Windows Server Assessment: Prerequisites and Configuration

This document explains the required steps to configure the Windows Server (Server, Security, Hyper-V, Failover Cluster and IIS) Assessment included with your Azure Log Analytics Workspace and entitled Microsoft On-Demand assessment.

There are configuration and setup tasks to be completed prior to executing the assessment setup tasks in this document. For all pre-work, follow the Getting Started with On-Demand Assessments in the Services Hub Resource Center.

Table of Contents

This document was last updated on January 25, 2020. To ensure you have the latest version of this document, check here: https://go.microsoft.com/fwlink/?linkid=865884

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

- This service is available for servers running on Windows Server 2016 or later.

Unsupported Versions

- IIS Server running with Shared Configuration (http://www.iis.net/learn/web-hosting/configuring-servers-in-the-windows-web-platform/shared-configuration_211).
- IIS Server running in workgroup (not domain joined). This scenario can be accomplished by running the collection process directly on each target server separately.

Common to Both Scenarios

- You will need a log analytics workspace
- **User account rights:**
    - A domain account with the following rights:
        - Member of the local Administrators group on all servers in the environment
        - Member of the local Administrators group of the tools machine
        - Unrestricted network access from the Tools machine to all servers

Data Collection Machine

- The **data collection machine** must be domain joined and have a Windows domain trust path to the domain joined servers to be assessed.
- Windows PowerShell 3.0 or later installed
- Log Parser 2.2 installed
- PowerShell Execution policy set to RemoteSigned
- **Data collection machine hardware:** Minimum 8 gigabytes (GB) of RAM, 2 gigahertz (GHz dual-core processor, minimum 5 GB of free disk space, plus up to 6 GB for every target server in the assessed environment during data collection.
- The **data collection machine** is used to connect to all servers and retrieve information from it, communicating over Remote Procedure Call (RPC), Server Message Block (SMB), WMI, remote registry, SQL Database, Lightweight Directory Access Protocol (LDAP) and Distributed Component Object Model (DCOM).
- The CLR version on the data collection machine should be using .NET 4.0 or greater. This can be verified by running $PSVersionTable.CLRVersion in the PowerShell prompt
- Microsoft .NET Framework 4.6.2 or newer installed and running Windows Server 2012 R2 or newer.
- The data collection machine must have the Microsoft Monitoring Agent installed and configured for one of the deployment scenarios at the beginning of this document.

Powershell Remoting

To complete the assessment with the 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 in-scope servers for the security update scan

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

Note1: *PowerShell version 2 or greater is required on target machines and comes installed by default starting with Windows Server 2008 R2.*
Note 2*: Windows Server 2016 and beyond have WinRM and PowerShellremoting enabled by default.*
The following configuration steps detailed below will only need to be implemented if the default configuration for target servers has been altered.

- 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

- Configure **WinRM / PowerShell remoting** via Group Policy (Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service)
  - "**Allow remote server management through WinRM**".
- Configure **WinRM service for automatic start** via Group Policy (Computer Configuration\Policies\Windows Settings\Security Settings\SystemServices)
  - Define **Windows Remote Management** (WS-Management) service for **Automatic startup mode**
- Configure **Inbound allow Firewall Rules:** This can be done individually in the local firewall policy of every in-scope target server or via a group policy which allow communication from the tools machine.

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 in-scope servers' organizational unit(s), 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**

