Microsoft® System Center

Building a Virtualized Network Solution, Second Edition

Nigel Cain, Michel Luescher, Damian Flynn, Alvin Morales
Mitch Tulloch, Series Editor

PUBLISHED BY
Microsoft Press
A division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number
ISBN: 978-0-7356-9580-1

Printed and bound in the United States of America.

First Printing

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Support at mspinput@microsoft.com. Please tell us what you think of this book at http://aka.ms/tellpress.

Acquisitions Editor: Karen Szall
Developmental Editor: Karen Szall
Editorial Production: Megan Smith-Creed
Copyeditor: Megan Smith-Creed
Cover: Twist Creative • Seattle

Contents

Introduction

According to the Hyper-V Network Virtualization overview at *http://technet.microsoft.com/en-us/library/jj134230.aspx,* Network Virtualization "provides virtual networks to virtual machines similar to how server virtualization provides virtual machines to the operating system. Network Virtualization decouples virtual networks from the physical network infrastructure and removes the constraints and limitations of VLANs and hierarchical IP address assignment from virtual machine provisioning. This flexibility makes it easy for customers to move to Infrastructure as a Service (IaaS) clouds and efficient for hosters and datacenter administrators to manage their infrastructure while maintaining the necessary multi-tenant isolation, security requirements, and supporting overlapping Virtual Machine IP addresses."

Although the benefits of this approach are very clear, designing and implementing a solution that delivers the promised benefits is both complex and challenging; architects, consultants, and fabric administrators alike often struggle to understand the different features and concepts that make up a solution.

Who should read this book?

Much of the current published material covering Network Virtualization is focused on the *how*, the set of tasks and things that you need to do (either in the console or through Windows PowerShell) to set up and configure the environment. In this book, we take a different approach and instead consider the *what*, with a view to helping private and hybrid cloud architects understand the overall architecture, the role each individual feature plays, and the key decision points, design considerations, and best practice recommendations they should adopt as they begin to design and build out a virtualized network solution using Windows Server and Microsoft System Center Virtual Machine Manager.

In summary, this book is specifically designed for architects and cloud fabric administrators who want to understand what decisions they need to make during the design process and the implications of those decisions, what constitutes best practice, and, ultimately, what they need to do to build out a virtualized network solution that meets today's business requirements while also providing a platform for future growth and expansion.

New to this second edition are chapters covering the Hyper-V Network Virtualization gateway, designing a solution that extends an on-premises virtualized network solution to an external (hosted) environment, details of how to troubleshoot and diagnose some of the key connectivity challenges, and a look at the Cloud Platform System (CPS) and some of the key considerations that went into designing and building the network architecture and solution for that environment.

In writing this book, we assume that, as architects and fabric administrators interested in Microsoft Network Virtualization, you are familiar with and have a good understanding of the networking features and capabilities of Windows Server, Hyper-V, and Virtual Machine Manager, as well as the Microsoft Cloud OS vision available at *http://www.microsoft.comen-us/server-cloud/cloud-os/default.aspx*.

What topics are included in this book?

The vast majority of the book is focused on architecture and design, highlighting key design decisions and providing best practice advice and guidance relating to each major feature of the solution.

- **Chapter 1: Key concepts** A virtualized network solution built on Windows Server and System Center depends on a number of different features. This chapter outlines the role each of these features plays in the overall solution and how they are interconnected.
- **Chapter 2: Logical networks** This chapter provides an overview of the key considerations, outlines some best practice guidance, and describes a process for identifying the set of logical networks that are needed in your environment.
- **Chapter 3: Hyper-V port profiles** This chapter discusses the different types of port profiles that are used in Virtual Machine Manager, outlines why you need them and what they are used for, and provides detailed guidance on how and when to create them.
- **Chapter 4: Logical switches** This chapter describes the function and purpose of logical switches, which are essentially templates that allow you to consistently apply the same settings and configuration across multiple hosts.
- **Chapter 5: Network Virtualization gateway** This chapter outlines key design choices and considerations for providing cross-premises connectivity from networks at tenant sites to virtual networks dedicated (per tenant) in a service provider network.
- **Chapter 6: Deployment** This chapter builds on the material discussed in previous chapters and walks through common deployment scenarios, highlighting known issues (and workarounds) relating to the deployment and use of logical switches in your environment.
- **Chapter 7: Operations** Even after having carefully planned a virtual network solution, things outside of your immediate control might force changes to your virtualized network solution. This chapter walks you through some relatively common scenarios and provides recommendations, advice, and guidance for how best to deal with them.

- **Chapter 8: Diagnosing Connectivity Issues** This chapter looks at how to approach a connectivity problem with a virtualized network solution, the process you should follow to troubleshoot the problem, and some actions you can take to remediate the issue and restore service.
- **Chapter 9: Cloud Platform System network architecture** This chapter reviews the design and key decision points for the network architecture and virtualized network solution within the Microsoft Cloud Platform System.

To recap, this book is mainly focused on architecture and design (what is needed to design a virtualized network solution) rather than on the actual steps required to deploy it in your environment. Other than in few chapters, you will find few examples of code. This is by design. Our focus here is not to provide details of how you achieve a specific goal but rather on what you need to do to build out a solution that meets the needs of your business and provides a platform for the future.

When you have designed a solution using the guidelines documented in this book, you will be able to make effective use of some of the excellent materials and examples available in the Building Clouds blog (*http://blogs.technet.com/b/privatecloud/*) to assist you with both solution deployment and ongoing management.

Acknowledgments

The authors would like to thank, once again, our original reviewers Stanislav Zhelyazkov (MVP), Hans Vredevoort (MVP), and Phillip Moss (NTTX) as well as Greg Cusanza, Thomas Roettinger, Artem Pronichkin, and Cristian Edwards Sabathe from Microsoft for providing valuable feedback and suggestions on the content of the book. We would also like to thank and show our appreciation to Nader Benmessaoud, Robert Davidson, Ricardo Machado, Kath McBride, and Larry Zhang (all from Microsoft) for their review, feedback, and comments specific to this second edition. Without their contributions, this book would not be as thorough nor as complete as you find it, so our thanks once again for their time and efforts in making this happen.

Free ebooks from Microsoft Press

From technical overviews to in-depth information on special topics, the free ebooks from Microsoft Press cover a wide range of topics. These ebooks are available in PDF, EPUB, and Mobi for Kindle formats, ready for you to download at:

http://aka.ms/mspressfree

Check back often to see what is new!

Errata, updates, & book support

We've made every effort to ensure the accuracy of this book and its companion content. You can access updates to this book—in the form of a list of submitted errata and their related corrections—at:

http://aka.ms/VNS2E

If you discover an error that is not already listed, please submit it to us at the same page.

If you need additional support, email Microsoft Press Book Support at *mspinput@microsoft.com*.

Please note that product support for Microsoft software and hardware is not offered through the previous addresses. For help with Microsoft software or hardware, go to *http://support.microsoft.com*.

We want to hear from you

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

http://aka.ms/tellpress

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in touch

Let's keep the conversation going! We're on Twitter: *http://twitter.com/MicrosoftPress*.

CHAPTER 1

Key concepts

A virtualized network solution built on Windows Server and Microsoft System Center depends on a number of different features. It is important to understand the role these features play in the solution and how they are interconnected, especially if you need to troubleshoot issues with connectivity or have to make changes to your solution to reflect updated business requirements.

This chapter will:

- Introduce Fabrikam, a provider trying to build a reliable and effective cloud solution based on Microsoft technologies
- Identify the different features of a virtualized network solution
- Provide an overview of each feature and its purpose

Introducing Fabrikam Ltd.

The chapters in this book describe the key design decisions for each of these features and how they all fit together. To help make the discussion a little more real and to put the key points into context, the chapters use a fictitious organization called Fabrikam.

Fabrikam is a service provider—otherwise known as a hoster—that offers infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) to customers from datacenters located in the United States (Seattle) and the United Kingdom (Reading). Fabrikam has more than 1,000 employees worldwide, with the majority of its employees employed in its development and operations center in Reading. Revenues in the last financial year topped £100 million for the first time. Fabrikam has decided to deploy and use the Microsoft Cloud Platform—essentially, Windows Server 2012 R2, Microsoft System Center 2012 R2, and the Windows Azure Pack—for its hosting services moving forward because the company recognizes this platform's ease of deployment and the cost and efficiency benefits in terms of infrastructure provisioning, infrastructure monitoring, application performance monitoring, automation and self-service, and IT service management.

Although your organization might not be a service provider and your business model and requirements might differ significantly from those of Fabrikam, the design processes, key decision points, and implications of certain design choices are applicable to all customers that plan to use Windows Server and System Center to create a cost-effective and highly efficient private or hybrid cloud solution.

Solution architecture

Figure 1-1 is a simplified diagram that illustrates the different layers and features that make up the architecture of a virtualized networking solution based on Windows Server and Microsoft System Center. Network virtualization transforms physical network devices into an automated pool of resources that can be controlled entirely in software. In this diagram, the physical network and Hyper-V host computers are at the bottom and the deployed virtual machines (VMs) and services are at the top. On the right are the names of each feature; the labels on the left describe how these features are connected. For example, a logical switch is connected to a logical network via a logical network.

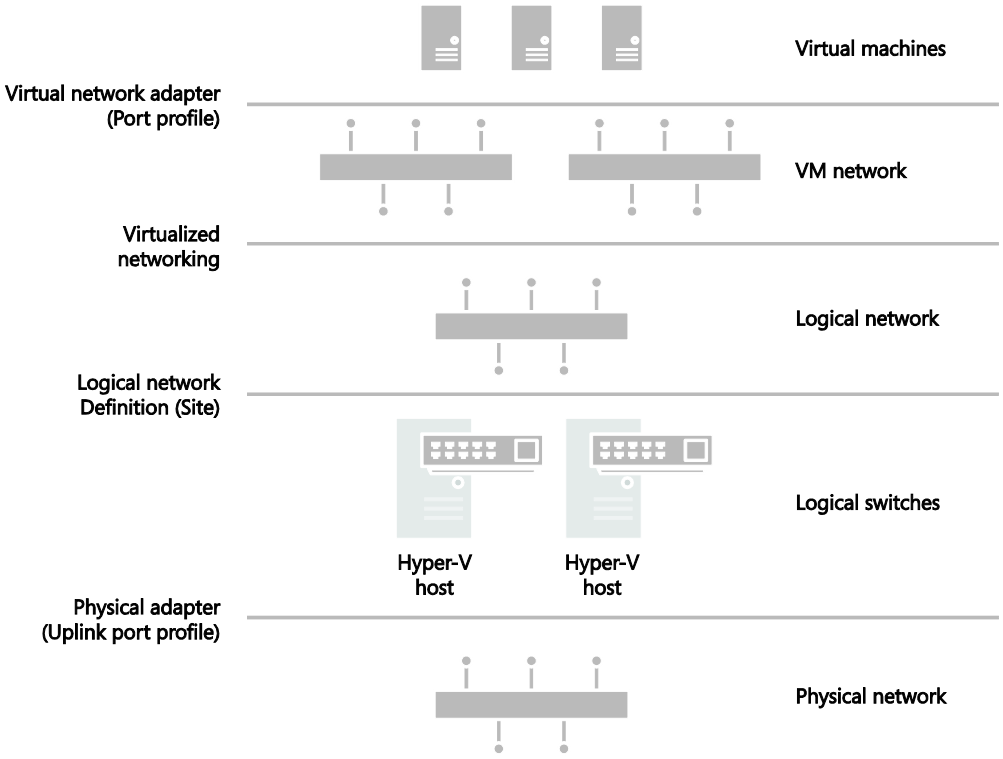


FIGURE 1-1 Architecture of a virtualized network solution

The sections that follow provide an overview of each of the major features shown in Figure 1-1 and explain what they are used for and how they connect to other features in the solution. Subsequent chapters go into more detail and explain how to deploy and use these features within your environment.

Logical networks

The Microsoft System Center Virtual Machine Manager (VMM) documentation says that "A logical network is used to organize and simplify network assignments for hosts, virtual machines, and services. As part of logical network creation, you can create network sites to define the VLANs, IP subnets, and IP subnet/VLAN pairs that are associated with the logical network in each physical location." The documentation goes on to state that logical networks can be used to describe networks with different purposes, to create traffic isolation, and even to support traffic with different types of service level agreements. You can find more information on logical networks and how to determine how many you need in your environment in Chapter 2, "Logical networks."

At Fabrikam, Hyper-V hosts supporting production workloads are situated in two physical locations, Reading and Seattle, with each site using a different VLAN and IP subnet range. VMs running production workloads on hosts in the Reading datacenter need to use VLAN 18 and have an IP address in the 192.168.99.0/24 subnet, where those in Seattle should use VLAN 100 and have IP address in the 192.168.199.0/24 subnet. To allow the Production logical network to be supported in both of these locations, two network sites must be defined, as shown in Figure 1-2.

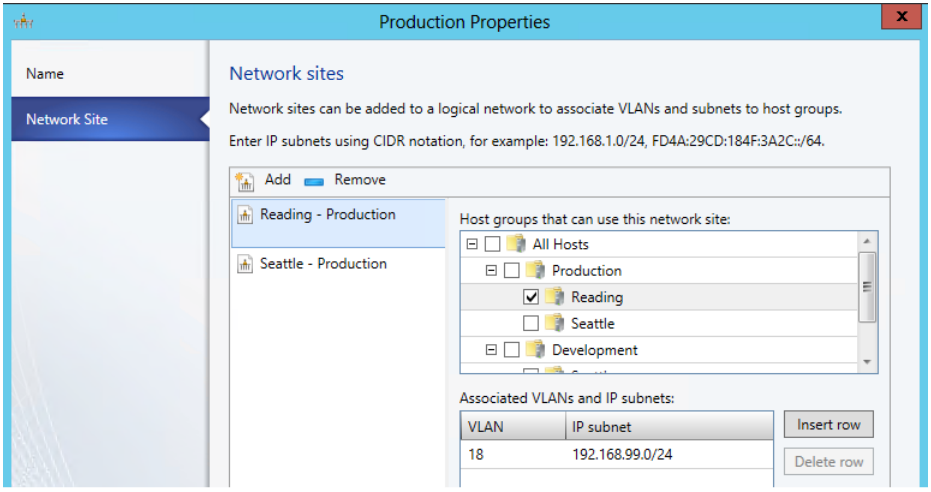


FIGURE 1-2 Defining sites within a logical network

The Reading network site is scoped to Hyper-V hosts deployed in Reading. It defines the VLAN and IP subnets that will be used by VMs that connect to the Production logical network when running on a Hyper-V host in the Reading location. The other network site is scoped to the Seattle host group and essentially defines the VLANs and subnets that will be used by VMs deployed in Seattle.

Note that scoping the logical network to a host group in the network site as shown above does not actually make the logical network available on any of the hosts within the group. It simply prevents the logical network from being associated with hosts that are not within the target groups. To make the logical network available on a given host, you need to associate the logical network with a physical network adapter on that host.

At Fabrikam, READING-VMH2 is one of the servers located in the Reading datacenter. The server is a member of the host group that is authorized for the Production logical network, and since this logical network has been successfully associated with one of the physical network adapters, as shown in Figure 1-3, it can be made available to VMs running on that host.

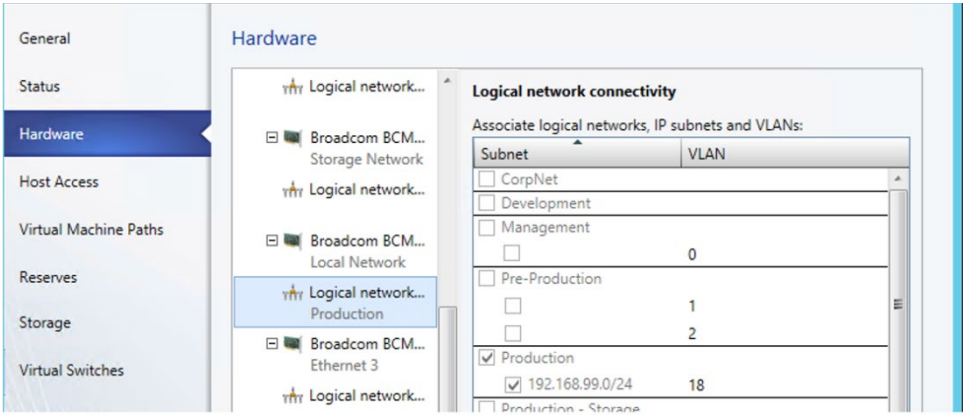


FIGURE 1-3 Logical networks associated with a physical network adapter

You might expect that the result of this configuration, when it has been deployed to hosts in both locations, would be that a VM connected to the Production logical network can be moved between hosts in Reading and Seattle without requiring any additional configuration. The destination Hyper-V host in the new location ensures that the VM is configured with the VLAN and IP address appropriate for the logical network in the new physical location.

Moving existing VMs between sites like this is certainly possible, but there are a few caveats. The main one is that the IP address assigned to the VM will not be changed during migration. If the physical network is a stretched LAN, meaning the same IP subnet is present in both locations, then the VM will continue to communicate on the network when it has been moved. If, as in the earlier example, each site has its own VLAN and IP subnet, then although you will be able to successfully move the VM to a new location, it will have an incorrect VLAN/IP address for that location.

NOTE **A VM connected to a VM network that uses network virtualization to isolate tenants may be moved between different host computers and will continue to be able to communicate on the network without requiring changes to its IP address.**

IP address pools

Static IP address pools (not shown in Figure 1-1) make it possible for VMM to automatically allocate static IP addresses to Windows-based VMs running on any managed Hyper-V, VMware ESX, or Citrix XenServer host.

VMM can automatically assign static IP addresses from the pool to stand-alone VMs and to VMs that are deployed as part of a service. It can also assign addresses to physical computers when you use VMM to deploy them as Hyper-V hosts or SMB v3 file servers. When you create a static IP address pool, you can also define a reserved range of IP addresses that can be assigned to load balancers as virtual IP addresses. VMM automatically assigns a virtual IP address to a load balancer during the deployment of a load-balanced service tier. If you define the IP address inside the VM manually, VMM detects the IP address and the pool to which it belongs (if defined) at the *next* refresh cycle. This process helps to ensure that VMM does not assign the selected IP address to another VM.

NOTE When isolating network traffic using network virtualization, which is covered in more detail in Chapter 2, you must create two IP pools, one for the logical network and one for (each) VM network associated with that logical network.

MAC address pools

Static MAC address pools (also not shown in Figure 1-1) make it possible for VMM to automatically allocate static MAC addresses to VMs running on any managed Hyper-V, VMware ESX, or Citrix XenServer host.

It is important to note that if you configure a VM to obtain an IP address from a static IP address pool, you must also configure the VM to use a static MAC address. In this case, you can either specify the MAC address manually or have VMM automatically assign a MAC address from either a central MAC address pool or one that you have created for a specific network site.

Uplink port profiles

Uplink port profiles are applied to physical network adapters as part of logical switch deployment and define the set of logical networks that should be associated with those network adapters. They also specify whether and how multiple network adapters (in a given host computer) using the same uplink port profile should be teamed.

In most cases, a single uplink port profile is required for each physical network unless you need to define custom connectivity rules, have multiple physical networks, or want to restrict logical networks to specific hosts within a given physical location, in which case you need to create additional uplink port profiles. You can find more details on uplink port profiles as well as a process to help you determine whether you need to create more than one of them in Chapter 3, "Hyper-V port profiles."

At Fabrikam, a number of hosts in Reading and Seattle have been dedicated to production workloads, and port profiles and logical switches (which are discussed in Chapter 4, "Logical switches") will be used to ensure the host computers in each location are configured consistently. Assuming that the servers in each location have the same type of physical connectivity, only a single uplink port profile should be required.

Figure 1-4 illustrates the network sites that have been configured for the Production uplink port profile. When this uplink is applied to one or more of the network adapters in a Hyper-V host computer in Reading, for example as part of logical switch deployment, it will associate those network adapters with the Production logical network and will also automatically configure the adapter with the VLANs and subnets (as listed in the Reading Production network site) that will be used by VMs in that location.

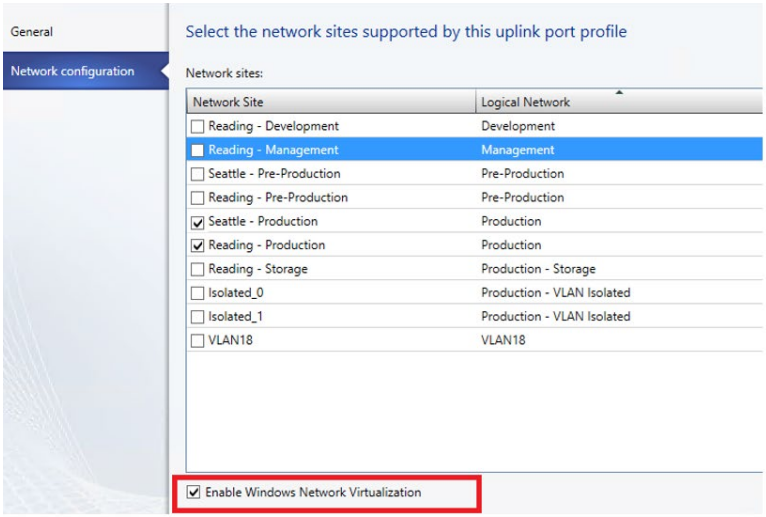


FIGURE 1-4 Defining network sites (and logical network connectivity) in an uplink port profile

In the example above, multiple network sites are supported by a single uplink profile. When the uplink port profile is applied to a physical network adapter as part of logical switch deployment, VMM checks each network site in the uplink to determine if the host is "in scope" for that site. If it is in scope, the network adapter will be associated with all of the logical networks that are defined in that network site.

Network adapter port profiles

Network adapter port profiles, which are called *native port profiles* for virtual network adapters in VMM 2012 SP1 and *Hyper-V port profiles* for virtual network adapters in the R2 release, are essentially templates that allow you to define offload and security settings for virtual network adapters. Network adapter port profiles allow you to define settings such as virtual machine queue (VMQ), IPsec task offloading, and single root I/O virtualization (SR-IOV) in one place and apply these settings to any virtual network adapter in your environment. You can

configure security settings, for example to prevent MAC spoofing, and you can set the bandwidth weight and minimum and maximum possible bandwidth allowed, as illustrated in Figure 1-5.

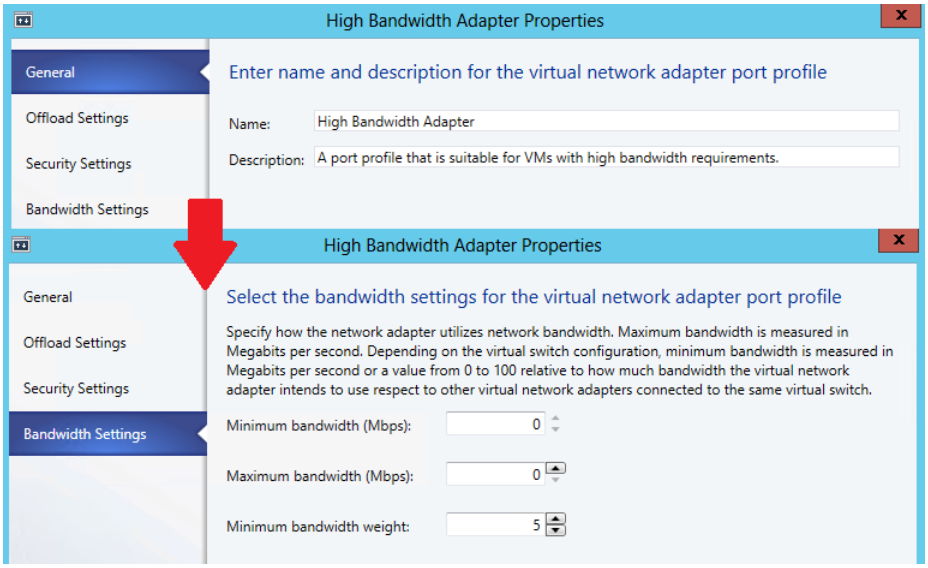


FIGURE 1-5 Defining bandwidth policy in port profiles

NOTE **Although native port profiles allow you to enable network settings for a virtual network adapter, to be effective, some of these (IPsec task offloading, for example) require additional configuration on the host computer.**

Network adapter port profiles and how you can configure and use them are covered in Chapter 3, but to summarize, network adaptor port profiles are used to define the quality of service (QoS) settings you want to apply to specific VMs and network cards that allow you to take advantage of some of the features provided by your host hardware.

Port classifications

Port classifications are not shown in Figure 1-1 but are linked to network adapter port profiles. They hide the details, settings, and configuration of a network adapter port profile from the end user. When connecting a VM to the network, end users will see a list of port classifications they can select from, for example "high bandwidth" or "low bandwidth," but they can't see the priority, bandwidth settings, and IEEE priority value behind a particular configuration. Port classifications are linked to network adapter port profiles and are discussed in Chapter 3.

Logical switches

A logical switch brings together all of the different uplink port profiles, native port profiles, port classifications, and switch extensions that are relevant to a particular physical network. A logical switch is essentially a template that contains an administrator-defined set of parameters you can use to create Hyper-V virtual switches on any of the host computers that connect to the network. When you use a logical switch to create a Hyper-V switch on a host computer, you select the most appropriate combination of port profiles, classifications, and switch extensions from the list of those defined in the logical switch. Generally, a new logical switch is required for every physical network in your environment. But if you plan to restrict some logical networks to a limited set of hosts, as in the example organization in this chapter, or if you have custom connectivity requirements, you may need to create additional logical switches. Chapter 4 covers the design rationale for logical switches.

Given that the example organization has three physical networks (Datacenter, Provider, and Storage), at least three logical switches must be created based on the above guidelines. However, only a limited number of hosts in Reading and Seattle will run production workloads that need to be associated with the Production logical network created earlier. The key question is whether an additional logical switch is required to support this environment.

Technically, the Production uplink port profile can be included in the logical switch created for the Datacenter network, and the administrator can choose the most appropriate settings and capabilities for the relevant host. VMM can even actively prevent administrators from using any of the Production uplinks when they use the logical switch to create a Hyper-V virtual switch on a host that should not be associated with the Production logical network.

The downside to this approach, however, is that a consistent configuration across hosts in Production is not guaranteed. Although uplink port profiles are restricted to certain hosts, administrators can choose from any of the network adapter port profiles, port classifications, and switch extensions that are available within the selected logical switch. In addition, you may find that capabilities you want offered only on production systems, such as network traffic tagged with IEEE high priority and given maximum bandwidth, are associated with other (non-production) systems because the administrator selected the wrong network adapter port profile during logical switch deployment. To avoid this issue, you should create a separate logical switch for Hyper-V hosts that will support production workloads (see Figure 1-6).

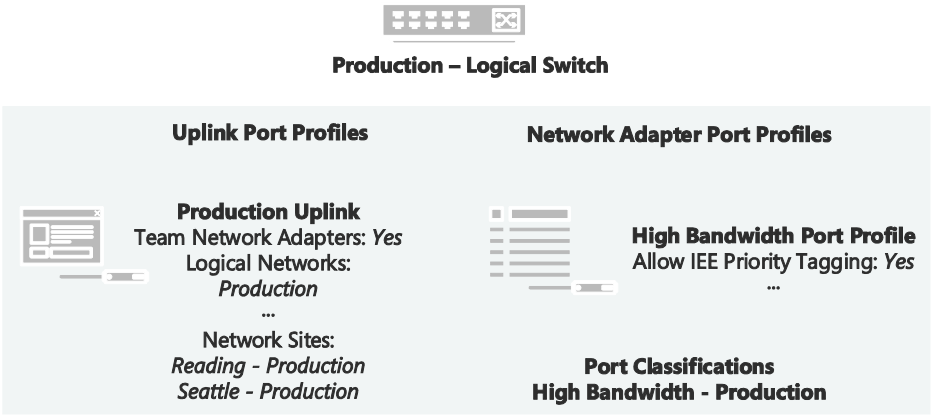


FIGURE 1-6 Contents of the Production logical switch

As shown in Figure 1-6, the new logical switch will contain the Production uplink port profile and a single network adapter port profile that will ensure that network traffic from these hosts and the VMs running on them are tagged with the required IEEE priority flags and are provided with the appropriate bandwidth guarantees. The port classification High Bandwidth - Production shown in Figure 1-6 is simply a friendly name for the network adapter port profile and will be displayed to users when they connect their VMs to the network.

NOTE **The previous example does not include any switch extensions; however, you might want to include these in your logical switch to allow you to monitor network traffic, use QoS to control how network bandwidth is used, enhance the level of security, or otherwise expand the capabilities of a Hyper-V virtual switch created on a host computer. If these enhanced services should be restricted or deployed only on a limited number of hosts, you might need to consider creating an additional logical switch.**

See also *You can find more information on Hyper-V virtual switches at http://technet.microsoft.com/library/hh831823.*

VM networks

In terms of overall architecture, VM networks are the final feature to consider in this short overview since they provide the (network) interface through which a VM connects to a particular logical network, as shown in Figure 1-7. You can find more details on VM networks in Chapter 2. Since all VMs must be connected to a VM network to be able to use and access network resources in VMM, it follows that you will need at least one VM network for each logical network.

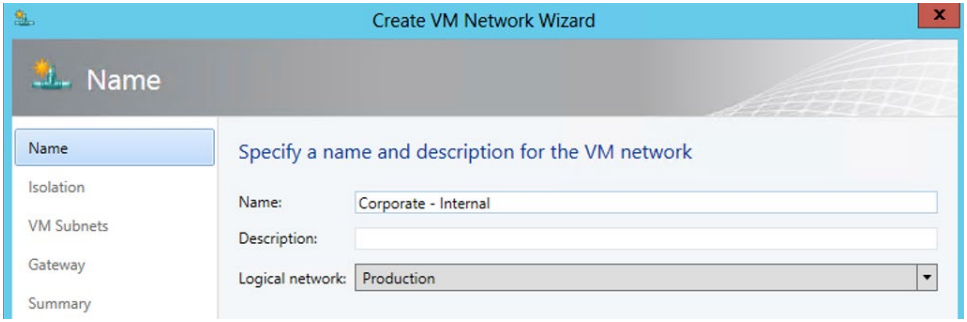


FIGURE 1-7 Mapping a VM network to a logical network

Multiple VM networks can be connected to the same logical network, with each one isolated from and totally unaware of the existence of any others. Isolation can be achieved through traditional VLAN (or isolated PVLAN) or through the concept of network virtualization. The isolation of tenant networks through either of these mechanisms provides the foundation for "bring-your-own-IP" or "bring-your-own-network" solutions currently offered by many service providers.

It is important to note that the relationship between a VM network and its (host) logical network is established when the VM network is initially created and cannot be changed afterward. To use a different logical network, you will need to delete the existing VM network and create a new one.

Hyper-V Network Virtualization gateways

In general, VMs connected to one network will need to connect to and leverage resources on another. For networks isolated through traditional VLAN and/or PVLAN technologies, this connectivity is achieved through the use of (physical) switches and routers.

For VM networks isolated using network virtualization, the Hyper-V Network Virtualization gateway (not shown in Figure 1-1) provides similar functionality—connecting resources on virtualized networks to resources on non-virtualized (external) networks—the key difference being the fact that this gateway can be implemented as a software-only solution.

Network virtualization works by adding additional information (tenant meta-data) to IP packets, which both identifies the IP packets belonging to the tenant and also isolates the traffic from other tenants. When VMs send packets to external networks, the tenant's meta-data needs to be removed. Similarly, when IP packets from external networks are destined for VMs isolated through network virtualization, the IP packets need to be tagged with tenant-specific meta-data to be routed to the correct VM network and to a specific VM(s) on that network. At its core, this is essentially what the Hyper-V Network Virtualization gateway does.

Connectivity to a public network is achieved through the use of network address translation (NAT), as shown in Figure 1-8, but the gateway also allows tenants to connect back to their own (on-premises) network through a VPN tunnel. It can also facilitate connectivity between multiple VM networks hosted by the service provider through the use of direct routing. You can find more details on each of these scenarios and how you would make use of them in Chapter 5, "Network Virtualization gateway."

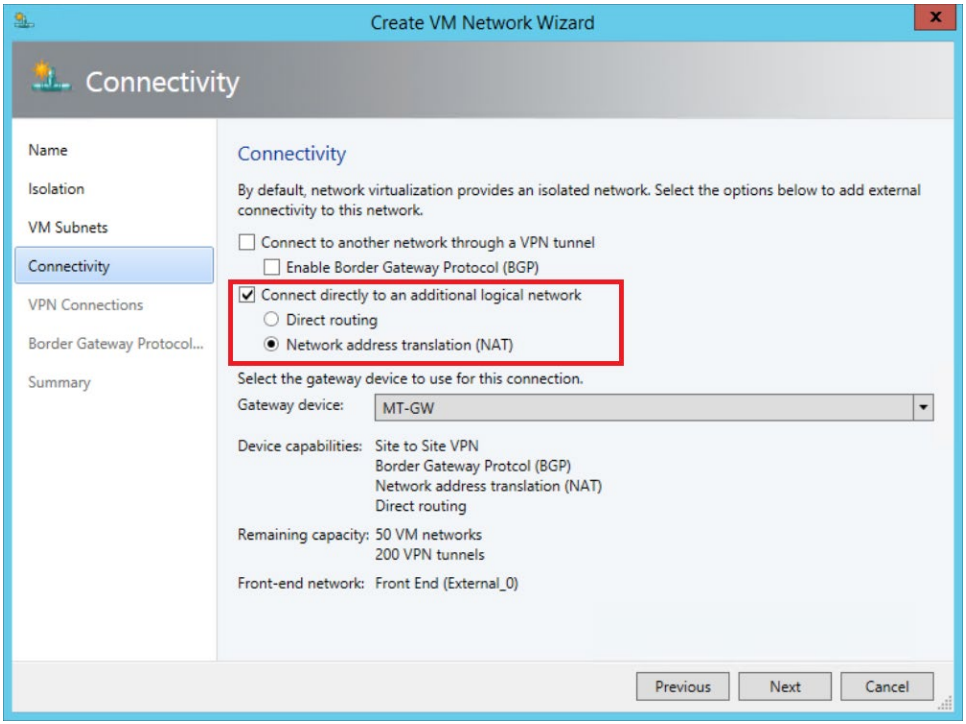


FIGURE 1-8 Using a gateway to connect a VM network to a public network

As discussed previously, the gateway can be implemented in software. Templates are provided for you to quickly and easily deploy the gateway using VMM, but they can also be incorporated into a top-of-rack (TOR) switch, put into an existing network appliance, or even serve as a stand-alone network appliance.

> ***See also*** *You can find more information on Hyper-V Network Virtualization gateways at https://technet.microsoft.com/en-us/library/jj618319.aspx.*

CHAPTER 2

Logical networks

Logical networks represent an abstraction of the underlying physical network infrastructure and enable you to model the network based on business needs and connectivity properties. Logical networks can be used to describe networks with different purposes, create traffic isolation, and even support traffic with different types of service level agreements.

This chapter will:

- Identify where logical networks fit into a virtualized network solution
- Determine how and why logical networks are created automatically
- Introduce a step-by-step process for logical network design
- Consider how to optimize design to support network traffic isolation
- Discuss the use of network sites, IP, and MAC address pools
- Try to help answer the question "How many logical networks do I really need?"

Reviewing key concepts

To help set context for this discussion, begin by referring to Figure 1-1 in Chapter 1, "Key concepts." This diagram illustrates the different layers that make up the architecture of a virtualized networking solution, highlighting logical networks and their connections to other features of the architecture. The key takeaways from this diagram for Chapter 2 are:

- Logical networks are connected to a logical switch via network sites (sometimes referred to as Logical Network Definitions) and to VM networks via virtualized networking.
- VM networks provide the network interface through which a VM connects to a particular logical network.

In addition, since all virtual machines (VMs) must be connected to a VM network to access network resources in Microsoft System Center Virtual Machine Manager (VMM), it follows that you will need to define *at least one* VM network for each logical network that will be accessed by VMs.

Although not shown in Figure 1-1, logical networks also have a relationship with clouds. VMM uses this relationship to scope or otherwise restrict the list of VM networks that are available to users during VM placement. To be placed in a cloud, the VM must be connected to a VM network that is linked to a logical network associated with that cloud. Chapter 6, "Deployment," examines this relationship and how it is used.

Logical network design

The goal in this chapter is to present a step-by-step approach to logical network design, starting from the basic principle that you should begin as simple as possible and then add additional logical networks only where there is a compelling business or technical reason to do so. The process can be summarized as follows:

1. First, define a set of logical networks that initially mirror the physical networks in your environment.
2. Define networks that have a specific purpose or perform a particular function within your environment.
3. Identify which logical networks need to be isolated and how that isolation will be enforced, either through physical separation, VLAN/PVLAN, or network virtualization.
4. Determine the network sites, VLANs, PVLANs, and IP pools that need to be defined for each logical network you have identified.
5. Finally, associate the logical network with the host computers that will support it (you will find details for doing so in Chapter 6).

As usual, defining and adhering to a sound naming convention for logical networks is important both to promote understanding and to help simplify management and reduce cost.

Introducing the Fabrikam network

To set this process in context, take a closer look at the physical network in Fabrikam and identify the set of logical networks that will be needed to support this company's specific business requirements. As shown in Figure 2-1, Fabrikam has three physical networks in each of its datacenters:

- Corporate (internal) workloads and services are hosted on the Datacenter network.
- Customer (tenant) network traffic is on the Provider network.
- Storage devices are accessed via the Storage network.

The physical separation between customer, storage, and corporate network traffic is designed to improve operational security, to simplify management, and to remove potential competition between different types of network traffic.

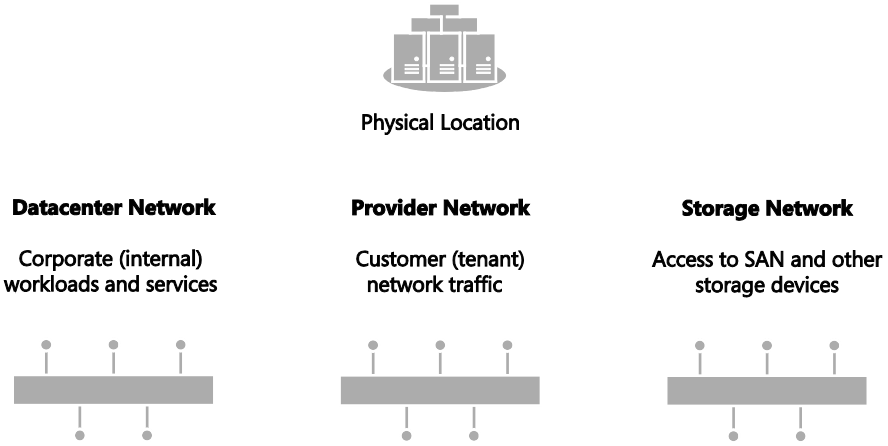


FIGURE 2-1 Physical networks in each Fabrikam datacenter

The following sections outline the five-step logical network design process for Fabrikam, identifying the set of logical networks the company needs to support its business and technical requirements and highlighting some of the key decision points and best practice recommendations along the way.

> **NOTE** **Although you may have a very different network architecture and business requirements from Fabrikam's, the process set out below and the key decision points are applicable to all environments.**

Step 1: Mirror physical networks

It seems reasonable to begin by creating logical networks that map to and mirror each of the physical networks in the environment, but you should expect to create many more logical networks than you have physical networks. Indeed, one of the key benefits of logical networks is that they provide flexibility, allowing you to separate computers and network services with different business purposes, isolate network traffic, or support different workloads with network service levels, all without having to change the physical network infrastructure. With that said, creating one logical network for each physical network is a very useful beginning.

Since Fabrikam has three physical networks (Datacenter, Provider, and Storage), the assumption is that three logical networks will be required to support this environment, one for each physical network, as shown in Figure 2-2. As you will discover in the sections that follow, you will likely need to create additional logical networks to support specific business and technical requirements. But as a guiding principle, you should always start with as simple a design as possible, adding logical networks only where there is a clear and justifiable reason for doing so.

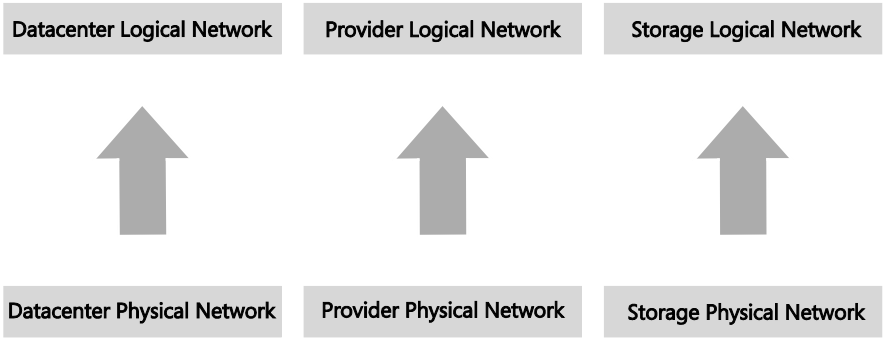


FIGURE 2-2 Logical networks that mirror the physical network

Step 2: Networks with different purposes

It's a basic assumption that computers and devices that connect to and use the same network should be able to communicate with each other with routers or gateway devices used to connect different networks should this be required. This general principle also holds true for logical networks, so the next step in the design process is to identify the different groups of users, applications, and network services that will use each of the physical networks and determine whether there is a need to separate them to enforce security, ensure privacy (isolation), simplify management and administration, or simply to ensure that network traffic from certain groups is provided with the required quality of service (QoS).

Step 1 started with the principle that a single logical network would be sufficient for each of Fabrikam's three physical networks, Datacenter, Provider, and Storage. Step 2 reviews each physical network to determine whether this design is appropriate for the groups of computers and network services that will use them.

Datacenter physical network

The Datacenter physical network at Fabrikam carries network traffic generated by corporate (internal) services and applications as well as network traffic needed to support and maintain the cloud fabric (infrastructure services such as host management, live migration, and cluster heartbeat). Step 1 established a single logical network, Datacenter. The question is whether this design is appropriate for the workloads on this network.

Corporate (internal) services If development, test, and production network traffic all share the same physical network, you will invariably want to differentiate these workloads. In the example Fabrikam environment, development, pre-production, and production workloads will coexist on the Datacenter physical network. To make this environment easier to manage, three separate logical networks are created, one for each workload type, as shown in Figure 2-3. Note that network adapter port profiles (explained in Chapter 3, "Hyper-V port profiles") will be used to apply the required properties and capabilities, including bandwidth limitations and IEEE priority tags, to VMs and services that connect those networks.

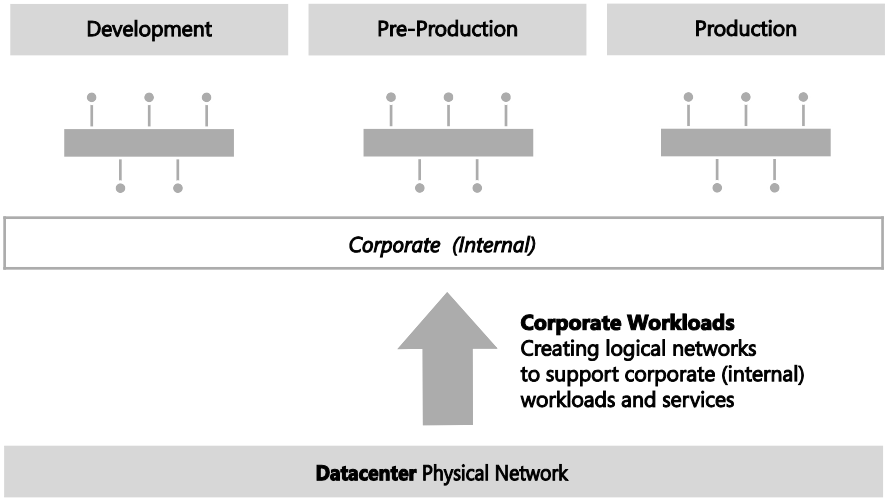


FIGURE 2-3 Using logical networks to differentiate workloads

NOTE **If corporate policy mandates that an application or workload can be hosted only on a particular group of host computers, you would start by defining a separate logical network and then using host groups and network sites to ensure that it is only associated with the selected host computers.**

The VMM documentation suggests that you also consider creating separate logical networks for the front end (web servers) and the back end (application and database servers) of multi-tier applications. The primary benefit of such an approach is that it allows you to use network sites to define the set of VLANs and IP subnets that will be used by VMs in each tier and, further, to apply a different set of security settings and capabilities to each network through the use of port profiles.

Since Fabrikam is expecting to deploy and use multi-tier applications, the logical network design for internal (corporate) workloads needs to be refined with the creation of separate logical networks for the front end and back end of these services, as shown in Figure 2-4. Note that production workloads that are not part of any multi-tier application will be expected to connect to and make use of the Back End logical network.

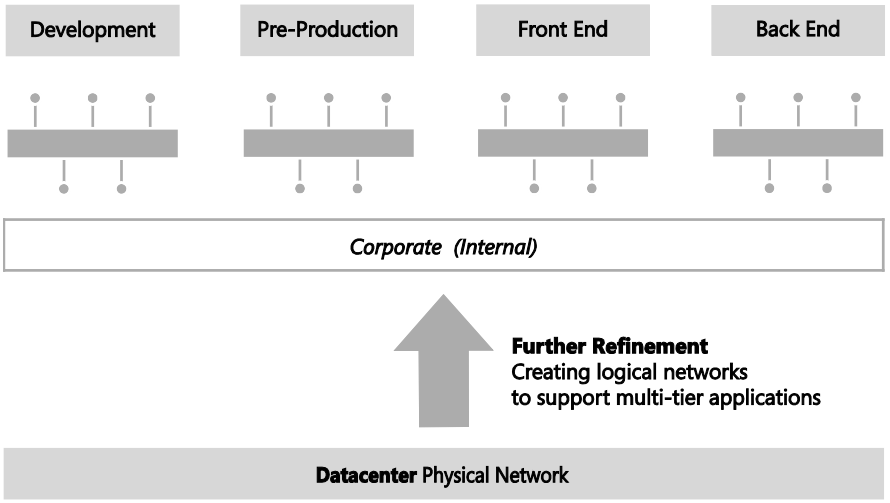


FIGURE 2-4 Dividing Production into front end and back end logical networks

Traffic prioritization, network bandwidth control, and support for multi-tier applications are just a few of the reasons why you might consider creating logical networks for corporate (internal) workloads. Security concerns, the requirement to isolate certain workloads, and the need to restrict the host computers on which a given business service can run are also key considerations. Consider each case on its merits, reviewing the business case as well the technical requirements, with the aim of creating logical networks only when really necessary and keeping the design as simple and as easy to understand as possible.

Cloud infrastructure As mentioned earlier, Fabrikam network traffic for cloud infrastructure (fabric) management will be on the same physical datacenter network as corporate workloads and will probably require a separate logical network to differentiate this traffic from anything else on the network. There are a number of different types of cloud infrastructure traffic, including backup operations, live migration, hardware management, and host/guest management. Will a single logical network suffice for all of these operations or will it be necessary to create logical networks for each type of infrastructure traffic?

In keeping with the guiding principle to create logical networks only when necessary, the key decision point is whether to apply different capabilities, bandwidth controls, and network traffic prioritization to each one of these services. If the answer is no, then a single logical network will suffice.

If the answer is yes for a limited number of these services (backup and live migration are normally good candidates), then a dedicated logical network for those services should be created, with the remainder using a shared infrastructure logical network.

Fabrikam has chosen to create logical networks for each of the infrastructure services, as shown in Figure 2-5. You might choose to implement this differently, adding or removing logical networks from the design based on your requirements and the capabilities of your network infrastructure.

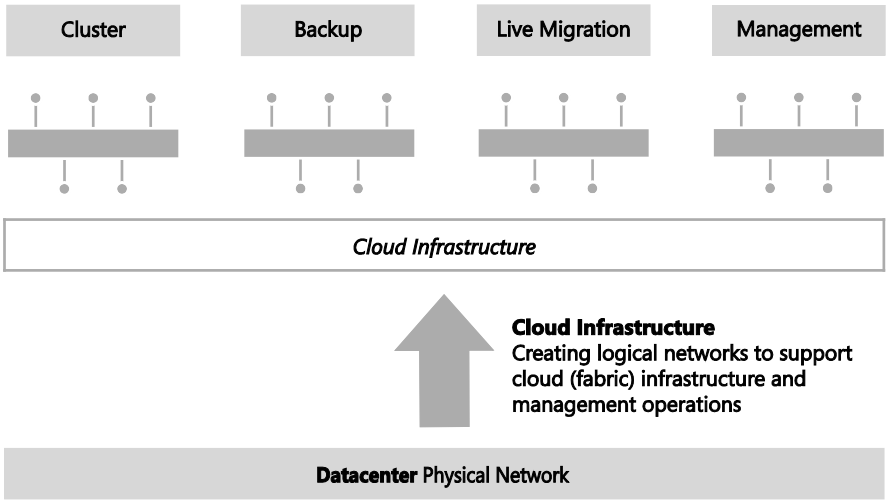


FIGURE 2-5 Using logical networks to differentiate cloud infrastructure services

Provider physical network

Fabrikam is a service provider (hoster) and offers hosted software and services, including web hosting, application hosting, messaging, collaboration, and platform infrastructure, to its end customers. The Provider network is dedicated to and used exclusively for customer (tenant) network traffic. The physical separation between customer network traffic on the Provider network and internal traffic on the Datacenter physical network improves security, simplifies management, and removes any potential competition between customer and corporate (internal) workloads.

MORE INFO **Some organizations are primarily service providers and others are enterprise customers who provide hosted software and services internally or externally to their customers or business partners. Planning to separate internal or cloud infrastructure and customer (tenant) workloads is relevant to both types of organization. You can find more details at** *http://technet.microsoft.com/en-us/library/hh831441.aspx.*

In designing a logical network solution for a provider network such as the one at Fabrikam, you should first consider the compute models the organization intends to support. Essentially, this means determining whether workloads from multiple customers will run on the same physical hardware (shared compute), if certain host computers and resources will be dedicated to a single customer (dedicated compute), or if both of these scenarios will be supported. A good starting point for the design is a single logical network for the shared compute workloads and a separate logical network for each customer that uses dedicated resources.

For customers with dedicated resources, host groups and network sites in VMM will associate the logical network with the host computers within each physical location that is allocated to (reserved for) that customer. This is covered in much more detail in Chapter 6.

Fabrikam, like many service providers, allows customers to choose which of these approaches works best for them. Customer workloads may be hosted on either shared or dedicated resources, with the latter attracting a price premium. Two customers have opted for physical servers dedicated to their workloads, with the remainder utilizing the shared compute model. The logical network design to support this model of operation (ignoring isolation) is shown in Figure 2-6. The design assumes that as new customers with dedicated compute requirements are onboarded to the service, additional logical networks will be added to the solution.

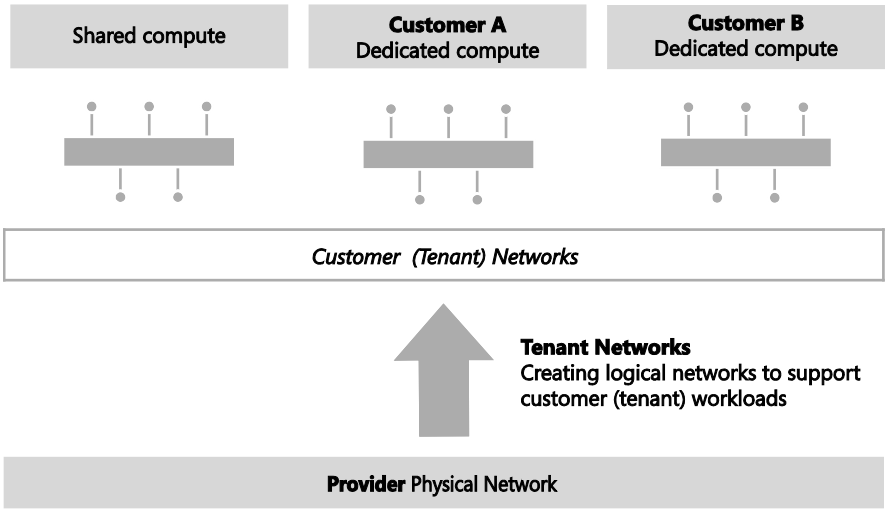


FIGURE 2-6 Logical networks for a mixed shared and dedicated compute model

In reality and as discussed for the Datacenter network, you may need to extend this initial model, breaking out (defining) additional logical networks to support a specific hosted service or to support a specific business or customer requirement.

Storage physical network

The final physical network for Fabrikam is dedicated to a single purpose: providing access to shared storage. A single logical network that maps directly to the physical network (as initially conceived in Step 1 of this process) is therefore quite appropriate. If Fabrikam were to use multiple IP-based storage technologies (such as iSCSI and SMB) on the physical network, they may decide to allocate each type of storage its own logical network, but this is not required.

Step 3: Determine isolation requirements

At the end of Step 2, you should have arrived at a set of candidate logical networks for each

physical network in your environment. The next step is to review the isolation requirements for each logical network you have identified so far, something which is clearly an important consideration for service providers hosting external customer workloads and enterprise customers needing to isolate network traffic from certain business groups or restricted (special) projects. These security requirements might lead you to create additional logical networks or at least further refine your logical network design.

To understand this concept, consider the basic assumption stated earlier: computers that connect to and use the same network should be able to communicate with each other through routers that connect different networks together, enabling inter-network communication. This principle holds in most cases. Indeed, where there is a business need to split off or otherwise isolate certain workloads for security, to improve performance, or simply to facilitate more effective control of network traffic, the best solution is to create a new network, either physically or via virtual networks (VLAN or PVLAN technology), place all of the appropriate computers and services on that network, and update the network routing tables and security policy to facilitate inter-network communication. This approach will be familiar to both enterprise customers and service providers, with the latter often using dedicated VLANs and PVLANs to isolate different customers from one another.

Historically, logical networks were used to model this behavior. However, each customer (tenant) supported by a service provider invariably needed an individual logical network—in some cases multiple logical networks—and as a result, service providers often needed to create hundreds if not thousands of them to support their customer base. As each logical network has a direct relationship with and needs to be assigned to network adapters in each (Hyper-V) host computer, the outcome was performance issues, increased cost, and management complexity.

The VM network concept introduced in VMM was designed to address this particular problem. Instead of connecting directly to a logical network, VMs connect to a VM network, with the VM network acting as an interface to a particular logical network, as shown in Figure 2-7. Since each VM network is associated with a logical network and not a physical host computer, adding, deleting, and making changes to VM networks is much simpler than making changes to logical networks.

Multiple VM networks may be associated with the same logical network, with VMs connected to one VM network being unaware of and completely isolated from any of the others—this isolation providing the foundation needed for service providers to offer customers services such as bring-your-own-IP and bring-your-own-network.

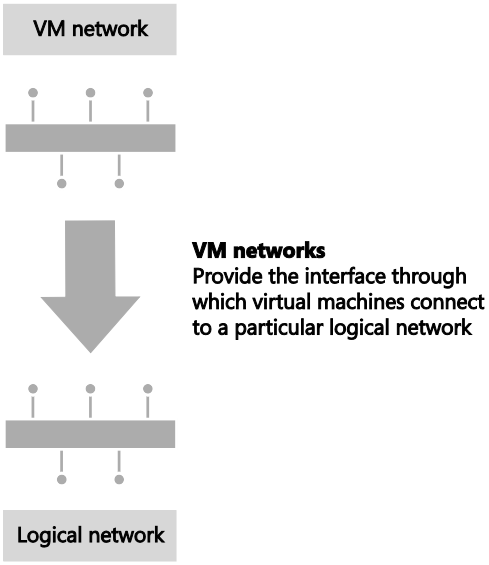


FIGURE 2-7 VM networks' relationship to logical networks

In VMM, you can isolate VM networks by using either traditional VLAN (or isolated PVLAN) solutions or by implementing network virtualization. The latter option addresses the scale limitations associated with a traditional VLAN solution.

You cannot mix and match different types of network isolation on the same logical network. It's impossible, for example, to isolate some VM networks by using VLAN/PVLAN technology and others by using network virtualization. Should you need to use multiple approaches in your environment, you will need to return to Step 2 above and create a separate logical network for each isolation method.

MORE INFO **There is a practical limit of approximately 2,000 tenants and 4,000 VM networks per VMM server. If you expect to approach either of these scale limitations, you will most likely need to introduce additional VMM servers and use Service Provider Foundation to manage this environment. You should follow the same process described in this section to identify and create logical and VM networks for each VMM server you deploy. You can find more information on Service Provider Foundation at** *http://technet.microsoft.com/en-us/library/jj642895.aspx*.

No isolation

Isolation is necessary only in cases where a logical network will be used by multiple customers (tenants). Logical networks created for corporate (internal) workloads, cloud infrastructure services, and logical networks that are *dedicated* to a specific customer are all single tenant, meaning that traffic isolation is optional.

