

The Lenovo logo is displayed in white text on a black rectangular background.

# Microsoft Storage Spaces Direct (S2D) Deployment Guide

Last Update: August 2023

---

**Includes detailed steps for deploying a Microsoft Azure Stack HCI solution based on Windows Server 2022**

---

**Updated for Lenovo ThinkAgile MX solutions running the Azure Stack HCI operating system**

---

**Includes deployment scenarios for RoCE and iWARP, as well as switched and direct-connected solutions**

---

**Provides validation steps along the way to ensure successful deployment**

**Dave Feisthammel**

**Mike Miller**

**David Ye**



# Abstract

Lenovo® and Microsoft have worked closely for many years to craft a software-defined storage solution leveraging the advanced feature sets of the Windows Server and Azure Stack HCI operating systems, combined with the flexibility of Lenovo ThinkSystem™ rack and edge servers. In addition, we have created Lenovo ThinkAgile™ MX solutions that contain only servers and server components that have been certified under the Microsoft Azure Stack HCI Program to run Microsoft Storage Spaces Direct (S2D) properly.

This solution provides a solid foundation for customers looking to consolidate both storage and compute capabilities on a single hardware platform, or for those enterprises that wish to have distinct storage and compute environments. In both situations, this solution provides outstanding performance, high availability protection and effortless scale out growth potential to accommodate evolving business needs.

This deployment guide makes extensive use of Windows PowerShell commands and scripts. It guides the reader through a set of well-proven procedures leading to readiness of the solution for production use. It covers multiple deployment scenarios, including RoCE and iWARP implementations of RDMA, as well as 2- and 3-node direct-connected deployments.

If you prefer to deploy an Azure Stack HCI cluster from the Windows Admin Center (WAC) deployment wizard, please refer to our companion document at the following URL:

<https://lenovopress.com/1p1524>

**Do you have the latest version?** Check whether you have the latest version of this document by clicking the **Check for Updates** button on the front page of the PDF. Pressing this button will take you to a web page that will tell you if you are reading the latest version of the document and give you a link to the latest if needed. While you're there, you can also sign up to get notified via email whenever we make an update.

# Contents

Storage Spaces Direct solution overview . . . . .	3
Solution configuration . . . . .	8
General hardware preparation . . . . .	12
Deployment scenarios . . . . .	22
Create failover cluster . . . . .	85
Enable and configure Storage Spaces Direct . . . . .	88
Cluster set creation . . . . .	94
Summary . . . . .	101
Lenovo Professional Services . . . . .	101
Change history . . . . .	101
Authors . . . . .	103
Notices . . . . .	105
Trademarks . . . . .	106

# Storage Spaces Direct solution overview

Microsoft Storage Spaces Direct (S2D) has become extremely popular with customers all over the world since its introduction with the release of Microsoft Windows Server 2016. This software-defined storage (SDS) technology leverages the concept of collecting a pool of affordable drives to form a large usable and shareable storage repository.

Lenovo continues to work closely with Microsoft to deliver the latest capabilities in the Windows Server 2022 and Azure Stack HCI operating systems. This document focuses on S2D/AzureStack HCI deployment on Lenovo’s rack servers. Special emphasis is given to Lenovo ThinkAgile MX Certified Nodes, which are certified under the Microsoft Azure Stack HCI Program for Storage Spaces Direct.

The example solutions shown in this paper were built using the Lenovo ThinkAgile MX Certified Node that is based on the ThinkSystem SR650 rack server. The SR650 server is used throughout this document as an example for Azure Stack HCI deployment tasks. As other rack servers are added to the ThinkAgile MX solution family (such as the SR630 V2 and SR650 V2), the steps required to deploy Azure Stack HCI on them are identical to those contained in this document.

Figure 1 shows an overview of the Storage Spaces Direct stack.

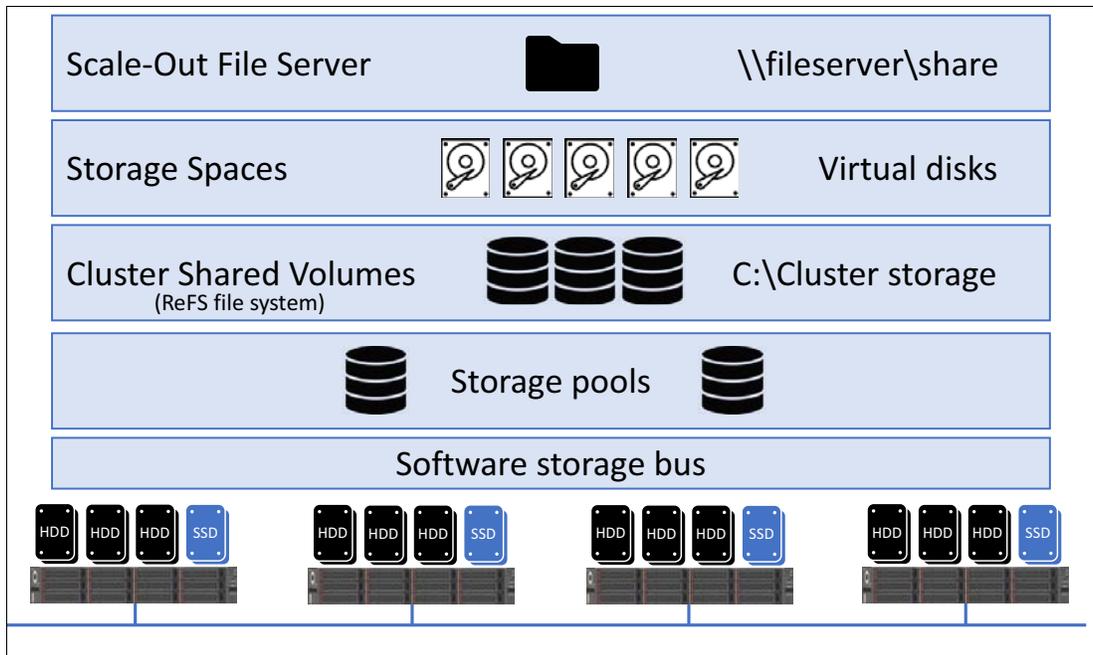


Figure 1 Storage Spaces Direct stack

Before getting into the technical details, the terminology used in this document must be made clear in order to avoid confusion between operating systems (such as Windows Server 2022 and Azure Stack HCI) and technologies (such as Storage Spaces Direct and Azure Stack HCI). In particular, there is often confusion regarding whether “Azure Stack HCI” refers to the technology feature set or the operating system. When referring to the feature set, we use the terms “S2D” or “Azure Stack HCI” and use “HCI OS” to clearly indicate that we are discussing the actual Azure Stack HCI operating system.

Related to this topic is the relationship between the Windows Server operating systems and the Azure Stack HCI operating systems. Although many S2D features are supported by the

Windows Server operating systems, HCI OS contains additional capabilities not found in Windows Server. Also, HCI OS “20H2” is based on Windows Server 2019, while HCI OS “21H2” and later are based on Windows Server 2022. This distinction is particularly important when looking for appropriate device drivers, since Lenovo does not designate distinct drivers for the HCI OSes. For HCI OS 20H2 (which is no longer supported by Microsoft), use drivers designated for Windows Server 2019 and for HCI OS 21H2 and later, use drivers designated for Windows Server 2022.

Key considerations for S2D are as follows:

- ▶ Windows Server vs. HCI OS

As mentioned in the previous paragraph, Windows Server and HCI OS are different operating systems with different requirements and feature sets. Therefore, one of the first decisions to be made is whether to use Windows Server or HCI OS. For example, if using an operating system with full GUI support (i.e. “Desktop Experience”) is a requirement, a Windows Server operating system must be used, since HCI OS supports only the Server Core option. On the other hand, if tight integration with Azure services like Azure Kubernetes Service (AKS) is a requirement, HCI OS should be used.

For additional guidance from Microsoft, refer to the following URLs:

<https://docs.microsoft.com/en-us/azure-stack/hci/concepts/compare-windows-server>  
<https://techcommunity.microsoft.com/t5/itops-talk-blog/azure-stack-hci-and-windows-server-2022/ba-p/3038229>

- ▶ Roles and features required for S2D

Multiple server roles and features that are not installed/enabled by default are required for Azure Stack HCI functionality. For all deployments, the Failover Clustering feature and a few File and Storage Services must be installed, including the “File and iSCSI Services” and “File Server” role services.

However, other roles and features might or might not be required, depending on a couple of factors and preferences. The Hyper-V role is required for true hyperconverged clusters, but is often installed even for converged (disaggregated) clusters. The Data Center Bridging feature is required only if using the RoCEv2 implementation of RDMA.

- ▶ S2D capacity and storage growth

Leveraging the hot-swap drive bays of Lenovo ThinkSystem rack servers such as the SR650, and high-capacity hard disk drives (HDD) with capacities up to 14TB that can be used in this solution, each server node is itself a JBOD (just a bunch of disks) repository. As demand for storage and/or compute resources grow, additional ThinkAgile MX Certified Nodes can be added into the environment to provide the necessary storage expansion.

- ▶ S2D performance

Using a combination of solid-state drives (SSD or NVMe) and regular HDDs as the building blocks of the storage volume, an effective method for storage tiering is available in Lenovo ThinkAgile MX Hybrid solutions. Faster-performing SSD or NVMe devices act as a cache repository to the capacity tier, which is usually placed on traditional HDDs in these solutions. Data is striped across multiple drives, thus allowing for very fast retrieval from multiple read points.

For even higher performance, ThinkAgile MX All-Flash solutions are available as well. These solutions do not use spinning disks. Rather, they are built using all SSD, all NVMe or a combination of NVMe devices acting as cache for the SSD capacity tier.

At the physical network layer, 10GbE, 25GbE, or 100GbE links are employed today. For most situations, the dual 10/25GbE network paths that contain both Windows Server operating system and storage replication traffic are more than sufficient to support the

workloads and show no indication of bandwidth saturation. However, for very high performance all-flash Azure Stack HCI clusters, a dual-port 100GbE network adapter that has been certified is also available.

► S2D resiliency (see Table 1 for a summary of supported resiliency types)

Traditional disk subsystem protection relies on RAID storage controllers. In S2D, high availability of the data is achieved using a non-RAID adapter and adopting redundancy measures provided by the operating system. S2D provides various resiliency types, depending on how many nodes make up the Azure Stack HCI cluster. Storage volumes can be configured as follows:

- Two-way mirror: Requires two cluster nodes. Keeps two copies of all data, one copy on the drives of each node. This results in storage efficiency of 50%, which means that 2TB of data will consume 4TB of storage pool capacity. Two-way mirroring can tolerate a single hardware failure (node or drive) at a time.
- Nested resiliency: Introduced in Windows Server 2019, requires exactly two cluster nodes and offers two options.
  - Nested two-way mirror: Two-way mirroring is used within each node, then further resiliency is provided by two-way mirroring between the two nodes. This essentially a four-way mirror, with two copies of all data on each node. Performance is optimal, but storage efficiency is low, at 25 percent.
  - Nested mirror-accelerated parity: Essentially, this method combines nested two-way mirroring with nested parity. Local resiliency for most data within a node is handled by single parity except for new writes, which use two-way mirroring for performance. Further resiliency is provided by a two-way mirror between the two nodes. Storage efficiency is approximately 35-40 percent, depending on the number of capacity drives in each node as well as the mix of mirror and parity that is specified for the volume.
  - For more information about nested resiliency, see the following Microsoft article:  
<https://learn.microsoft.com/en-us/azure-stack/hci/concepts/nested-resiliency>
- Three-way mirror: Requires three or more cluster nodes. Keeps three copies of all data, one copy on the drives of each of three nodes. This results in storage efficiency of 33 percent. Three-way mirroring can tolerate at least two hardware failures (node or drive) at a time.
- Dual parity: Also called “erasure coding,” requires four or more cluster nodes. Provides the same fault tolerance as three-way mirroring, but with better storage efficiency. Storage efficiency improves from 50% with four nodes to 80% with sixteen nodes in the cluster. However, since parity encoding is more compute intensive, the cost of this additional storage efficiency is performance. Dual parity can tolerate up to two hardware failures (node or drive) at a time.
- Mirror-accelerated parity: This is a combination of mirror and parity technologies. Writes land first in the mirrored portion and are gradually moved into the parity portion of the volume later. To mix three-way mirror and dual parity, at least 4 nodes are required. Storage efficiency of this option is between all mirror and all parity.
- For more information about fault tolerance and storage efficiency on HCI clusters, refer to the following Microsoft article:  
<https://learn.microsoft.com/en-us/azure-stack/hci/concepts/fault-tolerance>

Table 1 Resiliency types supported by S2D

Resiliency	Minimum required fault domains	Failure tolerance	Storage efficiency
Two-way mirror	2	1	50%
Three-way mirror	3	2	33.3%
Dual parity	4	3	50% - 80%
Mixed	4	2	33.3% - 80%

► S2D use cases

The importance of having a SAN in the enterprise space as the high-performance and high-resilience storage platform is changing – S2D can be a direct replacement for this role. Whether the primary function of the environment is to provide Windows applications or a Hyper-V virtual machine farm, Azure Stack HCI can be configured as the principal storage provider to these environments. Another use for S2D is as a repository for backup or archival of VHD(X) files. Wherever a shared volume is applicable for use, S2D can be the solution to support this function.

S2D supports two general deployment types, *converged* (sometimes called “disaggregated”) and *hyperconverged*. Both approaches provide storage for Hyper-V, specifically focusing on Hyper-V Infrastructure as a Service (IaaS) for service providers and enterprises.

In the converged/disaggregated approach, the environment is separated into compute and storage components. An independent pool of servers running Hyper-V acts to provide the CPU and memory resources (the “compute” component) for the running of VMs that reside on the storage environment. The “storage” component is built using S2D and Scale-Out File Server (SOFS) to provide an independently scalable storage repository for the running of VMs and applications. This method, as illustrated in Figure 2, allows for the independent scaling and expanding of the compute cluster (Hyper-V) and the storage cluster (S2D).

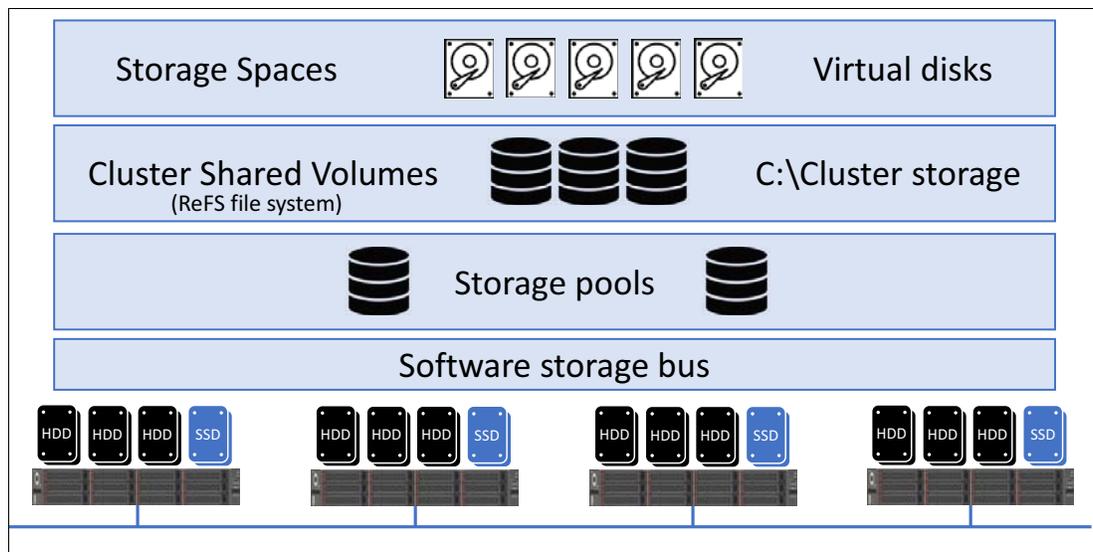


Figure 2 Converged/disaggregated S2D deployment type - nodes do not host VMs

For the hyperconverged approach, there is no separation between the resource pools for compute and storage. Instead, each server node provides hardware resources to support the

running of VMs under Hyper-V, as well as the allocation of its internal storage to contribute to the S2D storage repository.

Figure 3 on page 7 demonstrates this all-in-one configuration for a four-node hyperconverged solution. When it comes to growth, each additional node added to the environment will mean both compute and storage resources are increased together. Perhaps workload metrics dictate that a specific resource increase is sufficient to cure a bottleneck (e.g., CPU resources). Nevertheless, any scaling will mean the addition of both compute and storage resources. This is a fundamental limitation for all hyperconverged solutions.

