Managing Down-level Windows-based Servers from Server Manager in Windows Server 2012

★★★★☆
Gaby Kaplan
10 Sep 2012 7:26 PM

The multi-server-managing Server Manager that ships in Windows Server 2012 (and in its corresponding Remote Server Administration Tools for Windows 8 package) can be used to manage remote, downlevel servers that are running the following operating systems:

- Windows Server 2008 R2 SP1 (either full server or the Server Core installation options)
- Windows Server 2008 SP2 (full server only)

To manage these servers from Server Manager in Windows Server 2012, you need to install Windows Management Framework 3.0 (WMF 3.0) and its prerequisites on the managed servers (remote servers that are running the downlevel operating systems). No configuration is required on the client (the computer on which you're running Server Manager).

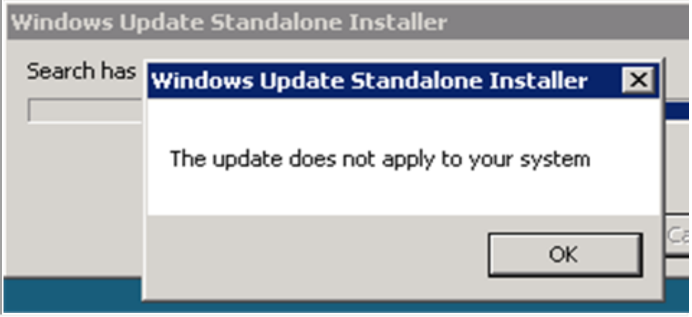
Suppose you want to manage two downlevel servers in Server Manager, and these servers do not have WMF 3.0 and its prerequisites. On the first server, the installed operating system is Windows Server 2008 R2 SP1, while on the second server, the installed operating system is Windows Server 2008 SP2. Both servers are in the same domain as the client (the computer on which you're running Server Manager).

The following table is a step-by-step guideline to ensure that downlevel servers are manageable from Server Manager. I will discuss both installation options of downlevel servers, full server and Server Core.

Managing downlevel servers that are running the full server installation option

Step	Action	Operating System-Specific Details		Comments
		WS08 R2	**WS08**	
1	I added my existing WS08 R2 & WS08 SP2 servers to Server Manager and received a	The manageability status in Server Manager: *"Online – Verify WinRM 3.0 service is*	The manageability status in Server Manager: *"Target server not accessible"*	When you added the downlevel servers, Server Manager attempted to connect to the Windows Remote Management (WinRM) port (if that fails, then the DCOM port) of the target server. If either the WinRM or DCOM connection is successful (depending on firewall settings, and whether WinRM is installed), Server Manager checks whether WMF 3.0 is installed on the target server. On the Windows Server 2008 R2 SP1 server, the

	manageability status alert.	*installed, running, and required firewall ports are open"*		WinRM connection attempt failed because either the WinRM service was not running, or WinRM inbound firewall rules were not enabled. The DCOM connection was successful, and Server Manager determined that WMF 3.0 is not installed on that server. So Server Manager displayed the manageability status of that server as “Online – Verify WinRM 3.0 service is installed, running, and required firewall ports are open.” On the Windows Server 2008 SP2 server, the connection attempt to both WinRM and DCOM ports failed (due to firewall restrictions, or no earlier WinRM version was installed), so Server Manager was unable to connect and retrieve data from the target server, and displayed manageability status as “Target server not accessible.”
2	Ensure the most current service pack for each operating system is installed.	WS08 R2 Service Pack 1 (or newer)	WS08 Service Pack 2 (or newer)	To get the major version of an installed service pack, open Command Prompt on the downlevel server, and run the following command: *wmic OS Get ServicePackMajorVersion*
3	Ensure .NET Framework 4 (or newer) is installed.	For servers that are running the full installation option of Windows Server 2008 R2 SP1, see Microsoft .NET Framework 4 (Standalone Installer). For Server Core, use Microsoft	For servers that are running the full installation option of Windows Server 2008 SP2, see Microsoft .NET Framework 4 (Standalone Installer). For Server Core, use Microsoft	You can use Knowledge base article 318785 to find out which version of .NET Framework is currently installed.