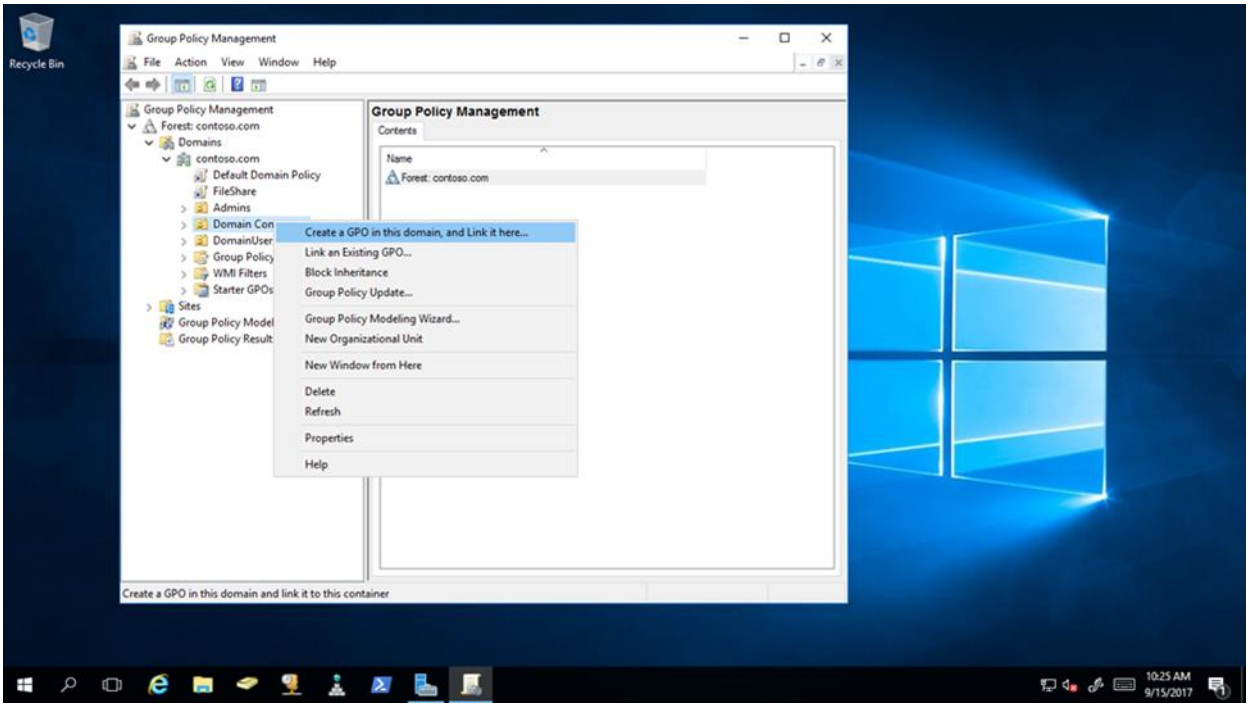
Subnet Mask . . . . . : X.X.X.X

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

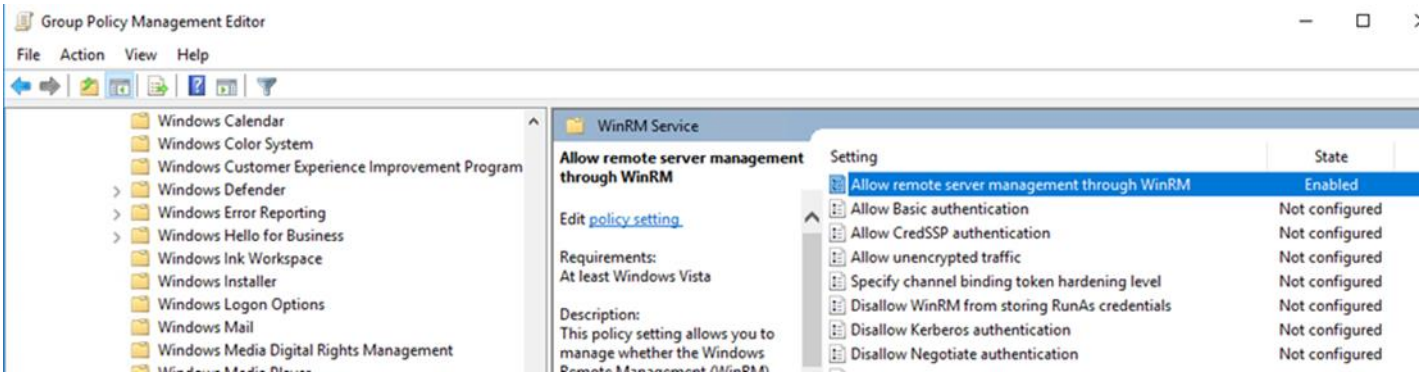
Make a note of the IPv4 address of your machine. The final step in the configuration will use this address to ensure only the data collection machine can communicate with the Windows Update Agent on the target servers.

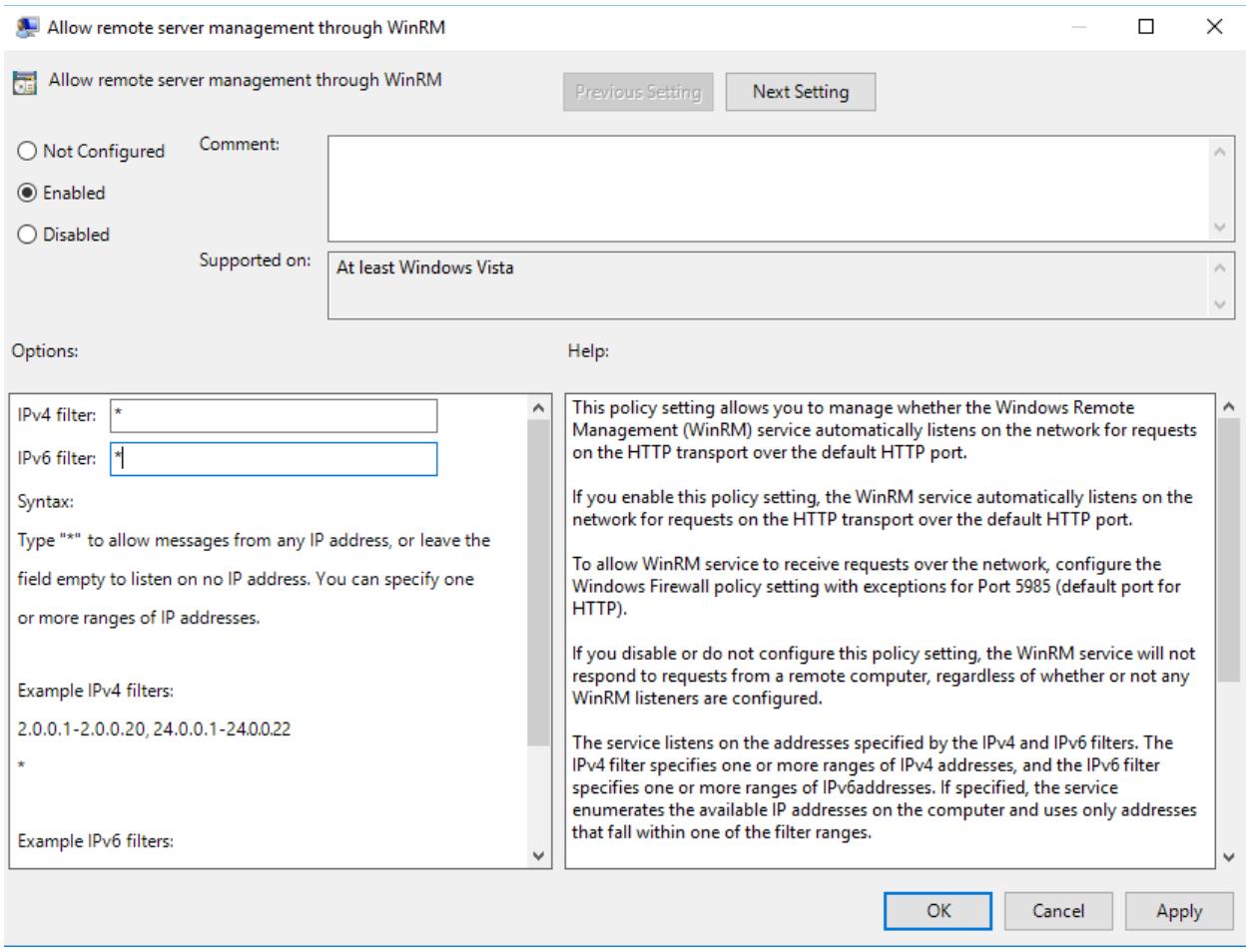
B.) Create, configure, and link a group policy object to the servers' OU(s) in each domain in the forest.

