Microsoft

Microsoft System Center

Building a Virtualized Network Solution

Nigel Cain ▪ Alvin Morales ▪ Michel Luescher ▪ Damian Flynn
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: 2014931254
ISBN: 978-0-7356-8310-5

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 Book Support at mspinput@microsoft.com. Please tell us what you think of this book at http://www.microsoft.com/learning/booksurvey.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions Editor: Anne Hamilton
Developmental Editor: Karen Szall
Editorial Production: Megan Smith-Creed
Copyeditor: Megan Smith-Creed
Cover Illustration: Twist Creative, Seattle

Contents

Introduction

According to the Hyper-V Network Virtualization Overview found 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 can often struggle to understand the different components and concepts that make up a solution.

Who should read this book?

Much of the published material covering Network Virtualization today is very much 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 very 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 component plays, and the key decision points, design considerations, and the best practice recommendations they should adopt as they begin to design and build out a virtualized network solution based on Windows Server 2012 and Microsoft System Center 2012 SP1 (or later).

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 in order to build out a virtualized network solution that that meets today's business requirements while also providing a platform for future growth and expansion.

In writing this book, we assume that as architects and fabric administrators interested in Microsoft Network Virtualization you are familiar and have a good understanding of the networking features and capabilities of Windows Server 2012 Hyper-V and System Center 2012 SP1, together with the Microsoft Cloud OS vision available at *http://www.microsoft.com/en-us/server-cloud/cloud-os/default.aspx*.

What topics are included in this book?

Although this book, part of a series of specialized guides on System Center, provides you with insight into the various components of a virtualized network solution primarily based upon Windows Server 2012 and System Center 2012 SP1, many of the concepts, advice, and guidance outlined in respect of best practice are unchanged for the R2 release.

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 component of the solution. The remaining chapters are more operational and discuss how to deploy and how to manage some of the common changes that might need to be made post deployment.

- **Chapter 1: Key concepts** A virtualized network solution built on Windows Server 2012 and System Center 2012 SP1 depends on a number of different components, and this chapter outlines the role each of these components plays in the overall solution and how they are interconnected.
- **Chapter 2: Logical networks** This chapter takes a look at some of the main reasons why you would (or would not) create a logical network, 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: Port profiles** This chapter discusses the different types of port profiles in Microsoft System Center 2012 Virtual Machine Manager (VMM)— uplink port profiles and network adapter port profiles—describes what they are used for, and provides detailed guidance for how and when to create them.
- **Chapter 4: Logical switches** This chapter covers logical switches, essentially templates for Hyper-V switches, which allow you to consistently apply the same settings and configuration across multiple hosts and ensure that any Hyper-V switches you deploy and configure using a logical switch remain compliant with it.
- **Chapter 5: 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 6: Operations** Even after having carefully planned a virtual network solution, things outside of your immediate control may 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.

To recap, this book is mainly focused on architecture and design, what is needed to design a virtualized network solution rather than the actual steps required to deploy it in your environment. Other than in Chapter 5, which focuses on deployment issues and considerations, and Chapter 6, which covers managing change to the environment post deployment, you will find very 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 to identify what you need to do to build out a solution that will meet the needs of your business and provide a platform for the future.

Once you have designed a solution using the guidelines documented in this book, you will be able to make effective use of the 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 Stanislav Zhelyazkov (MVP), Hans Vredevoort (MVP), Phillip Moss (NTTX), and Greg Cusanza, Thomas Roettinger, Artem Pronichkin, and Cristian Edwards Sabathe from Microsoft for providing valuable feedback and suggestions on the content of the book. Without their contributions this book would not be as thorough nor as complete; so our thanks once again for their time and efforts in making this happen.

Errata & book support

We’ve made every effort to ensure the accuracy of this content and its companion content. Any errors that have been reported since this content was published are listed on our Microsoft Press site at oreilly.com:

http://aka.ms/SCvirtnetsol/errata

If you find an error that is not already listed, you can report it to us through 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 is not offered through the addresses above.

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 components. It is important to understand the role these components 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 an example organization
- Identify the different components of a virtualized network solution
- Provide an overview of each component and how to configure it
- Describe how these components are used to configure virtualized networking on multiple Hyper-V host computers

Introducing Contoso Ltd.

Since a lot of planning considerations and best practice approaches are discussed in this book, we've use an fictitious organization (Contoso) to help put many of these points into context. Contoso Ltd. is a service provider—otherwise known as a hoster—that offers Infrastructure as 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).

Contoso 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. Contoso has decided to deploy and use Windows Server 2012 and Microsoft System Center 2012 SP1 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.

The chapters that follow discuss the architectural and design decisions that Contoso needs to make to build out the virtualized network component of their new hosting service and provide some best practice recommendations and guidance along the way. Although your organization may not be a service provider and your business requirements may be very different from Contoso's, the design processes, key decision points, and implications of certain design choices are applicable to all customers that want to use Windows Server and System Center to create a cost effective and highly efficient private or hybrid cloud solution.

Architecture

Figure 1-1 is a simplified diagram that illustrates the different layers and components that make up the architecture of a virtualized networking solution based on Windows Server and System Center. In this diagram, the physical network and Hyper-V host computers are at the bottom and the deployed virtual machines and services are at the top. On the right are the names of each component; the labels on the left describe how these components are connected. For example, a logical switch is connected to a logical network via a logical network.

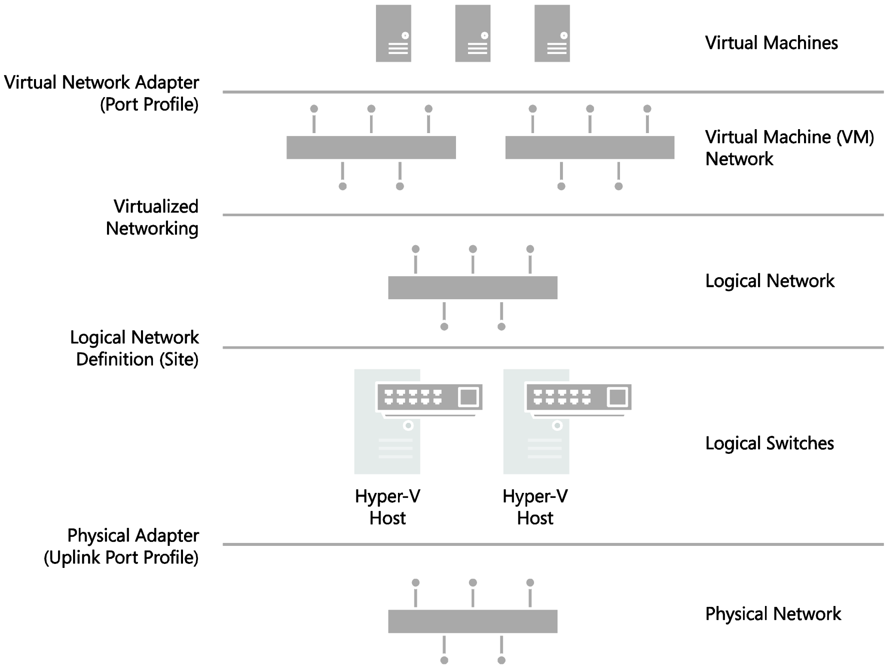

Virtual Machines
Virtual Network Adapter (Port Profile)
Virtual Machine (VM) Network
Virtualized Networking
Logical Network
Logical Network Definition (Site)
Logical Switches
Hyper-V Host
Hyper-V Host
Physical Adapter (Uplink Port Profile)
Physical Network

FIGURE 1-1 Architecture of a virtualized network solution.

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

Virtualized network components

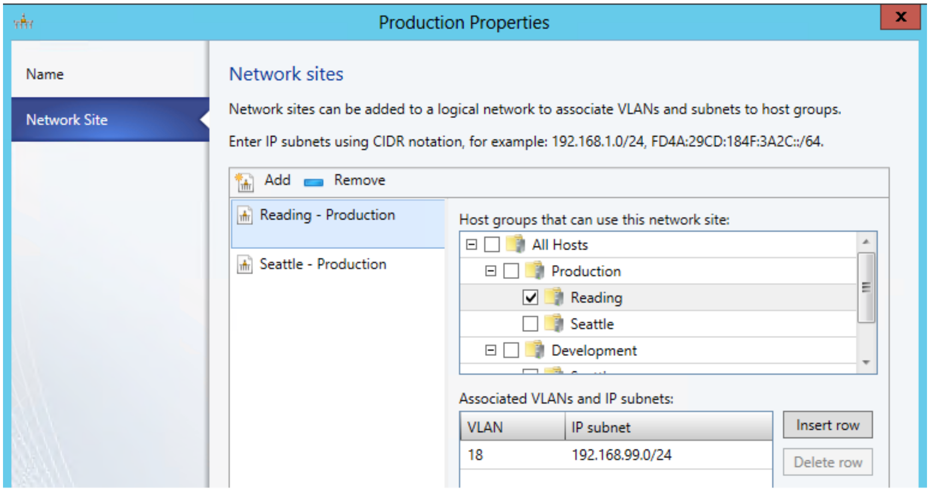
There are a number of different components in a virtualized network solution that must be defined and configured before you can begin to take full advantage of the features and flexibility provided by Windows Server 2012 Hyper-V and System Center 2012 SP1.

- **Logical networks** These represent an abstraction of the underlying physical network infrastructure and enable you to model the network based on business needs and connectivity requirements.
- **Uplink port profiles** These 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.
- **Network adapter port profiles** These are templates that define offload settings, bandwidth policies, and security settings for virtual network adapters.
- **Port classification** This is a user friendly label that can be linked to a network adapter port profile. (Port classifications are not shown in Figure 1-1.)
- **IP address pools** These allow VMM to automatically allocate static IP addresses to Windows-based virtual machines that are running on any managed Hyper-V, VMware ESX, or Citrix XenServer host.
- **MAC Address Pools** If virtual machines connected to the logical network will obtain IP addresses from a static IP address pool, you must also configure the virtual machine to use a static MAC address. You can either specify the MAC address manually or have VMM automatically assign a MAC address from a MAC address pool.
- **Logical switches** These bring together uplink port profiles, native port profiles, port classifications, and switch extensions that are relevant to a particular physical or logical network.
- **Virtual machine networks** These provide the network interface through which a virtual machine will connect to a particular logical network.

Logical network

The 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 Contoso, 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. Virtual machines running production workloads on hosts in the Reading Datacenter need to use VLAN18 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.



Network sites can be added to a logical network to associate VLANs and subnets to host groups.

Enter IP subnets using CIDR notation, for example: 192.168.1.0/24, FD4A:29CD:184F:3A2C::/64.

Host groups that can use this network site:

- ⊟ □ All Hosts
  - ⊟ □ Production
    - ☑ Reading
    - □ Seattle
  - ⊟ □ Development

Associated VLANs and IP subnets:

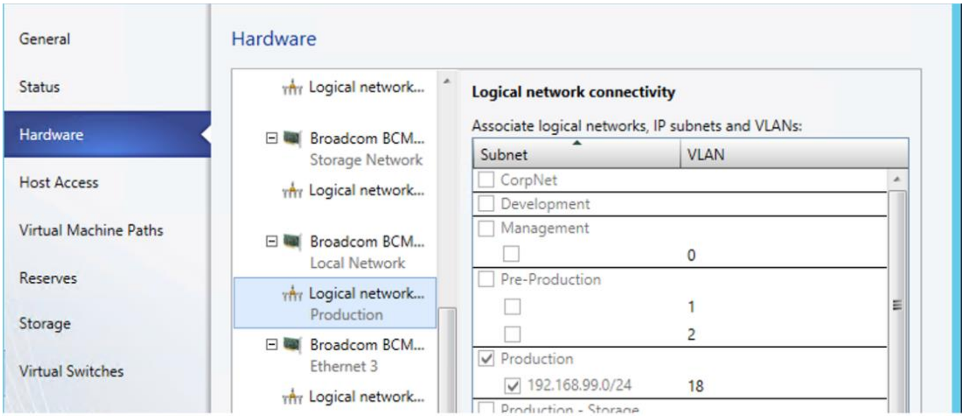
VLAN	IP subnet
18	192.168.99.0/24

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 virtual machines 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 virtual machines 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 Contoso, 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 virtual machines running on that host.



Logical network connectivity

Associate logical networks, IP subnets and VLANs:

Subnet	VLAN
□ CorpNet	
□ Development	
□ Management	
□	0
□ Pre-Production	
□	1
□	2
☑ Production	
☑ 192.168.99.0/24	18
□ Production - Storage	

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

You might expect that the result of this configuration, once it has been deployed to hosts in both locations, would be that a virtual machine 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 virtual machine is configured with the VLAN and IP address appropriate for the logical network in the new physical location.

Moving existing virtual machines between sites like this is certainly possible, but there are a few caveats. The main one is that the IP address assigned to the virtual machine 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 virtual machine will continue to communicate on the network once 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 virtual machine to a new location, it will have an incorrect VLAN/IP address for that location.

NOTE **A virtual machine connected to a virtual machine network that uses Network Virtualization where the Production logical network has been enabled can be moved between hosts in Reading and Seattle without requiring any additional configuration.**

IP and MAC 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 virtual machines 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 virtual machines and to virtual machines 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 virtual machine manually, VMM will detect 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 virtual machine.

NOTE **When isolating network traffic using Network Virtualization, which is covered in more detail in Chapter 2, the logical network also has a relationship with deployed virtual machines in that each machine must be allocated an IP address from one of the IP pools that have been defined for that logical network. An IP address from this pool, otherwise known as provider address (PA), is routable between Hyper-V hosts.**

If you configure a virtual machine to obtain its IP address from a static IP address pool, you must also configure the virtual machine 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.

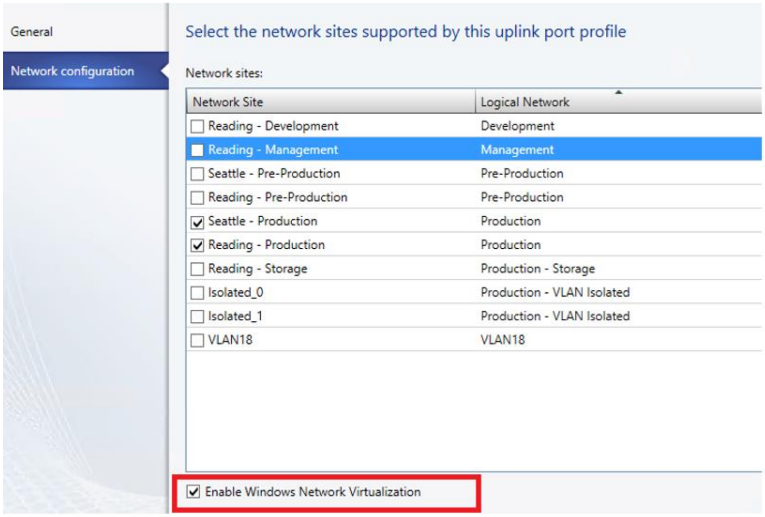
Uplink port profiles

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 will be required for each physical network unless you need to define custom connectivity rules, have multiple physical networks, or wish to restrict logical networks to specific hosts within a given physical location, in which case you will 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, "Port profiles."

At Contoso, 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 virtual machines in that location.



Select the network sites supported by this uplink port profile

Network sites:

Network Site	Logical Network
□ Reading - Development	Development
□ Reading - Management	Management
□ Seattle - Pre-Production	Pre-Production
□ Reading - Pre-Production	Pre-Production
☑ Seattle - Production	Production
☑ Reading - Production	Production
□ Reading - Storage	Production - Storage
□ Isolated_0	Production - VLAN Isolated
□ Isolated_1	Production - VLAN Isolated
□ VLAN18	VLAN18

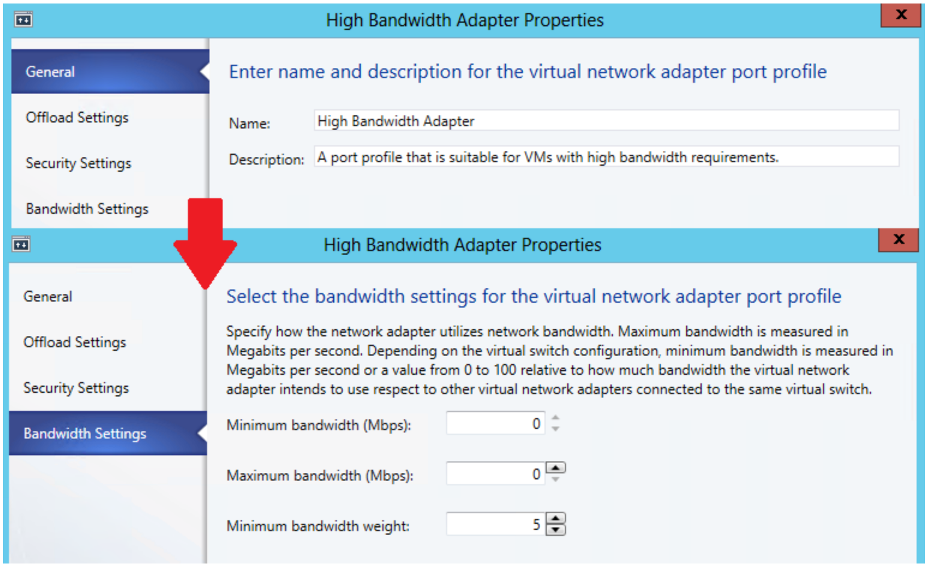
☑ Enable Windows Network Virtualization

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.



High Bandwidth Adapter Properties

Enter name and description for the virtual network adapter port profile

Name: High Bandwidth Adapter

Description: A port profile that is suitable for VMs with high bandwidth requirements.

High Bandwidth Adapter Properties

Select the bandwidth settings for the virtual network adapter port profile

Specify how the network adapter utilizes network bandwidth. Maximum bandwidth is measured in Megabits per second. Depending on the virtual switch configuration, minimum bandwidth is measured in Megabits per second or a value from 0 to 100 relative to how much bandwidth the virtual network adapter intends to use respect to other virtual network adapters connected to the same virtual switch.

Minimum bandwidth (Mbps): 0

Maximum bandwidth (Mbps): 0

Minimum bandwidth weight: 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) will 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 virtual machines and network cards that allow you to take advantage of some of the features provided by your host hardware.

Port classifications

Port classifications 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 virtual machine 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 will 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) we will need to create at least three logical switches 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).

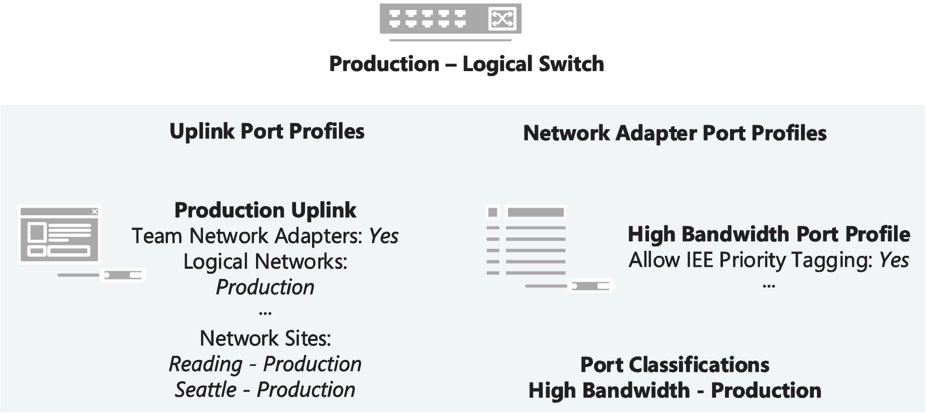

Production – Logical Switch
Uplink Port Profiles
Network Adapter Port Profiles
Production Uplink
Team Network Adapters: Yes
Logical Networks:
Production
...
Network Sites:
Reading - Production
Seattle - Production
High Bandwidth Port Profile
Allow IEE Priority Tagging: Yes
...
Port Classifications
High Bandwidth - Production

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 virtual machines 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 virtual machines to the network.

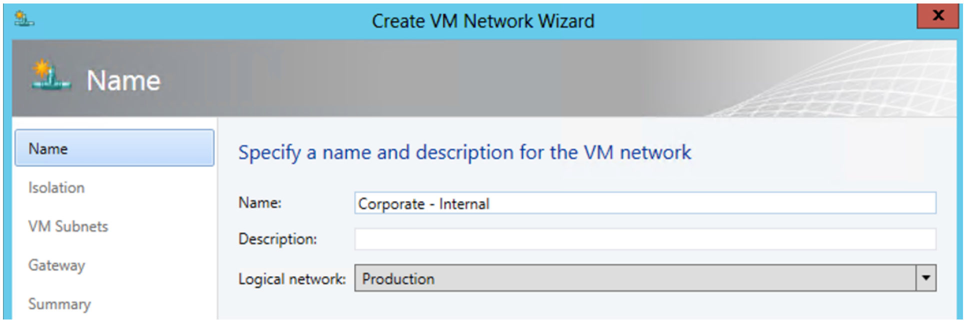
NOTE **The previous example does not include any switch extensions; however, you might want to include these in your logical switch to allow you to monitor network traffic, use quality of service (QoS) to control how network bandwidth is used, enhance the level of security, or otherwise expand the capabilities 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 may need to consider creating an additional logical switch.**

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

Virtual machine networks

In terms of overall architecture, virtual machine (VM) networks are the final component to consider in this short overview since they provide the (network) interface through which a virtual machine 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 virtual machines 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.

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. This concept is key to support multiple tenants (clients or customers) with their own networks and will be covered in much more detail in Chapter 6, "Operations."



Specify a name and description for the VM network

Name: Corporate - Internal

Description:

Logical network: Production

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

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.

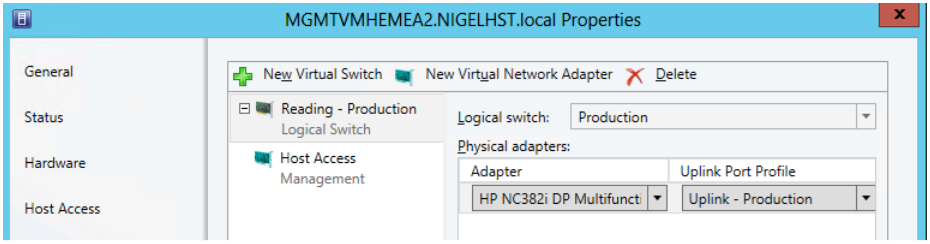
Deploying the solution

You can of course configure the network settings and properties on each Hyper-V host manually or by using Windows PowerShell, but 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 using VMM as described. When a logical switch is applied to a network adapter in a Hyper-V host, 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 and associate the network adapter with the required logical networks, VLAN, and IP subnets. It therefore follows that the host must be a member of a VMM 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 **If you apply the same logical switch and uplink port profile to two or more adapters, the two adapters will be teamed, assuming that this option has been defined in the logical switch. The option to add or remove adapters will be available only if Uplink Mode has been set to Team.**

Returning to the example organization, imagine a number of new Hyper-V hosts have been deployed in the Reading datacenter in response to increasing demand for computing capacity in production. Each one of these hosts must be configured for production workloads, meaning that its physical network adapters are teamed to provide maximum bandwidth and a degree of resilience to adapter failure.

Figure 1-8 shows the logical switch being applied on one of the new servers. The administrator has selected the Production uplink port profile to ensure that the selected network adapters are configured with the VLAN and IP Subnets that are appropriate for this location.



Logical switch: Production

Physical adapters:

Adapter	Uplink Port Profile
HP NC382i DP Multifuncti	Uplink - Production

FIGURE 1-8 Deploying a logical switch on a Hyper-V host.

Using this information, VMM will create a Hyper-V virtual switch on the host and use the logical networks, VLAN, and IP subnets from the uplink port profile to configure these properties on the selected network adapters. Once the switch has been deployed, the physical network adapter can no longer be configured through the UI or PowerShell; any further changes to the logical networks, VLAN, and IP subnets for the network adapter must be configured in the uplink port profile.

NOTE **A Hyper-V virtual switch deployed by VMM can be configured directly on the host computer using native Hyper-V and built-in operating system tools. However making changes to the switch in this way is strongly discouraged since it can lead to unexpected results.**

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. The Microsoft System Center 2012 Virtual Machine Manager (VMM) documentation indicates 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 virtual local area networks (VLANs), IP subnets, and IP subnet/VLAN pairs that are associated with the logical network in each physical location.” It goes on to state that 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 components of the architecture. The key takeaways from this diagram for Chapter 2 are:

- Logical networks are connected to a logical switch via a logical network definition (otherwise known as a network site) and to virtual machine (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 must be connected to a VM network to access network resources in VMM, it follows that you will need to define *at least one* VM network for each logical network that will be accessed by virtual machines.