As mentioned earlier, at least one VM network is required for logical networks that will be accessed by VMs. If there is no need to isolate network traffic on the logical network, only a single VM network, as shown in Figure 2-8, is required. The VM network in this example simply acts as a "pass through" to the logical network of the same name.

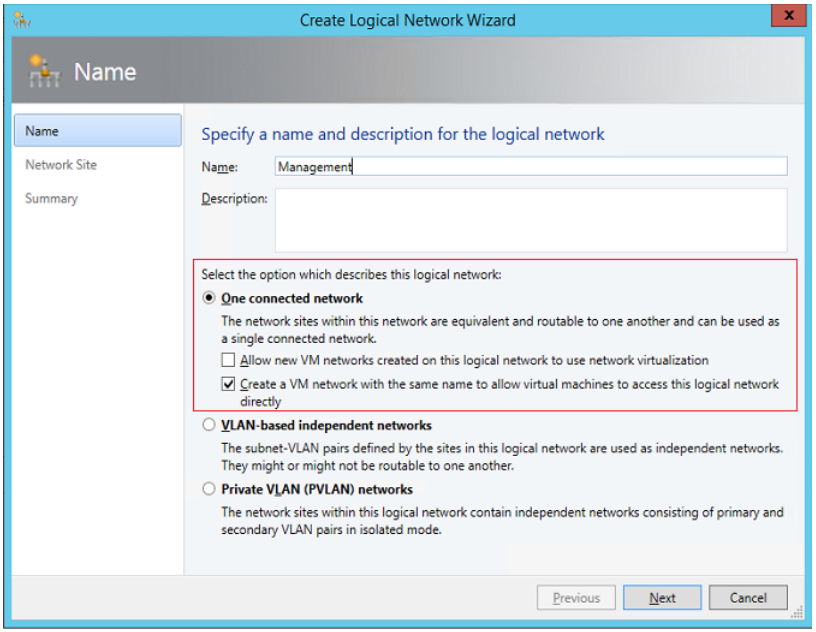


FIGURE 2-8 Creating a VM network with no isolation

This configuration establishes a one-to-one mapping between the VM network and the logical network, as shown in Figure 2-9. As a result, only one VM network per logical network can be configured for No Isolation. If VMs that connect to this VM network should be restricted from communicating with each other, you may need to consider breaking out an additional logical network to accommodate this requirement.

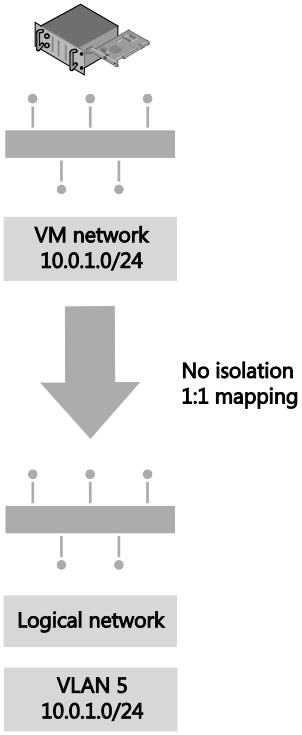


FIGURE 2-9 VM network direct access to the logical network

NOTE **For logical networks that will not be used by VMs, generally those dedicated to infrastructure services like storage and live migration, you may not need to create VM networks at all.**

VLAN isolation

As discussed earlier, VMs in VMM connect to a VM network, which acts as an interface to a particular logical network. Instead of creating a separate logical network for each customer that will be isolated from others using VLAN technology, you should create a single logical network for all of these customers, configuring the properties of the network, as shown in Figure 2-10, for VLAN-based independent networks. The VM network you associate with the logical network can be allocated a friendly name to clearly identify its purpose and which customer has access to it. You can also apply access control to restrict who can use it.

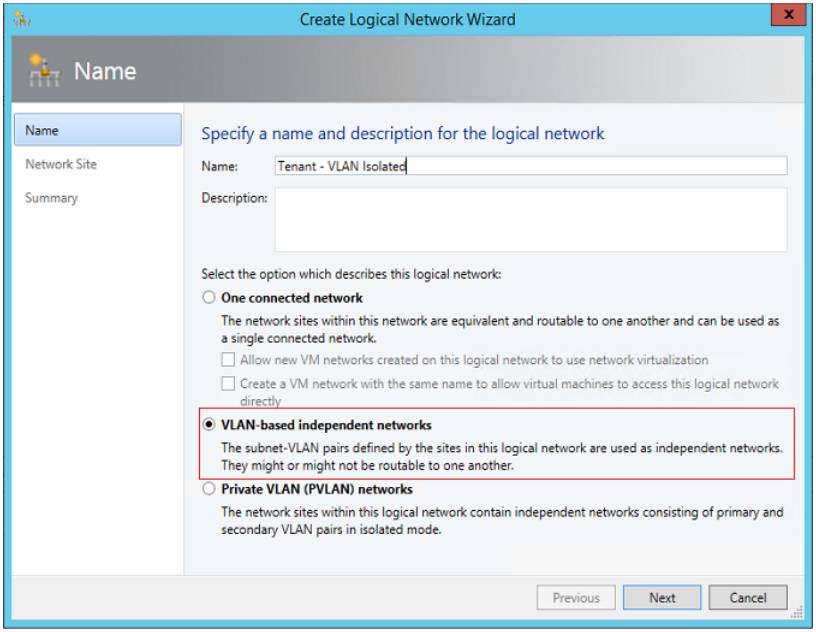


FIGURE 2-10 Configuring a logical network for VLAN isolation

Each VLAN must be allocated to a network site. Multiple VLANs can exist with the same site, as shown in Figure 2-11, but each one will be totally isolated from any of the others.

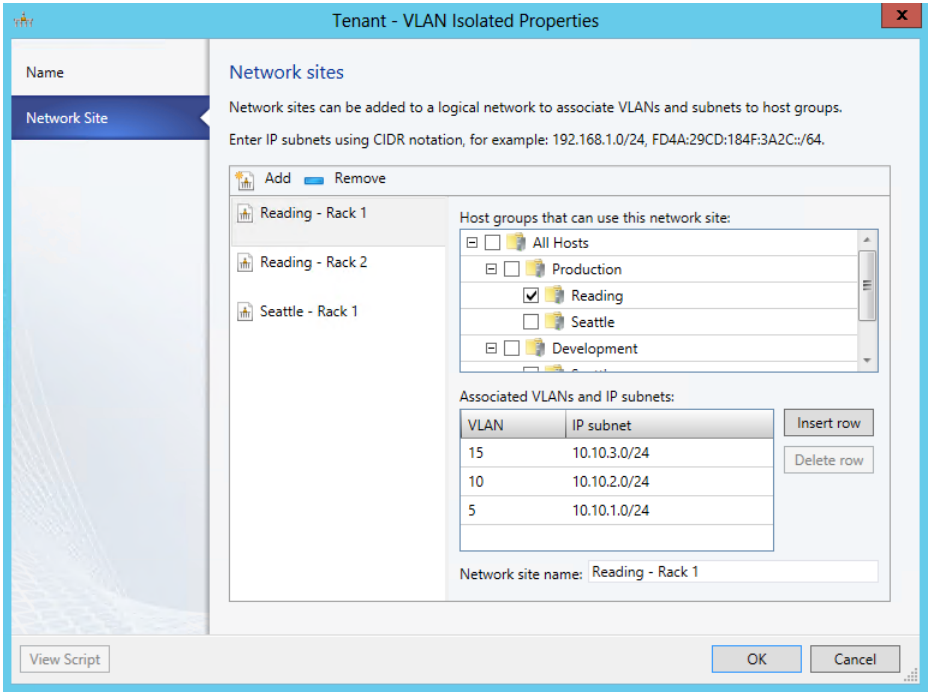


FIGURE 2-11 Defining network sites for VLAN isolation

Finally, VM networks need to be created to allow customer VMs to connect to and use the logical network. Each VM network you create is directly mapped to exactly one of the subnet VLANs that have been defined for a site in that logical network. As a result, you can have only as many VM networks as you have subnet VLANs. The Create VM Network Wizard, shown in Figure 2-12, will display only those network sites that have not already been allocated to an existing VM network.

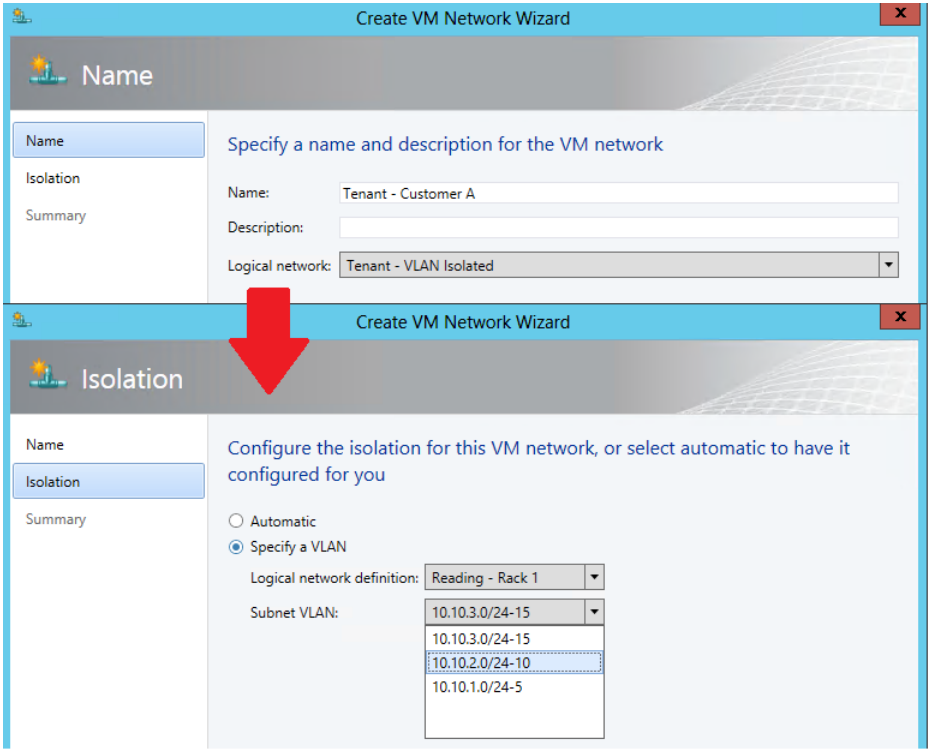


FIGURE 2-12 Allocating a VLAN (network site) to a VM network

Although you can manually choose which VLAN should be allocated to a VM network, VMM also provides for automatic assignment. This is useful where customers are allocated a VLAN from a pool rather than being given an assigned VLAN. In these cases, a VLAN is randomly assigned from the pool when you define a new VM network and is returned and available for re-use when that VM network is deleted. Note that once all of the available network sites have been allocated, no further VM networks may be linked to this logical network until additional VLANs are added or some of the existing VM networks are deleted.

To briefly summarize, create a single logical network by selecting the VLAN-Based Independent Networks option, create sites, and then specify the list of VLANs that exist in each site. Either create a VM network to represent each VLAN or create VM networks as needed using automatic assignment to allocate a network site (VLAN) to that VM network. The net result should be a one-to-one mapping between the VM network and the network site, as shown in Figure 2-13.

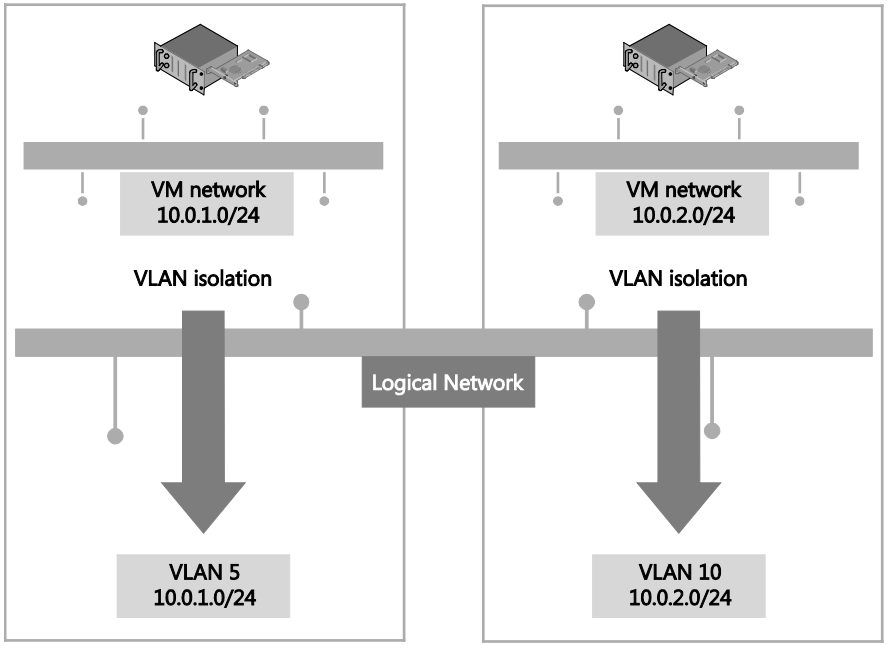


FIGURE 2-13 Logical network design for VLAN isolation

There are a number of limitations to using VLANs to isolate network traffic, most significantly the scalability limits. Only 4,095 VLANs are permitted per physical network. PVLANs may be used to work around this limitation, but at a cost of increased complexity. The cost of management, level of complexity, and the risk of error also increase significantly at high scale. These issues may not be of direct relevance to enterprise customers since, in general, they do not need to manage very large numbers of networks at this scale, but these are major considerations for service providers that provide hosted services to a large number of external customers.

VLAN isolation is expected to remain common practice in many enterprise deployments given its relative simplicity and ease of management at smaller scale. Service providers (hosters), however, can be expected to use alternative isolation technologies to help work around VLAN scale limitations given their need to manage a much larger number of networks.

PVLAN isolation

PVLANs are often used by service providers (hosters) to work around the scale limitations of VLANs. They essentially allow network administrators to divide a VLAN into a number of separate and isolated sub-networks, which can then be allocated to individual customers (tenants). PVLANs share the IP subnet that was allocated to the parent VLAN, as you might expect, but they require a router to communicate with each other and with resources on any other network.

A PVLAN consists of a primary and secondary VLAN pair. Each machine that is part of a PVLAN pair can be configured in one of three modes, as shown in Figure 2-14. In promiscuous mode, hosts are on the primary VLAN and can communicate directly with resources on both the primary and secondary VLANs. In community mode, the secondary VLAN represents a community. Direct communication is permitted only with hosts in the same community and those that are connected to the primary PVLAN in promiscuous mode. In isolated mode, direct communication is permitted only with promiscuous resources on the primary PVLAN.

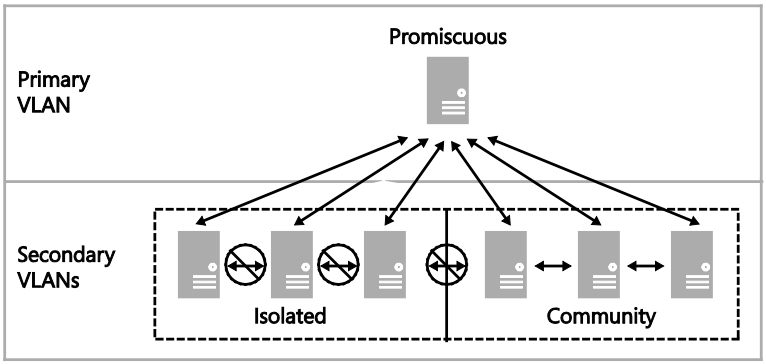


FIGURE 2-14 The three modes for PVLAN isolation

Hyper-V supports all modes of PVLAN isolation shown in Figure 2-14, but VMM supports only isolated mode. It has no concept of primary (promiscuous) or community modes. In practice, this means that a VM connected to a PVLAN in this release is completely isolated from any other resources on the network. The only device it can directly communicate with is the default IP gateway.

While this may feel like a severe limitation, a number of scenarios work quite well in this configuration, the most common example of which is front end web servers. In this specific scenario, all of the web servers in a web farm are placed on a single network subnet but are otherwise completely isolated from each other, PVLANs in this context, helping to simplify management and improve overall security.

NOTE **Similar functionality to community mode can be achieved by adding an additional network adapter to the VM and connecting this adapter to a VM network on which network virtualization has been enabled and to which all of the other community resources are also connected.**

To implement PVLAN isolation in VMM, you create a single logical network, configuring the properties of the network for PVLAN networks, as shown in Figure 2-15, and then you create VM networks. Each VM network you associate with the logical network can be allocated a friendly name, as mentioned previously, to clearly identify its purpose and which customer has access to it. You can also apply access control to restrict who can use it.

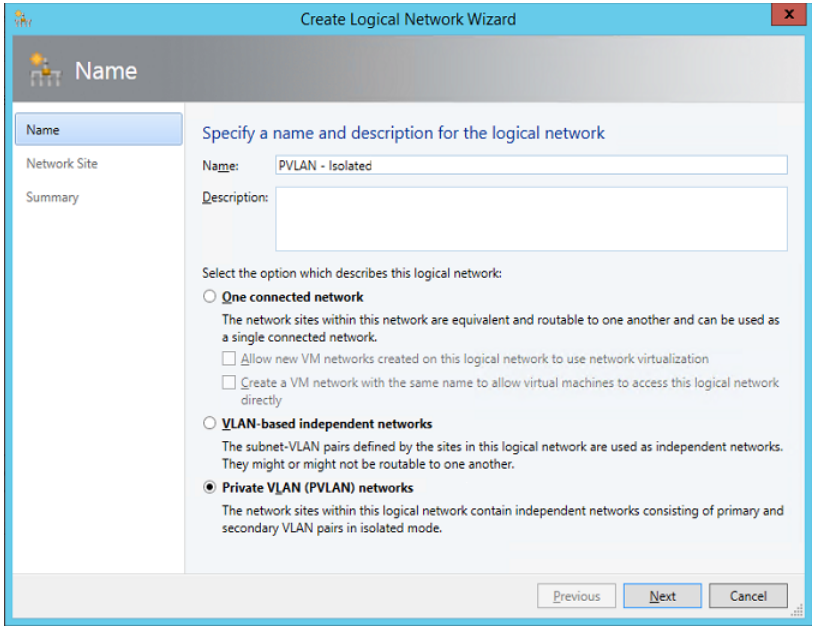


FIGURE 2-15 Enabling PVLAN isolation

The Network Site page of the Create Logical Network Wizard includes a subtle but important difference for PVLANs. In addition to the primary VLAN, the Associated VLANs and IP Subnets section contains an additional column called Secondary VLAN. You should associate each primary VLAN and secondary PVLAN with a network site within the logical network, as shown in Figure 2-16. You can also define multiple PVLANs in the same network site as needed.

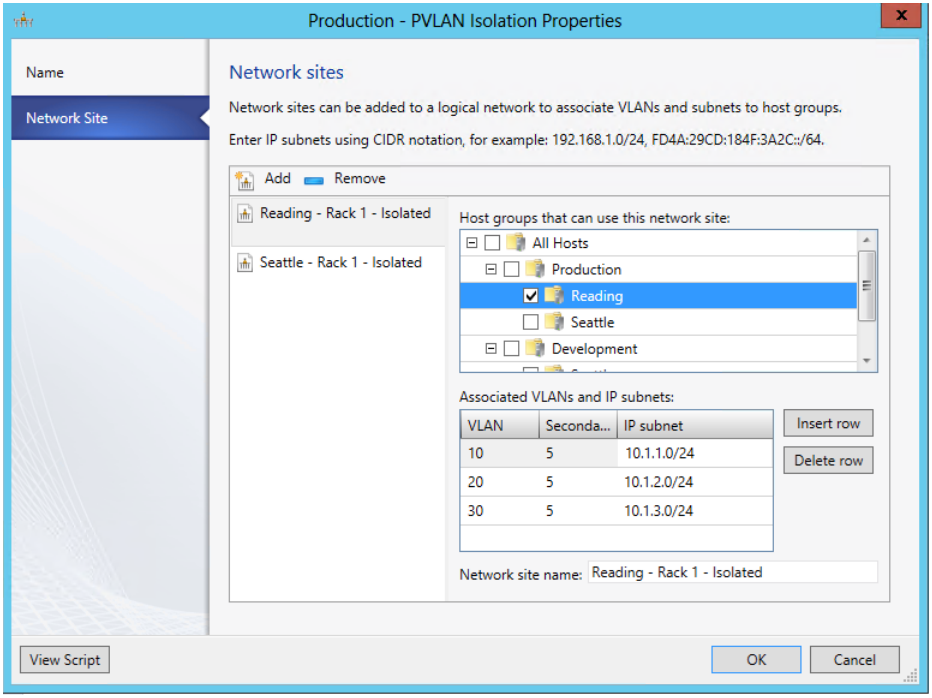


FIGURE 2-16 Network site configured for PVLAN isolation

Only one PVLAN can be in isolated mode per primary VLAN, and you should take care to ensure that a different primary VLAN ID is used in each network site you create. The ID you use for the PVLAN, however, may be the same in each site. In fact, using the same ID for the isolated PVLAN is actually recommended to ensure consistency and simplify management.

As discussed, VM networks are needed for VMs to connect to and use the logical network. Each VM network you create is directly mapped to exactly one of the PVLANs that have been defined for that logical network. As a result, you can have only as many VM networks as you have defined PVLANs. As shown in Figure 2-17, the Create VM Network Wizard displays only those PVLANs that have not already been allocated to an existing VM network. Note that the wizard does not offer the option for automatic assignment even though the UI suggests that this is possible.

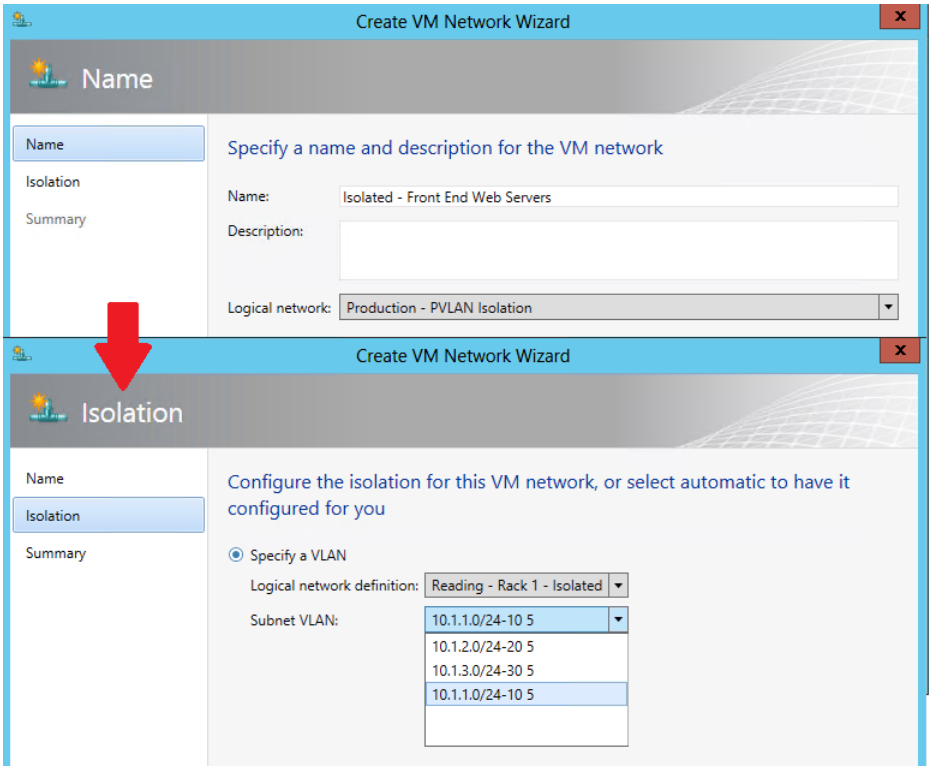


FIGURE 2-17 Allocating a PVLAN (network site) to a VM network

To briefly summarize, create a single logical network by selecting the Private VLAN (PVLAN) Networks option, create network sites, defining primary and secondary VLAN pairs, and create VM networks for each one, as shown in Figure 2-18. In this example, PVLAN 5 is designated as the isolated PVLAN for consistency across all primary VLANs. Your implementation may be different.

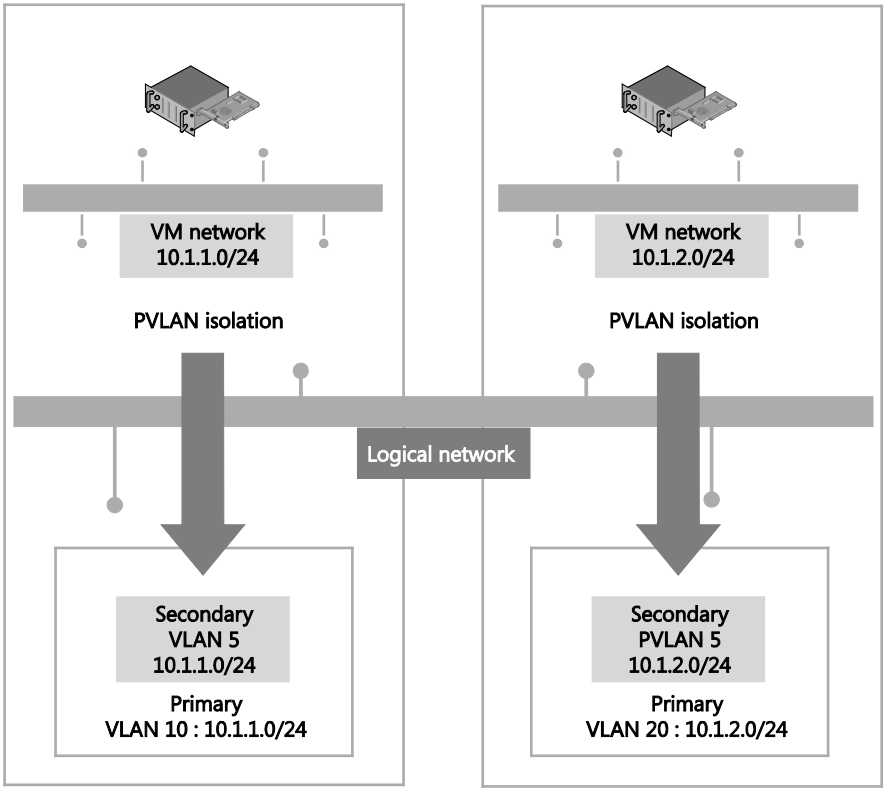


FIGURE 2-18 Logical network design for PVLAN isolation

Although each VM you connect to one of these VM networks will be assigned an IP address from the same subnet, it will be able to communicate only with the default IP gateway or with other network devices in promiscuous mode. Note that devices in promiscuous mode *must* be set up and configured outside of VMM. If all of the VMs are present on the same physical host, isolation will be enforced through the Hyper-V extensible switch. Otherwise you will need to make sure that each of the PVLANs you define in VMM are also configured for isolation mode on the physical switch. To avoid potential IP conflicts with resources that exist on the primary VLAN (and any community VLANs that were created outside of VMM), it is recommended that you reserve a set of IP addresses and create a separate IP pool for each PVLAN. The IP addresses you reserve must be part of the IP subnet that was allocated to the primary VLAN.

Network virtualization

Network virtualization provides administrators with the ability to create multiple virtual networks on a shared physical network. In this approach to isolation, each tenant receives a complete virtual network, which includes support for virtual subnets and virtual routing. Tenants can even use their own IP addresses and subnets in these virtual networks, even if these conflict with or overlap with those used by other tenants. Further, since virtual networks

are defined entirely in software, it is unnecessary to reconfigure the physical network (unlike VLAN and PVLAN solutions) to onboard or remove tenant networks or to make changes to reflect new business requirements.

NOTE **You can find more details on this approach at** *http://blogs.technet.com/b/windowsserver/archive/2012/08/22/software-defined-networking-enabled-in-windows-server-2012-and-system-center-2012-sp1-virtual-machine-manager.aspx*.

In Figure 2-19, Tenant A has two virtual subnets. Windows Server automatically creates a virtual router that connects the two subnets for this tenant and allows VMs on each subnet to communicate with each other. Tenant B has a single virtual subnet but still has its own virtual router. The virtual subnet ID and routing domain ID shown in the diagram are used by Hyper-V host computers to differentiate network traffic and routing for each of the tenants.

NOTE **The virtual router does not exist on any one host. It essentially spans all hosts that contain VMs that are part of a particular VM network.**

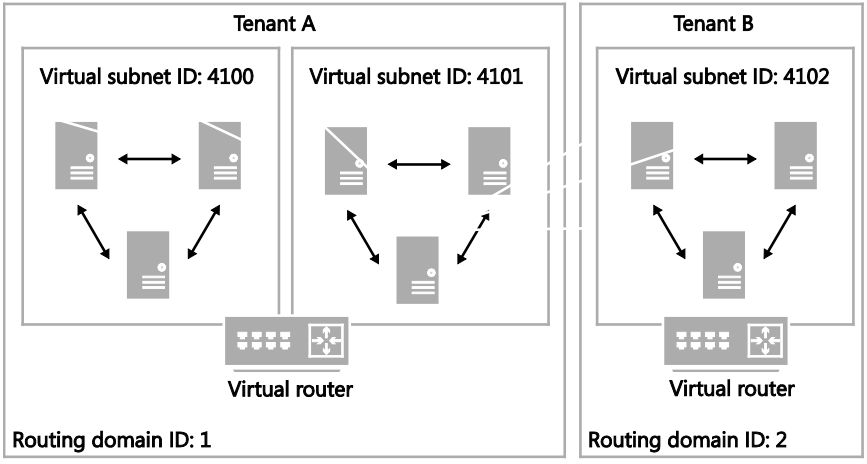


FIGURE 2-19 Logical network design for isolation using network virtualization

When using network virtualization, create a logical network by selecting the One Connected Network option and then selecting Allow New VM Networks Created On This Logical Network To Use Network Virtualization, as shown in Figure 2-20.

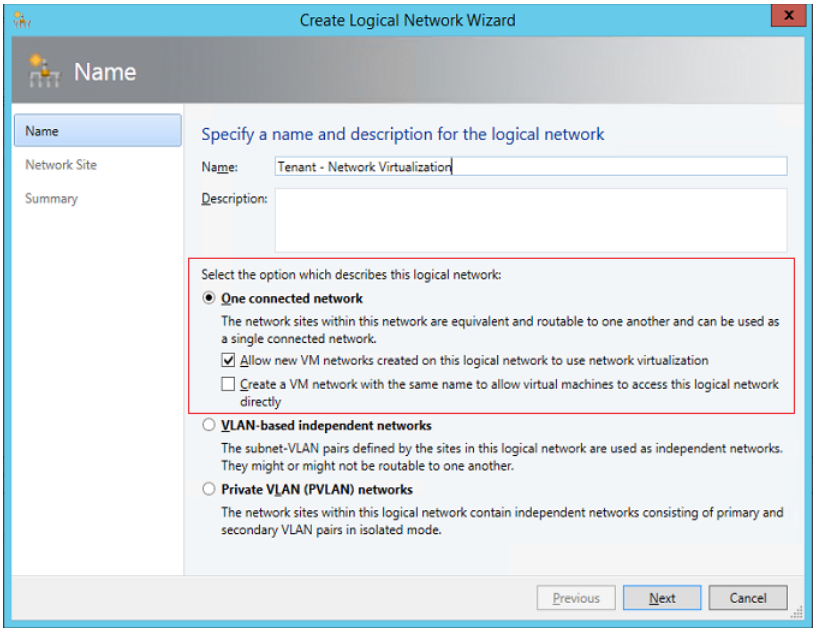


FIGURE 2-20 Configuring a logical network to support network virtualization

You need to create network sites to define the VLANs and IP subnets that are to be associated with the logical network in each physical location. Assuming you specify VLANs in your network sites, the physical network must be able to route network traffic between them. The VLANs in this case are used by the network administrator for ease of management and to control broadcast traffic; they are not used as an isolation mechanism. Note that these VLANs exist on the Hyper-V host server Parent Partition only—tenant VMs are unable to gain access to them.

An IP pool must be associated with every single network site linked to the logical network, as shown in Figure 2-21. The IP addresses from these pools, also known as provider address (PA) pools, must also be routable between all of the Hyper-V hosts associated with the logical network.

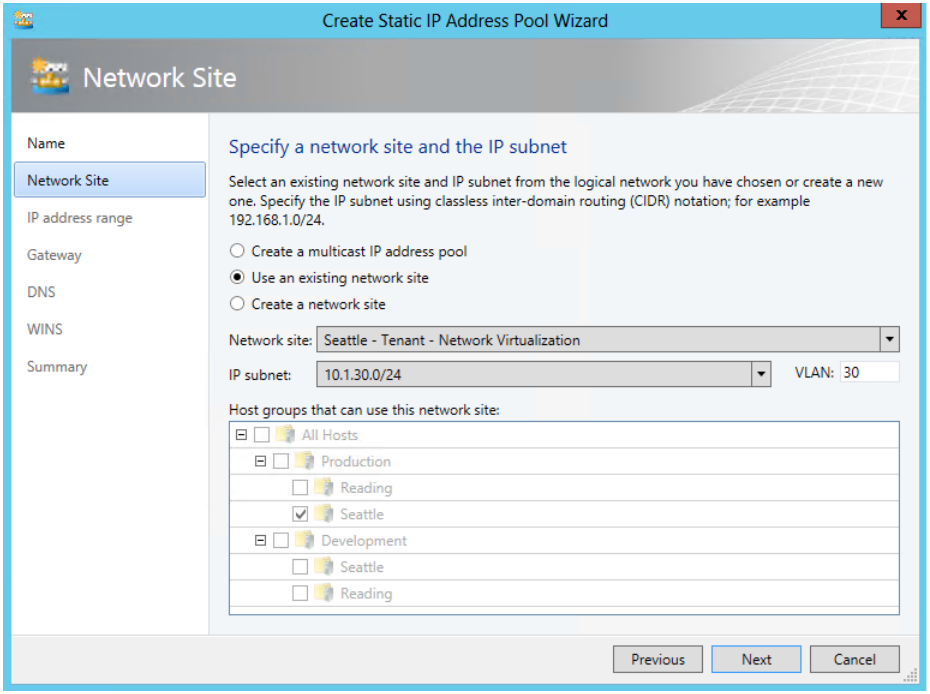


FIGURE 2-21 Defining a provider address IP pool for a network site

As with other isolation methods, you will also need to create VM networks to allow customer VMs to connect to and use the logical network, and you should define a separate VM network for each tenant, with each one of these VM networks configured to isolate using Hyper-V Network Virtualization, as shown in Figure 2-22. You can also select No Isolation if you want the VM network to provide VMs with direct access to the logical network. The option to enable isolation shown in Figure 2-22 is only available when provider address IP pools have been defined for the IP protocol (IPv4 or IPv6) supported by the logical network, as mentioned earlier.

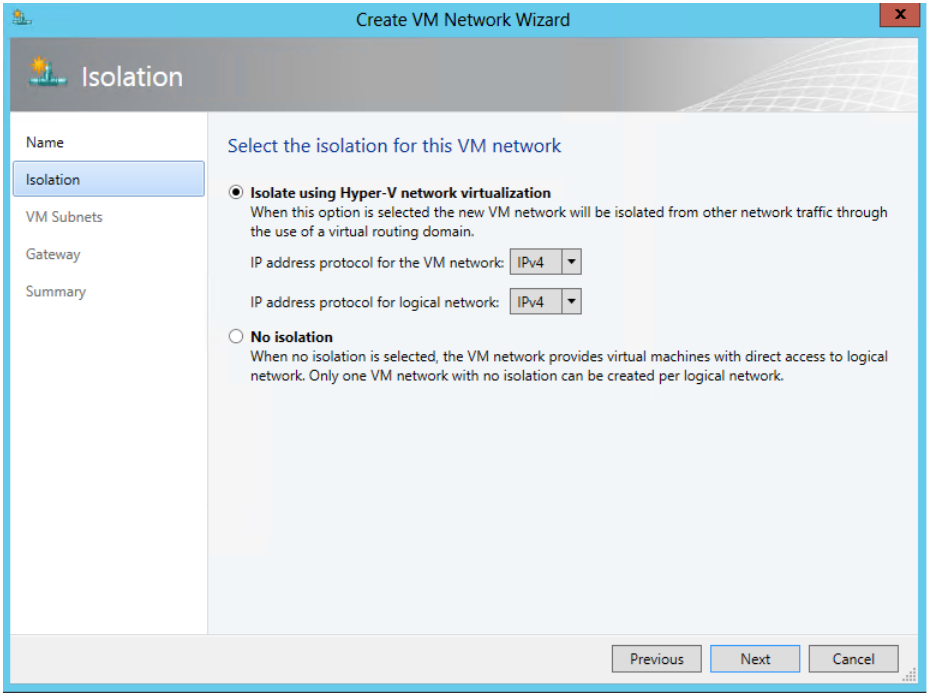


FIGURE 2-22 VM network isolation using Hyper-V Network Virtualization

You also need to define the IP subnets for each VM network, setting out the IP addresses that will be used by VMs connected to that network, as shown in Figure 2-23. These addresses, known as the consumer addresses (CA), are completely separate from any other tenant and from the logical network. Tenants can therefore use their own IP addresses and subnets in their virtual networks, even if these appear to conflict with or otherwise overlap with those used by other tenants. Again, each tenant may be allocated multiple subnets, as shown in Figure 2-23.

NOTE **VMM installs a DHCP Virtual Switch extension on each host that it manages. If a tenant's VM uses DHCP to request an IP address, the extension will respond by offering an IP address from the IP pool that has been defined for the VM network.**

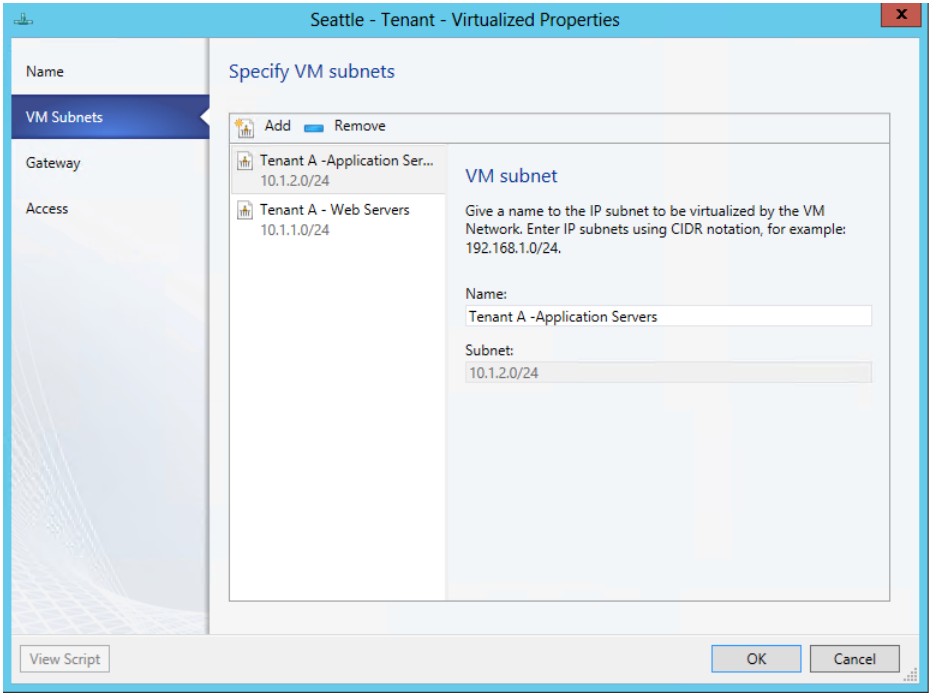


FIGURE 2-23 Defining consumer IP subnets

To summarize, create a logical network by selecting the One Connected Network option, and then select Allow New VM Networks Created On This Logical Network To Use Network Virtualization, and define network sites and IP pools for each location in which the network will be supported. You should then create VM networks for each tenant. The result should be a one-to-many mapping between the logical network and VM networks created to support each tenant, as shown in Figure 2-24.

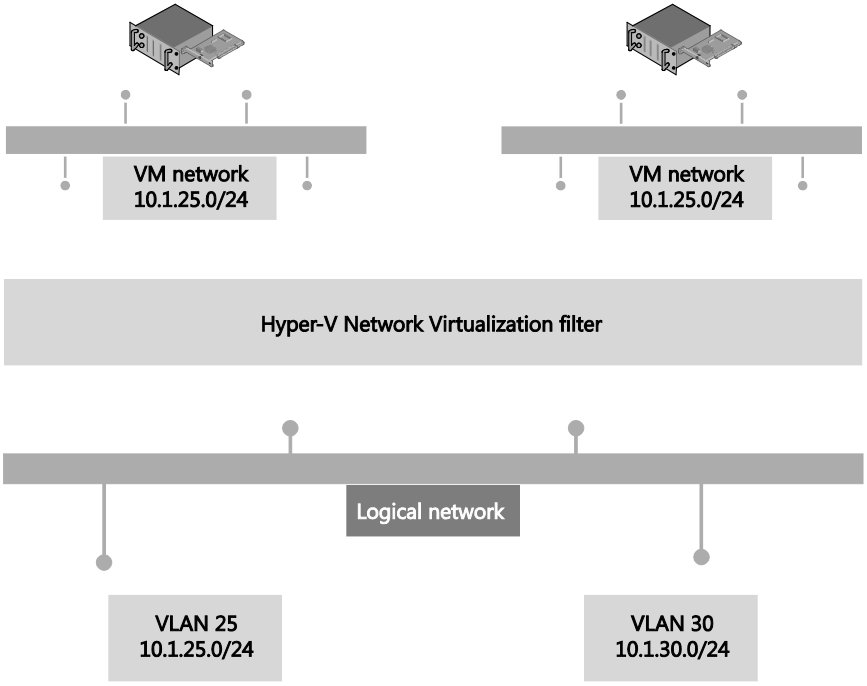


FIGURE 2-24 Logical network design for network virtualization

The virtual networks shown in Figure 2-24 have no external connectivity by default, meaning that VMs connected to them will be able to communicate only with other VMs on the same virtual network. You can use a Hyper-V gateway device (see Chapter 5, "Network Virtualization gateway") to provide connectivity to other networks.

Externally defined networks

Network administrators can also configure network settings or capabilities such as logical networks, network sites, and IP pools, by using a third-party (vendor) network management console. In this case, the VMM administrator uses a virtual switch extension manager to import the externally defined settings directly into VMM. This approach allows network specialists to focus on and define the logical network, leaving the VMM administrators free to concentrate on the VM networks and the services that are to be offered to end customers. In this context, the logical network becomes a "black box" to VMM administrators in that they can use networks imported through the virtual switch extension manager but have no insight into how the network is constructed, nor do they have any visibility into the method of network isolation that has been applied to a given network, as shown in Figure 2-25.

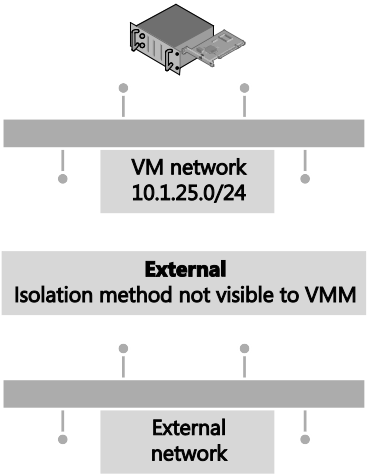


FIGURE 2-25 Externally defined networks with the isolation method invisible to VMM

Externally defined networks are included in this text only to note that VMM administrators need to work closely with their counterparts on the network team to make sure that a consistent model and design structure is being followed. Ideally, network administrators should plan the network configuration in partnership with VMM administrators to ensure that both parties agree on naming conventions and standards for how to define the fabric.

> ***See also*** *You can find more information on virtual switch extension managers in VMM and how to make use of them at http://technet.microsoft.com/en-us/library/jj614619.aspx.*

Key points

Considering the logical networks created for Fabrikam, there appears to be little or no requirement to isolate any of the logical networks defined on top of the Datacenter or Storage physical networks. That being said, you could easily justify using some form of isolation for front end web servers (assuming they were accessible from the public Internet) that were connected to the Datacenter network or for specialized servers and workloads that need to be isolated from others. You need to assess each logical network and determine what, if any, isolation methodologies you should apply in your environment.

The case for isolation for logical networks on the Provider network in the Fabrikam example is very clear, however, because there are multiple customers running workloads on the same physical infrastructure. Where a given physical network or VLANs have been dedicated to a particular customer, clearly no isolation will be required on the logical network since only that tenant's traffic will exist on the network. However, in the case of shared networks, you must consider which isolation method is best suited to the customers' requirements and is supported by the physical network. Network virtualization clearly offers the most comprehensive and scalable solution but requires NVGRE gateway devices to allow VMs to communicate with networks in the same datacenter or VPN gateway devices to facilitate communication with a defined external network. VLAN/PVLAN isolation can be readily used, is

well understood, and is supported by most existing network hardware, but has management issues at scale. The decision, ultimately, will be based on your business strategy, current and forecast growth patterns, and how quickly and easily you are able to move to software-defined networking.

Step 4: Define network sites

At this point in the process, you can start to consider implementation details, reviewing each of the logical networks that you identified during the earlier parts of the process to decide where (that is, in which physical location) they need to be deployed and to determine the set of network sites, IP pools, and MAC address pools needed to make the networks available and used in those locations.

Physical locations and host servers

It makes sense to make some logical networks available in all physical locations. Many of the cloud infrastructure networks, such as management, storage, and live migration, clearly are relevant to and should be made available everywhere while the availability of others should be restricted to specified locations. If your organization's development team is located in a single location, it might be reasonable to ensure that the logical network you create for development workloads would be available only on servers in that location.

Having identified the physical locations on which a given logical network will be available, the next decision point is which of the Hyper-V hosts in that location should be configured to support it. Again, it makes sense that some of the logical networks should be made available on all of the host computers in that location, the logical networks you define for cloud infrastructure such as Management and Storage clearly being the most obvious candidates, though there may well be exceptions to this general rule. For example, not all servers will have access to network-attached storage.

Servers may have been set aside for specific workloads or projects or allocated (dedicated) to specific tenants, and in these cases, you should ensure that the logical network that will carry those workloads is available only on those computers. To achieve this, you first need to define host groups. The recommended approach is to create a parent host group (for example, Production, as shown in Figure 2-26) that clearly identifies the group of dedicated servers and child host groups for each physical site where those servers will be located; Reading and Seattle in Figure 2-26 for example.

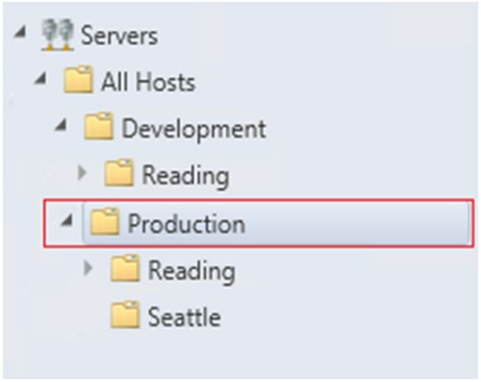


FIGURE 2-26 Creating host groups for dedicated servers

Network adapters in Hyper-V servers can be associated with multiple logical networks, but there is no internal routing between these networks by default. If you want to allow VMs and host services configured on one logical network to communicate with those on another, you must deploy a router or gateway device.

Network sites

Network sites, otherwise known as logical network definitions, are used to define the VLANs and IP subnets that are to be associated with the logical network in each physical location. However, it is unnecessary to define network sites for all of your logical networks. The following key points replicated from the Configuring Logical Networking in VMM Overview section of the VMM documentation (available at *http://technet.microsoft.com/en-us/library/jj721568.aspx*) set out the guidelines that will help you determine whether you need to define a network site for the logical network in a given physical location:

- If you want to use DHCP that is already available on the network, and you are not using VLANs, you *do not* have to create any network sites, but as a recommended best practice, you should always aim to do so.
- If using VM networks that use network virtualization, you must create at least one network site and associate at least one IP subnet with the site, as mentioned earlier. You can also assign a VLAN to the network site, as appropriate. Creating a network site with an IP subnet makes it possible to create an IP address pool for the logical network, which is necessary for network virtualization.

If the VM networks you create will not use network virtualization as an isolation mechanism, the following guidance applies:

- If you plan to use a load balancer that is managed by VMM to load balance a service tier, create at least one network site and associate at least one IP subnet with the network site.
- If you want to create static IP address pools that VMM manages, create at least one network site and associate at least one IP subnet with the network site.

- If you want to use DHCP that is already available on the network to assign IP addresses to virtual devices in a specified VLAN, create network sites with only VLANs assigned to them. With that said, it is strongly recommended that you fill in all the network properties in VMM, even if you're not going to use VMM for IP address assignment and management.

IP address pools

If you associate one or more IP subnets with a network site, you can also create static IP address pools for those subnets. Static IP address pools make it possible for VMM to automatically allocate static IP addresses to Windows-based VMs that are running on any managed Hyper-V, VMware ESX, or Citrix XenServer host. VMM can automatically assign static IP addresses from the pool to stand-alone VMs, to VMs that are deployed as part of a service, and to physical computers when you use VMM to deploy them as Hyper-V hosts. Additionally, when you create a static IP address pool, you can define a reserved range of IP addresses that can be assigned to load balancers as virtual IP addresses. VMM automatically assigns a virtual IP address to a load balancer during the deployment of a load-balanced service tier.

It is unnecessary to define IP address pools for all of your network sites. The following key points replicated from the VMM documentation at *http://technet.microsoft.com/en-us/library/jj721568.aspx* set out the guidelines that will help you determine whether you need to do so:

- If your network configuration includes VM networks that use network virtualization, you must create IP address pools on both the logical network that provides the foundation for those VM networks and on the VM networks themselves. If the VMs on the VM networks are configured to use DHCP, VMM will respond to the DHCP request with an address from an IP address pool.
- If you are using a VLAN-based network configuration, you can use either DHCP, if it is available, and/or IP address pools. To use IP address pools, create them on the logical network. They will automatically become available on the VM network.
- If you have a VM network that gives direct access to the underlying logical network, you can use either DHCP, if it is available, and/or IP address pools for that network. To use IP address pools, create them on the logical network. They will automatically become available on the VM network.

If you are using external networks that are implemented through a vendor network-management console (in other words, if you will use a virtual switch extension manager), your IP address pools will be imported from the vendor network-management database. Therefore, do not create IP address pools in VMM.

You can also add an IP Address Management (IPAM) server to the resources in VMM. The IP address settings that are associated with logical networks and VM networks made in VMM will be used to automatically update the settings stored in the IPAM server.

MAC address pools

If a VM connected to the logical network will obtain IP addresses from a static IP address pool, you must also configure the VM to use a static MAC address. You can either specify the MAC address manually or have VMM automatically assign a MAC address from either a central MAC address pool or one that you have created for a specific network site.

Step 5: Deployment

Having defined the logical network, the host groups, network sites, IP address pools, and, optionally, MAC address pools, the next step is to associate the network with the Hyper-V host computers. Although you can associate logical networks with each Hyper-V host manually or by using Windows PowerShell, to ensure consistency and simplify management across multiple hosts, it is far more efficient to define the required properties and capabilities within port profiles and logical switches. You'll find the details for this process in Chapter 6.

The default logical network

At least one logical network must be associated with a given host computer for it to support deployed VMs and services. To help ensure this is the case, VMM verifies that physical network adapters on all new host computers are associated with one or more logical networks. If no such association exists, VMM checks to see if a logical network exists with the same name as the first DNS suffix label on each network adapter. For example, in the case of a server called REA-HST-01.Corp.fabrikam.com, VMM would expect to find a pre-created logical network called Corp. If it does find a match, VMM will automatically associate the host network adapter with the selected logical network. If it does not, VMM will create a new logical network with that name (Corp) and make the necessary association with the host.

IMPORTANT **If the new host computer is connected to a number of different physical networks, VMM could potentially create a new logical network for every physical network the host is connected to.**

In a test or proof-of-concept environment, this type of behavior is perfectly acceptable since you want to get up and running as quickly and easily as possible. If you follow the guidance in this rest of this chapter, however, you will have carefully planned and structured your environment and will want to purposely associate a host with the required logical networks rather than rely on any default behaviors. Therefore, turning off this setting is recommended (as in Figure 2-27).

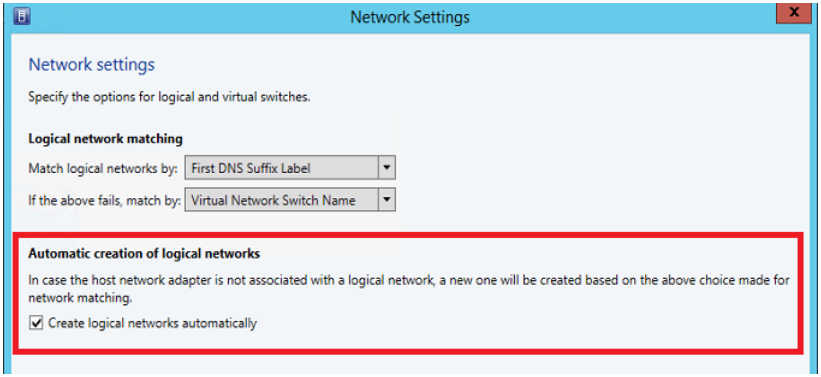


FIGURE 2-27 Turning off automatic logical network creation

If you leave this setting on initially, you can turn it off later, but be aware that VMM will not allow you to delete any default logical networks that have existing associations with network adapters in your host computers. You will need to associate these adapters with different logical networks first. You may also have to remove VM networks and any other objects that have dependencies on these logical networks before you can successfully delete them.

Naming conventions

As with everything else, defining and adhering to a naming convention for all the features of your virtualized networking solution is important. Logical network names should, as much as possible, help administrators clearly identify the main purpose and function and use a structure similar to the following:

[Environment] – [Optional Purpose]

Typical examples of this structure would be Production – Corporate, Production – Tenant, Development – Test, and so on. It is also strongly recommended that you add a high-level description to aid understanding.

Although network sites (logical network definitions) are created in the context of a logical network, naming conventions for these objects become particularly important when you start to use uplink port profiles and logical switches. Recall that when you create an uplink port profile, you select the network sites that represent connection to the required logical network. For example, if you have multiple sites called Reading, which one is linked to the logical network used for your corporate servers, which one is used by tenants, and which represents a connection to your development environment? Although the UI displays logical network names when you configure an uplink port profile, you need to pay close attention to make

sure you choose the correct one, and hence a poor naming convention can lead to potential misconfiguration. As a result, it is recommended that you use a naming convention for network sites similar to the following:

[Location] - [Logical Network Name]

Similarly, consider and arrive at an appropriate naming convention for the IP address and MAC address pools that are directly linked to the network site to help provide clarity around what a given address pool is used for. The following is a good starting point:

[Location] – [Logical Network Name] - [Purpose]

These are the recommended naming conventions for the logical network and the various features that depend on it, but this approach may not be appropriate for your specific environment. The point is, you need to arrive at a convention that clearly identifies the key features of the solution in your environment and what they are used for.

CHAPTER 3

Hyper-V port profiles

Hyper-V port profiles (and logical switches) act as containers, essentially templates, for the properties and capabilities to be applied to network adapters. Using these concepts, you can consistently apply the same settings and capabilities to network adapters across multiple hosts.

This chapter will:

- Review the role of Hyper-V port profiles in a virtualized network solution
- Discuss the different types of Hyper-V port profiles and how they are used
- Explain how to identify the set of Hyper-V port profiles you need in your environment
- Discuss the use of port classifications to hide implementation details

Uplink port profiles

There are two types of Hyper-V port profiles in Microsoft System Center Virtual Machine Manager (VMM): uplink port profiles and virtual network adapter port profiles. Uplink port profiles are applied to physical network adapters as part of logical switch deployment and define the set of logical networks that should be associated with those network adapters. They also specify how multiple network adapters in a given host computer using the same uplink port profile should be teamed. Virtual network adapter port profiles, in contrast, are applied to virtual network adapters and define specific capabilities, such as bandwidth limitations, priority, security settings, and so on.

If you have a simple environment that consists of a single physical network in one location, and if all host computers are configured the same way, have the same requirements, and use the same protocols for network adapter teaming, then a single uplink port profile might be all you need in your environment. In practice, however, such an environment is rare outside of a small business or test lab, and even then, the need to scope or restrict certain logical networks to a specific group of host computers can lead you to create multiple uplink port profiles. A process for identifying how many uplink port profiles you need is outlined in the section, "How many uplink port profiles do you need?" later in this chapter.

Figure 3-1 illustrates the different layers in the architecture of a virtualized networking solution, showing uplink port profiles and their connections to other features of the architecture.

