



Automation Arrives

At Ignite this month, talk will be about automation, as the introduction of containers to Microsoft platforms has paved the way to DevOps and support for cross-platform apps.

- > **Automate** *Page 1*
- > **Automate Using Windows PowerShell** *Page 12*
- > **First Look: Docker Machine for Hyper-V** *Page 19*
- > **vSphere 6.0 vs. Hyper-V** *Page 28*

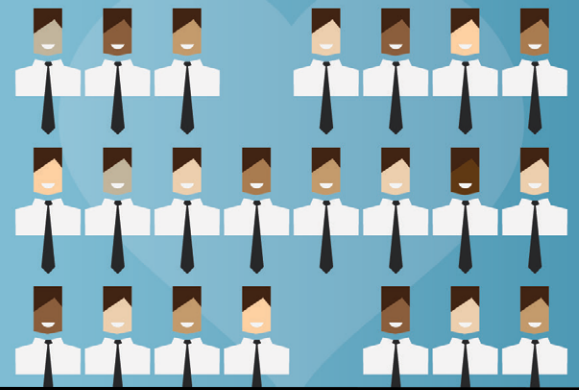
Are group management tasks blocking your arteries?

Visit Imanami
at Booth 310
for a free
diagnosis

Common Blockage Issues



A Healthy System with GroupID



Your Active Directory system may be a heart attack waiting to happen.

GroupID automatically completes mundane tasks,
so you can take care of your groups before it's too late.

With a suite of apps that automatically and dynamically manage groups in Active Directory, GroupID makes it easy to stay on top of all the changes, requests, and requirements that IT sees every day. The result:

- Distribution lists and security groups are always up to date.
- Users are empowered to perform common tasks independently.
- Security risks are significantly reduced or eliminated.

GroupID does all this for you, helping prevent potential group blockages from taking down your whole system. Visit imanami.com today for a free ebook and free trial. Or visit **Booth 310** for a free diagnosis.



GroupID
by *imanami*

*Automated Group Management
for Active Directory*

©2015 Imanami Corporation. All rights reserved.



The introduction of containers to Windows Server, Hyper-V and Microsoft Azure tied with Windows PowerShell automation paves the Microsoft path to DevOps and support for cross-platform modern apps.

BY JEFFREY SCHWARTZ

If the revelation of Microsoft's plans to develop container technology for Windows Server that would be compatible with Linux sounded like a far-out technical blueprint, you'll be surprised to hear the future is arriving. At least the early components of Microsoft's new journey to bring more automation and cross-platform support to its infrastructure software and tools are coming this month as Microsoft releases the second Technical Preview of Windows Server, code-named "vNext."

Bringing containers to the Windows platform is important because they provide lightweight services that can scale and transcend the functions of any one OS or virtual machine.

The new Windows Server preview introduces the support Microsoft promised last fall for Linux-based Docker Inc. containers in a once-unlikely partnership between Microsoft and a major open source player. The new Windows Server Technical Preview also will support recipes from Chef Software Inc. that enable automation of datacenter and desktop configuration management using the Desired State Configuration (DSC) enabled in the newest version of Windows PowerShell. Containers in Windows Server will function as lightweight runtime environments with many of the core components of a virtual machine (VM) and isolated services of an OS designed to package and execute as so-called micro-services.

Microsoft sees containers and the automation of routine processes as the key to providing the Web scale required for modern applications, while blurring the lines between what compute services are running in the on-premises datacenter and in Microsoft Azure and other providers' public clouds. This is not lost on Microsoft's rivals including Google Inc., Red Hat Inc. and VMware Inc. For example, Google just last month partnered with CoreOS Inc., operator of the default hub for all Linux containers including Docker. CoreOS, with help from a \$12 million investment from Google Ventures, is building a new platform called Tectonic, which will host the Google-developed and now open source Kubernetes. Tectonic aims to provide a fault-tolerant platform to deliver containers that provide APIs that manage such infrastructure services as search, load balancing and service discovery (Microsoft has also said it will support Kubernetes).

Shift in Computing

Bringing containers to the Windows platform is important because they provide lightweight services that can scale and transcend the functions of any one OS or VM. The quest to deliver automation is coming together in lockstep with the move to hybrid cloud architectures. Enabling both is critical so these new modern applications can scale.

"Right now, the biggest technology shift in the cloud is a rapid evolution from simple virtual machine (VM) hosting toward containerization, an incredibly efficient, portable, and lightweight cloud technology that saves significant operating system overhead costs and dramatically improves application time-to-market," wrote Mark



“We’re seeing orchestration technologies emerge for VMs and containers.”

*Mark Russinovich,
Microsoft Azure CTO*

Russinovich, the CTO for Azure and one of the most knowledgeable engineers on the internals of the Windows kernel, in a March blog post (bit.ly/1FsgXbB). “On top of that, we’re seeing orchestration technologies emerge for VMs and containers.”

This is all beginning to “blur the lines between traditional [Infrastructure-as-a-Service] (IaaS) and [Platform-as-a-Service] (PaaS) approaches to cloud computing, making it easier for customers to scale quickly and easily without sacrificing security or control,” Russinovich explained. Russinovich noted this is a major departure from the way developers have historically developed applications and maintained them in separate siloes. While each app could communicate across a network from a client, they couldn’t easily share with other apps. Now, developers are building applications that can be built, tested, deployed and monitored in a more “holistic” way.

“Everyone is driving toward a more simplified streamlined, deployment of resources,” says IDC analyst Al Gillen. “The nice thing is if you can get to a model where you use a really thin OS you can have a great deal of consistency from one image to another. Which means that you don’t have a lot of management overhead and you don’t have a lot of variable configurations that cost a lot of money to support. So the Holy Grail is to get to a one-image operating scenario, and the advent of the thin OS promises to remove the complexity from all the layers of the stack.”

Enabling DevOps

The benefit of getting to a single image for all of your workloads is that the management of the OS is minimized and potentially can go away because you’re only going to manage that one single image and only have to qualify the patches once because they’re insulated in the containers above the OS, he adds. “You need to have more automation simply because you need everything to just take care of itself. You can’t really do it manually, or you end up with this massive number of installations.” Getting to this stage is much more difficult than it may sound, Gillen warns. “It works great for net-new applications but for all the old stuff, it’s a lot harder to get a new thin OS implementation simply because you have to do a lot of code migration to get the old apps to run in that environment.”



“Microsoft has over 100 new configuration points that they added to their configuration management system.”

*Adam Edwards,
General Manager, Chef*

Mark Bowker, a senior analyst at Enterprise Strategy Group Inc., agrees, noting containers will allow developers to build applications that are more portable. The growth of environments that use containers also could benefit shops that are moving to a DevOps model, which involves processes where developers and IT pros work together so applications can become more infrastructure-aware. “So if I’m a developer and I need more capacity of some sorts, which could be compute or maybe it’s tied to networking, I may be able to have some PowerShell capabilities to include in my application and have that incorporated now instead of having to break through walls inside my IT org to get that capacity,” Bowker says.

Cooking Automation with Windows PowerShell

Indeed, Windows PowerShell and the DSC built into the newest version of the Microsoft Windows scripting tool promise to play a key role as containers become more mainstream. The inventor of Windows PowerShell Jeffrey Snover, a Microsoft distinguished engineer for Windows Server, has invested much of his efforts of late with the open source community, notably Chef, which has its own cross-platform architecture for automation using what the company calls Chef recipes (see a [Q&A with Snover on p. 6](#)).

DSC is a distributed, heterogeneous configuration management platform, as Snover described it to attendees at last month’s Chef-Conf in Santa Clara. “We work with all of the partners in the industry so if they have anything that needs to be configured, they write a [DSC] resource, and then we work with the configuration management tool industry to get their tools to consume the [DSC] management APIs, and when they do that, they can manage every element that plugs into the [DSC] platform.”

Adam Edwards, a general manager for the Windows vertical at Chef, explains the integration between DSC and Chef Server. “We essentially now use it as a pass-through where you can offer Chef recipes that will talk to their configuration management system and it exposes their configuration points as if they are part of Chef,” Edwards says. “Essentially we multiply. Microsoft has over 100 new configuration points that they’ve added to their configuration management system. We work closely with them, they’ve taken our feedback on some of the design of that configuration system so that

“The added benefit is that Docker is helping ops teams streamline the development process.”

– David Messina,
vice president of
marketing at Docker

we can make it work better with Chef. We actually work closely with their engineering team on that.”

Chef and Microsoft have worked closely together over the past year, according to Edwards. “Microsoft has been trying to make Windows more automatable. They wanted feedback from us and they’ve taken our feedback and as a result, we got pretty direct support for configuring Windows,” he adds.

These configuration capabilities enabled by DSC, tied to the release of the new Microsoft container technology, will bring about new requirements for automation, Russinovich tells Redmond. “PowerShell is great as a scripting tool for automation,” Russinovich says. “Our Azure Automation service executes user-supplied PowerShell scripts for imperative workflows. However, we believe that declarative, rather than imperative, automation allows the infrastructure to take over activities that users would otherwise have to manage themselves. For example, declaring that there should be three instances of a micro-service allows an orchestrator to take over the job of deploying those instances, determining on which servers to place them, and repairing or moving them when there are failures. Such tasks are very hard to script and lead to fragile systems that are hard to diagnose and change.”

David Messina, vice president of marketing at Docker, says the automation enabled by its container platform is bringing operational benefits to systems administrators by simplifying the deployment of distributed applications on any type of infrastructure, whether on-premises or in the cloud. “The added benefit is that Docker is helping ops teams streamline the development process,” Messina says. “We see operations teams setting up self-service capabilities for their development teams to rapidly build, ship, and run their apps and move them through all the stages of the application lifecycle seamlessly. Today this is true for Linux and moving forward, with all the great work that Microsoft is doing, it will be true for Windows Server, as well.”

Simplified Operational Model

Bringing containers to Windows Server, Hyper-V and Azure will usher in new capabilities that will enable automation of many mundane processes now performed by IT pros.



Jeffrey Snover

Q&A:

Jeffrey Snover Touts Cooking Up PowerShell Automation with Chef

The inventor of the popular Microsoft scripting engine discusses dev and ops people coming together and enabling automation.

By John K. Waters

As Microsoft and Chef Software Inc. forge ahead with their partnership, the two companies together hope to further enable cross-platform and cross-cloud automation in a way that brings development and operations teams together. That's now possible thanks to compatibility between Windows PowerShell and the Chef Automation Platform and the arrival of Windows containers.