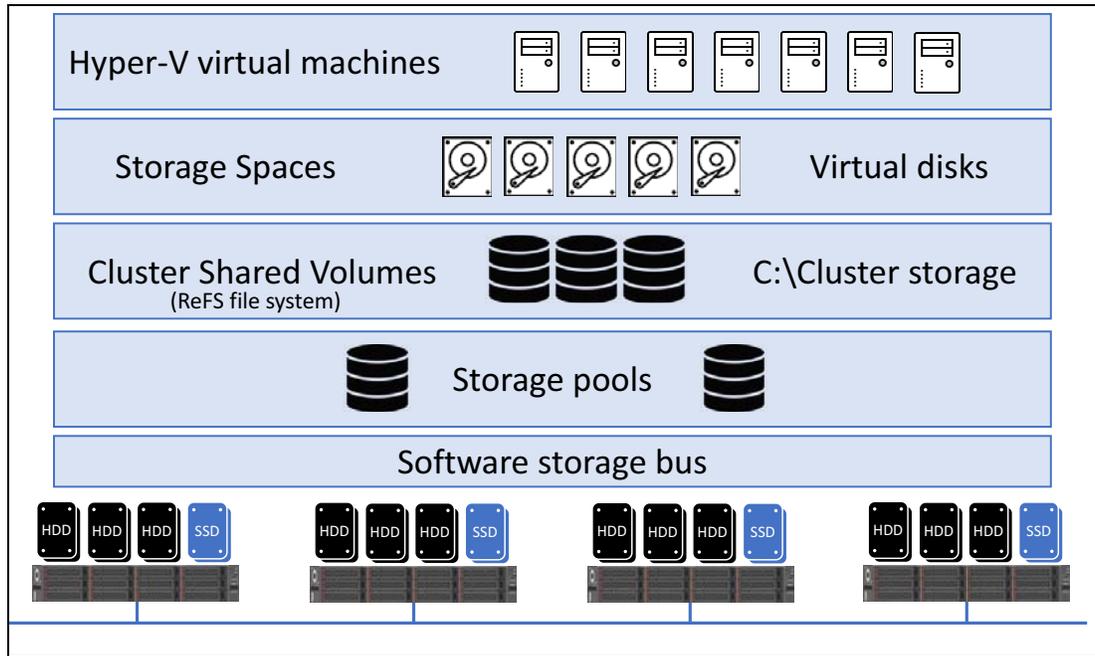


Figure 3 Hyperconverged S2D deployment type - nodes provide shared storage and Hyper-V hosting

Common to both converged and hyperconverged deployment types, S2D relies on Remote Direct Memory Access (RDMA) networking for storage (East-West) traffic inside the cluster. The two main implementations of RDMA that can be used for S2D are RDMA over Converged Ethernet version 2 (RoCEv2) and iWarp. Which implementation is chosen is primarily a personal preference. The key difference, in terms of the S2D deployment process, is that a RoCE implementation requires configuration of the network switches (if used) to enable Data Center Bridging (DCB), while iWARP does not require any special network switch configuration.

## Solution configuration

Configuring the converged and hyperconverged S2D deployment types is essentially identical. In this section, we begin by discussing, in general, the components used in our lab environment for the various deployment scenarios that are covered in this document. Next, in “General hardware preparation” on page 13, general installation and configuration steps required for all deployment scenarios, such as firmware updating, and OS installation and configuration are addressed. In the section “Deployment scenarios” on page 25, each of the deployment scenarios listed below are described in detail. Any special considerations, such as network cable diagrams and switch configuration are covered for each scenario.

Based on customer feedback, we have found a few key deployment scenarios to be the most widely adopted. The deployment scenarios covered in this document are based on the number of nodes contained in the Azure Stack HCI cluster, the number and type of network interfaces provided by each node, and whether or not a network switch is used for storage traffic. In addition, the steps required to deploy Azure Stack HCI depend on whether RDMA is implemented via RoCE using Mellanox NICs or via iWarp using Intel E810 NICs.

The deployment scenarios addressed in this document include the following:

- Two or more nodes using the RoCE implementation of RDMA (includes details for using one or two dual-port NICs in each node of the cluster)
- Two or three nodes using RoCE, direct-connected (no switch for storage traffic)
- Two or more nodes using the iWARP implementation of RDMA (includes details for using one or two dual-port NICs in each node of the cluster)
- Two or three nodes using iWARP, direct-connected (no switch for storage traffic)

The following components and information are relevant to the lab environment used to develop this guide. This solution consists of two key components, a high-throughput network infrastructure and a storage-dense high-performance server farm. Each of these components are described in further detail below. The examples and diagrams shown in this document are based on a Lenovo ThinkAgile MX solution using the ThinkSystem SR650 V1 rack server.

For details regarding Lenovo systems and components that have been certified for use with Azure Stack HCI, please refer to the Lenovo Press documents shown here.

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V1 Servers*, refer to the following URL:

<https://lenovopress.com/1p0866>

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V2 Servers*, refer to the following URL:

<https://lenovopress.com/1p1520>

These guides provide the latest details related to certification of Lenovo systems and components under the Microsoft Azure Stack HCI Program. Deploying Azure Stack HCI certified configurations takes the guesswork out of system configuration. You can rest assured that purchasing a ThinkAgile MX Certified Node or Appliance will provide a solid foundation with minimal obstacles along the way. These configurations are certified by Lenovo and validated by Microsoft for out-of-the-box optimization.

**Note:** It is strongly recommended to build S2D solutions based on Azure Stack HCI certified configurations and components. Deploying certified configurations ensures the highest levels of support from both Lenovo and Microsoft. The easiest way to ensure that configurations have been certified is to purchase Lenovo ThinkAgile MX solutions.

For more information about the Microsoft Azure Stack HCI program, see the following URL:

<https://docs.microsoft.com/en-us/windows-server/azure-stack-hci>

## Network infrastructure

The host network requirements for an HCI cluster are expressed by Microsoft based on the intended purpose of the network traffic type. Microsoft defines the following three traffic classifications:

- ▶ **Management traffic:** Traffic to or from outside the local cluster. For example, storage replica traffic or traffic used by the administrator for management of the cluster like Remote Desktop, Windows Admin Center, Active Directory, etc.
- ▶ **Compute traffic:** Traffic originating from or destined to a virtual machine.
- ▶ **Storage traffic:** Traffic using Server Message Block (SMB), including SMB-based live migration. This traffic is layer-2 traffic and is not routable.

Using a network adapter outside of its qualified traffic type is not supported. For more information, refer to the following Microsoft article:

<https://learn.microsoft.com/en-us/azure-stack/hci/concepts/host-network-requirements>

For 2- or 3-node clusters, it is possible to build a high-performance Azure Stack HCI solution without using network switches for East-West storage traffic inside the Azure Stack HCI cluster. In these solutions, the Mellanox (for RoCE configurations) or Intel E810 (for iWARP configurations) NICs are connected directly to each other, eliminating the need for a high-speed network switch architecture. This is particularly useful in small Remote Office / Branch Office (ROBO) environments or anywhere a small cluster would satisfy the need for high-performance storage. The sections “RoCE: 2-4 nodes, direct-connected” on page 46 and “iWARP: 2-4 nodes, direct-connected” on page 74 discuss these deployment scenarios in detail.

To build the Azure Stack HCI solutions described in this document that use a network switch for storage traffic, we used a pair of Lenovo ThinkSystem NE2572 RackSwitch network switches (Lenovo network switches are no longer available), which are connected to each node via 25GbE Direct Attach Copper (DAC) cables.

**Note:** We provide examples throughout this document that are based on deployments in our lab. Details related to IP subnetting, VLAN numbering, and similar environment-based parameters are shown for information purposes only. These types of parameters should be modified based on the requirements of your network environment.

For the deployment scenarios identified as “direct-connected,” no network switch is required to handle RDMA-based traffic. The only switches required in these scenarios is for North-South traffic between the Azure Stack HCI cluster and the organization’s intranet. Note that in this case, the network adapter used for North-South traffic must support both management and compute traffic, since there is no other connection between the HCI cluster and the outside world. Microsoft certification requirements are different for these two network traffic types, so make sure to choose an appropriate network adapter for this connectivity.

LOM ports in Lenovo V1 rack servers are based on the Intel x722 adapter, which is certified to carry management and compute traffic, as shown in Figure 4.



Figure 4 Microsoft HCI catalog showing Intel X722 network adapter certifications

## Server farm

To build the Azure Stack HCI solutions on which this document is based, we used between two and four Lenovo ThinkAgile MX Certified Nodes based on the ThinkSystem SR650 rack servers, equipped with multiple storage devices. Supported storage devices include HDD, SSD, and NVMe media types. A four-node cluster is the minimum configuration required to harness the failover capability of losing any two nodes, while sixteen nodes is the maximum number of nodes supported by Microsoft.

**Memory configurations:** Although many memory configurations are possible and supported, we highly recommend that you choose a balanced memory configuration. For more information, see one of the following Lenovo Press white papers:

For Lenovo ThinkSystem V1 servers, refer to *Balanced Memory Configurations with Second-Generation Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1089.pdf>

For Lenovo ThinkSystem V2 servers, refer to *Balanced Memory Configurations for 2-Socket Servers with 3rd-Gen Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1517.pdf>

**Use of RAID controllers:** Microsoft does not support any RAID controller attached to the storage devices used by Azure Stack HCI, regardless of a controller's ability to support "pass-through" or JBOD mode. As a result, the ThinkSystem 430-16i SAS/SATA HBAs are used in this solution. The ThinkSystem M.2 Mirroring Enablement Kit is used only for dual M.2 boot drives and has nothing to do with S2D functionality.

Lenovo has worked closely with Microsoft for many years to ensure our products perform smoothly and reliably with Microsoft operating systems and software. Our customers can leverage the benefits of our partnership with Microsoft by deploying Lenovo certified configurations for Microsoft Azure Stack HCI, which have been certified under the Microsoft Azure Stack HCI Program.

Deploying Lenovo ThinkAgile MX Certified Nodes for Azure Stack HCI solutions takes the guesswork out of system configuration. For details regarding Azure Stack HCI certified configurations for Azure Stack HCI, refer to the Lenovo Press documents shown below.

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V1 Servers*, refer to the following URL:

<https://lenovopress.com/1p0866>

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V2 Servers*, refer to the following URL:

<https://lenovopress.com/1p1520>

## Overview of installation tasks

This document specifically addresses the deployment of Storage Spaces Direct hyperconverged solutions. Although nearly all configuration steps presented apply to converged solutions as well, there are a few differences between these two deployment types. We have included notes regarding steps that do not apply to a converged solution. These notes are also included as comments in the PowerShell scripts shown.

A number of tasks need to be performed in order to configure this solution. If completed in a stepwise fashion, this is not a difficult endeavor. The high-level steps described in the remaining sections of this paper are as follows:

1. "General hardware preparation"
  - "Prepare servers and storage" on page 13
  - "Install operating system" on page 22
  - "Install Windows Server roles and features" on page 23
2. "Deployment scenarios" on page 25
  - "RoCE: 2-16 nodes with network switches" on page 25
  - "RoCE: 2-4 nodes, direct-connected" on page 46
  - "iWARP: 2-16 nodes with network switches" on page 56
  - "iWARP: 2-4 nodes, direct-connected" on page 74
3. "Create failover cluster" on page 84
  - "Perform Windows Update and join AD domain" on page 84
  - "Cluster validation and creation" on page 84
  - "Cluster file share witness" on page 85
4. "Enable and configure Storage Spaces Direct" on page 87
  - "Verify RDMA functionality" on page 87
  - "Create virtual disks" on page 90
5. "Cluster set creation" on page 93
  - "Introduction to cluster sets" on page 93
  - "Create the cluster set" on page 94

Figure 5 shows a process flow diagram that illustrates how to use this document to deploy any of the scenarios discussed.

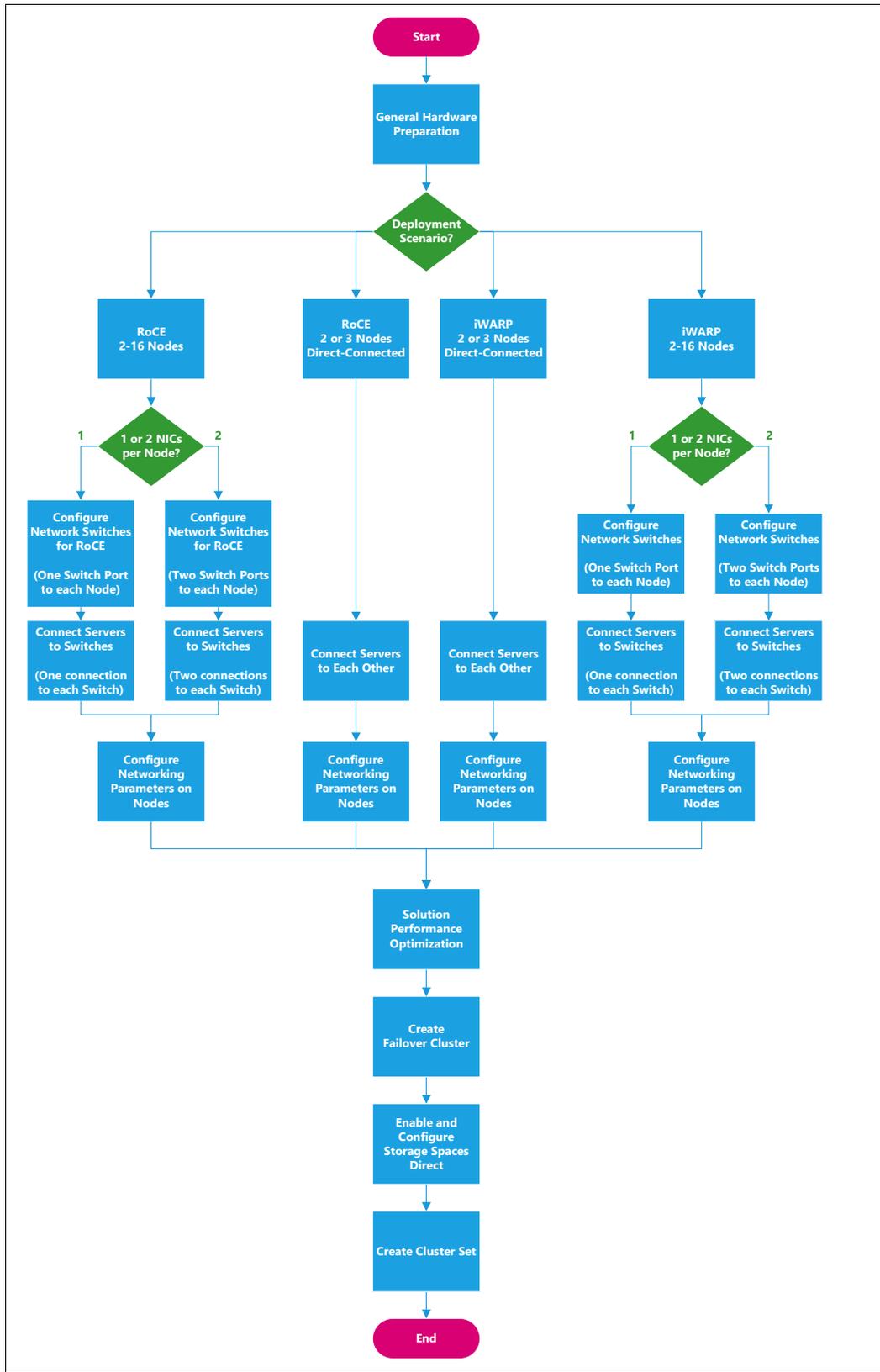


Figure 5 Process flow diagram illustrating flow through this document

# General hardware preparation

Several configuration steps must be followed for all servers that will become nodes in the Azure Stack HCI cluster, regardless of which deployment scenario is desired. These steps include updating system firmware, as well as installing and configuring the operating system.

## Prepare servers and storage

In this section, we describe the steps associated with preparing the physical servers and storage devices, which includes updating system firmware and configuring the RAID subsystem for the boot drive in the server nodes.

### Update system firmware according to Best Recipe

For Lenovo ThinkAgile MX Certified Nodes, ensure that the latest Best Recipe firmware and device driver versions are running on all nodes. For information regarding the current ThinkAgile MX Best Recipe, refer to the following URL:

<https://datacentersupport.lenovo.com/us/en/solutions/HT507406>

To simplify the process of downloading all firmware and device driver update packages for a given ThinkAgile MX Best Recipe, a single zip archive that includes all packages is available from the ThinkAgile MX Updates Repository site, which can be found at the following URL:

<https://thinkagile.lenovo.com/mx>

Lenovo offers multiple tools for updating firmware and device drivers on the nodes, including the Lenovo XClarity™ Integrator for Microsoft Windows Admin Center (LXCI for WAC), Lenovo XClarity Administrator (LXCA), Lenovo XClarity Provisioning Manager (LXPM), and Lenovo XClarity Essentials OneCLI. Since there are multiple benefits associated with using LXCI for WAC or LXCA to manage an Azure Stack HCI cluster, we recommend using one of these tools to update system firmware on the cluster nodes.