		.NET Framework 4 (Standalone Installer) for Server Core.	.NET Framework 4 (Standalone Installer) for Server Core.	
4	Install Windows Management Framework 2.0.	Not applicable	Applicable	
5	Install WMF 3.0 on downlevel servers.	Applicable	Applicable	Before this step, you’ll need to complete the last three prerequisite steps; they must be done in order. If you try to run the WMF 3.0 installer without performing the first three steps, the installation fails with an error message similar to the following:  Windows Update Standalone Installer Search has Windows Update Standalone Installer OK The update does not apply to your system
6	Install the performance update associated with Knowledge Base article 2682011 on downlevel servers.	Applicable	Applicable	To get performance counter data in the Server Manager console for servers running previous versions of Windows Server, you need to apply this update.
7	After installing all the prerequisites on a downlevel server,	Applicable	Applicable	This can be easily done by using command **winrm qc** in a command prompt window (which was run as Administrator on the downlevel server). If the WinRM service is not running on the downlevel server, or if the incoming connection attempts to WinRM service are denied by the firewall, when you add the downlevel server (with prerequisites) to the

	make sure that the WinRM service is running, and that its inbound firewall rules are enabled on each server.			Server Manager server pool, the connection attempt fails as described in step 1 in this table.
8	Enable Windows PowerShell remote management on each server.	Applicable	Applicable	To support Best Practices Analyzer (BPA) operations, Windows PowerShell remoting should also be enabled on the downlevel server. This can be done by running the command **Enable-PSremoting -Force** in a Windows PowerShell window (that was opened with administrative rights on the downlevel server). For more information about how to enable Windows PowerShell remote management on downlevel servers, see Configure Remote Management in Server Manager, or download the Server Manager Quick Reference Guide.

Managing downlevel servers that are running the Server Core installation option

WMF 3.0 can also be installed on the Server Core installation option of Windows Server 2008 R2 SP1. But it cannot be installed on the Server Core installation option of Windows Server 2008 SP2. Downlevel servers that are running the Server Core installation option of Windows Server 2008 R2 SP1 can be managed remotely from the new Server Manager. The installation process of Windows Management Framework 3.0 on Windows Server 2008 R2 SP1 Server Core is exactly the same as that for full server (already described above). However, before going through the aforementioned prerequisites, you must install or enable certain features in Server Core by using Deployment Image Servicing and Management (DISM) commands. The command names for the features are as follows:

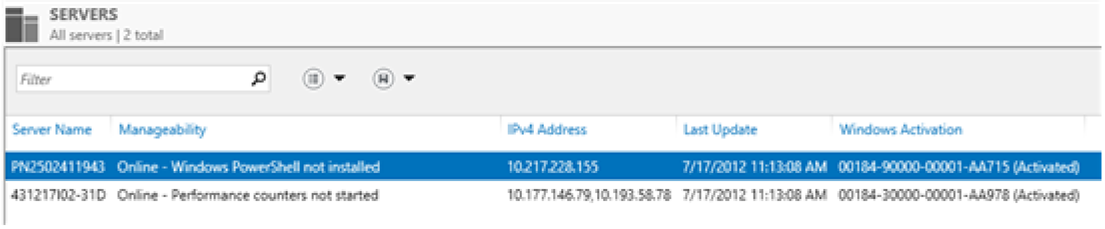
- **MicrosoftWindowsPowerShell**
- **MicrosoftWindowsPowerShell-WOW64**
- **NetFx2-ServerCore**
- **NetFx2-ServerCore-WOW64**

You can enable these features in Server Core by running the following command. Remember, DISM is case-sensitive:

```
Dism /online /enable-feature: <Feature Name>
```

After these features are installed on Windows Server 2008 R2 SP1 Server Core, the installation process of Windows Management Framework 3.0 is same as that for full server (already described above).