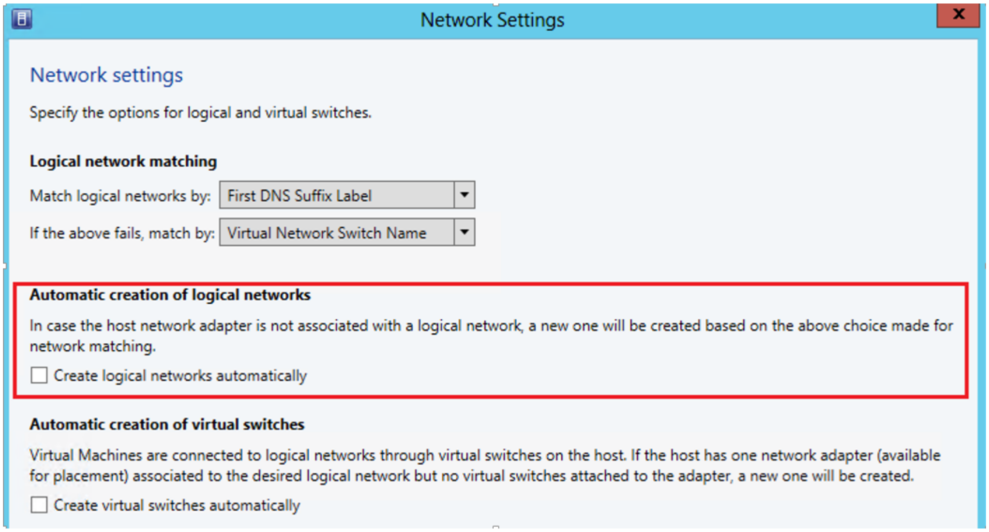
Although not shown in Figure 1-1, logical networks also have a relationship with clouds. VMM uses this relationship to scope or otherwise restrict the list of VM networks that are available to users during virtual machine placement. To be placed in a cloud, the virtual machine must be connected to a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

Getting started with logical networks

At least one logical network must be associated with a given host computer for it to support deployed virtual machines 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.contoso.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 2-1). The same is true of the option to automatically create virtual switches, which is discussed later in this chapter.



Network Settings

Network settings

Specify the options for logical and virtual switches.

Logical network matching

Match logical networks by: First DNS Suffix Label

If the above fails, match by: Virtual Network Switch Name

Automatic creation of logical networks

In case the host network adapter is not associated with a logical network, a new one will be created based on the above choice made for network matching.

□ Create logical networks automatically

Automatic creation of virtual switches

Virtual Machines are connected to logical networks through virtual switches on the host. If the host has one network adapter (available for placement) associated to the desired logical network but no virtual switches attached to the adapter, a new one will be created.

□ Create virtual switches automatically

FIGURE 2-1 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.

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 5, “Deployment”).

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 Contoso network

To set this process in context, take a closer look at the physical network in the example service provider, Contoso Ltd., and identify the set of logical networks that will be needed to support this company's specific business requirements. Although your own network layout and business requirements might be different from Contoso's, the design process, key decision points, and the implications of certain design choices are applicable in all cases.

As shown in Figure 2-2, Contoso 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 improves security, simplifies infrastructure management, and removes potential competition between different types of network traffic.

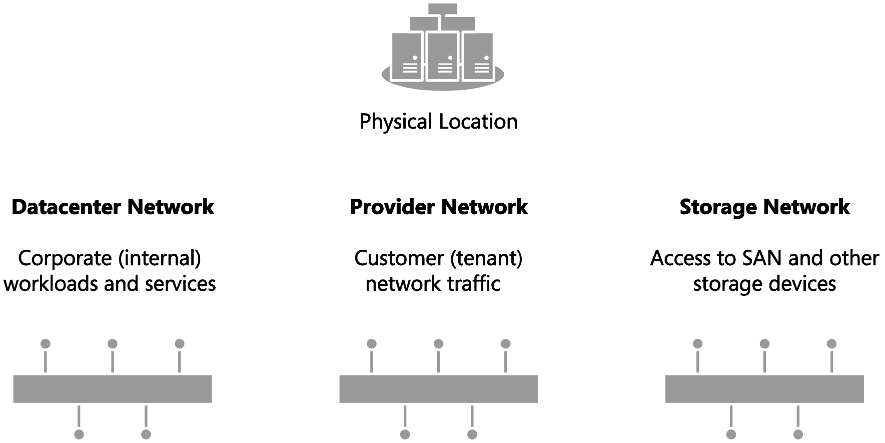

Physical Location
Datacenter Network
Corporate (internal) workloads and services
Provider Network
Customer (tenant) network traffic
Storage Network
Access to SAN and other storage devices

FIGURE 2-2 Physical networks in each Contoso datacenter.

The following sections outline the five step logical network design process for Contoso, 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.

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 Contoso has three physical networks (Datacenter, Provider, and Storage), the assumption is that that three logical networks will be required to support this environment, one for each physical network, as shown in Figure 2-3. 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.

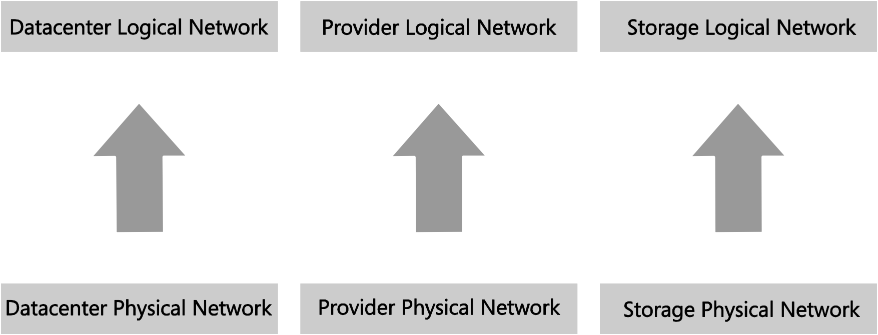

Datacenter Logical Network
Provider Logical Network
Storage Logical Network
Datacenter Physical Network
Provider Physical Network
Storage Physical Network

FIGURE 2-3 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 Contoso'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 Contoso Ltd. 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 Contoso 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 2-4. Note that network adapter port profiles (explained in Chapter 3, "Port profiles") will be used to apply the required properties and capabilities, including bandwidth limitations and IEEE priority tags, to virtual machines and services that connect those networks.

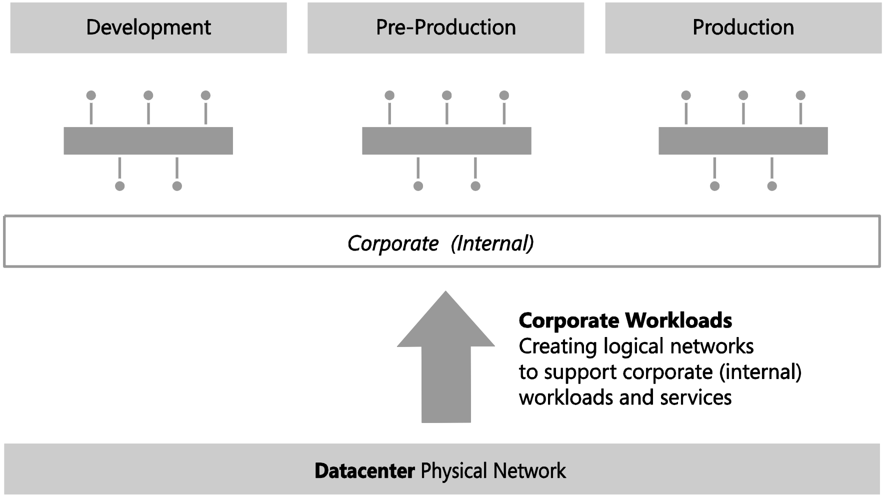


FIGURE 2-4 Using logical networks to differentiate workloads.

> ***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 components (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 virtual machines 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 Contoso 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 components of these services, as shown in Figure 2-5. 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.

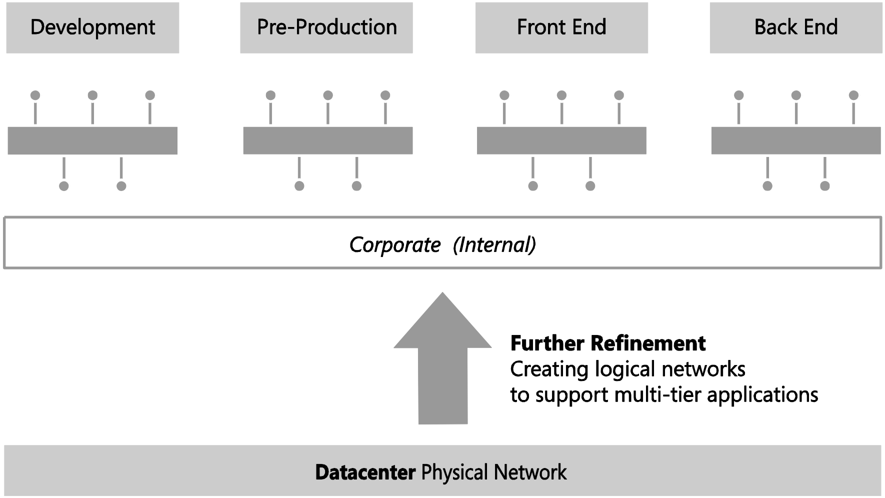

Development
Pre-Production
Front End
Back End
Corporate (Internal)
Further Refinement
Creating logical networks
to support multi-tier applications
Datacenter Physical Network

FIGURE 2-5 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, Contoso 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 CSV, 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.

Contoso has chosen to create logical networks for each of the infrastructure services, as shown in Figure 2-6. You might choose to implement this differently, adding or removing logical networks from the design based upon your requirements and the capabilities of your network infrastructure.

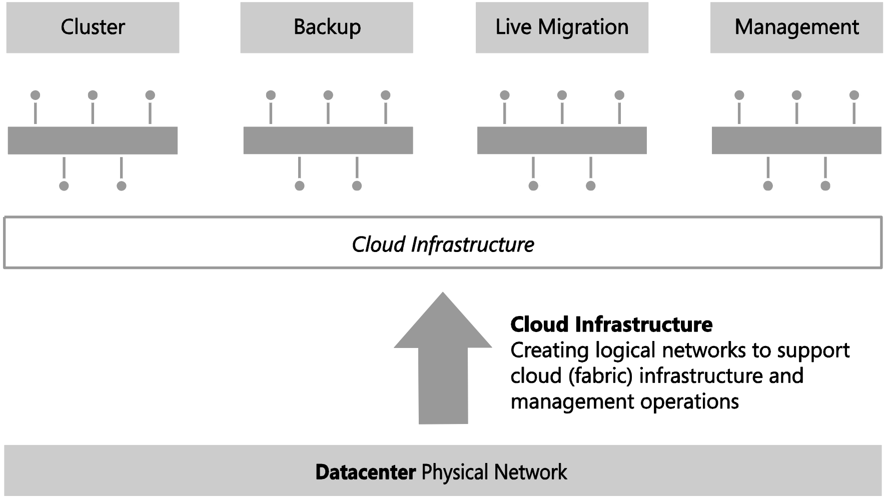

Cluster
Backup
Live Migration
Management
Cloud Infrastructure
Cloud Infrastructure
Creating logical networks to support cloud (fabric) infrastructure and management operations
Datacenter Physical Network

FIGURE 2-6 Using logical networks to differentiate cloud infrastructure services.

Provider physical network

Contoso 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, additionally, 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 Contoso, 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 you will support both of these scenarios. 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 has been allocated to (reserved for) that customer. This is covered in much more detail in Chapter 5, "Deployment."

Contoso, 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 2-7. 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.

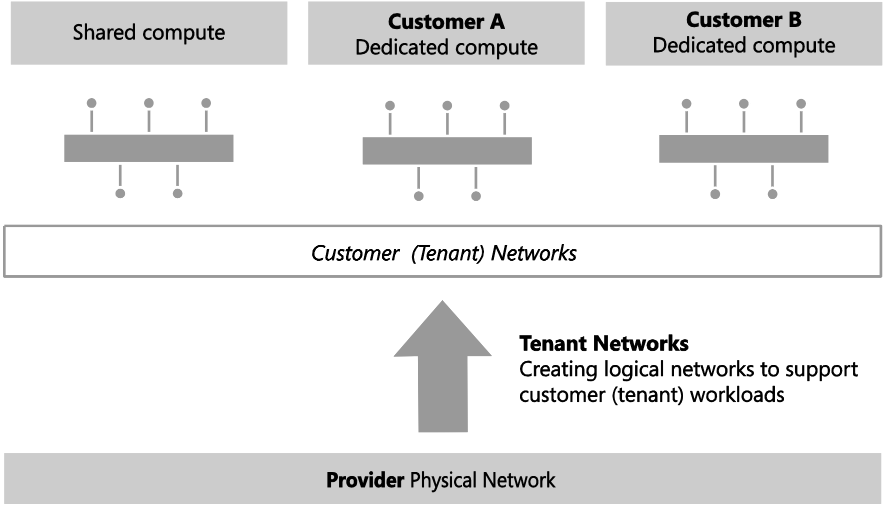

Shared compute
Customer A
Dedicated compute
Customer B
Dedicated compute
Customer (Tenant) Networks
Tenant Networks
Creating logical networks to support customer (tenant) workloads
Provider Physical Network

FIGURE 2-7 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 Contoso 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. Note that if Contoso were to use multiple IP-based storage technologies (such as iSCSI and SMB) on the physical network, each of these technologies would likely be allocated its own logical network.

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 may 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 in VMM effectively modeled this behavior by enabling resources connected to a given logical network to communicate with any other resource on that same network, with inter-network communication handled via a router or gateway device. The problem with this solution for service providers (hosters) was that each customer invariably required a distinct logical network, which led to the creation of hundreds if not thousands of logical networks within VMM, resulting in performance issues and increased complexity and management overhead.

VM networks were introduced to address this particular issue. Rather than connecting directly to a logical network, VMs in this release connect to a VM network, which acts as an interface to a particular part of a logical network, as shown in Figure 2-8. Since VM networks are linked to the logical network instead of associated with physical host computers, adding, deleting, and making changes to these VM networks is easier than making changes to logical networks. Multiple VM networks may be linked to each logical network, removing the need for service providers to create separate logical networks for each of their customers.

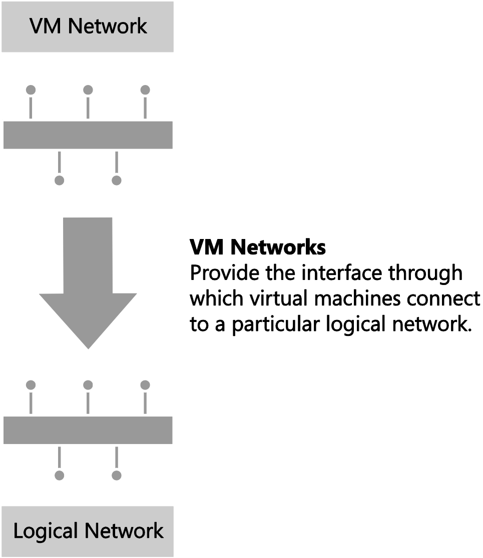

VM Network
VM Networks
Provide the interface through which virtual machines connect to a particular logical network.
Logical Network

FIGURE 2-8 VM networks' relationship to logical networks.

If there is no need to separate or otherwise isolate network traffic from certain machines, only a single VM network linked to the logical network is required. As described earlier, multiple VM networks are required to host workloads for multiple customers (tenants) on the same logical network, with each tenant isolated from and unaware of any others.

In VMM, you can isolate VM networks by using either traditional VLAN (or isolated PVLAN) solutions or, if you are using Windows Server 2012 as your host operating system, by implementing Network Virtualization. The latter option addresses the scale limitations associated with a traditional VLAN solution and allows tenants to "bring their own network" or otherwise extend their network into your environment.

MORE INFO **The different ways that VMM can isolate network traffic are well illustrated by the Logical Networks section of the "Networking in Virtual Machine Manager" poster that can be downloaded from the Microsoft Download Center at** ***http://www.microsoft.com/en-us/download/details.aspx?id=37137*****.**

Note that 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

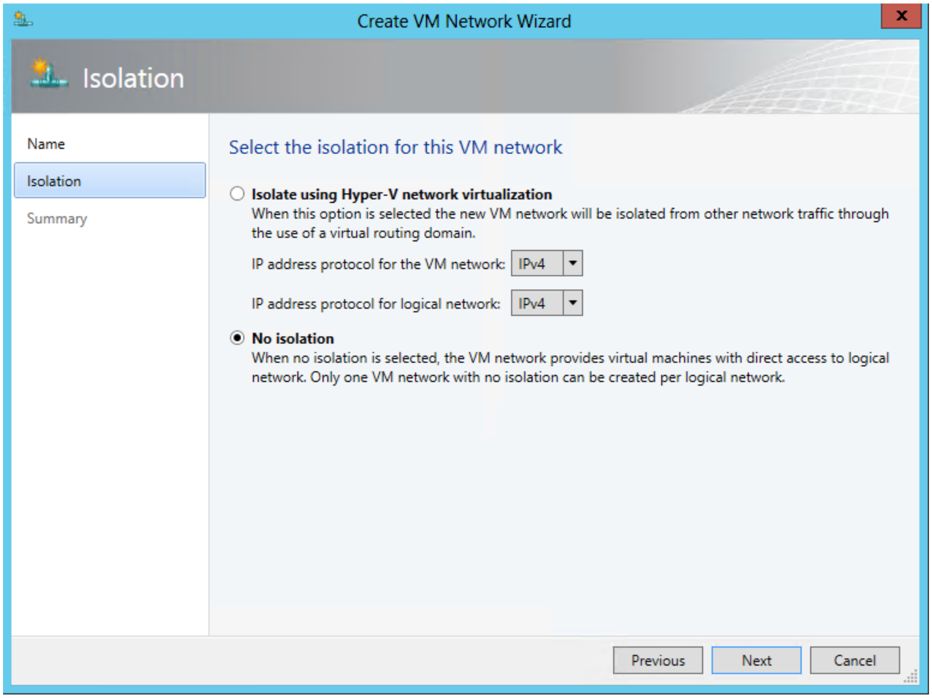
VLAN/PVLAN technology and others by using Network Virtualization. Should you need to use multiple approaches in your environment, you will need to return to Step 2 above and create a separate logical network for each isolation method.

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 configured for No Isolation, as shown in Figure 2-9, is required. The VM network in this example simply acts as a “pass through” to the logical network.



Select the isolation for this VM network

○ **Isolate using Hyper-V network virtualization**
When this option is selected the new VM network will be isolated from other network traffic through the use of a virtual routing domain.

IP address protocol for the VM network: IPv4

IP address protocol for logical network: IPv4

● **No isolation**
When no isolation is selected, the VM network provides virtual machines with direct access to logical network. Only one VM network with no isolation can be created per logical network.

FIGURE 2-9 Creating a VM network with no isolation.

> ***NOTE*** In VMM 2012, virtual machines were directly connected to logical networks. When customers using this release upgraded to SP1, VM networks configured for No Isolation were automatically created for each of these logical networks. Virtual machines that existed prior to the upgrade were then connected to the new VM network linked to their original logical network.

This configuration establishes a one-to-one mapping between the VM network and the logical network, as shown in Figure 2-10. As a result, only one VM network per logical network can be configured for No Isolation. If virtual machines 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.

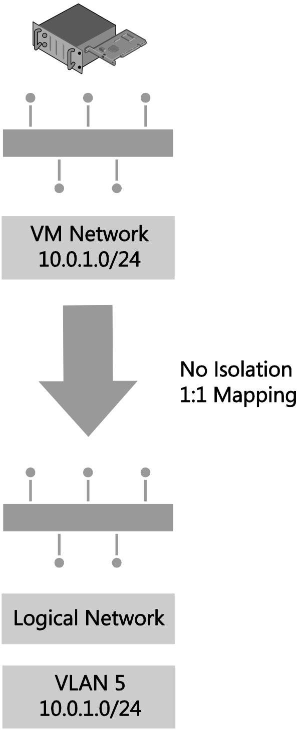
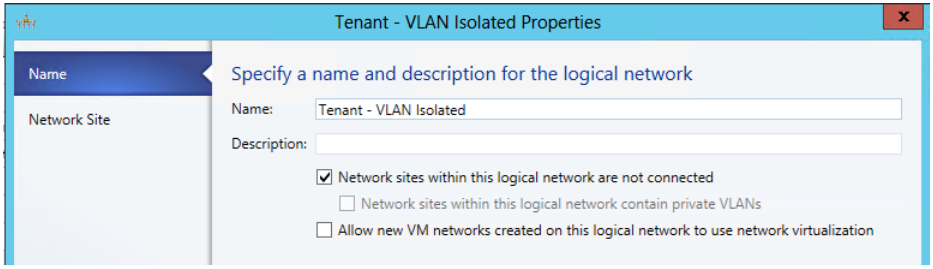


FIGURE 2-10 VM network direct access to the logical network.

> ***NOTE*** **For logical networks that will not be used by virtual machines, 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. Multiple VM networks may be linked to the same logical network if Network Virtualization is enabled, with each one of these VM networks separated from and unaware of any of the others. These improvements mean that instead of creating a separate logical network for each customer that will be isolated from others using VLAN technology, you can instead create a single logical network for all of these customers, configuring the properties of the network, as shown in Figure 2-11, to specify that sites within this logical network are not connected. Each VM 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.



Tenant - VLAN Isolated Properties

Specify a name and description for the logical network

Name: Tenant - VLAN Isolated

Description:

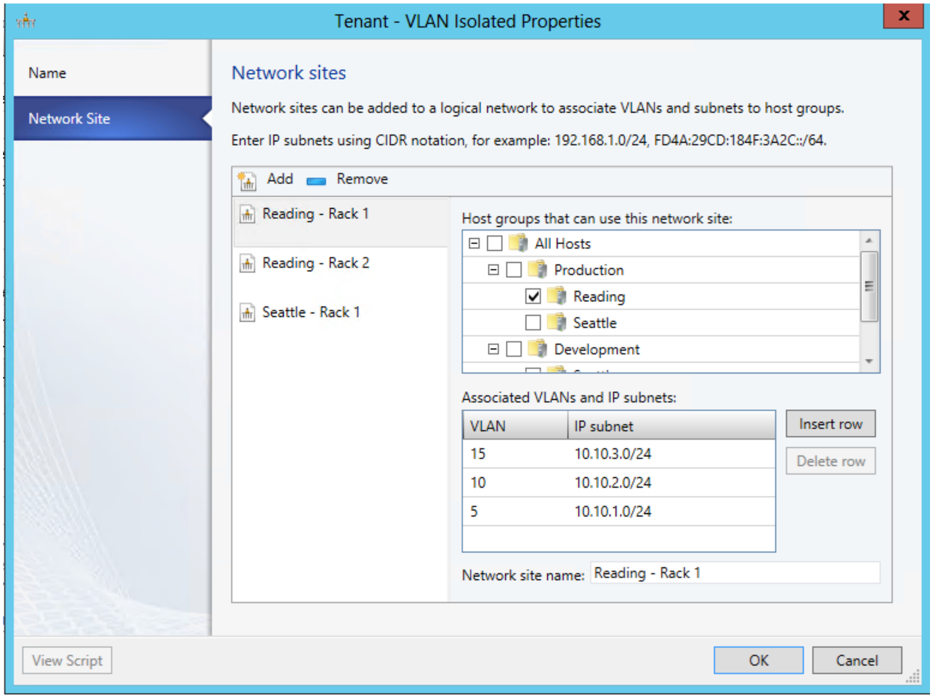
☑ Network sites within this logical network are not connected

□ Network sites within this logical network contain private VLANs

□ Allow new VM networks created on this logical network to use network virtualization

FIGURE 2-11 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 2-12, but each one will be totally isolated from any of the others.



Tenant - VLAN Isolated Properties

Network sites

Network sites can be added to a logical network to associate VLANs and subnets to host groups.

Enter IP subnets using CIDR notation, for example: 192.168.1.0/24, FD4A:29CD:184F:3A2C::/64.

- Reading - Rack 1
- Reading - Rack 2
- Seattle - Rack 1

Host groups that can use this network site:

- □ All Hosts
  - □ Production
    - ☑ Reading
    - □ Seattle
  - □ Development

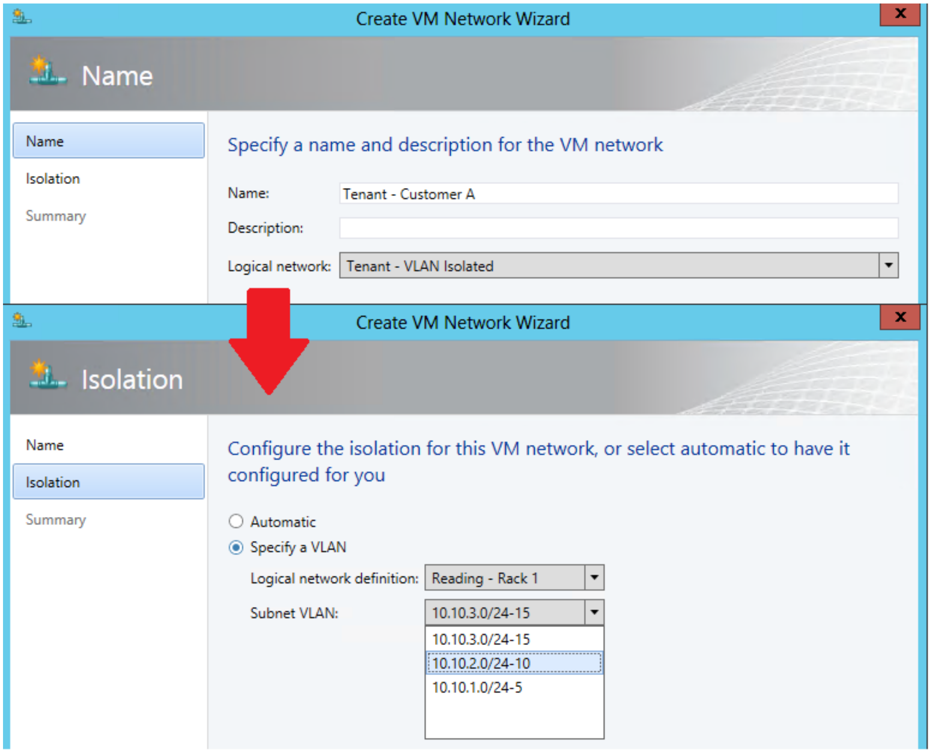
Associated VLANs and IP subnets:

VLAN	IP subnet
15	10.10.3.0/24
10	10.10.2.0/24
5	10.10.1.0/24

Network site name: Reading - Rack 1

FIGURE 2-12 Defining network sites for VLAN isolation.