1. Create a new GPO. Make sure the GPO applies to the servers' organizational unit(s). Give the new group policy a name based on your group policy naming convention or something that identifies its purpose similar to "Windows Server Assessment"

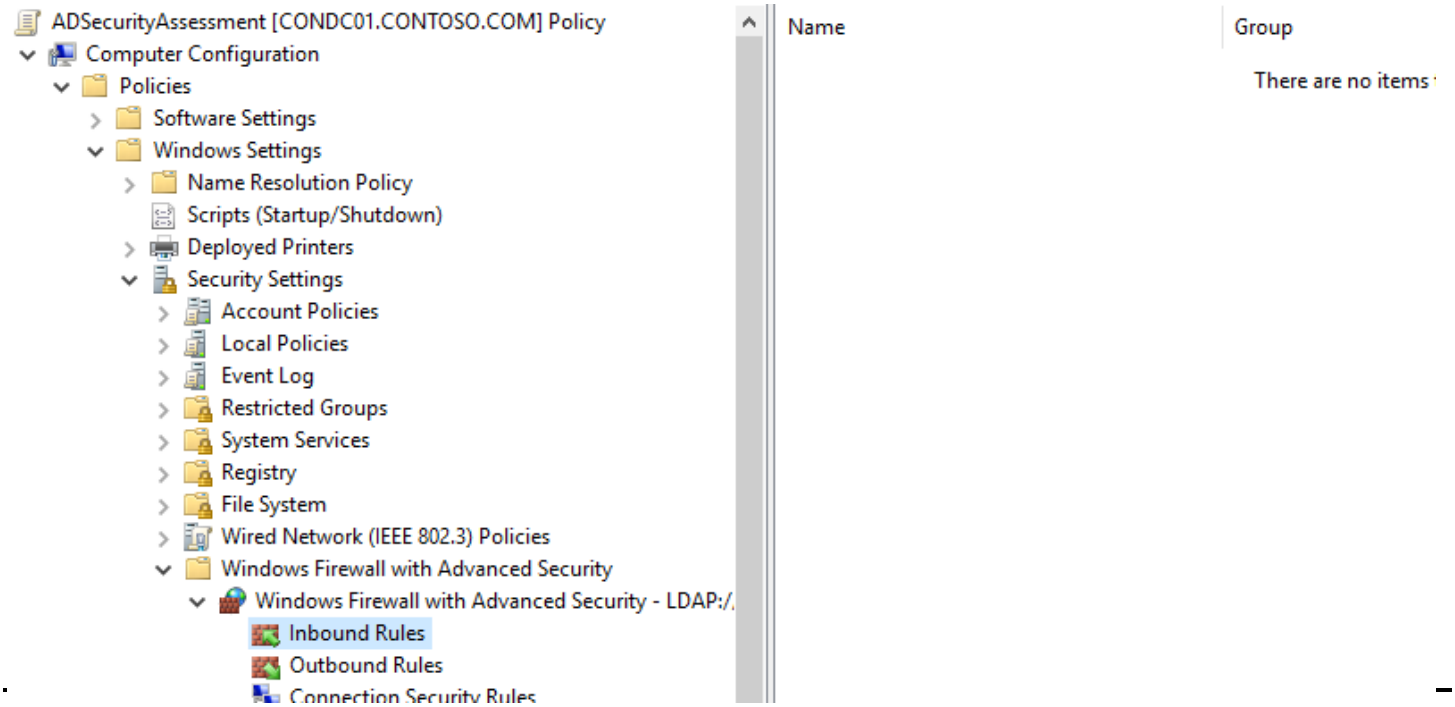


2. Within the GPO open: (Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management (WinRM)\WinRM Service). Enable "**Allow remote server management through WinRM**". You will need to specify IPv4 and IPv6 filters. ("*" will allow all inbound servers access, but specifying the IP address of the tools machine is preferred)