For the Server Core installation option of Windows Server 2008 R2 SP1, the system administrator can choose to uninstall Windows PowerShell, if it has been installed. If that's the case, we get a manageability status for the Server Core server (without Windows PowerShell) as follows:



SERVERS
All servers | 2 total

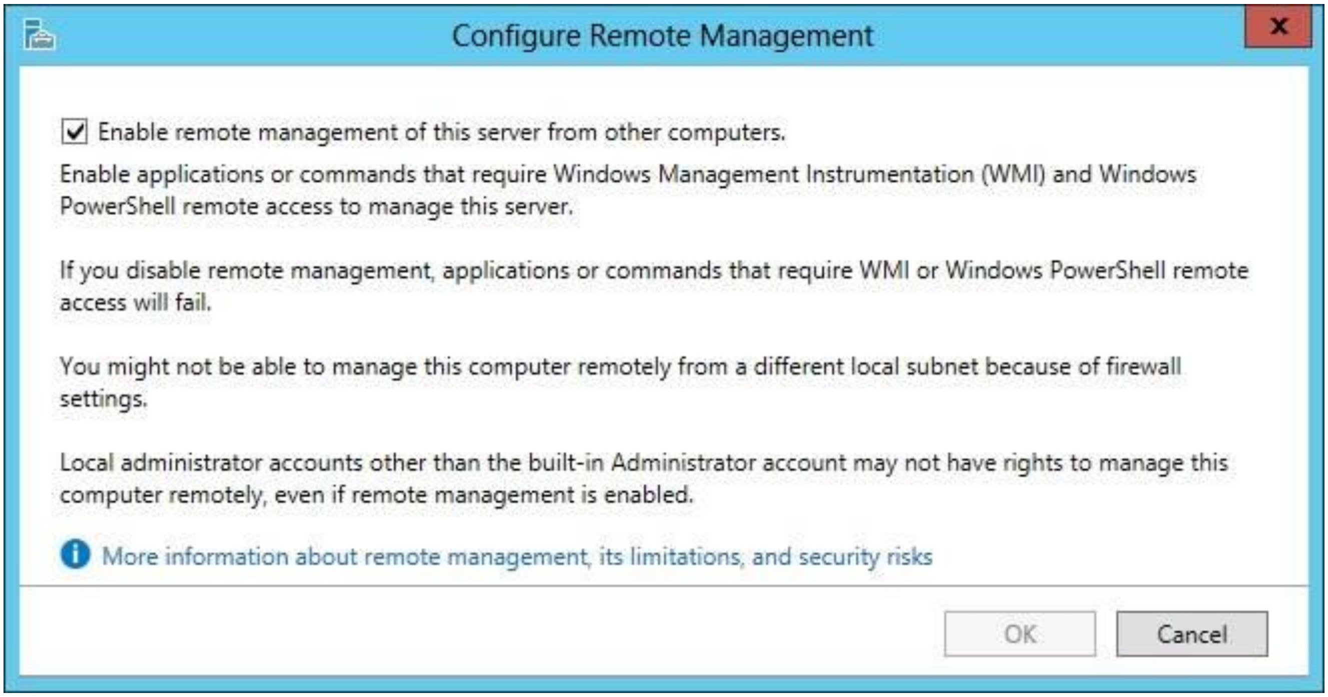
Server Name	Manageability	IPv4 Address	Last Update	Windows Activation
PN2502411943	Online - Windows PowerShell not installed	10.217.228.155	7/17/2012 11:13:08 AM	00184-90000-00001-AA715 (Activated)
431217I02-31D	Online - Performance counters not started	10.177.146.79,10.193.58.78	7/17/2012 11:13:08 AM	00184-30000-00001-AA978 (Activated)

After you have the prerequisite packages installed on your downlevel servers, if you haven't already done so, you're ready to add them to the pool of servers that you manage in the new Server Manager console. You can do this by clicking **Add Servers** on the **Manage** menu of the new console. For more information about how to add your servers, now that they're prepped for management with Server Manager, see Add Servers to Server Manager.

Enabling multi-server management in Windows Server 2012

Enable remote management

Before you can take advantage of multi-server management, you will have to enable remote management on the individual servers. To do so, open the Server Manager and select the **Local Server** container. Next, click on the **Disabled** link next to Remote Management. Doing so will cause Windows to display a dialog box that gives you the option of enabling remote management from other computers. Simply select the Enable Remote Management check box (figure 1) and click OK.



Configure Remote Management

☑ Enable remote management of this server from other computers.

Enable applications or commands that require Windows Management Instrumentation (WMI) and Windows PowerShell remote access to manage this server.

If you disable remote management, applications or commands that require WMI or Windows PowerShell remote access will fail.

You might not be able to manage this computer remotely from a different local subnet because of firewall settings.

Local administrator accounts other than the built-in Administrator account may not have rights to manage this computer remotely, even if remote management is enabled.