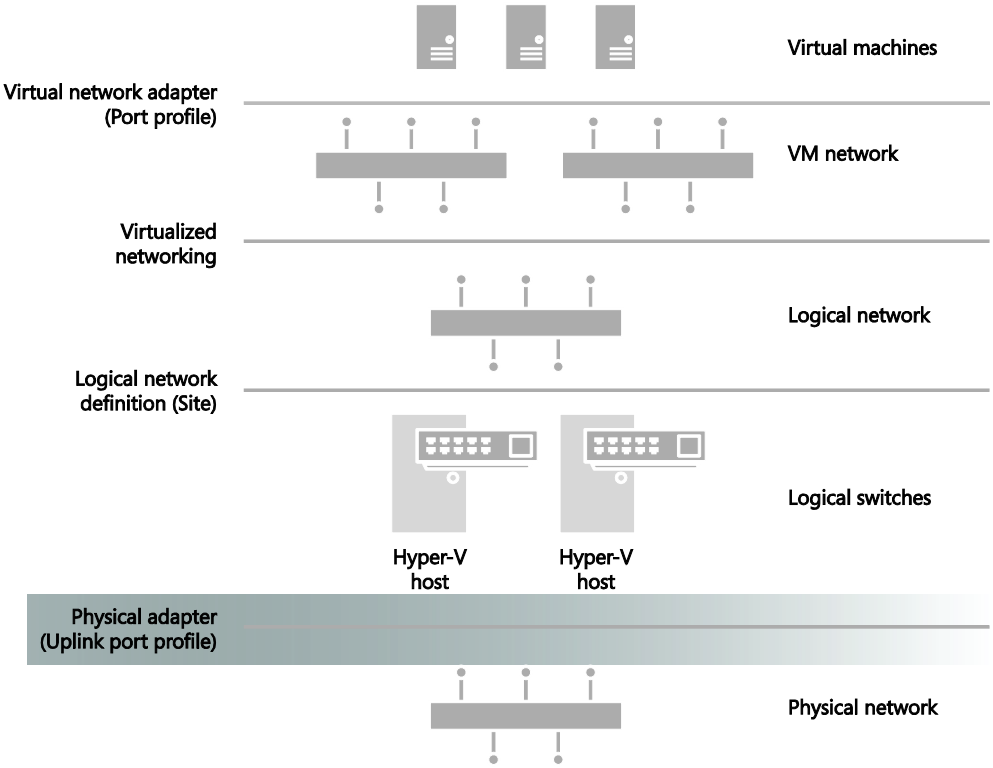


FIGURE 3-1 Position of uplink port profiles in the VMM network architecture

What is defined in an uplink port profile?

An uplink port profile defines the load balancing algorithm and the teaming mode that should be used by any of the physical network adapters on which it is applied, together with the set of logical networks that should be associated with those adapters. If you have host computers with differing requirements in terms of network adapter teaming or load balancing protocols, or if you need to scope logical networks to a specific group of host computers, you will need to create separate uplink port profiles for each one of these combinations.

Load balancing and teaming protocols

You can choose a number of different teaming modes and load balancing algorithms when defining an uplink port profile. For example, Figure 3-2 shows the default teaming mode selections. Host Default is selected for the load balancing algorithm which will either distribute network traffic based on the Hyper-V switch port identifier of the source VM or use a dynamic load balancing algorithm, depending on what the Hyper-V host computer can support. Switch Independent is defined for teaming mode which specifies that (physical) network switch configuration is not required and hence allows network adapters (within the team) to be connected to multiple (non-trunked) physical switches. You can use these default selections or

choose the settings most appropriate for the hosts and network adapters in your network.

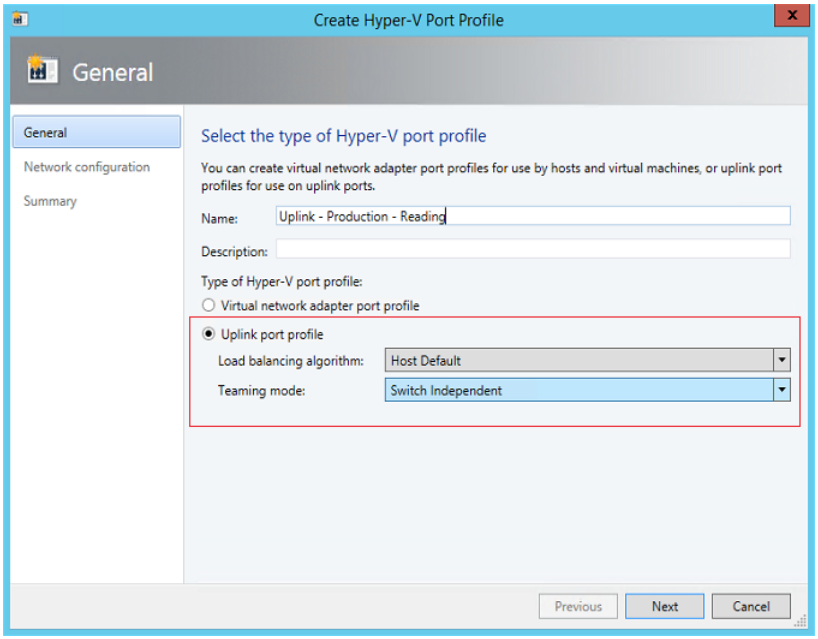


FIGURE 3-2 Load balancing and teaming mode selections in an uplink port profile

> ***See also*** *You can find more detailed information on the different load balancing and traffic distribution options and teaming modes that can be defined in an uplink port profile at http://technet.microsoft.com/library/hh831648.aspx and in the LBFO whitepaper at http://www.microsoft.com/en-us/download/details.aspx?id=30160.*

Network sites (logical networks)

Uplink port profiles also contain a list of network sites (otherwise known as logical network definitions), with each network site representing a link to a different logical network. When the uplink port profile is applied to a physical network adapter, for example as part of logical switch deployment, these network sites determine the set of logical networks that should be associated with the physical adapter and the VLANs and IP subnets that should be allocated to VMs and services that connect to those logical networks.

For example, Figure 3-3 shows the list of network sites that are configured in the uplink port profile called Production-Reading. When this uplink port profile is applied to a physical network adapter as part of logical switch deployment (see Chapter 4, "Logical switches"), the Management, Tenant - VLAN Isolated, and Tenant - PVLAN Isolated logical networks will be associated with that adapter. The VLANs and IP addresses for each logical network will be as defined in the respective network sites.

NOTE You should ensure that all network sites that will be included in a given uplink port profile are scoped to the same set (group) of host computers. Although no error is reported when you initially create the uplink, you will receive an out-of-scope error when you try to apply it (as part of logical switch deployment) to a computer that is not a member of the host groups defined in every one of the network sites included within the uplink.

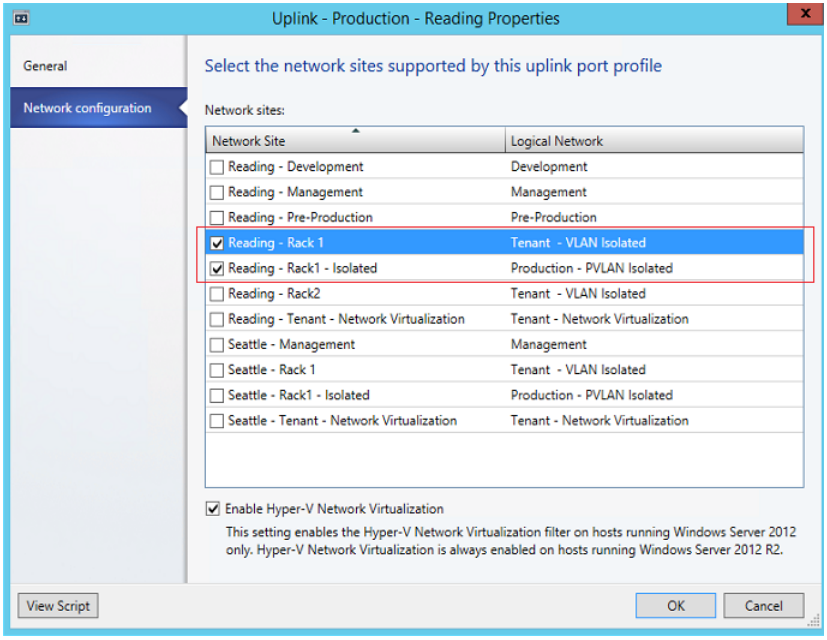


FIGURE 3-3 Network sites in an uplink port profile

As the example in Figure 3-3 shows, it is possible to create an uplink port profile that contains references to multiple network sites (and, hence, logical networks). The key point is that the VLANs and IP addresses defined within each of the selected sites should all be valid (routable) from the physical network adapter that the port profile has been applied to. In the example in Figure 3-3, the VLANs and IP addresses defined in the Reading - Rack 1 and Reading - Rack 1 - Isolated network sites are expected to be both valid and routable if applied to a host computer located in Rack 1 of the Reading datacenter.

You should try to ensure that each of the network sites that you add to an uplink port profile refers to a different logical network. The problem with doing otherwise basically is that *all* of the VLANs and IP subnets defined in those network sites will be associated with the logical network on any host computer on which the uplink port profile is applied. If you are not using VLAN isolation, the host computer has no way to establish which of the range of possible VLANs and IP subnets will be needed to allow VMs connected to the logical network to communicate on the physical network and will pick randomly from the list of those

available. As a consequence, some of the VMs might be allocated routable IP addresses while others are not.

How are uplink port profiles used?

When a logical switch is applied to a network adapter in a Hyper-V host, as shown in Figure 3-4, VMM uses the information contained in the logical switch and the selected uplink port profile to create a Hyper-V virtual switch on the host. The network sites referenced in the uplink port profile are used to determine which logical networks, VLAN, and IP subnets should be associated with that adapter.

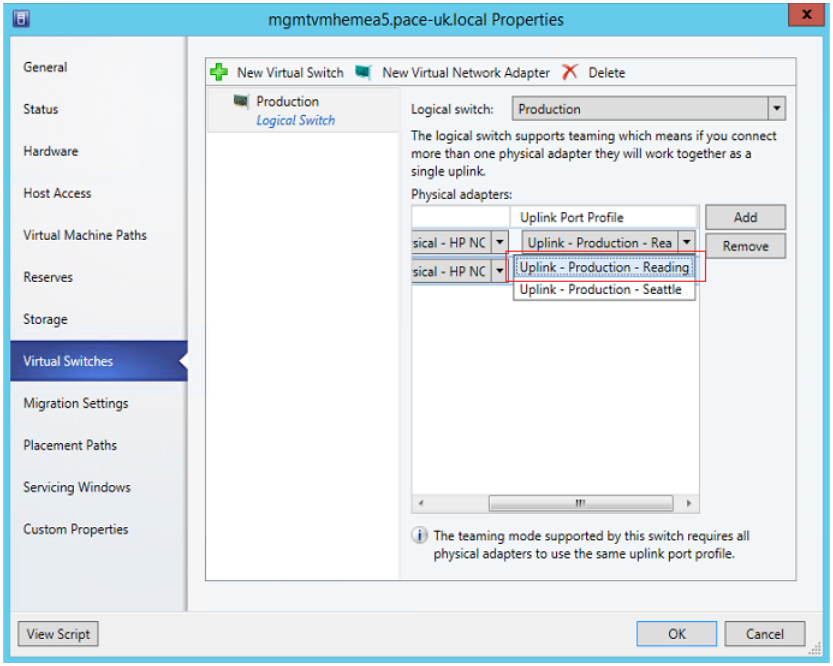


FIGURE 3-4 Selecting an uplink port profile as part of logical switch deployment

As you would expect, if the same logical switch and uplink port profile are applied to two or more adapters in a given host computer, those adapters will be teamed, assuming that this feature has been enabled in the selected logical switch. The teaming protocol used will be the one specified in the selected uplink port profile.

How many uplink port profiles do you need?

At least one uplink port profile needs to be created for you to be able to use logical switches (which are discussed in Chapter 4), but the following sections outline the process you should follow to identify uplink port profiles in your environment, look at some of the main reasons why you would (or would not) create an uplink port profile, and provide an overview of the

key considerations, best practice guidance, and key recommendations.

1. You need at least one uplink port profile for each physical network that exists within your environment.
2. For each of these networks, you need to define uplinks for each physical location that has its own VLAN and IP subnets.
3. If you plan to restrict or otherwise scope logical networks to a specific set of host computers, you will need to create uplinks for each group of computers.
4. You need separate uplink port profiles for groups of computers (in each physical location) that have different connectivity requirements or use different teaming protocols.
5. Finally, you might consider creating separate uplinks for networks that do not or will not support network virtualization.

As the list suggests, you will need a significant number of uplink port profiles in complex environments, so you should also consider a naming convention because a good naming convention can help promote understanding as well as simplify management, as discussed later in this chapter.

Multiple physical networks

Networks are introduced into an environment for many different reasons, but security and isolation are the most common reasons. Service providers (hosters) like Fabrikam might decide to use physical networks to separate tenant workloads from internal (corporate) workloads, for example, but the requirement to provide specific performance guarantees for certain types of network traffic or to mitigate potential network congestion might necessitate the use of separate physical networks.

The trend today is toward converged networking, which minimizes the need for separate physical networks even where traffic isolation and specific service levels are required for different types of network traffic. In a converged network, logical networks separate different types of network traffic on the physical network, and quality of service (QoS) policies ensure that each type of traffic is given the required prioritization and bandwidth.

With that said, regardless of whether you have adopted a converged networking solution, the process to determine how many uplink port profiles to create is the same. Uplink port profiles contain a list of network sites and, as discussed in Chapter 2, "Logical networks," network sites define the VLANs and IP subnets that are associated with a logical network in each physical location. Since each physical network (in a routed network) will have its own set of VLANs and IP subnets, it follows that at least one network site will need to be defined for each physical network.

The question is can all of these network sites be combined in a single uplink port profile or will multiple uplink port profiles be required. To answer this question, consider what happens when an uplink port profile is applied to a network adapter in a host computer as part of logical switch deployment. Essentially, the network sites listed in the uplink port profile are

used to determine which logical networks should be associated with the network adapter and the VLANs and subnets that should be used by VMs and services that use that adapter to connect to the physical network.

If you were to define and use a single uplink port profile, all of the possible VLANs and IP subnets linked to a given logical network would be associated with the physical network adapter on which that profile was applied. This is clearly not an ideal situation, and you should assume at least one uplink port profile will be required for each physical network that is routed (not bridged) to other internal or external networks.

Recall that Fabrikam, the example organization from Chapter 2, has three physical networks: all corporate (internal) workloads and services are hosted on the Datacenter network; storage devices are accessed via the Storage network, and customer (tenant) network traffic is on the Provider network. Leaving aside that Fabrikam has more than one physical location, a minimum of three uplink port profiles would be required to support this environment (see Figure 3-5).

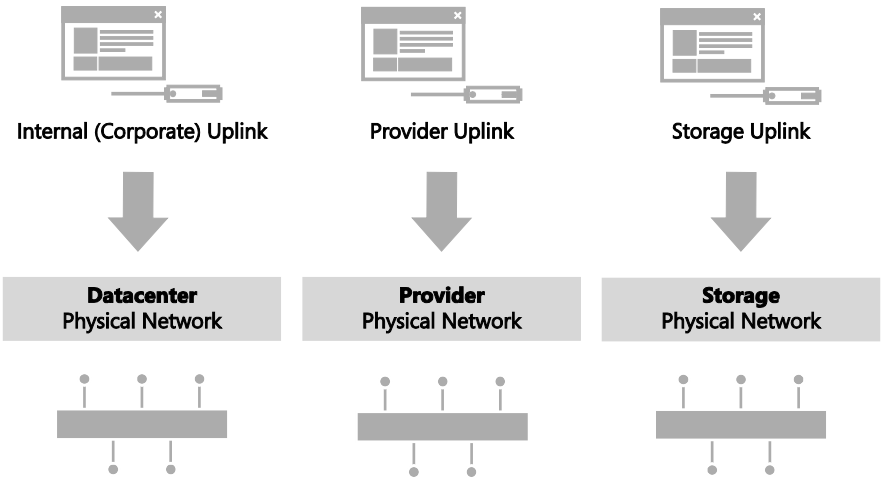


FIGURE 3-5 One uplink port profile for each physical network

Another reason to consider separate uplinks for each physical network is the scope of the logical networks you identified as part of the process discussed in Chapter 2. Although technically possible, it is difficult to identify a scenario in which a given logical network would need to be hosted on multiple physical networks. You would normally find multiple logical networks associated with a single physical network since this approach allows an administrator to separate computers and network services (on that network) with different business purposes, isolate certain types and groups of network traffic, and support workloads with differing QoS requirements and expected bandwidth.

Multiple physical locations

Having determined that at least one uplink port profile is required for each physical network, you can now consider and explore a more realistic network scenario, one in which each physical network is divided into different sites to minimize broadcast traffic and to optimize performance, with routers used to facilitate inter-site communication. In this scenario, each network site has unique VLANs and IP subnets, and VMs or services hosted in a given site will need to be provided with VLAN IDs and IP addresses that are both valid and routable within that site in order to communicate.

At Fabrikam, for example, Hyper-V hosts connected to the Datacenter network are situated in two physical locations, Reading and Seattle, with each of these locations allocated a different set of VLANs and IP subnets. Within each site, VLANs and IP subnets are used to separate different types of workload and help ensure QoS. At present, VMs and services running development workloads are based only in Reading and use VLAN 15 and subnet 192.170.15.0/24, while production workloads are supported in both datacenters. In Reading, production workloads use VLAN 18 and have an IP address in the 192.168.99.0/24 subnet, while those running production workloads in Seattle use VLAN 100 and have an IP address in the 192.168.199.0/24 subnet, as shown in Figure 3-6.

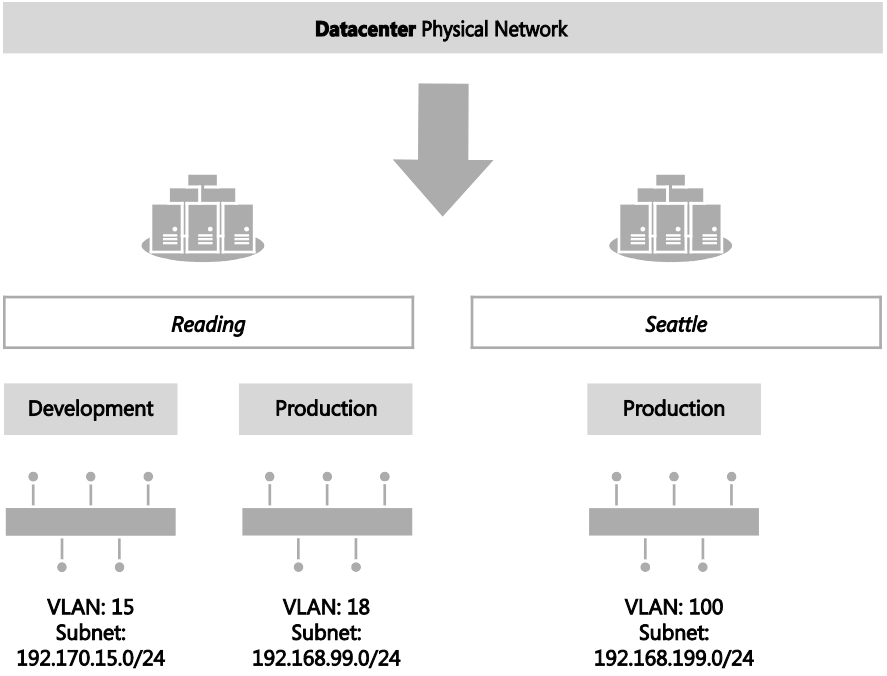


FIGURE 3-6 Workloads differentiated by VLAN and IP subnet

Since development activities are performed only in Reading, only a single network site called Reading-Development is needed to define the VLANs and IP subnets that are to be associated with the Development logical network. To support the Production logical network in multiple locations, however, two network sites are required, one for Reading and one for Seattle, as shown in Figure 3-7, since each of these locations requires a different VLAN and IP subnet for production workloads.

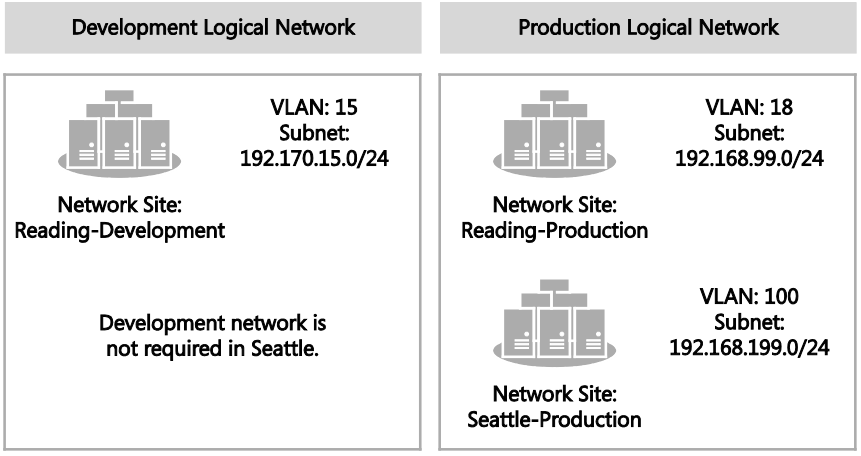


FIGURE 3-7 Network sites in a logical network

Having defined network sites for the logical networks that exist within your environment according to the guidelines set out in Chapter 2, you can begin to allocate those sites to uplink port profiles. Since putting all of these network sites into a single uplink port profile means that *all* of the possible VLANs and IP subnets linked to a given logical network would be associated with the physical network adapter on which that uplink profile was applied, it follows that multiple physical sites with their own set of VLANs and IP addresses, as in the Fabrikam example, will need a separate uplink port profile defined for each physical location.

NOTE **For the purposes of this discussion, multiple physical sites that are effectively linked together in a campus network or a stretched physical network operating across a number of different physical sites can be considered a single physical location.**

Multiple network sites, each one representing a *different* logical network, can be combined in the same uplink profile, assuming that all of the VLANs and IP subnets defined in those network sites are valid (and routable) on any host computer on which the uplink port profile is applied. Note that all of the logical networks (referenced by the selected network sites) will also be made available on each of these hosts. If you want to scope or otherwise restrict which host computers are associated with a given logical network, you will need to create additional uplink port profiles.

In the Fabrikam example, both the Production and Development logical networks are to be made available in Reading. Assuming that, in this simple environment, all hosts in this location have the same network adapter teaming and connectivity requirements and there are no issues with those hosts being associated with both of these logical networks, a single uplink port profile for Reading, referencing the Reading-Development and Reading-Production network sites, is sufficient. Since there is no requirement to support the Development logical network in Seattle, the uplink port profile for that location will contain only a single Network Site, Seattle-Production (see Figure 3-8).

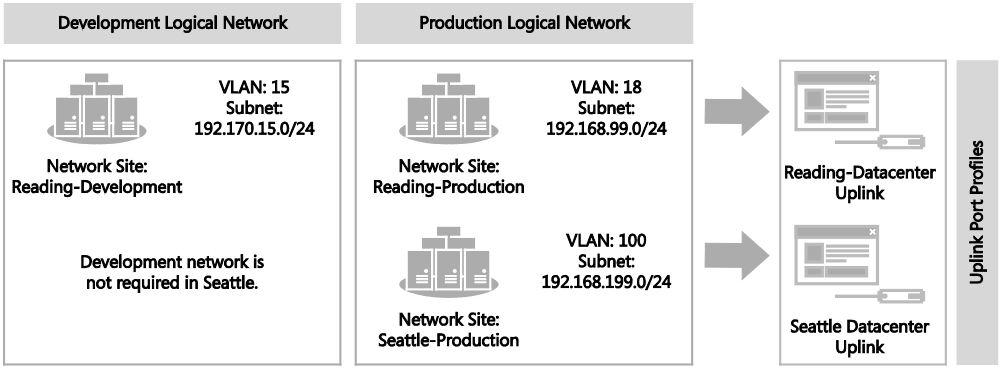


FIGURE 3-8 Defining uplink port profiles for each physical site

After identifying the sites and therefore the initial set of uplink port profiles required for one of the physical networks in your environment, repeat the process for each of the others. In the Fabrikam example, having determined the set of uplinks required for the Datacenter network, which supports internal (corporate) workloads, the administrator would next follow the same process for the Storage and Provider networks.

Restricting the scope of logical networks

To this point, this discussion has assumed that there are no business or technical reasons to restrict the set of logical networks to be included in a given uplink port profile. Yet, within each physical site, it is fairly common to find groups of host computers set aside (dedicated) to a specific purpose or type of workload (see Figure 3-9). In an enterprise environment, for example, the most powerful and generally the most expensive hosts are dedicated to running production workloads. At a service provider like Fabrikam, host computers run workloads for or on behalf of multiple external customers (shared compute model), while other hosts are allocated to and run workloads for a single customer only (dedicated compute model).

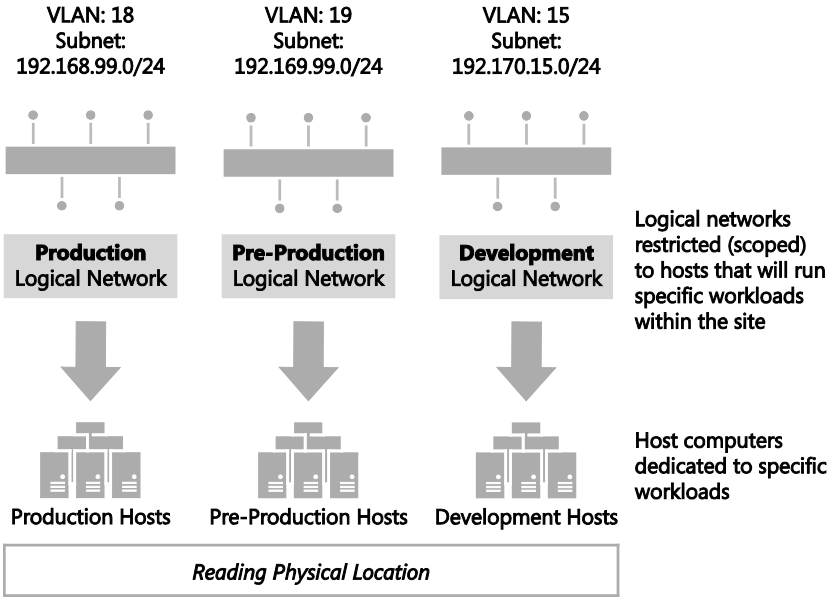


FIGURE 3-9 Associating logical networks with dedicated servers

How does this knowledge and the introduction of dedicated host computers into physical locations influence the approach to uplink port profiles? It is quite possible, as discussed earlier, to create an uplink port profile that contains multiple network sites, each of which represents a different logical network. The issue with such an approach is that all of the logical networks that are referenced in the uplink port profile will be associated with the host computers on which the uplink port profile is applied. In the dedicated compute model, shown in Figure 3-9, this is not appropriate. Fabrikam wants to make sure, for example, that only those hosts dedicated to production workloads are associated with the Production logical network. Therefore, to restrict or otherwise limit the set of host computers (within a physical site) that should be associated with specific logical networks, you will need to create a separate uplink port profile for those hosts.

Note that some logical networks, such as Management, Storage, Backup, and Live Migration for example, are generally common across all systems regardless of the workload they have been dedicated to. Since it is possible to include network sites for multiple logical networks in a single uplink port profile as long as the VLAN and IP subnets defined in these sites are valid within a particular physical location, you might choose to include these in your dedicated uplink port profile. You could also create a separate uplink port profile for common logical networks (within in a particular location) if you prefer. In the latter case, your host computers will require a dedicated physical network adapter on which you can apply this additional uplink port profile.

Fabrikam has three different types of workloads running in Reading, development, pre-production, and production, and for operational reasons, the company has decided to place each one of these different workloads onto a separate (dedicated) group of host computers.

Host computers should be associated with both a workload-specific logical network (such as Development) and the logical network required for host management, which is common to all three.

To support this environment, three uplink port profiles will be required, one for each group of dedicated computers, as mentioned earlier. Each of the three uplink port profiles will contain two network sites: one from the logical network specific to the type of workload (Reading-Development, for example) and the other from the Management logical network (Reading-Management, in this case), which is required on all hosts within Reading regardless of workload (see Figure 3-10).

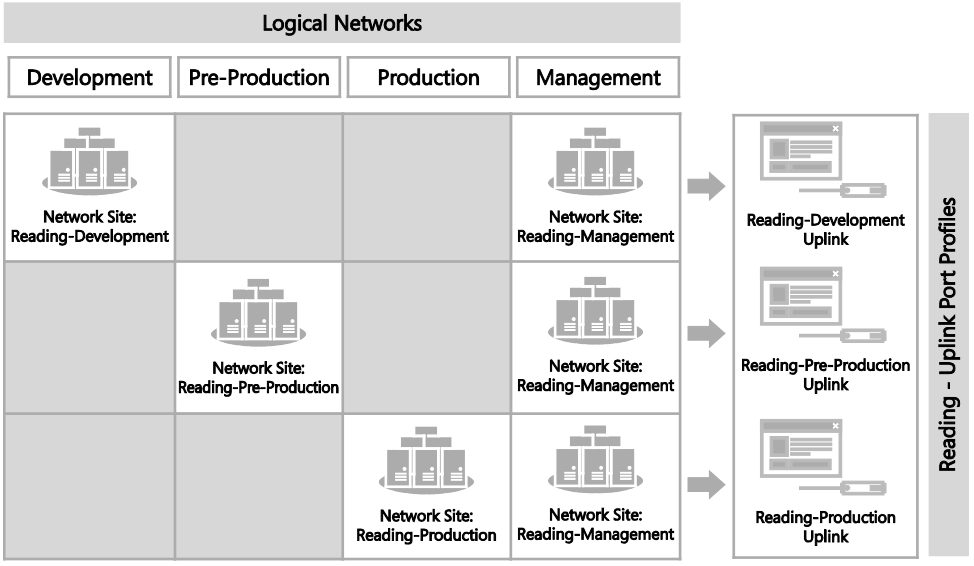


FIGURE 3-10 Define uplink port profiles for dedicated resources

NOTE **Although multiple logical networks might be associated with a host network adapter using a single uplink port profile, each of these networks will be isolated from each other. If you want to allow VMs and services on one logical network to communicate with those connected to another, you will need to use a router or gateway device.**

In each physical location, computers dedicated to specific workloads will normally be able to communicate with each other without the need for traffic routing, but in some environments, this is not the case for various reasons, including scale (insufficient IP addresses available in a given subnet), performance, and the need to manage broadcast traffic. It might therefore be necessary to place these host computers in different (routed) IP subnets.

At Fabrikam, for example, host computers dedicated to running production workloads in the Reading datacenter are located in one of three racks, with each rack allocated its own VLAN and IP subnet to allow the solution to scale. A router allows host computers, VMs, and services in one rack to communicate with any of the others. To support this environment, three network sites will need to be created in VMM for the Production logical network (in Reading), one for each rack, as shown in Figure 3-11, with additional network sites added as the solution scales above these three initial racks.

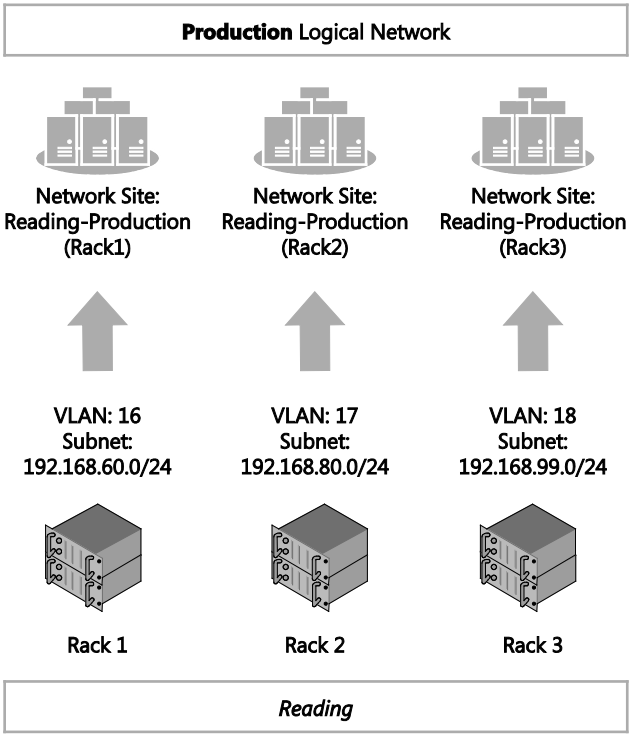


FIGURE 3-11 Multiple network sites in the same physical location

NOTE **In addition to those created for the Production logical network, it will be necessary to define network sites (one per rack) for any other logical networks that need to be associated with host computers located in these racks.**

All three of these network sites cannot be placed into a single uplink port profile since doing so would mean that *all* of the VLANs and IP subnets linked to the Production logical network would be associated with any physical network adapter on which the uplink profile was applied, which is clearly not desirable. It therefore follows that if you have a group of host computers within a physical location that have their own set of VLANs and IP addresses and use a switch or router to communicate with other resources on the network, as in the Fabrikam

example, then you will need to define uplink port profiles for host computers (within a site) that are on a separate routed subnet.

Returning to the Fabrikam example, there are now three separate network sites for the Production and Management logical networks in the Reading datacenter. These cannot be combined into a single uplink port profile for the reasons mentioned above, so individual uplink port profiles must be created for each rack, as shown in Figure 3-12.

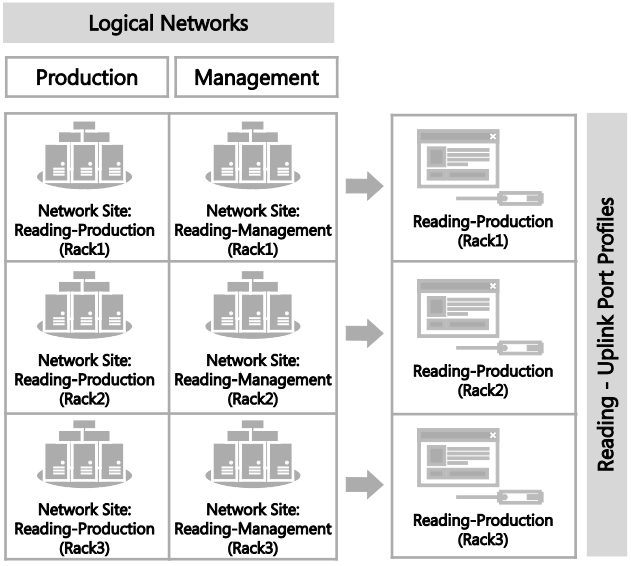


FIGURE 3-12 Uplink port profiles for sites within physical location

If you set aside host computers for specific workloads, projects, or tenants, you clearly do not want to permit someone to inadvertently apply an uplink port profile designed for host computers running production workloads to a host used only for development (Reading-Production (Rack1) in the example.

Unfortunately, there is no such thing as security groups or scoping for uplink port profiles. You can address this limitation, however, by including within your uplink port profile network sites that are scoped (restricted) to host computers that are members of a particular host group, as outlined in Chapter 2. Note that the scope of all of the network sites in a particular uplink port profile must be identical. If they are not, you might receive an out-of-scope error when you try to apply the uplink port profile (as part of logical switch deployment) to a computer that does not fit into the host groups used by all of the network sites referenced within the uplink.

Different connectivity requirements

After determining the initial set of uplink port profiles to be created for each group of host computers at a given physical location, the next step is to look at how each of these computers is physically connected to the network. In a software defined network (SDN), all hosts would be configured the same, all would have the same set of network adapters, all network traffic would co-exist on the same physical network, and logical networks and QoS policies would be used to differentiate and prioritize different types of traffic. In such an environment, known as a fully converged network, your original set of uplink port profiles might require little or no further refinement, as illustrated in Figure 3-13.

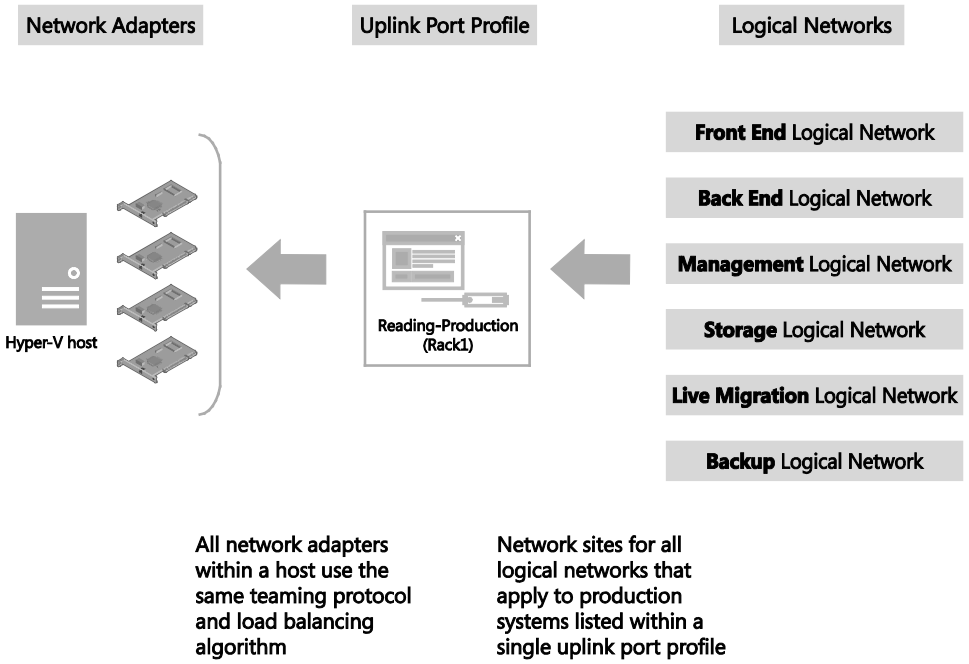


FIGURE 3-13 Uplink port profile assignment in a fully converged network.

In most environments, however, host computers are not all connected to the network in the same way. For example, connectivity often varies according to the role and type of workload expected to run on the host. Even in the same host computer, network adapters might be specialized—dedicated to specific functions like storage, host management, or tenant traffic, each of which might require different teaming modes and load balancing algorithms. Hosts might even contain a mixture of standard and specialist network capabilities, such as RDMA and SR-IOV, which might need to be managed differently.

NOTE **If you are using VMM 2012 or VMM 2012 R2, and are using RDMA network adapters, you should not create uplink port profiles for or deploy a logical switch on these adapters since releases of VMM up to and including 2012 R2 do not specifically recognize or support RDMA network adapters.**

Based on these considerations, you will likely need to refine your original list of uplink port profiles by creating additional uplink port profiles to associate logical networks with specific network adapters that optimize around workload and connectivity to the physical network.

At Fabrikam, for example, the majority of host computers in Rack 1 of the Reading datacenter have eight physical network adapters, with two dedicated to host management, four dedicated to storage, and the remaining two used by guest VMs. The adapter teaming and load balancing requirements for each of these different types of workload are very different. For example, the network adapters configured for storage in this environment are using LACP teaming to optimize around Fabrikam's use of SMB v3 and Multi-Path IO, while the SR-IOV adapters used by guest VMs do not support teaming due to the way these adapters operate.

Because a single uplink port profile will not work for this group of host computers, additional uplink port profiles must be created to account for the different connectivity requirements. Furthermore, since workloads like storage, backup, and live migration will not be present on all of the network adapters in a host (as in the converged network model), they will be present only on those adapters that have been allocated and optimized for that type of workload.

The diagram in Figure 3-14 shows the result of this optimization around connectivity. Instead of a single uplink port profile for Rack 1, there are now three, one for each different group of network adapters in the host, and instead of including network sites for all logical networks that apply to production systems, the new uplink port profiles contain only those that are relevant to the workloads that will be carried on those adapters.

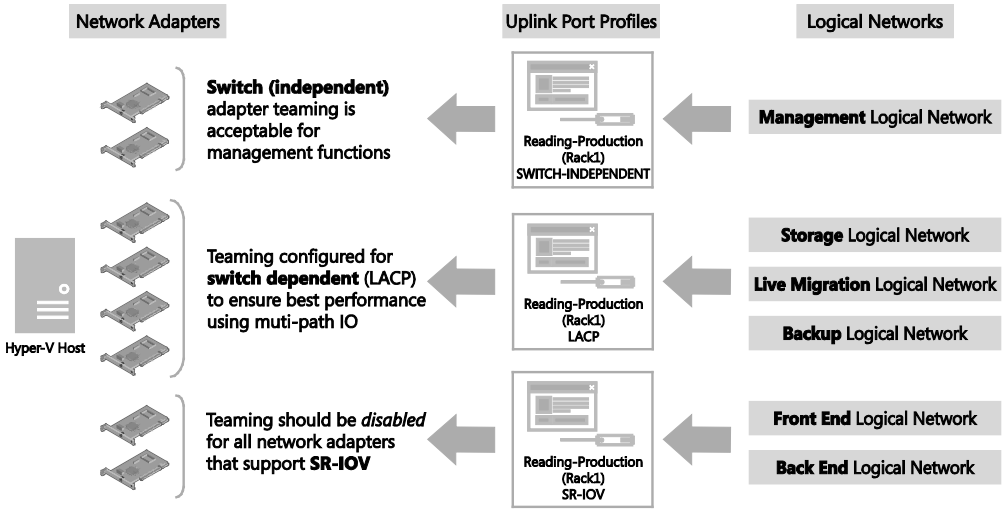


FIGURE 3-14 Identifying the need for uplinks based on connectivity

Host computers dedicated to specific tasks and workloads often have different requirements in terms of connectivity to other hosts in the datacenter, even when they are connected to the same networks or reside in the same rack. In each case, you need to review the type of network adapters in each host to determine if they require any special attention, creating new uplink port profiles (as discussed) where it make sense to do so.

Naming conventions

In complex multi-site environments, with multiple logical and physical networks and specialist host computers that have differing requirements in terms of connectivity and network adapter teaming, you can quickly build up a significant number of uplink port profiles. Without a sound naming convention, it can become difficult to determine which uplinks should be applied to which network adapters in a given host during logical switch deployment.

A naming convention like the following can help administrators clearly identify the scope and purpose of a given uplink and reduce management costs. Adding a high level description to aid understanding is strongly recommended.

[Location] - [Group] (Racknn) - [Connectivity]

Typical examples of this structure would be:

Reading - Production (Rack1) - LACP

Reading - Production (Rack1) - SRIOV

Reading - Development - NoTeam

This particular structure might be too detailed or complex for your specific environment, but the point is you need to arrive at a convention that clearly identifies the different uplink port profiles you have created and what they are used for.

Virtual network adapter port profiles

Virtual network adapter port profiles can be applied to network adapters defined within a guest VM or virtual network interface cards (vNICs) that are created within a logical switch deployed on a host computer. They define the processor offload settings that should be used (assuming that the physical hardware on which the virtual adapter is deployed is able to support those capabilities), the security settings that should be applied and how outgoing network bandwidth should be controlled and managed and if (from the R2 release) whether dynamic IP changes are allowed when using NVGRE.

Unlike uplink port profiles, a number of example virtual network adapter port profiles are provided out of the box. In practice, outside of a relatively small environment, it's likely that you will need to add to and refine this initial list to meet your specific requirements. A simple process to help you determine how many of these profiles you actually need is outlined later in this chapter.

It's important to note that users do not have direct access to network adapter port profiles. Instead, the user (and administrator in the case of a vNIC) selects a logical switch and then a port classification within that switch for each adapter's connection to the network. The selected classification is essentially mapped to one of the network adapter port profiles that is available within the selected logical switch.

The diagram in Figure 3-15 illustrates the different layers that make up the architecture of a virtualized networking solution, highlighting where network adapter port profiles fit into the architecture. Although not shown in the diagram, network adapter port profiles can also be applied to vNICs, but only when the vNIC is associated with a VM network that is *not* enabled for network virtualization (see Chapter 2 for more details).

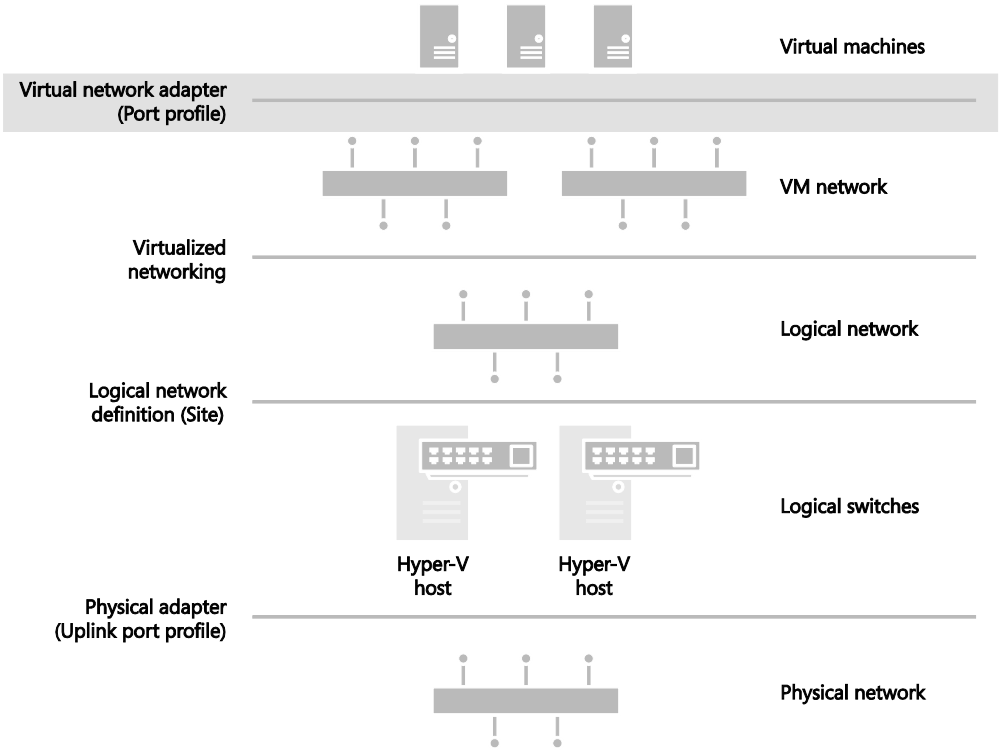


FIGURE 3-15 Architecture showing network adapter port profiles

What is defined in a virtual network adapter port profile?

Virtual network adapter port profiles define the processor offload settings, security settings, and bandwidth limitations to be enforced on network adapters within a guest VM or vNICs on which they are applied. You can find more detail on the range of different settings and capabilities that can be configured within one of these profiles on TechNet at *http://technet.microsoft.com/en-us/library/jj721570.aspx*.

If groups of host computers within your environment have differing requirements with respect to any of these settings and capabilities, you should consider separating network adapter port profiles for each different combination. This topic is covered in more detail later in the chapter.

How are virtual network adapter port profiles used?

As mentioned earlier, users (and administrators in the case of vNICs) do not interact with network adapter port profiles directly. To connect a VM network adapter to a VM network, the user selects a logical switch and, optionally, a port classification from the list of those available within the selected switch. The port classification (or the default if none is chosen by the user) maps to one of the network adapter port profiles within the same switch. The settings and

capabilities in this mapped port profile are then applied to the virtual network adapter.

In the example shown in Figure 3-16, the Reading-Production logical switch contains a number of network adapter port profiles and port classifications. One of the network adapters in VM WEBSVR-001 is connected to the Corporate VM network through this logical switch. Because the High Bandwidth port classification has been selected and is mapped (within the logical switch) to the High Bandwidth network adapter port profile, outbound network traffic on network adapter 1 would include the IEEE priority tag and to be allocated a minimum bandwidth weight of 10.

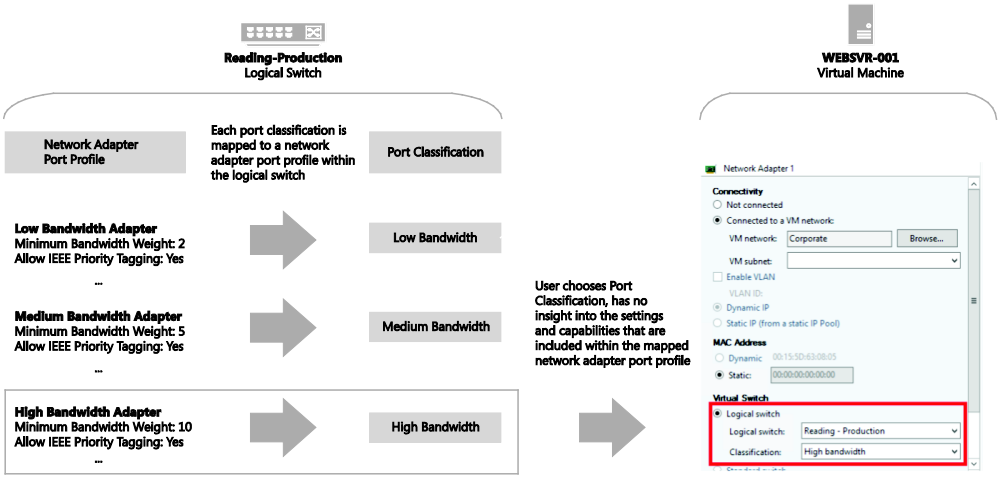


FIGURE 3-16 Applying a network adapter port profile

The port classification is simply a label. The end users have no insight into which network adapter port profile has been mapped to any of the port classifications they have access to or the different settings and capabilities that are defined within the port profile they choose to apply to their VMs.

How many virtual network adapter port profiles do you need?

Multiple virtual network adapter port profiles are required whenever your environment contains physical and virtual computers with differing QoS requirements, essentially the need to provide certain guarantees for outgoing network bandwidth and security policy, or when specialist network adapters (e.g., SR-IOV) that require additional configuration with VMM have been deployed.

Although a number of sample network adapter port profiles are provided out of the box, in practice, business requirements such as the need to enforce different security settings or ensure that certain workloads are prioritized above others likely require you to update and

refine the settings and capabilities within this initial set of profiles and to create additional ones.

The following process will help you decide whether to create new network adapter profiles in your environment and outlines some best practice guidance and key recommendations:

1. First, identify and build network adapter port profiles for workloads that need a guaranteed QoS, essentially where you need to control and manage outgoing network bandwidth.
2. Add additional network adapter port profiles to support VM networks that have different security requirements.
3. Finally, add network adapter port profiles for any physical network adapters that support either IPSec Task Offloading, SR-IOV, or Virtual Machine Queue (VMQ) processor offload capabilities.

As with the other features of your virtual networking architecture, after you have identified the required set of network adapter port profiles, you should identify a formal naming convention to help promote understanding.

Quality of Service

In a converged network, network traffic from a mixture of different workload types co-exist on and share the same physical network with QoS policies used to ensure network traffic related to the most important workloads, like production for example, will be prioritized above any of those considered of lesser importance. This distinction can help you to define your initial set of network adapter port profiles, one for each different priority (or weighting) value.

At Fabrikam, a number of different types of workload have been identified, as shown in Figure 3-17, with each one allocated a relative weighting or minimum bandwidth weight. Management and Cluster Heartbeat share the same value and are ranked highest in the list since these are required to keep the environment up and running. Live Migration and Production workloads follow in terms of relative priority. Network traffic related to development is viewed as the least important, at least in this ranking, and has the lowest bandwidth weight. A separate network adapter port profile should be created for each group of workloads with the same relative rating.

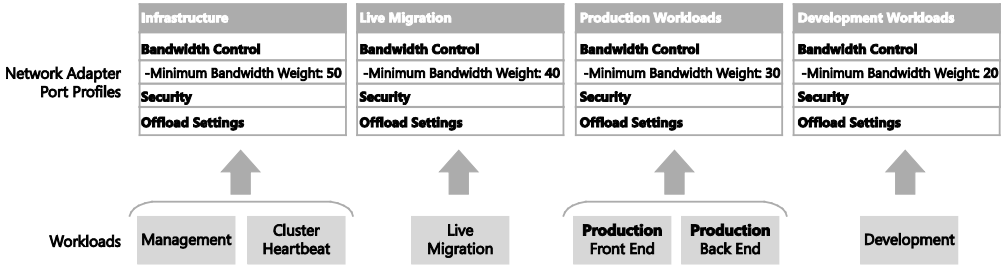


FIGURE 3-17 A network adapter port profile for each prioritized workload

Instead of allocating relative values to particular workloads, you can use the minimum and maximum bandwidth (in Mb) settings within the network adapter port profile to define specific thresholds for each workload type. This approach might provide more granular control, but relative priority is recommended since it allows you to make use of all available bandwidth.

NOTE **A given logical switch will be in either Relative Weights or Absolute Bandwidth Values mode with respect to bandwidth control. The default is Relative Weights. If some workloads will use relative weights and others will use fixed bandwidth limits, you need to create separate network adapter port profiles for each type of workload and ensure that the network adapter port profiles contained within a given logical switch match and align with the bandwidth control mode that has been defined for that switch. See Chapter 4 for more information on logical switches.**

Security settings

After creating an initial set of network adapter port profiles based on workload type and the priority of those workloads relative to others, the next step is to consider security settings. In general, most of the VMs and services that use a particular network adapter port profile will have the same general requirements in terms of network security, and in these cases, no further refinement to your solution is required since the appropriate security settings can be enabled in the selected port profile.

If, however, certain VMs and services within a given workload require different security settings, you will need to create a new network adapter port profile. The new profile will be identical in terms of bandwidth control since the underlying workloads are unchanged, but will contain the new security settings and configuration.

At Fabrikam, for example, most systems in production do not use or require support for guest teaming, so this security setting has been purposely disabled in the Production-Secure network adapter port profile. This security setting is clearly inappropriate for applications and services in production that rely on teamed in-guest network adapters for performance reasons, so a new network adapter port profile has been introduced, Production-Scale Out, to support this specific requirement. For similar reasons, a third port profile, Production-Tenant Interface, has been created for front-line production VMs on which MAC Spoofing needs to be enabled (see Figure 3-18).

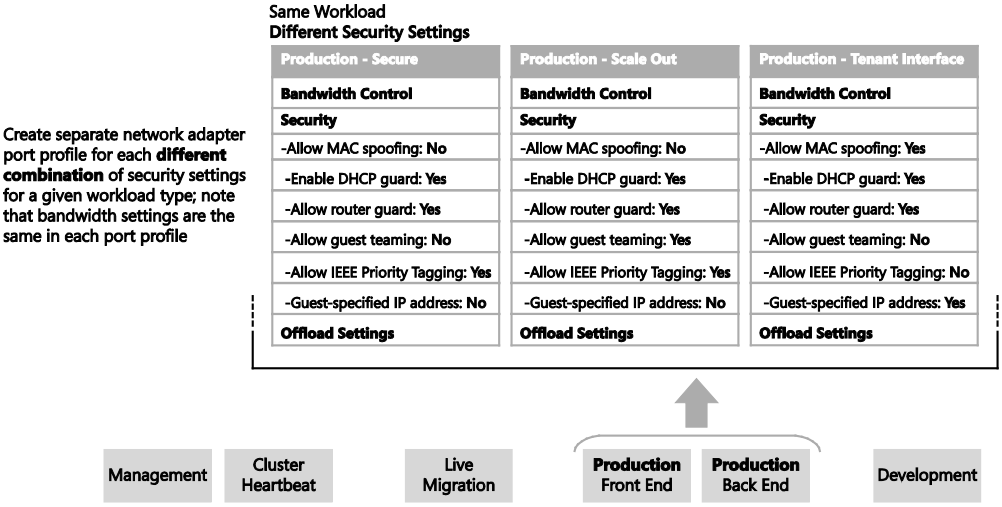


FIGURE 3-18 Refined solution for workloads with different security settings

Having identified the range of security settings required for one type of workload, you should review each of the others and repeat the process, creating additional network adapter port profiles as you identify different security requirements.

Support for processor offloading

In the final step, you optimize the set of port profiles for host computers containing physical network adapters that support either IPsec Task Offloading, SR-IOV, or VMQ. As before, if workloads that use a given network adapter port profile will be deployed only on host computers that have the same capabilities with respect to processor offload settings, then no further refinement is required; the appropriate settings can be made in the selected port profile.

If, however, a given workload, such as production, will run on hosts with a mixture of different processor offload capabilities, you will need to create a new network adapter port profile for each type. The new profile will be identical in terms of bandwidth control and security settings; the only difference will be the processor offload configuration.

At Fabrikam, for example, most systems in production do not use or require support for SR-IOV, so this capability has been purposely disabled in the Production-Secure-Standard network adapter port profile (see Figure 3-19). This port profile is clearly inappropriate for applications and services in production that use this feature, so a new network adapter port profile has been introduced, Production-Secure-SR-IOV, that supports this specific requirement.