LXCI for WAC provides IT administrators with a smooth and seamless experience in managing Lenovo servers. IT administrators can manage Azure Stack HCI clusters as well as Lenovo ThinkAgile MX appliances and certified nodes for Microsoft Azure Stack HCI through the LXCI snap-ins integrated into WAC's cluster creation and Cluster-Aware Update (CAU) functions. Of particular interest is the ability of this tool to recognize and apply firmware and device driver updates based on the current ThinkAgile MX Best Recipe. For more information about LXCI for WAC, see the following URL:

<https://support.lenovo.com/us/en/solutions/ht507549>

LXCA is a centralized resource management solution that is aimed at reducing complexity, speeding response, and enhancing the availability of Lenovo server systems and solutions. LXCA provides agent-free hardware management for our servers, storage, network switches, hyperconverged and ThinkAgile solutions. LXCA can be used to monitor Azure Stack HCI clusters as well as Lenovo ThinkAgile MX appliances and certified nodes for Microsoft Azure Stack HCI, as well as to maintain firmware compliance with a published Best Recipe. For more information about LXCA, see the *Lenovo XClarity Administrator Product Guide* at the following URL:

<https://lenovopress.com/tips1200-lenovo-xclarity-administrator>

## Configure M.2 boot drive

We recommend using a dual M.2 boot drive configuration for OS boot, since this allows all other storage devices to become part of the S2D shared storage pool. Alternatively, you can use a pair of devices attached to a RAID adapter for OS boot. If doing so, make sure to create a RAID-1 mirror using the correct two devices. There are multiple ways to configure the M.2 Mirroring Enablement Kit, including System Setup (graphical or text-based), LXCA, OneCLI, and the XCC browser interface. The following steps use the XCC browser interface to configure a RAID-1 array for the operating system on the M.2 devices via the ThinkSystem M.2 Mirroring Enablement Kit:

1. Log in to the XCC browser interface on the server.
2. In the left navigation pane, select Server Configuration > RAID Setup.

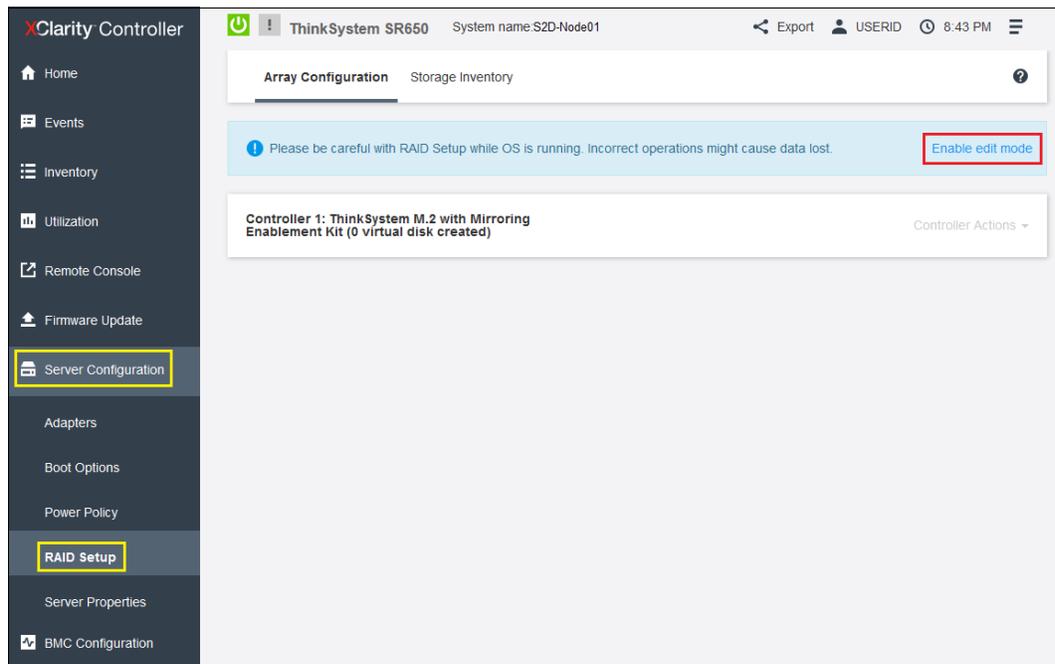


Figure 6 XCC browser interface with RAID Setup selected

3. Select the link to Enable edit mode.
4. For Controller 1, click the + Create Virtual Disk box.

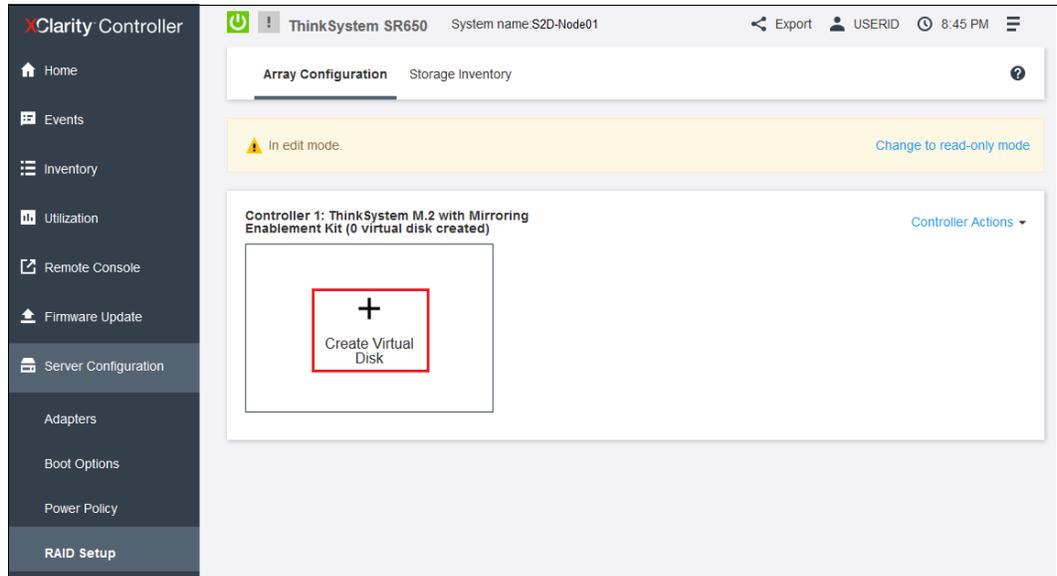


Figure 7 XCC browser interface showing Create Virtual Disk option for M.2 drives

5. On the Select Disk Drive/Disk Array page, select RAID 1 from the Select RAID level dropdown list.

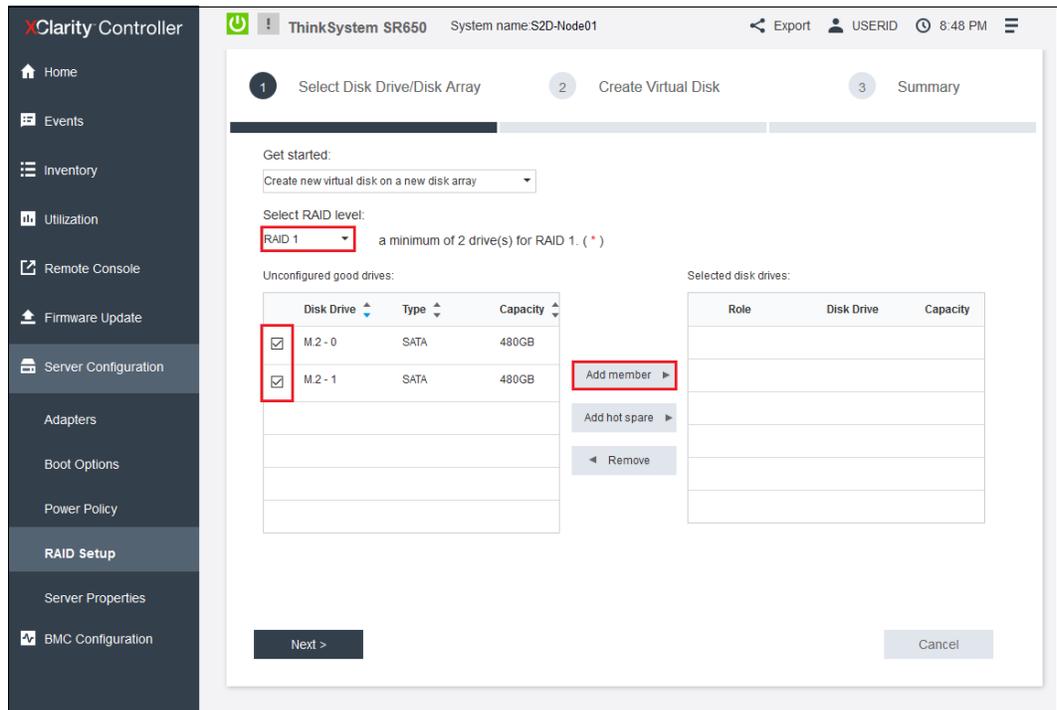


Figure 8 XCC browser interface while specifying M.2 drives to form a RAID-1 virtual disk

6. In the Unconfigured good drives list, select both M.2 drives and then click Add member.
7. With both M.2 drives moved to the Selected disk drives list on the right, click Next.

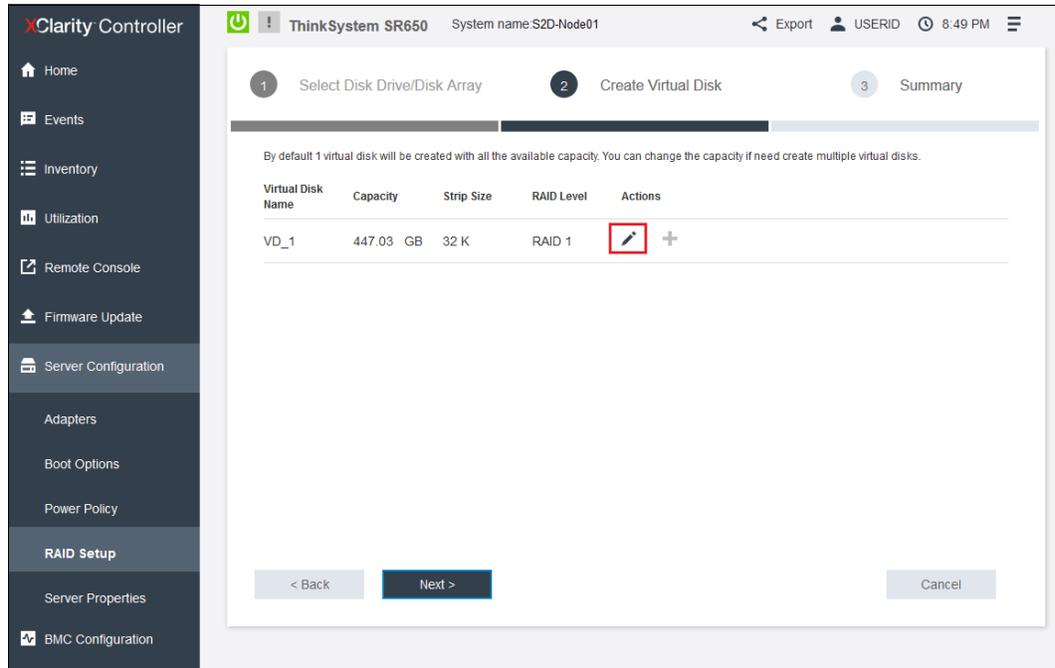


Figure 9 XVV browser interface during configuration of RAID-1 boot device

8. If you would like to give the virtual disk a name, click the pencil icon.
9. In the edit window that appears, change the Virtual Disk Name to something meaningful, like "Boot," click Apply and then click Next.

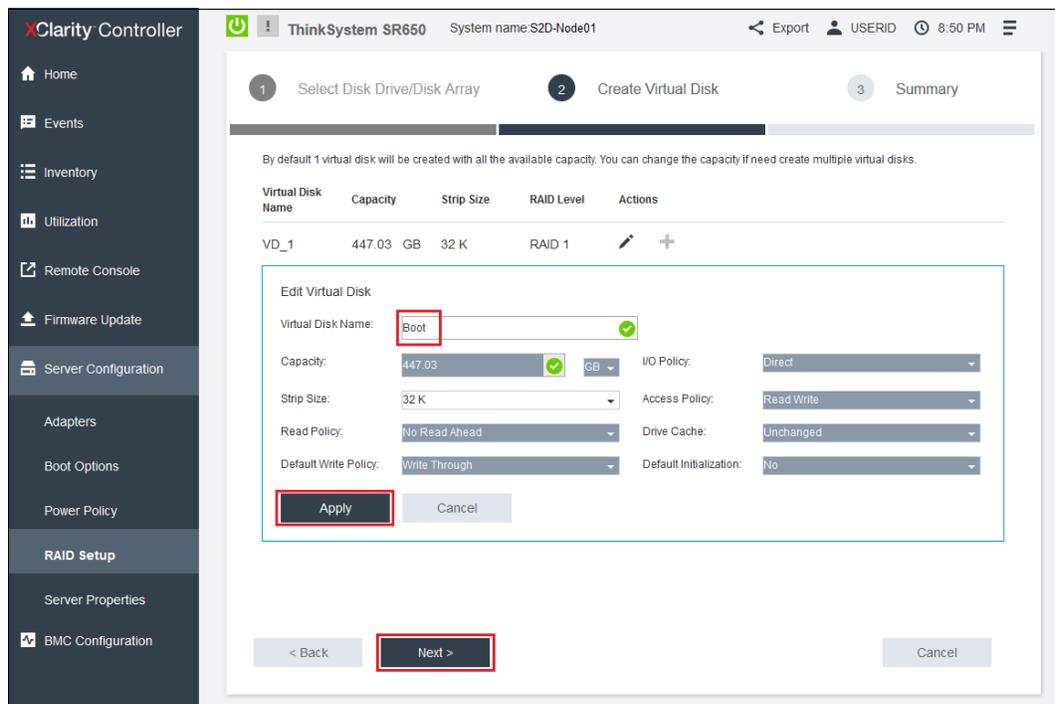


Figure 10 XCC browser interface with M.2 boot drive configured

10. Verify that all looks good and then click Start Creating.

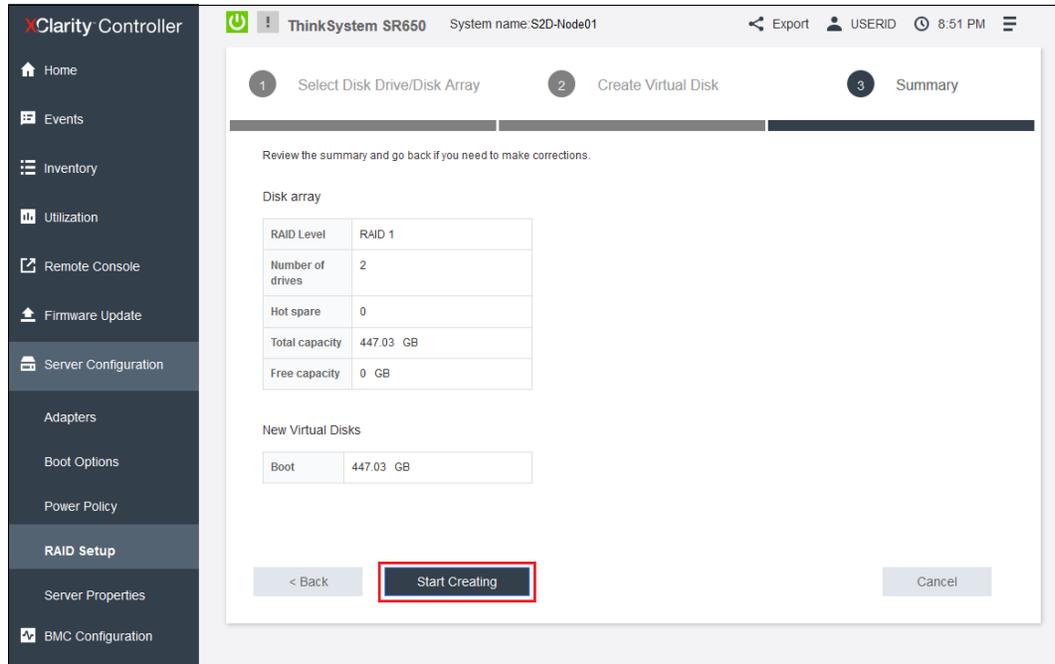


Figure 11 XCC browser interface with virtual disk ready for creation

11. After several seconds, the new virtual disk is shown. To protect this virtual disk, select Change to read-only mode.

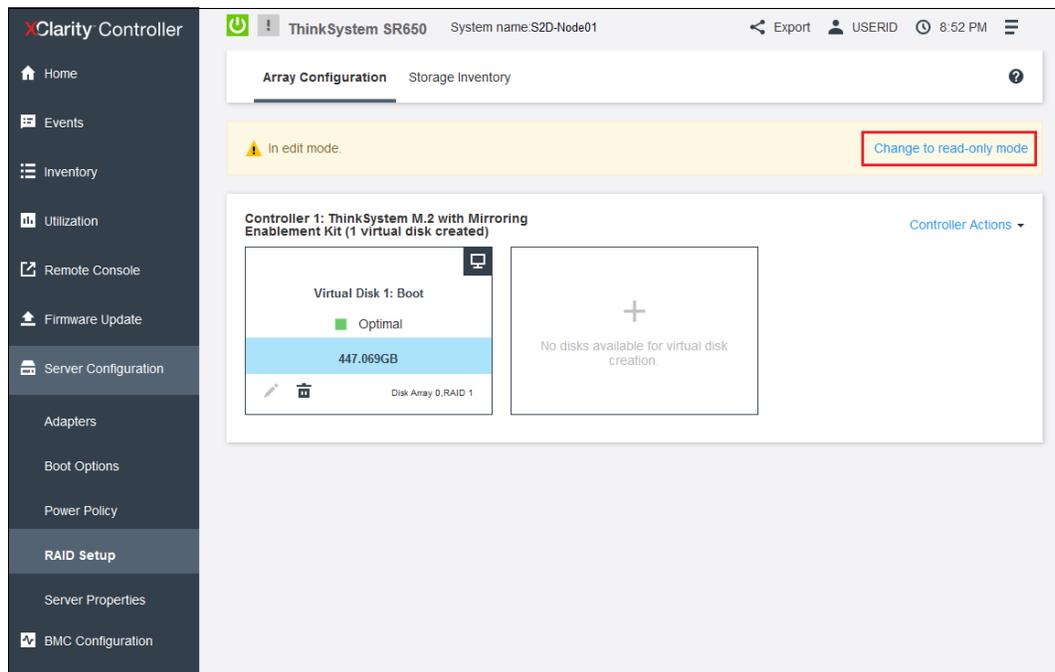


Figure 12 XCC browser interface with M.2 RAID-1 boot drive created

The mirrored M.2 boot disk is now ready for use. Leave all remaining storage devices (which are connected to an HBA) as unconfigured. They will be managed directly by the operating system when the time comes to create the storage pool.