The inventor of Windows PowerShell, Jeffrey Snover is a big fan of Chef and made that known at the company's annual developer conference last month. Snover, a Microsoft distinguished engineer and the lead architect for the Windows Server division, demonstrated his famous object-based, distributed automation engine, scripting language, and command-line shell to an audience of Chef users at the annual ChefConf gathering in San Clara, Calif., last month.

Steven Murawski, a Windows Server administrator at Chef who declared, "PowerShell has done wonders to bring a consistent command-line and automation story to the Windows platform," introduced Snover. "As Chef embraces PowerShell, our capabilities in the Microsoft stack continue to grow," Murawski said.

In his "PowerShell from the Ground Up" talk, Snover demonstrated a distinctly developer-focused capability: the new ability of Windows PowerShell 5.0 to define classes. "Keep in mind that we are doing this over the course of time and we're implementing just enough classes so you can write DSC resources very simply," Snover explained.

Although Snover admitted that Windows PowerShell is primarily a tool for IT pros and operations teams, his goal from the outset was to create a tool that would connect operators and developers. "We've got a term for that now—it's called DevOps."

Redmond magazine Contributing Editor John K. Waters talked with Snover after his presentation.

"PowerShell has done wonders to bring a consistent command-line and automation story to the Windows platform."

— Steven Murawski,
a Windows Server
administrator

The DevOps guys are definitely on the right track. But they represent an early vanguard of the industry.

Why is Microsoft partnering with Chef?

We are a platform company, and we want to enable companies like Chef to manage everything on the Windows ecosystem. We want to make it possible for a customer to go to these companies and say, hey, do you support DSC, and if they say yes, then they don't have to worry about deploying and managing configuration data for software services and managing the environment in which those services run. And Chef has been great.

You advocated strongly for automation 13 years ago in your Monad Manifesto (bit.ly/1qVIXjC), in which you articulated a long-term vision and started a development effort that became PowerShell. The Azure PowerShell module currently provides about 20 Azure Automation cmdlets, so it's clear that Microsoft is on board. But what about the rest of the industry? Are enterprises on the right track today? Do people get it now?

No, I'm afraid not. Just think about how many people are still on Windows Server 2003 clicking Next. The DevOps guys are definitely on the right track. But they represent an early vanguard of the industry. And when you talk to them, they'll tell you that they are, in fact, the vanguard in their organizations. So with DevOps, what we're seeing is really the vanguard of the vanguard.

Everyone seems to be unveiling a container strategy these days. Just last year Microsoft and containerization superstar Docker jointly announced plans to develop a native Windows Server based on the Docker engine (bit.ly/134r6Me). How do containers fit into the emerging automation picture?

OK, now we're talking about the vanguard of the vanguard of the vanguard. Exciting stuff, but anyone who tells you that they know how containers are going to play themselves out over the course of the next three to five years is making it up. We're all on a journey. It's amazing technology, and it's going to change things, but how it's going to do that, no one knows.

How might containerization change things?

One of the things I see is that a number of the configuration tasks currently done by operators late in the process are going to move forward and be done by developers as part of the build process.

Consumers may love drama, but drama and the enterprise don't go together.

In fact, you talked during your presentation about new support in PowerShell for developers. Is that part of an evolution or a response to a particular demand?

It's a logical next step. Along with the things I mentioned here, it's being integrated with Visual Studio, so you can now edit your PowerShell scripts in Visual Studio.

Which is perfectly in line with the hopes you expressed in the Manifesto, right?

Yes! If you look at the Manifesto, I talked about the problem being that the technology was keeping these two different groups of people [dev and ops] from integrating well together. We wanted one technology that made it easier. And, indeed, there were some things in PowerShell that developers found off-putting, and we've been addressing that.

What's the coolest thing you see on the horizon for PowerShell, beyond PowerShell 5.0?

Well, because we're now on this Agile footing, and a monthly release cadence, it's very difficult to have those big moments. But that's OK: Consumers may love drama, but drama and the enterprise don't go together. What you'll see is a steady pursuit of the core principles: applying IT to advance business goals, getting people to work together, making it a safer environment, and being able to secure systems.

PowerShell fans—of which there are bazillions—love its command-line interface. But there are critics out there who argue that wizard- and GUI-based approaches are more efficient and modern. How do you respond to that criticism?

Look, if you're not adding value through differentiated IT—if you're just using a GUI and doing click-next, click-next, click-next—you're probably in trouble. Those standardized values are going to be offered through the cloud at dramatically lower prices and higher service-level agreements. In that scenario, you're just not needed. But if you can figure out how to take those standardized offerings and turn them to your business advantage, there's a huge amount of value that one could add. Ask yourself, how can we take these technologies and better integrate them into our processes and use them to better advance our business agenda? There, I see a ton of value.

**Microsoft says
the new Windows
Server will support
a native Docker
client container.**

Russinovich says containers specifically enable “the decomposition of applications into containerized micro-services. Running in containers creates a stable environment for micro-services instances, which in turn results in a simplified operational model that’s amenable to automation. Such an architecture enables an orchestrator to automatically scale up and down micro-services, as well as to track their health and perform automated recovery.”

Containers also enable automation in that they allow for the sharing of kernel and critical system components. This provides quick start-up times and lower resource overhead, which make them suitable for cross-platform, services and application communications. The new Windows Server will enable sharing, publishing and shipping of containers “anywhere the next wave of Windows Server is running,” Microsoft Corporate VP Jason Zander explained last fall. “With this new technology millions of Windows developers familiar with technologies such as .NET, ASP.NET, PowerShell and more will be able to leverage container technology. No longer will developers have to choose between the advantages of containers and using Windows Server technologies.”

Docker Integration

Extending beyond Windows, though, Microsoft says the new Windows Server will support a native Docker client container. The Docker Engine for Windows Server, developed under the auspices of the Docker open source project, will be able to run Linux container-based applications available in the Docker Hub where Windows Server container images will also be available.

Meanwhile, the introduction of Hyper-V containers will offer a deployment option to running applications on Windows Server, which in the new version will also support containers. The Hyper-V containers will support the same development and management tools as those designed for Windows Server Containers, wrote Mike Neil, Microsoft general manager for Windows Server, in a blog post last month (bit.ly/1OauzuX). Moreover, he said developers don’t need to modify applications built for Windows Server Containers in order to run in Hyper-V containers.

“Nano Server provides just the components you need—nothing else, meaning smaller server images, which reduces deployment times, decreases network bandwidth consumption, and improves uptime and security.”

Mike Neil, General Manager for Windows Server, Microsoft

The addition of Hyper-V containers will offer a deployment option that offers extended isolation utilizing the attributes of not just the Windows OS, but Hyper-V virtualization, according to Neil.

“Virtualization has historically provided a valuable level of isolation that enables these scenarios, but there is now opportunity to blend the efficiency and density of the container model with the right level of isolation,” Neil said. “Microsoft will now offer containers with a new level of isolation previously reserved only for fully dedicated physical or virtual machines, while maintaining an agile and efficient experience with full Docker cross-platform integration. Through this new first-of-its-kind offering, Hyper-V Containers will ensure code running in one container remains isolated and cannot impact the host operating system or other containers running on the same host.”

Microsoft MVP Aidan Finn believes running containers in Hyper-V appears to be a more secure option than using the OS. “Hyper-V provides secure isolation for running each container, using the security of the hypervisor to create a boundary between each container,” he wrote in a blog post last month (bit.ly/1y9NYbu). “How this is accomplished has not been discussed publicly yet. We do know that Hyper-V containers will share the same management as Windows Server containers and that applications will be compatible with both.”

Neil pointed out that Microsoft also has made it easier to deploy to the newest Docker engine released earlier this year using Azure extensions to setup a Docker host on Azure Linux VMs and to deploy Docker-Managed VMs from the Azure marketplace.

Windows Nano Server

Container support is also coming to a scaled-down Nano Server for modern application scenarios where Hyper-V and Windows Server would be overkill, Neil said. Neil described the new Nano Server as “a minimal footprint installation option of Windows Server that is highly optimized for the cloud, including containers. Nano Server provides just the components you need—nothing else, meaning smaller server images, which reduces deployment times, decreases network bandwidth consumption, and improves uptime and security.



Automation is certainly not a new concept for IT, of course.

This small footprint makes Nano Server an ideal complement for Windows Server Containers and Hyper-V Containers, as well as other cloud-optimized scenarios.”

Finn warns the Nano Server is a bare-bones system. “The OS is beyond Server Core,” he says. “It’s not just Windows without the UI; it is Windows without the I [interface]. There is no log-on prompt and no remote desktop. This is a headless server installation option.”

Future Role of Automation

Automation is certainly not a new concept for IT, of course. It’s a matter of bringing it to new levels. In the end, though, the arrival of thin OSes and micro-services is sure to shed new light on automation. It’s a question of whether the drive for automation will usher in these new technologies or will be a mere outgrowth of them.

“It’s safe to say the concept of these thin OSes go hand in hand with the container strategy, which then enables the automation, and certainly you want to get to a model where you have more automation and it’s made possible by this,” IDC’s Gillen says. “As organizations wind up with more implementations of thin OSes, you need to have more automation simply because you need everything to just take care of itself. You can’t really do it manually or you end up with this massive number of installations.” **R**

Jeffrey Schwartz is editor in chief of Redmond.



Automate Using Windows PowerShell

New capabilities make automating IT processes easier and worthwhile. BY ADAM BERTRAM

It pros these days tend to wear many hats and handle seemingly random tasks on a daily basis. At any given time, you may find yourself dealing with an angry user about an Outlook problem, while bringing up a new virtual server and creating a few Active Directory user accounts for a new line-of-business application. IT operations people are the ultimate multitaskers.

What makes someone a “go-to” IT person?

Some are more successful than others in handling the multifaceted requirements of IT shops today. What makes someone a “go-to” IT person? It’s those who excel at getting work done who typically earn that reputation. They’re smart, work efficiently and always have enough time to help not only users, but also assist others in the IT organization. Yet they’re always able to leave at 5 p.m., while others are struggling to get work done. How do they do it? For many it’s their ability to bring automation to their tasks. And a growing number are bringing new levels of automation thanks to capabilities in the latest release of the Microsoft Windows PowerShell scripting environment.

The Windows scripting tool has gained some significant adoption across the IT infrastructure landscape since Microsoft first released

DSC is a critical new addition to Windows PowerShell because it allows IT pros to configure a server once and ensure it stays that way.

it in 2006. And now Microsoft is stepping up its emphasis of Windows PowerShell. At this month's Ignite conference in Chicago (the event replacing TechEd), Microsoft will present 18 technical sessions devoted to Windows PowerShell. That's a major increase over last year's TechEd North America in Houston when there were 10 sessions, and just four the year before in New Orleans. In 2012 Microsoft didn't hold any sessions covering Windows PowerShell.