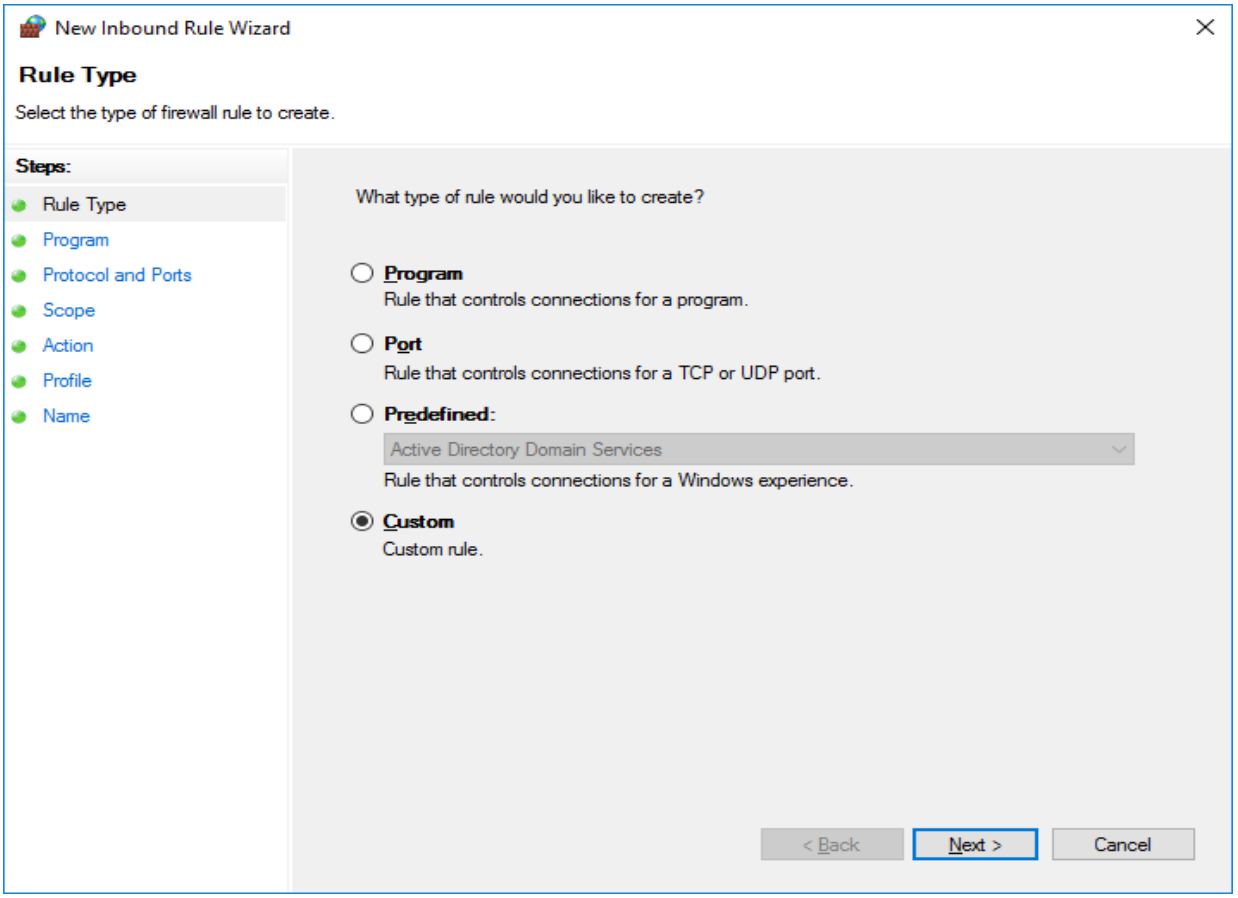




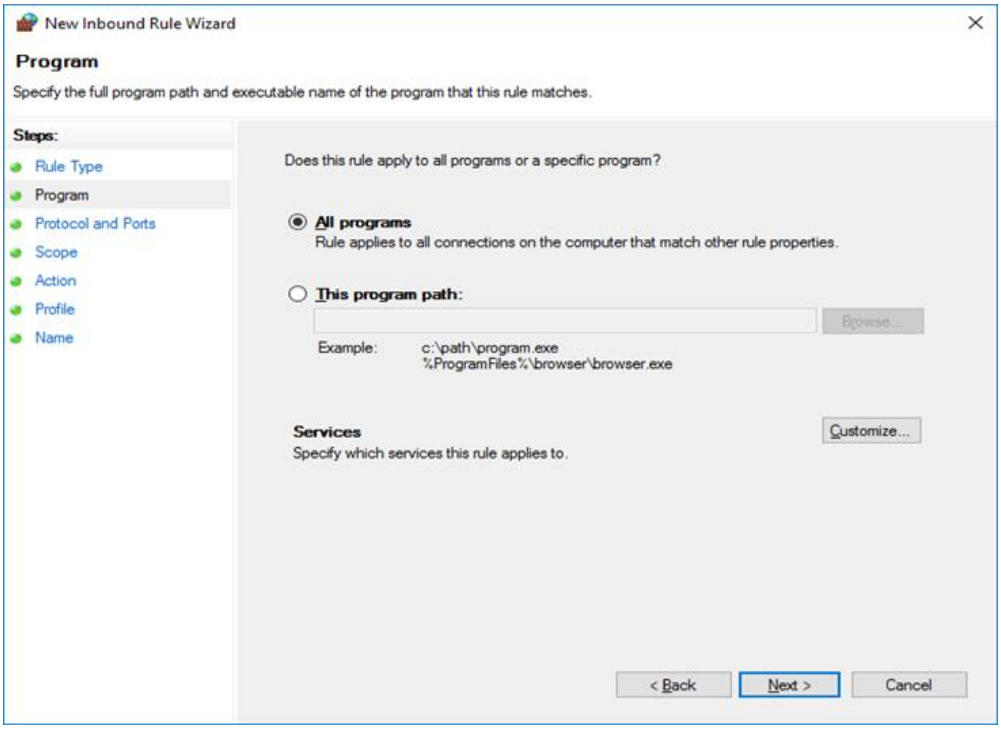
3. Create an advanced Inbound Firewall Rule to allow all network traffic from the tools machine to the target servers. This can be the applied to the same GPO that was used in step 1 above. (Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\Windows Firewall with Advanced Security –LDAP:/xxx\Inbound Rules)
4. To create the new rule, Right Click on “Inbound Rules” and select “New”