## LOM and OCP network ports

The terms “LOM” and “OCP” are used throughout this document when referring to a particular type of network port used in Lenovo V1 and V2 rack servers. LOM ports are LAN On Motherboard network ports that might be present in V1 servers, depending on the system configuration. For V1 servers, only certain Intel network adapters are available, which have not been certified to carry East-West (storage) traffic. Therefore, LOM ports can only be used for North-South (management and compute) traffic. For this reason, LOM ports are often disabled before deploying an Azure Stack HCI cluster on Lenovo V1 servers.

OCP ports are similar, but are available from multiple manufacturers for Lenovo V2 rack servers. The Open Compute Project (OCP) NIC 3.0 specification provides the foundation for this high-density network adapter form factor. For Lenovo V2 rack servers, multiple network adapters that comply with the OCP form factor standard are available, including the Mellanox ConnectX-6 (for RoCEv2 implementations) and Intel E810 (for iWARP scenarios) network adapters.

Regardless of the network adapter type (LOM, OCP, or PCIe), make sure to verify that it is certified for the use you intend to have it perform. See “Network infrastructure” on page 9 for more information on the three traffic types (management, compute, and storage) defined by Microsoft for HCI clusters.

Figure 13 shows the placement of LOM/OCP ports in Lenovo ThinkSystem SR650 V1 and V2 rack servers as well as an example of a PCIe network adapter location. Although LOM and OCP ports are always located in the lower left corner at the rear of the server, a PCIe network adapter can be placed in various PCIe slots. There is no operational or functional difference between the OCP and PCIe version of a given network adapter. Many customers order the OCP version of a network adapter in order to save a PCIe slot in the server for other options.

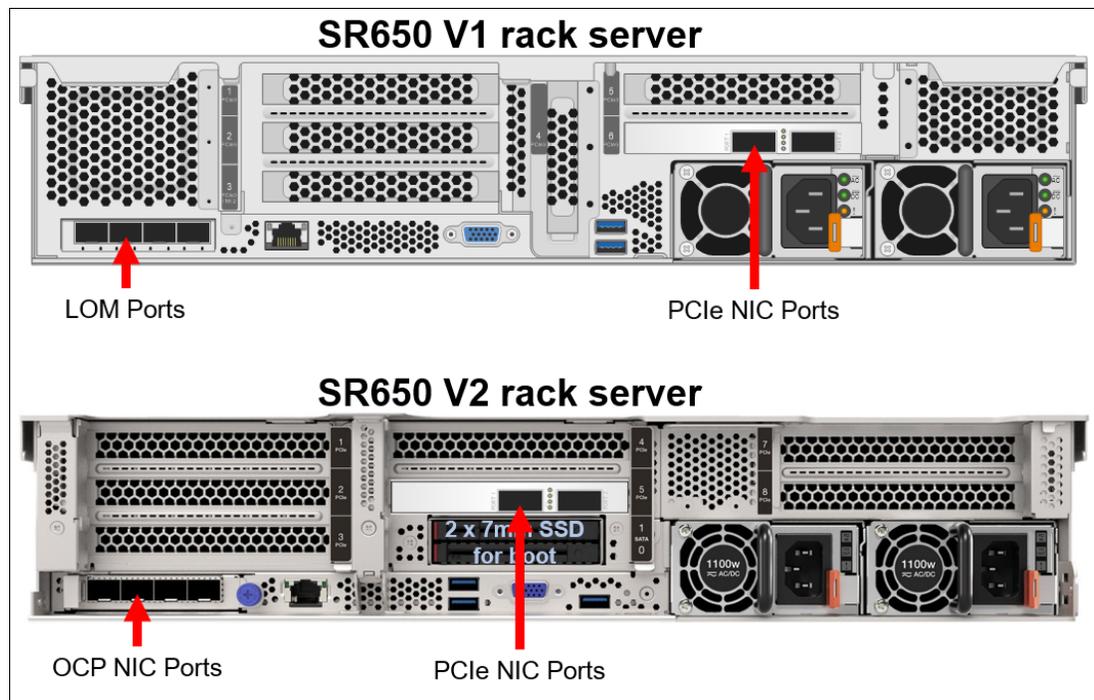


Figure 13 Placement of LOM/OCP ports in Lenovo ThinkSystem SR650 V1 and V2 rack servers

## Configure system settings

We need to change a couple of system settings to optimize system performance and also to ensure that unneeded network interfaces do not cause any issues with cluster validation and creation later.

### Operating Mode

The system Operating Mode should be changed to Maximum Performance to optimize system performance for its intended role as an Azure Stack HCI cluster node. To modify this system setting, follow these steps:

1. Reboot the server if necessary and enter the UEFI menu screen by pressing the **F1** key when prompted at the bottom of the screen.
  - a. If using the graphical system setup, navigate to **UEFI Setup > System Settings and then select Operating Modes**. Ensure that Choose Operating Mode is set to **Maximum Performance**. Once this setting change has been made, click the **Save** icon on the right, and then click **Back** to return to the System Settings screen. Proceed with Step 2 below.

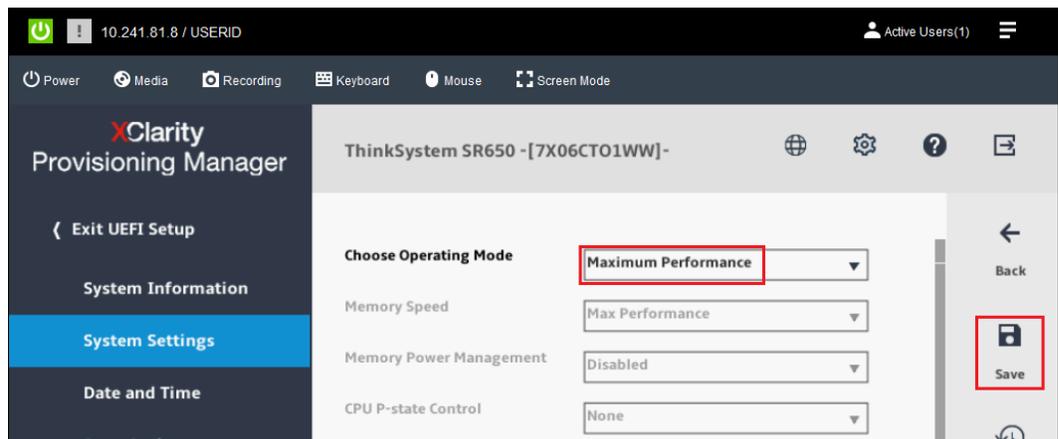


Figure 14 Graphical System Settings screen with Operating Mode set to Maximum Performance

- b. If using the text-based system setup, navigate to **System Settings > Operating Modes**. Ensure that Choose Operating Mode is set to **Maximum Performance**. If it is not, press Enter and use the arrow keys to select “Maximum Performance” before pressing Enter again. Once the setting change has been made, press the Esc key to return to the System Settings screen, and then proceed with Step 2 below.

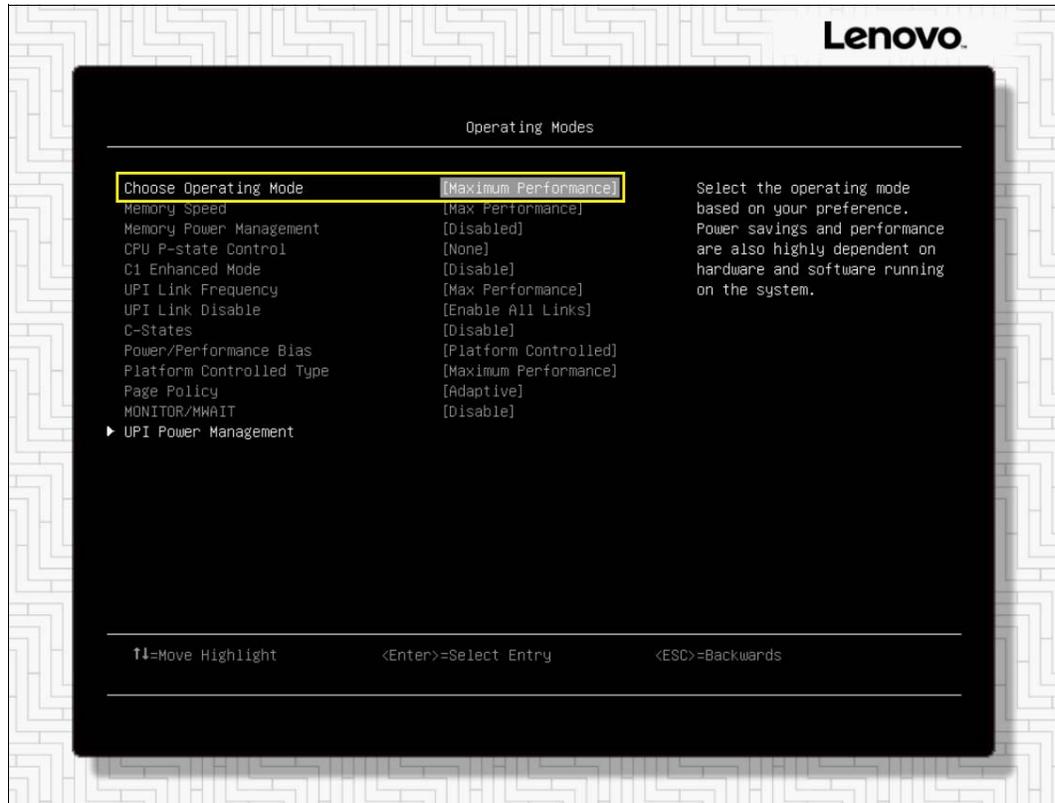


Figure 15 Text-based System Settings screen with Operating Mode set to Maximum Performance

2. Once the Operating Mode has been set to Maximum Performance, continue with the next steps to disable unneeded LOM ports if necessary. See “LOM and OCP network ports” on page 18 for a brief discussion regarding these types of network adapter ports. If no LOM ports are available in the nodes, exit System Setup, saving any changes made and reboot the system. Then proceed with Windows Server installation in “Install operating system” on page 22.

### **Disable unneeded LOM ports in UEFI (V1 servers)**

In order to avoid issues later with cluster validation and creation later, it is a good idea to disable any network interfaces that will not be used in the solution. For Lenovo ThinkSystem SR630 V1 and SR650 V1 servers, this typically includes LOM ports. See “LOM and OCP network ports” on page 18 for an overview of the similarities and differences between LOM ports in V1 servers and OCP ports in V2 servers.

The network interface for the “IBM USB Remote NDIS Network Device” that shows up after installing device drivers should be disabled, but from the OS - it should not be disabled in UEFI. More information is available about this device in each of the deployment scenarios.

For V1 servers, unused LOM ports should be disabled in UEFI so they are not visible to the OS. To disable unneeded LOM ports in UEFI, follow these steps:

1. From the System Setup screen, follow the instructions below based on whether you are using the graphical or text-based system setup.
  - a. If using the graphical system setup, in the main System Settings pane, navigate to **Devices and I/O Ports > Enable/Disable Onboard Device(s)** and scroll to the bottom of the device list.

- b. Disable either **Onboard LOM** to disable all LOM ports, or disable each unneeded port individually, as necessary. Once this setting change has been made, click the **Save** icon on the right, and then click **Exit UEFI Setup** to reboot the system.

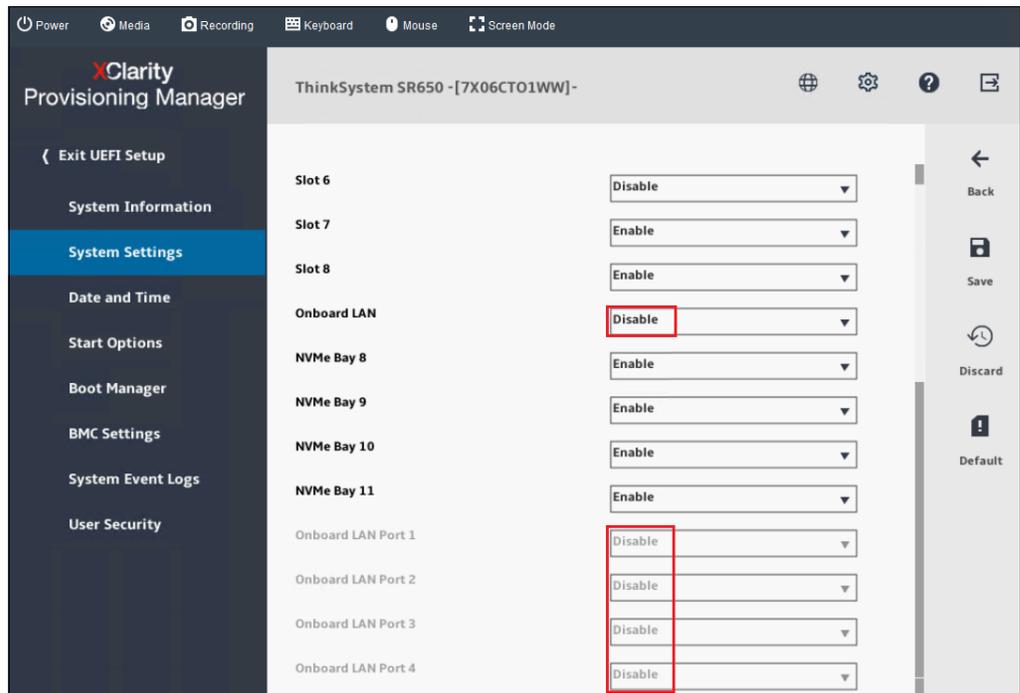


Figure 16 Graphical System Settings screen showing options for disabling LOM ports in UEFI

- c. If using the text-based system setup, from the System Settings page, navigate to **Devices and I/O Ports > Enable/Disable Onboard Device(s)**. Ensure that unneeded LOM ports are disabled. All LOM ports can be disabled at once by disabling **Onboard LAN**, or each individual LOM port can be disabled as necessary. Once this setting change has been made, press the Esc key repeatedly until prompted to save the new settings. Press the 'Y' key to save the settings and reboot the system.

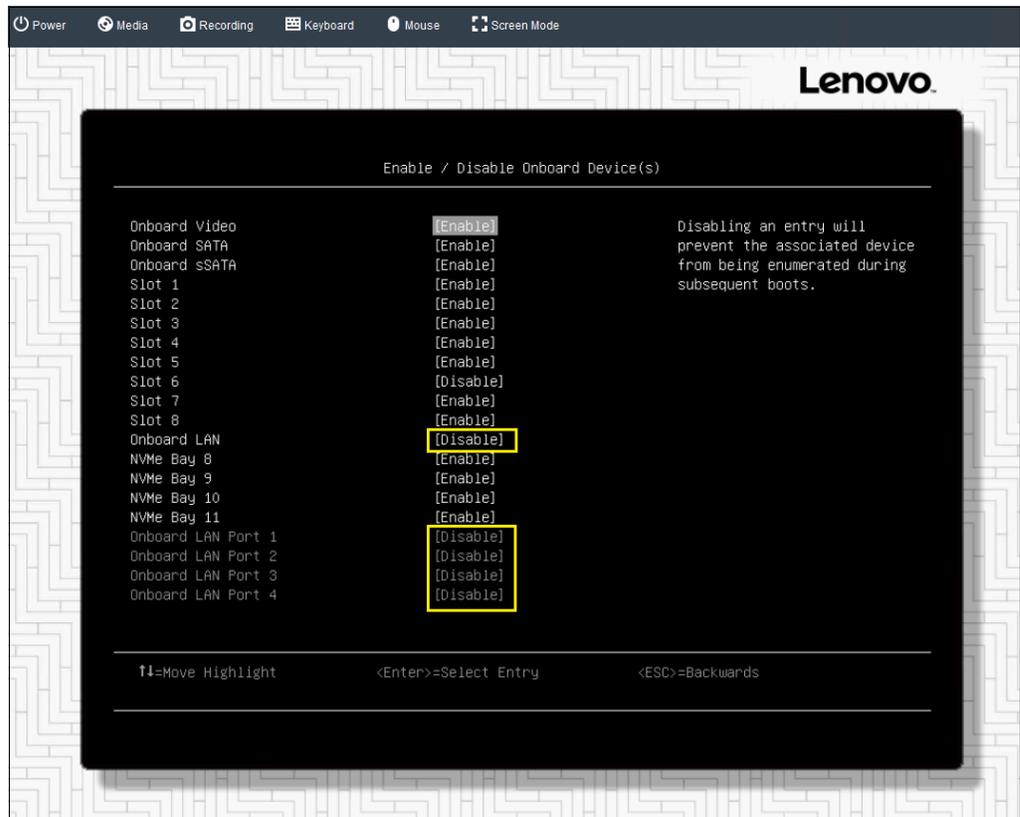


Figure 17 Text-based System Settings screen showing options for disabling LOM ports in UEFI

2. Once all system setting changes have been made, exit System Setup, saving changes when prompted. The system will reboot.

