ON sys.check_constraints TO [' + @UserName + ']
END
'

EXECUTE master.sys.sp_MSforeachdb @Command
```

You may need to remove the permissions granted after the assessment is run, in that case you may use the script provided here:

```
--Clean procedure. First log off any session using the RaaSUser
DECLARE @UserName nvarchar(255) = 'Domain\RaaSUser',
                @Command nvarchar(max)

SET @Command = 'USE [?];
        IF EXISTS(SELECT 1 FROM sys.database_principals WHERE name = '''+ @UserName +''')
                DROP USER [' + @UserName + '];
        '
EXECUTE master.sys.sp_MSforeachdb @Command

SET @Command = 'USE master;
                        DROP LOGIN [' + @UserName + ']'
EXEC sp_executesql @Command
```