IT pros who use the Windows PowerShell console and associated tools can replace the traditional command prompt to automate routines and other checks associated with PC and infrastructure management.

The Windows PowerShell Vision

To understand the momentum behind Windows PowerShell, it's important to understand its origins. Windows PowerShell is a Windows console replacement/scripting language originally concocted in 2002 by Jeffrey Snover, a distinguished engineer at Microsoft. Snover is now known as the father of Windows PowerShell and is the lead architect of Windows Server and System Center at Microsoft. Ironically, though as it came to pass fortuitously, Snover came to Microsoft in 1999 from Digital Equipment Corp. with a background in Unix, in which automation was easy due to its inherent architecture of using simple text files for configuration. Snover realized based on the way it was originally designed, Windows wasn't capable of the kind of automation Unix admins were leveraging. Administrators, myself included, at the time tried their best to automate Windows processes in VBScript. If you've ever written more than a few hundred lines in VBScript you'll understand why I say, "tried their best."

Snover believed Windows tasks could be as easy to automate as Unix using the ultimately popular Microsoft .NET Framework. He envisioned a platform that embraced standardization across OSes, allowed IT pros (not just developers) to leverage the power of the .NET Framework through objects and much more.

Initially, his vision fell on deaf ears. Not only was he not able to convince Microsoft of the virtues of automating Windows tasks, his efforts landed him a demotion. Undeterred, he finally prevailed with the release of Windows PowerShell 1.0 in 2006 with Windows XP SP2, Windows Server 2003 and Windows Vista.

Dozens, if not hundreds, of products now rely on Windows PowerShell as a management and automation platform.

From Rags to Riches

So much has changed since the first version of Windows PowerShell. The Windows PowerShell version 5 preview boasts a fully featured script editor (bit.ly/1CE7BbG) and a complete configuration management extension called Desired State Configuration (DSC), which you can read more about at bit.ly/1abB8PD. DSC is a critical new addition to Windows PowerShell because it allows IT pros to configure a server once and ensure it stays that way. It offers powerful remoting capabilities through the WS-MAN protocol that give a full-featured psexec-like experience and has a package manager similar to the Linux world's RPM Package Manager (RPM).

The Windows PowerShell team is consistently adding more features by routinely releasing new versions with subsequent versions of Windows, let alone the near-monthly DSC “waves.” Even though Windows PowerShell is rapidly becoming more robust and applicable for orchestration of cloud and multiplatform environments, many IT professionals are resisting the change that comes with automating tasks.

Dozens, if not hundreds, of products now rely on Windows PowerShell as a management and automation platform. Microsoft introduced Windows Server 2008 Core in 2008—the minimal installation instantiation of the server OS—and made Server Core the default install option for Windows Server 2012, which isn't even usable without Windows PowerShell. Yet many IT pros are struggling to change their work routine, put down the mouse and play in the command line as Unix geeks have for decades now.

Windows PowerShell is gaining adoption every year due to Microsoft and, thus by proxy, all of the other software products that have hooks into its software. Even though it took Snover a long time to convince Microsoft of this, as a company it now sees Windows PowerShell as the future of automating Windows and the Windows command line and encourages—if not forces—all future Microsoft products to have full Windows PowerShell support.

As you're about to see in further detail, Windows PowerShell today is on the rise, but hasn't quite achieved mainstream adoption. With time, however, you'll begin to see more and more companies require the kind of automation skills that Windows PowerShell can provide. The industry as a whole seemed OK with subpar automation for a while but it wasn't until the DevOps movement—the new way of

**Look how far
Windows
PowerShell has
come in less
than a decade.**

managing IT by bringing operational considerations and people into the software development, testing and deployment process—came along that the industry raised its expectations for automated operations. That's why knowing how to build and utilize Windows PowerShell scripts and learning its forthcoming new capabilities such as DSC are becoming critical for IT pros.

The Growth of Windows PowerShell

Look how far Windows PowerShell has come in less than a decade. Among searches for the most popular languages, the toddler Windows PowerShell has taken on old geezers in the scripting world, including VBScript, Perl, Bash and Batch, all of which have been around for decades. You'll notice a steady decline for all languages (see **Figure 1**) to eventually meet the Windows PowerShell interest this year (see **Figure 2**, page 16).

As a side effect of overall Windows PowerShell interest, you're beginning to see more IT hiring managers preferring, if not insisting on, job candidates with Windows PowerShell experience. Any job search site today will show hundreds to thousands of jobs calling for Windows PowerShell experience. What's more amazing is the job titles run the gamut from the typical Systems Administrator and Windows Systems Engineer to Enterprise Network Support, SQL Server DBA Engineer and even Security Analyst, according to a March 2015 job search on Monster.

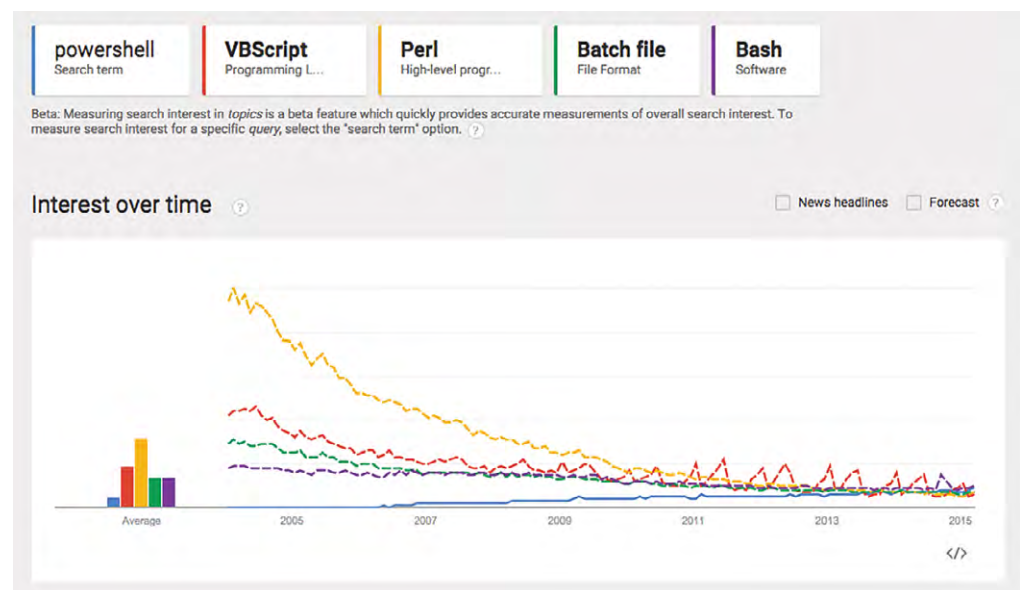


Figure 1. *The trajectory of Windows PowerShell versus other languages. Source: Google Trends*

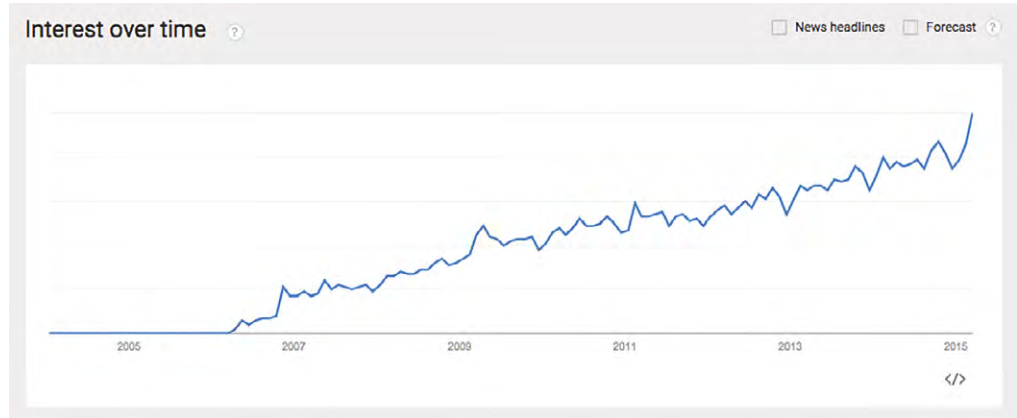


Figure 2. *Google Trends illustrates Windows PowerShell interest continues to rise.*

Windows PowerShell now has full support of nearly every object available in the Windows OS and all Microsoft technologies.

Vendor Adoption

Adding to the appeal of Windows PowerShell is its extensibility. Because it supports standards-based management (bit.ly/1EVO7wk), especially with its introduction of Open Management Infrastructure (bit.ly/1BF0iMd), hundreds of IT vendors large and small are seeing Windows PowerShell as the way to manage their products from the command line. Companies including Cisco Systems Inc., Dell Inc., EMC Corp. Hewlett-Packard Co., NetApp and many others now support Windows PowerShell and are extending PowerShell support to provide IT pros with not only Windows automation, but automation and full management across the entire infrastructure.

Imagine a day when you can write a set of WindowsPowerShell scripts that can provision and configure an entire datacenter. The reality is fast approaching. Windows PowerShell now has full support of nearly every object available in the Windows OS and all Microsoft technologies. This is a given, but who actually has all Microsoft technologies in their datacenter? Who really uses Routing and Remote Access Services (RRAS) over a tried-and-true Cisco ASA firewall?

With Windows PowerShell having growing support for standards, it may soon be possible to script out every layer of your datacenter from setting up VLANs on the switches, carving out LUNs on your SAN, configuring your Cisco UCS chassis, installing and configuring your VMware ESXi infrastructure on that chassis, to provisioning virtual servers and installing and configuring Windows (or Linux).

Never has any other technology come close to this kind of automation across so many different platforms.

Going to the Cloud

Another common concern is the cloud. IT pros who've honed their skills in an on-premises datacenter racking dozens or hundreds of physical servers, ensuring they remained cool while maintaining tight security, have that all-too-common feeling: outsourcing. If the traditional on-premises datacenter is going off to the elusive cloud,

then what work will be left to the IT pro? Plenty, if the IT pro evolves.