## Install operating system

ThinkSystem rack servers feature an advanced Baseboard Management Controller (BMC) called the "XClarity Controller" (XCC) to provide remote out-of-band management, including remote control and remote virtual media. You can install Windows Server from a variety of sources. Some examples include:

- ▶ Remote ISO media mount via the XCC
- ▶ Bootable USB media with the installation content
- ▶ Installation DVD
- ▶ Windows Deployment Services

Select the source that is appropriate for your situation. The following steps describe the installation:

1. With the method of OS installation selected, power the server on to begin the installation process for the OS.
2. Select the appropriate language pack, correct input device, and the geography, then select the desired OS edition, Desktop Experience (GUI) or Server Core, if necessary.
3. Select the virtual disk connected to the ThinkSystem M.2 Mirroring Enablement Kit as the target on which to install Windows Server.
4. Follow the prompts to complete installation of the OS.

## Install device drivers

It is critical to check the current ThinkAgile MX Best Recipe web page at the following URL to determine the exact versions of device drivers that have been certified for use by Azure Stack HCI:

<https://datacentersupport.lenovo.com/us/en/solutions/HT507406>

For any device driver that does not match the version shown in the latest ThinkAgile MX Best Recipe, download and run the EXE version of the device driver installer on each system that will become an Azure Stack HCI cluster node. Note that for HCI OS the EXE must be launched from the command line.

To simplify the process of downloading all firmware and device driver update packages for a given ThinkAgile MX Best Recipe, a single zip archive that includes all packages is available from the ThinkAgile MX Updates Repository site, which can be found at the following URL:

<https://thinkagile.lenovo.com/mx>

## Install Windows Server roles and features

Several Windows Server roles and features are used by this solution. It makes sense to install them all at the same time, then perform specific configuration tasks later. To make this installation quick and easy, use the PowerShell script shown in Example 1.

*Example 1 PowerShell script to install necessary server roles and features*

```
Install-WindowsFeature -Name File-Services
Install-WindowsFeature -Name Data-Center-Bridging # Optional for Intel E810 NICs
Install-WindowsFeature -Name Failover-Clustering -IncludeManagementTools
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart
```

**Note:** It is a good idea to install the Hyper-V role on all nodes even if you plan to implement the converged solution. Although you may not regularly use the storage cluster to host VMs, if the Hyper-V role is installed, you will have the option to deploy an occasional VM if the need arises.

Once the server has rebooted, check the status of all the required roles and features by using the PowerShell script shown in Example 2.

*Example 2 PowerShell script to check status of all required roles and features*

```
Get-WindowsFeature | ? installstate -eq installed | ? name -Like *File*
Get-WindowsFeature | ? installstate -eq installed | ? name -Like *Cluster*
Get-WindowsFeature | ? installstate -eq installed | ? name -Like Hyper-V*
Get-WindowsFeature | ? installstate -eq installed | ? name -Like *Bridging*
```

The returned output should be as shown in Example 3, with all roles and features shown as "Installed."

*Example 3 Expected status of required roles and features*

Display Name	Name	Install State
[X] File and Storage Services	FileAndStorage-Services	Installed
[X] File and iSCSI Services	File-Services	Installed
[X] File Server	FS-FileServer	Installed
[X] Failover Clustering	Failover-Clustering	Installed
[X] Failover Clustering Tools	RSAT-Clustering	Installed

	[X] Failover Cluster Management Tools	RSAT-Clustering-Mgmt	Installed
	[X] Failover Cluster Module for Windows ...	RSAT-Clustering-Powe...	Installed
[X] Hyper-V	[X] Hyper-V GUI Management Tools	Hyper-V-Tools	Installed
	[X] Hyper-V Module for Windows PowerShell	Hyper-V-PowerShell	Installed
[X] Data Center Bridging		Data-Center-Bridging	Installed

With the Windows Server roles and features installed on all systems that will become cluster nodes, we turn our attention to the particular deployment scenario that makes the most sense for the situation to be addressed. Figure 18 contains a portion of the process flow diagram from Figure 5 on page 12, showing this first decision point.

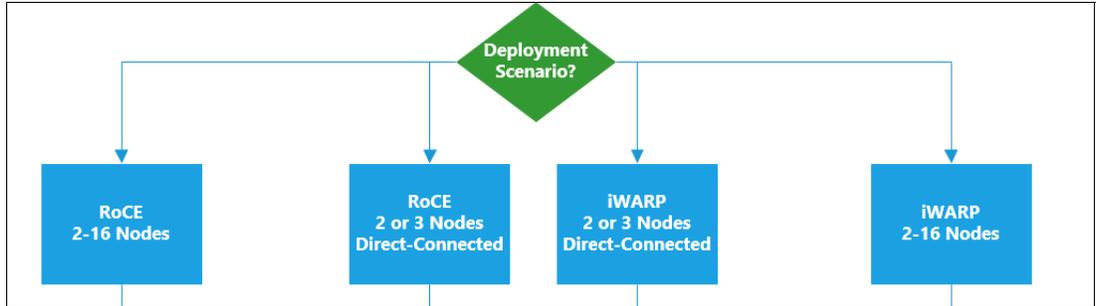


Figure 18 Portion of this document's process flow showing scenario decision point

# Deployment scenarios

This section contains multiple Azure Stack HCI deployment scenarios, which are based on the following variables:

- RoCE (Mellanox NICs) vs. iWARP (Intel E810 NICs) implementation of RDMA
- Number of nodes to be included in the Azure Stack HCI cluster
- Whether the nodes are direct-connected to each other

The RoCE specification requires specific capabilities and settings to be implemented on the network switches, including DCB, ETS, and PFC. The Lenovo switches mentioned in this document are fully compliant with these requirements and are extremely easy to configure.

For each scenario, the steps to deploy an Azure Stack HCI cluster are provided. There are specific hardware requirements for each scenario. For example, the only network adapters supported for any RoCE-based scenario are the Mellanox ConnectX-4 and ConnectX-6 adapters. For iWARP-based scenarios, the Intel E810 is the only NIC supported for Lenovo rack servers running a current operating system. A few Marvell (QLogic) adapters are supported only for Windows Server 2019. Make sure your hardware meets the requirements of the chosen deployment scenario before proceeding.

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V1 Servers*, refer to the following URL:

<https://lenovopress.com/1p0866>

To view or download the document *Lenovo Certified Configurations for Azure Stack HCI - V2 Servers*, refer to the following URL:

<https://lenovopress.com/1p1520>

To proceed with deployment, move to the scenario of interest and follow the steps outlined. The scenarios can be found in this document as follows:

- ▶ “RoCE: 2-16 nodes with network switches” on this page
- ▶ “RoCE: 2-4 nodes, direct-connected” on page 46
- ▶ “iWARP: 2-16 nodes with network switches” on page 56
- ▶ “iWARP: 2-4 nodes, direct-connected” on page 74

Once all configurations for your scenario have been successfully applied, continue the solution deployment by proceeding to “Create failover cluster” on page 84.

## RoCE: 2-16 nodes with network switches

This deployment scenario provides the steps to configure an Azure Stack HCI cluster that contains 2-16 nodes and uses the RoCE implementation of RDMA. In this scenario, the nodes are connected to network switches to carry all traffic. This scenario will cover the use of a single dual-port Mellanox network adapter in each node as well as two dual-port NICs. Using two NICs provides better performance and addresses the single point of failure issue that exists when using only a single network adapter. This is the deployment scenario on which previous versions of this document were based. Figure 19 shows a portion of the process flow diagram for this document and where this scenario fits.

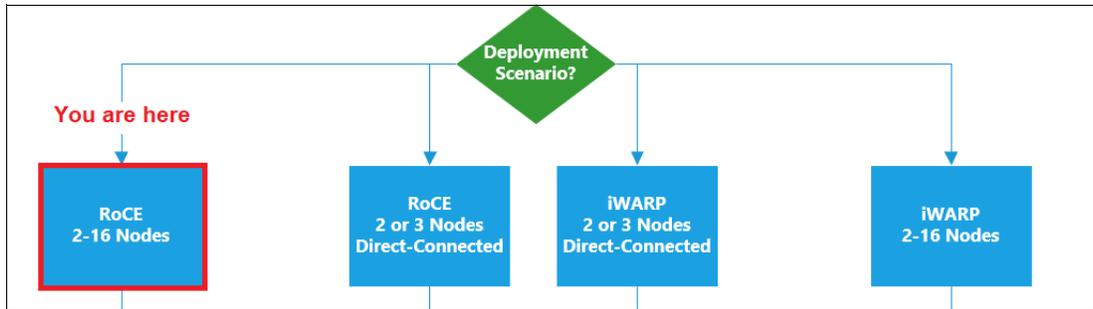


Figure 19 Portion of this document's process flow showing the general RoCE deployment scenario

## Overview

Figure 20 on page 26 shows high-level details of our 4-node configuration. The four server/storage nodes and two switches take up a combined total of 10 rack units of space. For smaller configurations that do not require network switches to handle RDMA traffic, such as a 2-node direct-connected configuration, the entire solution can take as little as 3.5" (2U) of vertical space.

	<p><b>Networking:</b> Two Lenovo ThinkSystem NE2572 RackSwitch network switches, each containing:</p> <ul style="list-style-type: none"> <li>▶ 48 ports at 10/25Gbps SFP28</li> <li>▶ 6 ports at 40/100Gbps QSFP28</li> </ul> <p><b>Compute:</b> Four Lenovo ThinkAgile MX Certified Nodes for S2D (in this case, SR650 servers), each containing:</p> <ul style="list-style-type: none"> <li>▶ Two Intel Xeon Platinum 8176 CPUs with 28 cores each, running at 2.10GHz</li> <li>▶ 384GB memory (balanced configuration, see Note below)</li> <li>▶ One or two dual-port 10/25GbE Mellanox ConnectX-4 Lx PCIe adapter(s) with RoCE support</li> </ul> <p><b>Storage</b> in each SR650 server:</p> <ul style="list-style-type: none"> <li>▶ Eight 3.5" hot swap HDDs and four SSDs at front</li> <li>▶ Two 3.5" hot swap HDDs at rear</li> <li>▶ ThinkSystem 430-16i SAS/SATA 12Gb HBA</li> <li>▶ M.2 Mirroring Kit with dual 480GB M.2 SSD for OS boot</li> </ul>
--	--

Figure 20 Solution configuration using ThinkAgile SXM Certified Nodes for S2D

**Note:** Although other memory configurations are possible, we highly recommend you choose a balanced memory configuration. For detailed information regarding what constitutes a balanced memory configuration, see the following documents.

For Lenovo V1 rack servers, refer to *Balanced Memory Configurations with Second-Generation Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1089>

For Lenovo V2 rack servers, refer to *Balanced Memory Configurations for 2-Socket Servers with 3rd-Generation Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1517>

Figure 21 shows the layout of the drives and Mellanox network adapters in a Lenovo ThinkSystem SR650 rack server. There are 14 x 3.5" hot-swap drive bays in the SR650, 12 at the front of the server and two at the rear of the server. Four bays contain 1.6TB SSD devices, while the remaining ten drives are 6TB SATA HDDs. These 14 drives form the tiered storage pool of Azure Stack HCI and are connected to the ThinkSystem 430-16i SAS/SATA 12Gb HBA. In addition to the storage devices that will be used by Azure Stack HCI, a dual 480GB M.2 SSD, residing inside the server, is configured as a mirrored (RAID-1) OS boot volume.

If a single dual-port Mellanox network adapter is used, it should be installed in PCI slot 6. If two dual-port Mellanox network adapters are used, the first NIC should be installed in PCI slot 6. Although the second NIC can be installed in PCI slots 1-4, the only available slot in our configuration is PCI slot 4, since slots 1-3 are not available when using the Rear HDD Kit. Figure 21 on page 27 shows this NIC placement, which ensures that CPU load for processing network traffic is balanced between physical processors.

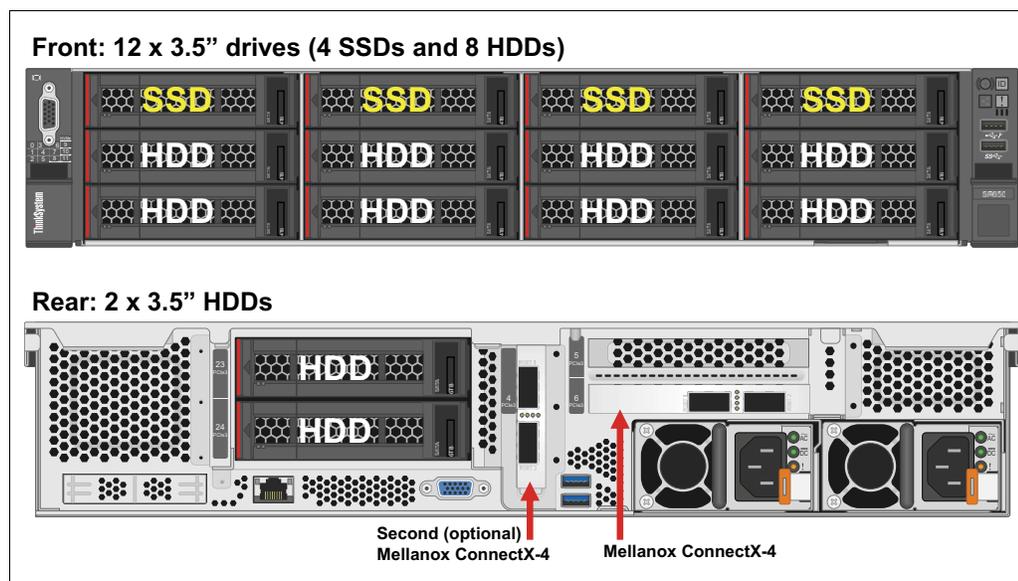


Figure 21 Lenovo ThinkSystem SR650 storage subsystem configured for RoCE

Network cabling of this solution is straight-forward, with each server being connected to each switch to enhance availability. Each system contains one or two dual-port 10/25GbE Mellanox ConnectX-4 Lx adapter(s) to handle operating system traffic and storage communication.

For a completely new environment including switches, we provide recommended network cable diagrams in “Connect servers to switches” on page 32. If using existing switches, servers can be connected to any properly configured port on each switch. If a single dual-port network adapter is used, each server is connected to each switch via a single network cable. If two dual-port network adapters are used, each server is connected to each switch twice, once from each NIC. As a best practice, network cabling is not completed until after the switches have been configured.

For the converged solution, the servers are configured with 192 GB of memory, rather than 384 GB, and the CPU has 16 cores instead of 28 cores. The higher-end specifications of the hyperconverged solution are to account for the dual functions of compute and storage that each server node will take on, whereas in the converged solution, there is a separation of duties, with one server farm dedicated to Azure Stack HCI and a second devoted to Hyper-V hosting.

## Configure network switches for RoCE

This section includes the steps required to configure Lenovo network switches to support RoCE for those deployment scenarios that require network switches to handle the East-West storage traffic between Azure Stack HCI cluster nodes.

In order to use RoCE for Azure Stack HCI, the network switches used by the solution must be configured appropriately. For Lenovo switches, this is an extremely easy task, as described below.

Windows Server 2019 includes a feature called SMB Direct, which supports the use of network adapters that have the Remote Direct Memory Access (RDMA) capability. Network adapters that support RDMA can function at full speed with very low latency, while using very little CPU. For workloads such as Hyper-V or Microsoft SQL Server, this enables a remote file server to resemble local storage.

SMB Direct provides the following benefits:

- ▶ **Increased throughput:** Leverages the full throughput of high speed networks where the network adapters coordinate the transfer of large amounts of data at line speed.
- ▶ **Low latency:** Provides extremely fast responses to network requests and, as a result, makes remote file storage feel as if it is directly attached block storage.
- ▶ **Low CPU utilization:** Uses fewer CPU cycles when transferring data over the network, which leaves more power available to server applications, including Hyper-V.

Leveraging the benefits of SMB Direct comes down to a few simple principles. First, using hardware that supports SMB Direct and RDMA is critical. This solution utilizes a pair of Lenovo ThinkSystem NE2572 RackSwitch Ethernet switches and one or two dual-port 10/25GbE Mellanox ConnectX-4 Lx PCIe adapter(s) for each node, depending on the high availability requirements of the organization. If using two dual-port network adapters in each server, some (but not all) of the commands shown in this section will need to be modified or repeated, as noted.

In order to leverage the SMB Direct benefits listed above, a set of cascading requirements must be met. Using RDMA over Converged Ethernet (RoCE) requires a lossless fabric, which is typically not provided by standard TCP/IP Ethernet network infrastructure, since the TCP protocol is designed as a “best-effort” transport protocol. Datacenter Bridging (DCB) is a set of enhancements to IP Ethernet, which is designed to eliminate loss due to queue overflow, as well as to allocate bandwidth between various traffic types.