Finally, VM networks need to be created to allow customer virtual machines 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 2-13, will display only those network sites that have not already been allocated to an existing VM network.



Create VM Network Wizard

Name

- Name
- Isolation
- Summary

Specify a name and description for the VM network

Name: Tenant - Customer A

Description:

Logical network: Tenant - VLAN Isolated

Create VM Network Wizard

Isolation

- Name
- Isolation
- Summary

Configure the isolation for this VM network, or select automatic to have it configured for you

○ Automatic

● Specify a VLAN

Logical network definition: Reading - Rack 1

Subnet VLAN: 10.10.3.0/24-15

- 10.10.3.0/24-15
- 10.10.2.0/24-10
- 10.10.1.0/24-5

FIGURE 2-13 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 configured with the Sites Within The Logical Network Are Not Connected option, create sites, and then specify the list of VLANs that exist in each site. Either create a VM networks 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 2-14.

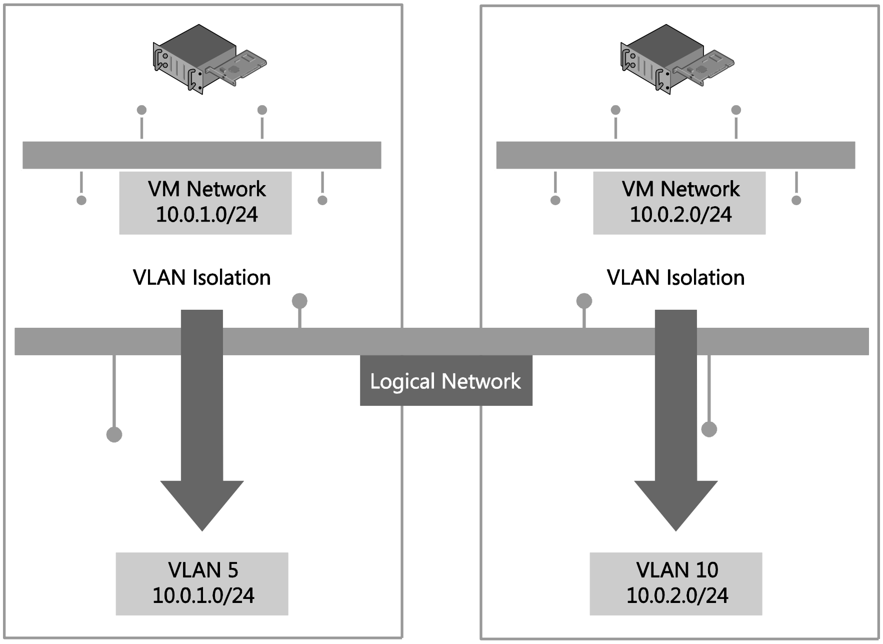


FIGURE 2-14 Logical network design for VLAN isolation.

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

Private virtual LANs (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 2-15. 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.

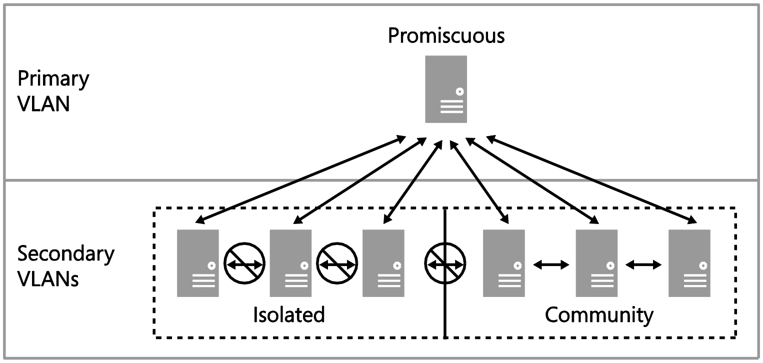
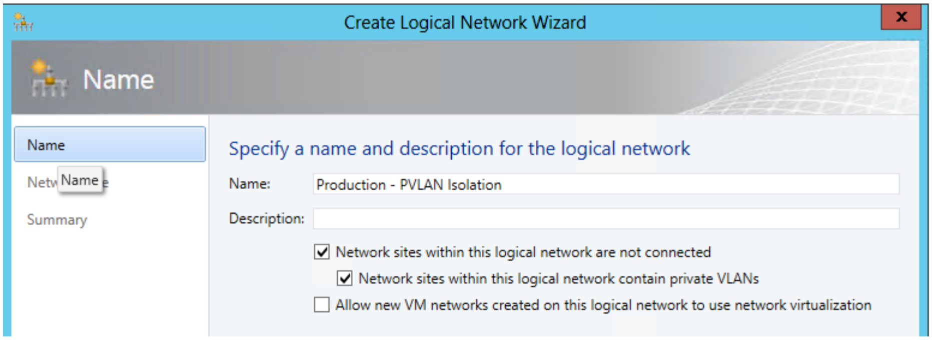


FIGURE 2-15 The three modes for PVLAN isolation.

VMM only supports isolation mode and has no concept of primary (promiscuous) or community modes. What this means in practice is that a virtual machine 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, there are a number of scenarios that 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.**

In terms of logical network design, you should create a single logical network when using PVLANs, configuring the properties of the logical network with the Sites Within This Logical Network Are Not Connected and Network Sites Within This Logical Network Contain Private VLANs options, as shown in Figure 2-16.



Specify a name and description for the logical network

Name: Production - PVLAN Isolation

Description:

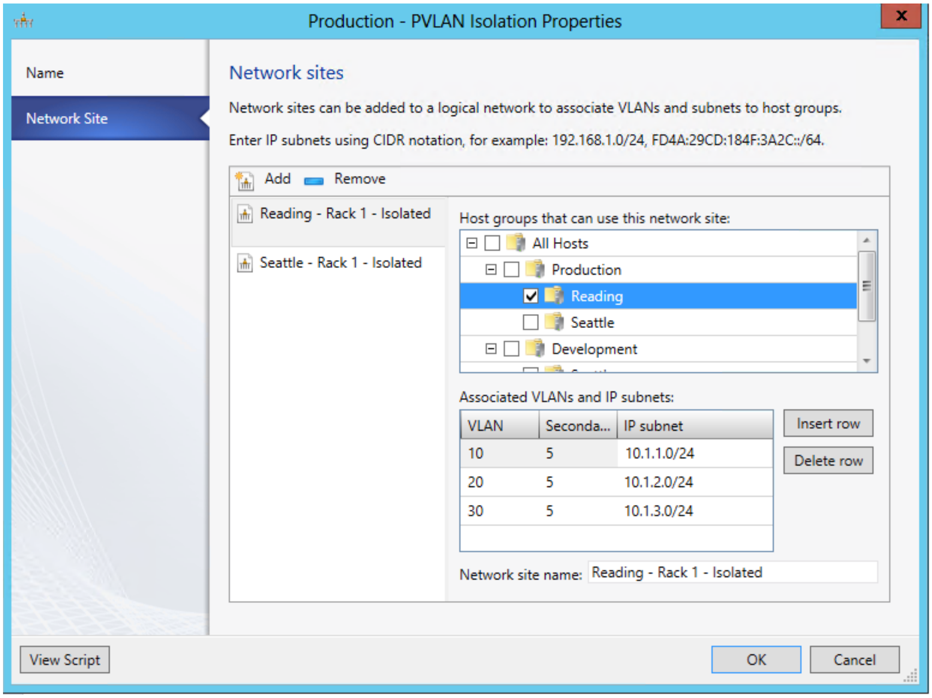
☑ Network sites within this logical network are not connected

☑ Network sites within this logical network contain private VLANs

☐ Allow new VM networks created on this logical network to use network virtualization

FIGURE 2-16 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 2-17. You can also define multiple PVLANS in the same network site as needed.



Network sites

Network sites can be added to a logical network to associate VLANs and subnets to host groups.

Enter IP subnets using CIDR notation, for example: 192.168.1.0/24, FD4A:29CD:184F:3A2C::/64.

Reading - Rack 1 - Isolated

Seattle - Rack 1 - Isolated

Host groups that can use this network site:

- ☐ All Hosts
  - ☐ Production
    - ☑ Reading
    - ☐ Seattle
  - ☐ Development

Associated VLANs and IP subnets:

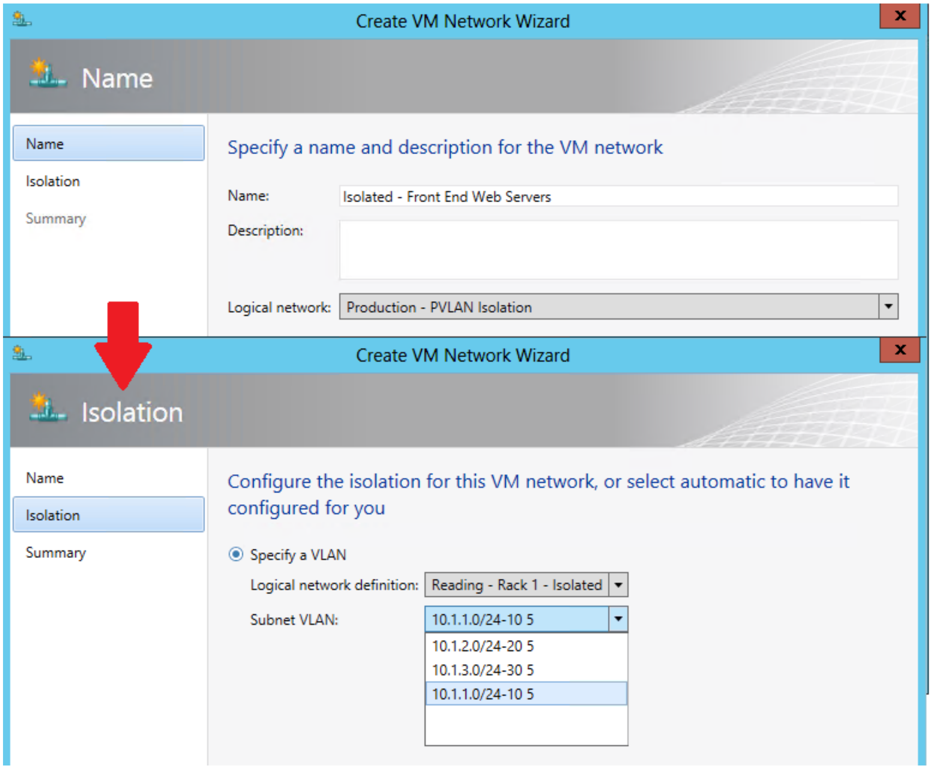
VLAN	Seconda...	IP subnet
10	5	10.1.1.0/24
20	5	10.1.2.0/24
30	5	10.1.3.0/24

Network site name: Reading - Rack 1 - Isolated

FIGURE 2-17 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 2-18, 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.



Specify a name and description for the VM network

Name: Isolated - Front End Web Servers

Description:

Logical network: Production - PVLAN Isolation

Configure the isolation for this VM network, or select automatic to have it configured for you

● Specify a VLAN

Logical network definition: Reading - Rack 1 - Isolated

Subnet VLAN: 10.1.1.0/24-10 5

10.1.2.0/24-20 5
10.1.3.0/24-30 5
10.1.1.0/24-10 5

FIGURE 2-18 Allocating a PVLAN (network site) to a VM network.

To briefly summarize, create a single logical network to support PVLAN isolation, then configure it with the options Sites Within The Logical Network Are Not Connected and Network Sites Within The Logical Network Contain Private VLANs. You should create a network site, define primary and secondary VLAN pairs, and create VM networks for each one, as shown in Figure 2-19. In this example, PVLAN 5 is designated as the isolated PVLAN for consistency across all primary VLANs. Your implementation may be different.

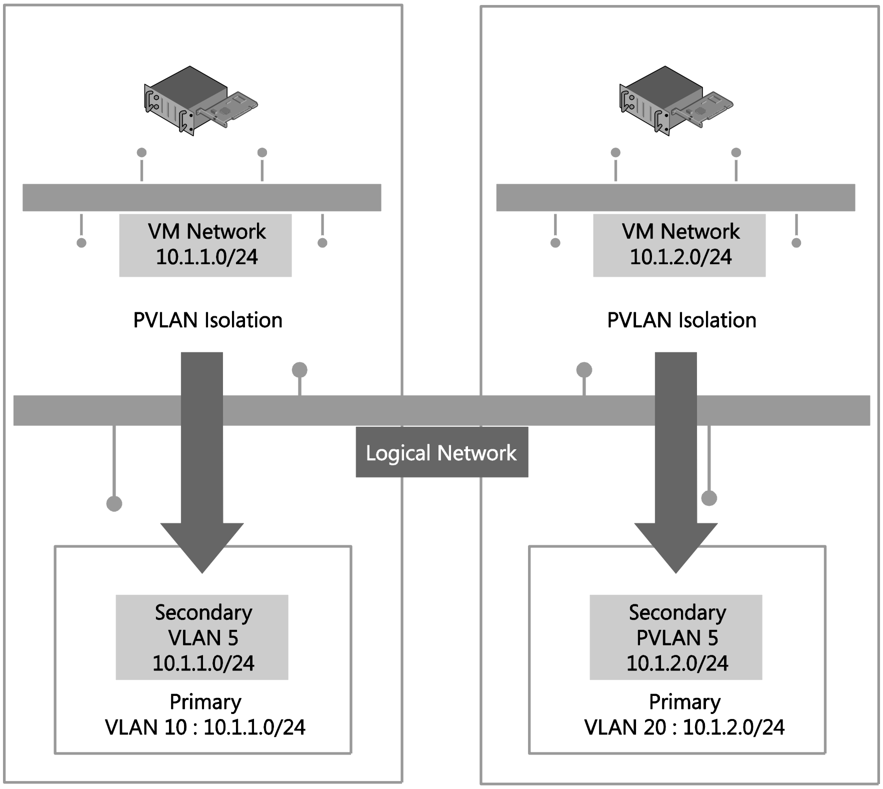


FIGURE 2-19 Logical network design for PVLAN isolation.

Although each virtual machine 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, but note that devices in promiscuous mode *must* be set up and configured outside of VMM. If all of the virtual machines 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 2-20, Tenant A has two virtual subnets. A virtual router automatically created by Windows Server 2012 Hyper-V 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.**

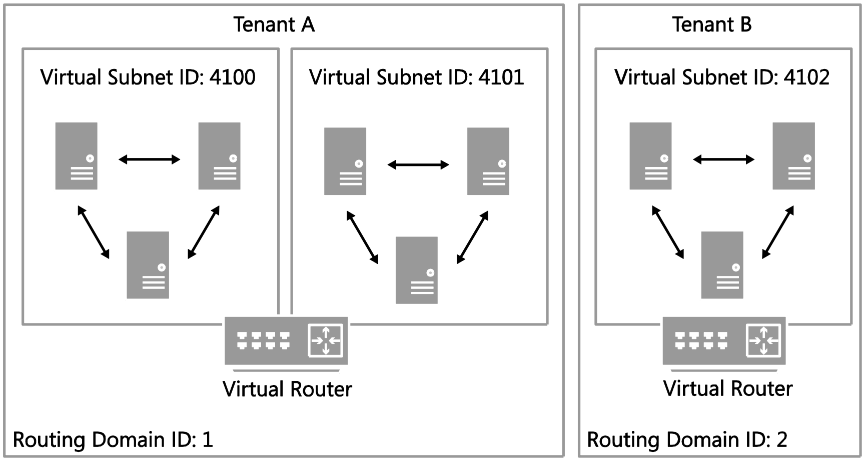
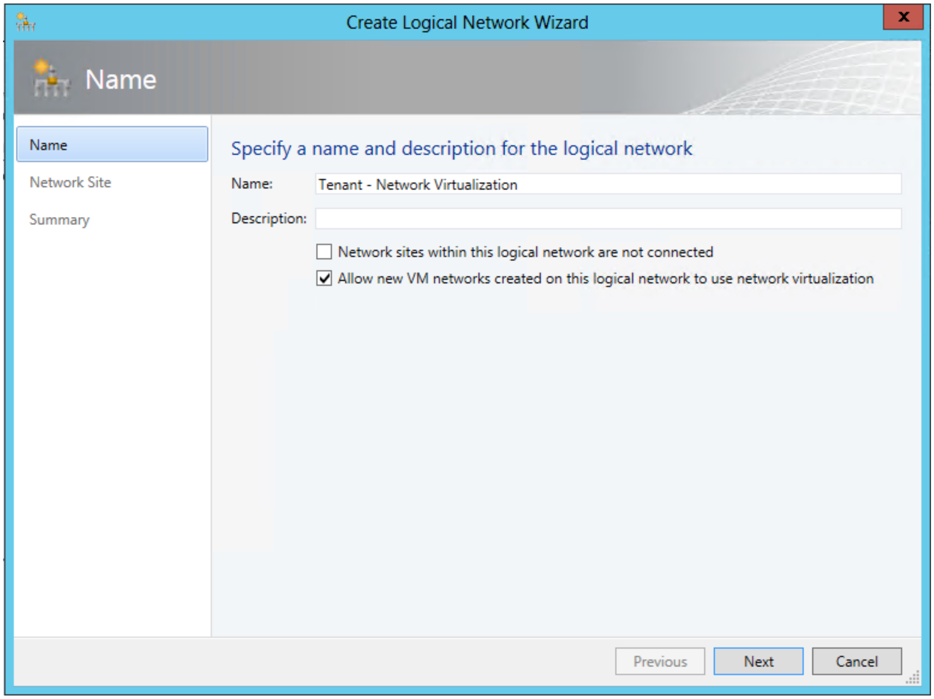


FIGURE 2-20 Logical network design for isolation using Network Virtualization.

When using Network Virtualization, the logical network design is relatively simple: create a single logical network for all of your customers that will be isolated from each other using Network Virtualization and configure the properties of the network with the Allow New VM Networks Created On This Network To Use Network Virtualization option, as shown in Figure 2-21.



Specify a name and description for the logical network

Name: Tenant - Network Virtualization

Description:

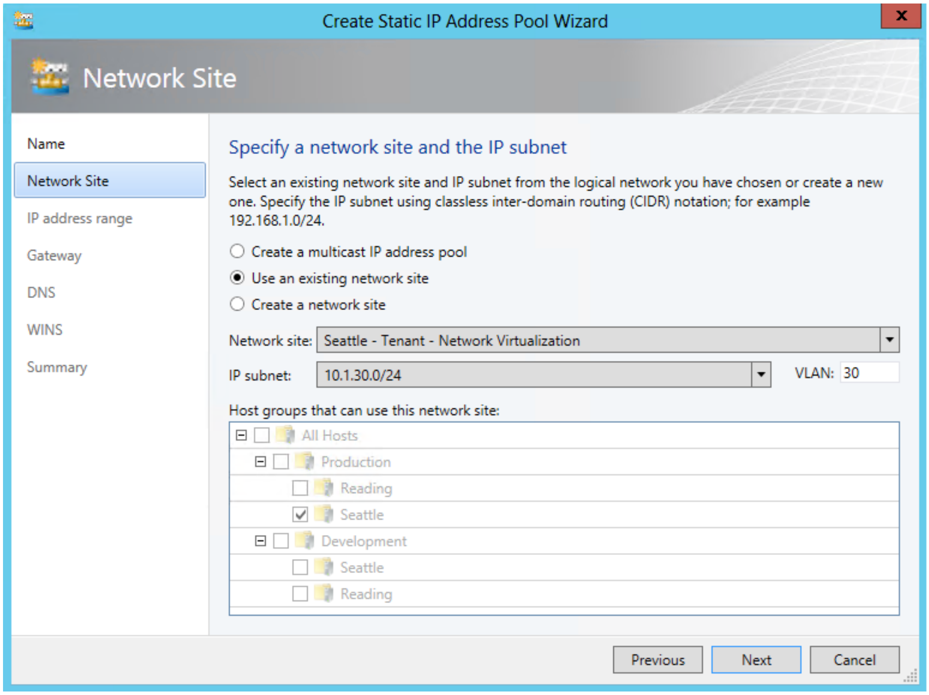
☐ Network sites within this logical network are not connected

☑ Allow new VM networks created on this logical network to use network virtualization

FIGURE 2-21 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.

Note that an IP pool must be associated with every single network site linked to the logical network, as shown in Figure 2-22. 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.



Specify a network site and the IP subnet

Select an existing network site and IP subnet from the logical network you have chosen or create a new one. Specify the IP subnet using classless inter-domain routing (CIDR) notation; for example 192.168.1.0/24.

○ Create a multicast IP address pool
● Use an existing network site
○ Create a network site

Network site: Seattle - Tenant - Network Virtualization

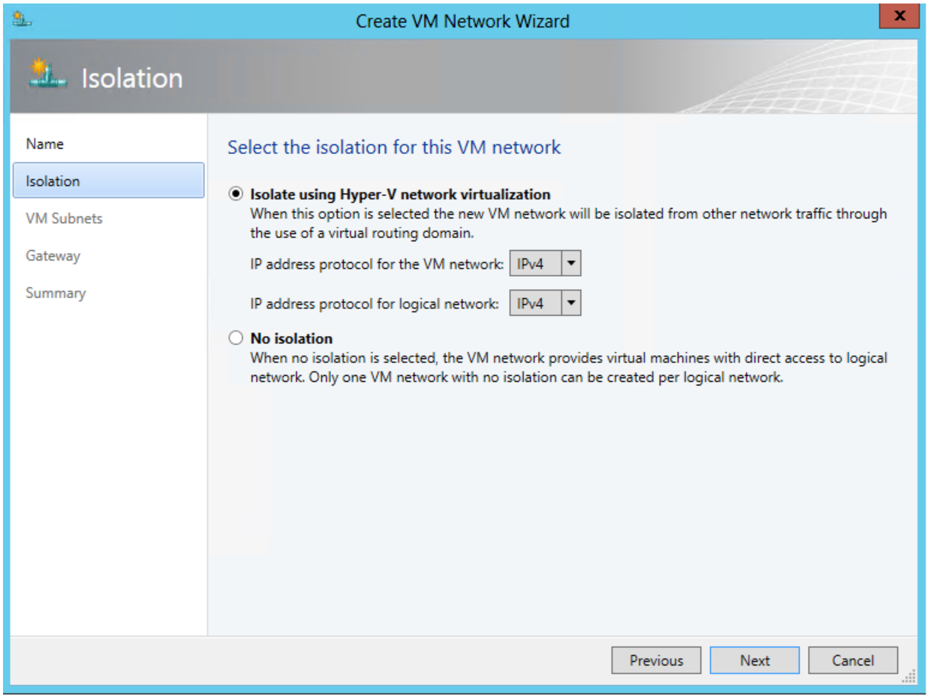
IP subnet: 10.1.30.0/24 VLAN: 30

Host groups that can use this network site:

- □ All Hosts
  - □ Production
    - □ Reading
    - ☑ Seattle
  - □ Development
    - □ Seattle
    - □ Reading

FIGURE 2-22 Defining a provider address IP pool for a network site.

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 2-23. You can also select No Isolation if you want the VM network to provide virtual machines with direct access to the logical network. Note that the option to enable isolation shown in Figure 2-23 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.



Select the isolation for this VM network

● **Isolate using Hyper-V network virtualization**
When this option is selected the new VM network will be isolated from other network traffic through the use of a virtual routing domain.

IP address protocol for the VM network: IPv4

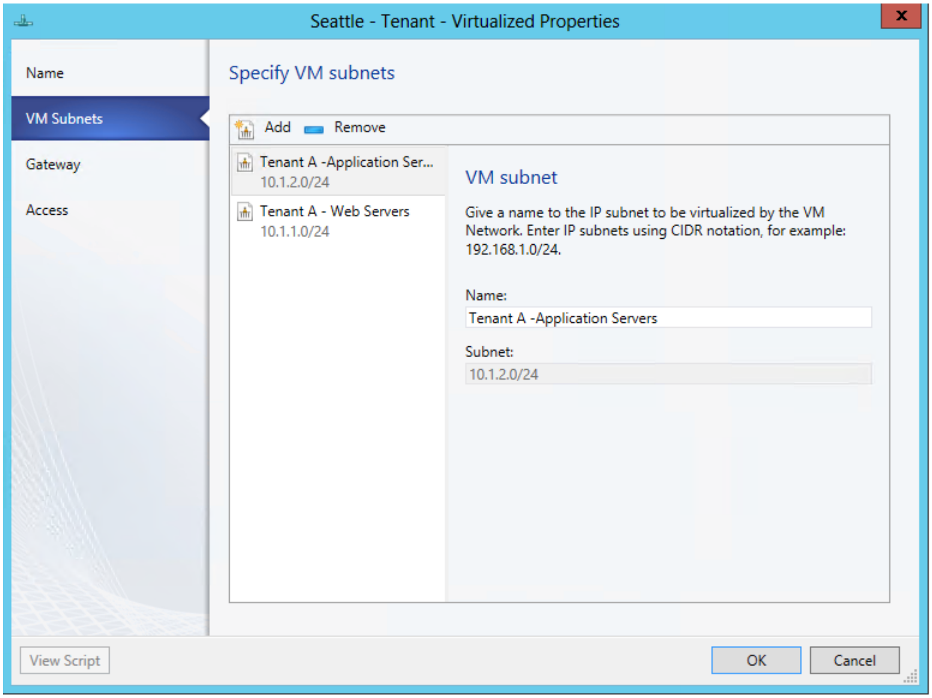
IP address protocol for logical network: IPv4

○ **No isolation**
When no isolation is selected, the VM network provides virtual machines with direct access to logical network. Only one VM network with no isolation can be created per logical network.