Getting Started with Windows PowerShell and DSC

Windows PowerShell has been out and in the mainstream IT pro space for many years now, but there are some IT pros that *still* have never written a single script. Here are some options on how to learn it:

1. **Find a book or course.** Some free options are Microsoft Virtual Academy's "Getting Started with PowerShell 3.0 Jump Start" by Jason Helmick and Jeffrey Snover (bit.ly/1hb0DT8) or the content in the Windows PowerShell Self-Study Guide (bit.ly/1I80jAt). A couple of paid options are the book, "Learn PowerShell in a Month of Lunches, Second Edition," by Don Jones and Jeffery Hicks (Manning Publications Co., 2012) or the book, "Windows PowerShell 3.0 First Steps" by Ed Wilson (Microsoft Press, 2013). Going through a course or book will provide a good foundation.
2. **Solve a real problem.** Once you become familiar enough with Windows PowerShell to run through the basics, taking a real problem you're running into in your job and using Windows PowerShell to solve it is an excellent way to get started.
3. **Ask the community.** The Windows PowerShell community is full of helpful people willing to help you learn the scripting language. Visit the powershell.org forums (bit.ly/1F3PQzL), StackOverflow (bit.ly/1HjLabl) or the powershell.com forums (bit.ly/1EFXO7e).
4. Attend conferences such as TechMentor Redmond this August where Windows PowerShell DSC experts such as Jones, Hicks, Helmick, myself and others will offer deep dives and training (bit.ly/1NeCElh). —A.B.

Popular Windows PowerShell proponent (and former Redmond columnist) Don Jones has a saying for how IT pros should look at it. "You need to be the one who makes the tools, not the one who clicks the tools." In other words, the most successful IT pros are those who build the Windows PowerShell scripts for the others who just want to "do their job." To survive as an IT pro, you must learn to adapt and that means embracing automation and stop clicking through the software wizards.

One of the best aspects of Windows PowerShell is its all-encompassing presence across the IT infrastructure. Windows PowerShell has its hooks in so many different technologies that learning and embracing it gives IT pros the freedom to take those skills

and apply them in numerous areas including the provisioning, orchestration and management of infrastructure and applications in public and hybrid cloud environments.

While IT pros may fear cloud services such as Microsoft Azure, Amazon Web Services and others will automate away their usefulness, IT pros are very much needed to manage these services and users simply want their applications to run without interruption. The difference is before the rise of the public cloud, those services were served by massive servers in on-site datacenters. Now the services have simply shifted to another datacenter somewhere else. Hence, with this shift, comes new ways to provision, connect to and manage these services.

To survive as an IT pro, you must learn to adapt and that means embracing automation and stop clicking through the software wizards.

It means “the server is down” no longer applies due to the distributed nature of the services. A more common scenario is something like five on-premises application servers with three back-end Azure SQL Databases with other file system resources stored in Amazon S3, but with full capability to utilize more cloud infrastructure services if a load gets too high on a moment’s notice. There’s no way this can be managed like traditional on-premises IT infrastructure. This entire orchestration needs to be automated and Windows PowerShell is emerging as the scripting environment for this new environment.

The good news is, nearly every cloud provider has some kind of Windows PowerShell support. If they don’t, it’s flexible enough to easily create your own—and indeed many are doing so. Windows PowerShell expertise is portable and prevalent, so you can take your skills with you to your next job. **R**

Adam Bertram, a Windows PowerShell MVP and Pluralsight author, is also a network engineer. A speaker at TechMentor, he’s giving a class in automating software installs with PowerShell at TechMentor Redmond in August. You can read his blog at adamtheautomator.com or find him on Twitter @adbertram.

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

First Look: Docker Machine for Hyper-V

The new Docker Machine preview lets you build Docker VM hosts on various compatible OS hosts, hypervisors and clouds. **BY BRIEN M. POSEY**

As Microsoft moves to embrace containers to provide virtual machine (VM) and application portability, the company is among numerous players that have aligned with Docker Inc. to enable cross-platform compatibility. Microsoft first announced support for Docker containers in the Microsoft Azure cloud service a year ago followed by last fall's agreement by the two companies to ensure Windows containers are compatible with Docker.

Experts say containers are poised to become a popular means of providing portability between server platforms, VMs and public clouds.

Experts say containers are poised to become a popular means of providing portability between server platforms, VMs and public clouds because the application doesn't need to undergo changes in order to run on a different platform. A virtualized application can, for instance, run on a laptop or in the cloud.

In spite of the compelling capabilities of Docker, many Windows Server shops might be reluctant to use Docker containers because they're designed to run in Linux environments. Docker recently created a utility called Docker Machine designed to deploy Docker containers in proprietary environments. For example, Docker Machine can be used to deploy Docker to Hyper-V, VMware hypervisors and even to Azure. With the release of the Docker Machine preview, it's now possible to run Docker containers in a Hyper-V environment.

Setting up the password file is really easy to do.

Deploying OpenSSH

Before you can use Docker Machine, you'll have to deploy OpenSSH on your Hyper-V server. One of the OpenSSH components (SSL-KeyGen) is used in the creation of the Docker Machine. Although OpenSSH is an open source application originally designed to run on Linux, there's a Windows version available (download from bit.ly/1Dnt5Md). Upon downloading the .zip file and extracting the application bits, run the installer and follow the prompts. The actual installation process is simple, but there are a couple of tasks you'll need to complete before you'll be able to use OpenSSH.

As you work through the installation wizard, you'll see a pop-up message that states:

"Before starting the OpenSSH service you MUST edit the C:\Program Files (x86)\OpenSSH\etc\passwd file. If you don't do this, you will not be able to log in through the SSH server. Please read the readme.txt or the quickstart.txt file for information regarding proper setup of the password file."

Setting up the password file (and the as-yet-unmentioned group file) is really easy to do. In fact, you only have to run four commands. The instructions can be found at C:\Program Files (x86)\OpenSSH\Docs\quickstart.txt and list the exact commands you'll need to use.

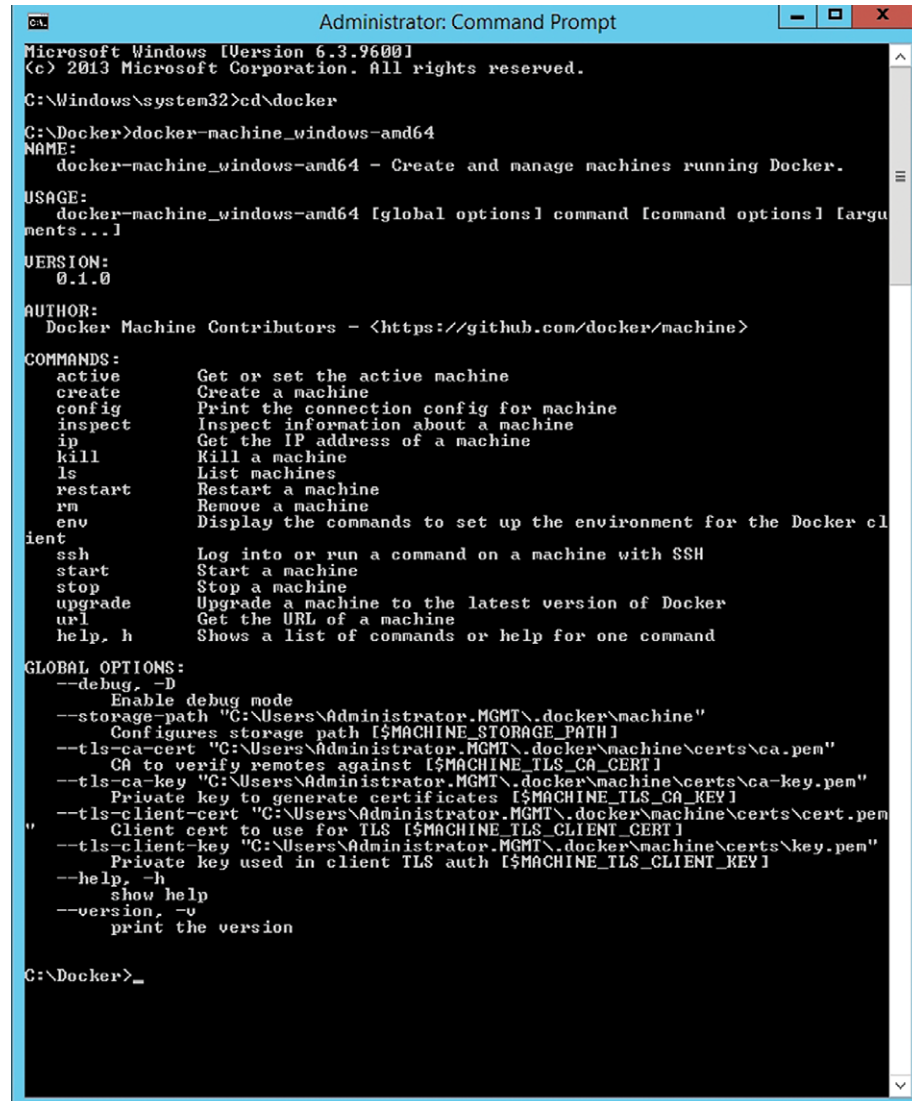
You also must add OpenSSH to your server's path. This is necessary because it'll allow Docker to locate the SSL-KeyGen utility.

To add OpenSSH to the server's path, right-click on the Start button and select the System command from the shortcut menu. When the System dialog box appears, click on Advanced System Settings. This will cause the System Properties dialog box to be displayed. Now, click on the Environment Variables button. You'll find the path in the list of System Variables (you have to scroll down). Edit the path and append C:\Program Files (x86)\OpenSSH\bin to the existing path. Click OK three times to complete the process. The new path won't be used until you reboot the server.

Using Docker Machine

Now that OpenSSH is installed, it's time to download Docker Machine (download from bit.ly/1C3hESI). This article is based on version 1.0, the beta release issued in February.