To sort out priorities and provide lossless performance for certain traffic types, DCB relies on Priority Flow Control (PFC). Rather than using the typical Global Pause method of standard Ethernet, PFC specifies individual pause parameters for eight separate priority classes. Since the priority class data is contained within the VLAN tag of any given traffic, VLAN tagging is also a requirement for RoCE and, therefore SMB Direct.

The following configuration commands need to be executed on *both* switches. We start by enabling Converged Enhanced Ethernet (CEE), which automatically enables Priority-Based Flow Control (PFC) for all Priority 3 traffic on all ports. Enabling CEE also automatically configures Enhanced Transmission Selection (ETS) so that at least 50% of the total bandwidth is always available for our storage traffic. These automatic default configurations provided by Lenovo switches are suitable for our solution. The commands shown in Example 4 apply whether using one or two dual-port NICs in each server.

*Example 4 Enable CEE on the switch*

---

```
enable  
configure
```

cee enable

---

**Note:** All switch configuration examples in this document use commands based on Lenovo switches running CNOS v10.10.1.0. The command syntax has changed significantly since the previous edition of this document.

In addition to the priorities that are configured by default with the “cee enable” command, it is a best practice to define a priority for the Azure Stack HCI cluster heartbeat (we will use priority 7). Specifying a minimal bandwidth reservation (of 1%) for this priority ensures that cluster heartbeat is not lost in the event of very high storage traffic through the switches. At the same time, we can configure precise bandwidth allocation to all priority groups and add descriptions to each to aid in troubleshooting in the future.

The commands shown in Example 5 on page 29 are used to perform these tasks. The final command in the example sets the bandwidth reservations for priorities 0, 3, and 7 to 49%, 50%, and 1%, respectively. These correspond to Default, RoCE, and Cluster Heartbeat traffic, respectively.

*Example 5 Commands used to configure ETS for RoCE on Lenovo NE2572 network switch*

---

```
cee ets priority-group pgid 0 priority 0 1 2 4 5 6
cee ets priority-group pgid 0 description Default
cee ets priority-group pgid 3 description RoCE
cee pfc priority 3 description RoCE
cee ets priority-group pgid 7 priority 7
cee ets priority-group pgid 7 description Cluster-HB
cee ets bandwidth-percentage 0 49 3 50 7 1
```

---

It is a good idea to do a quick check to see that all settings are as intended by executing the “show cee” command on the switch. Example 6 shows the first chunk of what is returned. You can press the “Q” key at the first “--More--” prompt to halt the rest of the return.

*Example 6 Results of “show cee” command after ETS has been set for RoCE and cluster heartbeat*

---

```
S2D-TOR1#show cee
```

```
CEE feature setting: On
```

```
ETS information:
```

```
ETS Global Admin Configuration:
```

PGID	BW%	COSq	Priorities	Description
0	49	0	0 1 2 4 5 6	Default
1	0	NA		
2	0	2		
3	50	3	3	RoCE
4	0	4		
5	0	5		
6	0	6		
7	1	7	7	Cluster-HB
15	NA	1		

---

After enabling CEE and configuring ETS for cluster heartbeat, we configure the VLANs. Although we could use multiple VLANs for different types of network traffic (storage, client, management, cluster heartbeat, Live Migration, etc.), the simplest choice is to use a single VLAN (we use VLAN 12 in our examples) to carry all our SMB Direct solution traffic.

Employing 25GbE links makes this a viable scenario. As previously noted, enabling VLAN tagging is important in this solution, since the RoCE implementation of RDMA requires it.

Example 7 shows the commands required. Make sure to adjust the “interface ethernet 1/1-4” command for the number of nodes and switch ports to be used. The example shows the command for 4 nodes using a single dual-port NIC. For more nodes or if using two dual-port NICs in each server, make sure to configure all required switch ports. For example, if configuring the switches for an 8-node Azure Stack HCI cluster in which all nodes contain two dual-port NICs, the command should be “interface ethernet 1/1-16” to configure enough switch ports to handle 2 connections from each of 8 nodes.

*Example 7 Establish VLAN for all solution traffic*

---

```
vlan 12
name RoCE
exit

interface ethernet 1/1-4
switchport mode trunk
switchport trunk allowed vlan 12
spanning-tree bpduguard enable
spanning-tree port type edge
no shutdown
exit
```

---

It is helpful to add a description to each switch port configuration to aid in troubleshooting later. Typically, the description would indicate the destination of the port connection. This could be as simple as the Azure Stack HCI node name or might also include details regarding to which server network adapter and port number the switch port is connected. To add a description to a switch port configuration, use the “description” command. Example 8 shows the commands used to specify a description for each of the switch ports (1-4) configured above.

*Example 8*

---

```
interface ethernet 1/1
description S2D-Node01
interface ethernet 1/2
description S2D-Node02
interface ethernet 1/3
description S2D-Node03
interface ethernet 1/4
description S2D-Node04
exit
```

---

**Note:** Another best practice that can be extremely helpful for troubleshooting is to label each network cable on both ends to indicate its source and destination. This can be invaluable if an internal server component must ever be replaced. If cable management arms are not in use, all cabling must be removed from the server in order to slide it out of the rack for component replacement. Having a label on each cable will help ensure that correct connections are made once the server is slid back into the rack.

The switch is now configured to carry RDMA traffic via RoCE. Next, we create a Link Aggregation Group (LAG) between two ports on each switch and then create an InterSwitch Link (ISL) between the pair of switches using this LAG. We then establish a virtual Link Aggregation Group (vLAG) across the ISL, which is used to ensure redundant connectivity when communicating with upstream switches in the corporate network. This creates an

automated network failover path from one switch to the other in case of entire switch or individual port failure.

The LAG is created between a pair of 100GbE ports on each switch. We use the first two 100GbE ports, 49 and 50, for this purpose. Physically, each port is connected to the same port on the other switch using a 100Gbps QSFP28 cable. Configuring the ISL is a simple matter of joining the two ports into a port trunk group. We establish a vLAG across this ISL, which extends network resiliency all the way to the Azure Stack HCI cluster nodes and their NIC teams using vLAG Instances. Example 9 shows the commands to run, which apply whether using one or two dual-port NICs in each server.

*Example 9 Configure an ISL between switches and establish a vLAG for resiliency*

---

```
interface ethernet 1/49-50
switchport mode trunk
switchport trunk allowed vlan all
channel-group 100 mode active
exit

interface port-channel 100
switchport mode trunk
switchport trunk allowed vlan all
exit

vlag tier-id 100
vlag isl port-channel 100
vlag enable
exit
```

---

Establishing the LAG, ISL, and vLAG as discussed above offers the following benefits:

- ▶ Enables Azure Stack HCI cluster nodes to use a LAG across two switches
- ▶ Spanning Tree Protocol (STP) blocked interfaces are eliminated
- ▶ Topology loops are also eliminated
- ▶ Enables the use of all available uplink bandwidth
- ▶ Allows fast convergence times in case of link or device failure
- ▶ Allows link-level resiliency
- ▶ Enables high availability

To verify the completed vLAG configuration, use the "show vlag information" command. A portion of the output of this command is shown in Example 10. Run this command on both switches and compare the outputs. There should be no differences between the Local and Peer switches in the "Mis-Match Information" section. Also, in the "Role Information" section, one switch should indicate that it has the Primary role and its Peer has the Secondary role. The other switch should indicate the opposite (i.e. it has the Secondary role and its Peer has the Primary role).

*Example 10 Verification of completed vLAG configuration*

---

```
show vlag information
Global State           : enabled
VRRP active/active mode : enabled
vLAG system MAC       : 08:17:f4:c3:dd:63
ISL Information:
  PCH   Ifindex   State   Previous State
  -----+-----+-----+-----
  100   100100    Active  Inactive

Mis-Match Information:
```

	Local	Peer
Match Result	: Match	: Match
Tier ID	: 100	: 100
System Type	: NE2572	: NE2572
OS Version	: 10.10.x.x	: 10.10.x.x

Role Information:

	Local	Peer
Admin Role	: Primary	: Secondary
Oper Role	: Primary	: Secondary
Priority	: 0	: 0
System MAC	: a4:8c:db:bb:7f:01	: a4:8c:db:bb:88:01

Consistency Checking Information:

State	: enabled
Strict Mode	: disabled
Final Result	: pass

Once the configuration is complete on the switch, we need to copy the running configuration to the startup configuration. Otherwise, our configuration changes would be lost once the switch is reset or reboots. This is achieved using the save or write command (they are equivalent), as shown in Example 11 below.

*Example 11 Use the write command to copy the running configuration to the startup configuration*

```
write
```

Repeat the entire set of commands above (Example 4 on page 28 through Example 11) on the other switch, defining the same VLAN and port trunk on that switch. Since we are using the same ports on both switches for identical purposes, the commands that are run on each switch are identical. Remember to commit the configuration changes on both switches using the "save" or "write" command.

**Note:** The steps and commands shown above are intended for use with Lenovo RackSwitch network switches running CNOS, including the G8272, NE2572, and NE10032. If the solution uses another switch model or switch vendor's equipment, it is essential to apply the equivalent command sets to the switches. The commands themselves may differ from what is stated above, but it is imperative that the same functions are executed on the switches to ensure proper operation of this solution.

### **Connect servers to switches**

To provide redundant network links in the event of a network port or external switch failure when using a single dual-port network adapter in each server, the recommendation calls for the connection from Port 1 on the Mellanox adapter to be connected to a port on the first switch ("Switch 1"), plus a connection from Port 2 on the same Mellanox adapter to be connected to an available port on the second switch ("Switch 2"). See Figure 22 on page 34 for the network cable diagram and Table 2 on page 35 for the network point-to-point connections for this scenario, using a single dual-port physical network adapter in each cluster node.

As a final bit of network cabling, we establish an ISL between our pair of switches to support the redundant node-to-switch cabling described above. To do this, we need redundant high-throughput connectivity between the switches, so we connect Ports 49 and 50 on each switch to each other using a pair of 100Gbps QSFP28 cables.

**Note:** In both network cable diagrams below, the port number indicators on the switches indicate the node to which they are connected. The ISL connections are between Ports 49 and 50 on each switch.

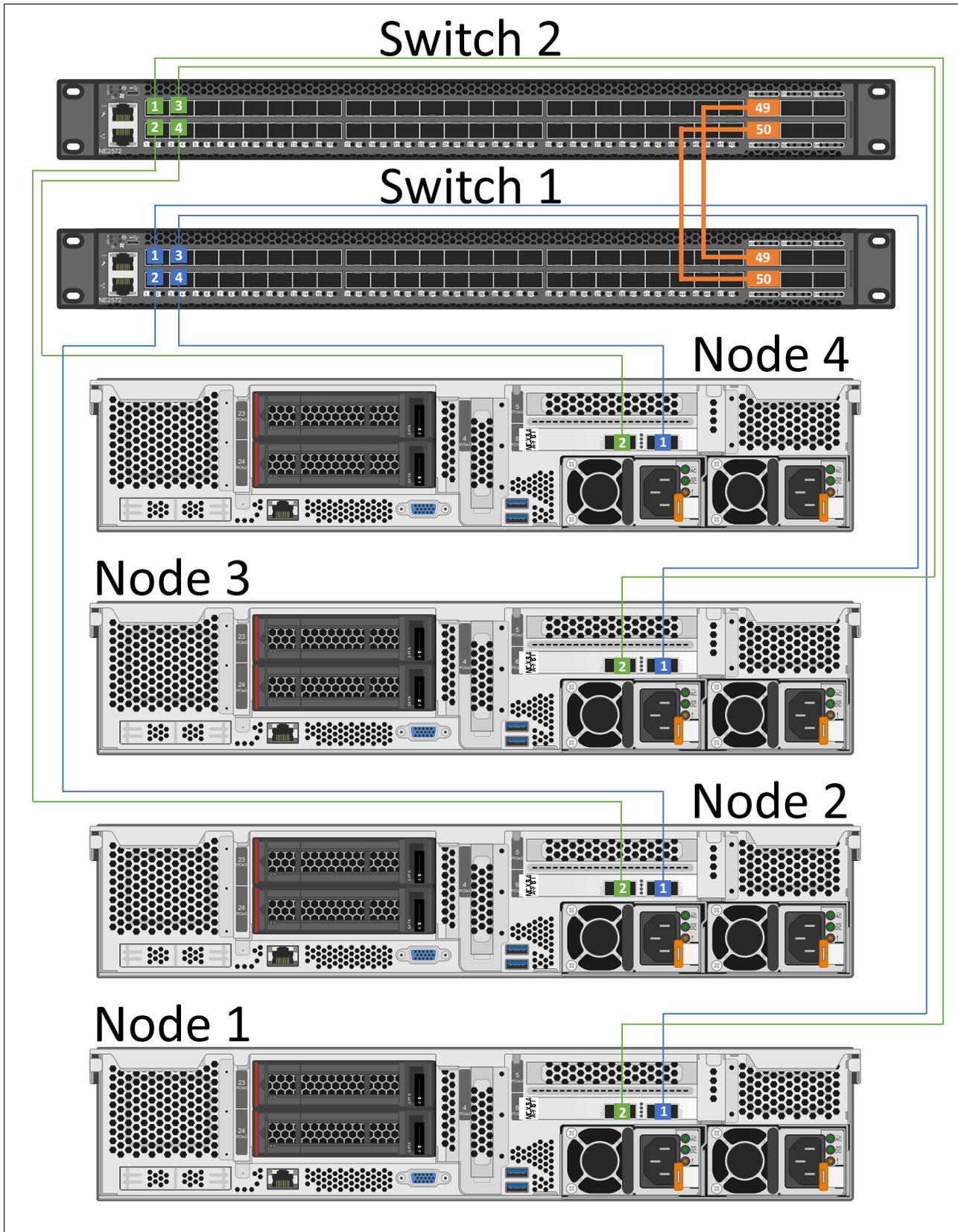


Figure 22 Switch to node connectivity using 10GbE / 25GbE AOC / DAC cables and a single dual-port NIC in each node

Table 2 shows the network point-to-point connections for this scenario when using a single dual-port physical network adapter in each cluster node.

*Table 2 Source and destination ports for a four-node cluster using a single dual-port network adapter*

<b>Source Device</b>	<b>Source Port</b>	<b>Destination Device</b>	<b>Destination Port</b>
Node 1	pNIC1-Port1	Switch 1	Port 1
Node 1	pNIC1-Port2	Switch 2	Port 1
Node 2	pNIC1-Port1	Switch 1	Port 2
Node 2	pNIC1-Port2	Switch 2	Port 2
Node 3	pNIC1-Port1	Switch 1	Port 3
Node 3	pNIC1-Port2	Switch 2	Port 3
Node 4	pNIC1-Port1	Switch 1	Port 4
Node 4	pNIC1-Port2	Switch 2	Port 4

To increase performance and availability even further and to avoid the single point of failure associated with a single network adapter carrying all traffic, it is possible to configure the servers with two dual-port Mellanox ConnectX-4 network adapters. Even if one of these NICs fails completely, impacting both of its ports, the second NIC will ensure that network traffic is maintained. Using two dual-port NICs has the additional benefit of doubling the network throughput of the solution. Adding a second dual-port NIC to each server simply means that each server is connected to each of the two network switches twice, once from each NIC, as shown in Figure 23. Nodes 3 and 4 are not shown in this Figure to make the network connection lines more clear. Note also that the switch-to-switch connections required for the ISL are identical, regardless of the number of NICs used in each server. Table 3 on page 36 shows the network point-to-point connections for this scenario, using two dual-port physical network adapters in each cluster node.