More information about remote management, its limitations, and security risks
OK
Cancel

Figure 1. To enable remote management on individual servers, open Server Manager, select Local Server container and click on Disabled next to Remote Management.

Using Server Manager

The most effective way to manage multiple servers through Server Manager is to create a server group. A server group is a collection of physical or virtual servers that perform the same tasks and should be managed or monitored together. To create a server group, open Server Manager and click on the **Dashboard** option, followed by the **Create a Server Group** option shown (figure 2).

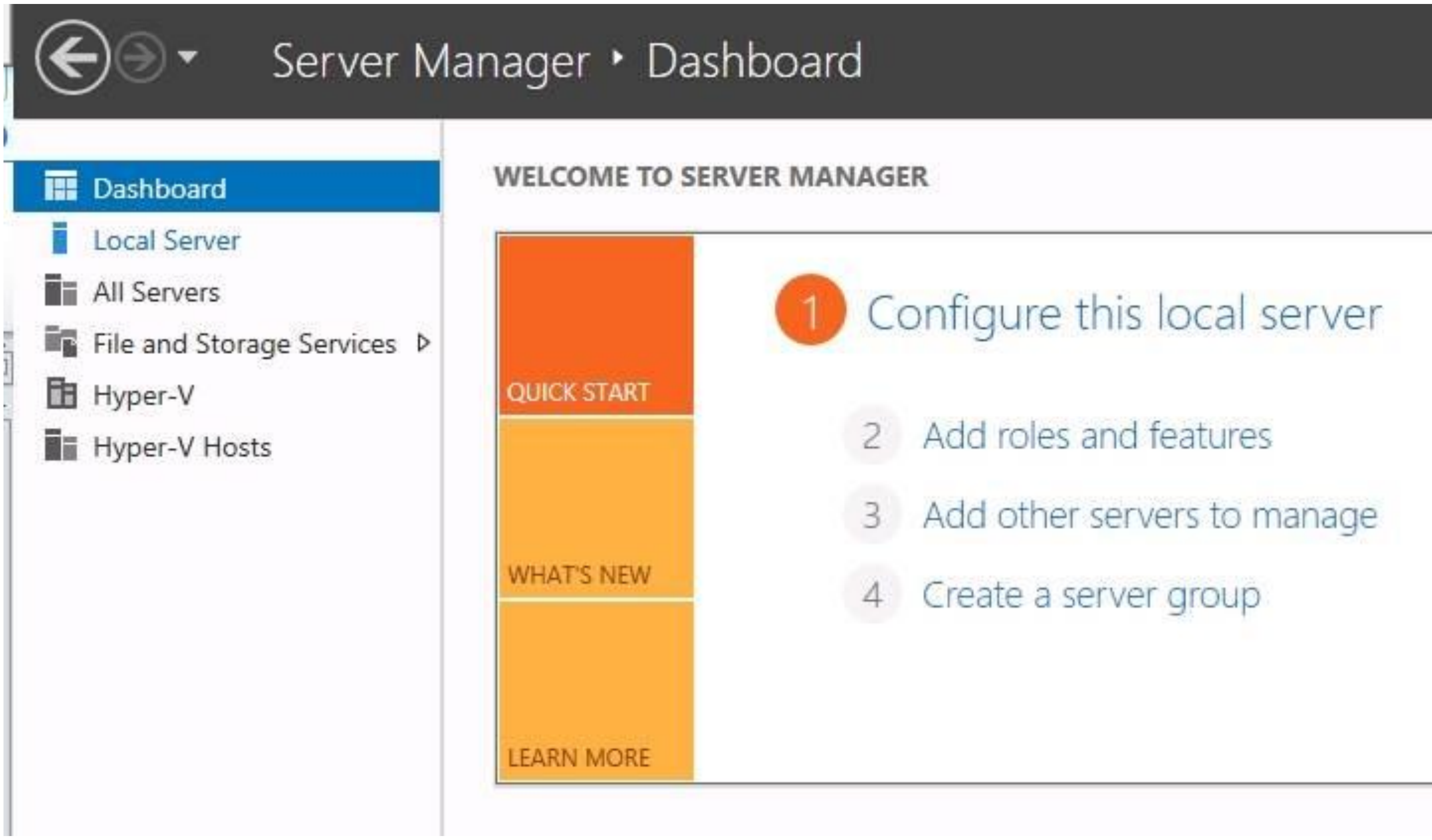
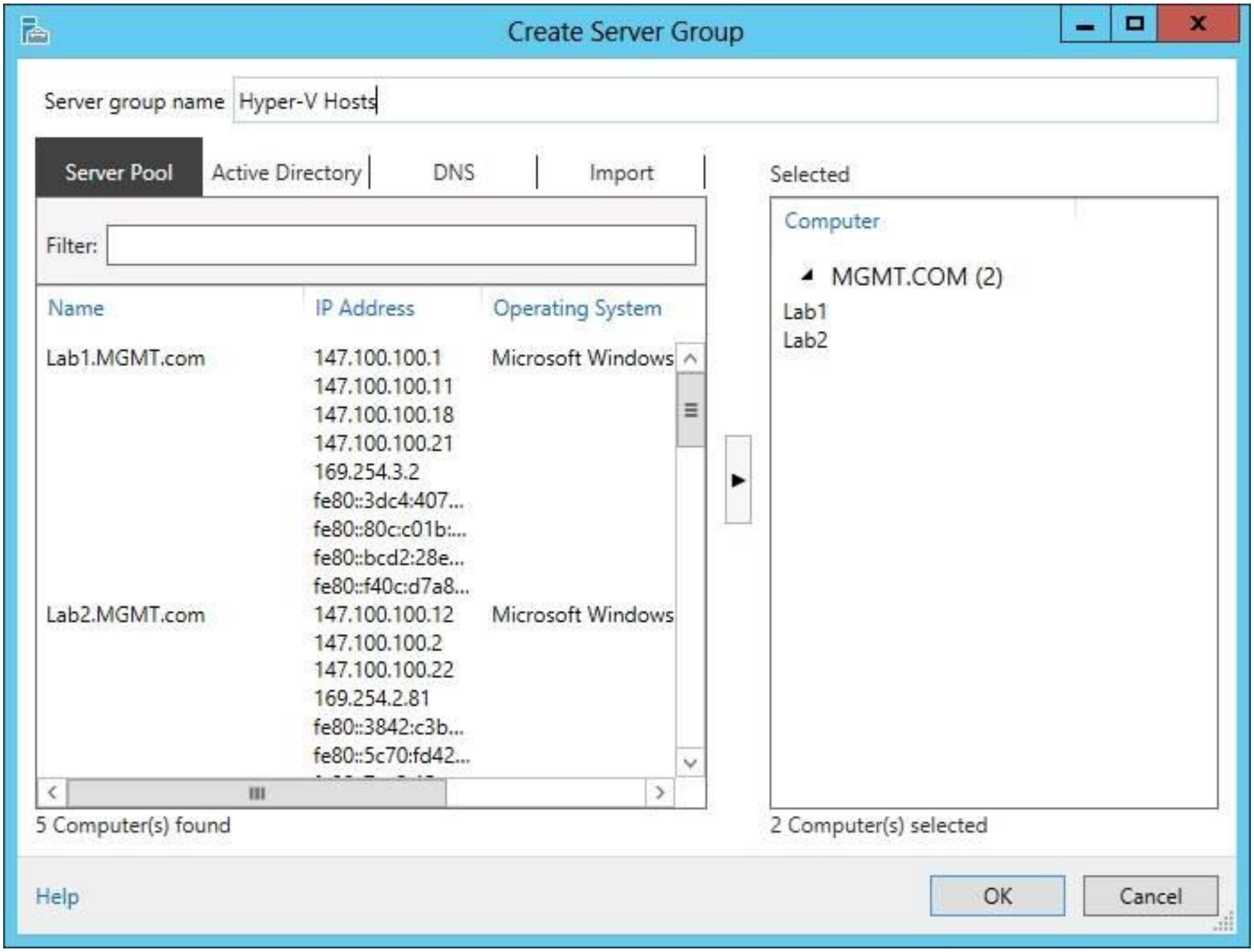

Server Manager ‣ Dashboard
Dashboard
Local Server
All Servers
File and Storage Services
Hyper-V
Hyper-V Hosts
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local server
2 Add roles and features
3 Add other servers to manage
4 Create a server group

Figure 2. To create a server group, open Server Manager, click on the Dashboard option to come to the Create a Server Group option.