Because the goal is to use Docker Machine to get the Docker environment up and running on Hyper-V, you have to use several different command-line switches.



```

Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd\docker
C:\Docker>docker-machine_windows-amd64
NAME:
  docker-machine_windows-amd64 - Create and manage machines running Docker.
USAGE:
  docker-machine_windows-amd64 [global options] command [command options] [arguments...]
VERSION:
  0.1.0
AUTHOR:
  Docker Machine Contributors - <https://github.com/docker/machine>
COMMANDS:
  active      Get or set the active machine
  create      Create a machine
  config      Print the connection config for machine
  inspect     Inspect information about a machine
  ip          Get the IP address of a machine
  kill        Kill a machine
  ls          List machines
  restart     Restart a machine
  rm          Remove a machine
  env         Display the commands to set up the environment for the Docker client
  ssh         Log into or run a command on a machine with SSH
  start       Start a machine
  stop        Stop a machine
  upgrade     Upgrade a machine to the latest version of Docker
  url         Get the URL of a machine
  help, h     Shows a list of commands or help for one command
GLOBAL OPTIONS:
  --debug, -D      Enable debug mode
  --storage-path "C:\Users\Administrator.MGMT\.docker\machine"
                  Configures storage path [MACHINE_STORAGE_PATH]
  --tls-ca-cert "C:\Users\Administrator.MGMT\.docker\machine\certs\ca.pem"
                  CA to verify remotes against [MACHINE_TLS_CA_CERT]
  --tls-ca-key "C:\Users\Administrator.MGMT\.docker\machine\certs\ca-key.pem"
                  Private key to generate certificates [MACHINE_TLS_CA_KEY]
  --tls-client-cert "C:\Users\Administrator.MGMT\.docker\machine\certs\cert.pem"
                  Client cert to use for TLS [MACHINE_TLS_CLIENT_CERT]
  --tls-client-key "C:\Users\Administrator.MGMT\.docker\machine\certs\key.pem"
                  Private key used in client TLS auth [MACHINE_TLS_CLIENT_KEY]
  --help, -h      show help
  --version, -v   print the version

C:\Docker>_

```

Figure 1. This is the Docker Machine command-line syntax.

Once the download completes, save the executable file to an easy-to-find folder (I put it in a folder named C:\Docker on my machine).

Now, open an elevated Command Prompt window and navigate to the folder where you saved the executable file. If you attempt to run the file, you'll see the file is meant to be used from the command line and there are a number of different switches available for you to use (see **Figure 1**).

Because the goal is to use Docker Machine to get the Docker environment up and running on Hyper-V, you have to use several different command-line switches. The first switch you'll need to use is Create. This tells Docker Machine you want to create a Docker host.

You can run Docker within Hyper-V or Azure because Docker Machine includes a Hyper-V driver and an Azure driver.

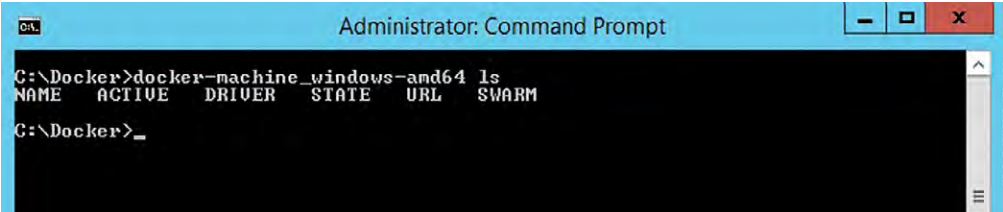
The next thing you'll have to provide is a driver. The driver is used to establish the platform where Docker will run. Docker normally runs inside VirtualBox. You can run Docker within Hyper-V or Azure because Docker Machine includes a Hyper-V driver and an Azure driver. You specify the driver by using the `-D` switch, followed by the driver name.

The last thing you have to specify is the name of the Docker Machine you want to create. For the purposes of this article, I'm going to call my Docker Machine LocalDev.

Next, you must verify no Docker Machines currently exist. To do so, enter the command:

```
Docker-machine_windows-amd64 ls
```

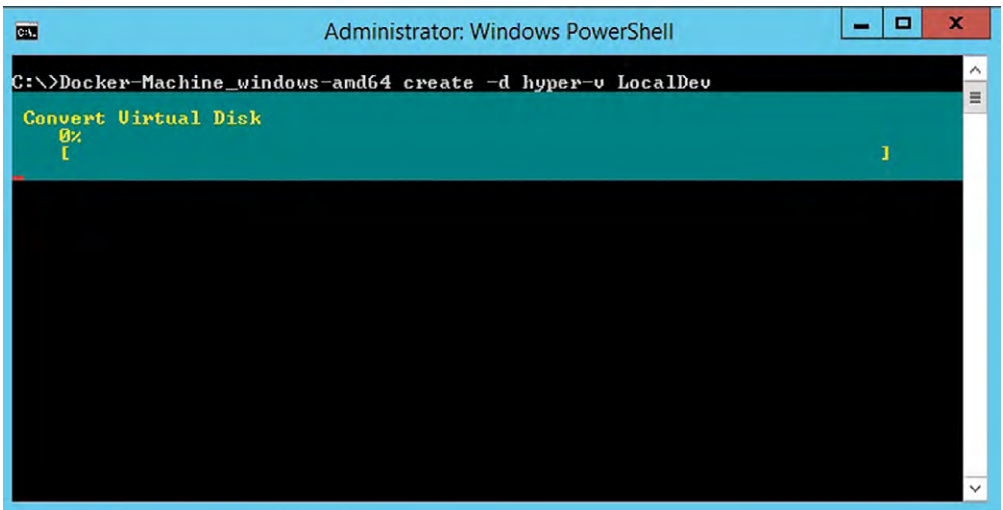
In this scenario no Docker Machines currently exist, which you can see in **Figure 2**.



```
Administrator: Command Prompt

C:\Docker>docker-machine_windows-amd64 ls
NAME    ACTIVE  DRIVER    STATE    URL     SWARM
C:\Docker>
```

Figure 2. No Docker Machines currently exist.



```
Administrator: Windows PowerShell

C:\>Docker-Machine_windows-amd64 create -d hyper-v LocalDev
Convert Virtual Disk
0%
1
```

Figure 3. After running the `Docker-machine_windows-amd64 create -d hyper-v LocalDev` command, Windows begins creating the Docker Machine.

Docker Machine is currently in beta and things can and sometimes do go wrong.

Now, create a Docker Machine using the command syntax. Because you're creating the Docker Machine on Hyper-V, you'll use a driver named hyper-v. The actual command you must use is:

```
Docker-machine_windows-amd64 create -d hyper-v LocalDev
```

If you've done everything correctly, then Windows will begin creating the Docker Machine (see **Figure 3**, p. 22).

When Things Go Wrong

Hopefully you now have a Docker Machine up and running. However, Docker Machine is currently in beta and things can and sometimes do go wrong. What should you do if you get an error message during the Docker Machine creation process? First, check if the Docker Machine was partially created. To do so, enter the same command as earlier, when you verified no Docker Machines exist on your machine:

```
Docker-machine_windows-amd64 ls
```

This command shows you the current state of any existing Docker Machines. Let's presume a Docker Machine is displayed within the results, but you received some errors during the creation process. Unfortunately, you can't just reattempt the creation because Docker Machine won't overwrite an existing machine.

In this type of situation, you should open the Hyper-V Manager. Depending on how far Docker Machine made it into the creation process, you may have a Hyper-V VM with the name you specified for the Docker Machine. This VM might even be running. The current Docker Machine build is a bit flakey and sometimes returns errors when no problem actually exists. During one of my tests, for example, I received an error message saying the Docker Machine had failed to start the VM. Even so, I was able to manually start the VM using Hyper-V Manager with no problems.

In other cases the errors returned by Docker are all too real and you'll have little choice but to recreate the Docker Machine. Before you can reattempt the Docker Machine creation process, you'll need to delete the existing VM, but don't do it through the Hyper-V Manager. You'll need to remove the VM using the command line instead.

Before you remove the Docker Machine, it's a good idea to look at the VM settings within the Hyper-V Manager and confirm the location of the virtual hard disk. In some cases it's necessary to manually remove VM components, so you need to know where the VM is being stored. The virtual hard disk will normally be stored within your user profile at `C:\Users\<your user name>\.docker\machine\machines\<Docker machine name>\disk.vhd`. You can see the location of my Docker Machine virtual hard disk in **Figure 4**.

To remove the Docker machine, go back to the Command Prompt window and enter the following command, `Docker-machine_ windows-amd64 rm LocalDev`, as shown in **Figure 5** (p. 25).

Before you remove the Docker Machine, it's a good idea to look at the VM settings within the Hyper-V Manager and confirm the location of the virtual hard disk.

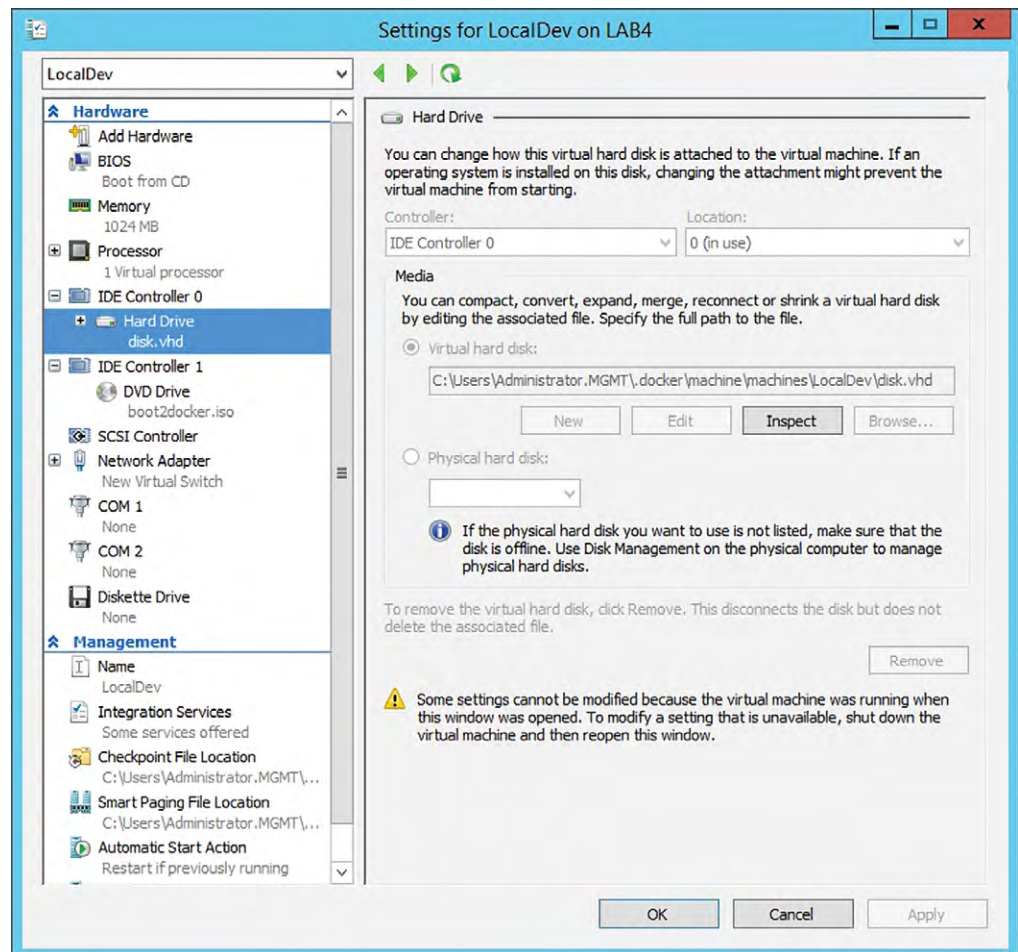


Figure 4. Be sure to make note of the Docker Machine's physical location in case you need to manually remove any virtual machine components.

```

Administrator: Command Prompt

C:\Windows\system32>docker-machine windows-amd64 ls
NAME      ACTIVE  DRIVER  STATE  URL             SWARM
LocalDev  ACTIVE  hyper-v  Running  tcp://147.100.100.250:2376
C:\Windows\system32>docker-machine windows-amd64 rm LocalDev
C:\Windows\system32>docker-machine windows-amd64 ls
NAME      ACTIVE  DRIVER  STATE  URL             SWARM
C:\Windows\system32>_

```

Figure 5. *This is how you remove a Docker Machine.*

Once you confirm the Docker machine is gone, you'll need to check to make sure the Hyper-V components have also been removed.