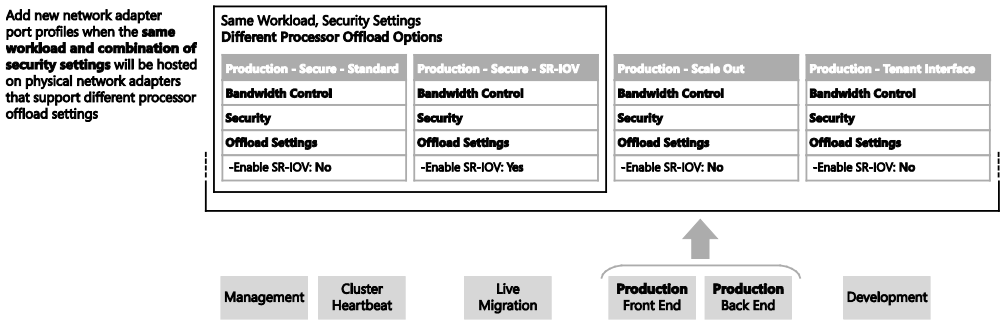


FIGURE 3-19 Adding profiles for network adapters that support offloading

NOTE **Enabling support in the network adapter port profile might not be sufficient for certain processor offload modes. It might be necessary to make changes in multiple places within the virtual network architecture, including in the physical host, the logical switch, and the uplink port profile, for this to work successfully, as is the case with SR-IOV.**

Naming conventions

It is likely that compared to some of the other objects covered so far, you will find that you need relatively few network adapter port profiles even in the largest of environments. But as with all things, it is still useful to develop a sound naming convention. The following convention below is a good starting point since it helps administrators clearly identify the scope and purpose of a given port profile. Adding a high level description to aid understanding is strongly recommended.

[Workload] - [Security] - [Connectivity]

Typical examples of this structure would be:

Infrastructure

Production - Secure

Production - Secure - SR-IOV

Development

Test

CHAPTER 4

Logical switches

A logical switch brings together all of the different elements, including uplink port profiles, native port profiles, port classifications, and switch extensions, that are relevant to a particular physical or logical network to create a combined model. Essentially, this is a template that contains a defined set of parameters (port profiles, classifications, and so on) that you can use to create Hyper-V virtual switches on any Windows Server 2012 or newer hosts that connect to the network.

This chapter covers the need for utilizing a logical switch, compares it to the distributed switch in VMware, considers some of the network configurations available, and details the procedures for deploying a logical switch. Finally, the text considers what happens in the event of a Virtual Machine Manager (VMM) failure and how the use of a logical switch is impacted when you are working with software-defined networks.

This chapter will:

- Review the role of logical switches in a virtualized network solution
- Discuss the differences between a standard (or virtual) switch, a logical switch, and a VMware distributed switch
- Introduce a step-by-step process for determining how many of these switches you need
- Explain how to configure a logical switch in your environment
- Describe how to maintain your logical switch and use it to update configuration across your hosts
- Help answer the question, "How many logical switches do I really require?"

Logical switches

As described previously in Chapter 1, "Key concepts," logical switches bring together all of the different uplink port profiles, native port profiles, port classifications, and switch extensions that are relevant to a particular physical or logical network. A logical switch is essentially a template that contains an administrator-defined set of parameters that you can use to create Hyper-V virtual switches on any of the host computers on which it is applied.

Figure 4-1 illustrates the different layers that make up the architecture of a virtualized networking solution with logical switches highlighted to show their connection to other features of the architecture.

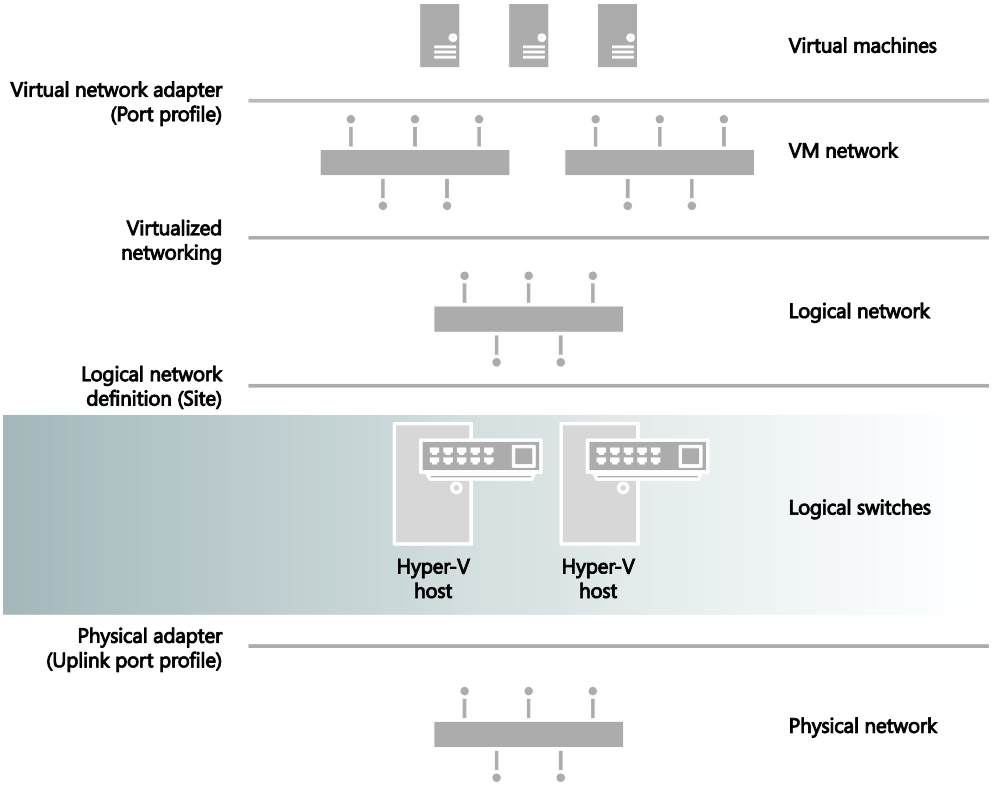


FIGURE 4-1 Architecture of a virtualized network solution showing how logical switches connect with other features

When you use a logical switch to create a Hyper-V switch on a host computer, you select the *most appropriate* combination of port profiles, classifications, and switch extensions from those defined in the logical switch. You can find more information on Hyper-V virtual switches at *http://technet.microsoft.com/en-us/library/hh831823.aspx*.

As a general principle, a new logical switch will be required for every physical network that exists in your environment, but if you plan to restrict some logical networks to a limited set of hosts, as with Fabrikam, the example organization introduced in previous chapters, and/or have custom connectivity requirements, you may find it necessary to create additional logical switches.

NOTE **A physical network is a network that is physically isolated from independent networking equipment stacks. Physically isolated networks therefore require that the host has a dedicated interface to establish a connection to these networks. Examples include highly secure line of business services or high-risk public networks.**

What is a logical switch?

To understand how a logical switch works, first contemplate your host and consider how it is configured, either via script or through the UI tools. Figure 4-2 shows a typical Windows Server 2012 fully converged network design. Using your preferred method of configuration, you will complete your host commissioning process by applying the appropriate network settings and running some validation tests.

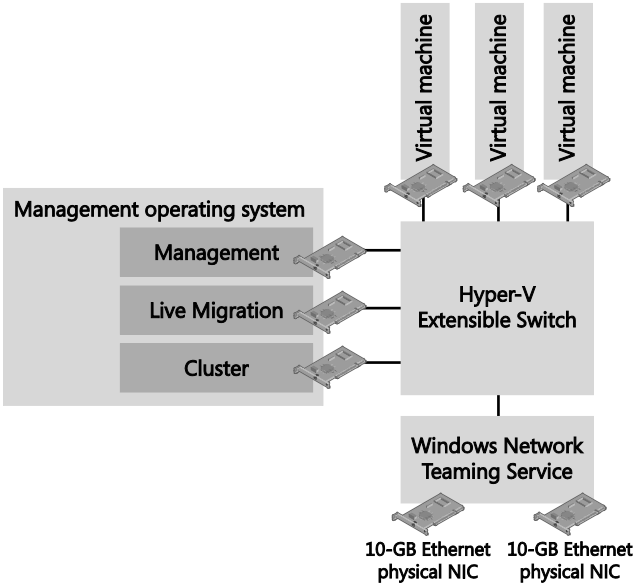


FIGURE 4-2 Sample network configuration for a Hyper-V host, fully converged

Of course, in reality you will not be configuring just a single host, but typically a number of similar hosts (see Figure 4-3), which may all be treated as standalones or clustered. Therefore, as each host is prepared and ready for configuration, you will need to repeat the same procedure.

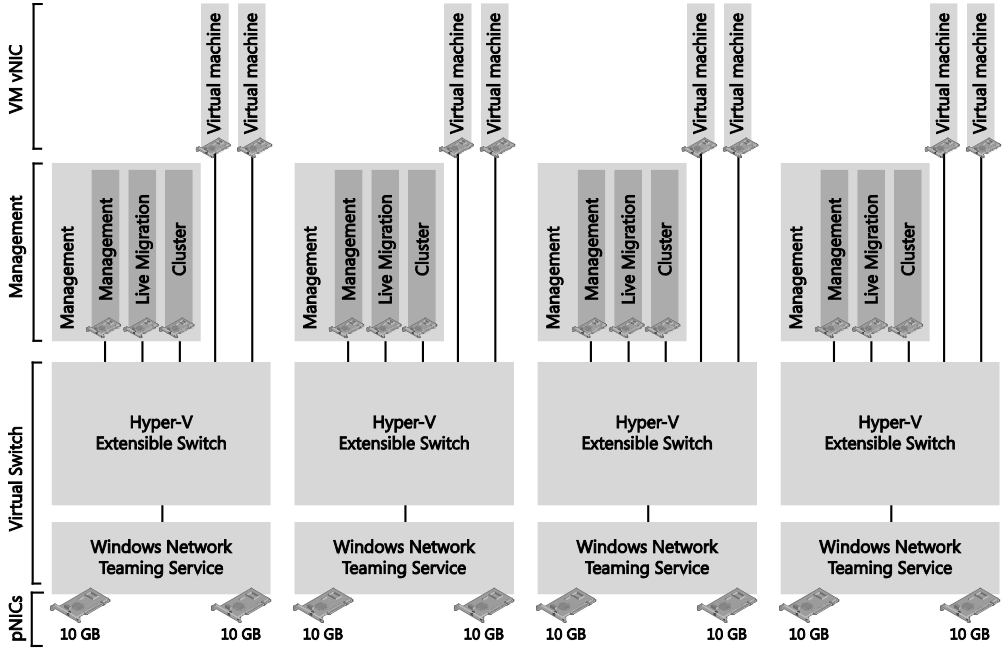


FIGURE 4-3 Network configuration must be implemented identically to scale hosts

Now consider for a moment that your virtual switch must support 1,000 different VLANs for your tenants. To do so, you must manually ensure that all of the hosts you have deployed in a high availability (HA) group (or cluster) have their virtual switches tagged exactly the same (manual configuration per host), otherwise the HA switch will not be presented and available for placement. Just one tiny mistake here and each host's switch will have to be inspected independently to identify the deviation.

The introduction of a logical switch, however, has changed this for the better. You can now define all of these detailed settings once and then bind them to your hosts' relevant physical adaptors. VMM will do the rest of the work for you.

Keeping in mind that the logical switch is essentially a template, the illustration in Figure 4-4 can help you visualize what this truly implies. The same template settings are applied to each of the hosts, ensuring that a centrally managed configuration is implemented and keeping everything consistent as long as host computers remain managed.

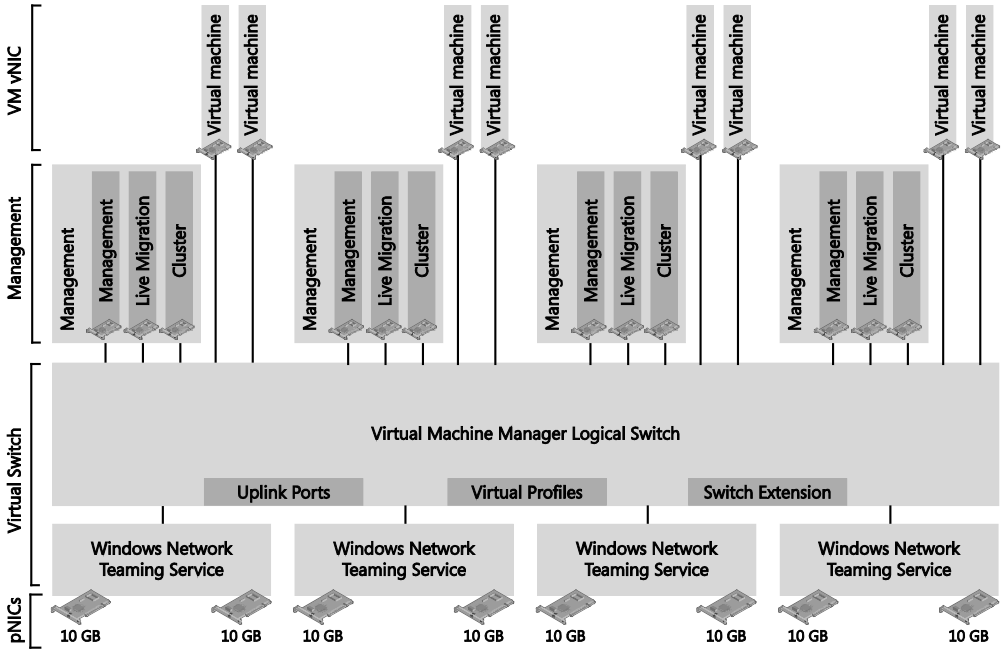


FIGURE 4-4 One logical switch configuration replacing multiple independent configurations

With this template-based approach, you can scale your environment with relative simplicity, confident that each new host will be implemented and configured in the same manner.

Logical switches versus virtual switches

A logical switch is a VMM concept or model to define a deployable template of network-related settings to a managed host. A virtual switch, on the other hand, is an operating system feature utilized by Hyper-V to process network traffic.

When a logical switch is applied to a Hyper-V host, VMM uses the information contained in the logical switch, including the selected uplink port profiles, to create a Hyper-V virtual switch on the host and associate the network adapter(s) with the required logical networks, VLAN, and IP subnets. It therefore follows that the host must be a member of a host group that has been scoped to those logical networks. If the host is not in an appropriate host group, deployment of the switch will fail with an Out of Scope error.

NOTE When you allow VMM to create and configure the virtual switch on your hosts, you may notice that if you then revisit the Hyper-V console on the configured host, the options for modifying any additional settings on the virtual switch are disabled. If you have the urge to tamper with the switch, you can use Windows PowerShell; however, any changes at this point will compromise the compliance of your switch.

If you apply the same logical switch and uplink port profile to two or more adapters, the adapters will be teamed, assuming that this option has been defined in the logical switch. The option to add or remove adapters described above will be available only if Uplink Mode has been set to Team.

Logical switches versus VMware distributed switches

It is fair to draw the conclusion that the VMM logical switch is analogous to the VMware vSphere distributed switch, but only in the sense that both permit centralized management of their respective hosts' virtual switches. There are some subtle differences in terminology and how the technology is implemented in both products. Table 4-1 identifies some of these differences.

TABLE 4-1 Comparison of Microsoft and VMware network switch terminology

VMM LOGICAL SWITCH	VMWARE DISTRIBUTED SWITCH
VMM logical switch templates the configuration options applied to Microsoft virtual switches.	vSphere Standard switch and vSphere Distributed switches are two totally independent virtual switch constructs.
Third-party extensions are added to the existing Microsoft virtual switch, e.g., Cisco Nexus 1000v.	Third-party extensions are implemented as new switches, e.g., Cisco Nexus 1000v.
Host virtual NICs are utilized for traffic classification, similar to HP FlexNIC or IBM vNIC (but without the logical limits of supporting only four).	
VM bandwidth management and isolation is implemented with virtual port profiles assigned to the VM vNIC.	Resource allocation is a combination of vNIC and port profiles leveraged from port classification.
Teaming is implemented via uplink port profiles, defining the load balancing algorithm and teaming mode to be implemented on the selected physical network interfaces.	Distributed switch uplinks are defined as dvUplinks, utilizing an uplink port profile.

Logical switch planning considerations

As you start to think about your environment and how many logical switches you will need to create to support your business requirements, there are a number of key considerations presented in this section that you should be sure to review as part of your planning process.

Hyper-V Server 2008 network architecture

In the original networking designs based on Hyper-V in Windows Server 2008, usually an NIC team would be created for each of the primary networks (Management, Cluster, and Live Migration), along with a team dedicated to the virtual machine (VM) traffic processed through the virtual switch. Additionally, the design would include a connection to the storage environment, possibly utilizing a pair of fibre channel interfaces, or when implementing iSCSI block storage, a pair of NICs distributed through multipath I/O (MPIO). On top of this, the design might include an additional team of NICs to segregate the traffic necessary for backup workloads. Figure 4-5 depicts a typical network configuration for Hyper-V Server 2008 with dedicated teams.

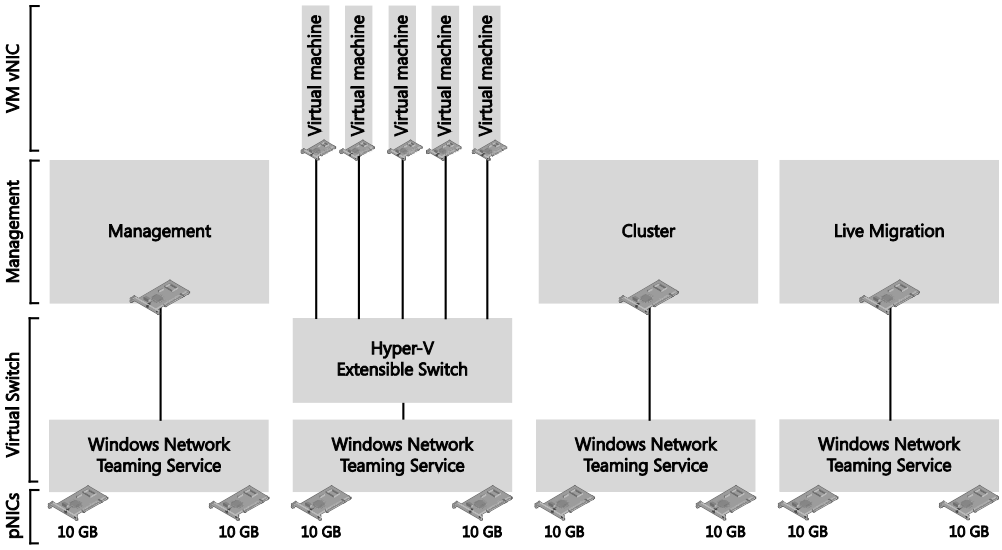


FIGURE 4-5 A typical network configuration for Hyper-V Server 2008 with dedicated teams

This legacy design was not only a challenge for the hosts, which required multiple quad-port network interfaces to present the desired number of teams on the limited number of physical bus slots, but also led to extremely complex physical networking requirements, with each host exposing an average of eight or more network ports, consuming switch space in large blocks, and vastly incrementing the potential for both misconfiguration and cabling mistakes.

Implementing this approach in VMM would require you to deploy a logical switch per team because each team in the configuration requires a dedicated logical switch that defines the configuration of the uplink interfaces based on the attached uplink port profiles. Therefore, reproducing this legacy approach as illustrated in Figure 4-5 would require four logical switches.

NOTE **With the introduction of Hyper-V 2012, using multiple network adapters to separate network traffic is no longer necessary. Different network traffic classifications can all use the same physical adapters in what is known as a converged network.**

Quality of service (QoS)

By using quality of service (QoS) mechanisms, you can use existing resources more efficiently to ensure the required level of service without reactively expanding or over-provisioning networking fabrics. Considering the needs of the different network workloads you support, you can define QoS by relative priority or in absolute terms. In Windows Server 2012, these QoS concepts are implemented as follows:

- **Bits Per Second (Absolute)** Bits per second rules are specific, guaranteeing a very clearly defined amount of bandwidth. This approach has its place when you need to communicate and understand the specific bandwidth allocations. However, absolute QoS is quite inflexible. For example, consider a VM that has been guaranteed a defined bandwidth. If this VM is moved to a host that also hosts additional VMs with guaranteed bandwidth, it quickly becomes possible to oversubscribe the available bandwidth, which will result in guarantee breaches or worse.
- **Weights (Relative)** Using a weight-based approach, you offer a share of the total available bandwidth on the network, with no considerations of the actual speed. For example, a VM guaranteed 50 percent of available bandwidth hosted on a 10-GB network would be offered 5 GB, but if moved to a host with a 1-GB network, the VM would be offered 512 MB. The relative approach is generally preferred over the absolute approach given that it provides more flexibility.

It's better if you do not configure the logical switch to use absolute values because there is no way to confirm that you will actually get the values you specified. The preferred mechanism is to use relative (weight-based) QoS settings as noted above, but there may be reasons why you need both approaches. You should note that placement will block live migration across logical switches that have different QoS settings.

Virtual network interface cards (vNICs)

Windows Server 2012 offers the ability to leverage the functions of QoS and combine them with another new feature, Virtual Network Interfaces, in the host operating system. Instead of creating a single NIC team (tNIC) from one or more physical NICs (pNIC) and binding a virtual switch to it, you can now use the new host vNIC feature to create a number of vNICs that can then be assigned to each of the primary workloads in the host operating system (that is, Cluster, Management, and Live Migration). This approach reduces the number of physical NICs required on your hosts and also lets you leverage higher capacity interfaces in a highly customizable manner. Figure 4-6 provides just one example of how host interfaces can be

easily configured to redesign a networking implementation. This approach of utilizing host-based vNICs to consolidate multiple networks to a set of host physical interfaces is commonly referred to as a converged network.

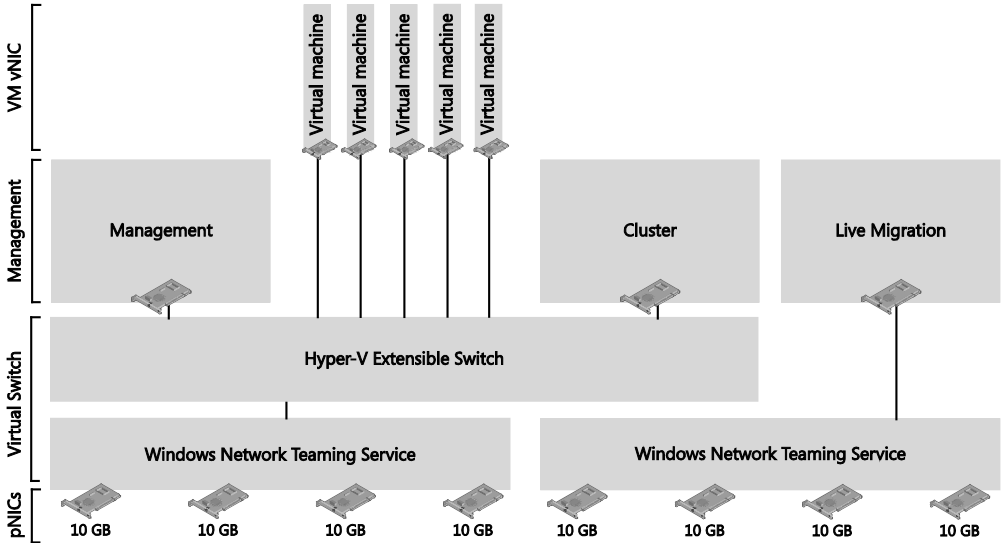


FIGURE 4-6 Utilizing vNICs to consolidate multiple networks

Combining this new ability with QoS, you can effectively define the bandwidth available on each network. Whereas previously you would add additional physical NICs to the desired team, now you can implement much larger scalability with one or more 10-GB NICs and distribute the capacity to the relevant networks through vNICs.

NOTE **The actual implementation methodology for establishing your management interface might be different based on your physical hardware. For example, it is common practice for rack mount servers to be supplied with two onboard interfaces, which are regularly allocated to the role of management. In a converged network implementation, the common practice is to disable these interfaces in favor of leveraging higher capacity interfaces and defining a virtual network interface on the converged configuration to host the management workload.**

Network adapter teaming

Generally, a NIC team will be created from two or more physical network adapters. But a team can also contain just a single adapter (essentially a team of one). This point is relevant to design considerations since, if you want to increase the bandwidth available to logical networks supported by a given switch, you can simply add a new physical network adapter to the host and join this adapter to the team.

Certain circumstances, such as with physical network adapters, support SR-IOV or are dedicated to SMB 3.0, where this kind of approach is not appropriate. For example, correctly implemented iSCSI workloads normally use two NICs on separate subnets, each of which are then connected to their respective storage target hosted on the subnets, while leveraging the features of multipath I/O (MPIO) to attain high availability.

As a second example, Windows Server 2012 introduced SMB 3.0, a file-based storage alternative to the iSCSI block storage previously relied upon for VM storage. SMB 3.0 includes a set of sophisticated, network-aware features that offer dramatic performance enhancements, for example SMB Multichannel (see *http://blogs.technet.com/b/josebda/archive/2012/05/13 / the-basics-of-smb-multichannel-a-feature-of-windows-server-2012-and-smb-3-0.aspx*). These multichannel concepts are implemented physically, similar to iSCSI, where the associated NICs are not teamed.

As a result of such considerations, you should carefully review each workload to determine whether or not physical network adapter teaming is appropriate and will help you to realize performance benefits.

NOTE **VMM is unable to create a host-based network team without also establishing a logical switch. This configuration would be inappropriate for the example iSCSI and SMB 3.0 workloads discussed in this section, which should obviously be implemented without virtual interfaces.**

In addition, when deploying teams for your hosts, you must also consider the networking switches your hosts will be uplinked with. These decisions will include the teaming mode and load balancing decisions, as addressed in the discussion of uplink port profiles in Chapter 3, "Hyper-V port profiles."

Another consideration with respect to teaming is the access mode you configure. You will need to choose either Trunk or Access mode and also ensure that this setting matches on both the host computer (as defined in the uplink port profile) and the physical switch.

Virtual high bandwidth adapters (HBAs)

Virtual HBA interfaces are not supported in VMM 2012 SP1, but are supported in VMM 2012 R2. These interfaces allow you run the Failover Clustering feature inside the guest operating system of a VM connected to share fibre channel storage. For more information on VMM support for HBAs, refer to *http://blogs.technet.com/b/privatecloud/archive/2013/07/23/hyper-v-virtual-fibre-channel-design-guide.aspx*. Virtual HBAs are *not* managed through the VMM logical switch environment, but are mentioned here for reference.

VMM availability and logical switches

If your highly available VMM environment fails, your first concern might be whether all of your logical switches will fail. Since logical switches are templates, they are only relevant at the point of deployment or change and really have no influence on the day-to-day traffic flow of your hosts. A more relevant concern might be determining how to keep the VM load optimally distributed on the hosts or how to ensure your tenants can access their clouds.

The point is that VMM does not just configure the network in this configuration, but also transparently manages the hosts to ensure that the network virtualization filters are capable of delivering the traffic to its destination. This function is quite complex, and is one of the primary reasons why you should not to try to manage software-defined networking without VMM and logical switches.

Behind the scenes, every time a VM that is connected to a virtualized network is moved between hosts, VMM updates the extensions on all relevant hosts with details of this environmental change. If VMM is not running, and a VM is moved by an external influence, that VM will effectively drop off the network. Of course, as soon as VMM recovers, it scans for environment changes and updates all the extensions as quickly as possible. Therefore, if VMM is unavailable, and you are using Virtual Networks, then you should aim to ensure that no external influence moves any VM connected to the virtual network. This will ensure everything remains healthy until VMM is restored.

How many logical switches do you need?

The goal in this section is to present a step-by-step approach to creating logical switches, starting from the basic principle that you should begin as simple as possible and then add additional logical switches only where there is a compelling business or technical reason to do so. The process can be summarized as follows:

1. Review the environment in which logical switches will be deployed.
2. Determine whether switch extensions need to be scoped to a specific host group.
3. Determine whether different QoS modes or traffic policies are required for logical networks.
4. Determine whether logical networks are restricted to a specific group of hosts.
5. Review the circumstances in which you should *not* create a logical switch.

As usual, defining and adhering to a sound naming convention for logical switches is important both to promote understanding and to help simplify management and reduce cost.

You need at least one logical switch to take advantage of the capabilities offered by VMM. This section examines some of the main reasons why you would (or would not) want to create additional switches and provides an overview of important considerations, best practices, and key hardware variances.

Step 1: Review the environment in which logical switches will be deployed

The first thing you need to do is consider the physical environment in which logical switches will be deployed, as well as some of the other technologies that are being utilized in the environment since these may have bearing on the number of switches you need.

Converged networks

Understanding some of the decision drivers associated with teaming host NICs, you can begin to appreciate some of the new networking options available. One option mentioned briefly at the beginning of the chapter is a fully converged design, where all the hosts' physical interfaces are bonded into a single team, which is uplinked to the logical switch. From the switch, you can carve a number of host virtual NICs, which you can then dedicate to your primary networks.

The fully converged configuration, as illustrated in Figure 4-7, offers a very simple host configuration with a lot of flexibility and is good for simple hosts that don't need to leverage too many hardware-based network enhancements or offloads.

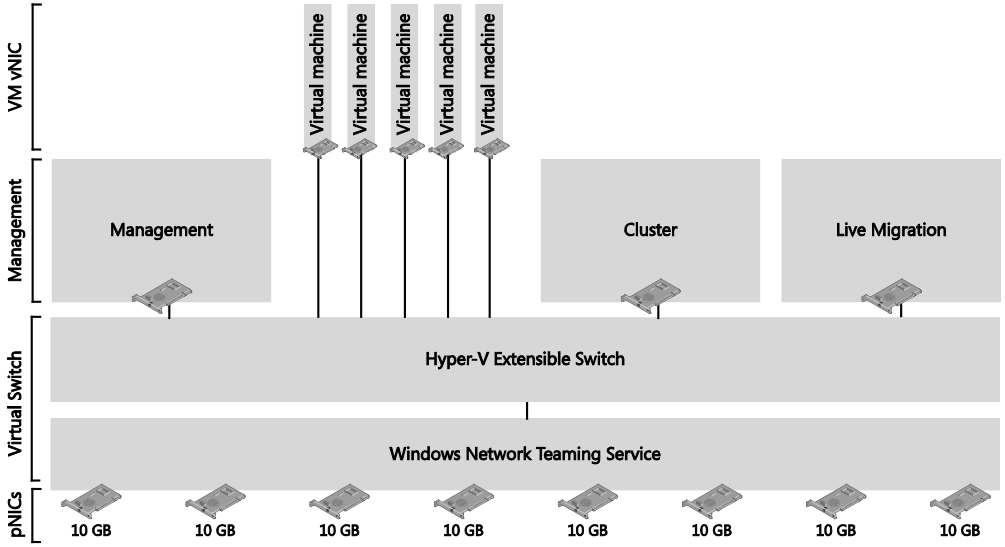


FIGURE 4-7 Fully converged network example

Dedicated VM switch

A common alternative design to the fully converged example is to implement the configuration utilizing two logical switches (see Figure 4-8), one dedicated to the traffic for VMs and a second logical switch to address the management-related traffic for hosts. In this scenario, QoS is implemented on the Management switch and the focus is on ensuring that the availably of the host workloads are running as optimally as possible. Similar to the fully

converged implementation, this approach again restricts the ability to leverage hardware enhancements and offloads.

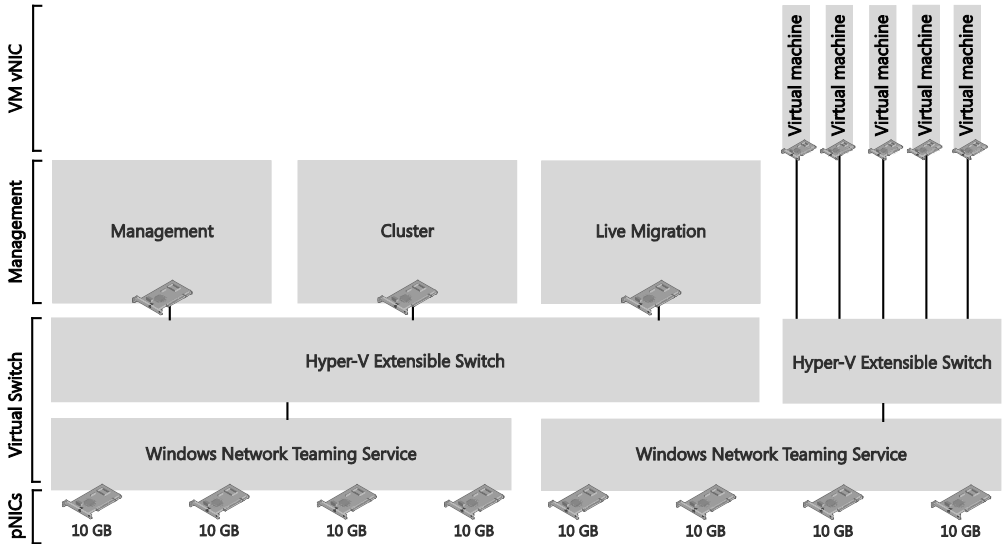


FIGURE 4-8 Implementing with dual logical switches

NOTE **You may need to take additional steps to optimize access to storage, effectively bypassing the Hyper-V switch for SMB v3 traffic. These optimizations are implemented outside of VMM. You can find more information at** *http://blogs.technet.com/b/josebda/archive/2013/10/09/networking-configurations-for-hyper-v-over-smb-in-windows-server-2012-and-windows-server-2012-r2.aspx*

Converged iSCSI

For access to iSCSI resources, network adapters are not normally teamed; however, it is possible to have these networks presented in a converged design, still presenting two vNICs to ensure that the requirement of separate subnets for each path is sustained. However, in doing so, you cannot guarantee that iSCSI traffic will be distributed on two different physical switches if that is a support requirement for your storage vendor (i.e., converged networking is not supported).

In this design, shown in Figure 4-9, you must consider implementing a minimum bandwidth constraint on the QoS or you will risk starving the storage, potentially on both paths. As with normal iSCSI configuration, your MPIO agent must still be configured.

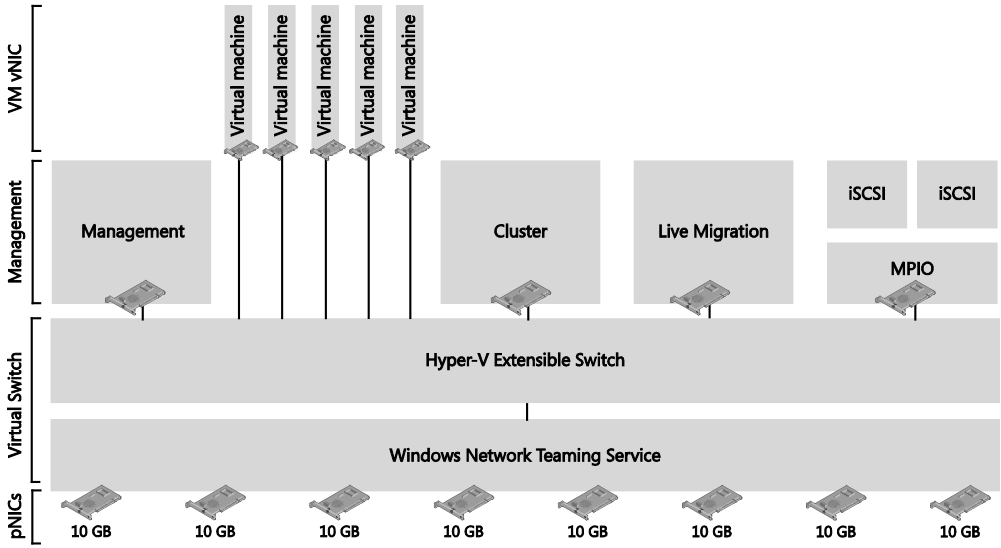


FIGURE 4-9 Converged iSCSI

SMB Direct and SMB Multichannel

Taking a slightly different approach, you can deviate from an iSCSI-based storage concept to a design that supports SMB 3.0 file-based storage. Essentially, the primary difference in this design, shown in Figure 4-10, is the segregation of storage-related interfaces from the rest of the primary and VM networks. As with the iSCSI design, the interfaces are not teamed, ensuring that SMB Multichannel (receive side scaling, or RSS) is supported.

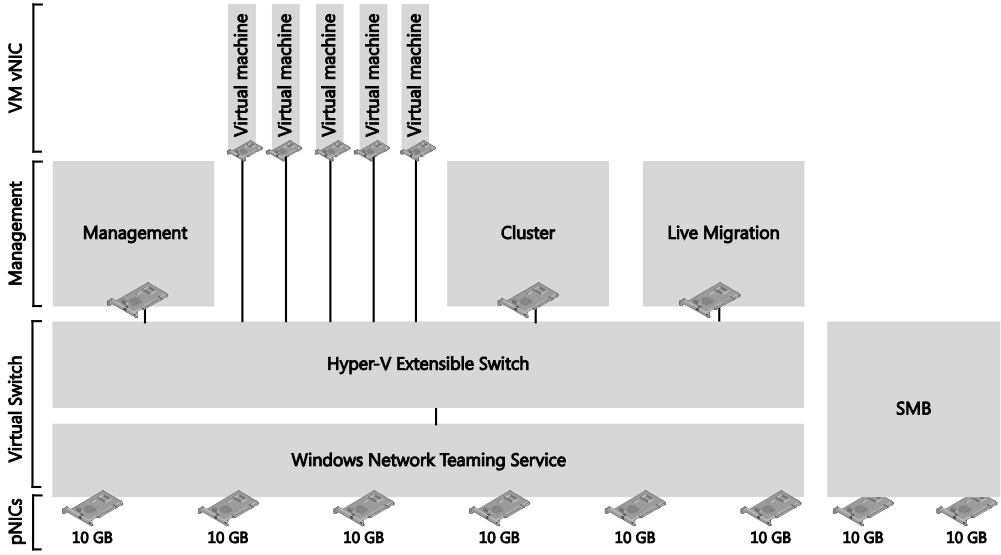


FIGURE 4-10 SMB 3.0 file-based storage

This design can also be implemented for iSCSI storage, again retaining support for MPIO. If at a later time you decide to migrate to SMB 3.0, you can add additional interfaces or repurpose one of the existing interfaces as you complete migrations.

Processor offload using SR-IOV

All the assumptions made in this chapter so far have been based on the premise of utilizing a software layer to connect hosted VMs to the underpinning physical network. A software-based design will never be as efficient as utilizing a hardware-based approach, which is now possible through the use of SR-IOV–enabled network interfaces, which are based on the PCI-SIG I/O Virtualization (IOV) specification published at *http://www.pcisig.com/specifications/iov/*.

When a Hyper-V virtual NIC is enabled for SR-IOV, it is no longer connected to the virtual switch. Instead, the virtual NIC connects to a PCIe feature on the SR-IOV NIC referred to as the Virtual Functions (VF), which then channels the traffic to the physical network. These VFs are isolated and secure, but have no configuration options, and the number of VFs available is determined by the SR-IOV NIC utilized. For example, most of the popular cards support 256 VFs, which simply implies that your host can have 256 virtual NICs.

NOTE **Due to SR-IOV bypassing most of the networking stack, SR-IOV NICs cannot be teamed on the host operating system, and you should not enable any policies, such as QoS.**

When you create a logical switch in VMM, the first page in the Create Logical Switch Wizard queries whether to enable SR-IOV for this switch, as shown in Figure 4-11. This is a onetime decision. If at a later time you re-consider, you will need to replace the switch since it is not possible to change this setting after it is defined.

If you may have multiple NICs on your host, you may determine that a portion of these are SR-IOV capable and want to leverage their abilities, while also utilizing the remaining NICs for other traffic. In this case, you should create at least two logical switches.

Finally, a maximum of eight SR-IOV–enabled connections can be assigned from your VM vNIC connections. You can find more details on SR-IOV at *http://blogs.technet.com/b/jhoward/archive/2012/03/12/everything-you-wanted-to-know-about-sr-iov-in-hyper-v-part-1.aspx*.

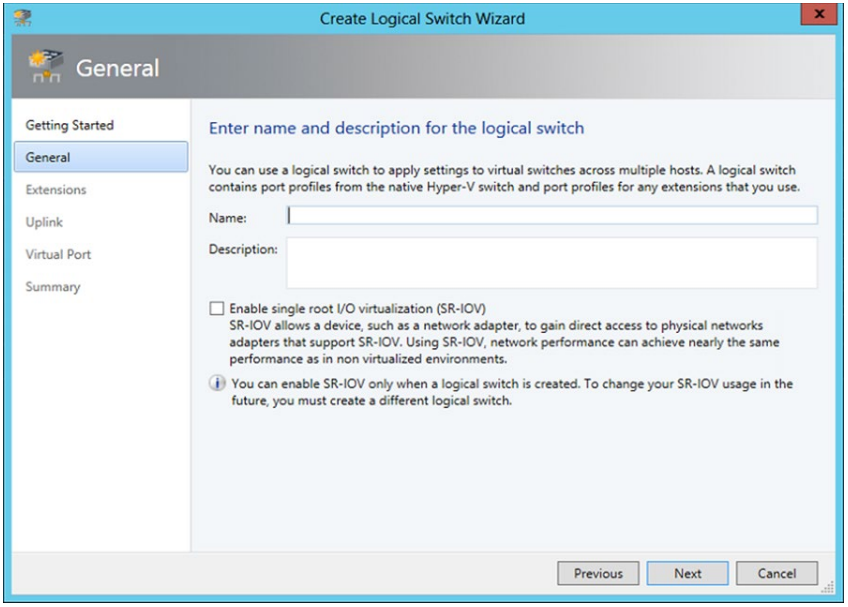


FIGURE 4-11 Option to enable SR-IOV during the creation of a logical switch

Remote Direct Memory Access (RDMA)

Assuming you want to utilize RDMA-enabled NICs, for example iWARP, RoCE, or InfiniBand, do not team these interfaces since by design these cards actually bypass the operating system teaming functions. Instead, connect these interfaces directly to the same subnet as your storage. Similar to iSCSI paths, each of these NICs will also reside on different subnets, ensuring that your storage will also operate with SMB Multichannel. As you will appreciate, there is no QoS on these interfaces since they bypass the operating system packet scheduler; however, this is not a problem since these cards will be leveraged for the purpose of storage traffic only. This design will offer you extremely scalable storage access, offering throughput of beyond 16 gigabytes (GB) per second, with just 5 percent of CPU utilization.

Another spin on the SMB Multichannel design is to segregate the Live Migration network so it also utilizes RDMA interfaces. This approach is akin to the Hyper-V 2008 usage of 10-GB interfaces to help accelerate the migration of VMs from your hosts for maintenance; however, the scale and performance of this design is so fast that the bottleneck is no longer between the network connection and the host, but instead the speed of the RAM bus itself.

What about the logical switch? You can connect RDMA interfaces to a logical switch, but associated VM NICs will not be RDMA capable. As the cost of these interfaces is still higher than regular 10-GB NICs, you will generally reserve these interfaces for host networks.

RDMA is enabled by default on Windows Server 2012. To disable and enable the feature on a specific interface, you can use Windows PowerShell as follows:

```
Disable-NetAdapterRdma <name>
Enable-NetAdapterRdma <name>
```

You can also disable RDMA globally on your host with the following command:

```
Set-NetOffloadGlobalSettings -NetworkDirect Disabled
Set-NetOffloadGlobalSettings -NetworkDirect Enabled
```

Although there are no switch extensions included in the preceding example, you might want to include these in your logical switch to allow you to monitor network traffic, use QoS to control how network bandwidth is used, enhance the level of security, or otherwise expand the capabilities of a Hyper-V virtual switch created on a host computer. If these enhanced services should be restricted to/or deployed on a limited number of hosts, you might need to consider creating an additional logical switch. You can begin utilizing logical switches without any advanced switch extensions, and as the environment matures or your requirements change, these can be added at a later time.

Step 2: Enhancing logical switch capabilities

The only extension considered in this document thus far has been the Network Virtualization extension, which, as the name suggests, is required for network virtualization. However, a number of third-party vendors have both free and commercial extensions readily available. The methods implemented for these switches are based on a single-integration API, as illustrated in Figure 4-12. This scenario uses one of the third-party extensions, in this case the free version of Cisco's Nexus 1000v.

When the Nexus 1000v is deployed, a new extension is added to the host logical switch, which vastly enhances the functionality of the virtual switch with features found in Cisco's enterprise Nexus range of physical switches. Among these features is what would be familiar to network administrators as the Cisco Network Management Console (or the Nexus OS CLI); it is from these interfaces your network administrators will conduct much of their daily network management duties, including configuring ports, profiles, switches, and so on.

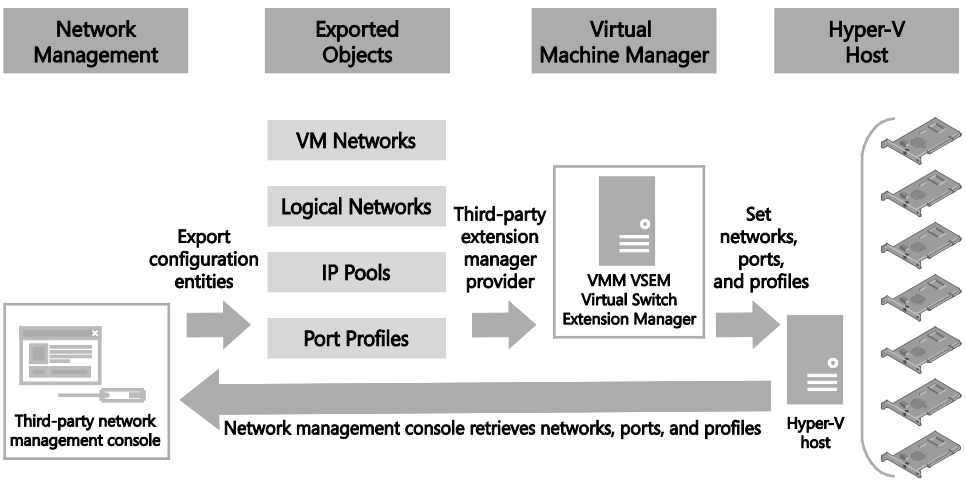


FIGURE 4-12 Extension Manager integration

When integrated with VMM, all the entities that the network administrator creates, manages, and maintains through the Cisco Network Management Console are transferred to VMM. VMM continues to manage its services as normal, deploying new VMs to the respective hosts, which are ultimately connected to a port on the virtual switch. This information is presented back to the network administrator who continues to manage the device in the native tools with enriched details added by VMM. For example, each virtual network port connected to a VM will include an always-up-to-date description field containing the name of the VM associated with the port.

This updating process continues in its cyclic fashion, permitting the network for clouds to be managed by the network administrators, leveraging the fabric administrators to focus on the rest of their responsibilities.

Deploying logical switch extensions

VMM continues to leverage much of the same infrastructure and agents that are already in place to manage its hosts and deploy logical switches as you consider adding third-party extensions to your environment. The installation of the third-party extensions are all relatively similar. You only need to add the product to VMM as a console add-in, networking services extension, or both, a process that is trivial in most cases and sometimes fully automated through the respective product installers.

When integrated with VMM, network extensions are added to your logical switch, which allows you to leverage the flexibility of potentially deploying multiple logical switches to your hosts. When you associate a logical switch with a host, VMM automatically discovers whether the logical switch has defined the use of an extension, checks whether the extension is deployed to the host, and then resolves the host when this is not the case. This simplified management approach orchestrated within VMM continues to ensure that all extensions are deployed and configured in a consistent manner, and only to hosts that require the extension to support their logical switch compliance.

Combining Hyper-V Network Virtualization with extensions

One of the key benefits introduced with the Hyper-V virtual switch extensibility features was the ability to deploy multiple extensions to the switch simultaneously. This design makes it possible, for example, to enable Hyper-V Network Virtualization while also supporting the Cisco Nexus 1000v.

NOTE **A problem was identified in the placement of extensions on the virtual switch in Windows Server 2012 that prevented the combined usage of NVGRE and Cisco Nexus 1000v. This problem is resolved with Windows Server 2012 R2.**

See also *The virtual switch design, including its ability to support multiple simulations extensions, is covered in detail on MSDN at http://msdn.microsoft.com/en-us/library /windows/hardware/hh582268(v=vs.85).aspx.*

Step 3: Determine whether different QoS modes or traffic policies are required for logical networks

Windows Server 2012 includes new QoS bandwidth management features that enable cloud hosting providers and enterprises to provide services that deliver predictable network performance to VMs on a server running Hyper-V. These settings are defined within the logical switch.

A logical switch can work in only one mode at a time, absolute or relative. By default, a logical switch functions in relative mode (weight based) rather than absolute. You can override this default, however, by using the following Windows PowerShell command:

```
Set-SCLogicalSwitch -MinimumBandwidthMode Absolute
```

It is not possible to change the minimum bandwidth mode for a logical switch using the VMM console. VMM defaults to relative mode for all of its logical switches. Should you require to support both relative and absolute QoS configurations in your environment, you must deploy a different logical switch for each minimum bandwidth mode.

To more closely consider these approaches and the value each offers, consider a simple example of a pair of 10-GB NICs, configured in a team. Figure 4-13 shows the distribution of the primary networks as they might be distributed across this sample team, implemented using either an absolute or relative approach.

In this example, the absolute assignment divides the total 20 GB of capacity into the following allocations:

- VM Traffic, set absolute to 10 GB
- Management, set absolute to 2 GB
- Cluster, set absolute to 2 GB
- Live Migration, set absolute to 6 GB

For the weight-based QoS approach, instead of assigning an absolute bandwidth value, the example distributes the bandwidth to each network from a combined total weight of 100 as follows:

- VM Traffic, set relative weight of 50 out of 100 = 20 GB * 50/100 = 10 GB
- Management, set relative weight of 10 out of 100 = 20 GB * 10/100 = 2 GB
- Cluster, set relative weight of 10 out of 100 = 20 GB * 10/100 = 2 GB
- Live Migration, set relative weight of 30 out of 100 = 20 GB * 30/100 = 6 GB

The final result of both configurations provides the same bandwidth distribution independent of the selected QoS approach (as in Figure 4-13).

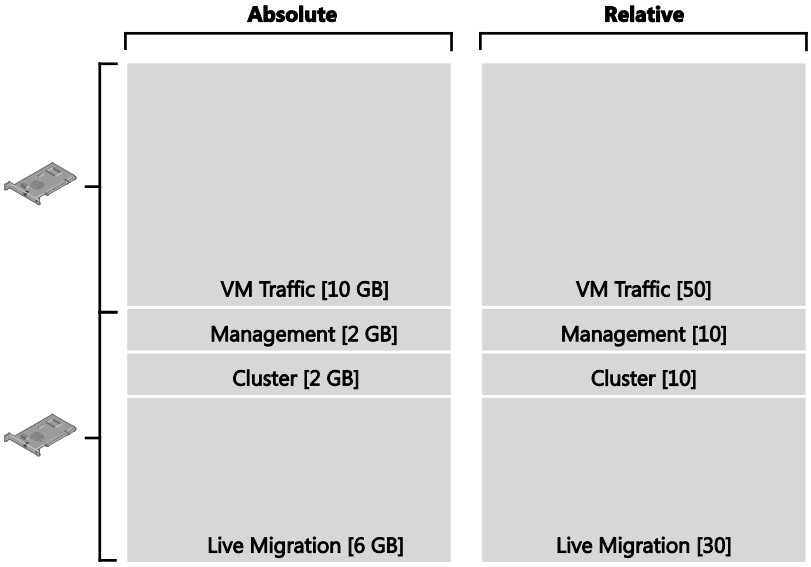


FIGURE 4-13 Distributing 20 GB of bandwidth using absolute and relative weighting

Alone, neither of these two concepts is perfect, but the flexibility of the relative approach makes it more appropriate for normal deployments. In addition, this approach offers the flexibility of applying minimum or maximum bandwidth settings. Maximum bandwidth settings restrict a VM from consuming more than a specified amount of capacity, which is useful in some scenarios, such as limiting tenants to the bandwidth they pay for. Minimum bandwidth settings, however, are far more useful, essentially enabling SLA guarantees. Using a weight-based approach combined with minimum bandwidth settings can guarantee enough capacity to deliver service, for example to important cluster networks, but still retain bandwidth capacity for use by other networks with higher weights.

Figure 4-14 shows a minimum bandwidth setting applied to relative weighted networks. In this example, 5 percent of the total bandwidth is defined for the default VM Traffic and the Custer networks, respectively.

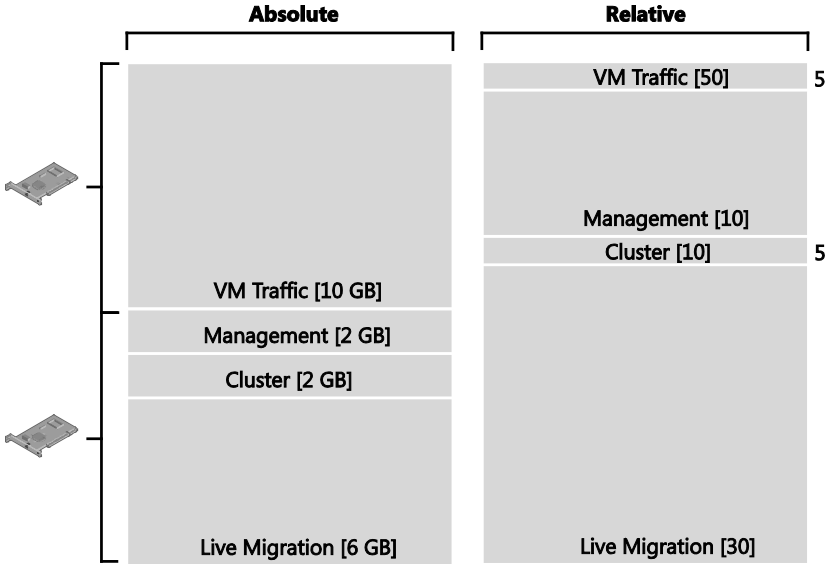


FIGURE 4-14 Adding minimum bandwidth to relative weights

Taking these reservations into account, the calculations of available bandwidth work as follows

- Ten percent of the overall 20 GB available is reserved and accounts for 2 GB of bandwidth, as follows:
  - VM Traffic, set to minimum reservation of 5 percent = 20 GB * 5% = 1 GB
  - Cluster, set to minimum reservation of 5 percent = 20 GB * 5% = 1 GB

The remaining 90 percent, or 18 GB, of bandwidth is allocated as a share of the total remaining weight, which in this case is 10 points and 30 points, respectively, or 40 points in total, as follows:

- Management, set relative weight of 10 out of 40 = 18 GB * 10/40 = 4.5 GB
- Live Migration, set relative weight of 30 out of 40 = 20 GB * 30/90 = 13.5 GB

If this information is combined with the previous weight-only approach, the variable nature of capacity to take advantage of can be defined as follows:

- VM Traffic, min of 1 GB and maximum of 10 GB
- Management, minimum of 0 GB, maximum of 4.5 GB, and full utilization reservation of 2 GB
- Cluster, minimum of 1 GB and maximum of 2 GB
- Live Migration, minimum of 0 GB, maximum of 13.5 GB, and full utilization reservation of 6 GB

These examples illustrate the flexibility offered by the combined use of weights and minimum reservations. However, to truly appreciate this value, consider the same example again, but this time imagine one of the two 10-GB network paths has failed (see Figure 4-15).

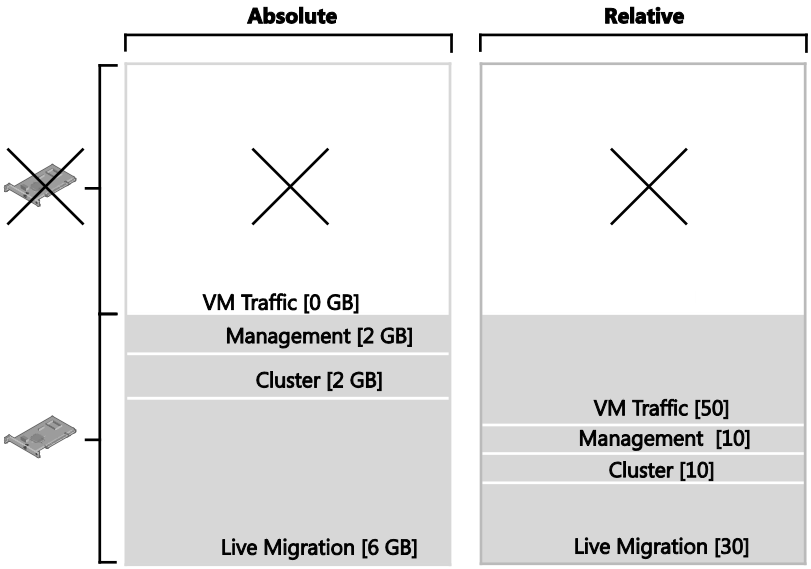


FIGURE 4-15 Bandwidth weights and minimum reservations with a 10-GB path failure

On first inspection of Figure 4-15, it is apparent that the absolute approach now has a major implementation issue. With the loss of 10 GB of bandwidth, the capacity to sustain the VM Traffic load is effectively lost. This would result in a service interruption or, worse, complete loss of VM Traffic.