At this point, Windows will display the Create Server Group dialog box. Enter a name for the server group you are creating and then specify the servers that should be included in the group. For example, I created a server group called Hyper-V hosts (figure 3). Click OK when you finish selecting the servers that should be included in the group.



Server group name: Hyper-V Hosts

Name	IP Address	Operating System
Lab1.MGMT.com	147.100.100.1 147.100.100.11 147.100.100.18 147.100.100.21 169.254.3.2 fe80::3dc4:407... fe80::80c:c01b:... fe80::bcd2:28e... fe80::f40c:d7a8...	Microsoft Windows
Lab2.MGMT.com	147.100.100.12 147.100.100.2 147.100.100.22 169.254.2.81 fe80::3842:c3b... fe80::5c70:fd42...	Microsoft Windows

5 Computer(s) found

Selected

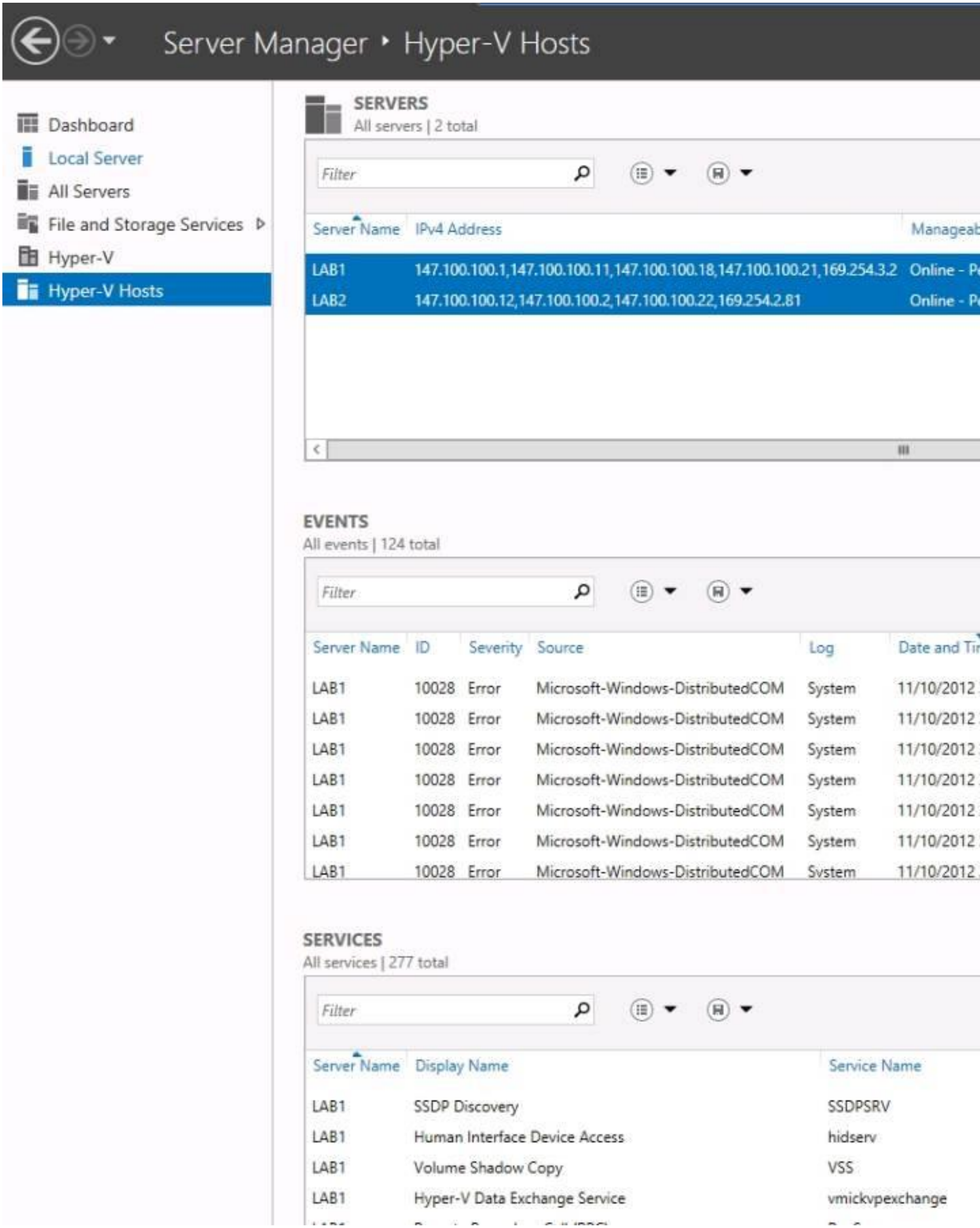
Computer

MGMT.COM (2)
Lab1
Lab2

2 Computer(s) selected

Figure 3. Enter a name for the server group and specify the servers to be included in the group.