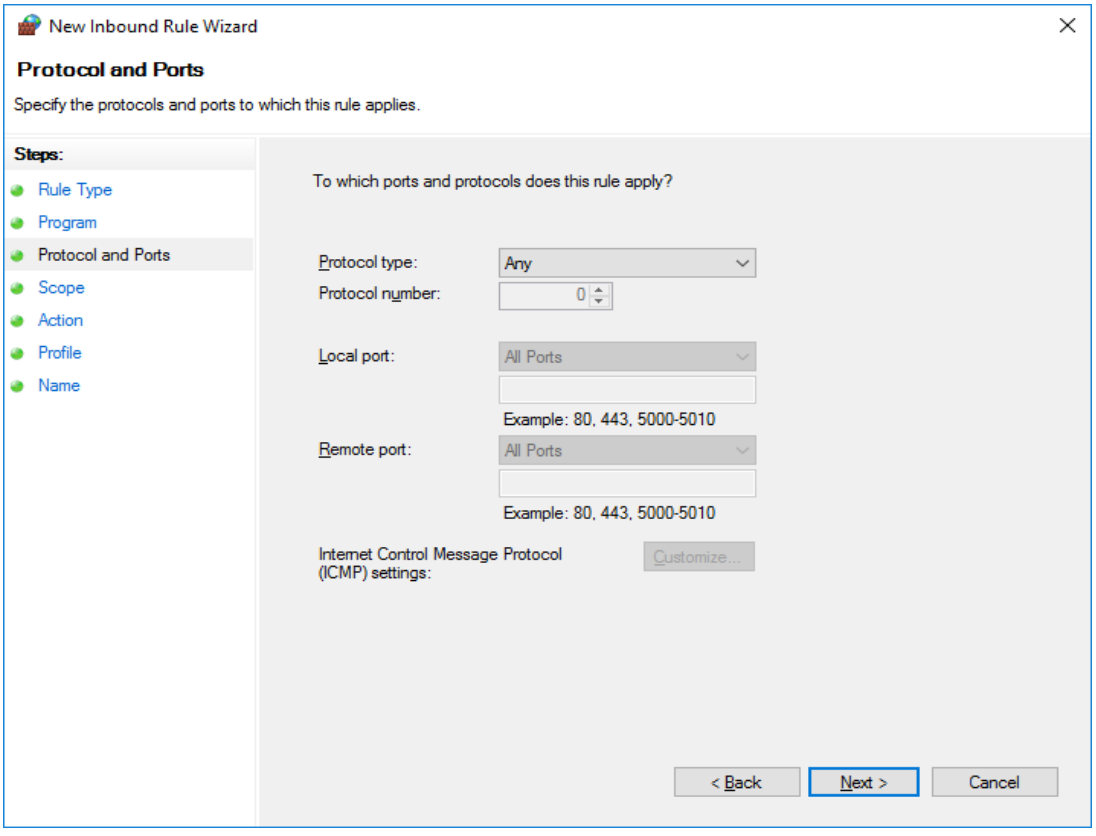
5. Create a custom rule and choose “Next”



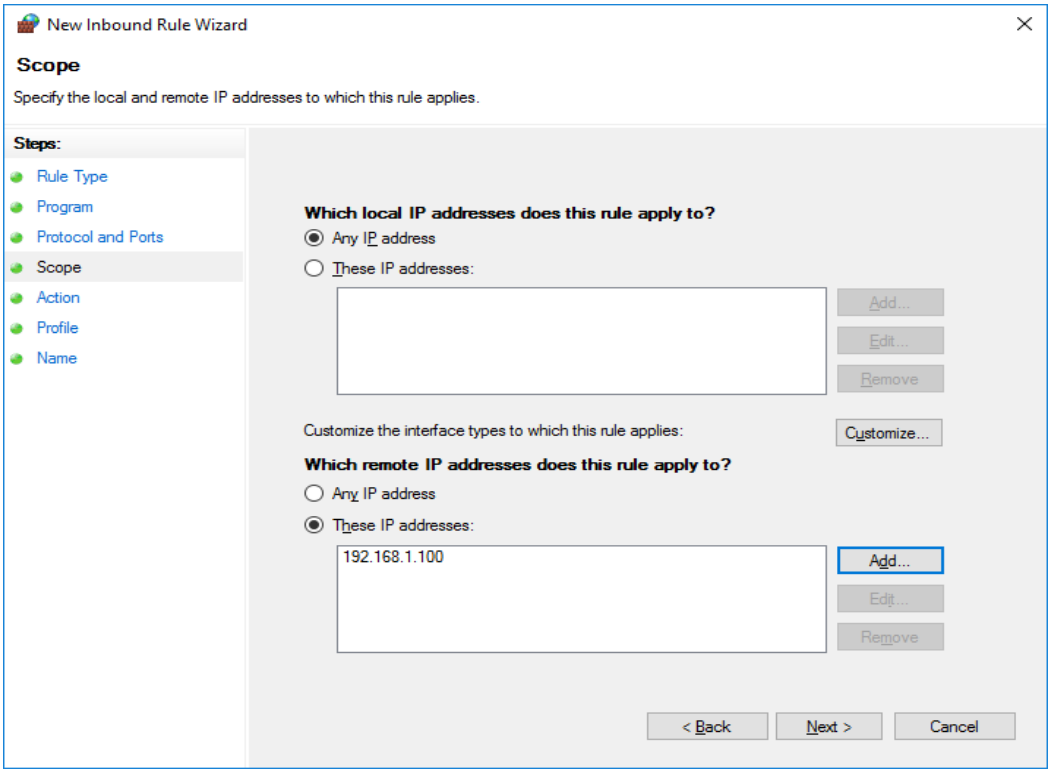
6. Allow “All programs” from the tools machine and click “Next”.