In some cases the removal process won't work as it should. This is especially true for failures that occur early on, before the Hyper-V VM is created. In those types of situations you can append the `-F` switch to the end of the removal command to force removal of the Docker Machine.

As you can see in **Figure 5**, you can confirm the removal by using the `Docker-machine_windows-amd64 ls` command. Once you confirm the Docker machine is gone, you'll need to check to make sure the Hyper-V components (such as the virtual hard disk) have also been removed. You can do this by opening File Explorer and navigating to the path of the VM. If you find any remnants of the VM, go ahead and remove them.

Testing Your Docker Machine

Once you get your Docker Machine up and running, you'll need to make sure it's working properly. When you created the Docker machine, you might have noticed the installer downloaded an ISO file called `Boot2Docker`. If you look back at **Figure 4**, page 24, you'll notice the installer has linked this ISO file to the VM DVD drive. The VM boots from this ISO file. If you open the VM console, you should see the `Boot2Docker` environment (see **Figure 6**, page 26).

If you've ever taken a programming class in school, you'll probably remember the old "Hello World" program. Believe it or not, Docker has its own Hello World program you can use to verify Docker is working properly. To run the Hello World test, enter the following command in the `Boot2Docker` VM:

The Hello World program does more than just echo a block of text.

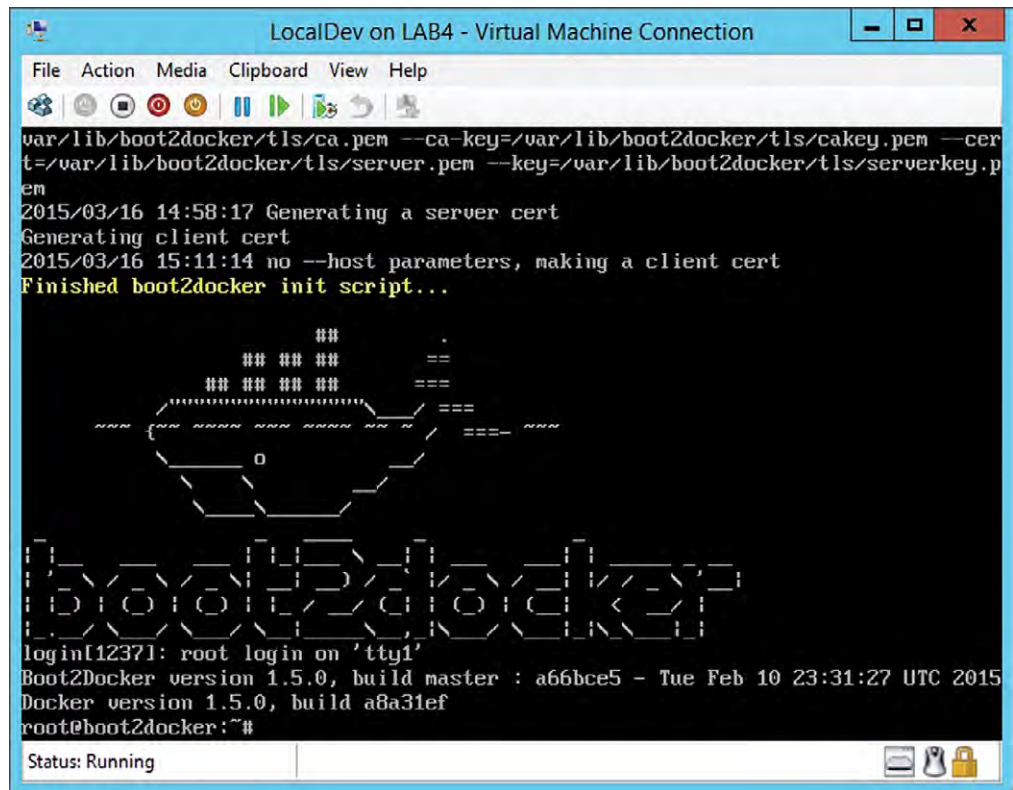


Figure 6. *This is what Boot2Docker looks like.*

Docker run hello-world

Upon executing this command, you should see a Hello from Docker message (see **Figure 7**, page 27). More important, Docker confirms four things have happened:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the Hello World image from the Docker Hub (unless it was locally available).
3. The Docker daemon created a new container from the Hello World image. This container is used to run the executable that produces the output shown in **Figure 7**, page 27.
4. The Docker daemon streamed the output to the Docker client, which caused the output to be displayed.

In other words, the Hello World program does more than just echo a block of text. It actually confirms the ability of the VM to create and use a Docker container.

It's relatively
easy to get Docker
up and running
in a Hyper-V
environment.

```

LocalDev on LAB4 - Virtual Machine Connection
File Action Media Clipboard View Help
511136ea3c5a: Pull complete
31cbccb51277: Pull complete
e45a5af57b00: Pull complete
hello-world:latest: The image you are pulling has been verified. Important: image
verification is a tech preview feature and should not be relied on to provide
security.
Status: Downloaded newer image for hello-world:latest
Hello from Docker.
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (Assuming it was not already locally available.)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

For more examples and ideas, visit:
http://docs.docker.com/userguide/
root@boot2docker:~#
Status: Running

```

Figure 7. The Hello World program confirms Docker is working properly.

Another way you can test your Docker Machine is to enter the Docker Info command. Doing so should return a number of different statistics about your Docker Machine.

Next Steps

As you can see, it's relatively easy to get Docker up and running in a Hyper-V environment. Yes, Docker Machine is still a little bit buggy, but those bugs should be fixed by the time Docker provides a full release of Docker Machine. For right now, though, you can use Docker in a lab to get used to working with the Docker environment. **R**

Brien M. Posey is a seven-time Microsoft MVP with more than two decades of IT experience. He's written thousands of articles and several dozen books on a wide variety of IT topics. Visit his Web site at brienposey.com.

vSphere 6.0



VMware has rolled out its first major new hypervisor platform in three years and introduces some features not yet in Hyper-V.

BY BRIEN M POSEY

Now that VMware has released vSphere 6.0, it's time to see how it compares to Hyper-V.

Since the release of Windows Server 2012 R2, which included the third generation of Hyper-V, the Microsoft hypervisor reached near parity with the vSphere virtualization platform from VMware Inc. Unquestionably, there are some things vSphere does better than Hyper-V, but there are also areas where Hyper-V does excel over vSphere.

And now that VMware has released vSphere 6.0, the first major upgrade of its hypervisor platform in three years, it's time to see how it compares to Hyper-V. While a feature-by-feature comparison would be too exhaustive for an overview, I'll emphasize the most important features in the new vSphere 6.0. VMware published a technical white paper with a complete list of new vSphere 6.0 features, which you can download from vmw.re/1gAGcdJ.

Scalability

VMware ESXi offers improved scalability over its predecessor. The hypervisor can now scale to support up to 64 hosts in a cluster,

double the previous limit of 32 hosts per cluster. A VMware 6.0 cluster can accommodate 8,000 virtual machines (VMs)—a twofold increase over the prior release.

The company has also improved the scalability of individual hosts. A single-host server can now accommodate up to 1,000 VMs. A host now supports up to 480 physical CPUs and up to 12TB of RAM. Windows Server 2012 R2 Hyper-V also supports up to 64 nodes per cluster and each cluster can accommodate up to 8,000 VMs. Individual Hyper-V hosts can accommodate up to 320 logical processors and up to 4TB of RAM. A single Hyper-V server can host up to 1,024 running VMs.

VMware has made significant security enhancements to its new hypervisor.

Security

VMware has made significant security enhancements to its new hypervisor. For starters, VMware enables central management of accounts and permissions for individual host servers. It's also now possible to centrally manage password complexity rules for hosts in a cluster. VMware has also introduced a couple new settings for the management of failed logon attempts with local accounts.

In contrast, Hyper-V is a Windows Server role, and Windows Server has long supported the central management of security and passwords via Active Directory.

Virtual Machines

Given VM support is the core function of any hypervisor, it should come as no surprise VMware has taken steps to improve VM functionality. In doing so, VMware has introduced VM hardware version 11. For those not familiar with this concept, VMware has a history of introducing a new VM hardware version with each major release. For example, hardware version 9 was introduced with vSphere 5.1 and hardware version 10 was introduced with vSphere 5.5.

The notion of VM hardware versions is something Microsoft has only recently adopted. In Windows Server 2012 R2, Microsoft introduced generation 2 VMs (see **Figure 1**, page 30). These second-generation VMs used different virtual hardware than first-generation VMs. Generation 2 VMs, for example, supported the ability to boot from SCSI virtual hard disks, used UEFI firmware and performed PXE boots from standard network adapters (among other things).

Microsoft has offered support for virtual NUMA since Windows Server 2012 Hyper-V.