FIGURE 2-23 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 2-24. These addresses, known as the Consumer Address (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 2-24.

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.**



Specify VM subnets

Tenant A -Application Ser...
10.1.2.0/24

Tenant A - Web Servers
10.1.1.0/24

VM subnet

Give a name to the IP subnet to be virtualized by the VM Network. Enter IP subnets using CIDR notation, for example: 192.168.1.0/24.

Name: Tenant A -Application Servers

Subnet: 10.1.2.0/24

FIGURE 2-24 Defining consumer IP subnets.

To briefly summarize, create a single logical network for tenants that are to be isolated using Network Virtualization, configured with the option Allow New VM Networks Created On This 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 2-25.

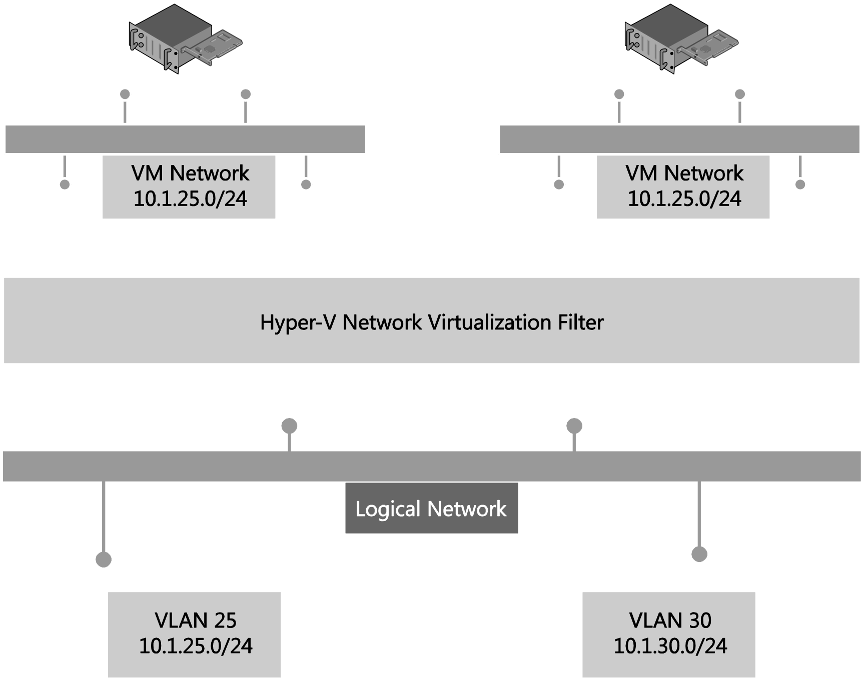


FIGURE 2-25 Logical network design for Network Virtualization.

The virtual networks shown in Figure 2-25 have no external connectivity by default, meaning that VMs connected to them will be able to communicate only with other virtual machines on the same virtual network. You can use a VPN gateway device to provide a VPN tunnel to a nominated external network or a Hyper-V Network Virtualization (HNV) gateway device to allow virtual machines on the virtual network to communicate with other networks in the local datacenter.

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 2-26.

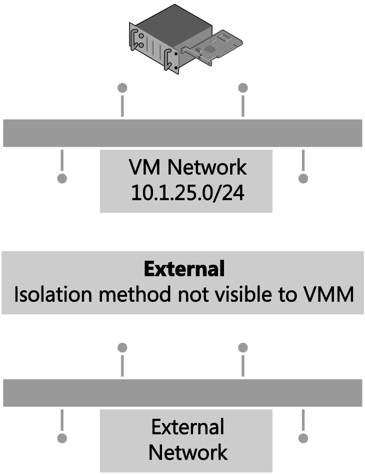


FIGURE 2-26 In externally defined networks, the isolation method is not visible to VMM.

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.

MORE INFO 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 Contoso, 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 Contoso 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 virtual machines 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 can acquire and deploy network gateways that support NVGRE and 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 (i.e., 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 recommend approach is to create a parent host group (for example Production as shown in Figure 2-27) 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 2-27, for example.

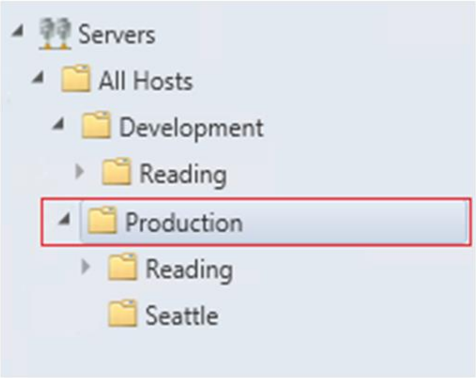


FIGURE 2-27 Creating host groups for dedicated servers.

Network adapters in Windows Server 2012 Hyper-V servers can be associated with multiple logical networks. However, there is no internal routing between them. If you want to allow virtual machines and host services configured on one logical network to communicate with those on another, you will need to deploy a router or gateway device.

Network sites (logical network definitions)

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 Dynamic Host Configuration Protocol (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 virtual machines, to virtual machines 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.

As with network sites, 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 virtual machines on the VM networks are configured to use DHCP, VMM will respond to the DHCP request with an address from an IP address pool.
- VLAN-based configuration: 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.
- VM network that gives direct access to the logical network ("no isolation"): 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.

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 addresses 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 5, "Deployment."

Naming conventions

As with everything else, defining and adhering to a naming convention for all the components 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 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 components 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 components of the solution in your environment and what they are used for.

CHAPTER 3

Port profiles

Port profiles (and logical switches) act as containers, essentially templates, for the properties and capabilities to be applied to network adapters. These tools allow you to consistently apply the same settings and capabilities to network adapters across multiple hosts. You can configure network adapter settings and capabilities on each host computer in your environment by making the necessary changes manually (or via Windows PowerShell), but if you've deployed Microsoft System Center 2012 SP1 Virtual Machine Manager (VMM), the recommended approach is use port profiles and logical switches to define the list of logical networks and the set of properties and capabilities you want to deploy.

This chapter will:

- Review the role of port profiles in a virtualized network solution
- Discuss the different types of port profiles and how they are used
- Explain how to identify the set of 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 port profiles in VMM: uplink port profiles and 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. 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 or a stretched or campus network with multiple physical locations, 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 profile may 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 components of the architecture.

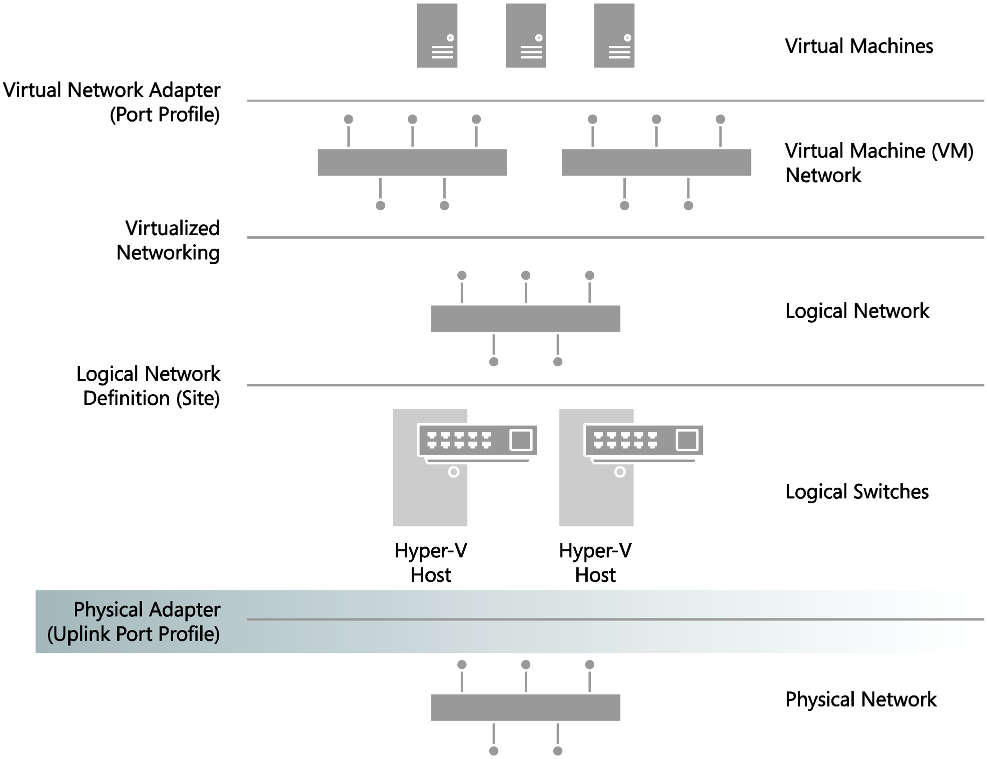

Virtual Machines
Virtual Network Adapter (Port Profile)
Virtual Machine (VM) Network
Virtualized Networking
Logical Network
Logical Network Definition (Site)
Logical Switches
Hyper-V Host
Hyper-V Host
Physical Adapter (Uplink Port Profile)
Physical Network

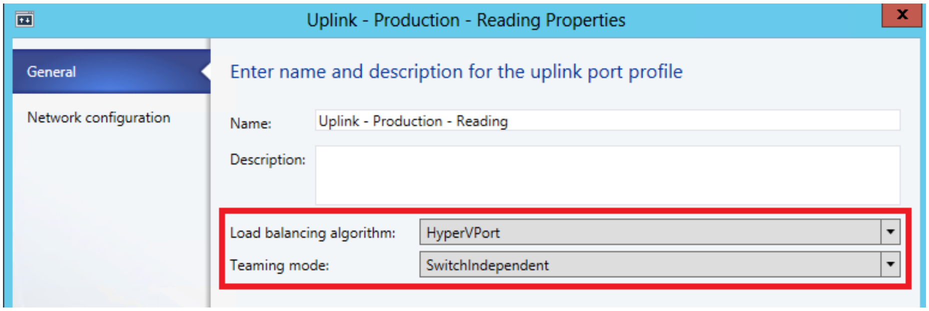
FIGURE 3-1 Position of uplink port profiles in the VMM network architecture.

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, load balancing protocols, or 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: HyperVPort for the load balancing algorithm, which distributes network traffic based on the Hyper-V switch port identifier of the source virtual machine, with SwitchIndependent 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 and configuration that will be most appropriate for the hosts and network adapters on which the uplink port profile will be applied.



Enter name and description for the uplink port profile

Name: Uplink - Production - Reading

Description:

Load balancing algorithm: HyperVPort

Teaming mode: SwitchIndependent

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

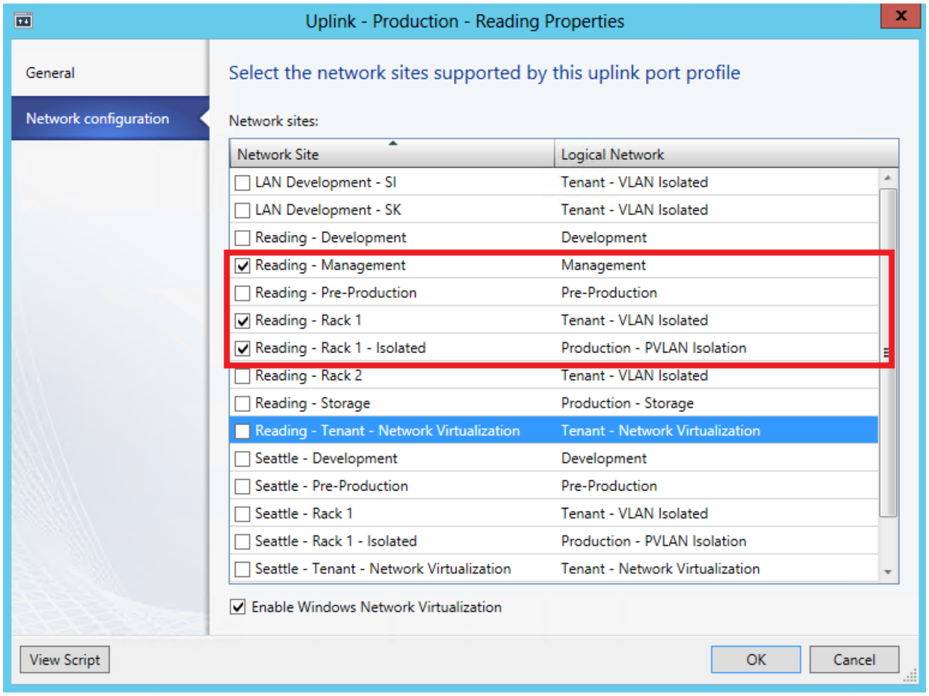
> ***MORE INFO*** **You can find more detailed information on the different load balancing and traffic distribution options and teaming modes that can be defined in an uplink port profile at *http://technet.microsoft.com/library/hh831648.aspx* and in the LBFO whitepaper at *http://www.microsoft.com/en-us/download/details.aspx?id=30160*.**

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 virtual machines (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.



Uplink - Production - Reading Properties

Select the network sites supported by this uplink port profile

Network sites:

Network Site	Logical Network
□ LAN Development - SI	Tenant - VLAN Isolated
□ LAN Development - SK	Tenant - VLAN Isolated
□ Reading - Development	Development
☑ Reading - Management	Management
□ Reading - Pre-Production	Pre-Production
☑ Reading - Rack 1	Tenant - VLAN Isolated
☑ Reading - Rack 1 - Isolated	Production - PVLAN Isolation
□ Reading - Rack 2	Tenant - VLAN Isolated
□ Reading - Storage	Production - Storage
□ Reading - Tenant - Network Virtualization	Tenant - Network Virtualization
□ Seattle - Development	Development
□ Seattle - Pre-Production	Pre-Production
□ Seattle - Rack 1	Tenant - VLAN Isolated
□ Seattle - Rack 1 - Isolated	Production - PVLAN Isolation
□ Seattle - Tenant - Network Virtualization	Tenant - Network Virtualization

☑ Enable Windows Network Virtualization

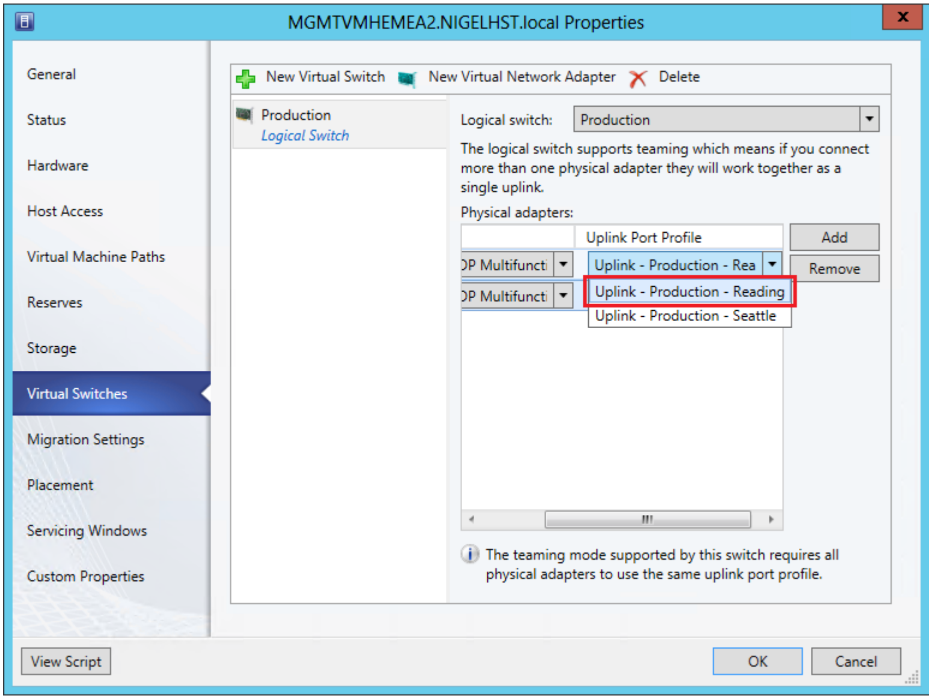
FIGURE 3-3 Network sites in an uplink port profile.

As the example in Figure 3-3 shows, it is reasonable for a VMM administrator 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 - Management, 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 refer 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. The problem, if you are not using VLAN isolation is that 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 may 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.



Logical switch: Production

The logical switch supports teaming which means if you connect more than one physical adapter they will work together as a single uplink.

Physical adapters:

	Uplink Port Profile
DP Multifuncti	Uplink - Production - Rea
DP Multifuncti	Uplink - Production - Reading Uplink - Production - Seattle

The teaming mode supported by this switch requires all physical adapters to use the same uplink port profile.

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 is 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 were 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 Contoso may 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 may 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 can 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 Contoso, 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 Contoso has more than one physical location, a minimum of three uplink port profiles would be required to support this environment (see Figure 3-5).

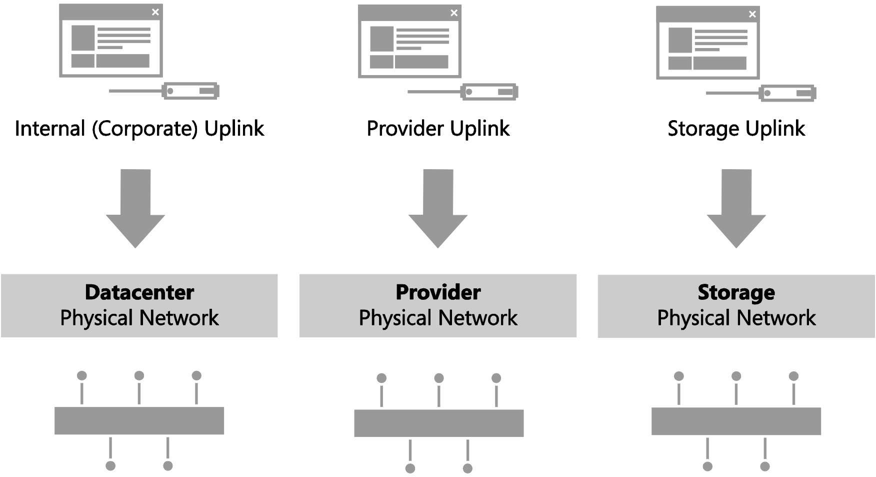


FIGURE 3-5 One uplink port profile for each physical network.

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 virtual machines 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 Contoso, 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, virtual machines 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 VLAN18 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.

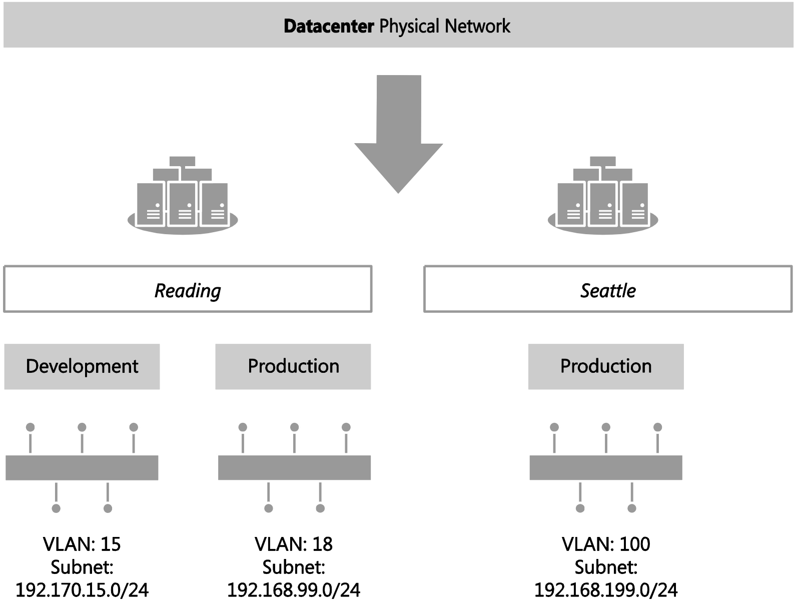


FIGURE 3-6 Workloads differentiated by VLAN and IP subnet.

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.

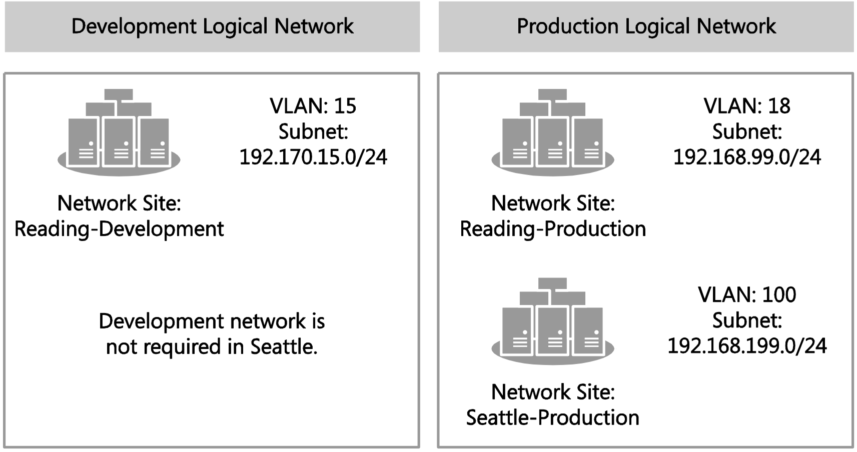


FIGURE 3-7 Network sites in a logical network.

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 Contoso 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 Contoso 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).

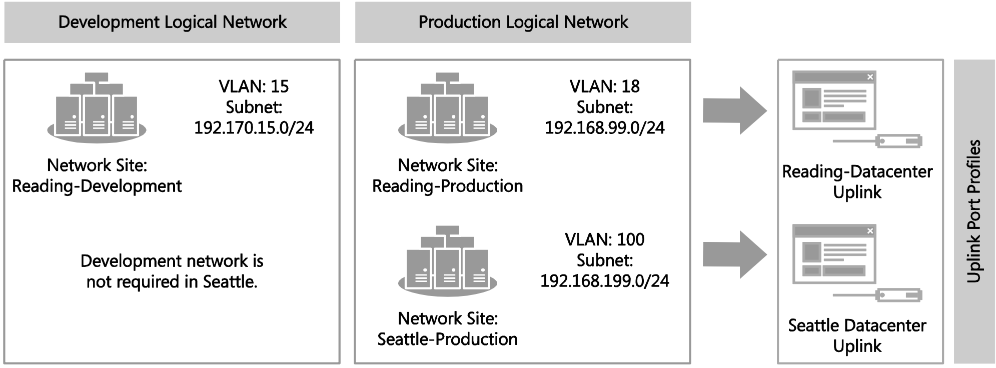


FIGURE 3-8 Defining uplink port profiles for each physical site.

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 Contoso 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 Contoso, 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).

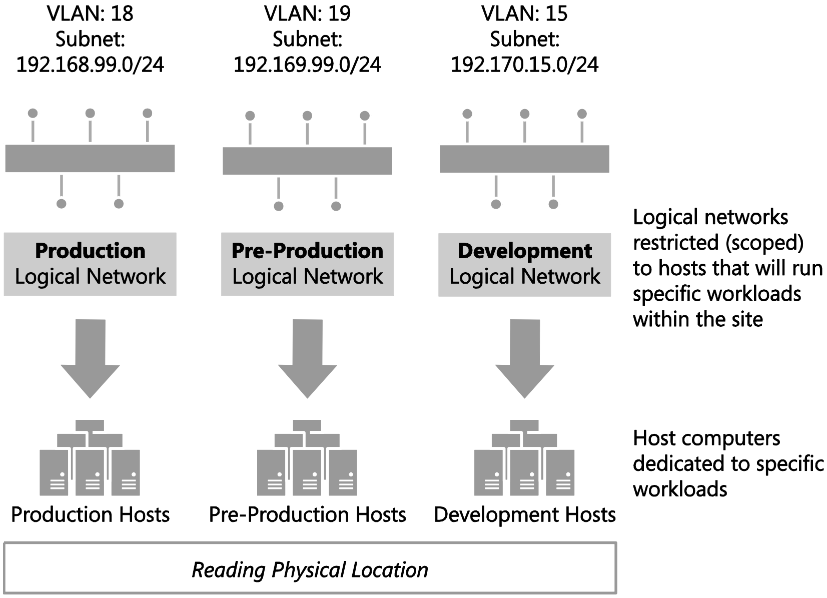


FIGURE 3-9 Associating logical networks with dedicated servers.

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. Contoso 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 may 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.

Contoso 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).