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



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



9. Choose to “Allow the connection” and click Next
10. Choose to select network profile “Domain” and click “Next”
11. Choose a name for the rule (Example: WindowsServerToolsMachine)

User Profile Service

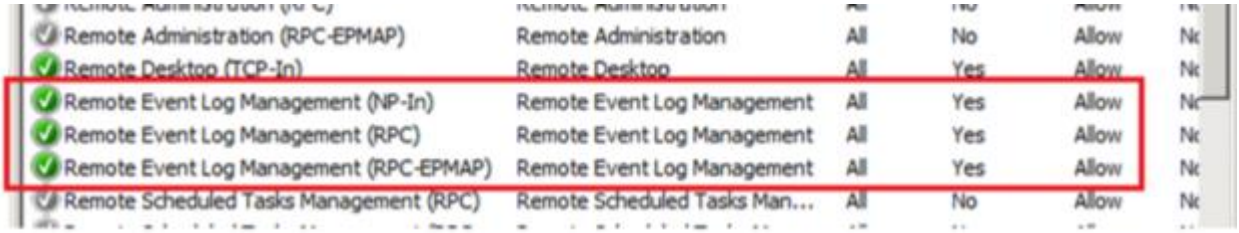
It is necessary to modify the default behavior of the User Profile Service as it relates to user logoff. Windows, by default, forcibly unloads user registry hive on logoff even if there are applications with open handles to the user registry hive. This default behavior interferes with remote Powershell 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 Management 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.

Remote Event Log Management

- **Configure the servers firewall to ensure all servers running Windows Server 2008/Windows Server 2008 R2 and later have Remote Event Log Management enabled**: Offline client might be unable to collect event log information from a Windows Server 2008/Windows Server 2008 R2 or later if **Remote Event Log Management** has not been allowed. When **Remote Management** is enabled, the rules that allow **Remote Event Log Management** are also enabled.



To test if the tool will be able to collect event log data from a Windows Server 20016/Windows Server 20016 R2 or later, you can try to connect to the Windows Server 2016or later using **eventvwr.msc**. If you are able to connect, collecting event log data is possible. If the remote connection is unsuccessful you may need to enable the Windows built -in firewall to allow **Remote Event Log Management**.

Before you can create firewall rules remotely on the server, the option **remote firewall management** must have been enabled on all Windows Server 2016 or later with the Advanced Firewall enabled. To allow Remote **Event Log Management,** create a new GPO:

Configure a GPO

1. Create a new GPO and link it to the corresponding OU for your servers.
   Within the GPO open **Computer Configuration\Policies\Windows Settings\Security Settings\Windows Firewall with Advanced Security\ Windows Firewall with Advanced Security**, right-click **Inbound Rules** and then click **New Rule**.
2. In the **New Inbound Rule Wizard**, on the **Rule Type** page, select **Predefined**. In the **rule** list, click **Remote Event Log Management,** and then click **Next**.
3. On the **Predefined Rules** page, select the **Remote Event Log Management (RPC)** rule check box, and click **Next**. *Note: the other two Remote Event Log Management rules are not required for the assessment but might be needed for Remote Event Log Management*
4. On the **Action** page, select **Allow the connection** and then click **Finish**.

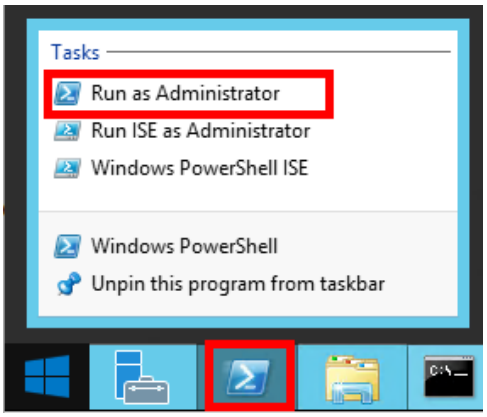
***NOTE**: Allow for this GPO to replicate and apply to all servers that are being assessed before starting data collection.*

Setting up the Windows Server Assessment

When you have finished the installation of the Microsoft Management Agent/OMS Gateway, you are ready to setup the Windows Server Assessment. There are two approaches to setting up the assessment scheduled task depending on whether the scheduled task account will be a managed service account or a user account (outlined in steps 2 and 3 below).

On the designated data collection machine, complete the following:

1. Open the Windows PowerShell command prompt as an Administrator



2. **Using a User Account:**
   Run the **Add-WindowsServerAssessmentTask -ServerName <YourServerNames> -WorkingDirectory <DirectoryPath>** command where <YourServerNames> is the semicolon separated FQDN or NetBIOS name of one or more of the servers in the environment and <*DirectoryPath*> is the path to an existing directory used to store the files created while collecting and analyzing the data from the environment.
   NOTE: If the directory does not exist, it must be created before you continue with the execution

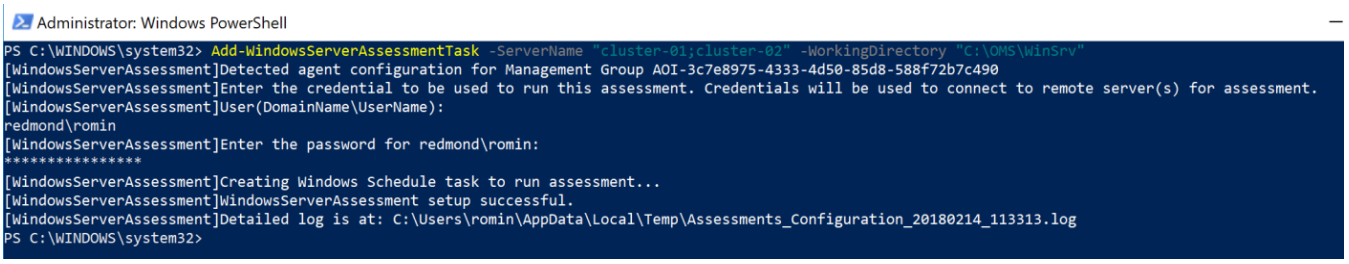
   You can also import a list of servers from a text file by using the below approach:
   PS C:\WINDOWS\system32> $Servers = Get-Content "C:\Docs\ServerList.txt"
   Add-WindowsServerAssessmentTask -ServerName $Servers -WorkingDirectory "C:\OMS\WinSrv"
   where the text file would contain a list of multiple servers that are semicolon separated for eg: "Server01;Server02;Server03".