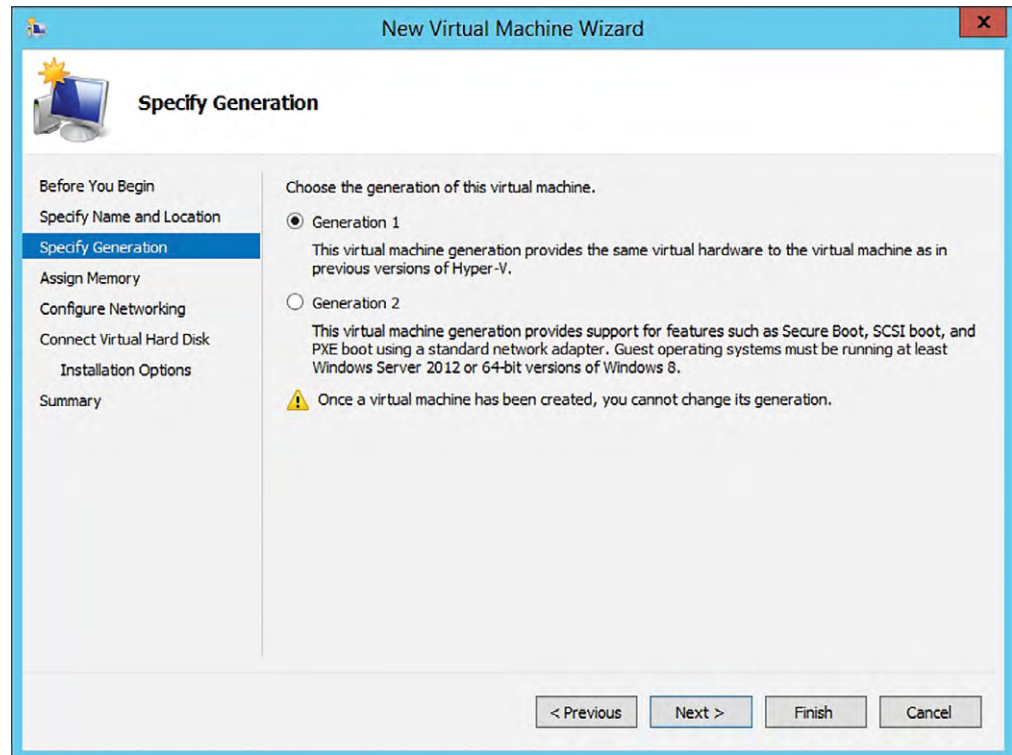


Figure 1. Microsoft introduced second-generation virtual machines in Windows Server 2012 R2 Hyper-V.

Similar to Microsoft VM generations, VMware VM hardware versions represent changes in the virtual hardware. Version 11 VMs offer some improvements in the way non-uniform memory access (NUMA) memory is used. When memory is hot-added to a VM, that memory is allocated equally across all NUMA regions. Before, the memory was allocated only to region zero. This makes it easier to scale a VM without taking it offline.

Microsoft has offered support for virtual NUMA since Windows Server 2012 Hyper-V. This virtual NUMA support allows a VM to span multiple NUMA nodes so the VM can scale to support larger workloads (see **Figure 2**, page 31) and can take advantage of NUMA-related performance optimizations.

Another enhancement to VMware VMs in virtual hardware version 11 is the increase to 32 serial ports that are supported. Also, vSphere 6.0 gives administrators the ability to remove unneeded serial and parallel ports.

This is one area VMware clearly has better support than Microsoft. For all practical purposes, Hyper-V doesn't support the use of

One more change VMware has made to its VMs is there are a number of new guest OSes that are now officially supported.

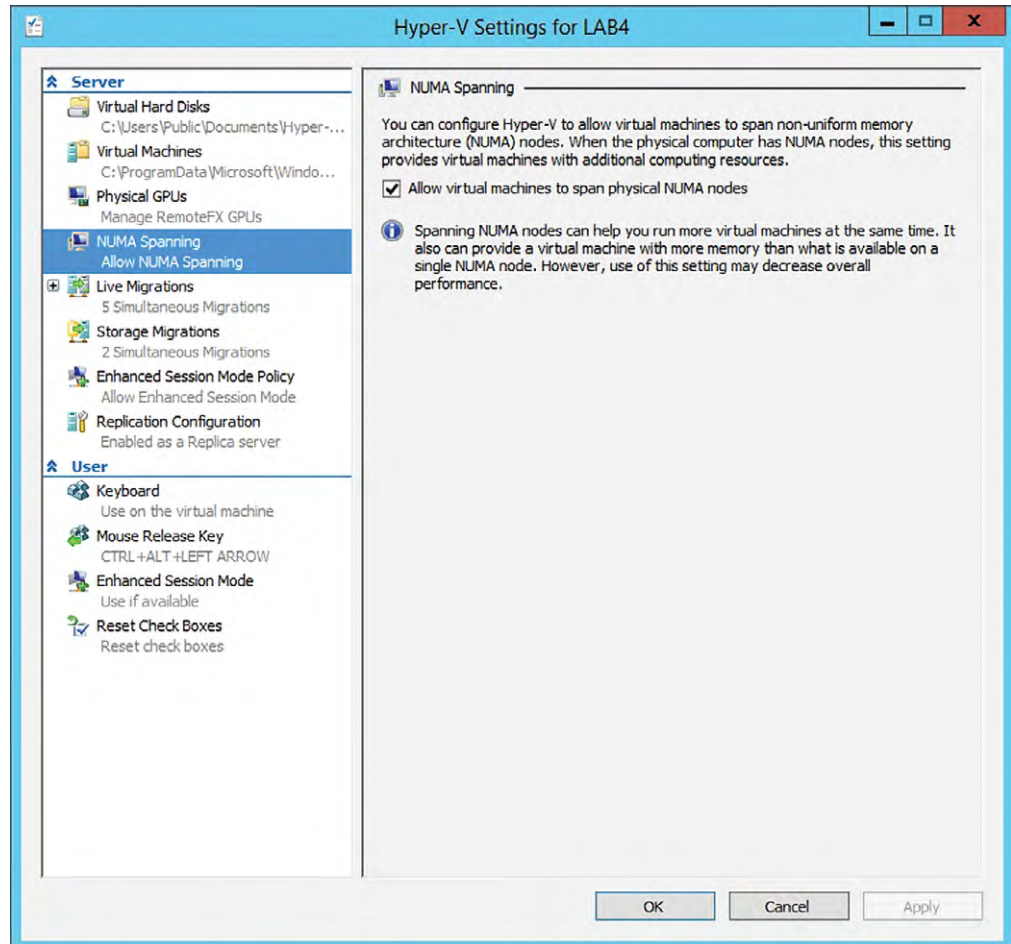


Figure 2. Hyper-V also supports spanning non-uniform memory access nodes.

physical serial or parallel ports by VMs. However, it's possible to configure a virtual COM port to communicate with a physical computer through a named pipe (see **Figure 3**, p. 32). Microsoft also provides a workaround for generation 2 VMs, which allows a serial port to be used for debugging purposes (bit.ly/1yGyeHO) even though virtual COM ports don't officially exist in generation 2 VMs.

One more change VMware has made to its VMs is there are a number of new guest OSes that are now officially supported. Among the newly supported OSes are:

- Oracle Unbreakable Enterprise Kernel Release 3 Quarterly Update 3
- Asianux 4 SP4
- Solaris 11.2

- Ubuntu 12.04.5
- Ubuntu 14.04.1
- Oracle Linux 7
- FreeBSD 9.3
- Mac OS X 10.10

VMware has published a complete list of supported OSes, which is available at vmw.re/1IUTXC7.

Like VMware, Microsoft supports a number of different guest OSes for use with Hyper-V. The supported Windows OSes (complete list available at bit.ly/1y3udTo) include:

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 SP1

**Like VMware,
Microsoft supports
a number of
different guest
OSes for use
with Hyper-V.**

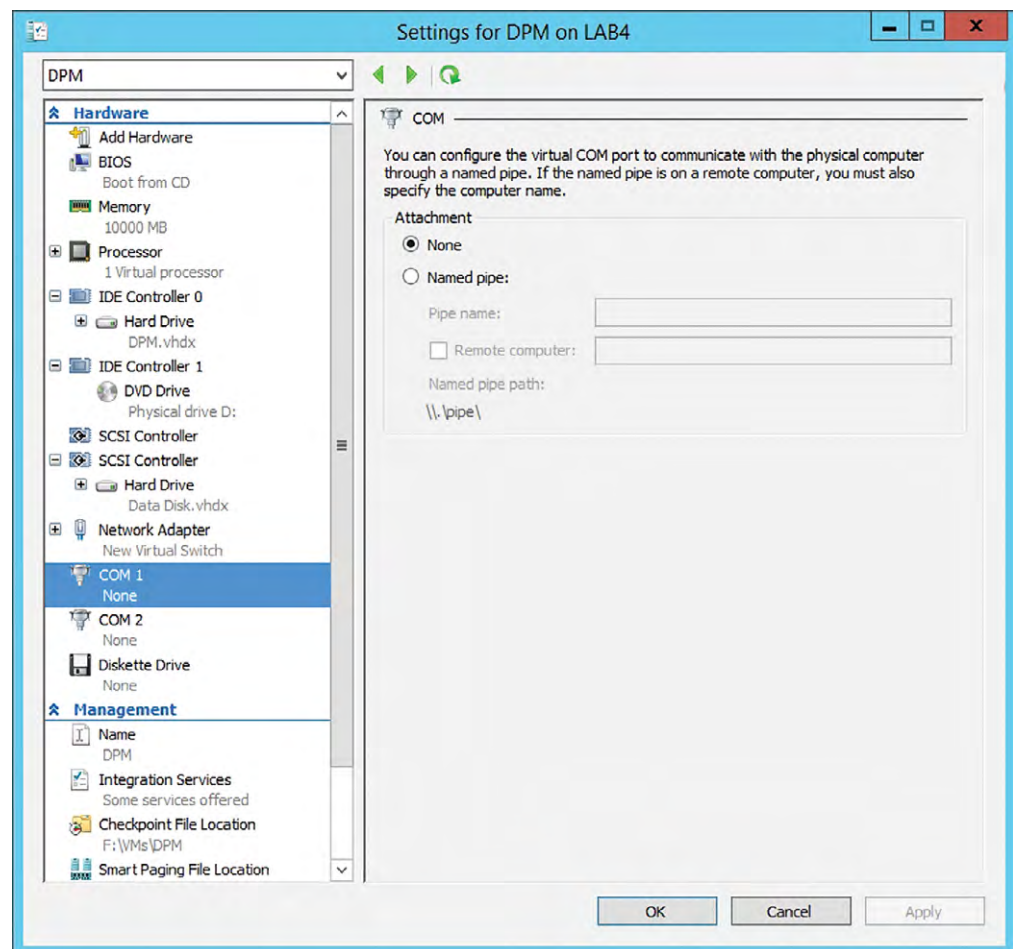


Figure 3. Virtual COM ports can communicate with physical computers through named pipes.