However, the weight-based QoS simply readjusts to the new capacity available, and, as the following calculations show, business can continue, although it might be a little slower:

- Ten percent of the overall 10 GB available is reserved and accounts for 1 GB of bandwidth, as follows:
- VM Traffic, set to minimum reservation of 5 percent = 10 GB * 5% = 0.5 GB
- Cluster, set to minimum reservation of 5 percent = 10 GB * 5% = 0.5 GB

The remaining 90 percent, or 9 GB, of bandwidth is allocated as a share of the total remaining weight, which in this case is 10 points and 30 points, respectively, or 40 points in total, as follows:

- Management, set relative weight of 10 out of 40 = 9 GB * 10/40 = 2.25 GB
- Live Migration, set relative weight of 30 out of 40 = 9 GB * 30/90 = 6.75 GB

Of course, weights offer flexibility. The following recalculations can help you understand the variable amounts of bandwidth available:

- VM Traffic, minimum of 0.5 GB and maximum of 5 GB
- Management, minimum of 0 GB, maximum of 2.25 GB, and full utilization reservation of 1 GB
- Cluster, minimum of 0.5 GB and maximum of 1 GB
- Live Migration, minimum of 0 GB, maximum of 6.75 GB, and full utilization reservation of 3 GB

As you embrace the new flexibility offered in Windows Server 2012 through the use of QoS, you can consider new approaches to designing your network implementations. You can apply QoS several different ways in Windows Server 2012, depending on the planned network configuration:

- Virtual switch
- Server network (that is, host networks not connected to a virtual switch)
- Windows QoS Packet Scheduler
- Data Center Bridging (DBC) for hardware-based QoS when all features end to end support DBC functionality. (More information on this technology is available at *http://msdn.microsoft.com/en-us/library/windows/hardware/hh440120(v=vs.85).aspx*)
- Offloaded networking, which bypasses the operating system stack, for example SR-IOV and RDMA

Step 4: Determine whether logical networks are restricted to a specific group of hosts

As a general principle, you will need a logical switch for each physical network that exists in your environment, but if you plan to restrict some logical networks to a limited set of hosts, as with the example organization in this chapter, or if you have custom connectivity requirements, you might find it necessary to create additional logical switches. Technically, nothing prevents you from including all of your uplink port profiles into the logical switch, allowing your administrators to choose the most appropriate settings and capabilities for the host they are working with.

The problem with this approach is that you cannot be sure of a consistent configuration across hosts in production. Although uplink port profiles are restricted to certain hosts, administrators can choose from any of the network adapter port profiles, port classifications, and switch extensions that are available within the selected logical switch. You might also find that capabilities you want offered only on production systems—network traffic tagged with IEEE high priority and given maximum bandwidth, for example—are associated with other (non-production) systems because the administrator selected the wrong network adapter port profile during logical switch deployment. To avoid this issue, you should consider creating logical switches for specific workload types.

NOTE **The host must be a member of a host group that has been scoped to those logical networks. If the host is not in an appropriate host group, deployment of the switch will fail with an Out of Scope error.**

Step 5: Review the circumstances in which you should *not* create a logical switch

For some specialist network adapters, it is not recommended or appropriate to use a logical switch because the enhanced benefits of these interfaces would be disabled or degraded. In the case of RDMA, for example, placing a logical switch on the adapter prevents the interface from leveraging direct memory access, reducing its speed significantly.

CHAPTER 5

Network Virtualization gateway

Most customer deployments require communication from the virtualized network environment to the non-virtualized network environment. Hyper-V Network Virtualization gateways are required to route between the two environments. Gateways take various forms. They can be built on Windows Server 2012, incorporated into a Top of Rack (TOR) switch, put into an existing network appliance, or exist as a stand-alone network appliance.

This chapter will:

- Provide a brief overview of how Network Virtualization works
- Explain how a gateway provides connectivity between VM networks and external networks
- Discuss different scenarios in which a gateway is required
- Introduce a step-by-step process for determining the factors that need to be considered when deploying the gateway

How Network Virtualization works

Hyper-V Network Virtualization provides virtual networks (called VM networks) to virtual machines (VMs) similar to how server virtualization (hypervisor) provides VMs to an operating system. Network Virtualization decouples virtual networks from the physical network infrastructure and removes the constraints of VLAN and hierarchical IP address assignment from VM provisioning. This flexibility makes it easy for customers to move to IaaS clouds and makes managing infrastructure efficient for hosters and datacenter administrators, while maintaining the necessary multi-tenant isolation and security requirements and supporting overlapping VM IP addresses.

Network Virtualization allows VM networks to be overlaid on logical networks by using NVGRE encapsulation. As explained in Chapter 2, "Logical networks," NVGRE encapsulation enables isolation and identification of traffic between multiple tenants' VMs with overlapping IP addresses.

Figure 5-1 shows an NVGRE-encapsulated packet. On the wire, NVGRE-encapsulated packets look like IP-over-Ethernet packets, except that the payload of the outer IP header is a GRE-encapsulated IP packet (including the Ethernet header). For more details, refer to Chapter 1 of the free ebook from Microsoft Press titled "Microsoft System Center: Network Virtualization and Cloud Computing," which can be downloaded from the Microsoft Virtual Academy (MVA) at *http://www.microsoftvirtualacademy.com/ebooks*.

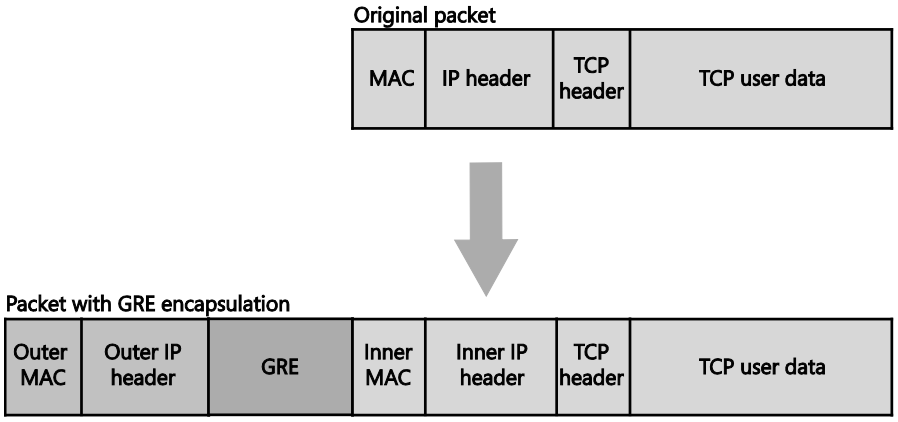


FIGURE 5-1 Network virtualization using GRE

Tenant VMs in a Network Virtualization network send Ethernet-encapsulated IP packets out of the VM NIC. The Network Virtualization feature in Hyper-V on each host adds GRE encapsulation. As long as the sender and receiver of IP packets are VMs in the Network Virtualization network, for example between VM-A and VM-B in Figure 5-2, Hyper-V hosts add and remove GRE encapsulation. When VMs in a Network Virtualization network send packets to non-Network Virtualization networks, such as to an Internet or hoster infrastructure network or a tenant on-premises network, NVGRE encapsulation must be removed and IP packets that are routable in the non-Network Virtualization network must be sent out.

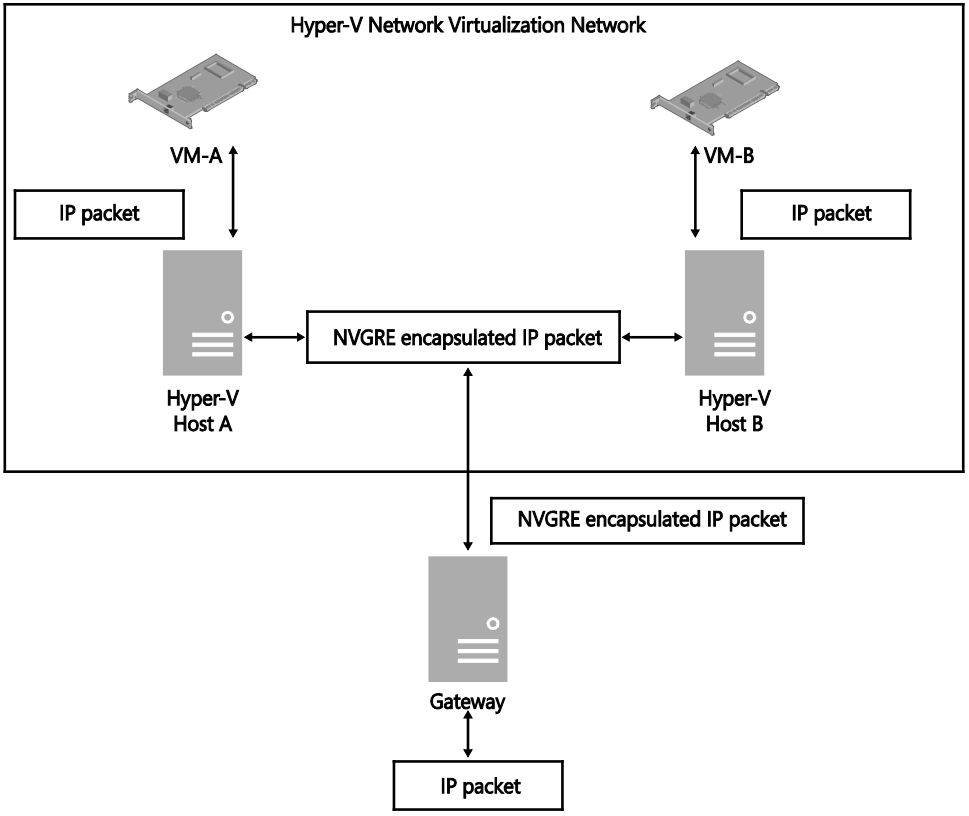


FIGURE 5-2 Overview of an NVGRE network

Similarly, when IP packets from an external network are sent to VMs in a Network Virtualization network, for example to VM-B in Figure 5-2, the gateway must add the necessary GRE encapsulation and route the packet to VM-B. While doing so, the gateway needs to determine which tenant the packet belongs to and set the appropriate GRE key in the GRE header so that the packet is delivered to the correct VM, even if IP addresses of different tenant VMs overlap. The exact packet processing the gateway must perform depends on the external network to which the tenant VM network connects.

Designing the virtualized network solution

The process for designing a virtualized network solution that supports inter-network connectivity can be summarized as follows:

1. Begin by understanding the connectivity requirements for different scenarios where external connectivity to VM networks is required.
2. Evaluate different mechanisms for enabling the scenarios using different features of the gateway.
3. Identify the factors that need to be considered in selecting the appropriate gateway functionality for the scenarios.
4. Understand deployment guidelines and constraints.

The following sections expand on each of these steps, provide insight into the use of the gateway in specific scenarios, and, where relevant and appropriate, call out best practice recommendations for the design and implementation of the gateway.

Understanding connectivity requirements: When is a gateway required?

Most customer deployments require communication from the network virtualized environment to the non-network virtualized environment. Hyper-V Network Virtualization gateways are required to bridge the two environments. Gateways can vary in form. This section covers different scenarios for enabling connectivity between VMs in VM networks and external networks. The following are the broad set of scenarios where a gateway is required:

- **Connectivity to enterprise applications** For most of the enterprises that leverage the benefit of public or hosted clouds, enabling secure connectivity to on-premises applications is essential. For enterprises that are born in the cloud, secure connectivity to VM networks is required for enterprise employees. This class of connectivity is covered in the section, "Connectivity to enterprise applications."
- **Internet connectivity and publishing** Businesses that have Internet-based applications can leverage the benefits of public or hosted clouds by deploying their applications in service provider premises. Service providers like Fabrikam, the example organization described in previous chapters, provide such an infrastructure for multiple businesses. They enable isolation for each tenant by deploying tenants' VMs in their respective VM networks. For the applications to be accessible from the Internet, a mechanism is needed to publish applications or services offered by VMs in VM networks. When VMs in a tenant VM network have to access Internet services like public DNS, the private IP addresses of the tenant VMs must be translated to Internet addresses. Details of this class of connectivity is covered in the section, "Internet connectivity and publishing."

- **Connectivity to shared services** VMs in VM networks typically require services such as DNS or protection. These services can be provided by service providers in their internal network. Details of this class of connectivity is covered in the section, "Connectivity to shared services."
- **Connectivity to legacy networks** While Network Virtualization is a new technology, some hosters and enterprises could already be leveraging VLAN-based isolation for tenant networks. For enterprises or hosters to gradually migrate tenant applications from VLAN-based networks to Network Virtualization-based networks in a phased manner, connectivity between Network Virtualization VM networks and VLAN networks is required. Details of this class of connectivity is covered in the section, "Connectivity to legacy networks."

Connectivity to enterprise applications

Access to VM networks is required when an enterprise has migrated some of its application workloads to the service provider infrastructure and there is a need for end users or workloads (either from the on-premises network or external networks) to access them.

> **NOTE** **In this case, "access" to workloads means the client applications like a SQL client querying a database server. Cloud service providers can allow administrative access to actual VMs in VM networks for diagnostic or management purposes to business administrators via Remote Desktop Protocol or similar mechanisms.**

Connectivity to enterprise applications and users from VM networks can be provided in the following ways:

- **Connectivity between enterprise networks and VM networks via site-to-site (S2S VPN)** This is network-to-network connectivity between enterprise networks in tenant premises and VM networks in hoster premises—generally required when enterprises have moved one or more workloads to the hoster and there is a need to access or connect to those workloads as if they were still located on-premises.
- **Enabling enterprise users to access VM networks (Remote Access VPN)** This is machine-to-network connectivity or point-to-site connectivity that is required when enterprise employees have to access workloads in VM networks from outside their corporate network.

Both modes of connectivity are typically referred to as *hybrid connectivity mode*.

Connectivity between enterprise networks and VM networks

A key advantage of Hyper-V Network Virtualization is that it can seamlessly extend an on-premises datacenter to a Windows Server 2012-based cloud datacenter. In this scenario, an internal server, such as a web server, is moved from the enterprise network into a cloud hoster's datacenter. Taking advantage of bring-your-own-IP-address offered by the hoster, the

enterprise does not need to change the network configuration of the web server VM. The hoster provides a secure link via a VPN gateway appliance in one of the two methods outlined below. The enterprise administrators only need to configure their on-premises VPN with the appropriate IP address.

- **Secure connectivity over public Internet** This mode of connectivity is a quick and inexpensive option to connect securely over public Internet since no new infrastructure is required in the enterprise site. It is generally suitable for applications with bandwidth requirements in the order of a few hundred megabits per second. Existing edge devices that support IPsec (IKEv2) VPN tunnels can be used as connection endpoints.
- **Connectivity over high speed dedicated tunnels** This mode of connectivity is generally provided by Internet service providers (ISP) through multi-protocol label switching (MPLS) circuits. These are typically of speeds in the order of gigabits per second. Since these circuits are dedicated for the tenant, ISPs typically charge a premium for this service.

To help illustrate these points, consider the example hoster Fabrikam. Contoso and Woodgrove Bank are two tenants that have created virtual networks in the Fabrikam cloud. The tenants want to access any resources connected to their specific virtual network and, just as importantly, to treat their virtual network as a seamless extension of their own on-premises infrastructure. To support this requirement, Fabrikam uses a multi-tenant VPN gateway on the Cloud GW Cluster, as shown in Figure 5-3.

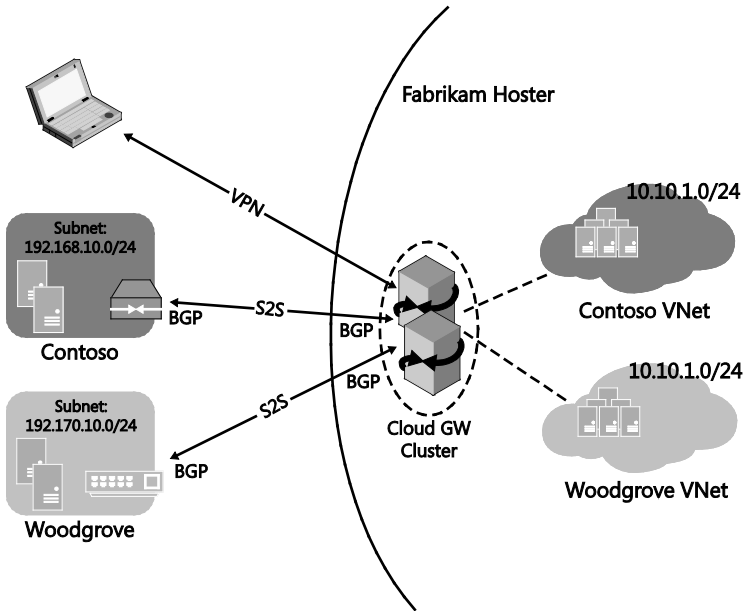


FIGURE 5-3 Enterprise-to-cloud connectivity

VPN tunnels over Internet Secure connectivity over public Internet is generally ensured through an IPsec VPN tunnel (also called site-to-site VPN or on-demand VPN) which is established between the customer's on-premises network and the service provider's cloud. IPsec VPN tunnels encrypt the traffic to ensure confidentiality. You'll find more details at *https://technet.microsoft.com/en-us/library/cc179879.aspx*.

To enable enterprise sites to connect to the cloud, Fabrikam deploys a Windows Server 2012 R2 multi-tenant gateway. A single multi-tenant gateway can serve multiple tenants and exposes a single public IP address for tenants to connect to. This greatly reduces the number of gateways and the number of public IPs that Fabrikam must deploy to cater to all of its tenants. The multi-tenant gateway can be configured differently for every tenant and also ensures traffic isolation between them. Contoso and Woodgrove deploy gateways at their on-premises networks and set up IKEv2 site-to-site VPN tunnels to the Fabrikam cloud gateway. There are a couple of options available for an on-premises gateway:

- Enterprises can deploy Windows Server 2012 or Windows Server 2012 R2, both of which support IKEv2 VPN tunnels.
- Enterprises can use a third-party edge router or other edge device that supports IKEv2 VPN tunnels. The Windows Server 2012 R2 gateway interoperates with routers and edge devices from leading vendors. You can find more information about this at *http://blogs.technet.com/b/networking/archive/2014/12/26/vpn-interoperability-guide-for-windows-server-2012-r2.aspx*.

The gateways on each side can be configured to either keep the IKEv2 tunnel permanently established or automatically set it up only when there is traffic.

To ensure high availability of the cloud gateway, given the business critical connections that it is terminating, Fabrikam can deploy another gateway and configure an active/standby cluster. When the active gateway fails, the standby gateway automatically assumes the IP address of the failed gateway. The site-to-site VPN configuration is already present on the standby gateway through periodic synchronization between active and standby, and connections are automatically re-established.

Multiple factors need to be considered when enabling site-to-site VPNs, including the following important factors:

Authentication The Windows Server 2012 R2 multi-tenant gateway supports multiple methods of authenticating tenants connecting to it. The tenant identification on the multi-tenant gateway also depends on the authentication method used.

- **Pre-shared Key (PSK)** Both parties use a secret key known only to them to authenticate each other. In the case of PSK authentication, the IP address of the on-premises gateway is used to identify the tenant, so a unique on-premises IP is required for every connection using PSK.
- **X.501 certificate** Both parties authenticate each other's machine certificate. This requires appropriate root certificates to be present on the two gateways. The subject name of the on-premises gateway's certificate is used to identify the tenant.

- **Extensible Authentication Protocol (EAP)** All username and certificate-based methods that are available inbox are supported. Tenant identification depends on the EAP method that is being used. EAP requires that you deploy the Microsoft RADIUS server, otherwise known as the Network Policy Server (NPS).

Routing A site-to-site VPN tunnel between on-premises and cloud only supports layer 3 connectivity, and as a result, the on-premises and VM network workloads must be in different subnets. Traffic between the two networks is routed over the tunnel, for which appropriate routes need to be configured on the gateway at both ends. There are two ways to configure these routes:

- **Static routes** The hoster and enterprise administrators manually configure the subnets reachable over the site-to-site tunnel on their respective gateways. In the scenario shown in Figure 5-3, the Fabrikam administrator would configure subnet 192.168.1.0/24 on the multi-tenant gateway to be reachable over the site-to-site tunnel to Contoso and the Contoso administrator would configure subnet 10.10.1.0/24 on the other on-premises gateway to be reachable over the tunnel. The multi-tenant gateway allows a separate set of routes for each of the tenants. These routes can also overlap. The disadvantage of this approach is that every time a new subnet is created in a tenant's VM network or on-premises, it must be manually configured on the gateway.
- **Dynamic routing** The routes referenced above can be dynamically exchanged between the two gateways by configuring BGP routing protocol on them. In this case, for every tenant, the on-premises network and cloud virtual network are configured as separate, autonomous systems. The multi-tenant gateway acts as a multi-tenant router. It emulates multiple virtual routers, one for every tenant, where each virtual router peers with the corresponding on-premises routers to exchange routes.

Although multiple sites of an enterprise can connect to a virtual network in the cloud over separate site-to-site tunnels, all of these tunnels terminate on the same cloud gateway, so this configuration can also provide failover capabilities. In Figure 5-4, Woodgrove's New York (NY) and San Francisco (SFO) branches are both connected to Woodgrove's virtual network in the Fabrikam cloud. Woodgrove has also established a site-to-site tunnel between the New York and San Francisco sites to allow resources in those sites to communicate directly with each other. If the site-to-site tunnel between San Francisco and the virtual network hosted by Fabrikam fails for any reason, all traffic between San Francisco and the virtual network will be routed via the New York site.

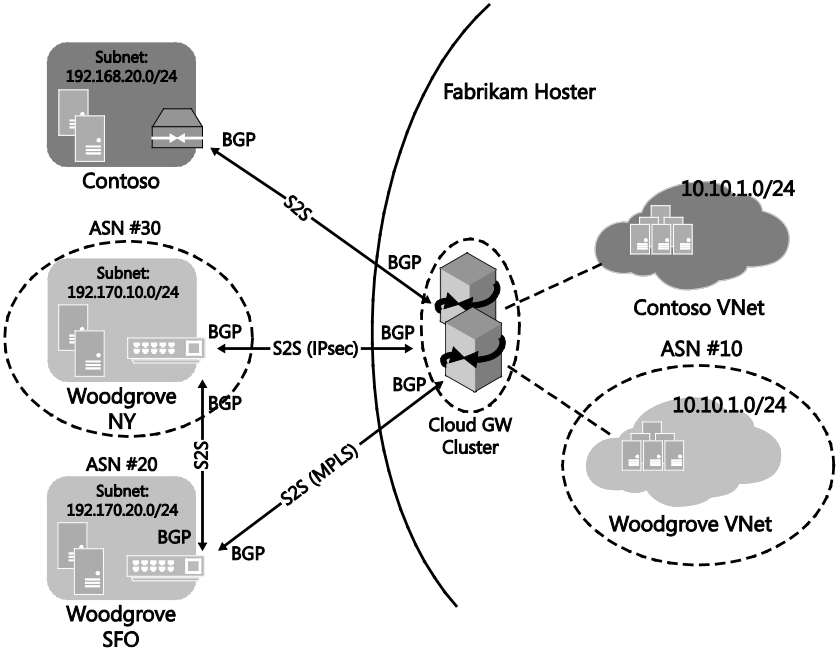


FIGURE 5-4 Connecting multiple sites of an enterprise to a virtual network

Virtual Machine Manager (VMM) can be used to deploy the Hyper-V Network Virtualization gateway and make it ready for tenants to use. Once this has been done, a self-service portal (like the Windows Azure Pack) provided to tenants can allow them to configure the gateway to suit their specific requirements.

High-speed dedicated tunnels Although Windows Server 2012 Gateway does not support Multiprotocol Label Switching (MPLS) directly, tenant MPLS networks can be integrated with a Network Virtualization gateway, as shown in Figure 5-5. In this example, the MPLS routers in the Contoso datacenter and Woodgrove Bank datacenters are connected to the MPLS router on the Fabrikam network.

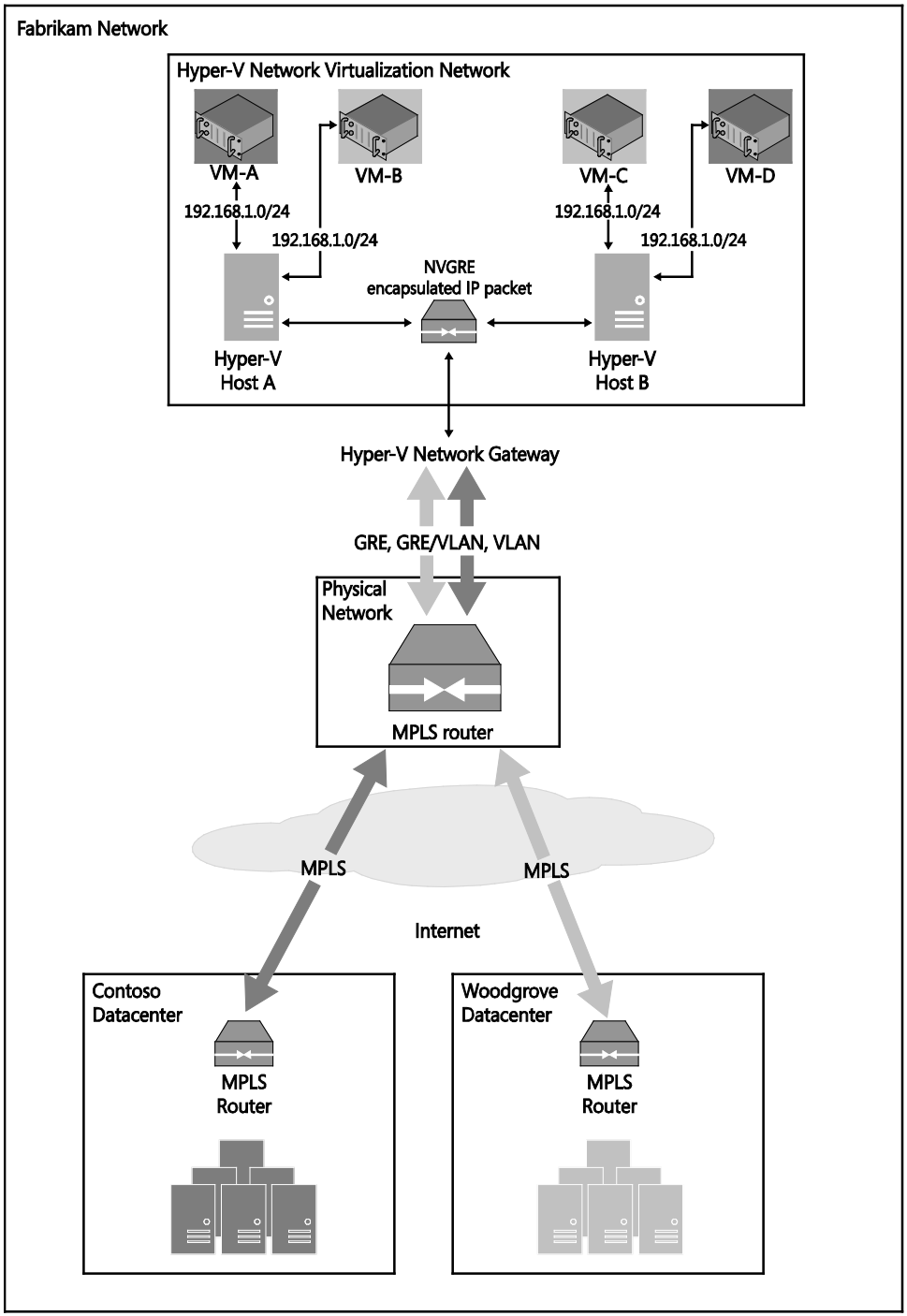


FIGURE 5-5 Integration with MPLS

In this specific example, both customers are using the same IP subnet (192.168.1.0/24), but whether tenants use overlapping addresses or not, Fabrikam must ensure isolation of traffic and make sure that packets are routed to and from the correct VM network. They can achieve this through either of the following two approaches outlined:

- **Establish GRE tunnels between an MPLS router and Network Virtualization gateway for each tunnel** Integration with MPLS networks can be accomplished by leveraging GRE tunneling support on the gateway. Refer to *https://technet.microsoft.com/en-in/library/dn765485.aspx* for more details on GRE tunnels in Windows Server 2012 R2. Multiple GRE tunnels can be established between MPLS routers and multi-tenant gateways, with each tunnel identified by a GRE key. Hence, between the MPLS router and the Network Virtualization gateway, multiple tenants' traffic with overlapping IP addresses can be isolated. The remaining mechanisms of integration with Network Virtualization networks are the same as in IPsec site-to-site VPNs as outlined at *http://blogs.technet.com/b/networking/archive/2013/09/29/multi-tenant-site-to-site-s2s-vpn-gateway-with-windows-server-2012-r2.aspx*.
- **Create VLANs for each tenant between an MPLS router and gateway** An alternative to GRE tunneling is deploying a forwarding gateway. A forwarding gateway assumes that separate VLANs per tenant are available in the physical network. While isolation in the virtual network is provided by Network Virtualization, isolation in the physical network must be provided by a VLAN. For each tenant, the VLAN must be configured on the gateway, the TOR switch, and any other routers in the physical network. This requires manual configuration of many switches and routers. Furthermore, since a separate forwarding gateway needs to be deployed for each tenant, a multi-tenant gateway cannot be used. Refer to *http://blogs.technet.com/b/networking/archive/2013/09/29/multi-tenant-site-to-site-s2s-vpn-gateway-with-windows-server-2012-r2.aspx* for more details on this scenario.

When enterprise employees access workloads in the cloud from outside the corporate network, they use VPN to access their enterprise VPN gateway and then access workloads via site-to-site VPN. If the enterprise VPN gateway is not available due to disaster or outage, enterprise employees can access their workloads by directly establishing a VPN connection with the cloud service provider gateway.

Businesses that do not have their own remote access (dial-in) VPN infrastructure can allow their employees to use VPN to access the cloud service provider gateway and access workloads in the cloud directly and access on-premises workloads via site-to-site VPN. Employees of businesses without infrastructure who have their entire workloads in the cloud must use VPN to access the cloud service provider and their internal workloads.

Remote access VPN is also required for disaster recovery for single-site enterprises. For example, Contoso replicates all of their business sensitive applications to the Fabrikam datacenter. In the event of a disaster, Fabrikam activates the application VMs of Contoso in the Contoso VM network. If Contoso had multiple premises, it would have been possible to access

Contoso VM networks over site-to-site VPN from one of its premises that is online. If Contoso's only premises is down, Contoso's VM network can be accessed over remote access VPN. You can find more details on this scenario at *http://blogs.technet.com/b/privatecloud/archive/2013/11/28/software-defined-networking-hybrid-cloud-using-hyper-v-network-virtualization.aspx*.

Two types of point-to-site VPN connections are supported:

- **SSTP** SSTP is the Microsoft SSL-based VPN tunnel solution. It is the preferred method to connect because it uses TCP port 443 and therefore can penetrate firewalls.
- **IKEv2** IPsec IKEv2 can also be used for VPN tunneling, but it requires specific ports to be opened in the firewall: ports 500 and 4500 for IKE traffic and IP protocol 50.

Multiple factors must be considered when enabling site-to-site VPNs, including authentication, IP address assignment, and client configuration.

Authentication The multi-tenant gateway supports all authentication methods that are included in the Windows Routing and Remote Access Service (RRAS) server and the built-in Windows VPN client, including username/password, certificates and token-based, with the latter provided through third-party EAP one-time password methods.

Credentials for authentication are assigned by the hoster Fabrikam using Active Directory and domain controller servers. Enterprises cannot bring their own credentials. Authentication can happen in one of two ways:

- **Using a RADIUS server** The gateway can be configured to use a local or remote RADIUS server for authentication. Microsoft NPS can be used. The advantage of using a RADIUS server is that tenant-specific network access policies can also be enforced.
- **Without using a RADIUS server** The gateway can be configured to directly authenticate a user with Active Directory.

Tenant identification is done using the username or the certificate subject name of the credential and can be configured to happen either on the gateway itself or on the Microsoft NPS RADIUS server. User/subject names are of the form <user name>@<organization name>. Fabrikam would have to use unique strings for one of these names for the purpose of tenant identification. For example, the following are some username options for a user Joe who belongs to Contoso:

- joe@contoso.com
- joecontoso@fabrikam.com
- joe@fabrikam.contoso.com

IP address assignment The IP address for the VPN clients is assigned through static IP pools. Both IPv4 and IPv6 are supported. A separate pool can be configured for every tenant on the multi-tenant gateway. These pools can overlap across tenants. Through BGP routing, the subnet defined for address assignment is also advertised as a reachable subnet from the gateway

Client configuration The built-in VPN client in Windows can be used to establish SSTP and IKEv2 connections to the cloud gateway. Following are two methods for setting up the client:

- **Network settings under Control Panel** Network settings allow a user to create a VPN profile in a few simple steps and uses smart defaults for some of the complex settings.
- **VPN client Windows PowerShell cmdlets** Windows PowerShell cmdlets provide more flexibility for specifying tunnel types, authentication protocol, encryption settings, and so on. An enterprise administrator can use these cmdlets to write a script to create a highly customized VPN profile.

Internet connectivity and publishing

This section covers scenarios where VMs in VM networks must access the Internet and must be accessed from the Internet. These scenarios are categorized as Internet connectivity and Internet publishing, respectively.

Internet connectivity of VM networks

Typically, tenant VMs in VM networks are assigned IP addresses from private IP address ranges. When applications running on these VMs must be made accessible from the Internet, the applications are published on public IP addresses. This process requires network address translation to be enabled on the gateway.

Consider Woodgrove Bank, which has a VM in a virtual network hosted by Fabrikam with the IP address 10.0.0.10. An application in this VM that binds to source port 5001 needs to access port 80 of an Internet server 65.10.10.100. Since IP address 10.0.0.10 is not routable in the Internet, when the packet exits the Fabrikam network, the IP address of the packet is translated to the public IP address 131.107.10.10. The mapping table on the gateway is as follows:

ORIGINAL PACKET						**TRANSLATED PACKET**				
Tenant	Source IP	Source Port	Destination IP	Destination Port	Protocol	Source IP	Source Port	Destination IP	Destination Port	Protocol
Woodgrove	10.0.0.10	5001	65.10.10.100	80	*	131.107.10.10	9001	65.10.10.100	80	*

When the return packet comes from 65.10.10.100 to the Fabrikam network, the following mapping table is used:

ORIGINAL PACKET					TRANSLATED PACKET					
Source IP	Source Port	Destination IP	Destination Port	Protocol	Source IP	Source Port	Destination IP	Destination Port	Protocol	Tenant
65.10.10.100	*	131.107.10.10	9001	*	65.10.10.100	*	10.0.0.10	5001	*	Woodgrove

As shown, for tenant VMs to access the Internet, NAT must be deployed at the edge of the VM network. Windows Server 2012 R2 supports multi-tenant NAT so that a single NAT VM can be used for multiple tenants with overlapping IP address space.

In a regular, single-tenant NAT, simply translating the destination IP address and port is good enough. However, with multi-tenancy, a number of tenants may use the same destination IP address (10.0.1.1 in our example). To ensure the packet is routed to the correct tenant, NAT maintains the mapping of the tenant's VSID with the packet.

Internet publishing from VM networks

Contoso has a mobile application. The first tier consists of web front ends while the second tier consists of backend application servers that service the mobile application. Consider the networking requirements of the Contoso deployment in the Fabrikam network. Contoso's mobile application should be reachable over the Internet. Fabrikam must design their infrastructure in a way that meets the requirements of Contoso and other customers with similar requests. Fabrikam creates a VM network for Contoso with the private IP address 10.1.1.0/24. Contoso deploys its front-end web server on VM 10.1.1.100. Fabrikam has Internet IP address space 131.107.10.10/28. Since Contoso's mobile application is https based, Fabrikam must reserve port 443 on one of its public IP addresses, for example 131.107.10.10, for Contoso. It maps 10.1.1.100:443 in the Contoso VM network to 131.107.10.10:443. Fabrikam does this by publishing the following NAT rule on the multi-tenant Network Virtualization gateway:

ORIGINAL PACKET					TRANSLATED PACKET					
Source IP	Source Port	Destination IP	Destination Port	Protocol	Source IP	Source Port	Destination IP	Destination Port	Protocol	Tenant
*	*	131.107.10.10	443	TCP	*	*	10.1.1.100	443	TCP	Contoso

When packets from the Contoso VM return on the Internet, the following NAT rule is applied:

ORIGINAL PACKET					TRANSLATED PACKET					
Source IP	Source Port	Destination IP	Destination Port	Protocol	Source IP	Source Port	Destination IP	Destination Port	Protocol	Tenant
*	*	*	*	*	131.107.10.10	*	*	*	TCP	Contoso

VMM can be used to deploy and configure NAT for the tenants. Tenants can also use a self-service portal (Windows Azure Pack) to configure NAT. An alternative is directly assigning public IP addresses to specific VMs. This requires multiple pools for public IPs for multiple tenants, wasting precious public IPs in the process of subnetting, and should generally be avoided.

Connectivity to shared services

The previous two sections covered connectivity solutions for scenarios where VMs in tenant VM networks connect to networks that are external to the hosting service provider. Service providers like the example organization Fabrikam often want to provide value-added services such as back-up, firewall, and management of VMs to their tenants. There are a number of different approaches to achieving this, but from a networking standpoint, the key question is whether tenants will potentially have overlapping IP addresses.

Tenant networks with non-overlapping IP addresses

Fabrikam has a shared backup and anti-virus service located on servers in a physical network that they want to provide to tenants. Assuming non-overlapping IP subnets, you would expect that tenants would have no issue routing packets to resources on the shared network. However, the multi-tenant Network Virtualization gateway is designed in such a way that packets in tenant compartments cannot be routed out directly. You can find more information on this constraint at *http://blogs.technet.com/b/networking/archive/2013/08/03/cloud-scale-multitenant-networking-stack-and-service.aspx*.

In short, packets in a tenant compartment can only be sent out to an external network via NAT, any tunnel (site-to-site or VPN), or a VLAN (single tenant forwarding gateway). The relative advantages and disadvantages of each of these methods for shared services are summarized as follows:

- **NAT** This requires publishing of ports for access from shared services to VM networks, so it is not ideal for all types of shared services and is generally not recommended.

- **Single tenant forwarding gateway** While this is technically a good solution, the fact that a separate gateway VM must be deployed for each tenant makes it a non-scalable option.
- **Site-to-site** Although creating a site-to-site connection for access within the same datacenter appears to be an unnecessary overhead, it is actually an elegant solution and represents the recommended way to deliver shared services to tenants with non-overlapping addresses.

Fabrikam plans to offer a number of shared services on IP subnet 172.22.1.0/24 to selected clients. One of the conditions for use of these services is that tenant IP subnets do not overlap with the IP address range allocated to the shared service. For Woodgrove Bank and Contoso, both tenants of Fabrikam, this constraint is not a factor: Woodgrove uses 192.168.1.0/24 and Contoso 192.168.2.0/24. Using the principles described in the section on establishing connectivity between enterprise networks and VM networks described previously, Fabrikam plans to allow tenants to "extend" their network into the VM network that hosts the shared services since this makes it appear as if those services are physically connected to the tenant's own network.

Fabrikam creates the shared services on the physical network (as shown in Figure 5-6) and then deploys a single tenant gateway to provide a connectivity "endpoint" for those services. Having successfully established this endpoint, Fabrikam then deploys a multi-tenant gateway for the tenants to use. When a tenant such as Woodgrove or Contoso want to use the shared services, the multi-tenant gateway connects to the endpoint offered by the single tenant gateway, and a secure tunnel is established between the tenant's network and the shared services network.

NOTE The reason two gateways are required to support this scenario is that the multi-tenant gateway can only route site-to-site packets to another gateway, so a single tenant gateway must be deployed to provide an endpoint in the hoster's VM network.

Although this approach to the problem of shared services works well, it does introduce a potential risk given that all tenants are essentially extending their networks through the same single tenant gateway. Since Fabrikam does not want to allow inter-tenant routing on that gateway, they also need to consider the use of filters. You can find more information on filters at *https://technet.microsoft.com/en-us/library/dn262682(v=wps.630).aspx*.

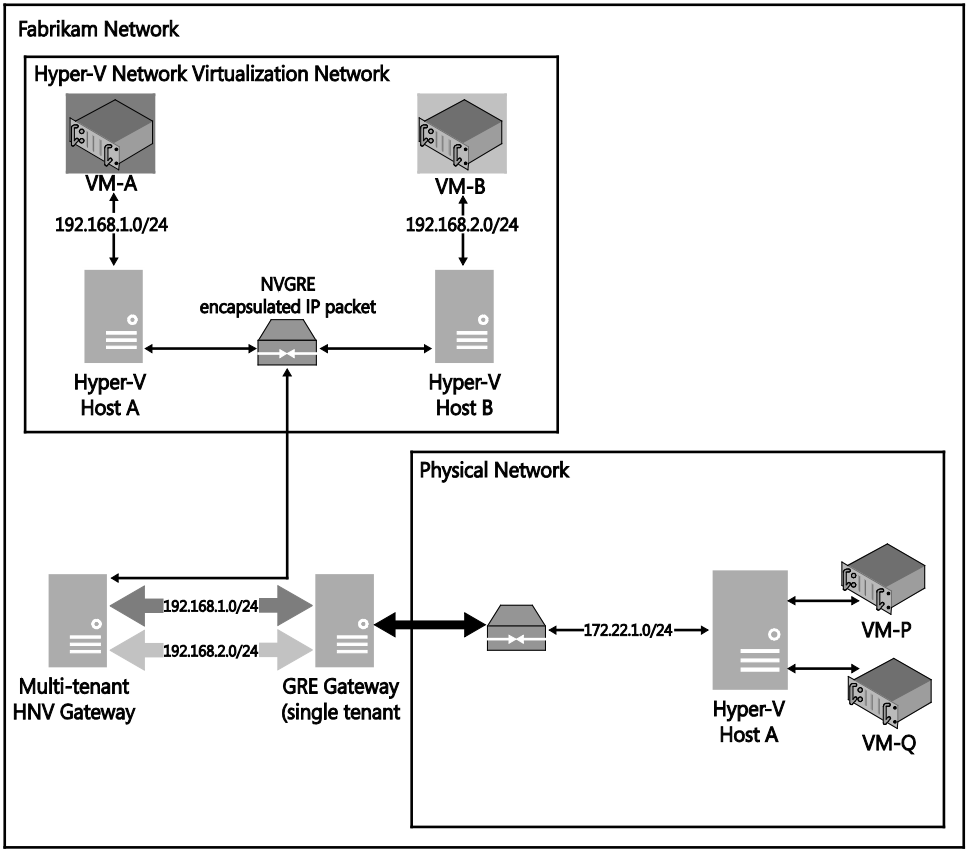


FIGURE 5-6 Integration with non-overlapping IP addresses

VM networks with overlapping IP addresses

In the previous scenario, only tenants with non-overlapping IP addresses can use the shared services offered by the hoster and although that approach works in principle, it might place limits on the hoster's ability to onboard or sell value-add services to some of its customers. In reality, it's good practice to think about the requirements and build out the necessary elements for a shared services solution that will work even if tenants do use IP address ranges that overlap with the address ranges allocated to the shared service, even if the current set of tenants do not.

Fabrikam wants to expand its solution offering and provide some additional shared services on IP subnet 192.168.1.0/24. The organization recognizes that the IP subnets used by some tenants (Woodgrove Bank in particular) overlap with this range but is unable to find a suitable range that does not overlap with any existing tenants. Fabrikam also recognizes that overlap at some point may become inevitable. As shown in Figure 5-7, the physical network in Fabrikam and the Woodgrove Bank VM network all use the same subnet, 192.168.1.0/24. To allow Woodgrove Bank to use these services, Fabrikam needs a different approach. As before, it

creates the shared services on the physical network (as shown in Figure 5-7) but this time uses a multi-tenant gateway configured for NAT to provide access. Whenever a tenant such as Woodgrove Bank wants to use the shared services, the multi-tenant gateway essentially translates the source and destination IP addresses and routes the packet to the required service as outlined below.

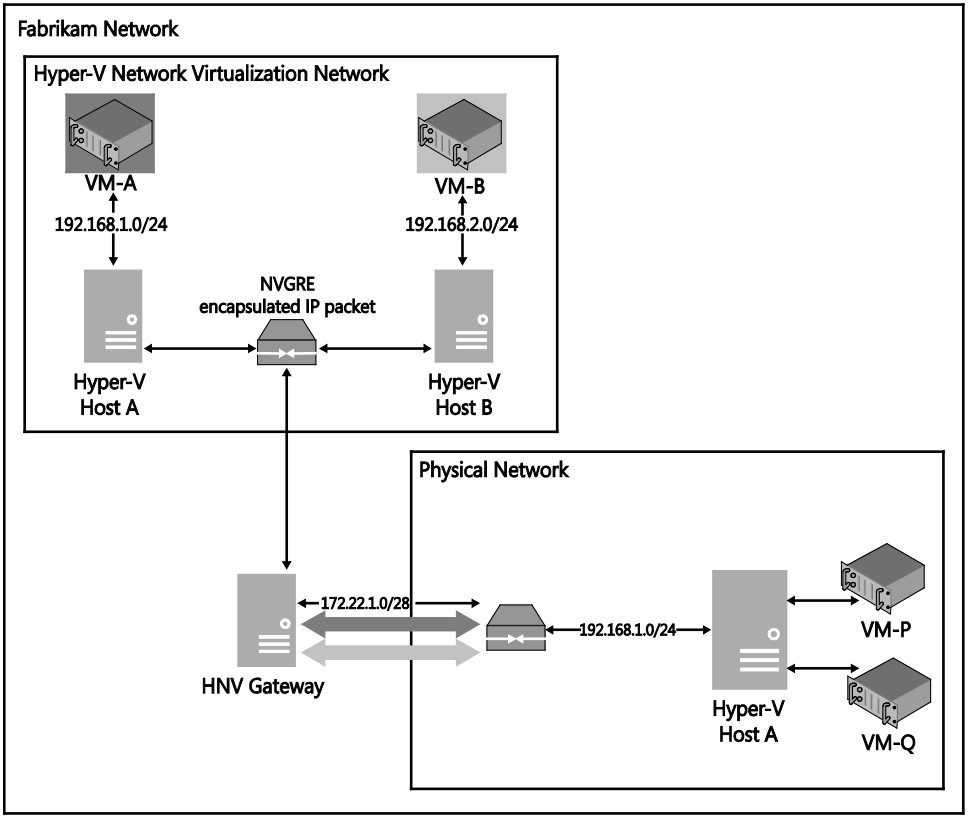


FIGURE 5-7 Integration with overlapping IP addresses

NAT of tenant packets In the Fabrikam example, the servers offering anti-virus as a service are in provider address space, but the IP addresses of those servers overlap with the tenant VMs. Furthermore, the anti-virus clients on the tenant VMs access the anti-virus servers using DNS names, so the tenant VMs must also be able to resolve the IP address of the service being offered by the hoster.

To avoid exposing anti-virus servers directly on the Internet, the hoster reserves a public DNS name, such as anti-virus.fabrikam.com, for the service. The hoster configures the public DNS to resolve anti-virus.fabrikam.com to a public IP, for example, 65.1.1.100. The hoster then deploys the anti-virus servers with a private IP address range in the infrastructure, for instance, 192.168.1.100. The hoster configures anti-virus clients on tenant VMs with server name anti-virus.fabrika.com. When the anti-virus clients try to resolve anti-virus.fabrikam.com, the query

is recursively resolved by the hoster's public facing DNS server to IP address 65.1.1.100. Since the DNS address is public, no DNS configuration changes are required on VMs in the VM network. When packets to 65.1.1.100 reach the Network Virtualization gateway, NAT mapping on the gateway translates the source and destination IP addresses and ports as follows:

NAT	VALUE IN PACKET	VALUE POST TRANSLATION
CONTOSO		
Source IP	*	192.168.1.254
Destination IP	65.1.1.100	192.168.1.100
Source port	*	65001
Destination port	*	* (same)
WOODGROVE		
Source IP	*	192.168.1.254
Destination IP	65.1.1.100	192.168.1.100
Source port	*	65002
Destination port	*	* (same)

Similar translation in the reverse direction ensures that the packets are returned to tenant VMs. The same solution is applicable for any service that the hoster offers. Although the service is resolvable to a public IP, the servers are not connected to the Internet and so there is no security issue.

Naming and addressing In the shared services with overlapping tenant VM networks example, naming and addressing of services plays a crucial role. In the previous example, Fabrikam exposed its services through a public name, anti-virus.fabrikam.com. This solves many problems related to DNS resolution. Assuming the tenant VMs cannot access the Internet to resolve public DNS name anti-virus.fabrikam.com, there are a number of different options.

Generally, VMs are assigned IP addresses through DHCP, with the DNS server address included in the information provided to the VM. For Contoso VMs to resolve anti-virus.fabrikam.com, the DNS server of Contoso should be able to resolve the DNS name. For this to happen, Contoso's DNS server should be modified to create a Zone Fabrikam.com populated with an A record for anti-virus. The question is, what should the IP address be in the A record. If it is a public IP address like 65.1.1.100, there is no problem (see *https://technet.microsoft.com/en-us/library/cc958825.aspx* for details on public and private IP addresses). What should you do if the hoster does not have enough public IP addresses for all of their servers or services? There are several options.

If the hoster chooses any of the three private IP ranges in 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16, there is a high chance that it will conflict with the IP address space of one or more tenants. Even subnets such as 10.254.254.0/24 are used by multiple enterprises for

different use cases. One alternative is to let each tenant configure its own non-conflicting IP address to anti-virus.fabrikam.com in the A record. For example, Contoso might configure 10.1.1.253 and Woodgrove might configure 172.16.0.253. The actual IP address of anti-virus.fabrikam.com can be 192.168.0.253. A third tenant assigns 192.168.0.253 to anti-virus.fabrikam.com. To ensure communication between multiple tenants' VMs and anti-virus.fabrikam.com, appropriate NAT mappings must be configured for each tenant. The following table shows NAT mappings for this example.

ORIGINAL PACKET					**TRANSLATED PACKET**					
Source IP	Source Port	Destination IP	Destination Port	Protocol	Source IP	Source Port	Destination IP	Destination Port	Protocol	Tenant
*	*	10.1.1.253	*	*	192.168.0.241	*	192.168.0.253	*	*	Contoso
*	*	172.16.0.253	*	*	192.168.0.242	*	192.168.0.253	*	*	Woodgrove
*	*	192.168.0.253	*	*	192.168.1.0.243	*	192.168.0.253	*	*	Third tenant

As shown in the preceding table, Fabrikam has assigned one IP address in its private IP address range for each tenant. Irrespective of what the destination IP is in the IP packet, NAT translates the destination to the IP address of anti-virus.fabrikam.com, or 192.168.0.253, and the source IP to the IP Fabrikam assigns to the respective tenant. So all that the hoster had to do is the following:

1. Allow each tenant to choose an IP address of their choice for anti-virus.fabrikam.com
2. Assign one IP address per tenant for anti-virus.fabrikam.com
3. Configure the NAT tables as shown
4. Configure routing in the shared service network so that packets with the IP address assigned in step 2 go to the appropriate gateway with the corresponding NAT mapping.

This mechanism elegantly solves naming and addressing problems of overlapping IP addresses for shared services. The following represent some of the other approaches a service provider could use to address this particular issue:

- To avoid changing enterprise DNS servers, they could use the Name Resolution Policy Table feature available in Windows Server clients. For more details, see *https://technet.microsoft.com/en-us/library/ee649207%28v=ws.10%29.aspx*.
- Use PVLANs as explained in Chapter 2, "Logical networks," instead of NAT.
- Use remote access VPN instead of NAT.

Connectivity to legacy networks

Network Virtualization is a new technology, and some hosters and enterprises already have VLAN-based tenant networks. For these enterprises or hosters to gradually migrate tenant applications from VLAN-based networks to Network Virtualization-based networks in a phased manner, connectivity between Network Virtualization VM networks and VLAN networks is required.

NOTE **A Network Virtualization gateway provides only layer 3 connectivity between Network Virtualization networks and VLAN networks, which means a subnet in a Network Virtualization network cannot overlap with a subnet in a VLAN network. For example, if a VM in a VLAN network has IP address 10.0.0.1 with subnet mask 255.255.255.0, VMs in the Network Virtualization network cannot have IPs in the same subnet 10.0.0.0/24. The IP subnet configured for the Network Virtualization-based VM network must be different, such as 10.0.1.0/24. This also means broadcast traffic in VLAN networks is not sent to Network Virtualization-based VM networks.**

Such connectivity enables a particular tenant with applications in one subnet based on Network Virtualization to communicate with another application in a different subnet based on VLAN. A Network Virtualization gateway provides interconnectivity between Network Virtualization VM networks and VLAN networks of the same tenant. There are two options for deploying a Network Virtualization gateway to support this scenario:

- **Dedicated forwarding gateway for each tenant** A Network Virtualization gateway can be deployed in forwarding gateway mode where a single tenant's Network Virtualization VM network traffic can be routed to that same tenant's subnet in a physical network. When deployed in this mode, a separate gateway must be configured for each tenant. The gateway maps VSID information in the NVGRE header to VLAN information in the Ethernet header.
- **Multi-tenant gateway with GRE tunnels** This mode of connectivity between physical networks and Network Virtualization networks operates via GRE tunnels on a Network Virtualization gateway. One endpoint of a GRE tunnel is configured on the multi-tenant gateway, and the other endpoint is configured on a third-party device on the physical network. Layer 3 traffic is routed between the VMs in the VM network and the third-party device on the physical network. If the GRE endpoint in the physical network allows virtual routing and forwarding, a physical network on the hosting provider network with multiple tenant network using VLAN-based isolation can be integrated with Network Virtualization networks. GRE tunnels, in addition to addressing the requirements specified, has the advantage that the intermediate network elements need not be modified. GRE tunnels established between Network Virtualization gateways and other devices provide isolation of overlapping networks without modifying the configuration on intermediate physical switches.

Returning to the example, Fabrikam is in the process of moving tenants away from traditional VLAN isolation methods to a Network Virtualization-based environment since it provides much more flexibility and control. Because not all of the tenants' VMs and services can be moved at once and, additionally, because Fabrikam wants to allow tenants to move when they are ready, Fabrikam first creates a new VM network for each tenant configured for Network Virtualization, deploys a GRE-capable device in the tenant's VLAN network, and, finally, sets up a multi-tenant gateway to connect this device, as shown in Figure 5-8. When this is done and the GRE tunnel between the networks is successfully established, VMs on either the Network Virtualization network or the physical network are able to communicate with each other successfully with no changes required to VM IP addresses or configuration.

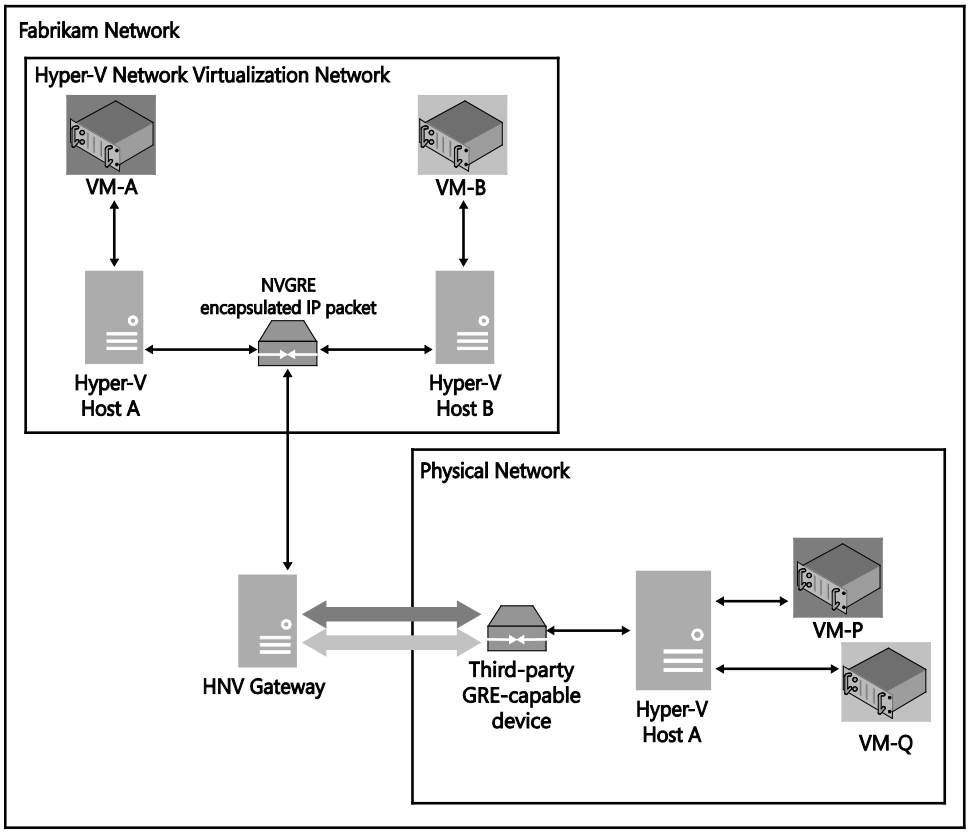


FIGURE 5-8 Connecting to a legacy network using a GRE tunnel

Deployment considerations

The following sections discuss some of the considerations for deploying the Hyper-V Network Virtualization gateway in your environment and what you need to do to enable the scenarios discussed previously. You can find more details at *https://technet.microsoft.com/en-us/library/dn606165.aspx*.