3. **Using a Managed Service Account:**
   Managed service accounts are the preferred option for running the assessment due to their credential management and security related benefits over standard user accounts. Managed service accounts must be provisioned in Active Directory Domain Services and authorized in the environment.
   a. Follow the instructions in the provisioning KB article
   b. Authorize the account with the necessary environmental access per the User account rights section in this document. On the designated data collection machine, complete the following in an admin powershell prompt:

   Add-WindowsServerAssessmentTask -ServerName <YourServerNames> -WorkingDirectory <DirectoryPath> -ScheduledTaskUsername <MSAname> -RunWithManagedServiceAccount $True

   command where <YourServerNames> is the semicolon separated FQDN or NetBIOS name of one or more of the servers in the environment and <DirectoryPath> is the path to an existing directory used to store the files created while collecting and analyzing the data from the environment and <MSAname> is the SAM account name (ending with a $ sign) of the provisioned and authorized managed service account.

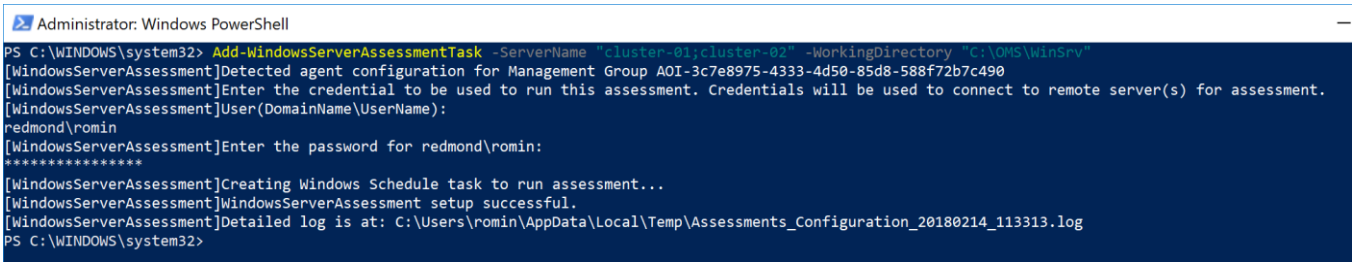
4. Provide the required user account credentials. These credentials are used to run the Windows Server Assessment.



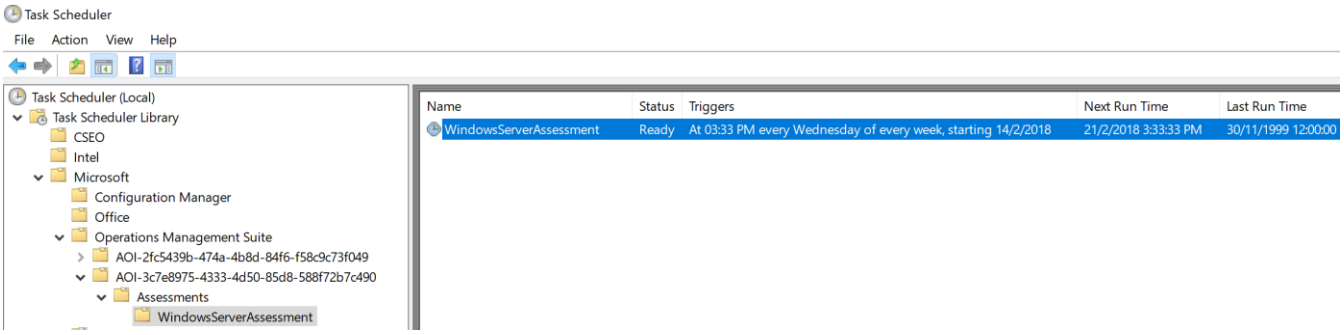
NOTE: This domain account must have all the following rights:

- Must be a local administrator on the data collection machine.
- Must be a local administrator on each of the target servers to be assessed.
- Unrestricted network access to every server to be assessed

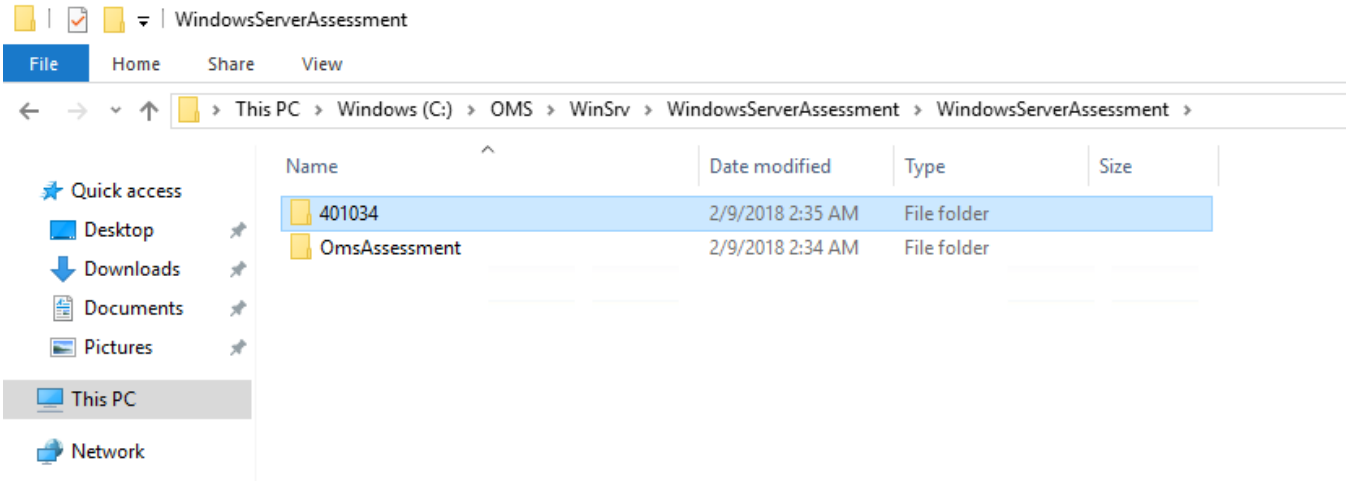
5. The script will continue with the necessary configuration. It will create a scheduled task that will trigger the data collection.