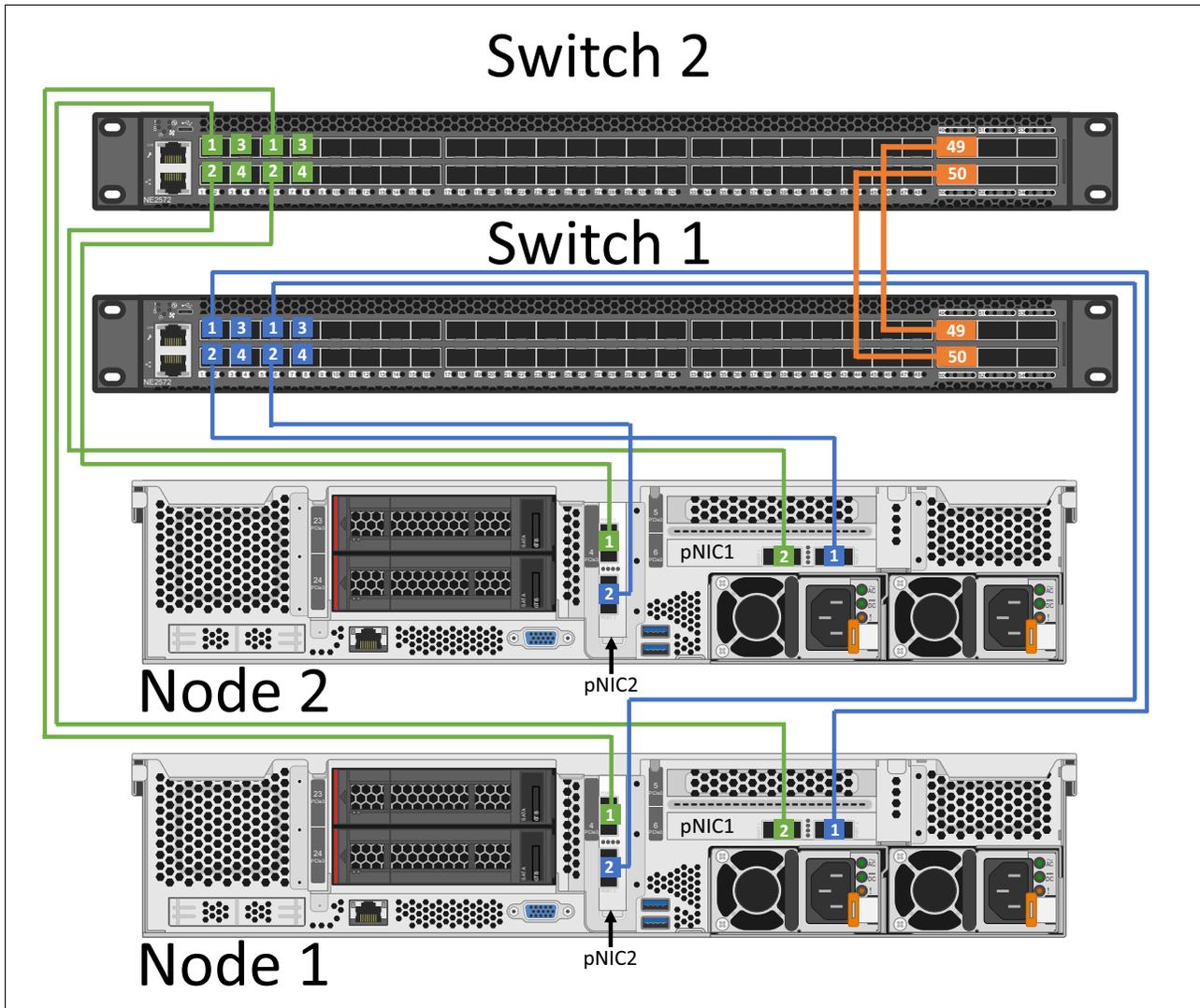


Figure 23 Switch to node connectivity using 10GbE / 25GbE AOC / DAC cables and two dual-port NICs in each node

Table 3 shows the network point-to-point connections for this scenario when using two dual-port physical network adapters in each cluster node.

Table 3 Source and destination ports for a four-node cluster using two dual-port network adapters

Source Device	Source Port	Destination Device	Destination Port
Node 1	pNIC1-Port1	Switch 1	Port 1
Node 1	pNIC1-Port2	Switch 2	Port 1
Node 1	pNIC2-Port1	Switch 2	Port 5
Node 1	pNIC2-Port2	Switch 1	Port 5
Node 2	pNIC1-Port1	Switch 1	Port 2
Node 2	pNIC1-Port2	Switch 2	Port 2
Node 2	pNIC2-Port1	Switch 2	Port 6
Node 2	pNIC2-Port2	Switch 1	Port 6

Source Device	Source Port	Destination Device	Destination Port
Node 3	pNIC1-Port1	Switch 1	Port 3
Node 3	pNIC1-Port2	Switch 2	Port 3
Node 3	pNIC2-Port1	Switch 2	Port 7
Node 3	pNIC2-Port2	Switch 1	Port 7
Node 4	pNIC1-Port1	Switch 1	Port 4
Node 4	pNIC1-Port2	Switch 2	Port 4
Node 4	pNIC2-Port1	Switch 2	Port 8
Node 4	pNIC2-Port2	Switch 1	Port 8

### Configure networking parameters

To increase performance and availability, we need to leverage the virtual network capabilities of Hyper-V on each host by creating SET-enabled teams from the 25GbE ports on the Mellanox adapter(s). From this a virtual switch (vSwitch) is defined and logical network adapters (vNICs) are created to facilitate the operating system and storage traffic. Note that for the converged solution, the SET team, vSwitch, and vNICs do not need to be created. However, we generally do this anyway, just in case we'd like to run a VM or two from the storage cluster occasionally.

We make extensive use of PowerShell commands and scripts throughout this document to configure various aspects of the Azure Stack HCI environment. The commands and scripts used to configure networking parameters on the servers in this section can be used with minimal modification if you take a moment now to name the physical network adapter ports according to Table 4 before working through this section. Alternatively, you can use your own naming convention for these ports, but in this case, remember to modify the PowerShell commands appropriately.

Table 4 Friendly names of network adapter ports used in this scenario

	Mellanox ConnectX-4	PCI Slot
First NIC, first port	"pNIC1-Port1"	6
First NIC, second port	"pNIC1-Port2"	6
Second NIC, first port (if used)	"pNIC2-Port1"	4
Second NIC, second port (if used)	"pNIC2-Port2"	4

### One dual-port Mellanox adapter in each server

If using one dual-port Mellanox NIC in each server, a single SET team is created across both ports of the network adapter. Figure 24 shows various details of this SET team and how it is used.

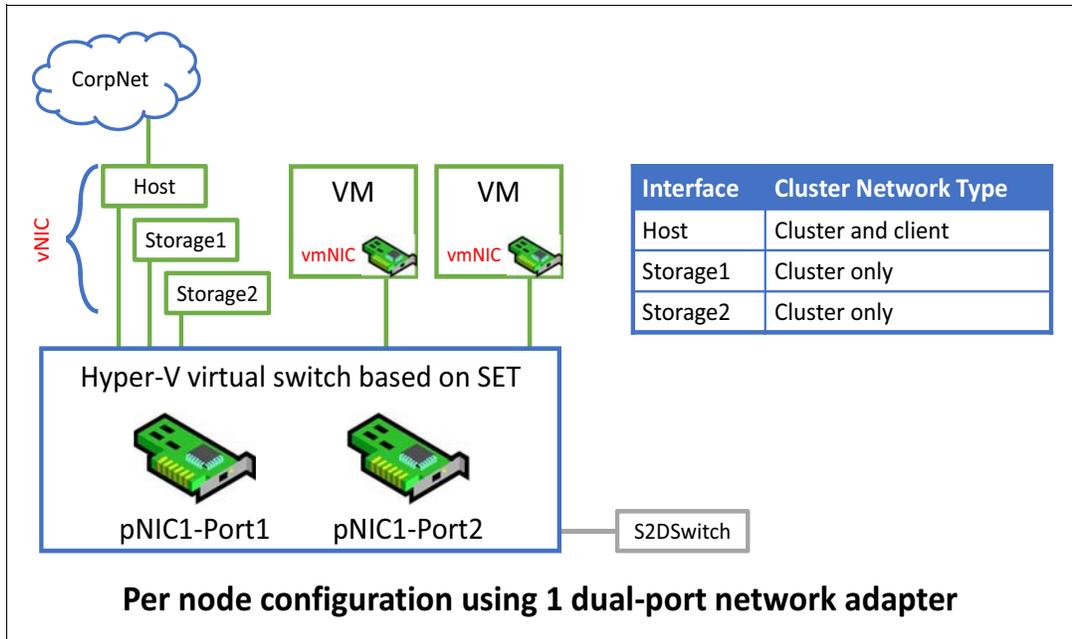


Figure 24 Diagram showing single SET team created from both ports on a single Mellanox NIC

We have already enabled the Data Center Bridging (DCB) feature in “Install Windows Server roles and features” on page 23. However, we do not want to use the DCB Exchange (DCBX) protocol to allow the OS to learn DCB settings from the switches, since the Windows operating system never looks at what settings the switch sent to the NIC. We configure the NIC to use specific settings, so it is safest to ensure that the NIC is told not to accept such settings from the network switch.

We now need to create a policy to establish network Quality of Service (QoS) to ensure that the Software Defined Storage system has enough bandwidth to communicate between the nodes (including cluster heartbeat), ensuring resiliency and performance. We also need to explicitly disable regular Flow Control (Global Pause) on the Mellanox adapters, since Priority Flow Control (PFC) and Global Pause cannot operate together on the same interface.

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 4 on page 37. For a solution that includes one dual-port Mellanox NIC in each server, three network interfaces should be displayed at this point, as shown in Figure 25.

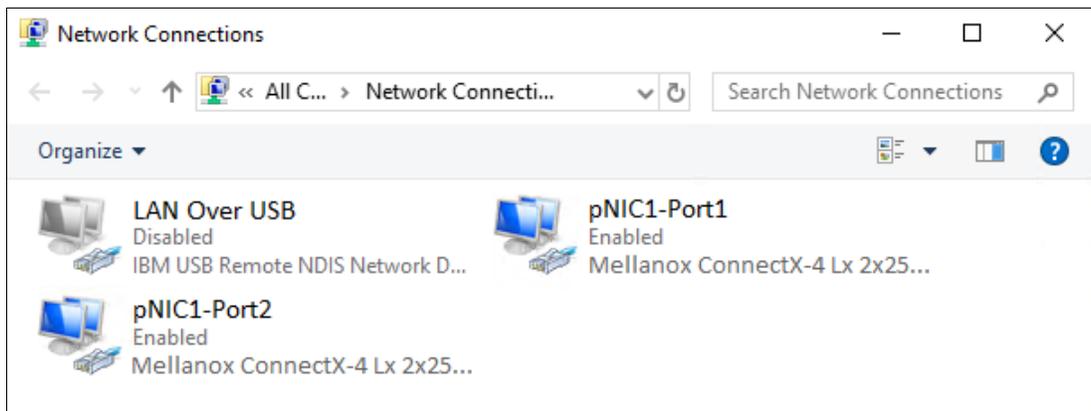


Figure 25 Network Connections control panel showing three interfaces that should exist at this point

As you can see, we have renamed the two network interfaces that we will use according to the tables above. We have also renamed the interface for the IBM USB Remote NDIS Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since LOM ports are not used in this scenario, they should be disabled in UEFI to avoid issues with cluster validation and creation later. If any LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

To make all these changes quickly and consistently on each of the servers that will become nodes in the Azure Stack HCI cluster, we use a PowerShell script. Example 12 on page 39 shows the script we used in our lab.

*Example 12 PowerShell script to configure required network parameters on servers*

---

```
# Block DCBX protocol between switches and nodes
Set-NetQosDcbxSetting -InterfaceAlias "pNIC1-Port1" -Willing $False
Set-NetQosDcbxSetting -InterfaceAlias "pNIC1-Port2" -Willing $False
# Configure QoS policies for SMB-Direct (RoCE), Cluster Heartbeat and Default (all other) traffic
New-NetQosPolicy -Name "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
New-NetQosPolicy -Name "Cluster-HB" -Cluster -PriorityValue8021Action 7
New-NetQosPolicy -Name "Default" -Default -PriorityValue8021Action 0
# Enable flow control for SMB-Direct (RoCE)
Enable-NetQosFlowControl -Priority 3
# Disable flow control for all other traffic
Disable-NetQosFlowControl -Priority 0,1,2,4,5,6,7
# Apply Quality of Service (QoS) policy to the target adapters
Enable-NetAdapterQos -Name "pNIC1-Port1"
Enable-NetAdapterQos -Name "pNIC1-Port2"
# Set minimum bandwidth - 50% for SMB-Direct, 1% for Cluster-HB
New-NetQosTrafficClass "SMB" -Priority 3 -BandwidthPercentage 50 -Algorithm ETS
New-NetQosTrafficClass "Cluster-HB" -Priority 7 -BandwidthPercentage 1 -Algorithm ETS
# Disable flow control (Global Pause) on physical adapters
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -RegistryKeyword "*FlowControl" -RegistryValue 0
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -RegistryKeyword "*FlowControl" -RegistryValue 0
```

---

For an Azure Stack HCI solution, we deploy a SET-enabled Hyper-V switch and add RDMA-enabled host virtual NICs to it for use by Hyper-V. Since many switches won't pass traffic class information on untagged VLAN traffic, we need to make sure that the vNICs using RDMA are on VLANs.

To keep this hyperconverged solution as simple as possible and since we are using dual-port 25GbE NICs, we will pass all traffic on a single VLAN. We use VLAN 12 in our examples, but the VLAN number can be determined by organizational policy. If you need to segment your network traffic more, for example to isolate virtual machine traffic, you can use additional VLANs.

As a best practice, we affinity the vNICs to the physical ports on the Mellanox ConnectX-4 network adapter. Without this step, both vNICs could become attached to the same physical NIC port, which would prevent bandwidth aggregation. It also makes sense to affinity the vNICs for troubleshooting purposes, since this makes it clear which port carries which vNIC traffic on all cluster nodes. Note that setting an affinity will not prevent failover to the other physical NIC port if the selected port encounters a failure. Affinity will be restored when the selected port is restored to operation.

Example 13 shows the PowerShell commands that can be used to perform the SET configuration, enable RDMA, assign VLANs to the vNICs, and affinitize the vNICs to the physical NIC ports.

*Example 13 PowerShell script to create a SET-enabled vSwitch and affinitize vNICs to physical NIC ports*

---

```
# Create SET-enabled vSwitch supporting multiple uplinks provided by Mellanox adapter
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "pNIC1-Port1", "pNIC1-Port2" -EnableEmbeddedTeaming $true `
  -AllowManagementOS $false
# Add host vNICs to the vSwitch just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Storage1" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Storage2" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Host" -ManagementOS
# Enable RDMA on Storage vNICs just created, but not on Host vNIC
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage1)"
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage2)"
# Assign Storage vNIC traffic to vLAN(s)
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage1" -VlanId 12 -Access -ManagementOS
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage2" -VlanId 12 -Access -ManagementOS
# Wait 5 seconds for previous commands to complete before proceeding
Start-Sleep -Seconds 5
# Affinitize vNICs to pNICs for consistency and better fault tolerance
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage1" -PhysicalNetAdapterName `
  "pNIC1-Port1" -ManagementOS
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage2" -PhysicalNetAdapterName `
  "pNIC1-Port2" -ManagementOS
```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

1. Configure a static IP address on the Storage1 vNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the Storage2 vNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. Perform a ping command from each interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both interfaces on each system should be able to communicate with both interfaces on all other systems.

Of course, PowerShell can be used to make IP address assignments if desired. Example 14 shows the commands used to specify static IP addresses and DNS server assignment for the interfaces on Node 1 in our environment. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 14 PowerShell commands used to configure IP settings on vNIC interfaces*

---

```
# Configure IP and subnet mask, no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
  -DefaultGateway 10.10.10.1
# Optional - Disable IPv6 on all interfaces if not needed
Get-NetAdapter | ? name -Like vEthernet* | Disable-NetAdapterBinding -ComponentID ms_tcpip6
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage1)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -ServerAddresses `
  ("10.10.10.5","10.10.10.6")
```

```
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage2)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
```

Figure 26 shows the network interfaces now configured in the server. Since the only interfaces that will be used in this solution are the interfaces derived from the physical Mellanox NIC ports, these are the only enabled interfaces that should be displayed.

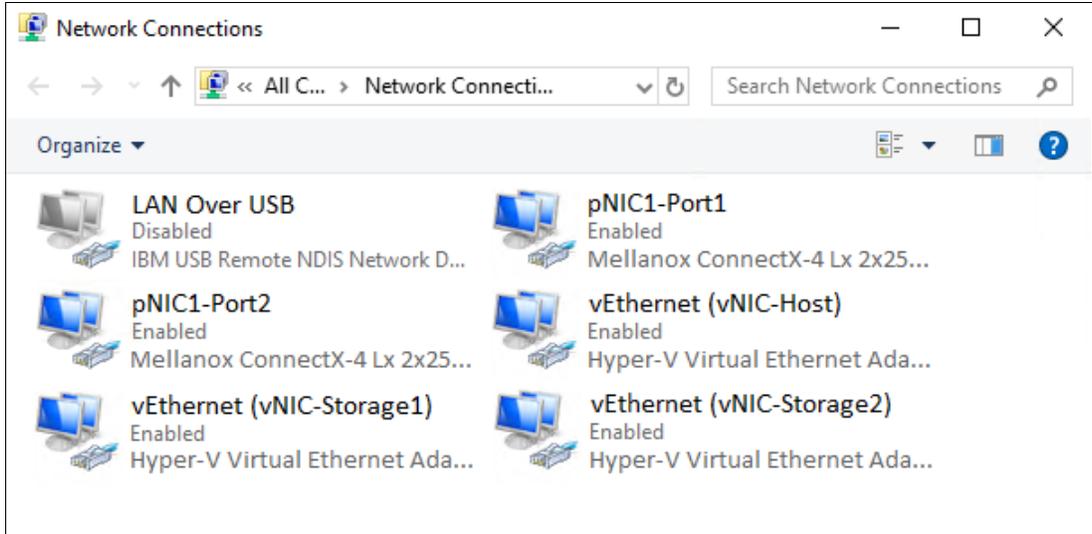


Figure 26 Final network interfaces from one dual-port Mellanox NIC

Execute the commands shown in Example 12 on page 39, Example 13 on page 40 and Example 14 on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

Since RDMA is so critical to the performance of the final solution, it is worthwhile to ensure that each piece of the configuration is correct as we move through the steps. We can't look for RDMA traffic yet, but we can verify that the vNICs (in a hyperconverged solution) have RDMA enabled. Example 15 shows the PowerShell command we use for this purpose.

Example 15 PowerShell command to verify that RDMA is enabled on the Storage interfaces

```
Get-NetAdapterRdma | ? Name -Like *Storage* | Format-Table Name, Enabled
```

Figure 27 shows the output of the above command in our environment.