When you finish creating the group, it will be listed in the Server Manager console. When you click on the group, Server Manager will provide you with an aggregate view of the servers that make up the group (figure 4). For example, you can see the event logs, services, performance and Best Practices Analyzer results for all of the servers in the group. You can even configure performance alerts across the various servers that make up the group.



SERVERS

All servers | 2 total

Server Name	IPv4 Address	Manageab
LAB1	147.100.100.1,147.100.100.11,147.100.100.18,147.100.100.21,169.254.3.2	Online - P
LAB2	147.100.100.12,147.100.100.2,147.100.100.22,169.254.2.81	Online - P

EVENTS

All events | 124 total

Server Name	ID	Severity	Source	Log	Date and Ti
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012
LAB1	10028	Error	Microsoft-Windows-DistributedCOM	System	11/10/2012

SERVICES

All services | 277 total

Server Name	Display Name	Service Name
LAB1	SSDP Discovery	SSDPSRV
LAB1	Human Interface Device Access	hidserv
LAB1	Volume Shadow Copy	VSS
LAB1	Hyper-V Data Exchange Service	vmickvpexchange

Figure 4. If you click on the group, Server Manager will give admins a view of servers in the group.

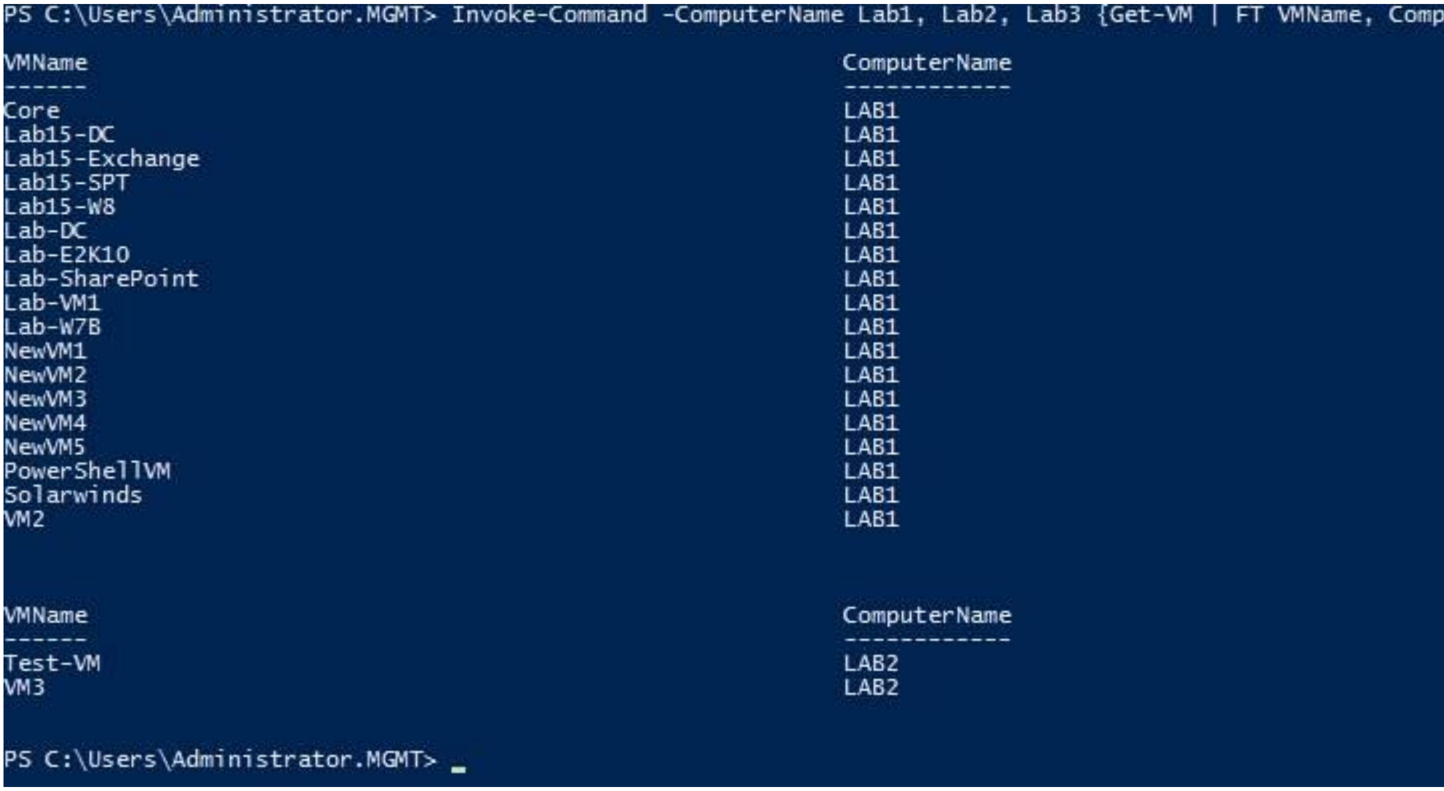
What about PowerShell?