**In the past,
VM storage has
primarily been
provisioned at
the LUN level.**

- Windows Server 2008 SP2
- Windows Home Server 2011
- Windows Small Business Server 2011
- Windows Server 2003 R2 SP2
- Windows Server 2003 SP2
- Windows 8.1
- Windows 8
- Windows 7 SP1
- Windows 7
- Windows Vista SP2
- Windows XP SP3
- Windows XP X64 Edition SP2

Microsoft also provides support for many of the more popular Linux distributions (bit.ly/1y3udTo), including:

- CentOS
- Red Hat Enterprise Linux
- Debian
- Oracle Linux
- SUSE
- Ubuntu
- FreeBSD

Virtual Volumes

The most prominent new feature in vSphere 6.0 is Virtual Volumes, which is essentially a mechanism for allowing array-based operations at the virtual disk level.

In the past, VM storage has primarily been provisioned at the LUN level. With releases prior to vSphere 6.0, when a VMware administrator would have to set up some new VMs, it involved calling or e-mailing the storage administrator and asking them to create a LUN according to certain specifications. The storage administrator created the LUN, making sure it would use the most appropriate storage type, RAID level and so on. Once this LUN was created, the virtualization administrator could create a datastore on the LUN and begin creating VMs. This process might have varied slightly from one organization to the next, but in every case there was abstraction between physical storage and the individual virtual hard disks.

The main advantage of using Virtual Volumes is it allows VMware administrators to leverage the storage hardware's native capabilities at the VM level.

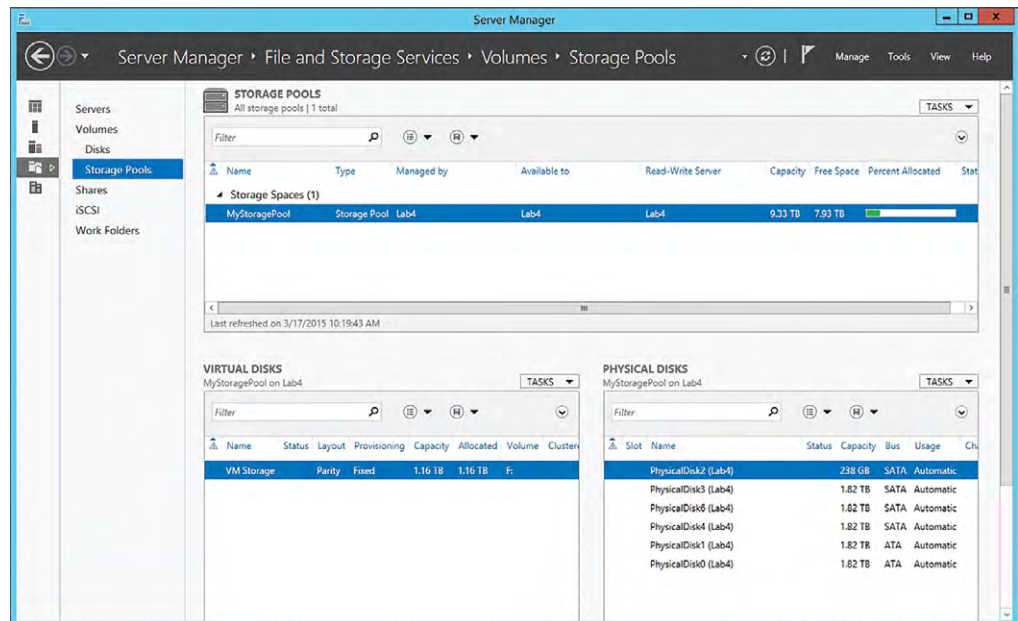


Figure 4. Windows Server supports the creation of virtual disks on physical storage.

Virtual Volumes is designed to change the way VMs interact with physical storage. This new feature allows VMDK files (VMware virtual hard disks) to natively interact with the physical storage array and allows the physical storage array's capabilities to be exposed through vCenter. This means a VMware admin can direct the array to create a VMDK that meets a specific set of requirements.

The main advantage of using Virtual Volumes is it allows VMware administrators to leverage the storage hardware's native capabilities at the VM level. Virtual Volumes makes it possible, for example, to clone, replicate or make a snapshot of a VMDK. Sure, VMware administrators already had the ability to clone, replicate or make snapshots of VMs, but the difference is the Virtual Volumes feature allows those functions to be performed at the storage level rather than at the hypervisor level.

Hyper-V doesn't have an equivalent feature. However, it's important to note Hyper-V is a Windows Server role, not a standalone product like VMware ESXi. Although Hyper-V itself doesn't have a VMware-style Virtual Volumes feature, the underlying Windows Server OS does provide native support for storage hardware. Windows Storage Spaces, for example, exposes storage to the OS. Virtual disks can be

Windows Server 2012 R2 is also designed to take advantage of native storage hardware capabilities by default.

created within a storage pool in a way that mimics various RAID levels. For instance, you can create a virtual disk on top of a storage pool made up of physical disks (see **Figure 4**, page 34).

Windows Server 2012 R2 is also designed to take advantage of native storage hardware capabilities by default. For example, storage arrays commonly offer a feature called Offload Data Transfer (ODX). Windows Server is able to leverage this feature for file copy operations, thereby allowing those operations to be handed off to the storage hardware rather than the operation being performed at the server level. The advantage to using this feature is the operation completes more quickly and it doesn't consume CPU and network bandwidth resources on servers.

vSphere Network I/O Control Enhancements

Yet another new capability introduced by VMware in vSphere 6.0 is vSphere network I/O control. This is a fancy way of saying VMware

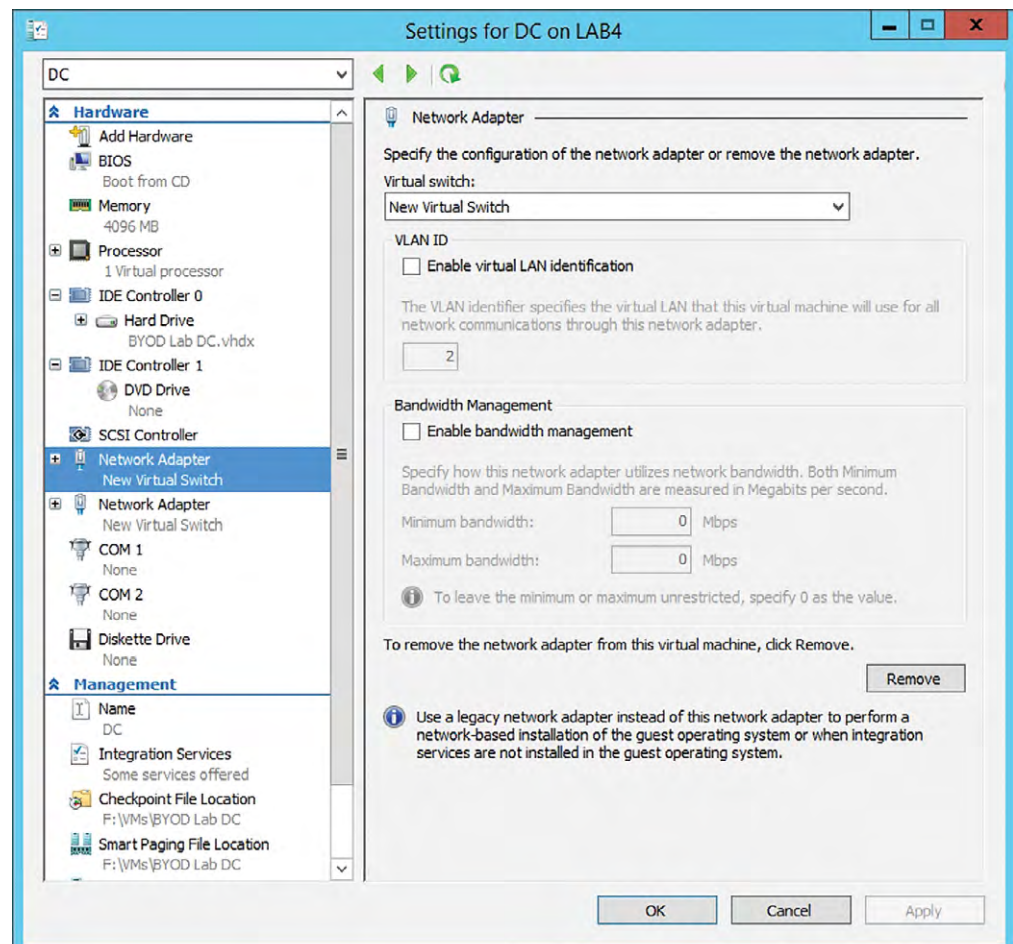


Figure 5. Hyper-V has its own Bandwidth Management feature.

VMware Ups the Ante

- Virtual machines (VMs) per cluster available double from 4,000 to 8,000
- Doubling the number of hosts per cluster from 32 to 64
- CPUs per host increase 50 percent from 320 to 480
- Supported RAM per host up 3x from 4TB to 12TB
- VMs per host expands 4x from 512 to 2,048
- Virtual CPUs per VM doubles from 64 to 128
- Virtual RAM per VM up 4x from 1TB to 4TB

allows administrators to reserve network bandwidth for use by a specific vNIC (or a distributed port group). The benefit to doing so is bandwidth reservations allow administrators to guarantee that VMs receive a specified minimum level of bandwidth regardless of the demand for network bandwidth made by other VMs.

It appears Microsoft engineers also understand the importance of bandwidth management, as Hyper-V already supports the reservation of network bandwidth. Hyper-V not only lets you reserve network bandwidth on a per-vNIC basis (see **Figure 5**, page 35), you can use the Bandwidth Management feature to cap network bandwidth consumption. This is especially useful for reining in chatty VMs because it's often easier to limit your chatty VMs than to reserve bandwidth for all of your other VMs.

The Winner Is ...

So how does vSphere 6.0 compare to the current version of Hyper-V? In my opinion, the two hypervisors are more or less on par with one another. VMware has added some important new features, however, at least some of those new features mimic capabilities that already exist in Hyper-V or in the Windows OS.

In all fairness, there are plenty of other new VMware features not covered here. Many revolve around multi-datacenter deployments. VMware outlines this in detail in the technical white paper I mentioned earlier.

Given the new feature set that VMware has brought to the table, it'll be interesting to see how Windows Server vNext Hyper-V compares upon its release next year. **R**

Brien M. Posey is a seven-time Microsoft MVP with more than two decades of IT experience. He's written thousands of articles and several dozen books on a wide variety of IT topics. Visit his Web site at brienposey.com.

GetMore Online

For a deeper look at what's new in vSphere 6.0, **Redmond** magazine sister publication **Virtualization Review** has extensive coverage and analysis of the new hypervisor platform. VirtualizationReview.com/vSphere6