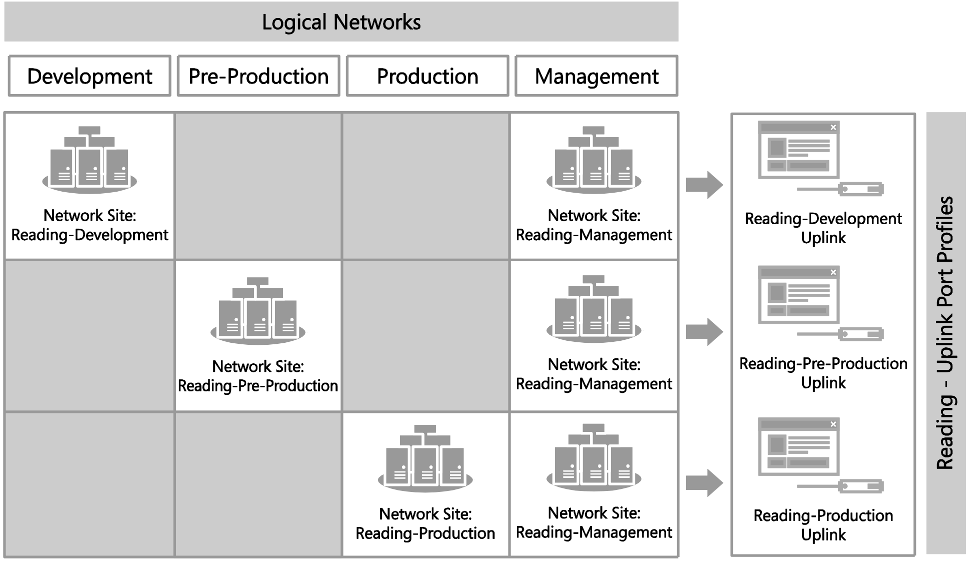

Logical Networks
Development
Pre-Production
Production
Management
Network Site: Reading-Development
Network Site: Reading-Management
Network Site: Reading-Pre-Production
Network Site: Reading-Management
Network Site: Reading-Production
Network Site: Reading-Management
Reading-Development Uplink
Reading-Pre-Production Uplink
Reading-Production Uplink
Reading - Uplink Port Profiles

FIGURE 3-10 Define uplink port profiles for dedicated resources.

NOTE **Although multiple logical networks may associated with a host network adapter using a single uplink port profile, each of these networks will be isolated from each other. If you wish to allow virtual machines 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 may therefore be necessary to place these host computers in different (routed) IP subnets.

At Contoso, 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, virtual machines, 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.

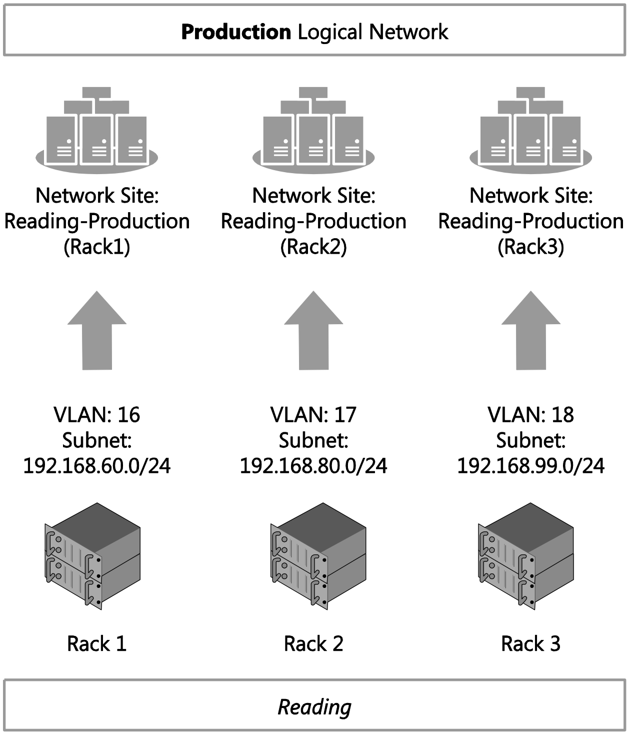

Production Logical Network
Network Site: Reading-Production (Rack1)
Network Site: Reading-Production (Rack2)
Network Site: Reading-Production (Rack3)
VLAN: 16 Subnet: 192.168.60.0/24
VLAN: 17 Subnet: 192.168.80.0/24
VLAN: 18 Subnet: 192.168.99.0/24
Rack 1
Rack 2
Rack 3
Reading

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.**

Because all three of these network sites cannot be placed into a single uplink port profile since doing so means 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 Contoso 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 Contoso 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.

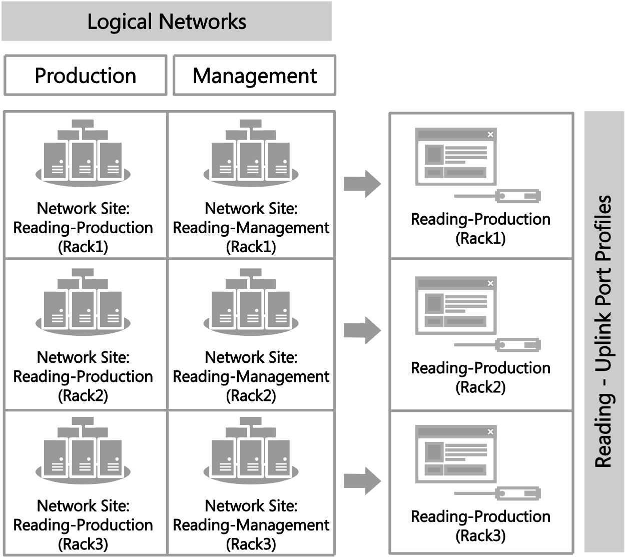

Logical Networks
Production
Management
Network Site: Reading-Production (Rack1)
Network Site: Reading-Management (Rack1)
Reading-Production (Rack1)
Network Site: Reading-Production (Rack2)
Network Site: Reading-Management (Rack2)
Reading-Production (Rack2)
Network Site: Reading-Production (Rack3)
Network Site: Reading-Management (Rack3)
Reading-Production (Rack3)
Reading - Uplink Port Profiles

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 may 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 may require little or no further refinement, as illustrated in Figure 3-13.

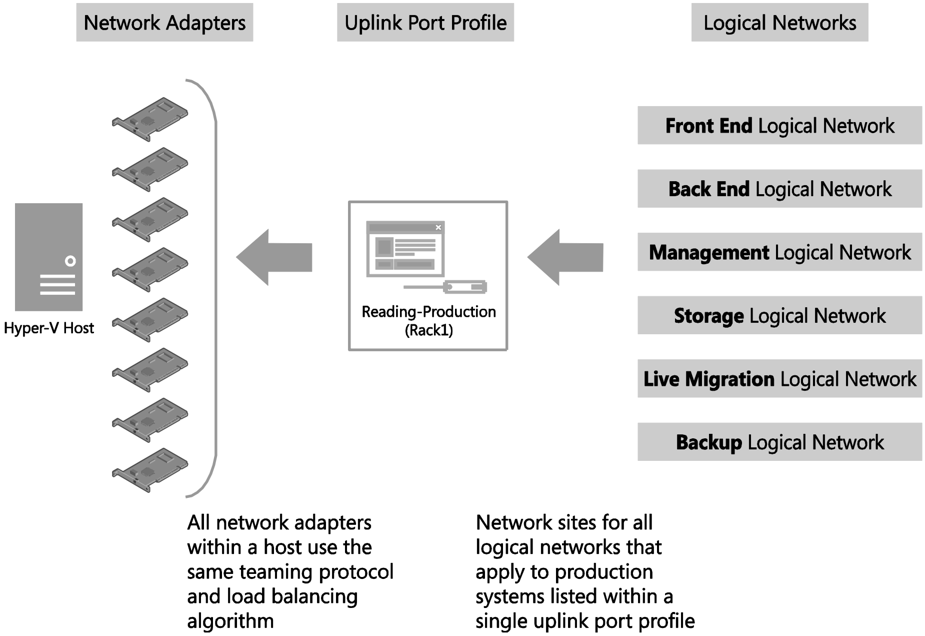

Network Adapters
Uplink Port Profile
Logical Networks
Hyper-V Host
Reading-Production (Rack1)
Front End Logical Network
Back End Logical Network
Management Logical Network
Storage Logical Network
Live Migration Logical Network
Backup Logical Network
All network adapters within a host use the same teaming protocol and load balancing algorithm
Network sites for all logical networks that apply to production systems listed within a single uplink port profile

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 may be dedicated to specific functions like storage, host management, or tenant traffic, each of which may require different teaming modes and load balancing algorithms. Hosts may even contain a mixture of standard and specialist network capabilities, such as RDMA and SR-IOV, which need to be managed differently. In the case of RDMA and SR-IOV, for example, network workloads are offloaded to the physical network adapter there bypassing the networking stack, and so it is not possible for the adapter team (which is part of the networking stack) to look at or redirect the data to another path.

NOTE **VMM does not specifically recognize an RDMA network adapter installed in a host computer, and there are no settings or configuration in VMM for RDMA. If you are using RDMA network adapters, you should not create uplink port profiles or deploy a logical switch on these 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 Contoso, 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 Contoso'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 take account of 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.

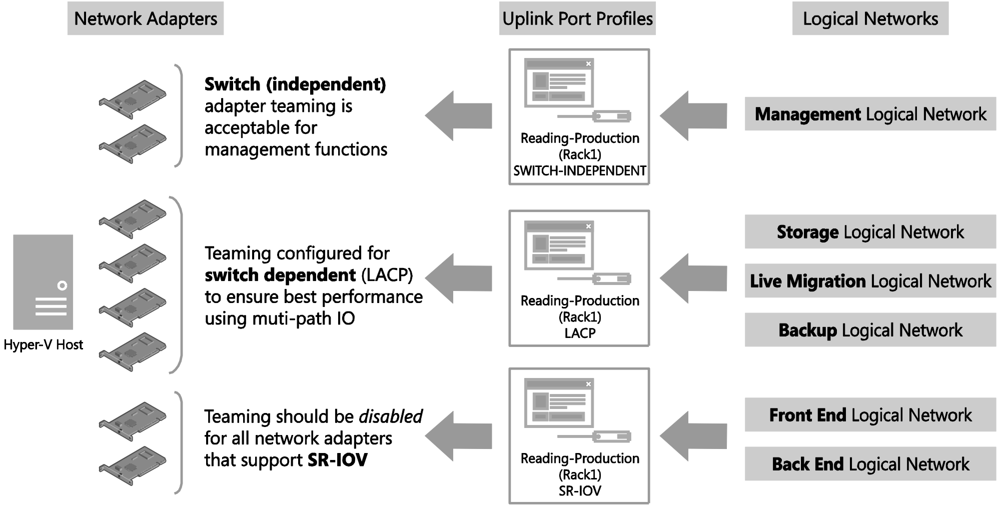
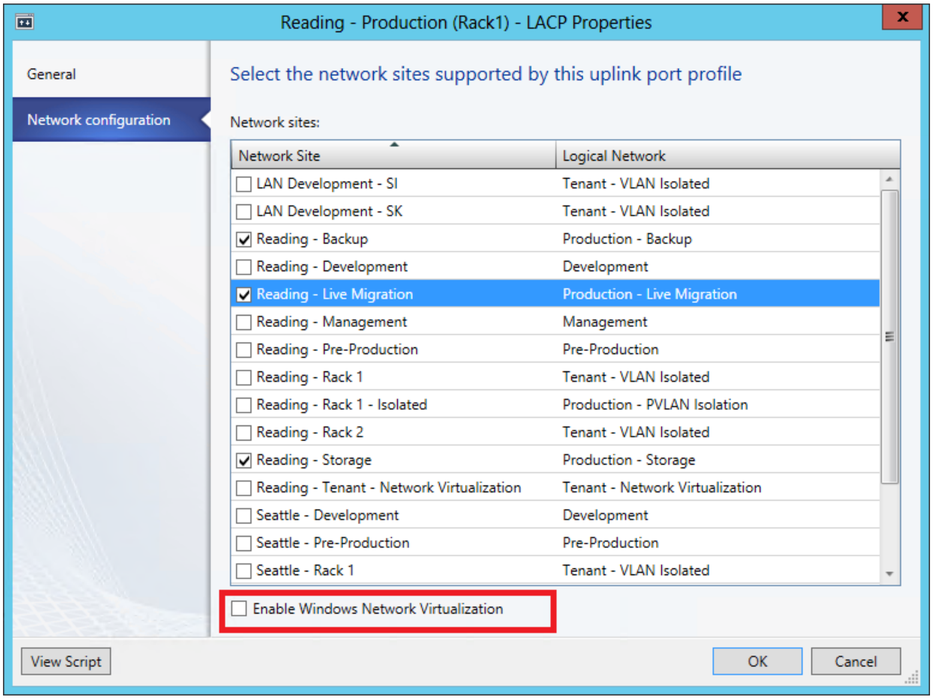

Network Adapters
Uplink Port Profiles
Logical Networks
Switch (independent) adapter teaming is acceptable for management functions
Reading-Production (Rack1) SWITCH-INDEPENDENT
Management Logical Network
Hyper-V Host
Teaming configured for switch dependent (LACP) to ensure best performance using muti-path IO
Reading-Production (Rack1) LACP
Storage Logical Network
Live Migration Logical Network
Backup Logical Network
Teaming should be disabled for all network adapters that support SR-IOV
Reading-Production (Rack1) SR-IOV
Front End Logical Network
Back End Logical Network

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.

Disabling Network Virtualization

Having refined the uplink port profile design, you may find a number of uplink port profiles contain network sites for logical networks that are used only by the host or that are required to manage the host. In such cases, it may be useful to clear the Enable Windows Network Virtualization option (shown in Figure 3-15) Deselecting the Enable Windows Network Virtualization option provides a useful scoping mechanism, helping to clarify which uplinks are designed primarily for infrastructure services and host management and which are designed for use by end user VMs.



Select the network sites supported by this uplink port profile

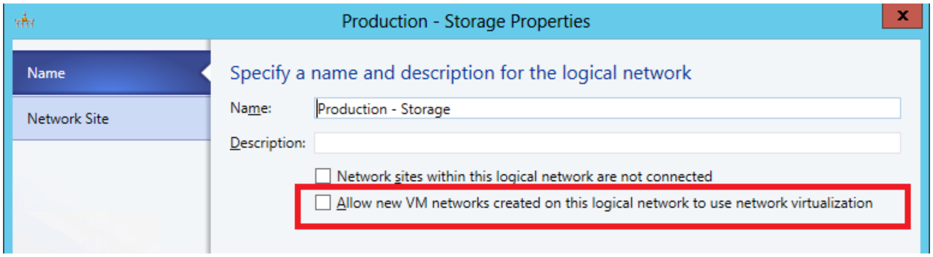
Network sites:

Network Site	Logical Network
☐ LAN Development - SI	Tenant - VLAN Isolated
☐ LAN Development - SK	Tenant - VLAN Isolated
☑ Reading - Backup	Production - Backup
☐ Reading - Development	Development
☑ Reading - Live Migration	Production - Live Migration
☐ Reading - Management	Management
☐ Reading - Pre-Production	Pre-Production
☐ Reading - Rack 1	Tenant - VLAN Isolated
☐ Reading - Rack 1 - Isolated	Production - PVLAN Isolation
☐ Reading - Rack 2	Tenant - VLAN Isolated
☑ Reading - Storage	Production - Storage
☐ Reading - Tenant - Network Virtualization	Tenant - Network Virtualization
☐ Seattle - Development	Development
☐ Seattle - Pre-Production	Pre-Production
☐ Seattle - Rack 1	Tenant - VLAN Isolated

☐ Enable Windows Network Virtualization

FIGURE 3-15 Disabling Network Virtualization in an uplink port profile.

With that said, since Network Virtualization service adds very little overhead to the host, it may be preferable to leave this setting on even in uplink port profiles that contain only infrastructure services, just for consistency. You can instead disable Network Virtualization at the logical network layer (shown in Figure 3-16).



Specify a name and description for the logical network

Name: Production - Storage

Description:

☐ Network sites within this logical network are not connected

☐ Allow new VM networks created on this logical network to use network virtualization

FIGURE 3-16 Disabling Network Virtualization in the logical network layer.

Ultimately, the choice of where to disable Network Virtualization comes down to whether you have uplink port profiles dedicated to management and infrastructure, in which case, either approach will be satisfactory. If, on the other hand, some uplink port profiles contain network sites from a mixture of infrastructure, management, and tenant or end user logical networks, the only option is to leave the Network Virtualization option enabled in the uplink port profile and disable it on those logical networks used for management and infrastructure.

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]

Where typical examples of this structure would be:

Reading - Production (Rack1) - LACP

Reading - Production (Rack1) - SRIOV

Reading - Development - NoTeam

This particular structure may 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.

Network adapter port profiles

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.

Unlike uplink port profiles, a number of example 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 3-17 illustrates the different layers that make up the architecture of a virtualized networking solution, highlighting where network adapter port profiles connect to other components of 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).

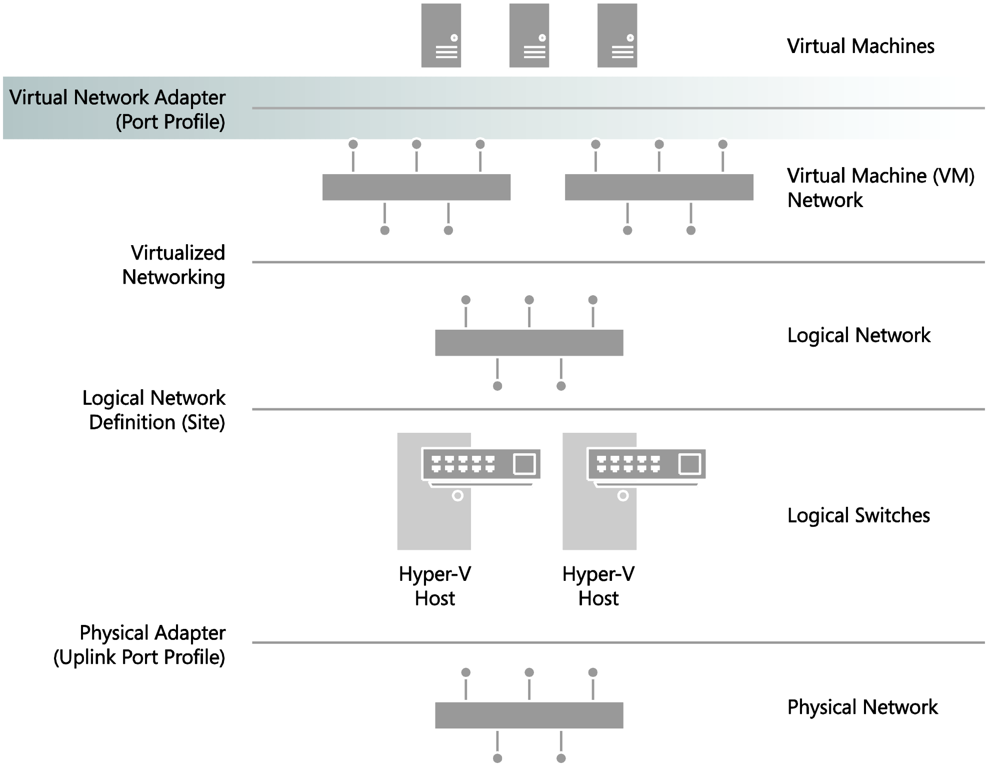

Virtual Machines
Virtual Network Adapter (Port Profile)
Virtual Machine (VM) Network
Virtualized Networking
Logical Network
Logical Network Definition (Site)
Logical Switches
Hyper-V Host
Hyper-V Host
Physical Adapter (Uplink Port Profile)
Physical Network

FIGURE 3-17 Architecture showing network adapter port profiles.

What is defined in a network adapter port profile?

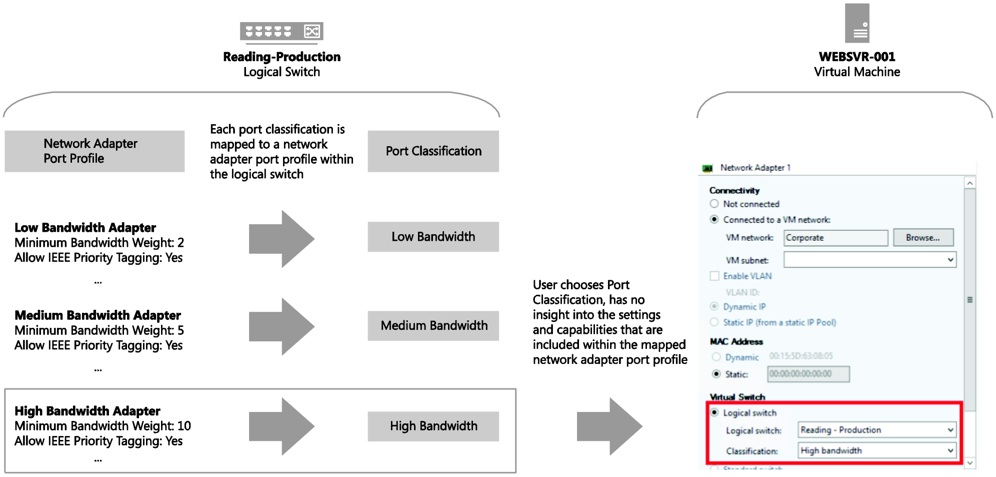
Network adapter port profiles define the processor offload settings, security settings, and bandwidth limitations to be enforced on network adapters within a guest VM or virtual network interface cards (vNICs) on which they are applied. You can find more detail on the range of different settings and 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 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 3-18, the Reading-Production logical switch contains a number of network adapter port profiles and port classifications. One of the network adapters in virtual machine 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.



Reading-Production
Logical Switch
WEBSVR-001
Virtual Machine
Network Adapter Port Profile
Each port classification is mapped to a network adapter port profile within the logical switch
Port Classification
Low Bandwidth Adapter
Minimum Bandwidth Weight: 2
Allow IEEE Priority Tagging: Yes
...
Low Bandwidth
Medium Bandwidth Adapter
Minimum Bandwidth Weight: 5
Allow IEEE Priority Tagging: Yes
...
Medium Bandwidth
High Bandwidth Adapter
Minimum Bandwidth Weight: 10
Allow IEEE Priority Tagging: Yes
...
High Bandwidth
User chooses Port Classification, has no insight into the settings and capabilities that are included within the mapped network adapter port profile

Network Adapter 1

Connectivity

○ Not connected

● Connected to a VM network:

VM network: Corporate [Browse...]

VM subnet:

□ Enable VLAN

VLAN ID:

● Dynamic IP

○ Static IP (from a static IP Pool)

MAC Address

○ Dynamic 00:15:5D:63:08:05

● Static: 00:00:00:00:00:00

Virtual Switch

● Logical switch

Logical switch: Reading - Production

Classification: High bandwidth

FIGURE 3-18 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 network adapter port profiles do you need?

Multiple 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 components 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 Contoso, a number of different types of workload have been identified, as shown in Figure 3-19, 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.

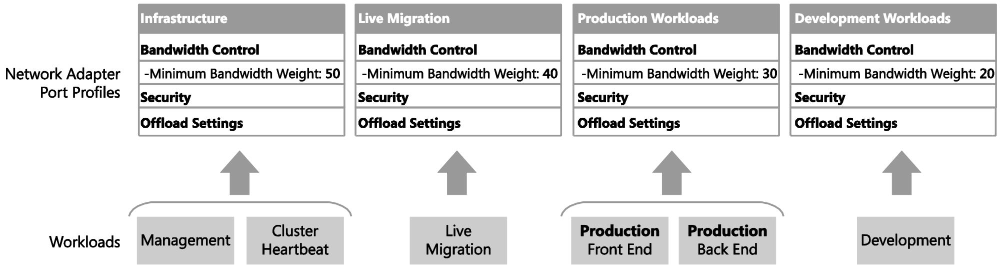


FIGURE 3-19 A network adapter port profile for each prioritized workload.

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 may provide more granular control, but relative priority is recommended since it allows you to make use of all available bandwidth.

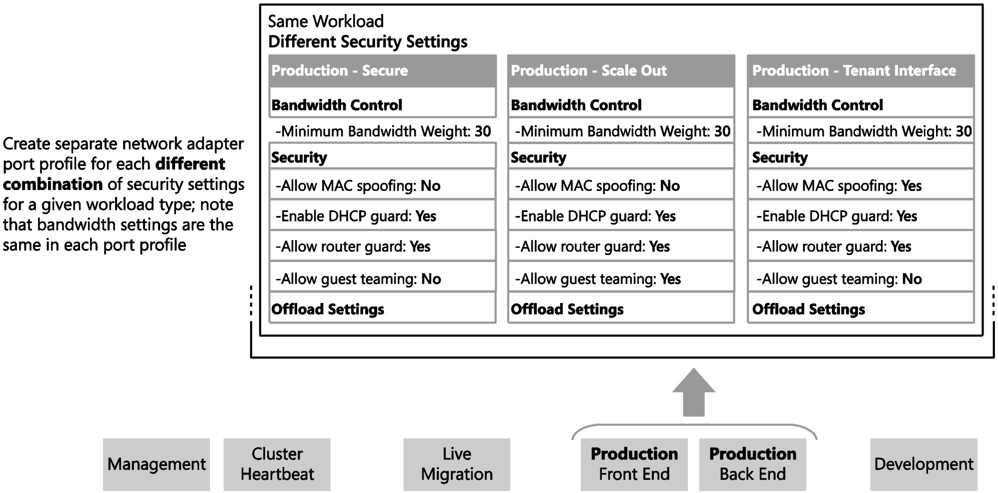
> ***NOTE*** **A given logical switch will be in either Relative Weights or Absolute Bandwidth Values mode with respect to bandwidth control. The default is Relative Weights. If some workloads will use relative weights and others will use fixed bandwidth limits, you need to create separate network adapter 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 Contoso, 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 3-20).