6. Data collection is triggered by the **scheduled task** named "**WindowsServerAssessment"** 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.

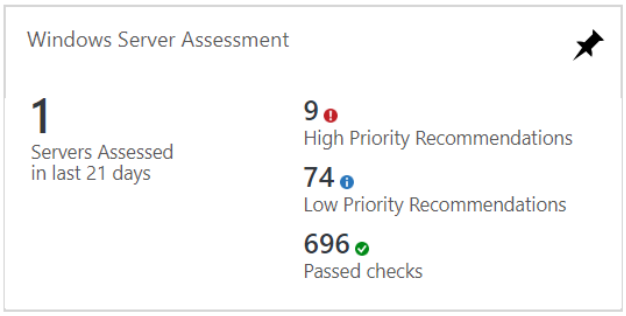


7. During collection and analysis, data is temporarily stored under the **WorkingDirectory** folder that was configured during setup, using the following structure:

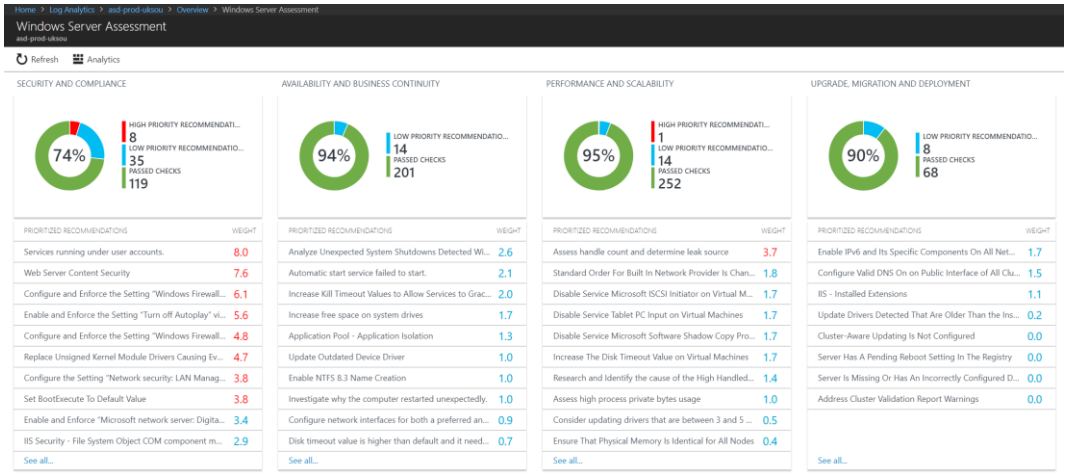


8. After data collection and analysis is completed on the tools machine, it will be submitted to your log analytics workspace depending on the scenario you have chosen:
    - **Directly** if the Data Collection Machine is connected to the Internet and configured to submit directly.
    - **Through to the OMS Gateway Server** if this option is configured, which will then submit the data to your log analytics workspace.

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



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



Appendix

Data Collection Methods

The **Windows Server Assessment in the log analytics workspace** 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. Registry Collectors
2. Xperf
3. EventLogCollector
4. Windows PowerShell
5. FileDataCollector
6. WMI
7. Nltest
8. LDAP Collectors
9. Custom C# Code
10. Validation

1. **Registry Collectors**
   Registry keys and values are read from the data collection machine and all servers. They include items such as:
   - Service information from HKLM\SYSTEM\CurrentControlSet\Services
   - Operating System information from HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion

2. **XPerf**
   Xperf is a tool that is part of the Windows Performance Toolkit that can create boot time statistics. With Xperf the boot time is evaluated and the top 10 processes that utilize disk and/or cpu most.

3. **EventLogCollector**
   Collects event logs from target machines. We mostly collect the last 7 days of different event logs.

4. **Windows PowerShell**
   Collects various information, such as:
   - BCD store boot configuration Data
   - Defragmentation rate

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

6. **Windows Management Instrumentation (WMI) Collectors**
   WMI is used to collect various information such as:
   - WIN32_Volume
     Collects information on volume settings for each server in scope. For example, the information is used to determine the system volume and drive letter, which allows the assessment to collect information on files located on the system drive.
   - Win32_Process
     Collect information on the processes running on each DC in the forest. The information provides insight on 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. We use the information to determine the amount of free space on the disk where the database or log files are located.

7. **Custom C# Code**

Collects information not captured using other collectors. The primary example here is the collection of effective user rights on the Windows servers.

8. **Validation**

Collects information not captured using other collectors. The primary example here is the collection of effective user rights on Servers.

Check computer Registry FQDN name and WMI against every target machine

```
get-wmiobject Win32_ComputerSystem -computer localhost | fl Name,Domain

Expected  Result:

Name   : <ComputerName>
Domain : dns.name
```

Check if administrative shares are available against every target machine

```
get-wmiobject WIN32_Share -computer "<ComputerName>" | ?{$_.Name -eq "C$"} | FL Name

Expected  Result: Name : C$
```

Check Scheduled Tasks access against every target machine

```
$([xml](schtasks /query /XML ONE /S "<ComputerName>")).Tasks.Task.Count

Expected  Result: > 0
```

Verifying PowerShell Remoting is enabled:

```
Enter-PSSession –Computer <ComputerName>
Expected Result: [ComputerName]: PS C:\Users\UserName\Documents>
```