In conjunction with the bandwidth factors discuss below, the encryption and encapsulation calculations that need to be performed can often consume a significant proportion of CPU on the host computer. You should carefully consider the hardware considerations outlined in the architectural guide at *https://technet.microsoft.com/en-us/library/jj618319.aspx* to ensure that the Hyper-V hosts are able to support your specific requirements.

Given these factors, you should aim to set aside a dedicated set of hosts for running gateway VMs. Refer to *https://technet.microsoft.com/en-us/library/dn423897.aspx* for more details. Since gateway service is critical for multiple scenarios, a cluster of gateway VMs for High Availability is recommended.

Hardware requirements for each type of gateway

Although it is possible to run multiple gateway VMs in the same Hyper-V host, each type of gateway has specific requirements with respect to hardware specification and loading patterns, and it is important to be aware of these points as you plan your deployment.

IPsec site-to-site VPN

IPsec site-to-site VPN is compute intensive because it requires encryption and decryption of traffic. A six-core gateway VM can process up to 1.5 Gbps of IPsec traffic. If each of the tenants of a hoster has site-to-site bandwidth requirements of 50 Mbps in each direction, then a single gateway VM can cater to up to 15 tenants. If NAT or GRE functionality must be deployed for the same tenants, the number of tenants a gateway can support is reduced accordingly.

Each gateway with site-to-site VPN must have a separate public IP address. The public IP address on the external IP address of the gateway is failed over from active to standby VMs by the clustering service.

To enable dynamic routing using BGP over site-to-site VPNs, a BGP router is enabled on the VSID interface of the tenant on the gateway. For a BGP peer to be established over a site-to-site interface, a single IP address route (otherwise known as a /32 route) of the remote BGP peer must be added on the site-to-site interface, over which peering must be enabled. You can find more details at *http://blogs.technet.com/b/networking/archive/2013/10/11/border-gateway-protocol-bgp-with-windows-server-2012-r2.aspx*.

Deploying GRE tunnels

Since GRE tunnel processing does not involve encryption, a six-core gateway VM can process up to 2 Gbps of GRE traffic. Multiple GRE tunnels can be configured between a gateway VM and its peer router. Each tunnel is uniquely identified by a key configured on both ends.

Remote access VPNs

Remote access VPNs leverage RADIUS servers for authentication and identification of tenants. A six-core gateway VM can process up to 1 Gbps of traffic aggregated over 1,000 point-to-site VPN connections. You can find more details on this in Chapter 2 in "Network Virtualization and Cloud Computing," a free ebook available at *http://blogs.msdn.com/b/microsoft_press/archive/2014/03/24/free-ebook-microsoft-system-center-network-virtualization-and-cloud-computing.aspx#comments*.

Network Address Translation (NAT)

A six-core gateway VM can process NAT traffic up to 6 Gbps. One constraint to be aware of in a Windows Server gateway is that traffic over a site-to-site interface cannot be configured to use NAT; therefore, a separate approach or different gateway is required for this specific case.

Single tenant forwarding gateway

Similar to NAT, a six-core gateway VM can process up to 6 Gbps of forwarding traffic. It should be noted that when deploying using VMM, a separate forwarding gateway must be deployed for each tenant or VM network.

How many gateways do you need?

Understanding the requirements of the different types of gateway functions, you next need to think about how may gateways you actually need to support your requirements. The following represents some basic guidelines.

One forwarding gateway for each tenant

For each tenant for which the hoster must enable forwarding gateway functionality, one gateway must be deployed. When a gateway is deployed for a tenant, all gateway functions, such as NAT and site-to-site VPN, must be enabled on the same gateway.

Aggregate bandwidth of tenants

Since VM network traffic cannot be spread across multiple gateways, the aggregate bandwidth requirements across all functions becomes the next criteria for the number of gateways. For example, consider a hoster whose tenants each require the following bandwidths:

- 150 Mbps of IPsec traffic
- 200 Mbps of GRE traffic
- 300 Mbps of NAT traffic

In this case, one gateway VM should be deployed for every four such tenants. Note that a six-core gateway VM can support 6 Gbps of NAT/forwarding traffic, 1.5 Gbps of IPsec site-to-site traffic, or 2 Gbps of GRE traffic. So to calculate the aggregate bandwidth that a gateway can support, simply determine how many gateways are required by first considering NAT/forwarding traffic and applying a multiplication factor of 4 for IPsec site-to-site and 3 for GRE site-to-site traffic. In the previous example, for instance, the following is true:

- 150 Mbps of IPsec site-to-site traffic is equivalent to 600 (150*4) Mbps of NAT traffic.
- 200 Mbps of GRE traffic is equivalent to 600 (200*3) Mbps of NAT traffic.

So, each tenant requires 600 + 600 + 300 = 1,500 Mbps of NAT traffic. On a six-core VM, 6,000 Mbps of NAT or equivalent traffic can be sent and received.

CHAPTER 6

Deployment

As explained in Chapter 4, "Logical switches," logical switches allow virtual machines (VMs) to communicate out through the physical network adapters and network teams configured on the Hyper-V host. Logical switches are also used to enforce standard configurations on the Hyper-V host to prevent workloads from experiencing downtime or other issues due to misconfigurations.

This chapter explains how logical switches can be deployed to a new Hyper-V host and how an existing standard switch can be migrated to use the new converged approach, as well as some of the known issues related to logical switch deployment. Therefore, this chapter explains the different methods for applying a logical switch to a Hyper-V host and how existing Hyper-V hosts with standard switches can be migrated. It also covers best practices from the real world and early adopter customers. In addition, the text addresses the common deployment scenarios and highlights known issues and workarounds regarding logical switches in System Center Virtual Machine Manager (VMM).

This chapter will:

- Review the requirements for logical switches
- Discuss the different options for deploying a logical switch to a Hyper-V host
- Explain how to migrate a standard switch to a logical switch
- List some known issues when deploying logical switches

Preparing for deployment

With System Center 2012 R2 Virtual Machine Manager (VMM), it is now possible to consistently configure identical capabilities for network adapters across multiple Hyper-V hosts by using logical switches. As explained in the previous chapters, a logical switch consists of port profiles, uplink profiles, and classifications and acts as a container for all of these properties, settings, and capabilities required by the underlying physical network adapters. You can almost think of it as a template including different building blocks for the Hyper-V host network adapters, which includes all the characteristics needed. When you deploy a logical switch, it matches this "template" instead of requiring configuration of individual properties or capabilities for each network adapter every time. This can simplify the configuration process dramatically.

Logical switch creation, however, involves a rather complex configuration wizard, especially the first time you go through this process. One reason it is complex is because a logical switch is made up of building blocks that you must define in advance. Before you deploy a logical switch, you need to prepare the following configurations and profiles in VMM:

- The following (or similar) logical networks should preferably be configured and represented in VMM (for more see Chapter 2, "Logical networks"):
  - Management (used by Hyper-V hosts)
  - Back End (used by Failover Cluster for Cluster Shared Volumes (CSV) and Live Migration)
  - Storage (used by iSCSI or SMB 3.0, if available)
  - Front End (used by VMs and tenants)
- VM networks (also discussed in Chapter 2)
- One or more uplink port profiles that defines how the NIC team should be configured (for more, see Chapter 3, "Hyper-V port profiles")
- At least the default virtual network adapter port profiles that will be used to control the characteristics of that virtual network adapter
- Port classifications to provide the global names for identifying different types of virtual network adapter port profiles
- (Optional) Additional Hyper-V switch extensions such as a Network Driver Interface Specification (NDIS) filter or Windows Filtering Platform (WFP) filter that runs inside the Hyper-V extensible switch

At this stage, all of the fundamental building blocks required for the creation of the logical switch are in place. As mentioned earlier, the logical switch brings port profiles, port classifications, and switch extensions together so that you can apply them consistently to network adapters on all Hyper-V hosts (for more see Chapter 4, "Logical switches")

NOTE **The reasons to have a logical network for "Storage" are debatable. If hardware offloads will be available, such as RDMA or iSCSI HBA, or a fully utilized 10-Gbps NIC, virtual switches cannot be used for these workloads. Thus, such NICs cannot be managed by VMM altogether. However, to aid troubleshooting and maintenance, those logical networks should also be created in VMM.**

Regardless of any port profiles and logical switches you plan to use in the configuration, each network adapter in a host can be allocated for use by VMs, for host management, for neither of these options, or for both of them. It is also important to review the prerequisites if you want to configure single-root I/O virtualization (SR-IOV) for network adapters on the host.

If you will not be using the bare-metal deployment capabilities of VMM to deploy your Hyper-V hosts, you will have to manually add all of your hosts to VMM. To perform this task, you must either be a member of the administrator user role or a delegated administrator. The steps for doing this are described on TechNet at *http://technet.microsoft.com/en-us/library/gg610605.aspx*. You can also use the Add-SCVMHost Windows PowerShell cmdlet:

```
$RunAsAccount = Get-SCRunAsAccount -Name My Administrator
Add-SCVMHost MyHyperVHost -RemoteConnectEnabled $True -RemoteConnectPort 5900
   -VMHostGroup MyHosts -Credential $RunAsAccount
```

TIP When you add a host to the VMM management server, by default VMM automatically creates logical networks for those host physical network adapters (pNICs) that do not have logical networks defined on them. You might therefore want to consider clearing this option as described in Chapter 2.

Deploying logical switches

A standard virtual switch can always be configured in the native Hyper-V management tools or by using Windows PowerShell cmdlets. But because they have been created outside of VMM, those virtual switches will show as "standard switches," which means, from a management perspective, those virtual switches do not offer any of the above mentioned advanced capabilities (for more see Chapter 4).

To ensure standardization and consistency across all managed Hyper-V hosts, the new concept of logical switches is much more powerful, mainly because it not only configures a virtual switch, it includes additional properties and capabilities within the different profiles. Based on this information and on building blocks, VMM creates the virtual switch on the Hyper-V host and uses the logical network(s), VLAN, and IP subnets from the uplink port profile to configure these properties on the selected network adapter(s).

For the deployment of a logical switch, you must consider the issues covered as part of this section where the deployment of a converged fabric logical switch is described. Before you proceed with the configuration, you should check if your Hyper-V host is using an a) untagged host management adapter or b) a tagged host management adapter. Tagged or untagged indicates how the management network adapter used by the management operating system partition (previously called the parent partition) is configured to access the network.

Tagged or untagged indicates the configuration of the VLAN ID for a particular network adapter. In a *tagged* deployment, every data packet that is sent from or to the management adapter is tagged with the configured VLAN ID as a numeric value, for example 120. This means that the virtual switch configured for the pNIC tags every package sent from and to the host management with a particular VLAN, in this case with the ID 120.

An *untagged* deployment doesn't require this VLAN ID. All packages sent from and to the host management are always untagged. However, this still means the network can be part of a specific VLAN. Therefore, the network port on the physical network switch (outside of Hyper-V) is configured for native VLAN on the trunk port. All untagged traffic is then treated automatically as the management VLAN, which consequentially doesn't require VLAN tagging on the management (virtual) network adapter. As mentioned previously, to deploy a logical switch, you must be a member of the administrator or delegated administrator user roles in VMM. In addition, when configuring virtual switches, delegated administrators can select only uplink port profiles that contain network sites that are in the administrative scope of their delegated privileges.

IMPORTANT When VMM creates the virtual switch, the host may temporarily lose network connectivity. This may have an adverse effect on other network operations in progress. As a result, the warning shown in Figure 6-1 appears and you must first acknowledge this warning before you can continue with the deployment process.

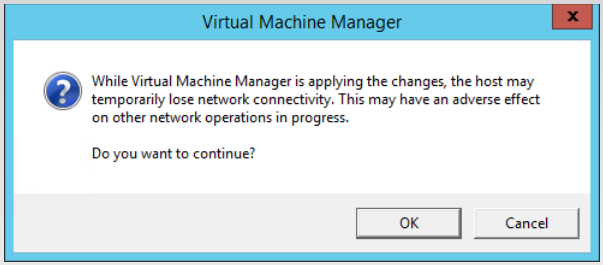


FIGURE 6-1 Dialog box warning that the host may temporarily lose network connectivity

Due to the temporary loss of network connectivity, there are some additional considerations when you are deploying logical switches onto a pNIC that will be used for host management, especially when management traffic should be carried only on a specific VLAN.

NOTE These network adapters require special attention since they are used by the host operating system (or parent partition) to access the network. The VMM Agent also uses these adapters to communicate with the VMM server, and to make this work, you need to define a logical network and a VM network for host management (as discussed in Chapter 2).

In a tagged deployment, every data packet related to host management that is sent from the network adapter must be tagged with a specific VLAN ID. As a result, a logical switch deployed on a pNIC used for management needs to be configured to add the appropriate VLAN ID to every packet that is sent from and to the host management logical network.

If the network port on the *physical* network switch has been configured for native VLAN on the trunk port, all untagged traffic is treated as destined for the host management VLAN, and as a result, all packets sent from and to the management logical network is unchanged by the logical switch.

Untagged host management network adapter

In an untagged scenario, management traffic is not tagged with a VLAN ID, as previously mentioned. This means two things from the perspective of physical network configuration:

- Management traffic generated by the host management adapter should not be tagged with a VLAN ID.
- The management network VLAN is actually set to be the native VLAN on the trunk port of the connected physical network switch.

NOTE The port on the physical switch can even be in access mode instead of trunk mode. In such a case, you would have only one management partition vNIC on the switch. This is typical across conservative designs, where a dedicated physical network just for host management is required.

To support untagged management traffic in VMM, define a logical network and set the VLAN ID for network site(s) within that network to 0 (as shown in Figure 6-2). VMM interprets this setting as meaning "no VLAN," and as a result, the logical switch will be configured to leave outbound network traffic unchanged.

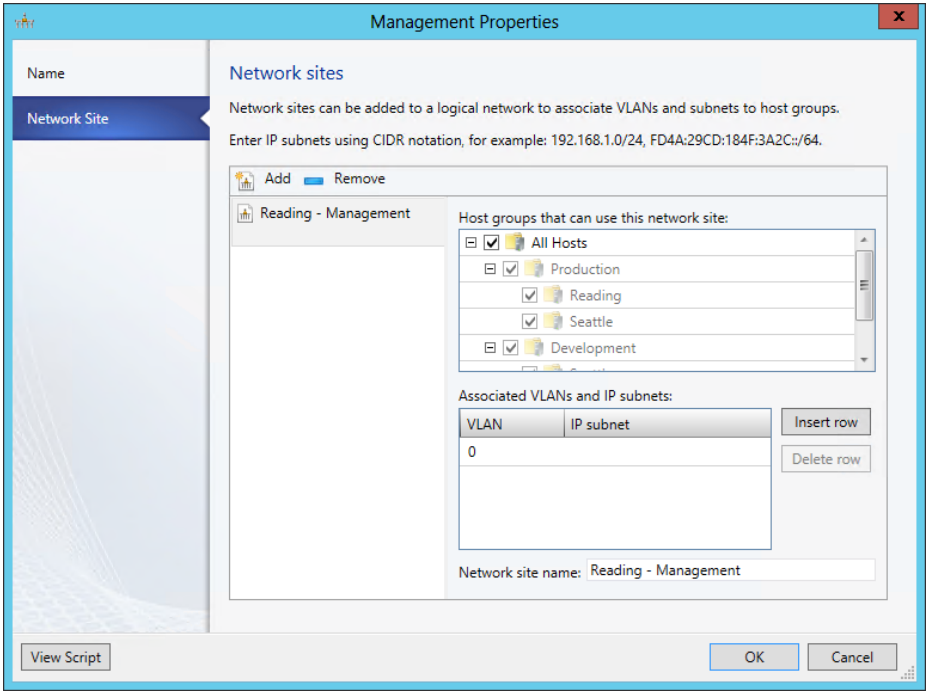


FIGURE 6-2 Example of an untagged host management logical network

The workflow for deploying a logical switch in a tagged scenario is as follows:

1. Open the VMM administrator console and switch to the Fabric workspace.
2. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, select the sub-host group where the host resides.
3. Select the target Hyper-V host and open the Properties dialog box.
4. In the Properties dialog box, select Virtual Switches.
5. Select New Virtual Switch and select New Logical Switch.
6. Select the corresponding logical switch from the drop-down list.
7. Under Physical Adapters, add the one (or more) network adapters that should be used for this logical switch. Note: It is possible to create a network team with a single network adapter only, and the network team then could be extended with another pNIC at a later time. But high availability deployments require at least two network adapters.
8. In the same view, next to the physical adapter, select the corresponding uplink port profile.
9. Select the logical switch and select New Virtual Network Adapter. This adds a virtual network adapter as part of the logical switch.

10. Select the newly added virtual network adapter and provide a meaningful name for the adapter.
11. For the management virtual network adapter, configure the following options as shown in Figure 6-3:
    - Name to be used for the virtual network adapter. If "Management" is used, then the vNIC would be named "vEthernet (Management)" on the host.
    - If the IP address from the physical network adapter should be reused, select This Virtual Network Adapter Inherits Settings From The Physical Management Adapter. Note that this option is available only for the first vNIC connected to the switch; this would typically be the Management vNIC, but this is not always the case.
    - Under Connectivity, select the corresponding VM network, for example Management. There should be no option to select a VLAN.
    - Under Port Profile, select the classification that matches the network, in this case Host Management.
    - Since the IP address will be reused from the pNIC, no additional settings are required.
12. For additional virtual network adapters, such as the one used by Live Migration, configure the following options:
    - After the name to be used for the virtual network adapter has been specified under Connectivity, select the corresponding VM network, for example Live Migration, and choose the appropriate VLAN if required.
    - For the IP address configuration, choose whether DHCP or Static will be used to configure the Live Migration adapter. When choosing Static, select IP Pool and specify the IPv4 address. If you don't specify an address, VMM will pick one automatically from the pool.
    - Under Port Profile, select the classification that matches the network, in this case Live Migration.
13. Repeat the previous step for all virtual network adapters required for this configuration.
14. After the configuration has been completed, click OK to close the Properties page. This initiates the logical switch creation on the Hyper-V host.
15. This job might take a while to finish. If you're connected to the Hyper-V host using RDP, you will most likely lose your connection.

16. When the job has finished, log on to the Hyper-V host and verify the configuration. If the IP address has been transferred from the physical to the virtual network adapter, make sure that the gateway and DNS Server settings were as well.

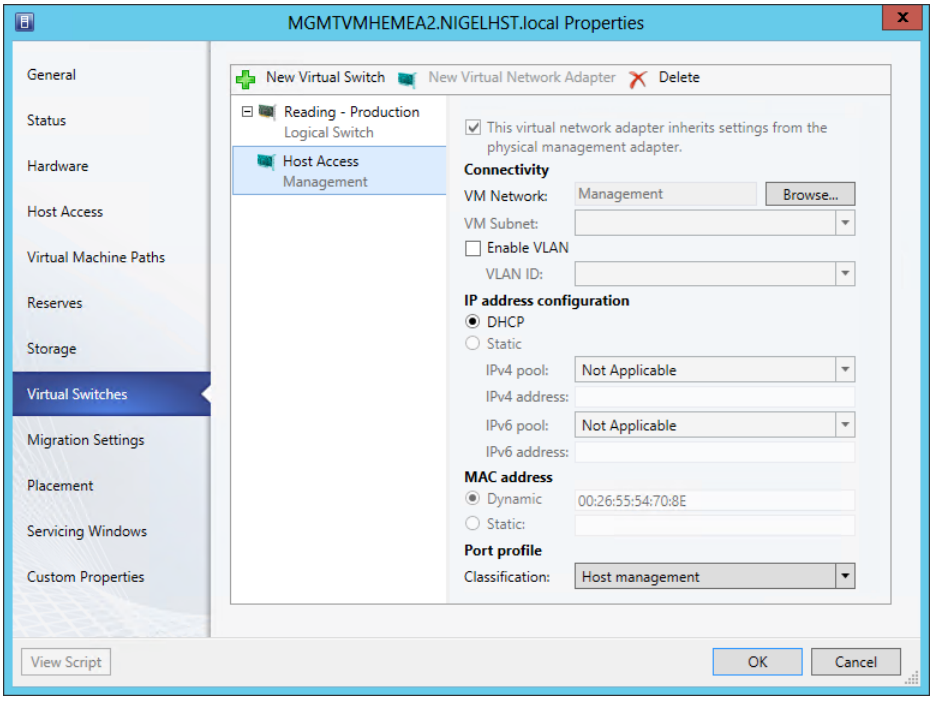


FIGURE 6-3 Logical switch deployment using untagged host management

NOTE **It is a good practice to add just one pNIC to the logical switch to make sure the right MAC address and configuration is used for the network team and management virtual network adapter.**

After the logical switch has been created and the configuration has been verified on the Hyper-V host, the host is ready for providing networking to VM workloads.

Tagged host management network adapter

In a tagged scenario, the physical network switch port is configured in *trunk mode*, and host management traffic is on a particular VLAN. To support tagged management traffic in VMM, define a logical network for management and set the VLAN ID for network sites within that network to the appropriate value (for example, 110 in Figure 6-4).

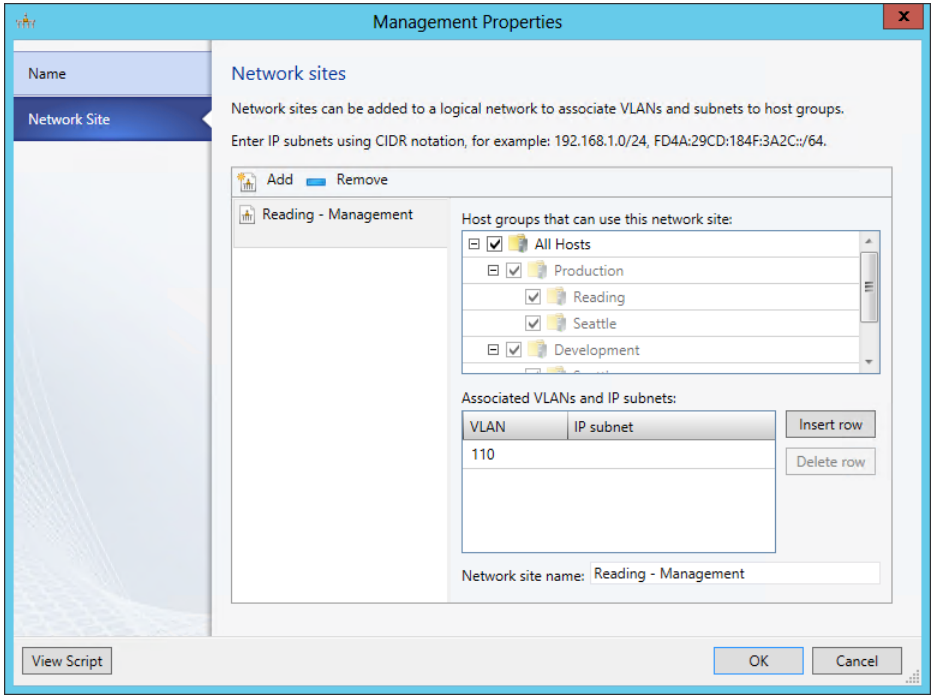


FIGURE 6-4 Tagged host management logical network

It's important to understand the subtle differences between the tagged VLAN scenario and the untagged scenario. In the tagged scenario, VMM has to tag the virtual network adapter used for host management with the particular VLAN ID specified in order for host management traffic to flow through the management virtual network adapter. As the VLAN configuration happens at the end of the virtual switch configuration, it is important that VMM has uninterrupted access to the Hyper-V host. This means that VMM requires connectivity through another management interface until it can complete the VLAN configuration of the new management virtual network adapter.

Complete the following steps to deploy a logical switch in a tagged scenario:

1. Deploy a logical switch with one pNIC as an uplink to reserve the other pNIC for management connectivity.
2. When the logical switch creation succeeds, add the other pNIC to the logical switch.

   Make sure that at least two pNICs are reserved to be used by management by completing the following steps.
3. In the VMM administrator console, switch to the Fabric workspace.
4. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, expand the sub-host group where the host resides.
5. Select the Hyper-V host that should be configured and open its Properties dialog box.

6. In the Properties dialog box, select Hardware.
7. Navigate to Network Adapters and select the pNIC that will be used for host management. Ensure that Used By Management is selected.
8. Repeat these steps for any other pNICs required for this configuration.

   You can reserve a single pNIC to be used for management and then start logical switch deployment from the other pNIC, regardless of whether it is reserved for use for management. If you have two pNICs active with valid IP addresses and expect both of them to be able to connect to VMM, both of must be registered with DNS.

   Next, proceed with the deployment of the logical switch to the Hyper-V host of choice. Complete the following steps to deploy the first virtual switch using the logical switch:
9. In VMM, click on the Fabric workspace.
10. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, expand the sub-host group where the host resides.
11. Select the target Hyper-V host and open its Properties dialog box.
12. In the Properties dialog box, select Virtual Switches.
13. Select New Virtual Switch and then select New Logical Switch.
14. Select the appropriate logical switch from the drop-down list.
15. Under Physical Adapters, add the first, and only the first, network adapter that should be used for this logical switch.
16. In the same view, next to the physical adapter, select the corresponding uplink port profile.
17. Select the logical switch and select New Virtual Network Adapter. This adds a virtual network adapter as part of the logical switch. Select the newly added virtual network adapter and provide a meaningful name to be used for the adapter. If "Management" is used, then the vNIC is named "vEthernet (Management)" on the host.
18. For the management virtual network adapter, configure the following options, as shown in Figure 6-5:
    - If the pNIC has a valid IP address, best practice is to select This Virtual Network Adapter Inherits Settings From The Physical Management Adapter. If the pNIC does not have a valid IP address, do not select this option.
    - Under Connectivity, select the corresponding VM network, for example Management, and select the appropriate VLAN (if required) as shown in Figure 6-5.
    - Under Port Profile, select the classification that matches the network, in this case Host Management.
    - If the pNIC does not have a valid IP address, select the IP pool. Otherwise, no additional settings are required.

19. For any additional virtual network adapters that you need to create on the selected logical switch, configure the following options:
    - After specifying the name for the virtual network adapter, select the corresponding VM network, for example Live Migration, and select the appropriate VLAN, if required.
    - For the IP address configuration, select DHCP or Static to configure the Live Migration adapter. When choosing Static, select IP Pool and specify the IPv4 address. If you don't specify an address, VMM picks one automatically from the pool.
    - Under Port Profile, select the classification that matches the network, in this case Live Migration.
20. After the configuration is completed, click OK to close the Properties dialog box. This initiates the logical switch creation on the Hyper-V host.
21. This job might take a while to complete. If you're connected to the Hyper-V host using RDP, you will most likely lose the connection.
22. When the job is finished, log on to the Hyper-V host and verify the configuration. If the IP address has been transferred from the physical to the virtual network adapter, make sure the gateway and DNS server were as well.

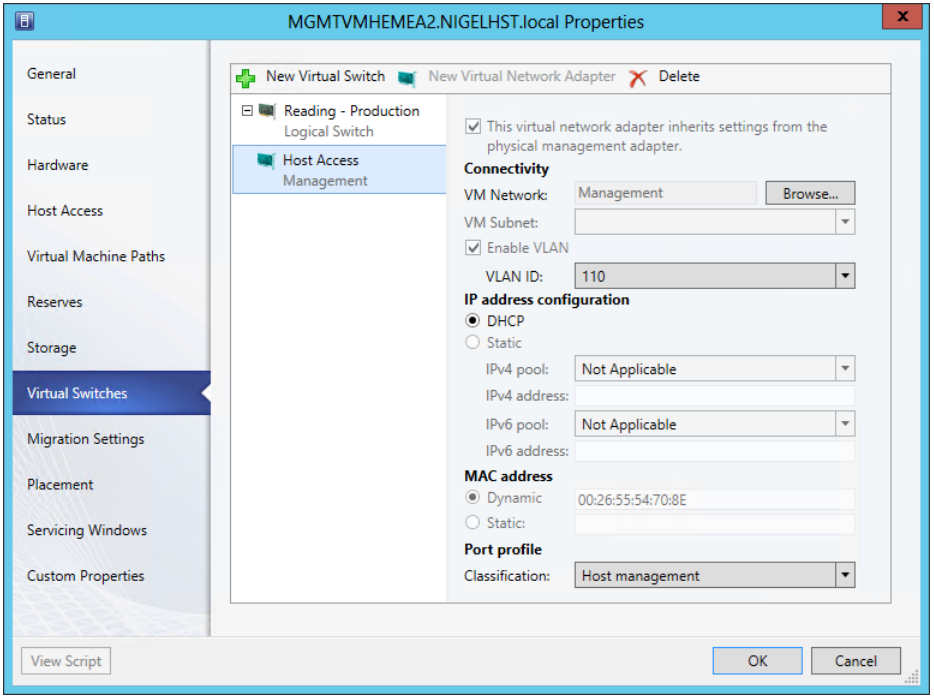


FIGURE 6-5 Logical switch deployment using tagged host management

When the logical switch has been created and the configuration has been verified on the Hyper-V host, the second pNIC can be added to the logical switch:

1. In VMM, select the previously configured Hyper-V host and open its Properties dialog box.
2. In the Properties dialog box, select Virtual Switches.
3. Select New Virtual Switch and select the logical switch that was previously deployed.
4. Under Physical Adapters, add the second pNIC that should be used for this logical switch.
5. In the same view, next to the physical adapter, select the corresponding uplink port profile.
6. After the configuration is completed, click OK to close the Properties dialog box. This initiates the logical switch creation on the Hyper-V host.

Finally, this Hyper-V host is ready for providing networking to VM workloads.

For automation and standardization, you might want to copy and customize the Windows PowerShell script that VMM can display at the end of the wizard before you click OK. For example, the following is a simple script to deploy without virtual network adapters:

```
$VMMServerName = MyVMMServer.fqdn
$HyperVName = MyHyperVHost.fqdn
$adapterName = MyEthernetAdapterName
$NativeUplinkPortProfileSetName = MyUplinkAdapterName
$LogicalSwitchName = MyLogicalSwitchName

$VMM = Get-SCVMMServer -ComputerName $VMMServerName
$vmHost = Get-SCVMHost -ComputerName $HyperVName -VMMServer $VMM
$networkAdapter = Get-SCVMHostNetworkAdapter -VMHost $vmHost | Where-Object
   -FilterScript { $PSItem.ConnectionName -eq $adapterName }
$uplinkPortProfileSet = Get-SCUplinkPortProfileSet -Name $NativeUplinkPortProfileSetName
   -VMMServer $VMM
$logicalSwitch = Get-SCLogicalSwitch -Name $LogicalSwitchName -VMMServer $VMM

Set-SCVMHostNetworkAdapter -VMHostNetworkAdapter $networkAdapter
   -UplinkPortProfileSet $uplinkPortProfileSet
New-SCVirtualNetwork -VMHost $vmHost -VMHostNetworkAdapters $networkAdapter
   -LogicalSwitch $logicalSwitch
```

Bare-metal deployment

Another way to deploy logical switches is by making use of the bare-metal deployment capabilities of VMM. VMM provides the capability to discover physical computers on the network, automatically install the Windows Server operating system on those computers, and convert them into managed Hyper-V hosts. This means the targeted physical computer can be a computer that does not have an operating system installed, often referred to as a bare-metal computer, or it can be a computer on which you want to overwrite an existing operating system. This chapter doesn't go into details about how to perform bare-metal deployment but instead highlights how to configure logical switches as part of bare-metal deployment.

The configuration required for bare-metal deployment is done in the physical computer profile located in the VMM Library. In VMM 2012 R2, this profile contains not only the Hyper-V configuration but also the entire physical and virtual network adapter configuration (see Figure 6-6).

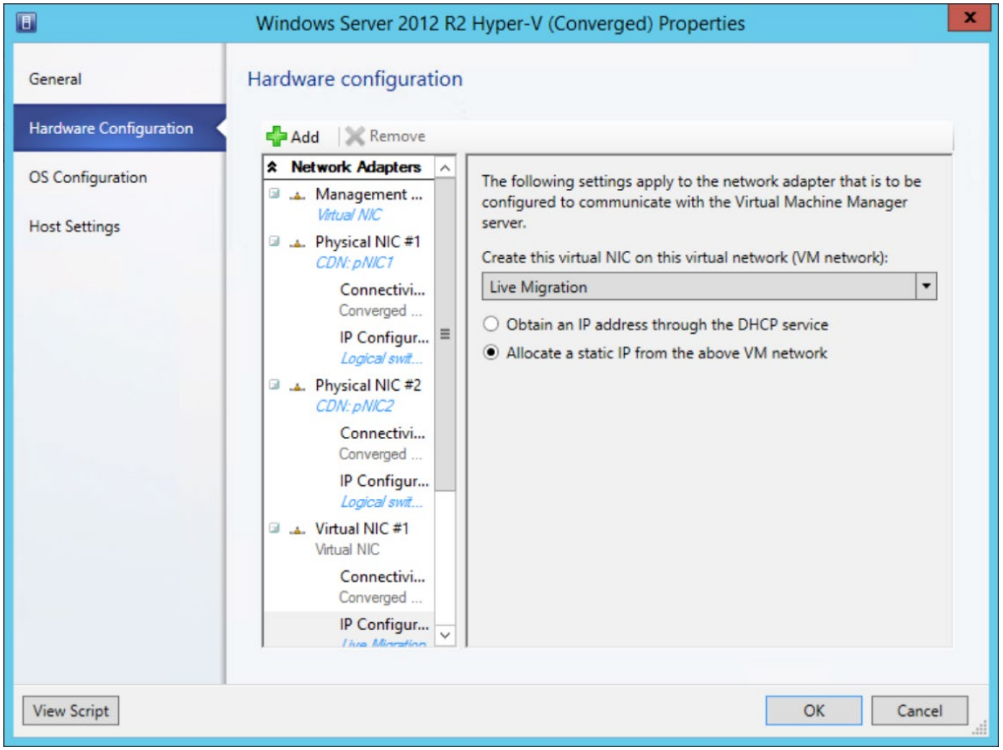


FIGURE 6-6 Hyper-V physical computer profile using a logical switch

If a virtual network adapter is used for management, the Create A Virtual Network Adapter As The Management NIC setting must be selected. Also, the IP configuration must specify whether DHCP or fixed IP addresses will be used. The option to use fixed IP addresses will be linked to the VM network, which will acquire an IP address from the available IP pool of this logical network.

When initiating a bare-metal deployment, the targeted Hyper-V host is restarted. This is initiated by an out-of-band management action. After the restart, the host boots into WinPE mode where a discovery of the host hardware is performed. This discovery is key to getting insights into how to apply the profile to the pNIC (see Figure 6-7). It's always good to have the MAC addresses available to easily identify the primary adapters.

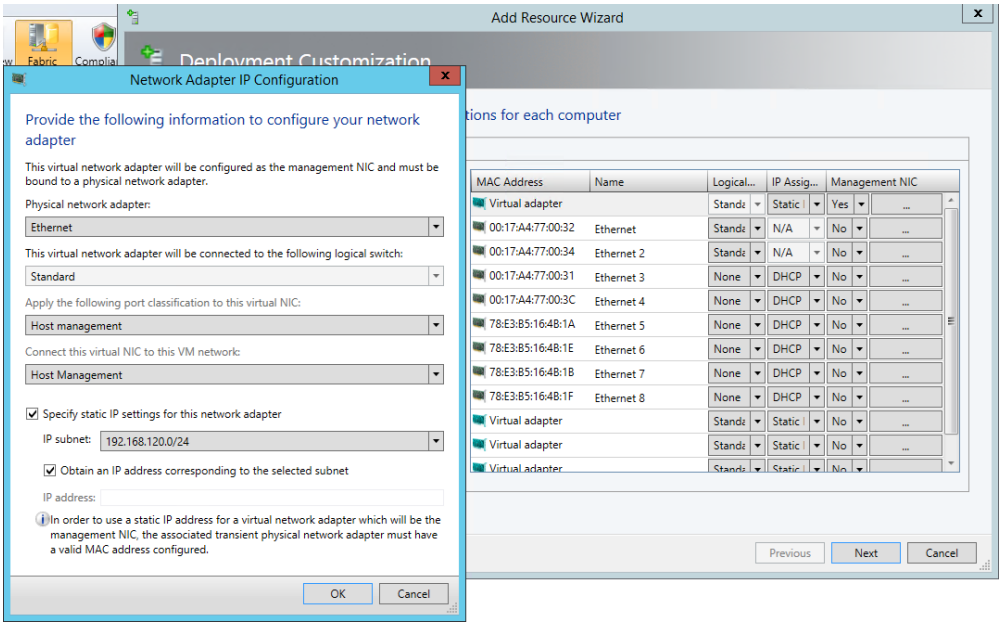


FIGURE 6-7 Bare-metal deployment network configuration

In summary, bare-metal deployment dramatically reduces the time required to install a Hyper-V host and simplifies the deployment of logical switches since this is already part of the installation and configuration process.

Update drivers and firmware on Hyper-V hosts

It's highly recommended that you check the firmware and drivers for the key hardware of a newly deployed Hyper-V host and update them if necessary. If existing hardware is repurposed, this step is even more important. This step ensures high performance and reliability of the configuration. The main focus should be on BIOS and BMC, but special attention should also be paid to the firmware and drivers for the physical network adapters—both the 1-Gbps and 10-Gbps adapters—for every Hyper-V host.

Refer to the hardware vendor's guidance for obtaining and applying any firmware updates. Don't put these hosts into production before the latest updates have applied and tested for an accepted period of time. For updates related to network adapters, make sure to include test cases with multiple VMs requiring high bandwidth over a few hours as well.

Migrating from a standard switch to a logical switch

Of course, not every environment is a "greenfield" in which you can build and deploy logical switches on brand new Hyper-V hosts. In established environments, you may find that standard Hyper-V switches have been deployed onto network adapters in a number of Hyper-V hosts. Although VMM will recognize and detect the presence of a standard switch (as shown in Figure 6-8), it provides the administrator with limited management capability. Unfortunately, once the pNIC has been associated with a standard switch, you cannot subsequently upgrade it to a logical switch. You must first *disconnect and remove* the standard switch and any associated vNIC from the network adapter before you begin to deploy the logical switch.

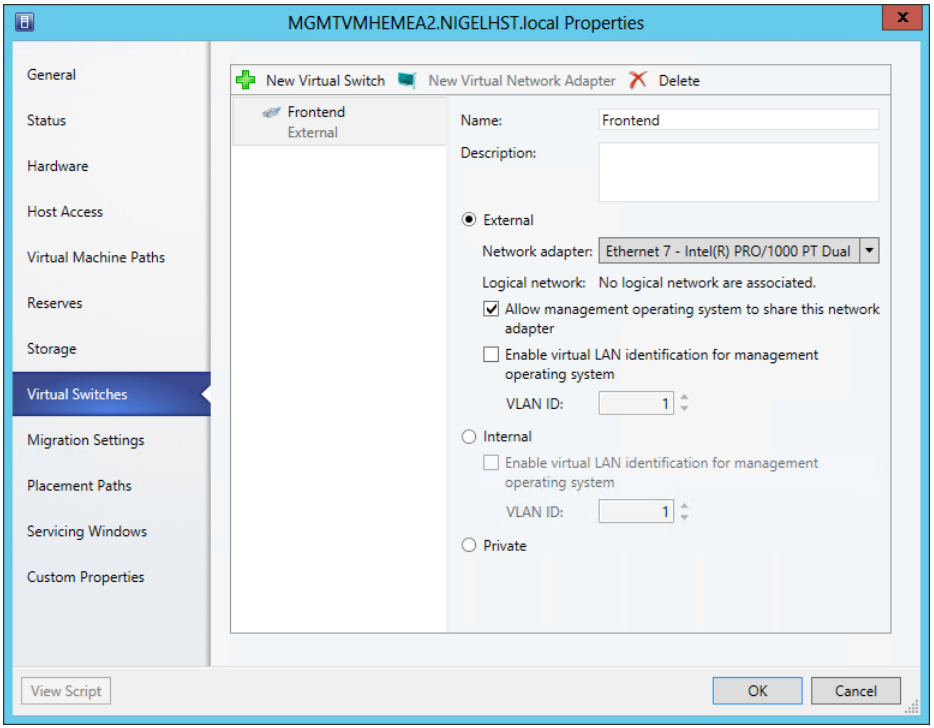


FIGURE 6-8 How a standard Hyper-V switch is represented in VMM

Preparation

You must first perform a few tasks before proceeding with the virtual switch migration. First, you must put the Hyper-V host into maintenance mode. To do so, in VMM, right-click the Hyper-V host and select Start Maintenance Mode, as shown in Figure 6-9, or select the option from the ribbon.

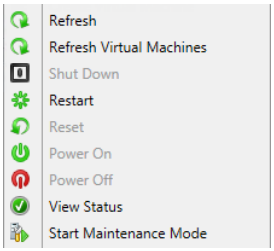


FIGURE 6-9 Enabling maintenance mode on a Hyper-V host in VMM

This evacuation process can be used to move all VMs from one host in a cluster to another host in the cluster by using Live Migration. If the host is not part of a cluster, or if no compatible Hyper-V host is available, the VMs will be put into saved state, which causes users to lose service. This method can also be used for non-highly available VMs running on a clustered Hyper-V host.

Even if the Hyper-V host is ready in theory for maintenance actions, be sure to check first that no VMs remain on the host. You can do this using Windows PowerShell by running the following command:

```
Disable-SCVMHost -VMHost MyHyperVHost -MoveWithinCluster
```

IMPORTANT **Don't confuse this command with Stop-SCVMHost, which would send a stop command to the baseboard management controller.**

In addition, whenever you perform intensive network changes, it is recommended that you connect to the console using an out-of-band interface rather than an RDP connection to the management network adapter.

NOTE **If VMM has been integrated with Microsoft System Center 2012 Operations Manager, the maintenance mode information will be passed to the monitoring system. This can help ensure that there are no unnecessary alerts when changing network connectivity or rebooting the system.**

Transitioning

Because there isn't an actual migration action to change from a standard switch to a logical switch, you actually first need to break the current configuration. To do this, delete the existing virtual switch using the Hyper-V management console or by using the Remove-VMSwitch cmdlet as follows:

```
Remove-VMSwitch -Name MyVirtualSwitchName
```

This operation also removes all virtual network adapters and their configuration. As mentioned in the previous section, be sure to use an out-of-band interface when applying this configuration change.

After the virtual switch has been successfully deleted, you can then remove the network team. This can be done in Server Manager or by using the Remove-NetLbfoTeam cmdlet as follows:

```
Remove-NetLbfoTeam -Name MyNetworkTeamName
```

Make sure that the first network adapter now has the management IP address configured. If this part of the configuration is lost, you need to configure it manually, including the DNS servers and gateway.

The host should now be back online, but make sure network connectivity and especially the connection to the VMM management server is working as expected.

To reflect these changes in VMM, the Hyper-V host configuration must be updated before deploying the logical switch. By default, the Host Refresh job (HostUpdateInterval) runs every 30 minutes, but to make sure the hardware changes are immediately represented in VMM, you can start a manual refresh job, as shown in Figure 6-10. When this is finished after a few seconds, be sure to verify the information in the Hyper-V host properties by checking the Virtual Switches tab. This should now be empty and should no longer display a standard switch.

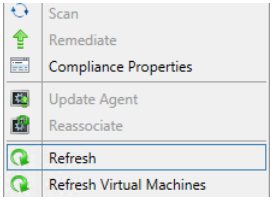


FIGURE 6-10 Refreshing a Hyper-V host in VMM

Once the host configuration has been cleaned, the logical switch can be deployed from the VMM console. The detailed steps for how to configure a logical switch with an untagged or tagged VLAN environment are described earlier in this chapter. You should disable maintenance mode once the logical switch has been deployed successfully to allow the VMs to be migrated back to the host.

Known deployment issues

The following sections describe some known issues concerning the deployment of logical switches using VMM.

Limitations for an existing NIC team

It's very important to know that VMM does not support the deployment of logical switches to Windows Server 2012 Hyper-V hosts that have already have been configured with a NIC team. This means before you can proceed with deployment, the existing NIC team must be removed. You can either remove one adapter from the NIC team, which can then be used for the logical switch, and do the clean-up later; or you can remove the complete NIC team and add both network adapters to the new logical switch.

Before proceeding with logical switch deployment, however, always first make sure that your Hyper-V hosts are configured with the correct IP address, subnet mask, gateway, and DNS servers. Also perform a host refresh in VMM to make sure the new configuration is reflected correctly. This can also be performed by using Windows PowerShell as follows:

```
Read-SCVMMHost -VMHost MyHyperVHost
```

An alternative and much more straightforward option is to leverage bare-metal deployment for Hyper-V host installation as described earlier in this chapter. In this scenario, logical switches can be provisioned right away as an integral step of the deployment workflow. This eliminates the need for swapping physical NICs one by one between teams.

To conclude, the only way to provision a logical switch to a Hyper-V host is to take over raw physical NICs that are not currently assigned to NIC teams or virtual switches.

Deployment fails if host is out-of-scope

When using host groups in VMM to organize your Hyper-V hosts, the deployment of a logical switch on a Hyper-V host can fail with error 26874:

Error (26874)
This operation is not permitted since uplink port profile set <adapterGUIDstring> in physical adapter <nameGUIDstring> on host <hostNameFQDN> would go out of scope for host

Recommended Action:
Delete the logical switch instance on the affected host(s) and retry the operation.

This can happen if the host on which you are attempting to deploy the logical switch is not a member of the host groups that are defined in *every one* of the network sites included within the selected uplink port profile. To resolve this issue, you simply need to add the host computer to the appropriate host groups.

Deployment fails when using different network adapter types

When deploying a logical switch on a Hyper-V host that has different types of network adapters, the deployment might fail with the following error message:

> *Warning (25259)*
> *Error while applying physical adapter network settings to teamed adapter. Error code details 2147942484*
>
> *Recommended Action*
> *Update the network settings on the host if the virtual network adapter is connected to the host.*

When deploying a logical switch to a host with two or more network adapters from different brands (Broadcom and Intel, for example), the job fails with the error 2912. Since VMM uses the first pNIC in the list and creates the NIC team with this network adapter, the switch inherits the capabilities of this network adapter, such as VMQ, SR-IOV, Task Offload, MTU size, and so on. If you add additional network adapters that do not support these capabilities to the NIC team, the job fails.

To work around this problem, the existing network team must be destroyed, which means that all existing network connections, and if configured, virtual network adapters, will fail (lose connectivity). To avoid this situation, always make sure to start with the pNIC that has the least possible capabilities, followed by other pNICs that have the same or better capabilities. This ensures the team works with different brands or adapter types.

For the record, the same thing happens in the default NIC teaming configuration wizard when you try to add a less capable pNIC to an existing network team.

CHAPTER 7

Operations

After logical switches have been deployed in the manner described in the Chapter 6, "Deployment," the focus moves to managing and operating the solution, making changes to the architecture, for example to accommodate new clients, new services, and business/environment changes, and monitoring network utilization to determine load patterns, peak periods, and which customers and services are generating the most traffic.

This chapter will:

- Describe how to approach some of the challenges associated with monitoring and reporting on network utilization
- Explain how to deal with common changes administrators need to make to a deployed virtualized network environment

Monitoring network utilization

It is often useful to understand network utilization and traffic patterns. Service providers, for example, want to know and understand load patterns, peak periods, and which customers and services are generating the most traffic. Tenants want insight into and details of what their virtual machines and services are doing on the network so that they ensure that they pay only for what is necessary.

The Hyper-V Management Pack for System Center Operations Manager does indeed provide counters which can help by allowing service providers to monitor and gain insight into the usage of the virtual switch, physical network adapter, and virtual network adapter, amongst others, and to create dashboards and reports that showcase this information in a more readily consumable fashion.

> ***See also*** *You can find more details on how System Center Operations Manager can be used to monitor network devices and network utilization at https://technet.microsoft.com/en-us/library/hh212935.aspx.*

However, service providers commonly want to understand the total amount of bandwidth used by a VM, (the total of both data downloaded from and data uploaded to the VM) and the peak bandwidth (the maximum bandwidth used at any time). The latter helps service providers to allow or support streaming so that they can bill their customers for this usage. To support this level of insight, obtain the required information by running the following Windows PowerShell commands for each VM on a given host computer:

```
$vm = get-vm [VMName]
Enable-vmresourcemetering $vm
Measure-vm $vm
```

It's important to note that to get a clear picture of utilization, the service provider might need to monitor a given VM or set of VMs over a significant period of time or at specific points in time. This is necessary to properly measure overall usage and any particular peaks or patterns. Ideally, a program or service would run the preceding Windows PowerShell commands over a period of time, collect the data, and run calculations and analytics against it. The bottom line is that monitoring usage involves developing a custom solution that can gather this data.

NOTE **Live migration of a VM from one host to another during the monitoring period might require restarting the monitoring exercise on the new host given that the measure-VM command is scoped to only a single host.**

Managing the environment

Factors outside of your immediate control, such as acquisitions, changing business requirements, and technology developments, might force you to review and make one or more changes to your virtualized network solution. The remainder of this chapter walks through some relatively common scenarios and provides some detailed recommendations, advice, and guidance around how to best address them. Although they may be read end to end for background, each scenario will probably be most useful to read when you experience that condition or need to prepare for that specific change within your own environment.

Logical switches

The following sections review some of the most common changes a logical switch might require after it has been deployed on a host computer, providing some guidance for successfully implementing those changes and a workaround if the change cannot be made directly.

Adding support for SR-IOV

With SR-IOV, network traffic bypasses the software switch layer of the Hyper-V virtualization stack. As a result, the I/O overhead in the software emulation layer is diminished while the network performance achieved using the interface is nearly the same as in non-virtualized environments.

As discussed in Chapter 3, "Hyper-V port profiles," enabling support for SR-IOV requires you to make changes in multiple places within your virtual network architecture, including the physical host and the uplink port profile and the logical switch in System Center 2012 Virtual Machine Manager (VMM). Assuming that you have enabled the settings required on the Hyper-V host and in the uplink port profile, but have forgotten to do so on the logical switch, the basic question is can you make those changes after the fact and enable SR-IOV on the deployed logical switch.

Unfortunately, although this appears to be a relatively frequent error for those using this form of processor offloading technology, changing the SR-IOV settings after the logical switch has been deployed is not possible. The option to enable SR-IOV is no longer available within the VMM administrator console after the logical switch is deployed, and any attempt to work around this limitation using Windows PowerShell simply fails with the following error message:

> *Error (25212): SR-IOV property (logical switch name) cannot be changed on this logical switch because there are sets of port profiles for virtual network adapters that refer to this property.*

The only remediation for this particular scenario is to remove the existing logical switch from any and all network adapters on which it has been deployed and to deploy a new logical switch on which SR-IOV support has been enabled.

> ***See also*** *You can find more details on SR-IOV at http://blogs.technet.com/b/privatecloud/archive/2012/05/14/increased-network-performance-using-sr-iov-in-windows-server-2012.aspx.*

Changing the assigned physical network adapter

As discussed in Chapter 3, a single uplink port profile may be applied to multiple physical network adapters in the same host computer (as part of logical switch deployment). The Load Balancing setting with the uplink port profile indicates whether each adapter should function standalone or should instead be configured to act as part of a team.

If one of the teamed physical network adapters fails or needs to be replaced, there is little or no real issue in the short term. You can leave the remaining adapters to provide service, albeit with reduced resiliency to failure and potentially some degradation in overall performance, until the next maintenance window. At that point, you simply replace the failed adapter and apply the same logical switch and uplink port profile to its replacement. The new adapter will automatically become a member of the existing team.

When network adapters are used in standalone mode, this process is clearly not as simple. Assuming that you have already added a replacement physical network adapter to the host computer, you cannot simply edit the logical switch and configure it to use the replacement since attempting to do so results in the following error:

> *Error (26864): Cannot change the uplink physical network adapter of a non-teamed logical switch instance (logical switch name) since it could lose connectivity—delete the logical switch instance and create a new logical switch instance with the desired uplink physical network adapter.*

As the above error message suggests, it will be necessary to delete the logical switch instance from the failed network adapter and deploy a new instance on the replacement. When the existing logical switch and the failed physical network adapter have been successfully removed from the Hyper-V host computer, you can either wait for the next automatic host refresh in VMM, or you can trigger this to occur on demand to force the VMM Agent to discover the new network adapter, at which point you can re-deploy the logical switch.

To avoid this situation in the future, it might be preferable to configure the majority of your uplink port profiles for teaming, and, in cases where a single physical network adapter has been dedicated to a specific function or operation, create a team of one. Then if something should happen, you can simply add a new physical network adapter to the host, join this adapter to the team, and remove the old one. This approach will allow you to recover from the problem without having to remove the logical switch as described earlier. There are some circumstances, physical network adapters dedicated to SMB 3.0 or that support SR-IOV for example, in which this workaround is not suitable, and you should make a point of reviewing each group of adapters in turn to determine the merits or otherwise of using this strategy to mitigate physical network adapter failure.

NOTE **The process for remediating this issue is very different if you are using a Hyper-V virtual switch (referred to as a standard switch in VMM) instead of a logical switch. If this is the case, you first need to change the standard switch mode from External to either Private or Internal. Having done so, change the mode back to External and then configure the standard switch to use the new physical network adapter.**

Converting from a standard switch to a logical switch

A Hyper-V virtual switch (known as a standard switch in VMM) is a software-based layer-2 network switch that becomes available once the Hyper-V server role is installed on a host computer. The standard switch includes programmatically managed and extensible capabilities to connect VMs to both virtual networks and the physical network and provides policy enforcement for security, isolation, and service levels.

The main issue with the Hyper-V switch is manageability since each switch is independent and must be configured separately. In VMM, the switch concept is greatly enhanced through the use of logical switches (essentially templates for Hyper-V switches) that allow you to consistently apply the same settings and configuration across multiple hosts and further to ensure that any Hyper-V switches deployed using the template remain compliant with it.

There is no easy migration path from a standard switch to a logical switch since after the physical network adapter has been associated with a standard switch, you cannot subsequently upgrade it to a logical switch. You must first disconnect and remove the standard switch and any associated virtual network interface cards (vNICs) from the network adapter and remove or break any pre-existing network adapter teams (as described below) before you begin to deploy the logical switch.

Handling pre-existing network adapter teams

Windows Server 2012 and subsequent releases allow you to combine multiple network adapters in the form of a NIC team to aggregate bandwidth and to provide for traffic failover, preventing connectivity loss in the event of a network component failure.

You can create a team on a Windows Server system manually from within Server Manager or by using Windows PowerShell. Having done so, however, you will be unable to deploy a logical switch to any of the network adapters that participate in that team. The fundamental issue is that VMM has no direct insight into how the network team was originally created or its current configuration. As a result, any attempts to assign a logical switch will fail with the following error:

> *Error (26900): A logical switch instance cannot be created on the physical network adapter (team name) because the adapter is a teamed adapter-delete the team from the host and create a logical switch instance on the physical network adapters.*

You can either leave the network team as is, with the understanding that these interfaces can only be used with standard Hyper-V switches, each team needs to be configured and managed separately and finally, that the team and corresponding network adapters fall out of the scope of management of VMM or remove the team and have VMM re-create it during logical switch deployment.

The primary benefits of moving from a team created directly on the Hyper-V host to one that is generated as a result of the deployment of a logical switch, as discussed earlier, are consistent configuration across a large number of hosts coupled with the ability to monitor compliance and to remediate (fix) deviations from expected configuration.

As the error message suggests, to deploy a logical switch to network adapters teamed directly on the host you must first break the existing team. Having done so and having forced a host refresh to allow VMM to discover the new configuration, you can then deploy a logical switch onto each network adapter that you want to team, with an uplink port profile used to define the teaming mode and load balancing port protocol (see Chapter 4 "Logical switches" for more details).

> ***See also*** *You can find an overview of NIC teaming at http://technet.microsoft.com/en-us/library/hh831648.aspx.*

Monitoring logical switch compliance

One of the advantages of logical switches compared to standard Hyper-V switches is that VMM can monitor the expected configuration across all host computers and remediate (fix) any differences. At each host refresh, VMM checks and verifies the configuration of the logical switch on each physical network adapter on which it has been deployed, reporting any deviation from the expected configuration, as shown in Figure 7-1.