Create separate network adapter port profile for each **different combination** of security settings for a given workload type; note that bandwidth settings are the same in each port profile

Same Workload
Different Security Settings

Production - Secure	Production - Scale Out	Production - Tenant Interface
Bandwidth Control	**Bandwidth Control**	**Bandwidth Control**
-Minimum Bandwidth Weight: **30**	-Minimum Bandwidth Weight: **30**	-Minimum Bandwidth Weight: **30**
Security	**Security**	**Security**
-Allow MAC spoofing: **No**	-Allow MAC spoofing: **No**	-Allow MAC spoofing: **Yes**
-Enable DHCP guard: **Yes**	-Enable DHCP guard: **Yes**	-Enable DHCP guard: **Yes**
-Allow router guard: **Yes**	-Allow router guard: **Yes**	-Allow router guard: **Yes**
-Allow guest teaming: **No**	-Allow guest teaming: **Yes**	-Allow guest teaming: **No**
Offload Settings	**Offload Settings**	**Offload Settings**

FIGURE 3-20 Refined solution for workloads with different security settings.

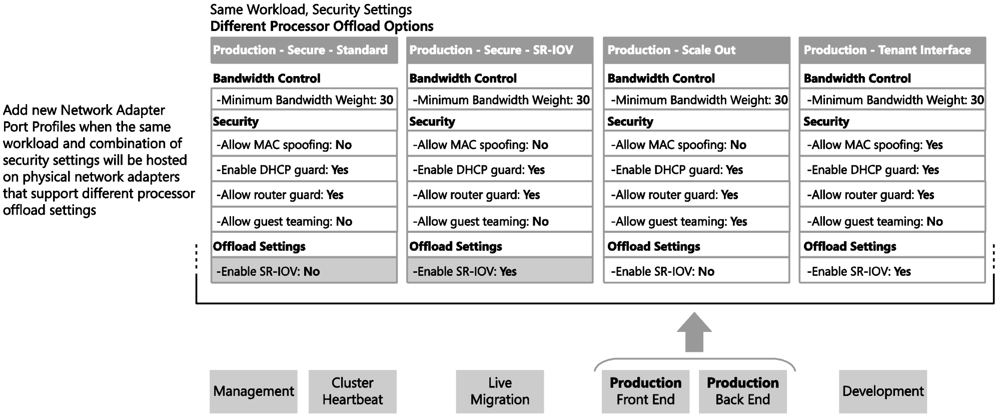
Having identified the range of security settings required for one type of workload, you should review each of the others and repeat the process, creating additional network adapter port profiles as you identify different security requirements.

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 may 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 Contoso, 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 3-21). 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.


Same Workload, Security Settings
Different Processor Offload Options
Add new Network Adapter Port Profiles when the same workload and combination of security settings will be hosted on physical network adapters that support different processor offload settings
Management
Cluster Heartbeat
Live Migration
Production Front End
Production Back End
Development

Production - Secure - Standard	Production - Secure - SR-IOV	Production - Scale Out	Production - Tenant Interface
Bandwidth Control	**Bandwidth Control**	**Bandwidth Control**	**Bandwidth Control**
-Minimum Bandwidth Weight: 30	-Minimum Bandwidth Weight: 30	-Minimum Bandwidth Weight: 30	-Minimum Bandwidth Weight: 30
Security	**Security**	**Security**	**Security**
-Allow MAC spoofing: **No**	-Allow MAC spoofing: **No**	-Allow MAC spoofing: **No**	-Allow MAC spoofing: **Yes**
-Enable DHCP guard: **Yes**	-Enable DHCP guard: **Yes**	-Enable DHCP guard: **Yes**	-Enable DHCP guard: **Yes**
-Allow router guard: **Yes**	-Allow router guard: **Yes**	-Allow router guard: **Yes**	-Allow router guard: **Yes**
-Allow guest teaming: **No**	-Allow guest teaming: **No**	-Allow guest teaming: **Yes**	-Allow guest teaming: **No**
Offload Settings	**Offload Settings**	**Offload Settings**	**Offload Settings**
-Enable SR-IOV: **No**	-Enable SR-IOV: **Yes**	-Enable SR-IOV: **No**	-Enable SR-IOV: **Yes**

FIGURE 3-21 Adding profiles for network adapters that support offloading.

> ***NOTE*** **Enabling support in the network adapter port profile may not be sufficient for certain processor offload modes. It may 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]

Where typical examples of this structure would be:

Infrastructure

Production - Secure

Production - Secure - SR-IOV

Development

Test

This particular structure may 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 network adapter port profiles you have created and what they are used for.

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 will consider what happens in the event of a 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
- 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 host

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 which 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 components of the architecture.

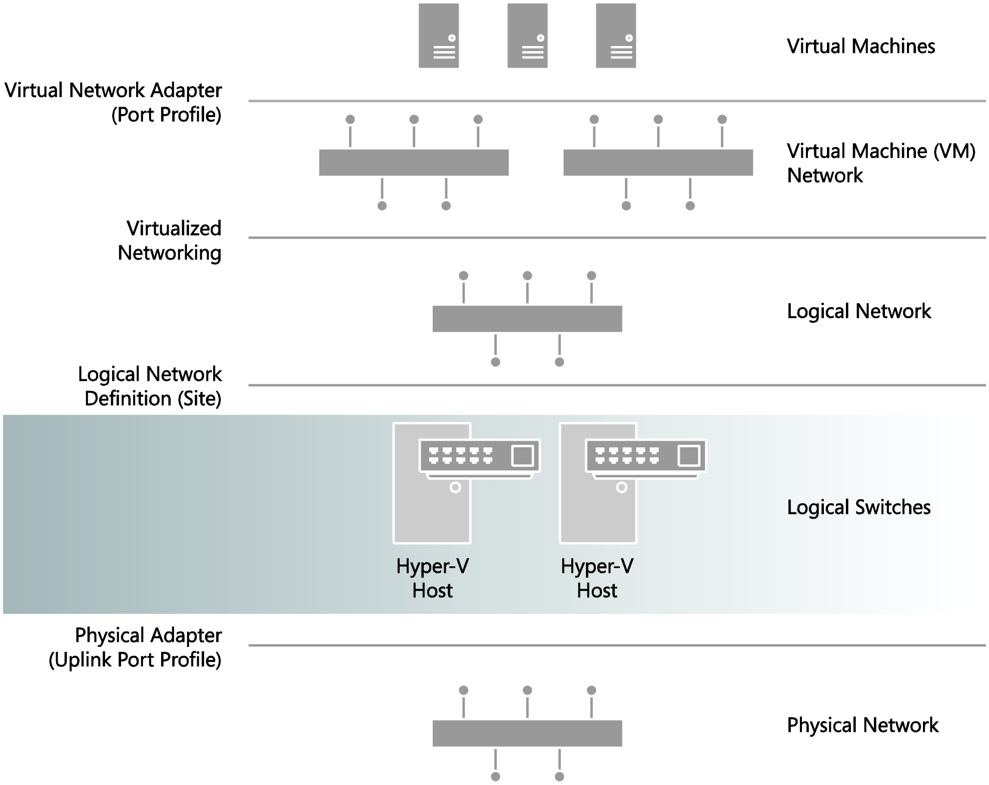

Virtual Machines
Virtual Network Adapter (Port Profile)
Virtual Machine (VM) Network
Virtualized Networking
Logical Network
Logical Network Definition (Site)
Logical Switches
Hyper-V Host
Hyper-V Host
Physical Adapter (Uplink Port Profile)
Physical Network

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

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 Contoso, the example organization introduced in previous chapters, and/or have custom connectivity requirements, you may find it necessary to create additional logical switches.

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.

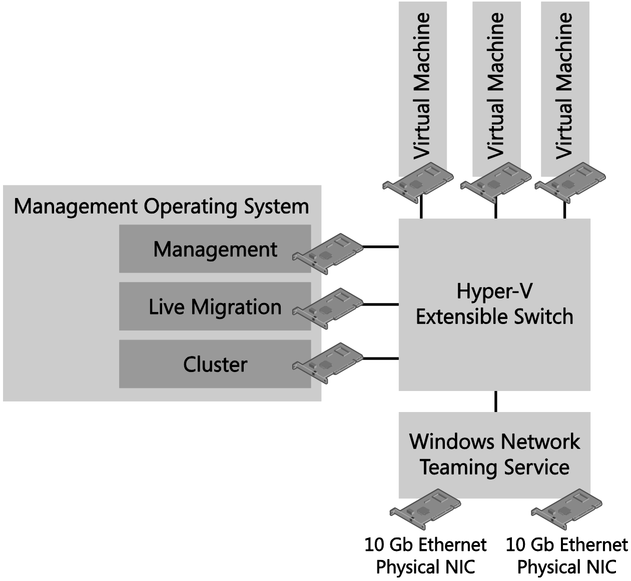


FIGURE 4-2 Sample network configuration for a Hyper-V host, fully converged.

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.

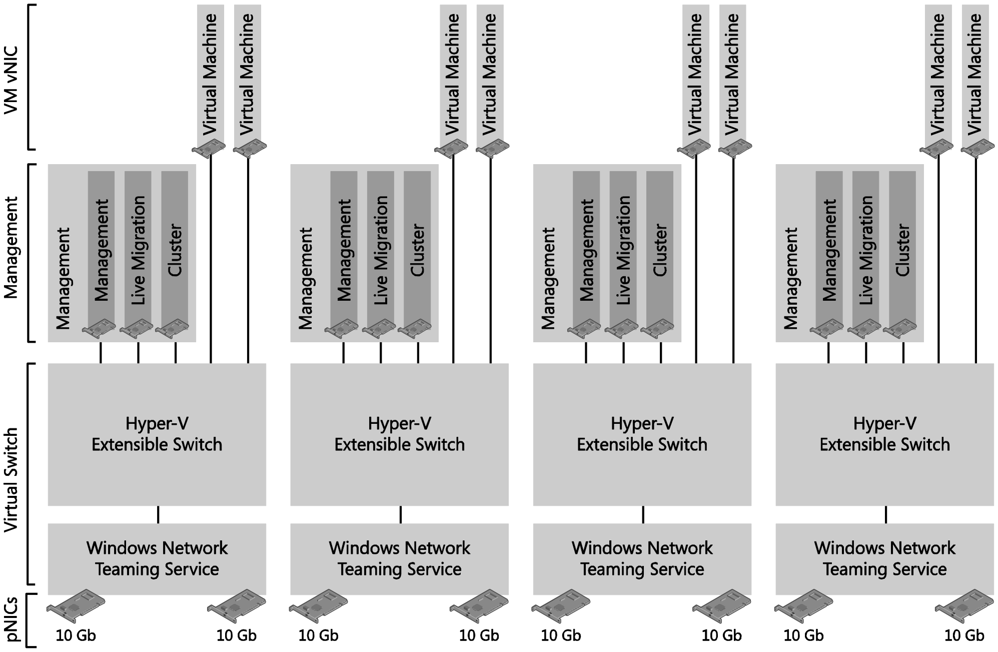


FIGURE 4-3 Network configuration must be implemented identically to scale hosts.

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.

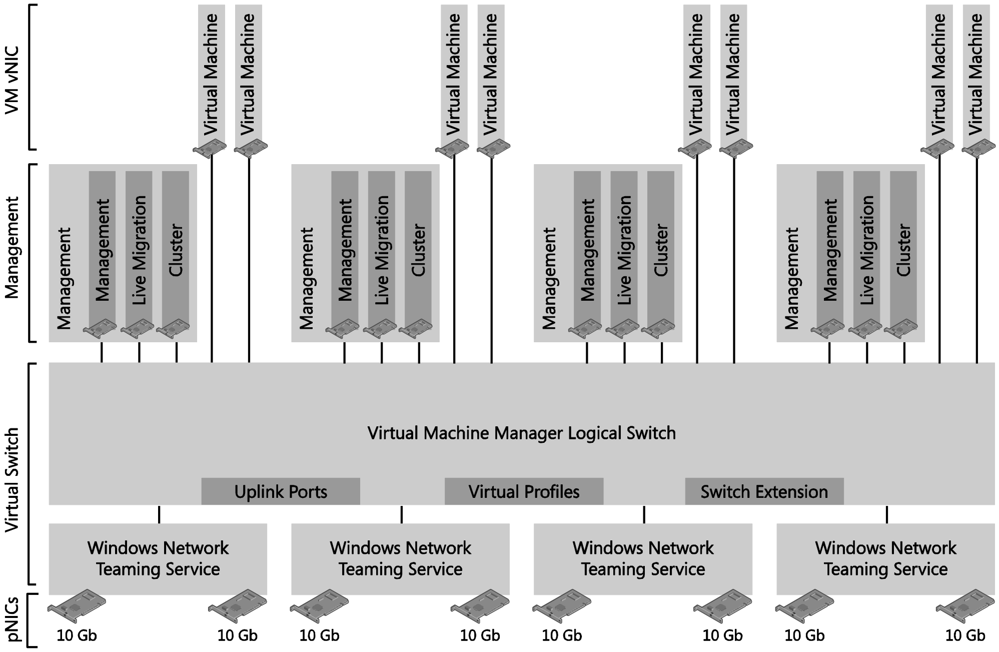


FIGURE 4-4 One logical switch configuration replacing multiple independent configurations.

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 component utilized by Hyper-V to process network traffic.

When a logical switch is applied to a network adapter on a Hyper-V host, 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 and associate the network adapter 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 now 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 host's 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.

Planning your logical switch design

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

Upgrading from Hyper-V Server 2008

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 VM traffic processed through the virtual switch. Additionally, the design would include a connection to the storage environment, possibly utilizing a pair of fiber 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.

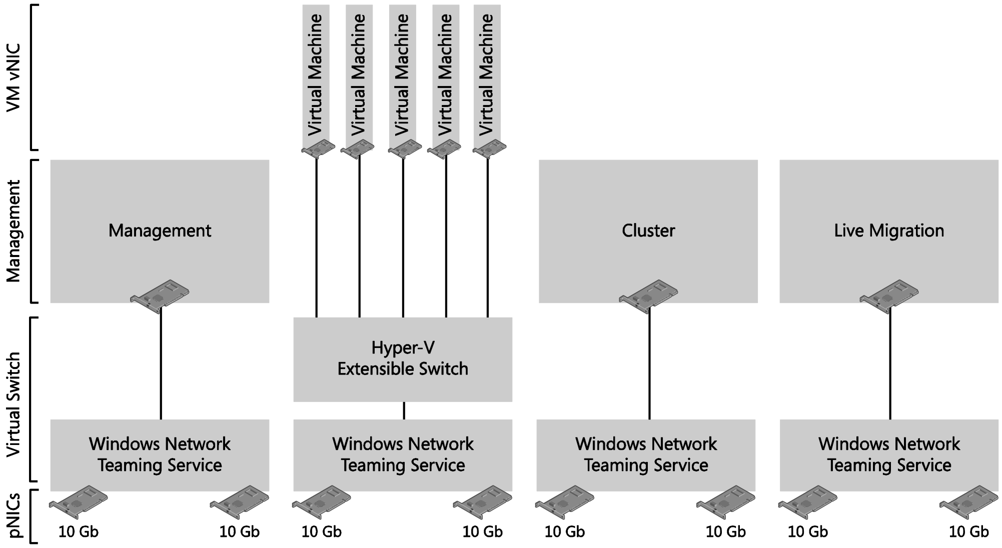


FIGURE 4-5 A typical network configuration for Hyper-V Server 2008 with dedicated teams.

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.

Quality of Service (QoS)

By using 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 which 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. Due to this adoptability and flexibility, the relative approach is normally preferred over the absolute approach.

A virtual switch can work in only one mode at a time (absolute or relative) and by default a logical switch will function in relative mode (weights) rather than absolute. You can override this however by using the following Windows PowerShell command:

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

It is not possible to change the mode though the VMM console.

> ***NOTE*** **In cases where you need to support both a relative and an absolute QoS configuration, you will need to use different logical switches.**

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, but you should note that placement will block live migration across logical switches.

To more closely consider these approaches and the values each offers, consider a simple example of a pair of 10-GB NICs, configured in a team. Figure 4-8 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 gigabytes 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 4-6).

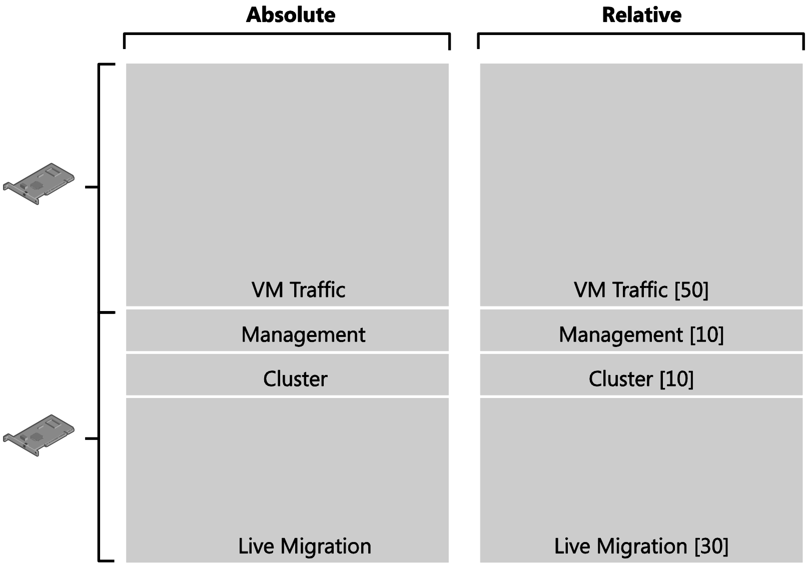


FIGURE 4-6 Distributing 20 gigabytes of bandwidth using absolute and relative weighting.

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 on the other hard 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 4-7 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.

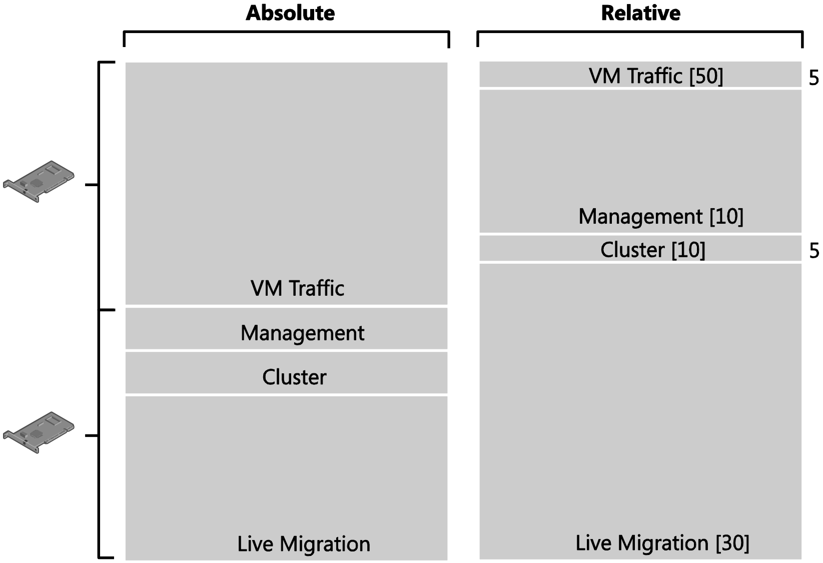


FIGURE 4-7 Adding minimum bandwidth to relative weights.

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.
  - 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:

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

- Live Migration, minimum of 0 GB, maximum of 13.5 GB, 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 4-8).

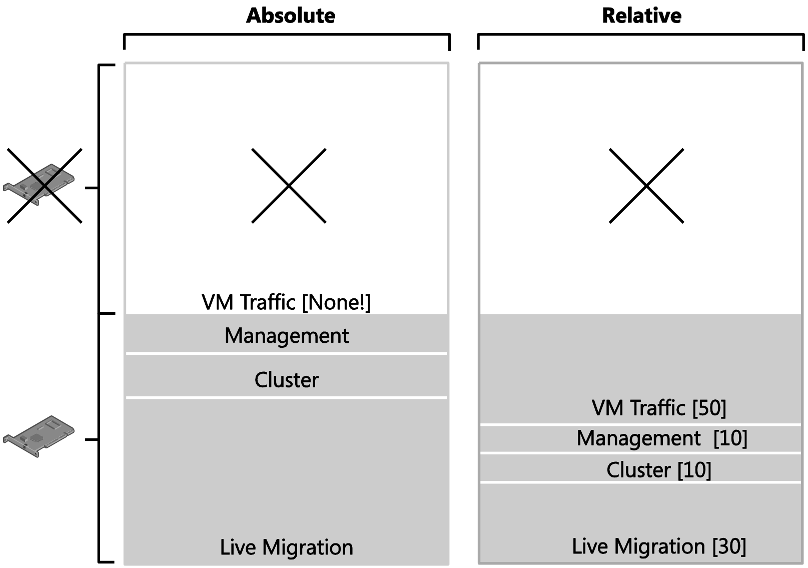


FIGURE 4-8 Bandwidth weights and minimum reservations with a 10-GB path failure.

On first inspection of Figure 4-8, 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.
  - 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, full utilization reservation of 1 GB
- Cluster, minimum of 0.5 GB, maximum of 1 GB
- Live Migration, minimum of 0 GB, maximum of 6.75 GB, 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 (i.e., host networks not connected to a virtual switch)
- Windows OS Network Packet Scheduler
- Data Center Bridging (DBC) for hardware-based QoS when all components 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

Virtual network interface cards (vNICs)

Many networking designs based on Hyper-V 2008 included a NIC team for each of the primary networks (Management, Cluster, and Live Migration) along with a team dedicated to the VM traffic processed via the virtual switch. Windows Server 2012 offers the ability to leverage the new 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 (e.g., 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 4-9 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.

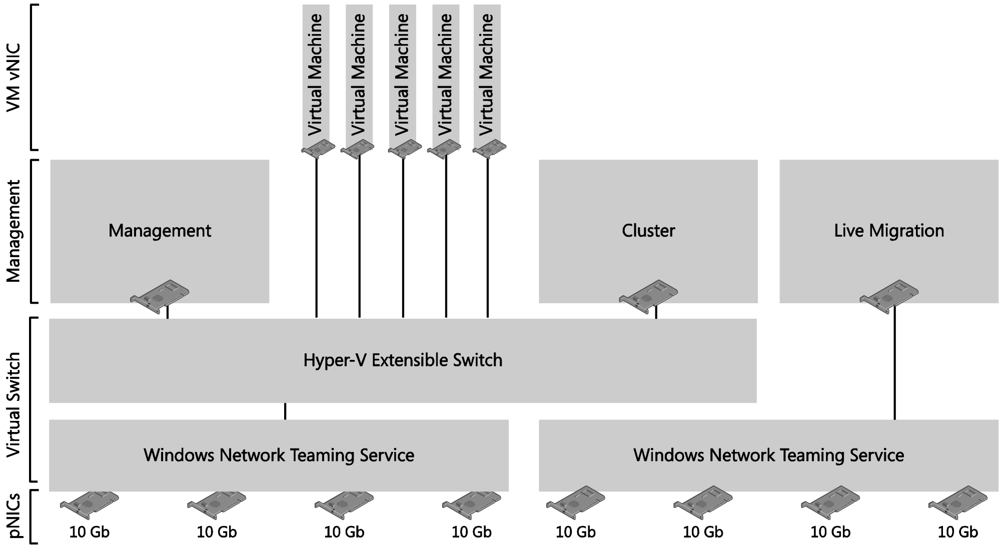


FIGURE 4-9 Utilizing vNICs to consolidate multiple networks.

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

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 very relevant to design considerations as 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.

There are certain circumstances, such as with physical network adapters which 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 virtual machine 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.

In addition, when deploying teams for your hosts, you must also consider the networking switches your hosts will be uplinked with. These ports will require some configuration to mirror the configuration settings you will implement on the teamed interfaces of your hosts. These decisions will include the teaming mode and load balancing decisions, as addressed in Chapter 5.

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. Note however that these are offered to your VMs if you use native Windows Server 2012 Hyper-V tools. If you choose to add these out-of-band, please be aware that VMM will be unable to manage them.

How many logical switches do you need?

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 recommendations.

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 4-10, 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.

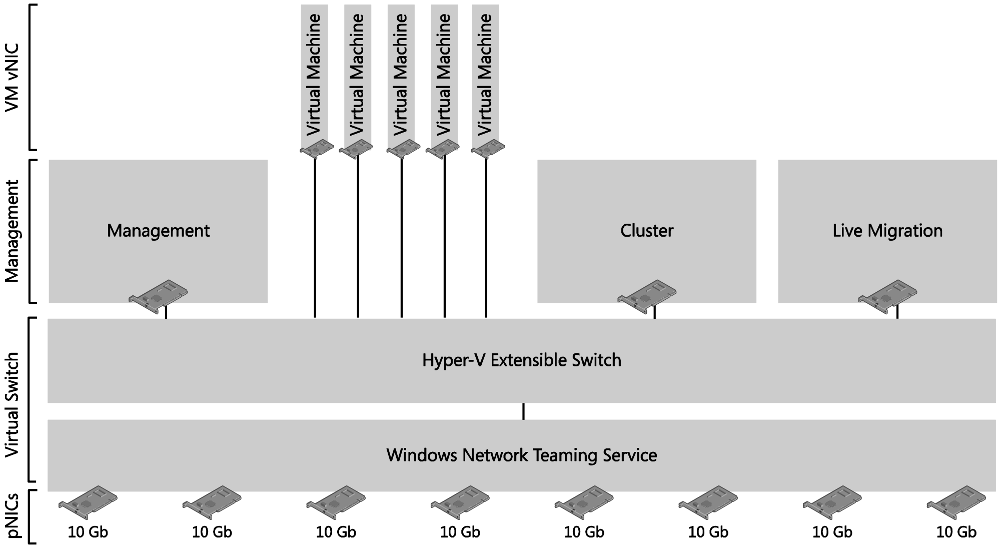


FIGURE 4-10 Fully converged network example.