Microsoft has long stated that Windows PowerShell is the preferred mechanism for managing Windows Server 2012. It should come as no surprise that most of the multi-server management capabilities are only exposed through PowerShell.

There are a number of different techniques that can be used to simultaneously manage multiple computers. The easiest of these techniques involves using the *Invoke-Command* cmdlet. There are three parts to the *Invoke-Command* cmdlet. First, there is the *Invoke-Command* cmdlet itself. Next, you must provide the *ComputerName* switch, followed by a list of the computers on which you wish to run the command. The last part of the command is the actual command that you want to run against the remote machines.

To see how this technique is useful, imagine that you had a number of Hyper-V servers and you wanted to see the names of the virtual machines residing on each Hyper-V host. Normally, the command you would use to create a list of virtual machines and the host server each is running on is:

```
Get-VM | FT VMName, ComputerName
```



```
PS C:\Users\Administrator.MGMT> Invoke-Command -ComputerName Lab1, Lab2, Lab3 {Get-VM | FT VMName, Comp

VMName                                                      ComputerName
------                                                      ------------
Core                                                        LAB1
Lab15-DC                                                    LAB1
Lab15-Exchange                                              LAB1
Lab15-SPT                                                   LAB1
Lab15-W8                                                    LAB1
Lab-DC                                                      LAB1
Lab-E2K10                                                   LAB1
Lab-SharePoint                                              LAB1
Lab-VM1                                                     LAB1
Lab-W7B                                                     LAB1
NewVM1                                                      LAB1
NewVM2                                                      LAB1
NewVM3                                                      LAB1
NewVM4                                                      LAB1
NewVM5                                                      LAB1
PowerShellVM                                                LAB1
Solarwinds                                                  LAB1
VM2                                                         LAB1


VMName                                                      ComputerName
------                                                      ------------
Test-VM                                                     LAB2
VM3                                                         LAB2

PS C:\Users\Administrator.MGMT> _
```

Figure 5. The PowerShell command being run against the remote servers is in brackets at the end.

The problem with this command is that it only looks at the virtual machines on the local server. If we wanted to run the command against multiple servers we would need to use the *Invoke-Command* cmdlet. To show how this works, imaging that we wanted to analyze three servers named Lab1, Lab2 and Lab3. To do so, we would use the following command:

```
Invoke-Command –ComputerName Lab1, Lab2, Lab3 {Get-VM | FT VMName,
ComputerName}
```

We started out by issuing the *Invoke-Command* cmdlet. Next, we used the *ComputerName* switch and provided the names of the servers that we wanted to run the command against. The actual command that is being run against the remote servers is encased in brackets at the end of the command (figure 5).

One important caveat when thinking about using Windows Server 2012 for multi-server management: In most cases the remote hosts must be running Windows Server 2012 in order to be remotely managed