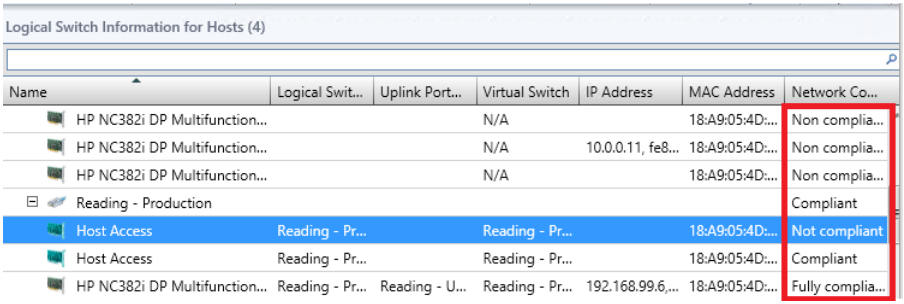


FIGURE 7-1 Logical switch compliance report

For each network adapter on which the logical switch has been deployed, the report indicates one of the following status values:

- **Fully Compliant or Compliant** This indicates that the settings on the host are consistent with the expected configuration in VMM.
- **Partially Compliant** This indicates that there is only a partial match between the settings on the host and expected configuration.
- **Not Compliant** This indicates that the deployed logical switch is significantly different from the expected configuration. This state is most likely caused by a modification directly performed on the Hyper-V host, such as adding or removing an additional virtual network adapter or changing the bandwidth control mode outside of VMM.

For any logical switch that shows as either Partially Compliant or Not Complaint, the reason for the discrepancy will appear in the Compliance Errors section. The Remediate option available through the VMM administrator console can be used to address and resolve any of the issues that have been discovered. Note that you might find that resolving one issue triggers subsequent discovery of another. If this occurs, you should continue with remediation until all network adapters show as Fully Compliant.

Depending on the nature of the property values that are changed as part of the Remediate action, connectivity for guest VMs and even the host itself may be disrupted. As a consequence, it is recommended that you review compliance errors reported and arrange to remediate partially or non-compliant logical switches, place the host into maintenance mode (to evacuate the VMs), and then take the necessary steps to remediate the issue.

Logical networks

The following sections review some of the common changes that might be required for logical networks and network sites, provide some advice and guidance for successfully implementing those changes, and explain how to work around the problem if necessary.

Moving from VLAN isolation to network virtualization

When using network virtualization as an isolation mechanism, virtual networks are defined entirely in software. As a result, it is unnecessary to reconfigure the physical network (unlike VLAN and PVLAN solutions) to onboard or remove new tenant networks or to make changes to reflect new business requirements. The benefits of such an approach are clear, but having configured a logical network to use either VLAN or PVLAN isolation as described in Chapter 2, "Logical networks," there is unfortunately no way to change it. To use network virtualization, therefore, you must create and deploy a completely new logical network, together with network sites, IP pools, and associated VM networks.

If the original VLAN (or PVLAN) logical network was associated with host network adapters through logical switches, you may be able to simply add the new network sites to the appropriate uplink port profiles defined within each logical switch. VMM will automatically update all of the host computers using the updated uplink port profiles and ensure that the hosts are associated with the new logical network. You can then migrate all of the VMs and services, disconnecting them from the existing VM network and connecting them to one that is associated with the new logical network. Of course, some downtime should be anticipated during this process.

The downtime in this instance is a result of the need to shut down the VM and change the network adapter configuration given that any network adapter connected to a VM network configured for Network Virtualization requires a static MAC address. The reason for this is relatively simple—since the VM can be live migrated between different host computers, each could allocate the VM a different provider address (PA). The only guaranteed, unique, and unchanging attribute in this environment is the machine's MAC address.

NOTE It is not necessary to create a static MAC address pool since one is provided by default, but if you would like to create one, you can find the steps to do so at *https://technet.microsoft.com/en-us/library/gg610632.aspx*.

The MAC address itself can either be manually specified or automatically allocated from a MAC address pool. Note that placing the value of "00:00:00:00:00:00" in the static MAC address field (as shown in Figure 7-2) indicates that VMM should allocate an address from the MAC pool.

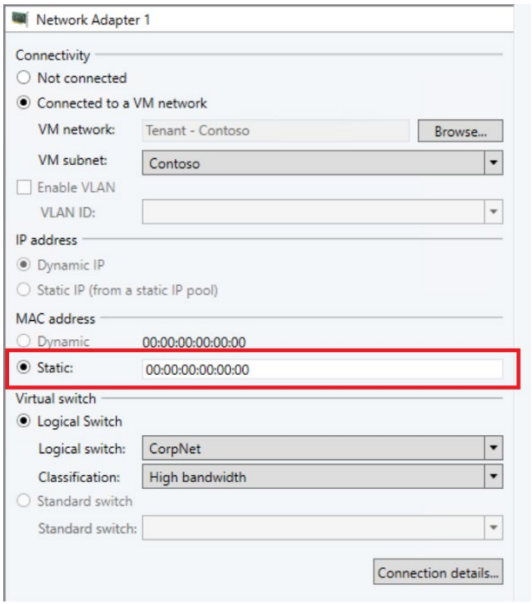


FIGURE 7-2 Configuring a network adapter to use a MAC address from a MAC address pool

When all of the VMs have been successfully migrated to the new VM network, you can remove the VLAN or PVLAN isolated logical network as described in the "Deleting a logical network" section later in this chapter.

Changing VLAN and PVLAN ID numbers

In environments that are using VLANs or PVLANs to isolate network traffic, it may become necessary at some point to change the VLAN ID numbers allocated to specific networks. The reasons for doing this can vary considerably, but all such changes will involve some form of disruption to normal service while switches and routing tables are updated to reflect the changes.

As you would expect, making such fundamental changes to the underlying network fabric will require you to make a number of corresponding changes to the solution you designed as part of the process described in Chapter 2 through Chapter 5. The open questions, therefore, are what needs to be changed to reflect the new environment and how can you make those changes with minimal effort, keeping downtime to a minimum.

To support VLAN isolation, a logical network must be configured such that sites within the logical network are not connected. In addition, each individual VLAN ID must be allocated to a network site, as discussed in Chapter 2. To allow VMs and services to connect to the selected logical network using the Network ID, each VLAN must be associated with a specific VM network.

If no such association currently exists, you are free to update and make changes to the network ID within the network site without issues until or unless you have created an IP pool linked to that site. In that case, the option to change the VLAN ID for the site is no longer available within the VMM administrator console, and your only recourse is to remove the IP pool and recreate it after the VLAN ID has been changed.

To remove the IP pool, you might first have to revoke the IP addresses that have been allocated to the VMs and services using the logical network. In most cases, IP addresses should be automatically returned to the pool as each VM and vNIC is disconnected, but there can be exceptions. For example, you can use the Inactive Addresses tab of the IP Pool Properties page to view and release any IP addresses that are no longer in use but were never returned to the pool. If there are a lot of allocated but inactive addresses, you can use the following Windows PowerShell script to return any of these addresses to the pool prior to removing the pool itself:

```
$ip = Get-SCIPAddress –IPAddress <IP Address>
$ip | Revoke-SCIPAddress
```

In cases where you have established an association between the network site and a VM network, the option to make changes to the VLAN ID within the network site will also be unavailable. If you attempt to change this via Windows PowerShell, the following error is returned:

Error (25176): The specified Subnet VLAN cannot be removed because it is being used by VM subnets—remove the referenced VM subnets and try again.

The steps required to mitigate this particular condition can be significantly more impactful than the previous case. As the error message suggests, you cannot simply change the VLAN ID without first deleting the existing VM network. Since the VM network in question may be used by any number of VMs, each of which would remain disconnected from the network until the changes to the network site have being made and a new VM network has been created, the following is the recommended way to mitigate this specific issue.

Instead of changing the existing network site as described previously, plan to add the new VLAN ID and subnet to the existing network site. You can then create a VM network tied to this VLAN ID and gradually migrate all of the VMs and services from the old VM network to the new one. This approach also provides a fallback position in the sense that the existing VM network still exists and can be used until you confirm that the new configuration is working as expected.

NOTE **This process works only when both the VLAN ID and the IP subnet is changed because VMM does not allow you to create a VLAN that has the same subnet as another. In such cases, the remediation steps are more extensive, requiring that you use a temporary (interim) subnet during the transition period.**

You can follow a similar process, in essence creating a new site and mapping to a new VM network, whenever you need to change either of the values defined for the Primary VLAN ID or the Secondary VLAN ID in a network site that is part of a logical network configured to support PVLANs.

Deploying new logical networks

You can add network sites for any new logical networks to the uplink port profiles defined within a logical switch at any time. VMM will automatically update all of the host computers that are using the updated uplink port profiles and ensure that network adapters in those hosts are correctly associated with the new logical network. No additional configuration is required.

Deleting a logical network

As described in Chapter 2, logical networks are connected to a significant number of objects within your virtual network architecture. As a result, the process to remove them requires careful coordination; VMM does not allow you to remove a logical network while one or more other objects have a direct dependency on it. To discover which objects are preventing successful deletion, you can use the dependency action within the VMM console. An example of this is shown in Figure 7-3. Note that this list must be empty before you can successfully delete the logical network.

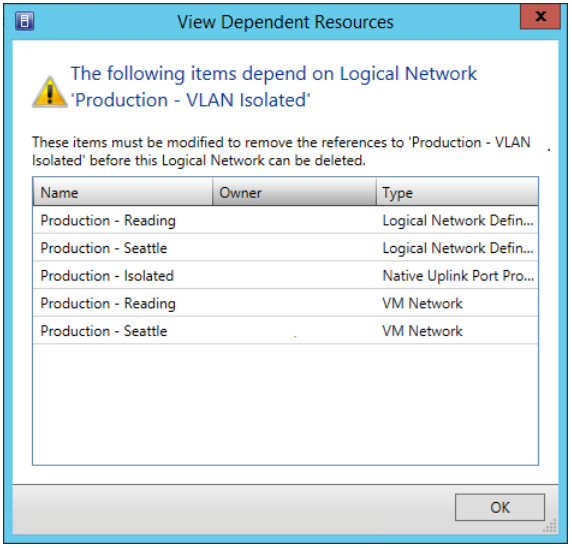


FIGURE 7-3 Checking for dependent resources prior to logical network deletion

The list of dependencies can include objects such as network sites (note that these are listed under the Type column as logical network definitions), load balancers, IP address pools, hosts, VMs, services, and any templates that exist in the library. As you would expect, before you can successfully delete the logical network, you must first modify or delete all of these dependent items.

The same issue with respect to deletion or removal of a dependency chain is true of most objects within VMM. To ensure that you can actually delete any an object, you must first review and remove or disconnect any objects that have dependencies upon it.

VM networks

This final section reviews the two most common scenarios relating to VM networks: the need to map an existing VM network to a different logical network and how to effectively delete a VM network.

Mapping a VM network to a new logical network

The relationship between a VM network and its host logical network is established when the VM network is initially created and cannot be changed afterward. To use a different logical network, you should first create a new VM network linked to the correct logical network and connect VMs and services to this VM network. You can then safely remove the previous VM network.

Removing a VM network

The proper way to delete a VM network is to start by deleting or disconnecting all of the virtual network adapters associated with the VM network. This includes VMs and service templates that have virtual network adapters associated with the VM network (see Figure 7-4). You then delete any IP pools and finally the VM network itself.

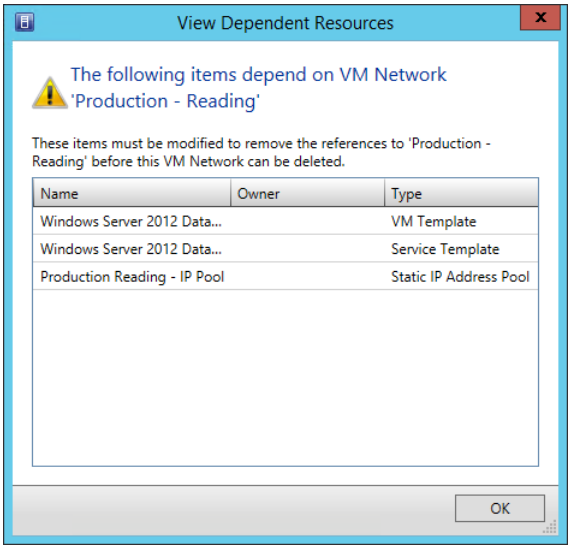


FIGURE 7-4 Checking for dependent resources prior to VM network deletion

As with logical networks, to remove the IP pool, you might have to revoke the IP addresses that have been allocated to VMs and services using the logical network. In most cases, IP addresses should be automatically returned to the pool as each VM or vNIC is disconnected, but there might be exceptions. For these specific cases, you will need to use Windows PowerShell, as shown in the following example, to return any of these addresses to the pool prior to removing the pool itself.

```
$ip = Get-SCIPAddress -IPAddress <IP Address>
$ip | Revoke-SCIPAddress
```

The Revoke-SCIPAddress command removes the IP address from the list of assigned IP addresses. When the command completes successfully, you can then delete the IP pool for the VM network, then the site, and then the VM network.

CHAPTER 8

Diagnosing connectivity issues

Networks are all about connectivity. Computers that connect to and use the same network need to be able to communicate with each other and, through the use of routers and gateway devices, to resources on external networks. An inability to connect to resources either on the same network or on an external network is clearly a problem.

As you would expect when dealing with network connectivity issues, the first and most important steps in fault diagnosis involve checking and confirming that a host machine and the virtual machines (VMs) that run on it do indeed have valid IP addresses. After verification of the IP addresses, the next step involves validating communication with other devices on the local network, then the local gateway, and, finally, devices on a remote (external) network. The process and the tools network administrators use to test this basic level of connectivity, including ipconfig, ping, and tracert (trace route), are all well-known and well understood, so this chapter assumes that you've carried out these initial checks and that some form of connectivity issue still remains. This chapter discusses how to approach a connectivity problem with a virtualized network solution, the processes you should follow to troubleshoot and diagnose what has happened, and, where appropriate, suggests some actions you can take to remediate the problem.

This chapter will:

- Outline a process for troubleshooting and diagnosing issues with connectivity
- Provide details of the Windows PowerShell commands and troubleshooting tools that you can use to find out determine why connectivity has failed or is failing
- Outline some of the actions you can take to restore connectivity

Where is the failure?

As discussed in Chapter 1, "Key concepts," a virtualized network solution is essentially constructed of a physical layer consisting of physical networks, host computers, switches, routers, and gateways and a virtual layer made up of logical networks, logical switches, VM networks, and (optionally) virtual appliances.

While identifying, diagnosing, and remediating issues in the physical layer are well beyond the scope of this book, the cause of a failure or connectivity problems in the virtualized network can often be attributed to some underlying faults that have developed in the physical network, so it is always worth beginning the process of fault diagnosis by starting from basic principles, validating and confirming that the physical host computer is able to communicate successfully before starting to explore and consider more detailed explanations. Examples of potential problems in the physical layer include disconnected or faulty physical network cables or misconfigured IP addresses on the host computer; incorrectly configured network appliances, such as gateways and routers in the wider network; and outages in service provider and telco services that provide access to the Internet and to external networks.

Having verified that the physical host is able to successfully communicate, the focus of troubleshooting efforts moves to the virtual environment. Some of the basic issues are easy to verify, for example, confirming that a given VM is running, is connected to the required network, and has a valid IP address/subnet mask. However, investigating and diagnosing more complex issues, especially given a requirement to establish connectivity to resources on different internal and external networks, is somewhat more challenging.

The remainder of this chapter describes how you should approach a connectivity problem in a virtualized network solution, the process you should follow and some of the tools you can use to help fault diagnosis, and some actions you can take to remediate and reestablish connectivity.

A step-by-step approach

The goal in this chapter is to present a step-by-step approach to diagnosing and troubleshooting issues with virtual network connectivity, starting from the underlying principle that it is best to assume a simple root cause initially, especially when the issue relates to connectivity to resources on other internal and external networks, and move from there toward a more complex and involved diagnosis.

The recommended process for diagnosing and troubleshooting issues can be roughly summarized as follows:

1. Validate that the Hyper-V host computers are configured correctly and able to communicate with other internal and external resources.
2. Confirm that the host is able to support tenant network services and that, in the case of network virtualization, the appropriate provider IP addresses have been allocated to each VM.
3. Validate that the VM is connected to the network and has been allocated an IP address for that network and that the address, subnet mask, gateway, and so on are correct.

4. Confirm that the Hyper-V Network Virtualization gateway VM is operating and has been configured correctly.
5. Only if needed, perform a detailed analysis into network packets.

The following sections expand on each one of these steps and, where relevant and appropriate, calls out the set of Windows PowerShell cmdlets and tools that you can use to discover more information, find out what is happening behind the scenes, and ultimately identify the root cause of a particular connectivity problem.

Step 1: Confirm host connectivity and physical configuration

When deployed solutions that were working begin to experience issues, connectivity failures might be related to system outages, as mentioned previously, or due to changes in the internal or external environment, either scheduled or unscheduled. Therefore, the first step in the troubleshooting process should be some degree of investigation to determine whether external factors are at play; the failure or root cause of a given failure might have nothing to do with the virtualized network solution itself.

A lack of connectivity in a newly deployed systems architecture might also be related to such changes in the environment but could equally be a consequence of design choices, some previously unknown environmental constraints, or some elements of physical configuration. In those cases, it might be worth reviewing the design and physical implementation and reconfirming some of the base assumptions and constraints, especially where network security and firewalls are concerned. You can find additional information and a checklist outlining best recommendations for Hyper-V host setup and configuration at *http://blogs.technet.com/b/askpfeplat/archive/2013/03/10/windows-server-2012-hyper-v-best-practices-in-easy-checklist-form.aspx*.

Identifying, diagnosing, and remediating the majority of issues with the physical network and the host itself are beyond the scope of this book, but assuming that the health and configuration of the physical network and the Hyper-V host computers are actively monitored through a management tool such as System Center Operations Manager, connectivity issues encountered at this level generally will have been identified and reported, along with information and outline notes suggesting remedial action to restore service.

> ***See also*** *System Center Operations Manager can be used to monitor both the health of the physical network as well as the Hyper-V 2012 host servers, including the configuration of critical services and disks, Hyper-V VMs, virtual features, and virtual hardware. You can find more information on the Hyper-V Management Pack for Operations Manager at https://www.microsoft.com/en-us/download/details.aspx?id=36438.*

With that said, when connectivity issues occur that don't appear to be related to the physical network, you should check that the Virtual Machine Manager (VMM) server is working, ensure that the Hyper-V virtual switches deployed on each host are set up and

configured correctly, and make sure that the VMM DHCPv4 Server Switch Extension, which allows VMM to dynamically assign IP addresses to VMs, is installed and enabled.

Confirm that the VMM server is available

It's important to understand that Hyper-V provides the platform for network virtualization, but VMM is really the controller for how networks and routing policies are distributed to Hyper-V hosts. Every time a VM connected to a virtualized network is moved between hosts, VMM updates the extensions on all relevant hosts with details of this environmental change. If VMM is not running and a VM is moved via Hyper-V or Windows PowerShell, that VM effectively drops off the VM network that it is connected to. Of course, as soon as VMM recovers, it scans for environment changes and updates all the extensions as quickly as possible. Therefore, if VMM is unavailable and you are using virtualized networks, you should ensure that no VMs are moved while you are troubleshooting an issue with the VMM server. This will ensure everything remains healthy until VMM is restored.

Hyper-V virtual switch

The Hyper-V virtual switch (known as a standard switch in VMM) is a software-based layer 2 network switch that becomes available when the Hyper-V server role is installed on a host computer. Each instance of this switch is independent, meaning that it is deployed and configured separately, so it is relatively easy to introduce minor errors or misconfiguration between switches deployed on network adapters on different hosts or even within the same host, potentially leading to unpredictable results. For more details, see Chapter 7, "Operations."

If logical switches have been deployed on the Hyper-V hosts as described in Chapter 6, "Deployment," VMM can monitor the expected configuration across all host computers and remediate any differences. At each host refresh, VMM checks and verifies the configuration of the logical switch on each physical network adapter on which it has been deployed, reporting any deviation from the expected configuration. The Remediate option available through the VMM administrator console can be used to address and resolve any issues that are discovered.

VMM DHCPv4 Server Switch Extension

The VMM DHCPv4 Server Switch Extension allows VMM to dynamically manage and assign IP addresses from a static IP address pool. Without it, no IP addresses are allocated and, depending on the configuration, VMs requiring an IP address revert to the standard default address, which basically means they are unable to connect to other network resources. A properly configured VMM DHCPv4 Server Switch Extension is shown in the Hyper-V Switch properties in Figure 8-1.

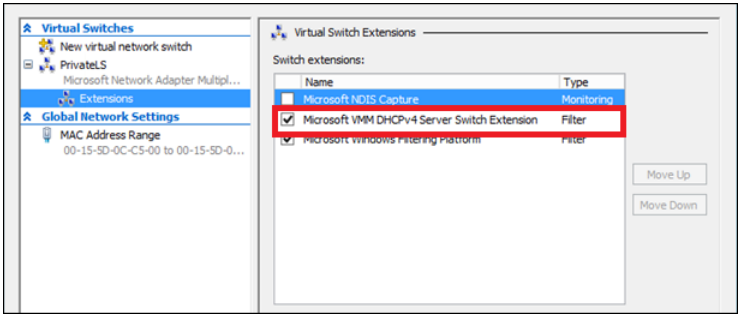


FIGURE 8-1 A Hyper-V switch with the VMM DHCPv4 Server Switch Extension enabled

You can obtain the same information from each host by using the Windows PowerShell command Get-VMSwitchExtension, providing the name of the host and the Hyper-V switch, and then filtering the results to Microsoft VMM DHCPv4 Server Switch Extension, as shown next:

```
PS C:\> get-vmswitch -computername "cl5007a-th" | Get-VMSwitchExtension | where {$_.name
-match "Microsoft VMM DHCP*"} | fl computername, switchname, name, enabled, running
```

In this example, the command returns information on all of the switches running on host CL5007A-TH as shown in the output below. The advantage of using Windows PowerShell for is that you can quickly and easily determine the status of the extension for one host or multiple hosts by simply changing the -ComputerName parameter. Note that the command may return multiple entries, one for each Logical or Virtual Switch that has been configured.

```
ComputerName : cl5007a-th
SwitchName   : CorpNet vSwitch
Name         : Microsoft VMM DHCPv4 Server Switch Extension
Enabled      : True
Running      : True
```

If for some reason the switch extension is not present on a given Hyper-V host, you can install it from the VMM server under the directory C:\Program Files\Microsoft System Center 2012 R2\Virtual Machine Manager\SwExtn\DHCPExtn.msi.

Step 2: Confirm host is providing tenant network services

Having confirmed that the Hyper-V host is able to connect to the network successfully, that the virtual switches associated with each network adapter in that host are configured correctly, and that the VMM DHCPv4 Server Switch Extension is enabled, the next step is to confirm that the network services required to isolate tenant VMs are being provided by the host computer.

VLAN/PVLAN isolated networks

It is still relatively common to find VLAN isolation technology used to isolate and prioritize different types of workloads on the physical network even in today's world of converged networks. VLANs are even used in the Microsoft Cloud Platform System (CPS), discussed in Chapter 9, "Cloud Platform System network architecture," to keep management traffic (including host management) separate from virtualized tenant traffic on the same physical network.

In an environment where VLANs are in use, it's important to ensure that the VLAN or PVLAN ID associated with a given host (or VM) is correct, meaning that it matches what is both configured and expected by the switches and routers on the physical network. You can find details of how to verify and define both VLAN tagged and untagged network adapters for host management at *http://blogs.technet.com/b/scvmm/archive/2013/11/07/logical-switch-deployment-with-virtual-network-adapters-for-host-management-in-vmm-2012-sp1.aspx*.

You should also verify the VLAN and PVLAN assignments for all of the logical networks associated with the host to ensure that when VMs are connected to those networks via a VM network, the correct tags are applied to network traffic coming from those machines. (See Chapter 2, "Logical networks," for more details.)

Note that it might be difficult to diagnose issues when VLANs are used because, in some cases, the VLAN tag is assigned on the physical switch, meaning that the network interfaces (on the host) are generally left blank. In these situations, you must query the physical switch, locate the ports used by a specific host, and review the port configuration to confirm whether this is set up correctly.

Linux guests

When working with Linux guests, a VM might fail to communicate on the virtual network even though everything appears to be configured correctly. The problem is that in some versions of Linux, the networking configuration can be lost when a new MAC address is assigned to the virtual network adapter, a situation that could occur when the VM is migrated to another host and the option to use dynamic MAC addresses has been selected. In such cases, you should ensure that each virtual network adapter has a static MAC address. You can configure the MAC address by editing the settings of the VM in VMM or Hyper-V Manager.

Virtualized networks

As discussed in Chapter 2, where VM networks have been isolated using network virtualization, tenants are assigned individual virtual networks and are able to use their own IP addresses and subnets, even if these conflict or overlap with those used by other tenants. As a consequence, diagnosing and troubleshooting problems with connectivity becomes a little more challenging given that, unlike the VLAN/PVLAN model described previously, all of the configuration and routing rules are performed in software rather than in physical hardware.

Confirm host supports network virtualization If you are experiencing connectivity issues with VM networks configured for network virtualization, you should first validate that the host is enabled for and is currently configured to support this method of isolation and that the required VM networks are available on that host. You can obtain this information relatively simply by using the Windows PowerShell command Get-NetVirtualizationCustomerRoute.

```
PS C:\> Get-NetVirtualizationCustomerRoute -CimSession "CL5007a-TH"
```

This command returns the full list of VM networks supported by a given host along with some useful summary information about each hosted network, including the virtual subnet ID and the customer IP address range (designated by the DestinationPrefix field), the NextHop address (which shows either 0.0.0.0 or the IP address of the gateway used to provide access to external networks), and the metric or route cost as shown next:

```
RoutingDomainID      : {66971027-72C7-44F1-B551-3CE76BA5954A}
VirtualSubnetID      : 14879496
DestinationPrefix    : 192.168.0.0/24
Next Hop             : 0.0.0.0
Metric               : 0
PSComputerName       : CL5007a-TH

RoutingDomainID      : {D3AE7AAC-E457-4AFA-A410-F2C269C3D814}
VirtualSubnetID      : 13573848
DestinationPrefix    : 192.168.100.0/24
Next Hop             : 0.0.0.0
Metric               : 0
PSComputerName       : CL5007a-TH
```

If nothing is returned when you run this command, either network virtualization has not been configured on the host, or the Windows Network Virtualization Filter is not installed or has been purposely disabled by the administrator.

Check that provider address and customer addresses are being assigned As noted previously, network virtualization allows you to run multiple network infrastructures on the same physical network with each of these networks isolated from and totally independent of each other, potentially using the same or an overlapping set of IP addresses. To achieve this, VMs connected to VM networks configured for network virtualization are actually assigned two IP addresses:

- **Customer address (CA)** This IP address is visible to the VM and is used by customers to communicate with the VM.
- **Provider address (PA)** This IP address is used by the Hyper-V computer that hosts the VM and is visible only to the host.

PAs are assigned from an IP address pool associated with the logical network, while CAs are assigned from an IP pool linked to a given VM network. For VMs to communicate on a given VM network, both of these addresses must be successfully allocated.

If network virtualization is enabled, you can use the Get-NetVirtualizationLookupRecord Windows PowerShell cmdlet to validate and confirm that the host computer is correctly allocating CAs and PAs to VMs as shown below. It is often useful to filter the results to a single VM so that you can quickly determine whether these addresses are being successfully allocated and whether they are valid. The CA should be valid/in scope of the IP pool assigned to the VM network and the PA should be valid/in scope of the IP pool that has been assigned to the logical network.

```
PS C:\> Get-NetVirtualizationLookupRecord | where {$_.vmname -match "ws_10074"} | fl
CustomerAddress, VirtualSubnetID, MacAddress, ProviderAddress
```

As you would expect, a VM needs a CA and PA for each VM network that it is connected to. If, for example, a VM is multi-homed, meaning that it is connected to two or more VM networks at the same time, you should expect to find a CA/PA pair for each of the VM networks.

```
CustomerAddress     : 192.168.0.2
VirtualSubnetID     : 14879496
MACAddress          : 001dd8b71c1b
ProviderAddress     : 172.172.10.4
```

If you find no mapping of PA to CA when one is expected, and if you have already validated and confirmed that the VMM DHCPv4 Server Switch Extension is installed and configured, the most likely cause of the problem is the Hyper-V Network Virtualization Policy job that runs on the VMM server. There is, unfortunately, no way to reset or restart this job; to fix this problem, you must restart the VMM Agent on the HyperV Host and then re-confirm that the PA and CAs are being allocated correctly.

NOTE **It is sometimes useful to know which IP address pool a given VM's PA was taken from, and you can obtain this using the Get-SCIPAddress command from the VMM server (or from a machine installed with the VMM Windows PowerShell extensions).**

Confirm that the interface to the physical network (the logical switch) is active

Assuming PAs are being successfully allocated, you can use Windows PowerShell to identify the network interface and from there the logical switch that the VM is using to gain access to the virtualized network. From there, you can validate that the switch is available and that at least one of the host network adapters associated with the switch has a valid connection to the physical network.

Having successfully determined that the PA is allocated to the VM, 172.172.10.4 in the earlier example, you can use Get-NetVirtualizationProviderAddress to identify the network interface (essentially, the logical switch) on which that address is valid, as shown below. The output from this comment returns the InterfaceIndex (or identifier) of the logical switch, in this case 11.

```
PS C:\> Get-NetworkVirtualizationProviderAddress | Where {$_.ProviderAddress – eq
"172.172.10.4"

ProviderAddress     : 172.172.10.4
InterfaceIndex      : 11
PrefixLength        : 24
VLanID              : 0
AddressState        : Preferred
MACAddress          : 001dd8b71c1d
ManagedByCluster    : False
```

A host should have one PA for each tenant network that has at least one VM on that host. The AddressState field shown above should be listed as Preferred. If the value of this field is displayed as Duplicate, this is an issue. Although duplicate IP addresses can be revoked, revoking an IP address puts the address back into the pool for reassignment. If the issue that caused the address to be duplicated is not corrected, the address could be duplicated once again. If no valid PA is returned from Get-NetVirtualizationProviderAddress, the target VM you are working with might have been moved to a different host computer. You should retry using different VM.

The next step is to use the InterfaceIndex number as the parameter for the GetNetAdapter Windows PowerShell command to retrieve the (friendly) name of the logical switch, its current status, and the link speed, as shown next:

```
PS C:\> Get-NetAdapter | Where {$_.ifIndex -eq "11"}

Name         InterfaceDescription                  ifIndex Status   MacAddress
LinkSpeed
----         --------------------                  ------- ------   ----------         -
--------
APJI Switch  Microsoft Network Adapter Multiplexo…      11 Up       3C-4A-92-DC-B0-62
2 Gbps
```

Finally, use the Get-NetAdapterStatistics cmdlet to confirm that the host is sending or receiving data through the logical switch, assuming that the status of the switch indicates that it is available (status of "up").

```
PS C:\> Get-NetAdapterStatistics | where {$_.name -eq "APJI Switch"}

Name         ReceivedBytes  ReceivedUnicastPackets       SentBytes  SentUnicastPackets
----         -------------  ----------------------       ---------  ------------------
APJI Switch  3569941705563              4312485002   3253298313944          3188105292
```

If no statistics are shown or the send or the received values are zero, unlike in the example below, the physical adapters associated with the logical switch might be disconnected or the physical network adapter settings might be incorrect.

Check logical switch compliance As discussed previously, the main benefits of using logical switches as compared to standard Hyper-V switches is consistent configuration across a large number of hosts coupled with the ability to monitor compliance and to identify and remediate deviations from expected configuration. It is quite common to find problems with host (and guest) connectivity occurring when a logical switch falls out of compliance. After confirming and validating host configuration to this point, the next step is to check compliance, either through Windows PowerShell, as shown next, or through the VMM console.

```
PS C:\> Get-SCVirtualnetwork -name "PrivateLs" | fl logicalnetworks, logicalswitch,
vmhostnetworkadapters, logicalswitchcompliance*

LogicalNetworks               : {Tenant - Virtualized}
LogicalSwitch                 : APJI Switch
VMHostNetworkAdapters         : {Ethernet – HP NC362i Integrated DP Gigabit Server
Adapter,
                                Ethernet 2 – HP NC362i Integrated DP Gigabit Server
Adapter #2}
LogicalSwitchComplianceStatus : Compliant
LogicalSwitchComplianceErrors : {}
```

If the logical switch on one or more hosts is out of compliance, you should take the actions necessary to remediate the issue as described in Chapter 7 before you proceed with any further troubleshooting and diagnostics steps.

Confirm ability to communicate from PA to all hosts that support a given logical network As discussed in Chapter 2, multiple Hyper-V hosts can be associated with a logical network, and having validated the configuration of the logical switch, the next troubleshooting step is to confirm that the host is able to route traffic to and from VMs that are connected to the same logical network but are on a different host.

To confirm host connectivity to the physical network and routing between different hosts, run the ping –p command from the host computer, first targeted at the PA of a VM running *on that host* to validate local connectivity.

```
PS C:\>Get-NetVirtualizationProviderAddress –cimsession "cl5007a-th" | where
{S_.ProviderAddress –eq "172.172.10.4"}

ProviderAddress     : 172.172.10.4
InterfaceIndex      : 11
PrefixLength        : 24
VLanID              : 0
AddressState        : Preferred
MACAddress          : 001dd8b71c1d
ManagedByCluster    : False
```

```
PSComputerName      : cl5007a-th

PS C:\> ping –p 172.172.10.4

Pinging 172.172.10.4 with 32 bytes of data
Reply from 172.172.10.4: bytes=32 time<1ms TTL=128
Reply from 172.172.10.4: bytes=32 time<1ms TTL=128
Reply from 172.172.10.4: bytes=32 time<1ms TTL=128
Reply from 172.172.10.4: bytes=32 time<1ms TTL=128
```

You should repeat the process, but this time, target the PA of a VM that is located and running on a different *remote host computer* but which (crucially) uses the same logical switch for connectivity.

```
PS C:\>Get-NetVirtualizationProviderAddress –cimsession "cl5007e-th" | where
{S_.ProviderAddress –eq "172.172.10.2"}

ProviderAddress     : 172.172.10.2
InterfaceIndex      : 13
PrefixLength        : 24
VLanID              : 0
AddressState        : Preferred
MACAddress          : 001dd8b71c19
ManagedByCluster    : False
PSComputerName      : cl5007e-th

PS C:\> ping –p 172.172.10.2

Pinging 172.172.10.2 with 32 bytes of data
Reply from 172.172.10.2: bytes=32 time<1ms TTL=128
Reply from 172.172.10.2: bytes=32 time<1ms TTL=128
Reply from 172.172.10.2: bytes=32 time<1ms TTL=128
Reply from 172.172.10.2: bytes=32 time<1ms TTL=128
```

If, after performing all of the previous verification steps, there is no response when you run ping –p with the PA for a VM located on a different host, the most likely cause is incorrect settings on the host network adapters, for example a disabled network adapter or an incorrect or invalid VLAN assignment.

NOTE **The PA is valid for a specific logical network and can be accessed only from Hyper-V hosts that are associated with that logical network (see Chapter 2 for more details). You cannot access the address from any other device within your network. Trying to ping the PA without the –p parameter, regardless of which host you attempt this from, also returns no results.**

Assuming these simple cases can be eliminated, more detailed investigation is required given that the cause of the connectivity problem is probably related to the physical network, the connection from the host to the physical switch or router, or some failure in the physical network.

Step 3: Check guest network settings and configuration

After you confirm that the Hyper-V host computers are working as expected, the focus of troubleshooting and diagnostics moves on to the configuration and setup of the individual guest VMs that are experiencing connectivity issues. The first and most important step is to confirm that the VM has an active network adapter(s) and has been assigned or is using IP addresses valid for the network(s) that it is connected to. From there, attention moves away from the machine itself to checking and confirming communication with other resources, either on the same network or on external networks.

Review guest network connection

As the service provider or owner of the Hyper-V hosts on which the VM is running, the first thing to do prior to asking the tenant to check the configuration and settings of the network adapters within the guest operating system is to confirm that the VM is actually connected to the network and, assuming an IP pool is set up, that it is being allocated a correct CA from that pool.

Outlined below is a combination of Windows PowerShell cmdlets that can be used to determine the list of network adapters within a VM called WS_100074_DC_en_us_Gen2, output the current status of those adapters, and show any customer IP address associated with those adapters. The results indicate that this VM has two network adapters, one of which is working normally and another that is disconnected (indicated by no data). In this example, restoring connectivity might be as simple as re-enabling the network adapter.

```
PS C:\> Get-VM | where {$_.name -match "WS_10074*"} | Get-VMNetworkAdapter | sort-object
MacAddress | fl VMName, MacAddress, SwitchName, Status, IPAddresses

VMName          : WS_10074_DC_en-us_Gen2
MacAddress      : 001DD8B71C1B
SwitchName      : APJI Switch
Status          :
IPAddresses     : {}

VMName          : WS_10074_DC_en-us_Gen2
MacAddress      : 001DD8B71C1C
SwitchName      : APJI Switch
Status          : {Ok}
IPAddresses     : {192.168.0.2, fe80::503c:af9c:9e66:7bd}
```

With the state of each adapter verified, you can then use VMM Windows PowerShell extensions to obtain the MAC address of each network adapter and the logical switch, logical network, and VM network it is using to connect and the list of IP addresses that are allocated to it. In this example, the first network adapter shows no results for the virtual network, logical network, and VM network since it has been disconnected as discovered previously. Much more information is available for the second adapter since it has a valid connection to the network.

```
PS C:\> Get-SCVirtualMachine | where {$_.name -match "WS_10074*"} | Get-
SCVirtualNetworkAdapter | sort-object | fl MacAddress, VirtualNetwork, LogicalNetwork,
IPv4Addresses, IPv4Subnets

MacAddress      : 00:1D:D8:B7:1C:1C
Virtual Network :
Logical Network :
VM Network      :
IPv4Addresses   : {192.168.100.21}
IPv4Subnets     : {192.168.100.0/24}

MacAddress      : 00:1D:D8:B7:1C:1C
Virtual Network : APJI Switch
Logical Network : Tenant Virtualized
VM Network      : Test
IPv4Addresses   : {192.168.0.2}
IPv4Subnets     : {192.168.0.0/24}
```

Connectivity problems can occur if the address is invalid or incorrect on the network, and you should verify that the IP address, subnet mask and VLAN ID, if appropriate, is indeed valid for the selected VM network. In the case of connections to VM networks isolated using network virtualization, the allocated IP address must also fall within the range of addresses that are defined in the IP address pool linked to that VM network.

NOTE **If you are using or have configured VM networks for VLAN isolation or if the underlying network is using VLANs to both segregate and prioritize network traffic, you can use Get-VMNetworkAdapterVLAN to identify and report the VLAN tags that are associated with network adapters in a VM.**

To confirm that the assigned address has been (correctly) allocated from an IP Pool, first identify the appropriate IP pool using the Get-SCStaticIPAddressPool VMM Windows PowerShell extension as shown below, filtering the results to show the set of IP pools that are associated with the VM network identified in the preceding step. As shown, the output from this command includes the range of valid addresses and the default gateways defined in the selected IP pools. The gateway information will prove useful later in this chapter to investigate why a VM with a valid IP address appears unable to communicate with external resources.

```
PS C:\> Get-SCStaticIPAddressPool | where {$_.VMSubnet -match "test"} | fl VMSubnet,
Name, IPAddressRangeStart, IPAddressRangeEnd, DefaultGateways

VMSubnet                    : test
Name                        : Test IP Pool
IPAddressRangeStart         : 192.168.0.2
IPAddressRangeEnd           : 192.168.0.250
DefaultGateways             : {192.168.0.1}
```

Having identified the IP pool, you can use Get-SCIPAddress to identify whether the address allocated to the network adapter is actually valid on the VM network, meaning that it falls within the defined range of addresses. If this command returns no results, the address is incorrect and further investigation is required to determine the root cause. In general, failure to obtain an address from an IP Pool can be attributed to manual changes to the IP configuration of the adapter (that is, assigning a fixed or static address that falls outside of the defined range) or certain changes made to the IP pool by the VMM administrator, such as revising the start/end range or changing the list of reserved addresses. Assuming there are sufficient free addresses in the pool, the first case may be resolved by assigning a different fixed address or by forcing the network adapter to use DHCP. In the latter two cases, simply renewing the IP address using IPConfig / renew should result in a new address being allocated to the network adapter.

```
PS C:\> Get-SCIPAddress | where {$_.name -eq "192.168.0.2"} | fl Address,
AllocatingAddressPool, State, Description

Address                     : 192.168.0.2
AllocatingAddressPool       : Test IP Pool (192.168.0.2 - 192.168.0.250}
IPAddressRangeStart         : 192.168.0.2
State                       : Assigned
Description                 : HNV CA DHCP
```

It is often necessary to assign a static IP address to a particular VM. This requirement is fully supported by VMM, which automatically discovers fixed addresses and tries to reserve those addresses in the pool. Problems can occur if the fixed address is already allocated to an existing VM, in which case, you should revoke the existing use of that address if it was allocated via DHCP or use a different fixed address. For more details on this process, see *https://technet.microsoft.com/en-us/library/hh801383.aspx*. The key point here is that static addresses are supported by VMM but to avoid connectivity issues, you must ensure that the assigned address falls within the scope of the assigned IP pool if using network virtualization as an isolation mechanism.

Check VM access to the network gateway

From the guest network connection, attention turns to the IP gateway since this device (either physical or virtual) provides the mechanism to allow VMs to reach other networks, with an inability or failure to communicate with the gateway understandably leading to a loss of access to external resources. For connections to VM networks isolated using network virtualization, a default gateway IP address is automatically assigned to the network adapter, although it is possible for the VMM administrator to add additional gateways if necessary. For VM networks that are not isolated or that leverage VLANs/PVLANs for traffic isolation, it is important to check that the correct gateway IP address is being assigned to the adapter and that the VM is able to successfully connect to that gateway.

Assuming that the network is available and that the gateway device or service is running, the most common reasons for lack of connectivity is an incorrect or invalid gateway IP address on the network adapter or conflict issues when multiple gateways and therefore multiple routes are available to the VM. As noted previously, it is relatively simple to validate that the gateway IP address for each network adapter is correct for the selected VM network, making any necessary changes either directly on the virtual network adapter (if a single machine is impacted) or to the gateway settings in the IP address pool or on an external DHCP server (where multiple machines have the same issue).

When multiple gateways are available to a VM, as is often the case when that VM has multiple network adapters, the steps for identifying and resolving connectivity issues are a little more involved. Until recently, the recommended way to address this issue (generally referred to as the multi-homed computer problem) was to define and use only one gateway and to ensure that, no matter how many network adapters were included, only one of these was the gateway specified with the setting left blank on all others. This approach works well until there is real need to use multiple gateways to access different networks or to provide alternatives if the primary goes offline or if you want to use VM networks configured for network virtualization. In the latter case, the default gateway setting is automatically populated on all network adapters connecting to that network and there is no way to remove it.

The solution to both of these situations is to specify a metric (or route cost) for each gateway, as shown in Figure 8-2. When the VM has connections to multiple VM networks, each of which with its own gateway, the gateway offering the lowest metric (or cost) to external resources becomes the default, with any other gateways checked in cost order, moving from lowest to highest, if the default gateway is unable to offer the VM a route to the target destination.

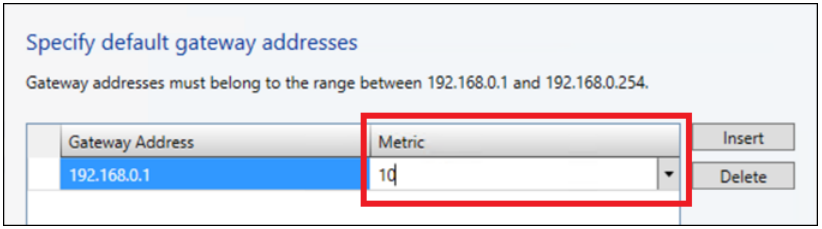


FIGURE 8-2 Configuring route metrics to avoid issues with multi-homed computers

This approach requires careful planning and a clear understanding of which VM networks a given VM might connect to. With that said, it is rare to find VMs that need to connect to more than two or three VM networks at the same time, so it should not be as complex as it might first appear to architect a solution based on this approach. As a general note to best practice, especially for service providers that want to offer connectivity to shared or common services, you should consider making the gateway to the tenants' network the lowest cost since most of the traffic they generate is likely to be routed (via their VM network gateway) back to their home network.

To this point, the focus of fault finding and diagnostics has been on verifying that the Hyper-V host and guest VMs have been configured correctly, that IP addresses, subnet masks, and gateway settings are correct and valid, and that there appears to be no reason why, based on these factors, a given VM should not be able to communicate with either local or remote network resources. Assuming that connectivity problems still exist, you can assume that the root cause of the issue lies elsewhere in the network. Therefore, the attention from here onward moves from checking configuration settings to looking specifically at issues with connectivity, with a specific focus later in the chapter on troubleshooting issues that involve the Hyper-V Network Virtualization gateway.

NOTE **In terms of a methodical process, you should consider first testing connectivity to network resources that exist on local, directly attached networks, moving on to resources that exist on external networks only after you have confirmed access to local resources is working as expected.**

A variety of network diagnostic tools are available to help troubleshoot connectivity, with more details available in the article titled, "New Networking Diagnostics with PowerShell in Windows Server 2012 R2" (see *http://blogs.technet.com/b/networking/archive/2013/07/31/new-networking-diagnostics-with-powershell-in-windows-server-r2.aspx*) with the output from the Test-VMNetworkAdapter providing a useful starting point given that it allows you to test and obtain insight into connectivity issues that might exist between a specific network adapter and sending/receiving packets to and from local and external network resources. Since service providers generally cannot log on to the tenants' VMs this tool is especially useful because it injects Internet Control Message Protocol (ICMP) packets on the port that the VM is attached

to and awaits an echo response from the receiver. If this command succeeds, this can indicate that the problem lies within the VM and not in the tenant network itself.

Step 4: Check Hyper-V Network Virtualization gateway settings

This final section assumes basic connectivity is working as expected but that challenges remain when trying to use the Hyper-V Network Virtualization gateway to establish communication between VM networks using network virtualization and external, non-virtualized networks.

Confirm that the right gateway is being used

The first thing to confirm is that a Hyper-V Network Virtualization gateway has actually been associated with the selected VM network. Unless this is set up correctly, VMs on that network will be isolated with no way to communicate with external network resources. You can use the Get-SCVMNetwork command, included in the VMM Windows PowerShell extensions, to obtain detailed information about a particular VM network, including the isolation type (see Chapter 2 for more details), the logical network to which it is linked, and, for the purpose of troubleshooting external communication, the friendly name of the Network Virtualization gateway

```
PS C:\> get-scvmnetwork -name "test" | fl name, LogicalNetwork, VMSubnet,
VMNetworkGateways, VPNConnections, NATConnections, RoutingDomainID, IsolationType,
HasGatewayConnection
```

In the following example output, the Test VM network has been configured to use the Test_Gateway.

```
Name                 : Test
LogicalNetwork       : Tenant Virtualized
VMSubnet             : {Test}
VMNetworkGateways    : {Test_Gateway}
VPNConnections       : {}
NATConnections       : {Test_NatConnection}
RoutingDomainId      : ec0db49f-6fc1-4a75-9905-e08c8dfa0a25
IsolationType        : WindowsNetworkVirtualization
HasGatewayConnection : True
```

Assuming that a gateway has been associated with the VM network, the next step is to use the friendly name to identify the VM that hosts the Network Virtualization gateway and the IP address of the gateway, both of which are essential for the troubleshooting steps that follow. You can do this by using the Get-SCVMNetworkGateway command, as shown next.

```
PS C:\> $vmnet = get-scvmnetwork -name "test"
PS C:\> get-scvmnetworkgateway -vmnetwork $vmnet | fl name, IPv4Address, IPv4Subnet,
IPAddresses, IPSubnets, EnableBGP, EffectiveRoutes, VPNConnections, NetworkGateway,
BGPPeers, VMNetwork

Name            : Test_Gateway
IPv4Address     : 10.254.254.2
IPv4Subnet      : 10.254.254.0/29
IPAddresses     : {10.254.254.2}
IPSubnets       : {10.254.254.0/29}
EnableBGP       : False
EffectiveRoutes : {}
VPNConnections  : {}
NetworkGateway  : txtstgwclu01
BGPPeers        : {}
VMNetwork       : Test
```

If VMs on the selected VM network are able to ping the address, it confirms that the gateway VM is available, that the gateway service on that machine is currently running, and that the routing domain (or compartment) for the tenant has been successfully created within the gateway. If external resources on the customer's own network are also able to ping the address, this confirms that the connection from the external network has been successfully established. Failure to receive a response from one side or the other suggests a failure in connection or that the gateway VM or the gateway service running on that machine is unavailable.

NOTE **Rather than accepting a VM network name, the Get-SCVMNetworkGateway Windows PowerShell command requires a VM network object. You can obtain this using Get-SCVMNetwork as shown in the example. You can find more information on Get_SCNetworkGateway, expected parameters, and what it returns at** *https://technet.microsoft.com/en-us/library/jj654382(v=sc.20).aspx.*

Assuming there is no response when you ping the address, the first thing to check is that the VM is actually up and running and, just as importantly, that the RemoteAccess service is actually running on that VM. To verify this, you can use the Get-Service command with the gateway name as a parameter, as shown below. The output should indicate that the service has been found and is running successfully. If no results are returned, the VM or the gateway service itself might be unavailable.

```
PS C:\Users\nigelc> get-service -computername "txtstgwclu01" remoteaccess

Status   Name               DisplayName
------   ----               -----------
Running  remoteaccess       Routing and Remote Access
```

The next step is to determine which routing domain has been allocated to the tenant and, from there, to verify that this routing domain is actually registered within the selected gateway. The starting point for this is the Get-SCVMNetwork Windows PowerShell command mentioned previously since, in addition to providing details about the gateway associated with the VM network, this command also returns the routing domain ID, as shown next.

```
PS C:\> get-scvmnetwork -name "test" | fl name, LogicalNetwork, VMSubnet,
VMNetworkGateways, VPNConnections, NATConnections, RoutingDomainID, IsolationType,
HasGatewayConnection

Name                  : Test
LogicalNetwork        : Tenant Virtualized
VMSubnet              : {Test}
VMNetworkGateways     : {Test_Gateway}
VPNConnections        : {}
NATConnections        : {Test_NatConnection}
RoutingDomainId       : ec0db49f-6fc1-4a75-9905-e08c8dfa0a25
IsolationType         : WindowsNetworkVirtualization
HasGatewayConnection  : True
```

In VM networks isolated using network virtualization, the routing domain represents the boundary or extent of the tenant's network (as discussed in Chapter 2), with the gateway providing the means for VMs operating within that boundary to connect with external resources. Assuming that the gateway is available, you can use this information to find the routing domain (also known as a compartment) in the gateway by using the Windows PowerShell command Get-NetCompartment, filtered to the selected routing domain (since there may be more than one within any given gateway) as shown in Figure 8-16. This command provides some useful information about the gateway, including the description (shown as CompartmentDescription in the output in Figure 8-16), which will be needed in further troubleshooting steps.

```
PS C:\> get-netcompartment -cimsession "txtstgwclu01" | where {$_.CompartmentGuid -match
"ec0db49f-6fc1-4a75-9905-e08c8dfa0a25"} | fl *

PSShowComputerName     : True
CompartmentType        : RoutingDomain
Caption                :
Description            :
ElementName            :
InstanceID             : =55;
CompartmentDescription : Testec0db49f-6fc1-4a75-9905-e08c8dfa0a25
CompartmentGuid        : {ec0db49f-6fc1-4a75-9905-e08c8dfa0a25}
CompartmentId          : 3
PSComputerName         : txtstgwclu01
```

Verify configuration for each type of gateway in use

To this point, you have confirmed that the VM hosting the gateway is available, that the gateway service on that machine is running, and that the tenant's routing domain is successfully registered in the gateway as a compartment. Assuming communication challenges still exist, the next step is to confirm the type of gateway being used by that tenant, either site-to-site, network address translation (NAT), or direct access, and to follow the specific troubleshooting steps for that type of gateway as described in the following sections.

Site-to-site gateway As discussed in Chapter 5, a site-to-site gateway (or VPN tunnel) between on-premises and the cloud supports only level 3 connectivity, which requires the on-premises and VM network workloads to be in different subnets. Traffic between the two networks is routed over the tunnel, for which appropriate routes must be configured on the gateway at both ends.

When dealing with issues with a site-to-site gateway, the first things to check are that site-to-site capability has in fact been enabled for the selected tenant routing domain using Get-RemoteAccessRoutingDomain as shown next (limited set output reproduced):

```
PS C:\ > get-remoteaccessroutingdomain -cimsession "txtstgwclu01" -name "Testec0db49f-
6fc1-4a75-9905-e08c8dfa0a25"

RoutingDomain                       : Testec0db49f-6fc1-4a75-9905-e08c8dfa0a25
RoutingDomainID                     : {EC0DB49F-6FC1-4A75-9905-E08C8DFA0A25}
RoutingDomainStatus                 : Enabled and Available
VpnStatus                           : Enabled
VpnS2SStatus                        : Enabled
RoutingStatus                       : Enabled
EnableQoS                           : Enabled
TxBandwidthKbps                     : 5120
RxBandwidthKbps                     : 5120
IPRange                             :
IPv6Prefix                          :
IdleDisconnect(s)                   : 300
SaRenegotiationDataSize(KB)         : 33553408
SALifeTime(s)                       : 28800
InterimAccounting(s)                :
AuthenticationTransformConstant     : SHA196
CipherTransformConstant             : AES256
CustomPolicy                        : True
DHGroup                             : Group2
EncryptionMethod                    : AES256
IntegrityCheckMethod                : SHA1
```

In this specific case, remote access VPN is also enabled (as denoted by the VpnStatus field). Border gateway protocol (BGP) is always enabled by default for a routing domain (as denoted

by the RoutingStatus field), but BGP settings must be explicitly configured for automatic routing over BGP to take effect.

> **NOTE** If you receive a command not found error when running Get-RemoteAccessRoutingDomain, this means that the remote access Windows PowerShell cmdlets are not available. You must import these by first using the Add-WindowsFeature –name "RSAT-RemoteAccessMgmt" to import the management tools and then IPMO RemoteAccess to import the Windows PowerShell module.

Having confirmed that the gateway is capable of providing site-to-site connectivity, the next thing to check is whether a site-to-site interface has been added. You can confirm this using the Get-VPNS2SInterface Windows PowerShell command as shown in Figure 8-18. In this example, a site-to-site interface is present for the selected routing domain, but a connection hasn't been established yet, as denoted by the value of the ConnectionState field.

```
PS C:\ > get-VpnS2SInterface -cimsession "txtstgwclu01" -name "Testec0db49f-6fc1-4a75-
9905-e08c8dfa0a25" | fl *

PSShowComputerName                  : True
IPv4TriggerFilterAction             :
IPv6TriggerFilterAction             :
Certificate                         : Enabled
AuthenticatopmTransformConstants    : SHA196
CipherTransformContents             : AES256
DHGroup                             : Group2
EncryptionMethod                    : AES256
IntegrityCheckMethod                : SHA1
PfsGroup                            : PFS2048
EnableQoS                           : Disabled
AdminStatus                         : True
AuthenticationMethod                : PSKOnly
ConnectionState                     : Connecting
Destination                         : [Destination IP Address]
EapMethod                           :
IdleDisconnect                      : 0
InitiateConfigPayload               : False
InterfaceType                       : FullRouter
```

When the connection is made, the correct routing must be in place so that traffic can be successfully routed between on-premises and a cloud virtual network. The routing could be static or via BGP. Static routes configured are present in the IPv4Subnet and IPv6Subnet fields that are output from the get-VPNS2SInterface command. These routes are plumbed even before the connection is established and trigger the site-to-site connection (for example, traffic with destinations matching the routes trigger the connection to be established). Note that routes can also be configured to be plumbed after a connection is established, and these are indicated in the PostConnectionIPv4Subnet and PostConnectionIPv6Subnet fields.