Dedicated VM switch

A common alternative design to the fully converged example is to implement the configuration utilizing two logical switches (see Figure 4-11), 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.

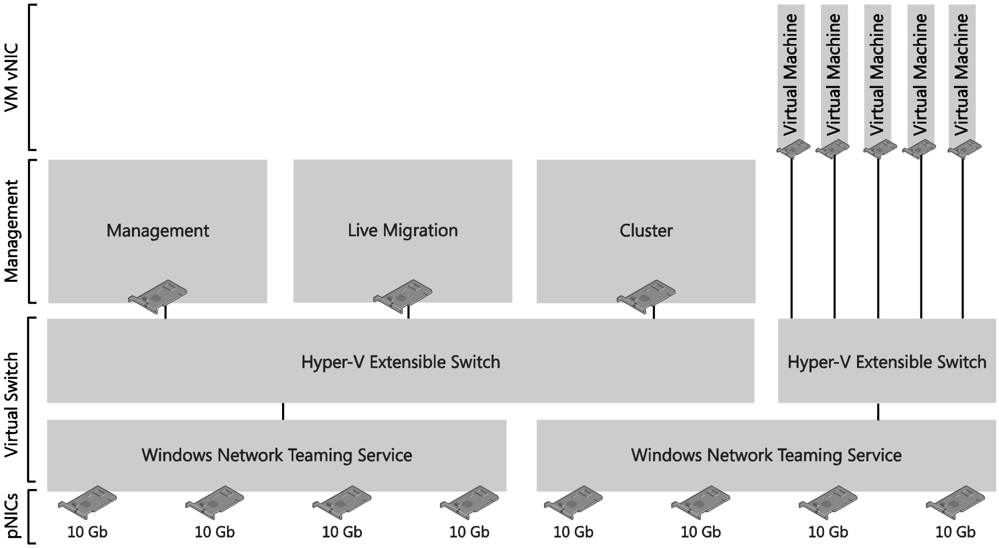


FIGURE 4-11 Implementing with dual logical switches.

> ***NOTE*** **You may need to take additional steps to optimize the above for very high performance environments to effectively bypass 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

As mentioned in the teaming section earlier, iSCSI networks 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 4-12, 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.

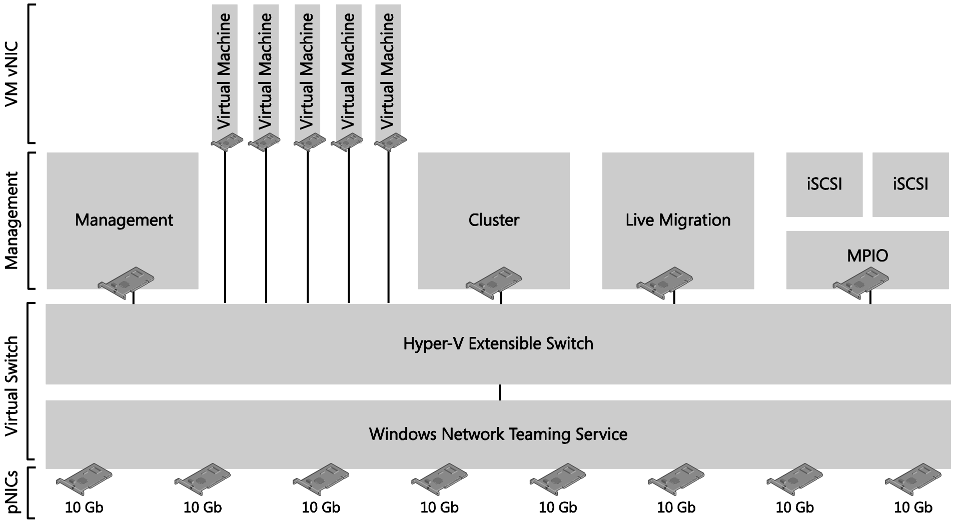


FIGURE 4-12 Converged iSCSI.

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 4-13, 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.

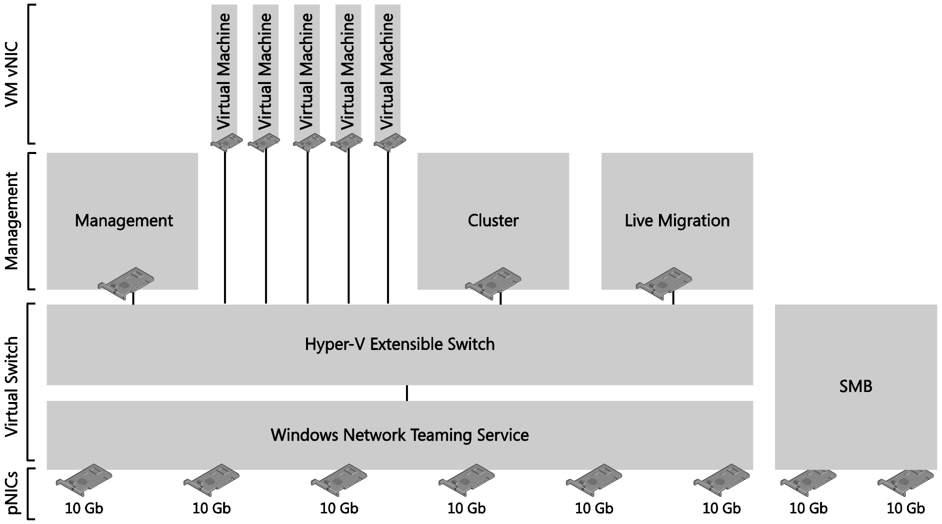


FIGURE 4-13 SMB 3.0 file-based storage.

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/*.

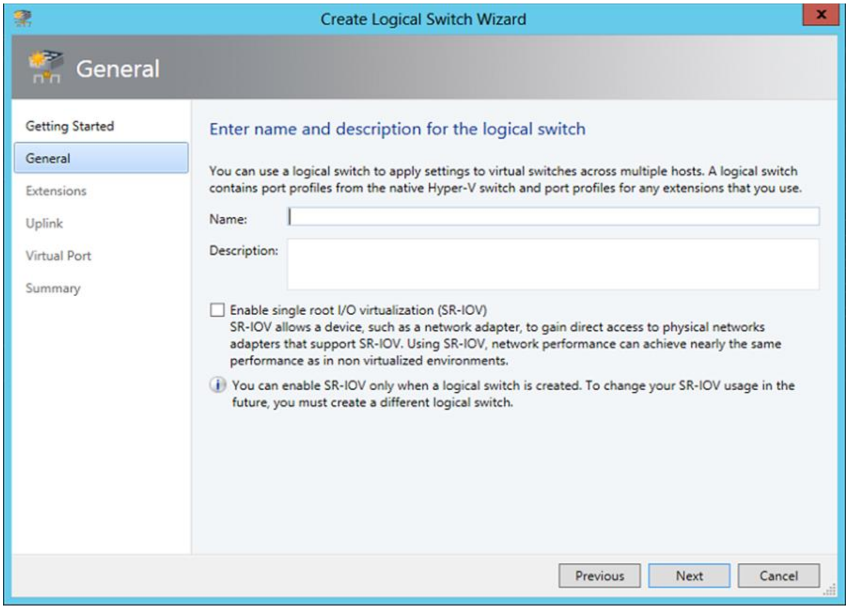
When a Hyper-V virtual NIC is enabled for SR-IOV it is no longer connected to the virtual switch. Instead, the virtual NIC connects to a PCIe feature on the SR-IOV NIC referred to as the Virtual Functions (VF), which then channels the traffic to the physical network. These VFs are isolated and secure, 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 also 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 4-14. 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.*



Enter name and description for the logical switch

You can use a logical switch to apply settings to virtual switches across multiple hosts. A logical switch contains port profiles from the native Hyper-V switch and port profiles for any extensions that you use.

Name:

Description:

□ Enable single root I/O virtualization (SR-IOV)
SR-IOV allows a device, such as a network adapter, to gain direct access to physical networks adapters that support SR-IOV. Using SR-IOV, network performance can achieve nearly the same performance as in non virtualized environments.

You can enable SR-IOV only when a logical switch is created. To change your SR-IOV usage in the future, you must create a different logical switch.

FIGURE 4-14 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, you will not team these interfaces since by design these cards actually bypass the operating system teaming functions. Instead, you will 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 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
```

Dedicated host

If you are designing your environment to use a dedicated host for production, staging, and development, then you should define switches that are dedicated to these specific types of overall workloads. This approach will help deliver a greater level of segregation for fabric administration while attaining standardization on the hosts dedicated to each environment.

Enhancing logical switch capabilities

Extensions can be used to add enhance the features and functionality available in logical switch capabilities provided out of the box. Extensions can allow you to monitor network traffic passing through the switch, provide increased granularity and control in relation to quality of service (QoS), or enhance the level of security. If these enhanced services should be restricted to or deployed on only a limited number of hosts, you may need to consider creating an additional logical switch.

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 4-15. 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 execute much of their daily network management duties, including configuring ports, profiles, switches, and so on.

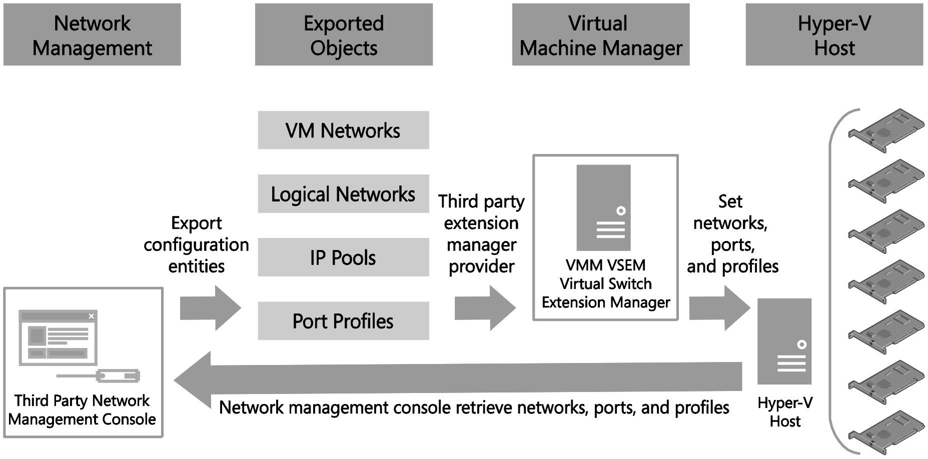

Network Management
Exported Objects
Virtual Machine Manager
Hyper-V Host
VM Networks
Logical Networks
IP Pools
Port Profiles
Export configuration entities
Third party extension manager provider
VMM VSEM Virtual Switch Extension Manager
Set networks, ports, and profiles
Third Party Network Management Console
Network management console retrieve networks, ports, and profiles
Hyper-V Host

FIGURE 4-15 Extension Manager integration.

Once integrated with VMM, all the entities that the network administrator creates, manages, and maintains Cisco Network Management Console are transferred to VMM, including but not limited to VLAN and PVLAN networks, profiles, and pools are represented as similar objects ready for association with VMs. VMM will then continue to manage its services as normal, deploying new VMs to the respective hosts, which will ultimately be connected to a port on the virtual switch. This information will be 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 contain an always up-to-date description field, which contains 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.

Once 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 will automatically discover if the logical switch has defined the use of an extension, will check whether the extension is deployed to the host, and then resolve 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 that 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 which prevented the combined usage of NVGRE and Cisco Nexus 1000v. This problem is resolved with Windows Server 2012 R2.**

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*.

VMM unavailability

Assuming your highly available VMM environment fails, your first concern may be whether all 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 related to figuring out how to keep the VM load optimally distributed on the hosts, or how to ensure your tenants can access their clouds.

There is, however, a real networking reason to be concerned. Recall the discussion about whether you truly need a logical switch to leverage SDN. The text made a strong case for using a logical switch, since VMM not just configure the network in this configuration, but will also transparently manage the hosts to ensure that the network virtualization filters are capable of delivering the traffic to its destination. This function is quite complex, and one of the primary reasons not to try SDN without VMM and logical switches.

Behind the scenes, every time a VM is moved between hosts, and that VM is connected to a virtualized network, 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 will scan for environment changes and update 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.

CHAPTER 5

Deployment

This chapter explains not only how logical switches can be deployed to a new Hyper-V host, but also how an existing standard switch can be migrated to use the new converged approach. The chapter will explain 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 will address the common deployment scenarios and highlight known issues and workarounds regarding logical switches in VMM.

This chapter will:

- Review the requirements for logical switches
- Discuss the different options to deploy 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 VMM it is now possible to consistently configure identical capabilities for network adapters across multiple Hyper-V hosts by using logical switches, which consist of port profiles, uplink profiles, and classifications. Logical switches basically act as containers for the properties or capabilities that you want to deploy and configure for your network adapters.

To be able to deploy a logical switch, you need to prepare the following settings in VMM:

- Logical networks represented in VMM (see Chapter 2, "Logical networks") such as the following:
  - 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 virtual machines (VMs) and tenants)
- Appropriate VM networks since they will be used not only by the VMs but also by the uplink profiles (see Chapter 2)
- At least one uplink profile that contains that defines how the NIC team should be

configured (see Chapter 3, “Port profiles”)

- A logical switch that contains all objects and specifies the network configuration that will be used on all Hyper-V hosts (see Chapter 4, “Logical switches”)
- Hyper-V hosts managed by 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, neither of these options, or 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 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 PowerShell cmdlet:

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

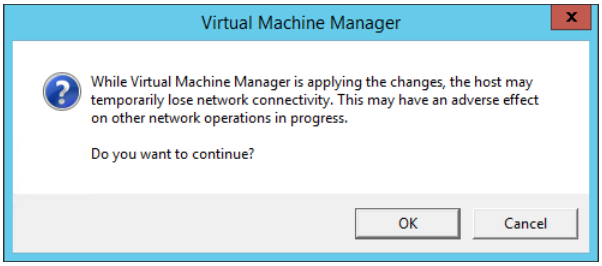
TIP **When you add a host to the VMM management server, by default VMM automatically creates logical networks for those host physical network adapters that do not have logical networks defined on them. You might therefore want to consider clearing this option as described in Chapter 2.**

Deploying logical switches

As discussed in Chapter 2, when a logical switch is applied to a network adapter in a Hyper-V host, 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 and associate the network adapter with the required logical networks, VLAN, and IP subnets.

To deploy a logical switch, you must be a member of the Administrator or Delegated Administrator user role 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 5-1 will appear and you must first acknowledge this warning before you can continue with the deployment process.**



While Virtual Machine Manager is applying the changes, the host may temporarily lose network connectivity. This may have an adverse effect on other network operations in progress.

Do you want to continue?

FIGURE 5-1 Dialog box warning that the host may temporarily lose network connectivity.

There are some additional considerations when you are deploying logical switches onto physical network adapters that will be used for host management, especially when management traffic should only be carried on a *specific* VLAN.

> ***NOTE*** **These network adapters require special attention as 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 physical network adapter 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.

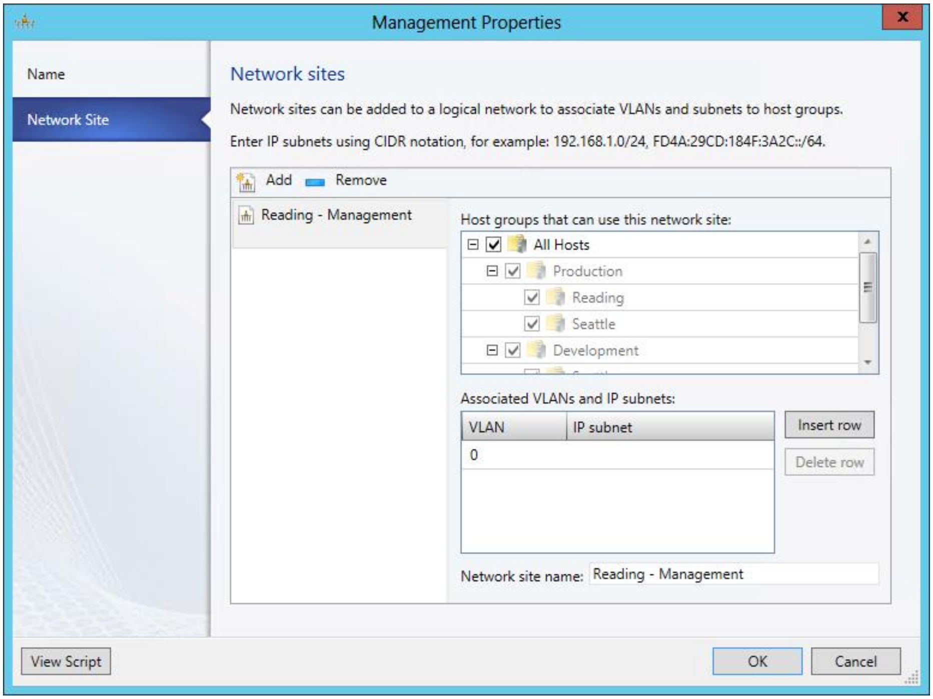
If the network port on the *physical* network switch has been configured for native VLAN on the trunk port, all untagged traffic is treated as destined for the Host Management VLAN. As a result, all packets sent from and to the management logical network will be unchanged by the logical switch.

Untagged host management network adapter

In an untagged scenario, management traffic is not tagged with a VLAN ID as mentioned above. 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.

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 5-2). VMM interprets this setting as meaning “no VLAN” and as a result, the logical switch will be configured to leave outbound network traffic unchanged



Network sites

Network sites can be added to a logical network to associate VLANs and subnets to host groups.

Enter IP subnets using CIDR notation, for example: 192.168.1.0/24, FD4A:29CD:184F:3A2C::/64.

Reading - Management

Host groups that can use this network site:

- ☑ All Hosts
  - ☑ Production
    - ☑ Reading
    - ☑ Seattle
  - ☑ Development

Associated VLANs and IP subnets:

VLAN	IP subnet
0	

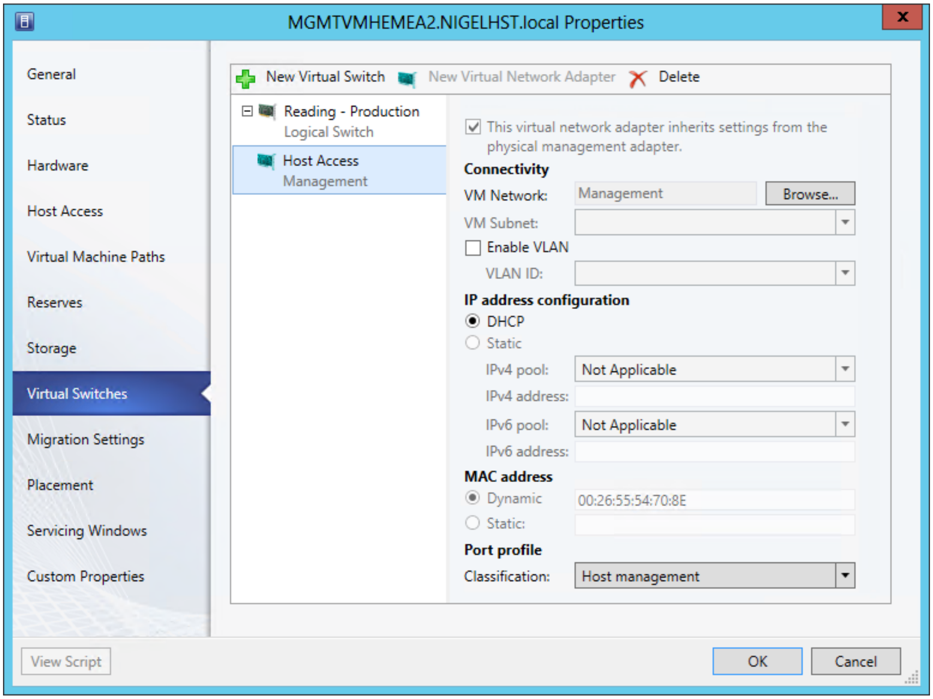
Network site name: Reading - Management

FIGURE 5-2 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 admin console and switch to the Fabric workspace.
2. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, 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 choose New Logical Switch.
6. Select the corresponding logical switch from the drop-down list.
7. Under Physical Adapters, add the two (or more) network adapters that should be used for this logical switch.
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 5-3:
    - If the IP address from the physical network adapter should be reused, select the option "This Virtual Network Adapter Inherits Settings From The Physical Management Adapter." Note that this option is only available 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.
    - As the IP Address will be reused from the physical network adapter, 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 above 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 will initiate the logical switch creation on the Hyper-V host.
15. This job might take a while to finish and 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.



Reading - Production
Logical Switch

Host Access
Management

☑ This virtual network adapter inherits settings from the physical management adapter.

Connectivity

VM Network: Management

VM Subnet:

□ Enable VLAN

VLAN ID:

IP address configuration

● DHCP

○ Static

IPv4 pool: Not Applicable

IPv4 address:

IPv6 pool: Not Applicable

IPv6 address:

MAC address

● Dynamic 00:26:55:54:70:8E

○ Static:

Port profile

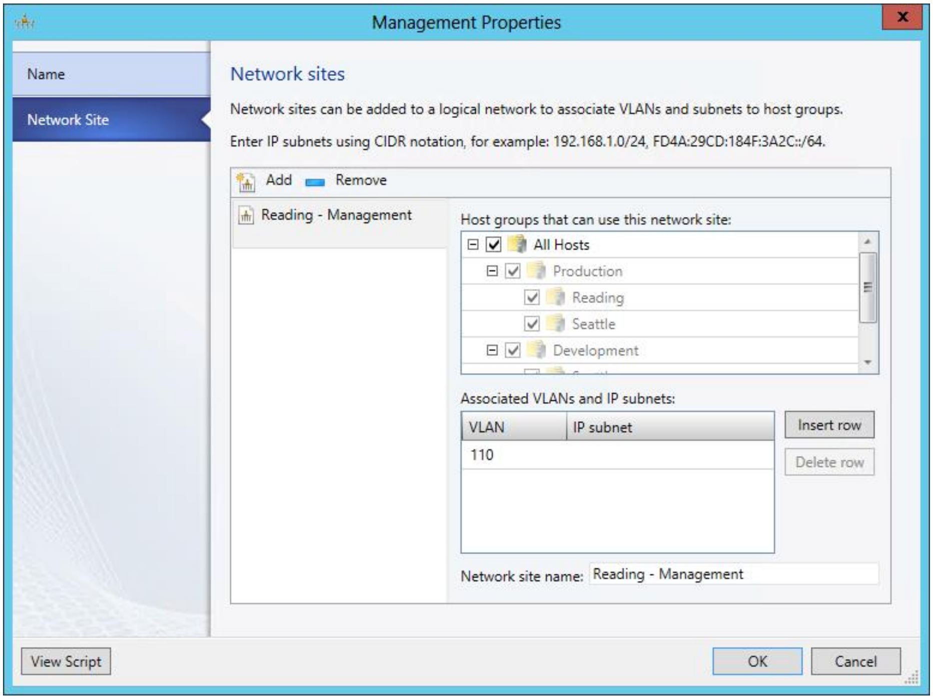
Classification: Host management

FIGURE 5-3 Logical switch deployment using untagged host management.

Once 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, you should define a logical network for management and set the VLAN ID for Network Sites within that network to the appropriate value (110 in Figure 5-4).



Network sites

Network sites can be added to a logical network to associate VLANs and subnets to host groups.

Enter IP subnets using CIDR notation, for example: 192.168.1.0/24, FD4A:29CD:184F:3A2C::/64.

Reading - Management

Host groups that can use this network site:

- ☑ All Hosts
  - ☑ Production
    - ☑ Reading
    - ☑ Seattle
  - ☑ Development

Associated VLANs and IP subnets:

VLAN	IP subnet
110	

Network site name: Reading - Management

FIGURE 5-4 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.

The workflow for deploying a logical switch in a tagged scenario will look like the following:

1. Deploy a logical switch with one physical network adapter as an uplink to keep the other physical network adapter for management connectivity.
2. When the logical switch creation succeeds, add the other physical network adapter to the logical switch.

Before proceeding, therefore, you must make sure that there are at least two physical network adapters that are marked as Used By Management:

1. In the VMM admin console, switch to the Fabric workspace.
2. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, the sub-host group where the host resides.

3. Select the Hyper-V host that should be configured and open its Properties dialog box.
4. In the Properties dialog box, select Hardware.
5. Navigate to Network Adapters and select the physical network adapter that will be used for host management. Ensure the option Used By Management is selected.
6. Repeat these steps for any other physical network adapters required for this configuration.

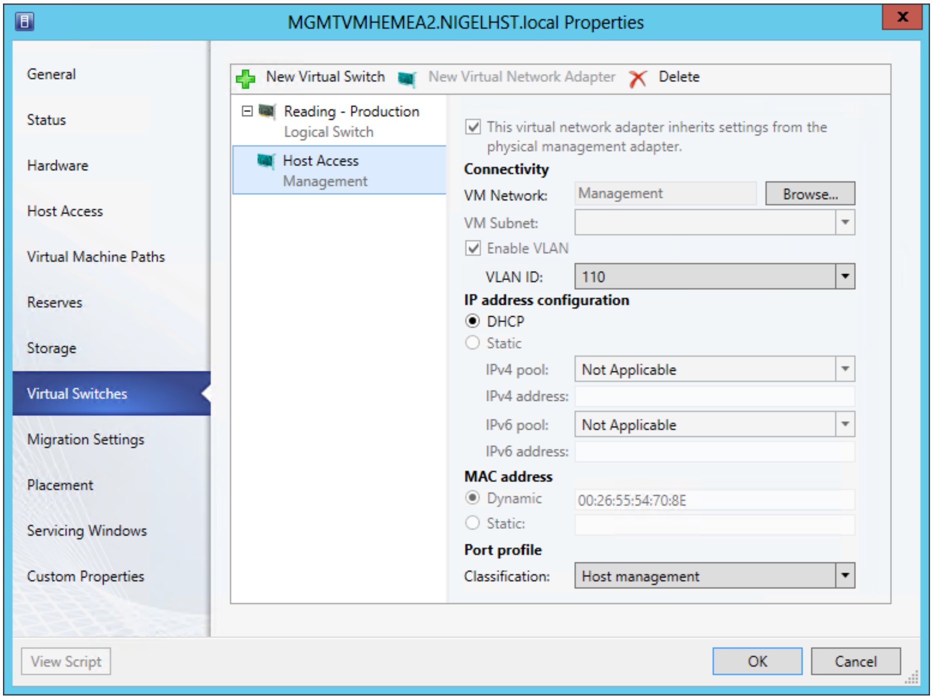
Next, proceed with the deployment of the logical switch to the Hyper-V host of choice. Follow these steps to deploy the first virtual switch using the logical switch:

1. In VMM, change to the Fabric workspace.
2. In the Fabric workspace, expand Servers, select All Hosts, and, if needed, the sub-host group where the host resides.
3. Select the target Hyper-V host and open its Properties dialog box.
4. In the Properties dialog box and select Virtual Switches.
5. Select New Virtual Switch and choose New Logical Switch.
6. Select the appropriate logical switch from the drop-down list.
7. Under Physical Adapters, add the first, and only the first, network adapter that should be used for this logical switch.
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 to be used for the adapter.
11. For the management virtual network adapter, configure the following options as shown in Figure 5-5:
    - If possible and if not affecting the other connectivity, select the This Virtual Network Adapter Inherits Settings From The Physical Management Adapter option.
    - Under Connectivity, select the corresponding VM network, for example Management. There should be the option to select the required VLAN.
    - Under Port Profile, select the classification that matches the network, in this case Host Management.
    - As the IP address will be reused from the physical network adapter, no additional settings are required.
12. For the additional virtual network adapters, such as the one used for Live Migration, configure the following options:
    - After the name to be used for the virtual network adapter has been specified, select

the corresponding VM network, for example Live Migration, and choose the appropriate VLAN if required.

- For the IP address configuration, choose 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 will pick one automatically from the pool.
- Under Port Profile, select the classification that matches the network, in this case Live Migration.

13. Repeat these steps for all virtual network adapters required for this configuration.
14. After the configuration has been completed, click OK to close the Properties page. This will initiate the logical switch creation on the Hyper-V host.
15. This job might take a while to complete, and if you're connected to the Hyper-V host using RDP you will most likely lose the 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 the gateway and DNS Server were as well.



☑ This virtual network adapter inherits settings from the physical management adapter.

Connectivity

VM Network: Management

VM Subnet:

☑ Enable VLAN

VLAN ID: 110

IP address configuration

● DHCP

○ Static

IPv4 pool: Not Applicable

IPv4 address:

IPv6 pool: Not Applicable

IPv6 address:

MAC address

● Dynamic 00:26:55:54:70:8E

○ Static:

Port profile

Classification: Host management

FIGURE 5-5 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 host is ready for providing networking to VM workloads.

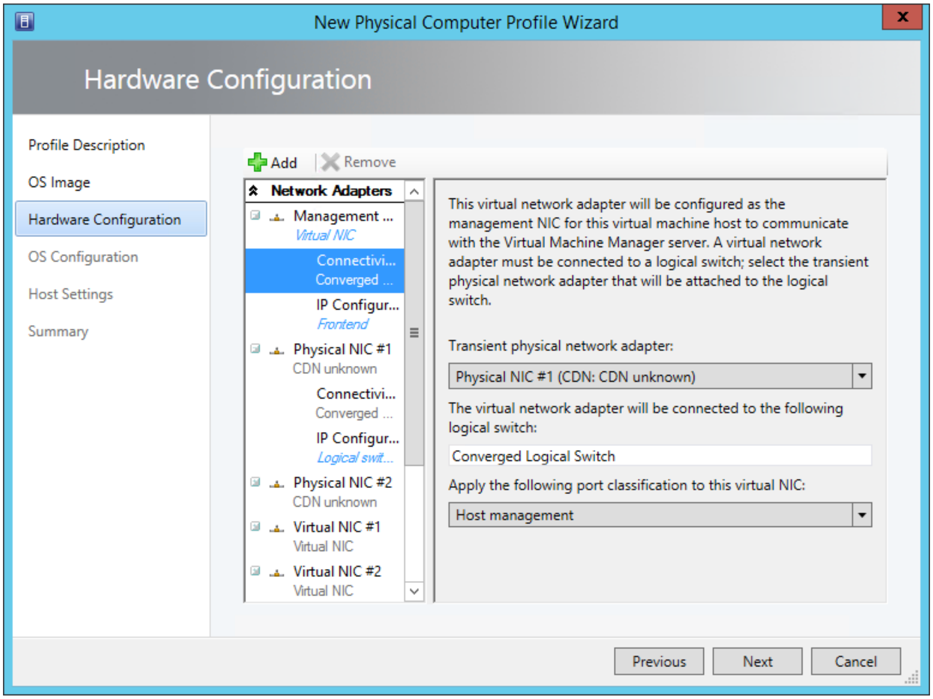
For automation and standardization, you might want to copy and customize the Windows PowerShell script that VMM can display at the end of the wizard before you click OK. An example of a simple script to deploy without virtual network adapters might look like the following:

```
$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 and then 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 will highlight how to configure logical switches as part of bare-metal deployment.

The configuration required for bare-metal deployment is done in the host profile located in the VMM Library. In VMM 2012 SP1 this profile contains not only the Hyper-V configuration but also the entire physical and virtual network adapter configuration (see Figure 5-6).



This virtual network adapter will be configured as the management NIC for this virtual machine host to communicate with the Virtual Machine Manager server. A virtual network adapter must be connected to a logical switch; select the transient physical network adapter that will be attached to the logical switch.

Transient physical network adapter: Physical NIC #1 (CDN: CDN unknown)

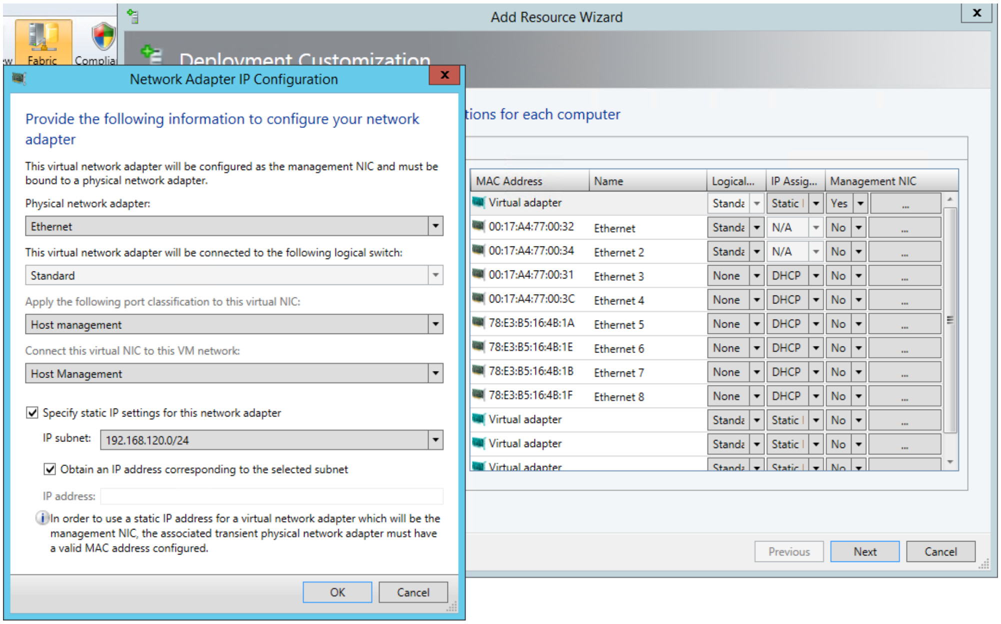
The virtual network adapter will be connected to the following logical switch: Converged Logical Switch

Apply the following port classification to this virtual NIC: Host management

FIGURE 5-6 Hyper-V host profile using logical switch.

If a virtual network adapter is used for management, the setting Create A Virtual Network Adapter As The Management NIC has to be selected. Also, the IP Configuration must specify whether DHCP or fixed IP addresses will be used.

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 physical network adapters (see Figure 5-7). It’s always good to have the MAC addresses available to easily identify the primary adapters.



Network Adapter IP Configuration

Provide the following information to configure your network adapter

This virtual network adapter will be configured as the management NIC and must be bound to a physical network adapter.

Physical network adapter: Ethernet

This virtual network adapter will be connected to the following logical switch: Standard

Apply the following port classification to this virtual NIC: Host management

Connect this virtual NIC to this VM network: Host Management

☑ Specify static IP settings for this network adapter

IP subnet: 192.168.120.0/24

☑ Obtain an IP address corresponding to the selected subnet

IP address:

In order to use a static IP address for a virtual network adapter which will be the management NIC, the associated transient physical network adapter must have a valid MAC address configured.

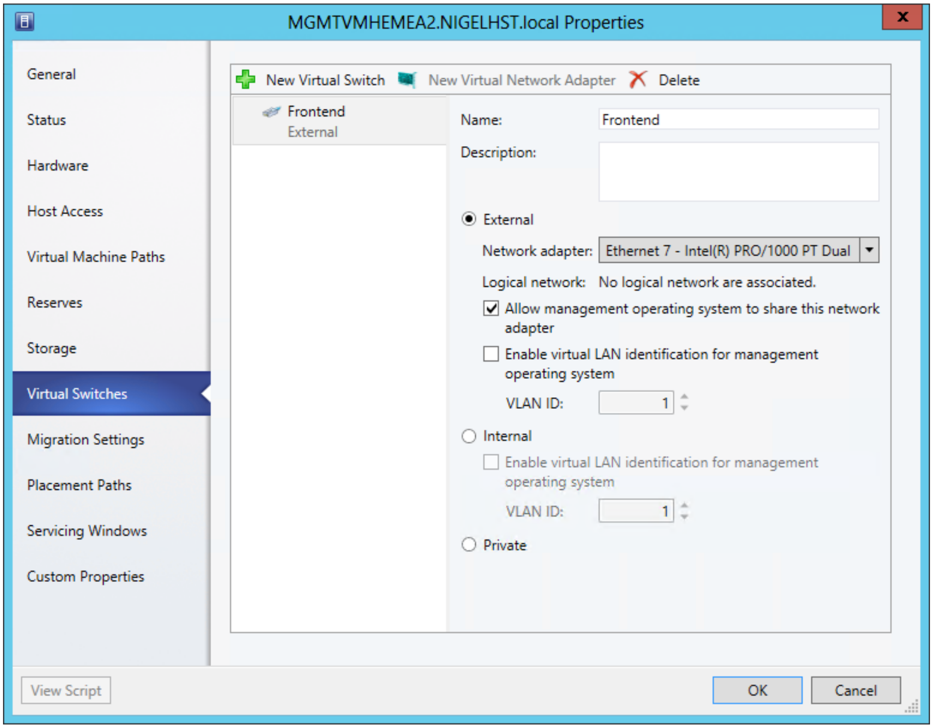
MAC Address	Name	Logical...	IP Assig...	Management NIC	
Virtual adapter		Standa	Static	Yes	...
00:17:A4:77:00:32	Ethernet	Standa	N/A	No	...
00:17:A4:77:00:34	Ethernet 2	Standa	N/A	No	...
00:17:A4:77:00:31	Ethernet 3	None	DHCP	No	...
00:17:A4:77:00:3C	Ethernet 4	None	DHCP	No	...
78:E3:B5:16:4B:1A	Ethernet 5	None	DHCP	No	...
78:E3:B5:16:4B:1E	Ethernet 6	None	DHCP	No	...
78:E3:B5:16:4B:1B	Ethernet 7	None	DHCP	No	...
78:E3:B5:16:4B:1F	Ethernet 8	None	DHCP	No	...
Virtual adapter		Standa	Static	No	...
Virtual adapter		Standa	Static	No	...
Virtual adapter		Standa	Static	No	

FIGURE 5-7 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.

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 you can see in Figure 5-8), it provides the administrator with limited management capability. Unfortunately, once 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 (vNIC) from the network adapter before you begin to deploy the logical switch.



Name: Frontend

Description:

● External

Network adapter: Ethernet 7 - Intel(R) PRO/1000 PT Dual

Logical network: No logical network are associated.

☑ Allow management operating system to share this network adapter

□ Enable virtual LAN identification for management operating system

VLAN ID: 1

○ Internal

□ Enable virtual LAN identification for management operating system

VLAN ID: 1

○ Private

FIGURE 5-8 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 5-9 or selecting the option from the ribbon.

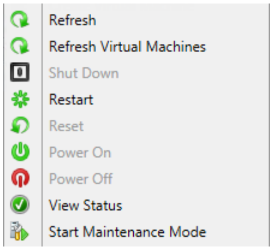


FIGURE 5-9 Enabling maintenance mode on a Hyper-V host in VMM.

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 this 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 like this:

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

This operation will also remove 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 like this:

```
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 will 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 has to 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 5-10. 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.

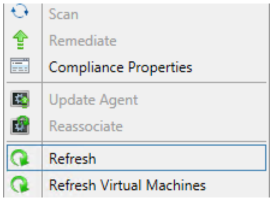


FIGURE 5-10 Refreshing a Hyper-V host in VMM.

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 has already been described earlier in this chapter. You should disable maintenance mode once the logical switch has been deployed successfully to allow the virtual machines 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-SCVM Host -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 physical network adapter 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 will fail.

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 physical network adapter that has the least possible capabilities followed by other physical network adapters that have the same or better capabilities. This will ensure the team works with different brands or adapter types.

Just for the record, the same thing happens in the default NIC Teaming configuration wizard when you try to add a less capable physical network adapter to an existing network team.

CHAPTER 6

Operations

After logical switches have been deployed in the manner described in the Chapter 5, “Deployment,” many of the settings and capabilities defined in those switches cannot be changed without first removing the logical switch and all of the objects that depend on it. The main reason for this is stability, given that changing certain settings and capabilities can significantly influence how the network is presented to virtual machines (VMs) and services that connect through the switch.

With that in mind, no virtual networking solution survives for very long even after being carefully planned based upon what you know at its inception. Factors outside of your immediate control like acquisitions, changing business requirements, and technology developments may force you to review and make one or more changes to your architecture. This chapter walks through some relatively common change scenarios that may occur and provides some detailed recommendations, advice, and guidance around how best to deal with them, noting where logical switch removal and replacement is required to address a particular scenario.

Specifically, this chapter covers what you should do if you need to:

- Add support for single root I/O virtualization (SR-IOV)
- Change the network adapter assigned to a logical switch
- Handle pre-existing network adapter teams
- Convert a standard (virtual) switch to a logical switch
- Manage logical switch compliance
- Make changes to VLAN and PVLAN numbers
- Move from VLAN isolation to Network Virtualization
- Delete a logical network
- Move a VM network to a new logical network
- Delete a VM network

Operational scenarios

The sections in this chapter are broken out into a number of distinct and totally separate scenarios that represent some of the most common changes that an IT administrator will need to perform after a virtual network solution has been deployed. 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 perform that specific change within your own environment.

Logical switches

The following sections review some of the most common changes a logical switch may 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.

MORE INFO **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*.**

As discussed in Chapter 3, "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 Microsoft 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 will no longer be available within the VMM administrator console after the logical switch is deployed, and any attempt to work around this limitation using Windows PowerShell will simply fail 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.

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 may 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 above. 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.

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

You can create a team on a Windows Server computer 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.*

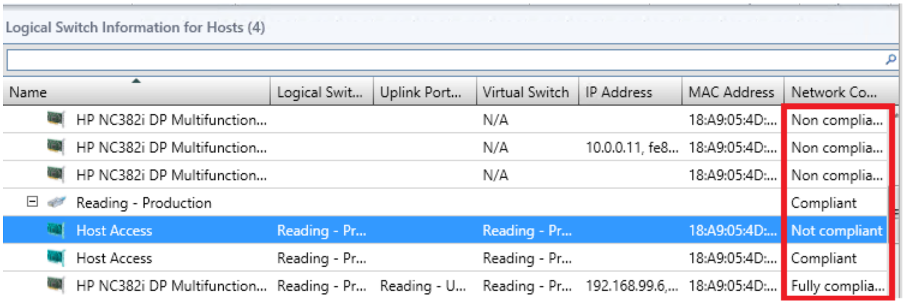
You can either leave the network team as is, with the understanding that these interfaces can only be used with standard Hyper-V switches, each team needs to be configured and managed separately and finally, that the team and corresponding network adapters fall out of the scope of management of VMM or 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, " Logical switches and network design," 4 and Chapter 5 for more details).

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 6-1.



Logical Switch Information for Hosts (4)

Name	Logical Swit...	Uplink Port...	Virtual Switch	IP Address	MAC Address	Network Co...
HP NC382i DP Multifunction...			N/A		18:A9:05:4D:...	Non complia...
HP NC382i DP Multifunction...			N/A	10.0.0.11, fe8...	18:A9:05:4D:...	Non complia...
HP NC382i DP Multifunction...			N/A		18:A9:05:4D:...	Non complia...
⊟ Reading - Production						Compliant
Host Access	Reading - Pr...		Reading - Pr...		18:A9:05:4D:...	Not compliant
Host Access	Reading - Pr...		Reading - Pr...		18:A9:05:4D:...	Compliant
HP NC382i DP Multifunction...	Reading - Pr...	Reading - U...	Reading - Pr...	192.168.99.6,...	18:A9:05:4D:...	Fully complia...

FIGURE 6-1 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 indicates that the settings on the host are consistent with the expected configuration in VMM.
- Partially Compliant indicates that there is only a partial match between the settings on the host and expected configuration.
- Not Compliant 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 admin console can be used to address and resolve any of the issues that may have been discovered. Note that you may 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 virtual machines), and then take the steps to necessary to remediate the issue.

Logical networks

The following sections review some of the common changes that may 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, there is unfortunately no way to change it. To use Network Virtualization, therefore, you will need to 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, but the outage should be relatively minor. Once the migration has been successfully completed, you can remove the VLAN or PVLAN isolated logical network as described in the "Deleting a logical network" section 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 Chapters 2 through 4. 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 needs to be configured such that sites within the logical network are not connected. In addition, each individual VLAN ID needs to 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 needs to 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 will no longer be available within the VMM admin 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 will be 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 above, you should 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 you with 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 will work only when both the VLAN ID and the IP subnet is changed because VMM will not allow you to create a VLAN that has the same subnet as another. In such cases, the remediation steps will be more extensive, requiring you use a temporary (interim) subnet during the transition period.**

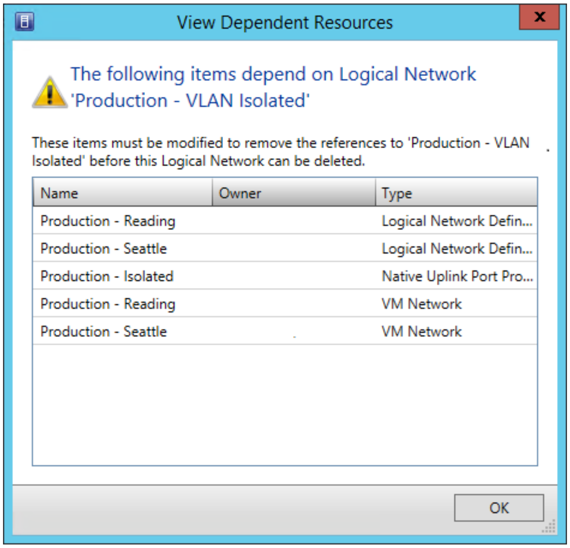
You can follow a similar process to the above, in essence creating a new site and mapping to a new VM network, whenever you need to change either of the values defined for the Primary VLAN ID or the Secondary VLAN ID in a network site that is part of a logical network configured to support PVLANs.

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 will 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 6-2. Note that this list must be empty before you can successfully delete the logical network.



The following items depend on Logical Network 'Production - VLAN Isolated'

These items must be modified to remove the references to 'Production - VLAN Isolated' before this Logical Network can be deleted.

Name	Owner	Type
Production - Reading		Logical Network Defin...
Production - Seattle		Logical Network Defin...
Production - Isolated		Native Uplink Port Pro...
Production - Reading		VM Network
Production - Seattle		VM Network

FIGURE 6-2 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 in respect of 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

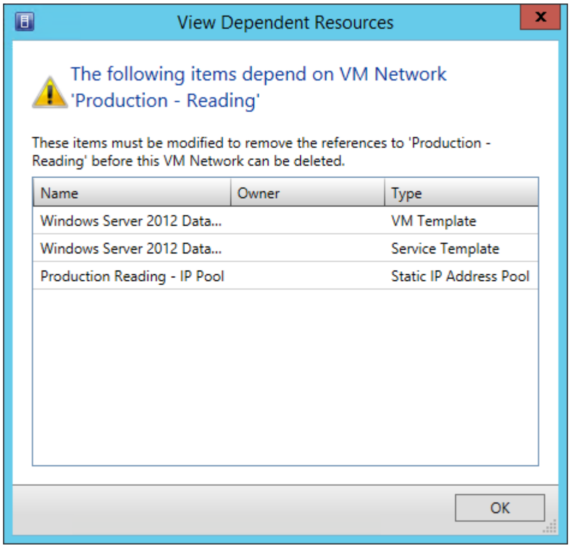
This final section reviews the two most common scenarios relating to VM networks: the need to map an existing VM network to a different logical network and how to effectively delete a VM network.

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 6-3). You then delete any IP pools and finally the VM network itself.



View Dependent Resources

The following items depend on VM Network 'Production - Reading'

These items must be modified to remove the references to 'Production - Reading' before this VM Network can be deleted.

Name	Owner	Type
Windows Server 2012 Data...		VM Template
Windows Server 2012 Data...		Service Template
Production Reading - IP Pool		Static IP Address Pool

FIGURE 6-3 Checking for dependent resources prior to VM network deletion.

As with logical networks, to remove the IP pool you may 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 may be exceptions. For these specific cases, you will need to use Windows PowerShell as in the example below 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 will remove 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.

About the authors

NIGEL CAIN leads the Windows and System Center Customer, Architecture, and Technology (CAT) team in the Asia-Pacific region. He and his team work closely with service providers (hosters) and enterprise customers, helping them take full advantage of Windows Server and Microsoft 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 and hybrid clouds at a number of industry events. Nigel graduated with an MBA from Warwick Business School in 2010. For more information, see *http://uk.linkedin.com/in/nigelcain/*.

DAMIAN FLYNN, MVP, Microsoft System Center and Datacenter, is the Infrastructure Technical Architect for Lionbridge Technologies (a localization, logo-certification, search, and content-services company and Microsoft Certified Gold Partner). Damian works closely with the business stakeholders, IT team, and partners, while also incubating new projects. His current focus is on software defined networks (SDN) with the Windows Azure Pack, with perspective on orchestration of repeatable processes in Development-Operations (DevOps) scenarios. Damian has presented sessions on private and hybrid clouds at numerous industry events and is a co-author of books focusing on Microsoft cloud solutions. He is active in many Microsoft programs, blogs at *www.damianflynn.com* and *www.petri.co.il*, tweets at @damian_flynn, and has published a number of white papers and technical articles.

ALVIN MORALES is a senior IT operations engineer at Microsoft 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 is currently working on his MBA in cybersecurity from the University of Dallas. For more information see *http://www.linkedin.com/in/alvinmorales*.

MICHEL LUESCHER is a senior consultant on the datacenter team in the Enterprise Services Division at Microsoft Switzerland, primarily focused on datacenter architectures. He joined Microsoft at the beginning of January 2009 and since then has been activity engaged with a number of different divisions and communities across the company. Michel is a well-known virtualization specialist, presenting at different internal and external events, and co-authored the *Windows Server 2012 Hyper-V Installation and Configuration Guide* (John Wiley & Sons, 2013). Michel writes regularly about Microsoft virtualization and private cloud computing in his blog *http://www.server-talk.eu*. You can also find him on Twitter as @michelluescher.

About the series editor

MITCH TULLOCH is a well-known expert on Windows Server administration and virtualization. He has published hundreds of articles on a wide variety of technology sites and has written or contributed to over two dozen books, including *Windows 7 Resource Kit* (Microsoft Press, 2009), for which he was lead author*; Understanding Microsoft Virtualization Solutions: From the Desktop to the Datacenter* (Microsoft Press, 2010); and *Introducing Windows Server 2012* (Microsoft Press, 2012), a free ebook that has been downloaded almost three quarters of a million times.

Mitch has been repeatedly awarded Most Valuable Professional (MVP) status by Microsoft for his outstanding contributions to supporting the global IT community. He is a nine-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 (*http://www.wservernews.com*), a weekly newsletter focused on system administration and security issues for the Windows Server platform. With more than 100,000 IT pro subscribers worldwide, WServerNews is the largest Windows Server–focused newsletter in the world.

Mitch 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 at *http://twitter.com/mitchtulloch* or like him on Facebook at *http://www.facebook.com/mitchtulloch*.

Now that you've read the book...

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