If dynamic routing through BGP is configured for the selected site-to-site gateway, you also need to check that the BGP configuration is correct. Although the connection might have come up successfully, communication between resources on the selected VM network and any external networks can fail if the routing information is invalid. You can confirm BGP status for the routing domain using the Get-BGPRouter Windows PowerShell command, as shown next:

```
PS C:\ > Get-BGPRouter -cimsession "txtstgwclu01" -RoutingDomain "Testec0db49f-6fc1-
4a75-9905-e08c8dfa0a25"
*
RoutingDomain             : Testec0db49f-6fc1-4a75-9905-e08c8dfa0a25
BGPIdentifier             : 10.254.254.2
LocalASN                  : 100
CompareMEDAcrossASN       : False
DefaultGatewayRouting     : True
IPv6Routing               : Disabled
LocalIPv6Address          :
PeerName                  : {172.23.10.100400}
PolicyName                :
PSComputerName            : txtstgwclu01
```

Having confirmed that BGP routing is configured for the routing domain (the command will fail if BGP is not configured), you can use the Get-BGPPeer command, shown below, to confirm that the gateway's peer (the end of the connection in the customer's site) has come up and is ready and able to connect. In this example, BGP has been configured for the selected routing domain. Its peering point is also known, but the peering itself has not come up yet because peering happens over the site-to-site tunnel, which hasn't been established yet.

```
PS C:\> Get-BGPPeer -cimsession "txtstgwclu01" -RoutingDomain "Testec0db49f-6fc1-4a75-
9905-e08c8dfa0a25"

PeerName          Local IP Address   PeerIP Address   Peer ASN   Operation Mode   ConnectivityStatus
PSComputer Name
--------          ----------------   --------------   --------   --------------   ------------------   ---------
------
172.23.10.100400  10.254.254.2       172.23.10.100    400        Mixed            Connecting
Txtstgwclu01
```

NAT-enabled gateway Applications and VMs running in the customer's virtualized network can connect to external sites through a NAT gateway -instead of through the site-to-site connection. Resources on external networks can also be permitted to connect with VMs and services on the VM network with the internal IP addressing scheme completely hidden from the external resource via address translation.

As you would expect, the steps required to troubleshoot and diagnose problems on a NAT-enabled gateway differ from those required for a site-to-site gateway. A sequence of Windows PowerShell commands can be used to quickly obtain details of the NAT configuration and the set of rules that have been set for a particular gateway:

```
PS C:\Users\nigelc> $vmnetwork = get-scvmnetwork test
PS C:\Users\nigelc> $vmnetworkgateway = get-SCVMNetworkGateway -VMNetwork $vmnetwork
PS C:\Users\nigelc> $nat = get-scnatconnection -VMNetworkGateway $vmnetworkgateway
PS C:\Users\nigelc> get-scnatrule -natconnection $nat
```

The output of the final command in this sequence, Get-SCNATRule, provides details of the external IP address used and presented to the external network in addition to all of the inbound NAT rules that have been defined for the selected VM network.

In the following example output, the NAT gateway has two rules. The first, called Test, is in an inbound rule that indicates that external devices communicating with IP address 3.3.254.89 and port 4001 will, in essence, be communicating with the VM with the internal address 192.168.0.2 and port 3389. The second rule is essentially a default outbound rule for the gateway, meaning that the IP address of VMs on the VM network will appear as 3.3.254.89 to all external resources they are connected to.

```
Name              : Test
ExternalIPAddress : 3.3.254.89
ExternalPort      : 4001
InternalIPAddress : 192.168.0.2
InternalPort      : 3389
Protocol          : TCP
NATConnection     : Test_NatConnection
ServerConnection  :
Microsoft.SystemCenter.VirtualMachineManager.Remoting.ServerConnection
ID                : 0b0243d4-164b-4b52-b2c5-a753701b2254
IsViewOnly        : False
ObjectType        : NATRule
MarkedForDeletion : False
IsFullyCached     : True

Name              : 3.3.254.89
ExternalIPAddress : 3.3.254.89
ExternalPort      : 0
InternalIPAddress :
InternalPort      :
Protocol          : TCP
NATConnection     : Test_NatConnection
ServerConnection  :
Microsoft.SystemCenter.VirtualMachineManager.Remoting.ServerConnection
ID                : d7405304-63b3-4ecf-af90-b883ba39e16e
IsViewOnly        : False
ObjectType        : NATRule
MarkedForDeletion : False
IsFullyCached     : True
```

Having confirmed that the configured NAT rules are valid, you should attempt to ping the external address to confirm that it is reachable from resources that reside on external networks. If no response is received, this suggests that the gateway VM is offline, that the gateway service has stopped, or that there is some other fundamental communication issue.

The Get-NetNat command can be used to obtain more information about the NAT gateway, including whether it is actually clustered or hosted in a single instance. The Active property is also particularly important in this context, given that a value of False suggests a configuration failure or that NAT is not in use on this specific gateway.

One further Windows PowerShell command that can be useful when troubleshooting issues with NAT gateways is Get-NetNATExternalAddress, which, when run on the gateway VM, allows you to compare the external IP that is used for NAT on the gateway with what VMM has registered. There should be one entry in the output per IP address used on the gateway.

Finally, you should check that it is possible for the gateway VMs to reach the tenant routing domains (compartments) within the gateway by using the ping –c command, which essentially allows you to perform a ping from a compartment on the gateway. Note that if you are using clustered gateways, you must use the ping –c command on the node that is the owner of the Hyper-V Network Virtualization gateway role in the gateway clusters.

Layer 3 gateway (direct route) Computers can exchange network traffic with a VM by using a CA within the virtual network. network virtualization uses provider routes to direct network traffic on the physical network. Remember that with direct routing only one VM network can be assigned to a network service.

The following sample script determines which VM networks are set to direct routing. This can be useful in cases where you have many VM networks and are not sure which one has direct routing enabled.

```
$outfile = "c:\log\vmnetworks.txt"
$vmns = get-scvmnetwork
foreach ($vmn in $vmns)
{if ($vmn.VMNetworkGateways[0] -ne $null -and $vmn.natconnections[0] -eq $null)
   {$vmn.name |out-file -width 500 -filepath $outfile
   $VmNetworkGateway = Get-SCVMNetworkGateway -Name $vmn.VMNetworkGateways[0] -vmnetwork
$vmn
   }
}
```

Step 5: Perform a network packet analysis

If you have reached this point and are still experiencing communications issues with the gateway, a more detailed analysis of network packets is required to identify and diagnose the root cause of the problem. A comprehensive walkthrough of the tools and techniques that can be used to perform such an analysis is available in the blog post at *http://blogs.technet.com/b/networking/archive/2015/03/26/troubleshooting-site-to-site-s2s-vpn-connections-on-hnv-gateway.aspx*.

CHAPTER 9

Cloud Platform System network architecture

The Microsoft Cloud Platform System (CPS), released to the market in 2014, is an Azure-consistent "cloud-in-a-box" that runs in your datacenter. CPS is based on Windows Server, Hyper-V technology, and System Center—the very same set of technologies discussed in the preceding chapters. The design decisions and architecture choices that went into building out the network architecture for CPS are interesting and directly relevant to organizations that want the flexibility to design and build their own virtualized network solutions.

This chapter moves away from a general discussion of virtualized network features and the process and steps that you should follow to create a customized solution and instead focuses on the physical and virtual network architecture within CPS and the reasons and motivations behind some of the design decisions made by the engineering team.

This chapter will:

- Introduce CPS and the different features that make up the solution architecture
- Describe the network topology, essentially how reliable and scalable connectivity is established between different racks within a CPS stamp
- Review the physical and logical networks that exist within a rack, the reasons why each one of those networks exist, and what they are used for
- Explain how CPS provides cloud services and tenant virtual machines (VMs) with access to resources on external networks
- Discuss how physical servers within the rack are physically connected to the network infrastructure

Introduction

In general, service providers and IT departments that act as service providers to their internal customers want to build cost-effective, flexible, and highly scalable cloud solutions. However, selecting and integrating the right hardware, installing and configuring the software, and optimizing the solution for specific performance, scale, and reliability goals takes considerable time and effort. Given this complexity, it's small wonder that these projects often fail to deliver all of the promised benefits or take longer to deploy and provide value than originally anticipated.

CPS is a joint effort between Microsoft and its hardware partner, Dell, to deliver a pre-configured combination of hardware and software that reduces the complexity and risk of implementing a cloud. When customers deploy CPS, they begin with a reliable, scalable, and manageable cloud fabric on top of approved hardware and can very quickly begin to build out (and deliver) the cloud services they plan to offer to their customers on top of that fabric.

> ***See also*** *More detailed information on CPS and its capabilities can be found at http://www.microsoft.com/en-us/server-cloud/products/cloud-platform-system/.*

Solution architecture

CPS is composed of standard Microsoft software, Windows Server 2012 R2, System Center 2012 R2, and Windows Azure Pack combined with standard off-the-shelf hardware from Dell (see Figure 9-1) optimized for the provision of cloud infrastructure as a service (IaaS) and platform as a service (PaaS). The overall design and configuration of the system is based on learnings and experiences derived from delivering services to customers on the Microsoft Azure public cloud.

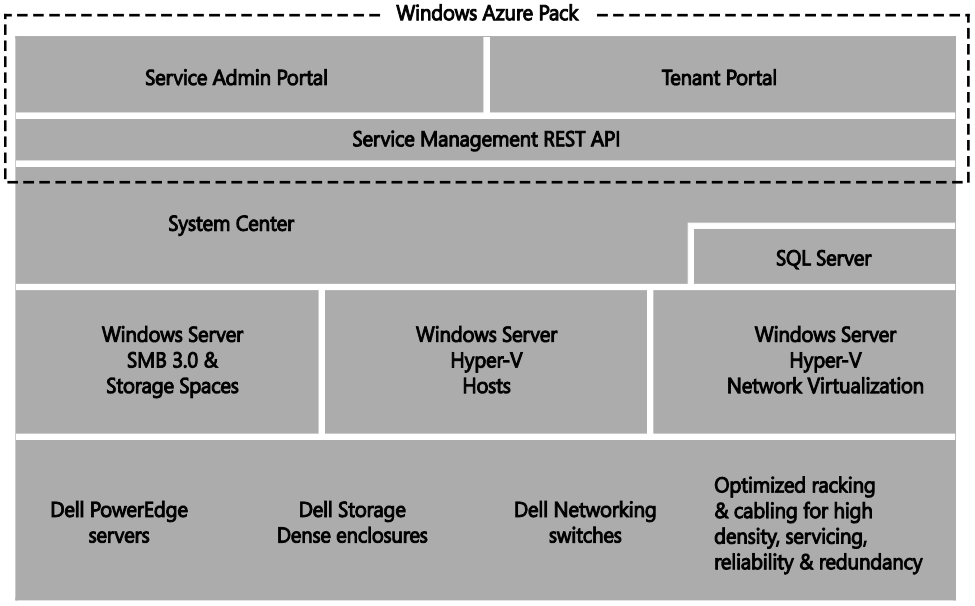


FIGURE 9-1 CPS logical architecture

In common with all cloud solutions that leverage Windows Azure Pack, CPS comes with two portals: the Service Administration Portal and the Tenant Portal. The Service Administration Portal allows administrators to configure and manage the cloud services and resources that are to be made available to tenants. The tenants are able to gain access to and consume those services through the Tenant Portal, essentially a self-service experience consistent with Azure

that allows them to manage their accounts, subscribe to service plans, and view their resource utilization.

The Tenant and Service Administration Portals are built on a REST-based API, known as the Service Management API, which enables providers who do not want to use the provided portals to build out their own user experience. Advanced users are also able to leverage this API to integrate their own applications and services with CPS resources.

Management capabilities required by the service provider, such as deployment of new resources, system configuration, monitoring, and automation, are provided by Microsoft System Center, with overall system configuration stored in SQL Server 2012 R2.

From a tenant networking perspective, VMs that are provisioned through IaaS are created on isolated virtual networks using Windows Server Hyper-V Network Virtualization. This allows tenants to create networks on demand without providers having to reprogram the physical networks. This isolation provides data channel isolation at the network layer and creates flexibility for tenants and administrators who don't have to worry about things such as overlapping IP addresses, conflicting machines, or configuration errors when working with physical networking.

CPS implements software-defined storage virtualization technology delivered through Windows Scale-Out File Server, Windows Server SMB 3.0, and Storage Spaces. All of the Windows Server Hyper-V hosts place their VMs on an SMB-backed share that is built on top of storage pools created from the underlying physical storage.

In terms of the physical features, Dell PowerEdge servers are used to provide compute resource, Dell PowerVault Dense Enclosures are used to house the just-a-bunch-of-disk (JBOD) configuration for system storage, and Dell Force10 Networking switches make up the networking layer.

CPS ships in units referred to as stamps. A stamp consists of at least one rack, up to a maximum of four (as shown in Figure 9-2) with each rack containing aggregation and access switches required for connectivity, a compute cluster, a storage cluster, and an optional (edge) cluster. The first rack in the stamp contains the management cluster. As you would expect, it is possible to start with a single rack and scale up to the maximum of four, which allows the service provider to expand the pool of compute, storage, and network resources within the stamp to meet demand.

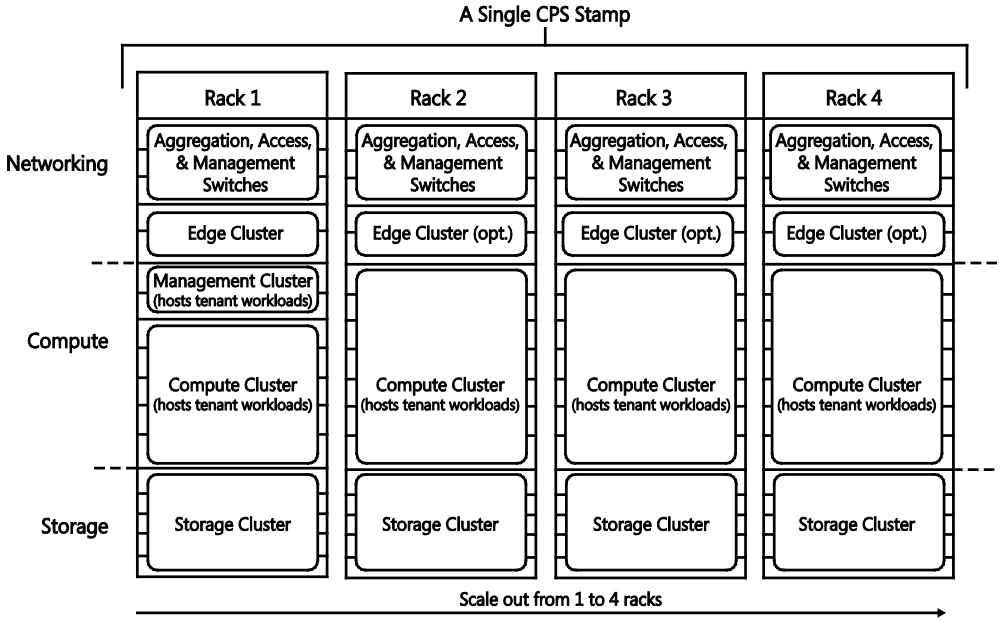


FIGURE 9-2 Features in a CPS stamp

From a service provider's perspective, the sole purpose of cloud infrastructure (otherwise known as cloud fabric) is to provide manageable, scalable, and reliable capacity to support the cloud services that are to be offered to tenants. To support this basic requirement, physical servers with each CPS rack are organized into a number of failover clusters.

- **Edge cluster** This provides network edge services, facilitating tenant and service provider access to resources and networks that are external to the stamp. It provides software-based network edge services such as the Hyper-V Network Virtualization (HNV) forwarding gateway, Network Address Translation (NAT) gateway, and site-to-site VPN gateway discussed in Chapter 5, "Network Virtualization gateway." The edge cluster is essentially a single, independent two-node cluster per rack, which resides in the same domain as all of the other management elements within the stamp. It hosts multi-tenant gateways to the "external" network and forwarding gateways each of which are connected to their own front end with the latter isolated from the rest of the stamp through physical switch ACLs. Note that the edge cluster contains only gateway VMs; no tenant or other management VMs are present on this cluster.
- **Management cluster** This houses all of the management resources for the stamp, as well as the VMs required for the Tenant Portal. Management in this context includes System Center services—Virtual Machine Manager (VMM) and Operations Manager—as well as infrastructure services such as Windows Server Update Services (WSUS) and Windows Deployment Services (WDS), DNS, Active Directory, and DHCP.

- **Compute cluster** This houses all tenant services utilizing Hyper-V virtualization—including the IaaS services and PaaS services provided as part of the basic CPS installation—together with any additional services added by the service provider.
- **Storage cluster** This provides continuously available storage services through a set of storage nodes combined into a single failover cluster. A single storage cluster can provide storage services to any other cluster within the stamp.

As can be seen in Figure 9-2, clusters within CPS do not span across multiple racks, and, hence, if the stamp contains four racks, there will be four separate compute clusters, four storage clusters, and, potentially, four edge clusters. VMM is used to aggregate all of these disparate resources into private clouds, hiding the complexity of the physical implementation and infrastructure from the services that need to make use of them. (For more details on creating a private cloud in VMM, see *http://technet.microsoft.com/en-us/library/gg610625.aspx*.)

A key feature of any cloud infrastructure is its resiliency to failure and its ability to provide an acceptable level of service to tenants even if such a failure occurs. These characteristics are generally achieved through a combination of architectural design/feature choice and the deployment and use of redundant (spare) capacity.

CPS is certainly no different in this regard. The solution is divided into racks (fault zones), and where possible, active-active infrastructure is used within the rack to enable full utilization of redundant resources and to facilitate faster failover if a fault occurs. Failure of any one feature in the architecture is designed to cause minimal loss in capability. Indeed, for many types of faults, two failures can be accommodated (N+2 fault tolerance) without causing a service down event.

To summarize, service providers deploying CPS begin with highly managed, scalable, and reliable cloud infrastructure, which supplies the compute, storage, and networking capacity needed by the services they want to offer to their customers. Note that support for IaaS is installed by default, but a service provider can use the Service Administration Portal to add PaaS capabilities, including support for high-density websites, queues, and databases.

A closer look at CPS network architecture

As previously discussed, selecting and integrating the right hardware for your cloud fabric, installing and configuring the software, and, finally, optimizing the solution for specific performance, scale, and reliability goals takes considerable time and effort. A complete and preconfigured solution, CPS minimizes the complexity and risk associated with such a project and significantly reduces the critical time to value—literally, the time required from the decision to proceed with the project to the cloud service(s) being made available to customers.

As can be seen from the logical architecture diagram in Figure 9-1, a number of different software and hardware features were needed to build out CPS, each one optimized for the

performance, reliability, and scalability goals of the IaaS and PaaS services and tenant workloads that are expected to run on it. A complete review and discussion of the design rationale, choices, and optimizations for all of these features is clearly well beyond the scope of this book, but in the context of a discussion of the best way to design and build a virtual network solution, it is both useful and instructive to understand how the engineering team approached this problem and to consider some of the key decision points and best practices that went into the CPS design.

NOTE **Since your physical environment and the type of services and workloads you plan to offer to your tenants might differ from those targeted by CPS, it is important to view the contents of this chapter as examples and pointers to best practice rather than as a design blueprint or prescriptive guidance for your own solution.**

The chapter begins with a look at network topology, essentially how connectivity is established between and across racks within a stamp, and from there reviews the physical networks that exist within a rack, the reasons why each one of those networks exists, and what they are used for. From there, the chapter moves on to logical networks and the need for network isolation, how CPS provides cloud services and tenant VMs access to resources that exist off stamp—external networks—and concludes with a review of how servers within the stamp are physically connected to the network infrastructure.

Network topology

CPS allows service providers to add capacity to the stamp, essentially a new rack, to meet current or expected demand with zero downtime for services already running on that stamp. To facilitate this, each rack is self-contained, with an internal network used to connect all of the compute and storage resources within that rack.

The primary goal of the CPS network topology is therefore to provide fast, efficient, and reliable connectivity between resources that exist in different racks and to enable connectivity off stamp to external resources and services, a topic covered later in this chapter.

Assuming a hierarchical topology is an appropriate starting point for CPS, a hierarchy of switches (as shown in Figure 9-3) can be used to achieve connectivity between resources within and across racks in the stamp, with the higher level switch in Figure 9-3 containing higher-speed uplinks to aggregate traffic.

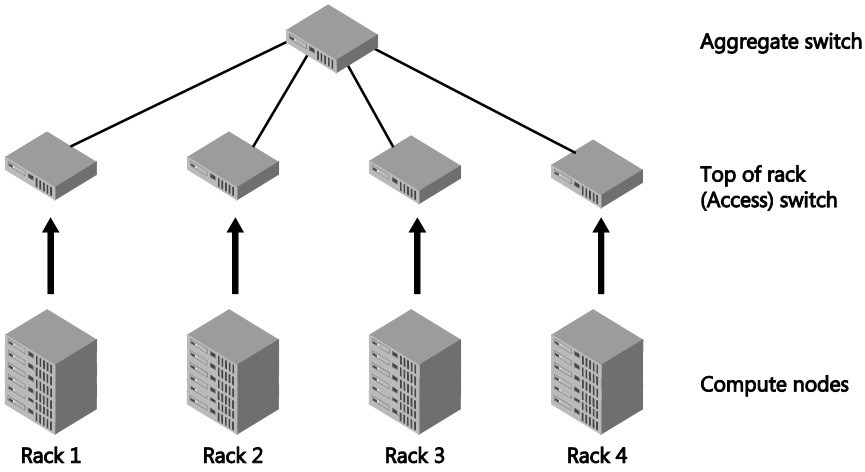


FIGURE 9-3 Using a hierarchy of switches to connect resources across the stamp

Although such an approach satisfies the scale-out requirement, allowing the service provider to add racks on demand, the use of a high-performance aggregate switch adds significant cost to the solution. Further, the network bandwidth between resources in a hierarchical topology is not guaranteed to be uniform; the bandwidth between resources located within the same rack is likely to be much higher than the worst case bandwidth available between those in different racks, and, as a result, tenant services would need to be constrained to and located within a given rack. Splitting a service (set of VMs) across multiple racks or placing a VM in one rack and its storage in another leads to reduced levels of performance.

A flat network topology based on the "CLOS" network architecture, in contrast, appears to have approximately uniform bandwidth from any one resource in the stamp to any other resource in the stamp, removing the constraints associated with the hierarchical model and allowing tenant services to be placed on and leverage compute and storage resources from any location within the stamp. Microsoft published some of the original research in this area, leveraging what is now commonly called a layer 2 flat network in Windows Azure. For more information on this topic, see *http://azure.microsoft.com/blog/2012/11/02/windows-azures-flat-network-storage-and-2012-scalability-targets/*.

A further refinement from Microsoft's experience with Windows Azure and the network topology ultimately chosen to connect racks within the CPS stamp is the concept of a routed flat network in which all layer 2 traffic remains within the rack, with network traffic destined for outside the rack routed using IP networking. An Internet draft written by Microsoft Corporation and Arista (see *http://sandbox.ietf.org/doc/draft-ietf-rtgwg-bgp-routing-large-dc/*) describes this concept in more detail.

To ensure packets destined for outside a given rack are routed efficiently, CPS leverages the Equal Cost Multipath routing protocol (ECMP), which allows multiple redundant paths across multiple switches to be used at the same time. The combination of a flat network design with

ECMP allows CPS to use simple, low cost switches at both the aggregation layer and the access layer and to use multiple switches for fault tolerance as shown in Figure 9-4. The access and aggregation switches are included in each rack and load balance/fail over between the links as necessary using ECMP.

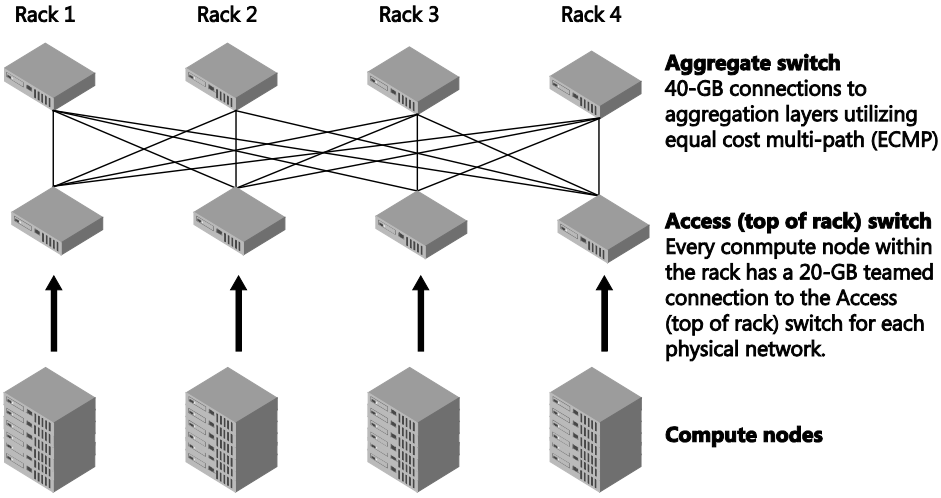


FIGURE 9-4 Flat network topology used by CPS to connect resources across the stamp

In short, all traffic destined for resources within the rack is switched using standard layer 2. Traffic that needs to exit the rack is routed through layer 3. Given that multiple possible routes exist over layer 3 to each other rack (a mesh architecture as shown in Figure 9-4), the most efficient route is selected using the ECMP routing protocol. This network architecture, combined with mechanisms in Windows Server such as Hyper-V Live Migration and Hyper-V Storage Migration provides fast, reliable connectivity and low cost fault tolerance and allows all compute and storage resources to be utilized, regardless of their physical location within the stamp.

Each new rack adds an additional set of cross-rack interconnects to each of the existing racks in the stamp, so the amount of bandwidth increases linearly with each new rack in the stamp. So not only is there zero down time when new racks are added, there is also zero performance impact to the existing workloads from the new rack.

Physical networks

As discussed in Chapter 2, "Logical networks," multiple physical networks are often used when organizations want to improve security, simplify management, or to remove potential competition between different types of network traffic. The engineering team used these same principles when deciding how many physical networks to use within CPS.

Since storage operations such as file sharing and live migration generally take up a lot of bandwidth, it makes sense to come up with a design that optimizes and isolates these

workloads from other network traffic. Although clearly the quality of service (QoS) features in Windows Server could be used to ensure that each workload on the same network is appropriately prioritized, the issue comes down to the capacity (throughput and maximum bandwidth) of the physical network and the potential to optimize the CPS architecture to ensure that the maximum possible bandwidth is available to each type of operation. The engineering team also decided early on to use RDMA to achieve optimal SMB storage performance, recognizing that in Windows Server 2012 R2, RDMA does not operate over a Hyper-V virtual switch, a constraint that really drove the separation of tenant workloads and datacenter workloads onto different physical networks.

As shown in Figure 9-5, CPS has two primary physical networks: the Datacenter network dedicated to providing maximum throughput for storage operations and the Tenant network, which carries all other network traffic, including that generated by tenant VMs, service provider (shared) services, and stamp management. To isolate device management from other workloads and as a means to ensure administrators can gain access to the stamp in the event of a network failure or configuration issue on either of the primary networks, a third network, the Management network, was introduced to provide direct access to switches and the baseboard management controller (BMC) in each node.

NOTE **Because there are three physical networks in each rack but only two switches, logical separation is configured on each switch –with individual ports within the switch dedicated to and configured to support a specific network within the rack.**

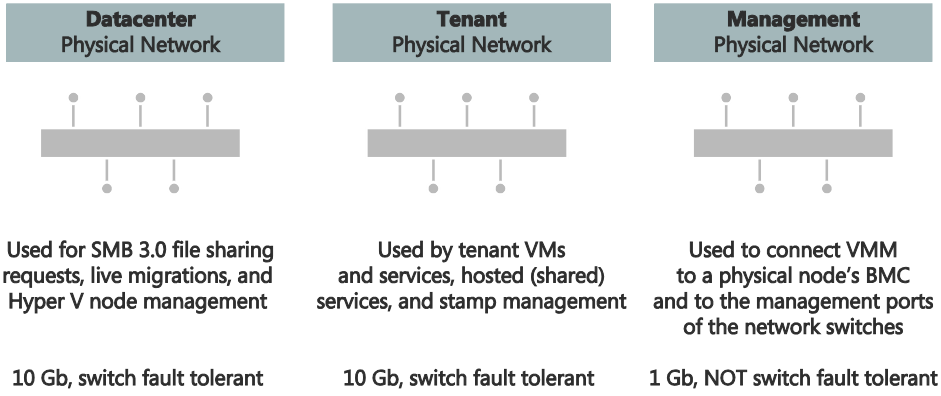


FIGURE 9-5 The three physical networks present in a CPS rack

Compute and storage nodes have a total of five network interfaces to provide connectivity to each of the three physical networks. Four of the interfaces in each node are capable of 10-gigabit (Gb) Ethernet, two of which support RDMA and are connected to the Datacenter network, with the remaining two 10-Gb Ethernet interfaces combined into a NIC team and connected to the Tenant network. The remaining interface is used for management and is capable of 1 Gb only.

Datacenter network

In line with the architecture recommendations related to building a converged datacenter with file server storage, CPS uses a dedicated network for storage traffic with this network configured to host file sharing requests, VM live migration, and server backup operations. (See *https://technet.microsoft.com/en-us/library/hh831738.aspx* for more details.)

Apart from providing a degree of isolation for the purposes of security and ease of management, the use of a dedicated network that separates storage traffic, cluster heartbeat, all management access to the host, and even PXE booting (if a host is re-imaged during field replacement) from VM traffic ensures that these workloads do not compete with each other. One of the other benefits of a separate network is that it allowed the CPS design team to optimize around access to storage, leading to the introduction and use of RDMA network adapters in both storage and compute nodes (as discussed earlier) and the use of the SMB Direct protocol and SMB multichannel features introduced in Windows Server 2012, both of which are designed to optimize data transfer to and from shared storage when combined with RDMA adapters.

In short, the creation of a dedicated datacenter network and the deployment and use of RDMA network adapters (combined with SMB Direct and SMB Multichannel) in CPS led to increased throughput for storage related requests. RDMA network adapters coordinate the transfer of large amounts of data at line speed, with low latency. They provide extremely fast responses to network requests (remote file storage responsiveness can often appear similar to directly attached block storage) and lower CPU utilization with fewer CPU cycles required to transfer data over the network. This approach also provides fault tolerance for CPS storage without the need to team RDMA network adapters (which is currently not supported in Windows Server 2012 R2) given that both SMB v3 for file sharing and SMB Direct for RDMA data access are able to use multiple network paths to reach a destination simultaneously.

NOTE **As you would expect, design choices for one part of a solution can often have implications for others In the case of the storage network, the decision to build a network topology that includes a layer 3 routed element meant that only RDMA network adapters that support the iWARP standard could be used in CPS given that all of the other standards—InfiniBand, and RoCE v1—do not support routing between RDMA endpoints.**

See also *You can find more details on how the use of RDMA network adapters, SMB Direct and SMB Multichannel, can improve file server performance at https://technet.microsoft.com/en-us/library/jj134210.aspx.*

Tenant network

The Tenant network, as the name suggests, is used by all VMs running within the stamp, with logical networks (to be discussed later in the chapter) used to separate traffic that is destined for resources external to the stamp, traffic that needs to be routed to a hardware load balancer, traffic between a tenant's VMs or the Hyper-V Network Virtualization gateway, traffic

created as part of a PaaS offering, and traffic from the Management network to the management cluster.

Traffic isolation is a key requirement for the tenant network given that it is used for a number of different purposes and the fact that multiple tenants (customers) will also be using it. Clearly, no tenant should be able to see another tenant's network traffic or be able to view or access the traffic created to manage the fabric resources that make up the CPS infrastructure. As discussed in Chapter 2, a variety of technologies can be used to provide traffic isolation including virtual local area networks (VLAN), private virtual local area networks (PVLAN), and Network Virtualization, but CPS actually uses a combination of different approaches. Network Virtualization is used to isolate tenants from one another, VLANs are used for access isolation or congestion isolation between any logical networks that share the same physical network, and physical switch ACLs are used for layer 3 isolation, as follows:

- Isolation of BMC and switch management port traffic from all other traffic
- Isolation of the networks that can be subject to DOS attacks, specifically, isolation of the External network and Load Balancer network, which may be connected to the Internet
- Isolation of networks with different security models, specifically, the Network Virtualization network and the Services network.
- Isolation of the service provider's Infrastructure network from all tenant traffic, specifically, isolation of the Infrastructure network from the Network Virtualization network, External network, Load Balancer network, and Services network

The only specific hardware requirement for connecting compute and storage nodes to the tenant network is that physical adapters on the tenant network must support NVGRE task offload. This is important because that's how the CPS team is able to achieve 18 Gbps for VM-to-VM traffic and 9 Gbps for North-South traffic through the gateways. Network adapters that are connected to this network are combined in each CPS node to provide fault tolerance, bandwidth aggregation, and traffic failover—in other words, to maintain network connectivity even if the network fails between the node and the access switch).

NOTE **CPS requires Link Aggregated Control Protocol (LACP) switch dependent teaming because of its superior failover times. LACP (also known as IEEE 802.1ax) teaming specifically is required because it is easier to configure, is standards based, and enables a heartbeat mechanism that verifies link integrity beyond just verifying that a signal was received by the node.**

Storage nodes are physically connected to the Tenant network, but since there are no VMs on the storage nodes, the adapters are disabled, so this connection is effectively unused by the storage nodes. The fact that storage nodes don't host virtual machines is the primary reason why the team did not install Mellanox Network Adapters with NVGRE task offload in the storage nodes.

Management network

The third physical network in the stamp is the Management network. It connects VMM to a physical node's BMC and to the management ports of the network switches, but, unlike the other two networks, the Management network doesn't have any redundancy/it is not switch fault tolerant. Since there are no significant bandwidth requirements, from a management perspective, the link speed and available bandwidth on this network at 1 Gb is significantly lower than the other two.

Logical networks

Logical networks represent an abstraction of the underlying physical network infrastructure. They are used to organize and simplify network assignments for hosts, VMs, and services as discussed in Chapter 2. As part of logical network creation, network sites define the VLANs, IP subnets, and IP subnet/VLAN pairs that are associated with the logical network in each physical location.

The logical networks within CPS are External, Infrastructure, Load Balancer, Network Virtualization, Forwarding Gateway FrontEnd, and Services, as shown in Figure 9-6. Figure 9-6 also indicates which VMs and services connect to and use these virtual networks. The function and purpose of each logical network, how they are used, and whether they are available to tenants as part of the tenant cloud provided through Windows Azure Pack (WAP) running on the stamp are explained in Table 9-1. The "Network sites" section explains the network sites used within each logical network and their respective VLAN(s) and IP subnet ranges.

NOTE **Neither the physical Datacenter network (containing the logical Infrastructure network) nor the physical Management network (containing the logical BMC Management and Switch Management networks) connect directly to VMs. Instead, they are routed onto the physical tenant network via the infrastructure VLAN.**

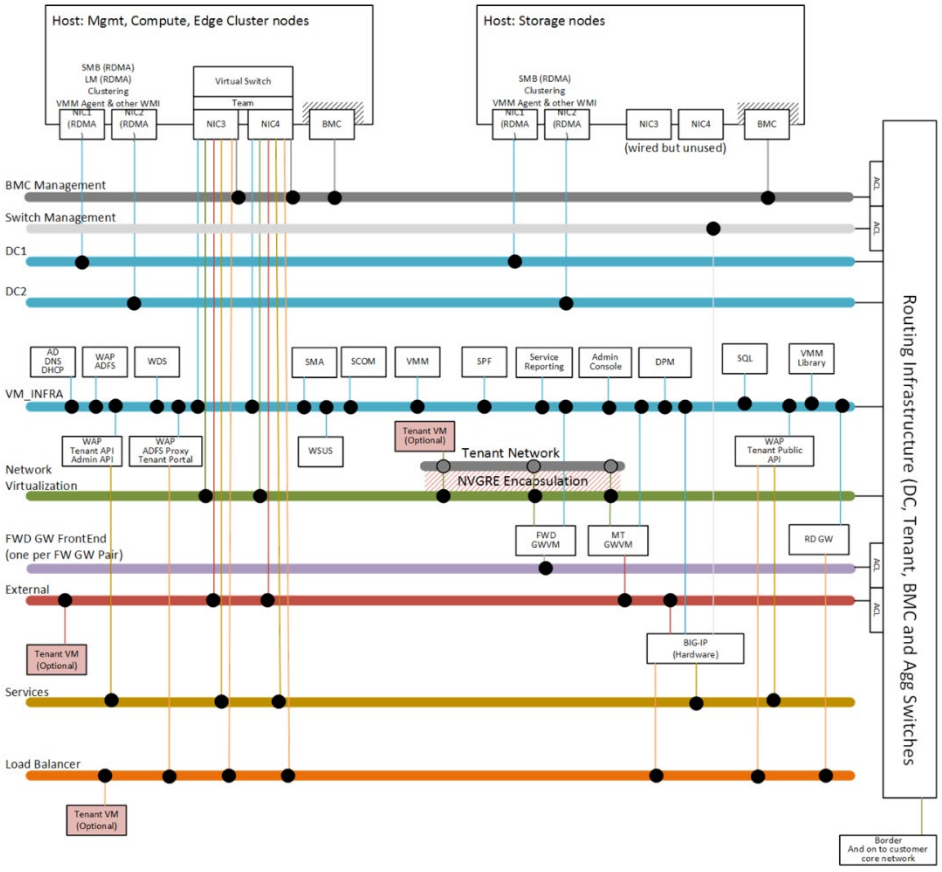


FIGURE 9-6 CPS logical network overview

TABLE 9-1 Function and purpose of each logical network

LOGICAL NETWORK NAME	DESCRIPTION	AVAILABLE TO TENANT CLOUD?
External	Used for off-stamp network connectivity, including Internet connectivity. There is no tenant isolation on the External network. From a tenant perspective, this network is used for: ■ Site-to-site endpoint IP addresses ■ Load balancer virtual IP addresses (VIPs) ■ Network address translation (NAT) IP addresses for virtual networks ■ Tenant VMs that need direct connectivity to the external network with full inbound access	Yes

LOGICAL NETWORK NAME	DESCRIPTION	AVAILABLE TO TENANT CLOUD?
Infrastructure	Used for service provider infrastructure, including host management, live migration, failover clustering, and remote storage. It cannot be accessed directly by tenants.	No
Load Balancer	Connects tenant or service provider VMs to the hardware load balancer included within CPS. A tenant or service provider service tier is connected to a hardware load balancer to provide virtual IP addresses for a set of VMs. Multiple tenants share this network—there is no tenant isolation on the Load Balancer network. From a tenant perspective, this network: ■ Uses static IP addresses ■ Has outbound access to the external network via the load balancer ■ Has inbound access that is restricted to only the ports that are exposed through the VIPs on the load balancer	Yes
Network Virtualization	Used in general for IaaS tenant-to-tenant traffic and tenant-to-off-stamp traffic (exceptions include if the Load Balancer network is used or if the service provider chooses to allow the External network to connect directly to a tenant VM). All traffic on the Network Virtualization network uses Hyper-V Network Virtualization. All traffic on this network is encapsulated using NVGRE. From a tenant perspective: ■ This network is automatically used for allocating provider addresses when a VM that is connected to a virtual network is placed onto a host. ■ Only the gateway VMs connect to this directly. ■ Tenant VMs connect to their own VM network. Each tenant's VM network is connected to the Network Virtualization logical network. ■ A tenant VM will never connect to this directly. ■ Static IP addresses are automatically assigned. Note that any load-balanced workload cannot be placed onto this network.	Yes
ScaleUnit#-Forwardingx (where # is the rack number, and x = 1 through 4)	Associated with forwarding gateways, which require one logical network per gateway. For each forwarding gateway, a logical network is associated with its respective scale unit and forwarding gateway.	No

LOGICAL NETWORK NAME	DESCRIPTION	AVAILABLE TO TENANT CLOUD?
Services	The Services network is used for connectivity between services in the stamp by public-facing Windows Azure Pack features, and for SQL Server and MySQL Database DbaaS deployments. All deployments on the Services network are behind the load balancer and accessed through a virtual IP (VIP) on the load balancer. This logical network is also designed to provide support for any service provider-owned service and is likely to be used by high-density web servers initially, but potentially many other services over time.	No

Network sites

For each of the logical networks, network sites are used to scope or otherwise limit the network to a specific group of hosts (either within a rack or across the stamp) and to define the VLANs, IP pools, and MAC address pools that are needed to make those networks available and usable on the selected hosts. Table 9-2 sets out the complete set of network sites used within CPS, the logical networks they support, and the VLAN/IP subnet ranges.

TABLE 9-2 Network sites used within CPS

NETWORK SITE NAME	LOGICAL NETWORK	HOST GROUPS	VLAN - IP SUBNET
ScaleUnit1-VMInfrastructure	Infrastructure	All Hosts	8 - 10.228.194.0/24
ScalUnit2-VMInfrastructure	Infrastructure	All Hosts	8 - 10.228.195.0/24
ScaleUnit1-External	External	Compute Clusters Edge Clusters	1001 - 10.228.204.0/24
ScaleUnit2-External	External	Compute Clusters EdgeClusters	1001 - 10.228.205.0/24
ScaleUnit1-LoadBalancer	Load Balancer	Compute Clusters EdgeClusters	2101 - 10.10.12.0/23
ScaleUnit2-LoadBalancer	Load Balancer	Compute Clusters EdgeClusters	2101 - 10.10.14.0/23
LoadBalancer-FrontEnd_0	Load Balancer FrontEnd	All Hosts	0 - 10.184.106.0/23
ScaleUnit1-NetworkVirtualization	Network Virtualization	All Hosts	2000 – 10.10.0.0/22

NETWORK SITE NAME	LOGICAL NETWORK	HOST GROUPS	VLAN - IP SUBNET
ScaleUnit2-NetworkVirtualization	Network Virtualization	All Hosts	2000 – 10.10.4.0/22
ScaleUnit2-Services	Services	Management Cluster Compute Clusters	0 – 10.10.11.0/24
ScaleUnit1-Services	Services	Management Cluster Compute Clusters	2201 – 10.10.10.0/24
ScaleUnit1-Forwarding1	ScaleUnit1-Forwarding1	Edge Clusters	21 – 10.228.192.128/29
ScaleUnit1-Forwarding2	ScaleUnit1-Forwarding2	Edge Clusters	22 – 10.228.192.136/29
ScaleUnit1-Forwarding3	ScaleUnit1-Forwarding3	Edge Clusters	23 – 10.228.192.144/29
ScaleUnit1-Forwarding4	ScaleUnit1-Forwarding4	Edge Clusters	24 – 10.228.192.152/29
ScaleUnit2-Forwarding1	ScaleUnit2-Forwarding1	Edge Clusters	21 – 10.228.192.192/29
ScaleUnit2-Forwarding2	ScaleUnit2-Forwarding2	Edge Clusters	22 – 10.228.192.200/29
ScaleUnit2-Forwarding3	ScaleUnit2-Forwarding3	Edge Clusters	23 – 10.228.192.208/29
ScaleUnit2-Forwarding4	ScaleUnit2-Forwarding4	Edge Clusters	24 – 10.228.192.216/29

Logical switches

As discussed in Chapter 4, "Logical switches," the Hyper-V virtual switch in Windows Server 2012 is a layer 2 virtual network switch that provides programmatically managed and extensible capabilities to connect VMs to the physical network. The switch is compatible with most networking features in Windows apart from RDMA, and, as a consequence, Hyper-V virtual switches are used for access and connectivity to the tenant network but not the Datacenter network.

NOTE **The Hyper-V virtual switch architecture in Windows Server 2012 is an open framework that allows third parties to add new functionality such as capture, forwarding, and filtering to the virtual switch. Extensions are implemented using Network Device Interface Specification (NDIS) filter drivers and Windows Filtering Platform (WFP) callout drivers. However, for CPS, third party extensions to the virtual switch are not supported because the end-to-end testing does not include them.**

As indicated in Chapter 8, "Diagnosing connectivity issues," the switch concept is greatly enhanced through the use of logical switches (essentially templates for Hyper-V switches) that allow you to consistently apply the same settings and configuration across multiple hosts and further to ensure that any Hyper-V switches deployed using the template remain compliant with it. As an aid to management, all of the Hyper-V virtual switches configured on compute and storage nodes were deployed using logical switches, and, hence, it is relatively easy for the CPS administrator to verify compliance as part of the regular set of management activities.

External connectivity

Each edge cluster has two physical nodes running Hyper-V. The nodes run Windows Server 2012 R2 Datacenter edition (Server Core configuration). There is one edge cluster per rack. The software gateways run as guest clusters (two VMs in each guest cluster). For each rack, there are five multitenant site-to-site/NAT gateway guest clusters and four forwarding gateway guest clusters. The gateway's two modes are used to address two different technical scenarios:

- **Site-to-site/NAT gateway** This allows VM networks to be created with external connectivity to a remote site by using a site-to-site VPN tunnel or direct connectivity through NAT. In this mode, up to 100 VM networks can share a single gateway guest cluster. Additional gateway guest clusters can be deployed to the edge cluster until its capacity (network, CPU, or memory) is reached. Site-to-site/NAT is the default configuration for a gateway guest cluster deployment.

  From a tenant perspective, a site-to-site/NAT gateway is used to connect their on-premises network to their hosted VM network. A public IP address is associated with each gateway, and tenants use that IP address as their site-to-site VPN endpoint address.

- **Forwarding gateway** This forwards traffic between a single VM network and an external network. You must configure the external network with static routes into this gateway for the subnets that the VM network contains. Each gateway guest cluster in this mode can serve only a single VM network, although that VM network can contain as many VM subnets as required from the external network. These subnets must be taken from the address space that is routable from the external network, and care must be taken to make sure the address space within this VM network does not overlap with any other parts of the external network, including any other VM networks that route onto the external network. Only the administrator can create a VM network to use a forwarding gateway.

  From a tenant perspective, a forwarding gateway is typically used to connect the VM network to resources on the corporate network.

Monitoring

As discussed in Chapter 7, "Operations." it is often useful to understand network utilization and traffic patterns. Service providers, for example, want to know and understand load patterns, peak periods, and which customers and services are generating the most traffic. Tenants (customers) want insight into and details of what their VMs and services are doing on the network to ensure that they pay only for what is necessary.

CPS is no exception, and, as Table 9-3 shows, several monitoring dashboards are provided in CPS to allow administrators to quickly see whether any issues require attention. Used together, these dashboards consolidate health state and alerting views for all major fabric features and are the best places to view CPS health, including networking.

TABLE 9-3 Monitoring dashboards provided in CPS

MONITORING DASHBOARD	WHAT IT SHOWS
CPS System Health	Health state rollup from: ■ Compute Clusters and Storage ■ Edge Cluster Two health state rollups from Management Cluster: ■ VM Health (a key underpinning of the cloud) ■ All other management features
Cloud Health Dashboard	Health state rollup of each cloud.
Fabric Health Dashboard	Detailed overview of the health of the cloud you select on the Cloud Health Dashboard and the fabric (storage, compute, and networking) that services that cloud
Edge Cluster Dashboard	Health state views for all software features and VM guests and hosts that support the edge cluster. A critical alerts view displays only alerts that relate to the edge cluster selected.
Management Cluster Dashboard	Health State views for all software features and VM guests and hosts that support the management cluster. Also, a critical alerts view displays only alerts that relate to the management cluster.
Storage Health Dashboard	Health state views for all storage features

CPS uses System Center Operations Manager to monitor cloud infrastructure. You can install the System Center Operations Manager console on any computer that meets the System Center 2012 R2 Operations Manager system requirements (*http://technet.microsoft.com /en-us/library/dn249696.aspx*). It is also recommended that you apply the latest System Center 2012 R2 Operations Manager Update Rollup. You can find more information at *http://technet.microsoft.com/library/hh298607.aspx*.

The Hyper-V Management Pack for System Center Operations Manager, installed by default, provides counters that can help by allowing service providers to monitor and gain insight into the usage of the virtual switch, physical network adapter, and virtual network adapter among others and create dashboards and reports that showcase this in a more readily consumable fashion.

System Center 2012 R2 Operations Manager also provides network monitoring of the access and aggregation switches in the stamp via simple network management protocol (SNMP). It implements basic monitoring support (device up/down and port monitoring) for devices that support the Interface MIB (IETF RFC 2863) and MIB-II (IETF RFC 1213) standards.

System Center 2012 R2 Operations Manager also supports monitoring of the CPU and memory consumption of network devices. To enable this additional capability, the original equipment manufacturer (OEM) is responsible for creating a System Center 2012 R2 Operations Manager management pack to cover CPU and memory monitoring for their particular network device. A sample is provided at *http://gallery.technet.microsoft.com/Sample-Network-Management-a061608c* . The created management pack must be verified under the System Center Marketplace validation detailed at *http://systemcenter.pinpoint.microsoft.com/en-US/partnercenter*.

About the authors

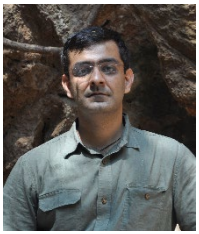
NIGEL CAIN (lead author) leads the Customer, Architecture, and Technology (CAT) team for the Microsoft Enterprise Cloud Group in Asia Pacific, Japan, and India. He and his team work closely with service providers (hosters) and enterprise customers, helping them take full advantage of Windows Server and System Center. He has a keen interest in cloud computing from both a business strategy and technical viewpoint and has presented sessions on building and managing private/hybrid clouds at a number of industry events. Nigel is the lead author of the ebook *Microsoft System Center: Building a Virtualized Network Solution* (Microsoft Press, 2013) of which the current ebook is an updated edition. Nigel graduated MBA from Warwick Business School in 2010. For more information and to connect with Nigel, see *https://www.linkedin.com/in/nigelcain*.

ALVIN MORALES (co-author) is a senior IT operations engineer at Microsoft CSS Labs and works closely with the Windows Server and System Center engineering team. His current focus is on integrating Microsoft System Center in the datacenter and private and hybrid cloud computing in service providers (hosters) and enterprise customers. He has presented sessions from an operational standpoint to help enterprise customers manage private and hybrid clouds. Alvin graduated from the University of Puerto Rico at Mayaguez Campus, and he completed his MBA in cybersecurity at the University of Dallas. For more information, see *http://www.linkedin.com/in/alvinmorales*.

MICHEL LUESCHER (co-author) is a solution architect in the worldwide Datacenter & Cloud Infrastructure Center of Excellence (CoE) at Microsoft Corporation based out of Switzerland. Primarily, Michel is focused on hybrid cloud solutions (Hyper-V, System Center, and Microsoft Azure) and works with Microsoft's enterprise customers and service providers to define and guide the new landscape and architecture. Michel joined Microsoft in 2009 and works closely with the different divisions, including the various product groups. He is a well-known virtualization specialist, presenting at different events, and his is also the publisher of books on cloud and virtualization. Internally at Microsoft, he is considered a Subject Matter Expert (SME) for datacenter and is involved in initiatives such as the Cloud OS Network (COSN). Michel also has a blog called Server Talk (*www.server-talk.eu*) where he posts technical articles about the Microsoft cloud platform. You can follow him on Twitter at @michelluescher.

DAMIAN FLYNN (co-author) is a Microsoft MVP (System Center and Datacenter) and a Cisco Champion. He is an Infrastructure Technical Architect for a large multi-national organization, and a freelance consultant focused on cloud technologies in the converged and hybrid datacenter, with perspective on service management automation for repeatable processes in dev/ops scenarios, leveraging software defined networks (SDN) and Microsoft Azure Pack. He has a keen interest in cloud computing from both a business strategy and a technical viewpoint and has presented sessions on building and managing private/hybrid clouds at a number of industry events. Damian is co-author of titles, including *Microsoft Private Cloud Computing* (Sybex), *Windows Server 2012 Hyper-V Installation and Configuration Guide* (Sybex), and *Microsoft System Center: Building a Virtualized Network Solution* (Microsoft Press, 2013). Damian is active in many technology preview programs, blogs at *www.damianflynn.com*, tweets at @damian_flynn, and has published a number of white papers, technical articles, and webinars. His motto is "Making incredible software incredibly simple."

UMA MAHESH MUDIGONDA (contributing author) is a Senior Program Manager in the Enterprise Cloud Group at Microsoft India Development Center, Hyderabad. His areas of expertise include cloud computing, software defined networking (SDN), routing, virtual private networking (VPN), IPsec, IPV6, domain name systems (DNS), distributed systems, and optical networks. He holds multiple patents and research publications, and he co-authored the ebook *Microsoft System Center: Network Virtualization and Cloud Computing* (Microsoft Press, 2013). He has a bachelor's degree in computer science from Osmania University Hyderabad and master's degree from the Indian Institute of Technology Madras.



AANAND RAMACHANDRAN (contributing author) is a Senior Program Manager in the Windows Server Networking organization where he leads the multi-tenant cloud gateway and remote access efforts. He has been with Microsoft for over 9 years, working on various networking technologies, such as Remote Access Client and Server, VPN NAP, DirectAccess, and the Extensible Authentication Protocol framework. Prior to joining Microsoft, Aanand worked at T-Mobile, U.S.A as a network engineer in the Data Platforms Engineering Group, managing T-Mobile's nationwide ATM network, and at Cisco Systems, U.S.A, where he worked on multi-service switching products. Aanand has a master's degree in computer networking and telecommunications from the University of Missouri – Kansas City and a bachelor's degree in computer science and engineering from Pondicherry University, India.

About the series editor

MITCH TULLOCH is a well-known expert on Windows Server administration and cloud computing technologies. He has published hundreds of articles on a wide variety of technology sites and has written, contributed to or been series editor for over 50 books. Mitch is one of the most popular authors at Microsoft Press—the almost two dozen ebooks on Windows Server and System Center he either wrote or was Series Editor on have been downloaded more than 2.5 million times! For a complete list of free ebooks from Microsoft Press, visit the Microsoft Virtual Academy at *http://www.microsoftvirtualacademy.com/ebooks*.

Mitch has repeatedly received Microsoft's Most Valuable Professional (MVP) award for his outstanding contributions to supporting the global IT community. He is a ten-time MVP in the technology area of Windows Server Software Packaging, Deployment & Servicing. You can find his MVP Profile page at *http://mvp.microsoft.com/en-us/mvp/Mitch%20Tulloch-21182*.

Mitch is also Senior Editor of WServerNews, a weekly newsletter focused on system admin and security issues for the Windows Server platform. With almost 100,000 IT pro subscribers worldwide, WServerNews is the most popular Windows Server–focused newsletter in the world. Visit *http://www.wservernews.com* and subscribe to WServerNews today!

Mitch also runs an IT content development business based in Winnipeg, Canada, that produces white papers and other collateral for the business decision maker (BDM) and technical decision maker (TDM) audiences. His published content ranges from white papers about Microsoft cloud technologies to reviews of third-party products designed for the Windows Server platform. Before starting his own business in 1998, Mitch worked as a Microsoft Certified Trainer (MCT) for Productivity Point.

For more information about Mitch, visit his website at *http://www.mtit.com*. You can also follow Mitch on Twitter @mitchtulloch.

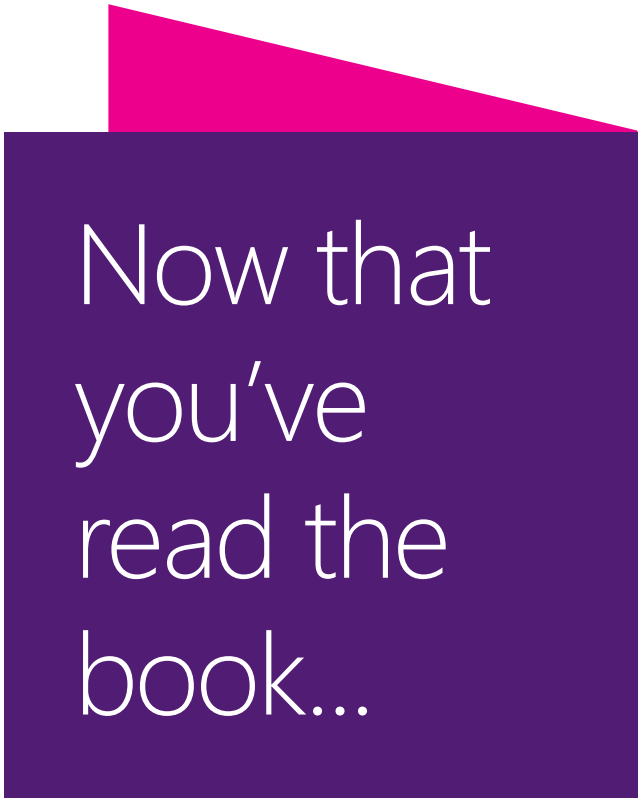
From technical overviews to drilldowns on special topics, get *free* ebooks from Microsoft Press at:

www.microsoftvirtualacademy.com/ebooks

Download your free ebooks in PDF, EPUB, and/or Mobi for Kindle formats.

Look for other great resources at Microsoft Virtual Academy, where you can learn new skills and help advance your career with free Microsoft training delivered by experts.

Microsoft Press



Tell us what you think!

Was it useful?
Did it teach you what you wanted to learn?
Was there room for improvement?

Let us know at http://aka.ms/tellpress

Your feedback goes directly to the staff at Microsoft Press,
and we read every one of your responses. Thanks in advance!

Microsoft