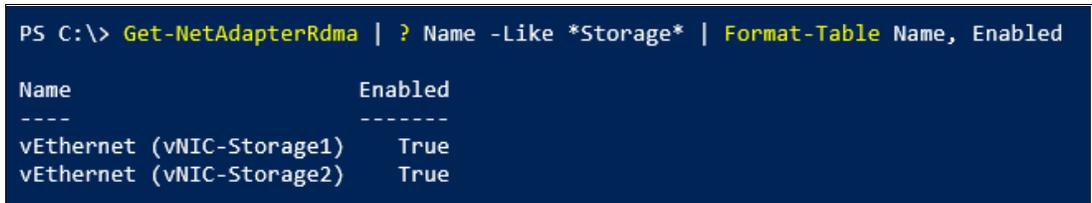


Figure 27 PowerShell command verifies that RDMA is enabled on the Storage interfaces

The next piece of preparing the infrastructure for Azure Stack HCI is to perform a few optimizations to ensure the best performance possible. Proceed to "Create failover cluster" on page 84 for detailed instructions.

## Two dual-port Mellanox adapters in each server

If using two dual-port NICs, we create two SET teams and Hyper-V switches; one across Port 1 on both NICs and another across Port 2 on both NICs. Figure 28 shows various details of these SET teams and how they are used. In this case, storage traffic can be isolated to one of the teams, while all other traffic, including VM Live Migration and all traffic in and out of the cluster, is carried over the other team. For best redundancy, assure that one port from each NIC is added to each SET team.

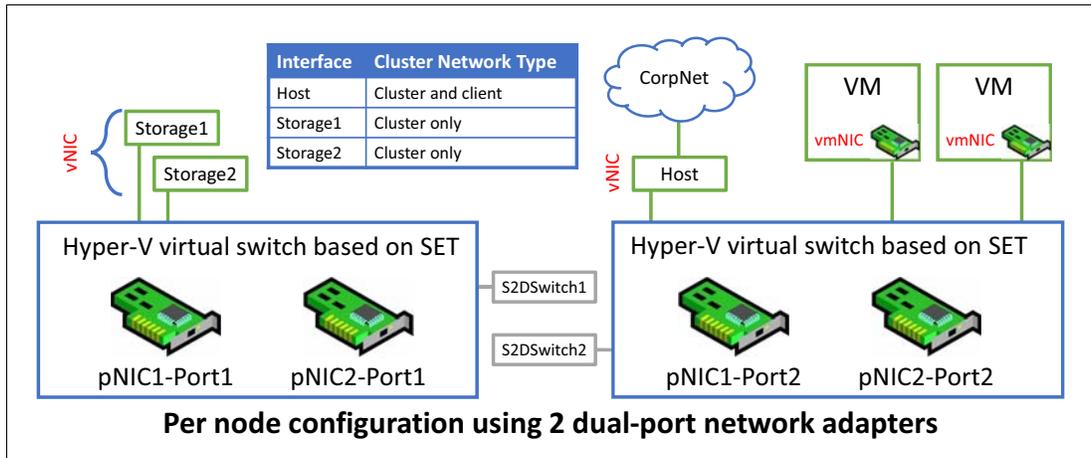


Figure 28 Diagram showing two SET teams created from two dual-port network adapters

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 4 on page 37. For a solution that includes two dual-port Mellanox NICs in each server, five network interfaces should be displayed at this point, as shown in Figure 29.

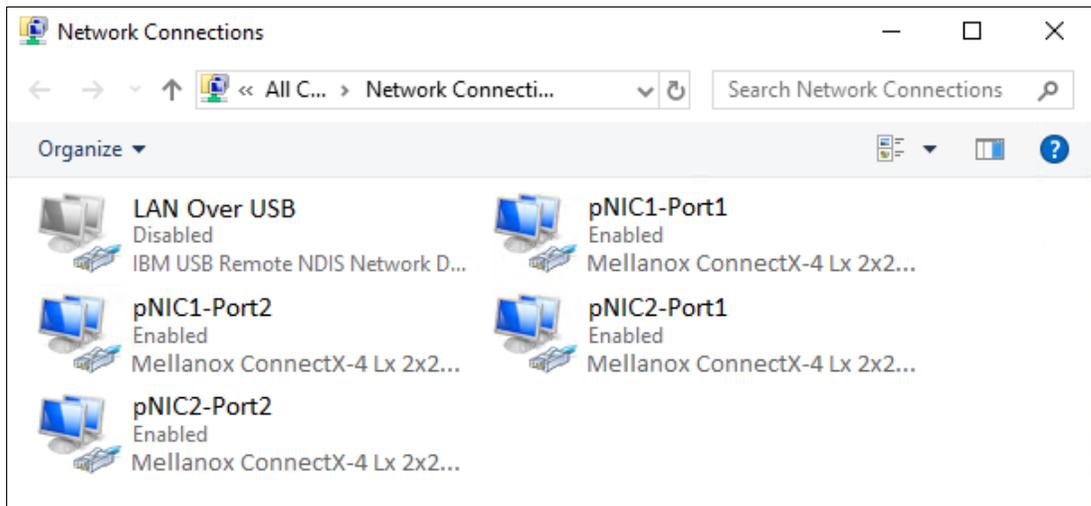


Figure 29 Network Connections control panel showing the five interfaces that should exist at this point

As you can see, we have renamed the four network interfaces that we will use according to the table. We have also renamed the interface for the IBM USB Remote NDIS Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since LOM ports are not used in this scenario, they should be disabled in UEFI to avoid issues with cluster validation and creation later. If any LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

The process and commands used to configure two dual-port Mellanox network adapters (4 physical network ports total) are nearly identical to those shown in the previous section. In this section we show only the required commands and a few notes. For more detail about exactly what is being configured and why, refer to the previous section.

To create the QoS policy and disable PFC, use the commands shown in Example 16. These commands are nearly identical to those shown in the previous section for a single dual-port network adapter. The only difference is that four NIC ports need to be configured rather than two. The adapter names indicate the physical ports to which they refer. For example, “pNIC1-Port1” refers to Port 1 on Mellanox NIC 1, while “pNIC2-Port1” refers to Port 1 on Mellanox NIC 2.

*Example 16 PowerShell script to configure required network parameters on servers*

---

```
# Block DCBX protocol between switches and nodes
Set-NetQosDcbxSetting -InterfaceAlias "pNIC1-Port1" -Willing $False
Set-NetQosDcbxSetting -InterfaceAlias "pNIC1-Port2" -Willing $False
Set-NetQosDcbxSetting -InterfaceAlias "pNIC2-Port1" -Willing $False
Set-NetQosDcbxSetting -InterfaceAlias "pNIC2-Port2" -Willing $False
# Configure QoS policies for SMB-Direct (RoCE), Cluster Heartbeat and Default (all other) traffic
New-NetQosPolicy -Name "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
New-NetQosPolicy -Name "Cluster-HB" -Cluster -PriorityValue8021Action 7
New-NetQosPolicy -Name "Default" -Default -PriorityValue8021Action 0
# Enable flow control for SMB-Direct (RoCE)
Enable-NetQosFlowControl -Priority 3
# Disable flow control for all other traffic
Disable-NetQosFlowControl -Priority 0,1,2,4,5,6,7
# Apply QoS policies to target NIC ports
Enable-NetAdapterQos -Name "pNIC1-Port1"
Enable-NetAdapterQos -Name "pNIC1-Port2"
Enable-NetAdapterQos -Name "pNIC2-Port1"
Enable-NetAdapterQos -Name "pNIC2-Port2"
# Set minimum bandwidth - 50% for SMB-Direct, 1% for Cluster-HB
New-NetQosTrafficClass "SMB" -Priority 3 -BandwidthPercentage 50 -Algorithm ETS
New-NetQosTrafficClass "Cluster-HB" -Priority 7 -BandwidthPercentage 1 -Algorithm ETS
# Disable flow control (Global Pause) on physical adapters
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -RegistryKeyword "*FlowControl" -RegistryValue 0
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -RegistryKeyword "*FlowControl" -RegistryValue 0
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port1" -RegistryKeyword "*FlowControl" -RegistryValue 0
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port2" -RegistryKeyword "*FlowControl" -RegistryValue 0
```

---

Example 17 shows PowerShell commands used to perform the SET configuration, enable RDMA, assign VLANs to the vNICs, and affinitize the vNICs to the physical NIC ports.

*Example 17 PowerShell script to create a SET-enabled vSwitch and affinitize vNICs to physical NIC ports*

---

```
# Create SET-enabled vSwitches supporting multiple uplinks provided by Mellanox NICs
New-VMSwitch -Name "S2DSwitch1" -NetAdapterName "pNIC1-Port1", "pNIC2-Port1" -EnableEmbeddedTeaming $true -AllowManagementOS $false
New-VMSwitch -Name "S2DSwitch2" -NetAdapterName "pNIC1-Port2", "pNIC2-Port2" -EnableEmbeddedTeaming $true -AllowManagementOS $false
# Add host vNICs to the vSwitches just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch1" -Name "vNIC-Storage1" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch1" -Name "vNIC-Storage2" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch2" -Name "vNIC-Host" -ManagementOS
```

```

# Enable RDMA on Storage vNICs just created, but not on Host vNIC
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage1)"
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage2)"
# Assign vNIC traffic to vLAN(s)
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage1" -VlanId 12 -Access -ManagementOS
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage2" -VlanId 12 -Access -ManagementOS
# Affinitize vNICs to pNICs for consistency and better fault tolerance
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage1" -PhysicalNetAdapterName `
    "pNIC1-Port1" -ManagementOS
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage2" -PhysicalNetAdapterName `
    "pNIC2-Port1" -ManagementOS

```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

1. Configure a static IP address on the Storage1 vNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the Storage2 vNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. Configure a static IP address on the Host vNIC, using a different subnet if desired. Since this interface will carry network traffic into and out of the Azure Stack HCI cluster (North-South traffic), this will likely be a “CorpNet” subnet. You must specify a DNS server and register this interface with DNS. You must also configure a default gateway for this interface.
4. Perform a ping command from each Storage interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both Storage interfaces on each system should be able to communicate with both Storage interfaces on all other systems and the Host interface on each system should be able to communicate with the Host interface on all other systems.
5. Example 18 shows the commands used to specify static IP addresses and DNS server assignment for each interface on Node 1 in our environment. These are exactly the same commands that are used if only one dual-port Mellanox network adapter is installed in each server. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 18 PowerShell commands used to configure IP settings on vNIC interfaces*

---

```

# Optional: Disable IPv6 on all vNICs if it is not needed
Get-NetAdapter | ? name -Like vEthernet* | Disable-NetAdapterBinding -ComponentID ms_tcpip6
# Configure IP and subnet mask, but no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
    -DefaultGateway 10.10.10.1
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage1)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -ServerAddresses `
    ("10.10.10.5","10.10.10.6")
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage2)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -ServerAddresses `
    ("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
    ("10.10.10.5","10.10.10.6")

```

---

Figure 30 shows the network interfaces now configured in the server. Since the only interfaces that will be used in this solution are the interfaces derived from the physical Mellanox NIC ports, these are the only enabled interfaces that should be displayed.

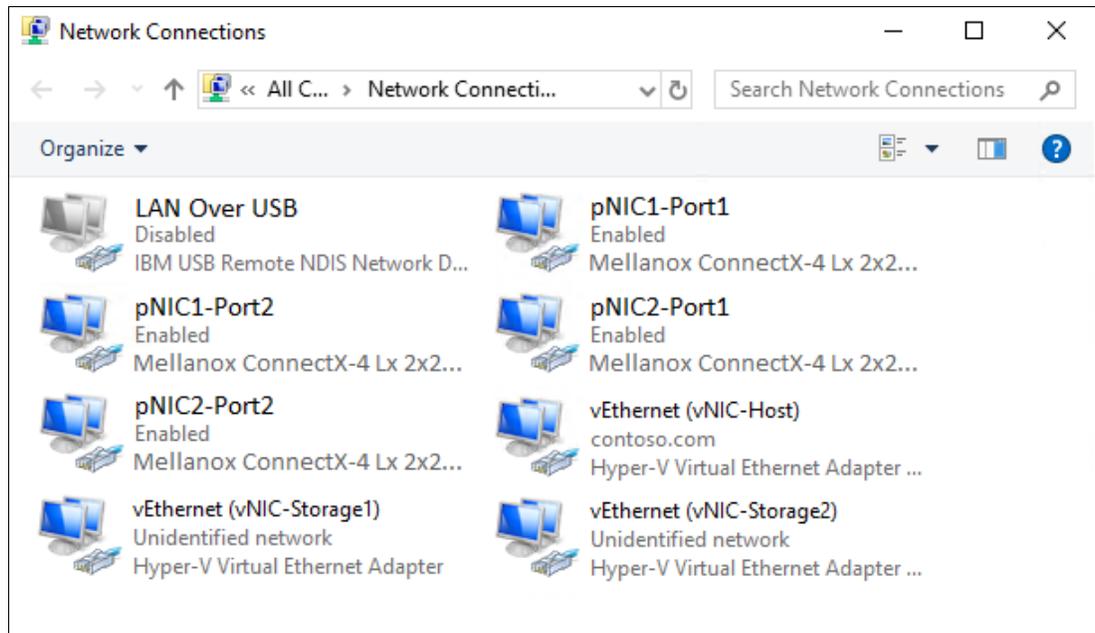


Figure 30 Final network interfaces from two dual-port Mellanox NICs

Execute the commands shown in Example 16 on page 43, Example 17 on page 43 and Example 18 on page 44 on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

Example 19 shows the PowerShell command we use to confirm that RDMA is enabled on the appropriate interfaces.

*Example 19 PowerShell command to verify that RDMA is enabled on the Storage interfaces*

```
Get-NetAdapterRdma | ? Name -Like *Storage* | Format-Table Name, Enabled
```

Figure 31 shows the output of the above command in our environment.

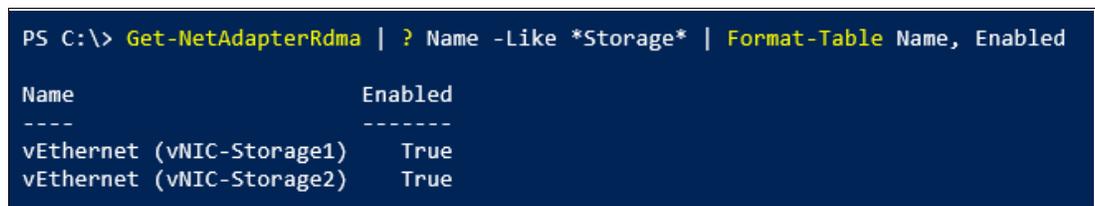


Figure 31 PowerShell command verifies that RDMA is enabled on the Storage interfaces

The next piece of preparing the infrastructure for Azure Stack HCI is to perform a few optimizations to ensure the best performance possible. Proceed to “Create failover cluster” on page 84 for detailed instructions.

## RoCE: 2-4 nodes, direct-connected

This deployment scenario provides the steps to configure an Azure Stack HCI cluster that contains 2 to 4 nodes that are direct-connected for East-West storage traffic, and uses the RoCE implementation of RDMA. Figure 32 shows a portion of the process flow diagram for this document and where this scenario fits. Although the diagram still refers to “2 or 3 Nodes” since detailed instructions are provided for 2- and 3-node clusters, a 4-node direct-connected cluster can be deployed by extrapolating these instructions.

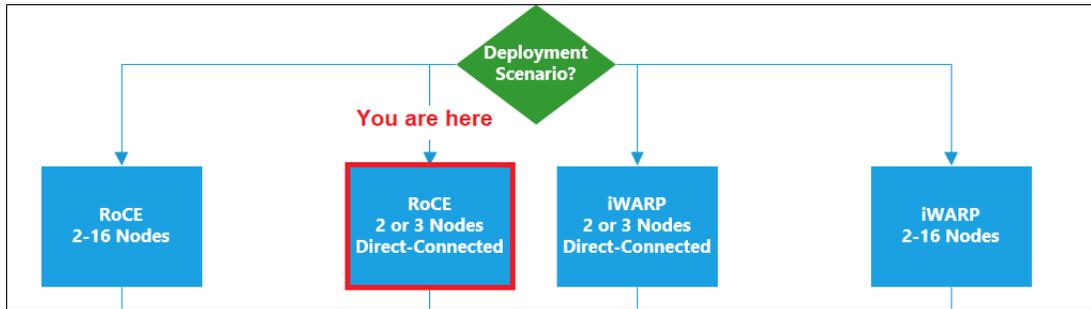


Figure 32 Portion of this document's process flow showing the direct-connected RoCE scenario

**Note:** Although these instructions can be used to deploy a 2-node direct-connected cluster of ThinkAgile MX1021 on SE350 Certified Nodes, we have published a separate document dedicated to this solution. Please refer to the *ThinkAgile MX1021 on SE350 Azure Stack HCI (S2D) Deployment Guide*, which can be found at the following URL:

<https://lenovopress.com/lp1298>

### Overview

By “direct-connected” we refer to the dual-port Mellanox NICs in the nodes being connected directly to each other, without a switch between them, for storage traffic. Note that a switch is still needed to connect the nodes to the corporate network (“CorpNet”) to pass North-South traffic into and out of the cluster.

Figure 33 on page 47 shows diagrams of various network connectivity models between cluster nodes for storage traffic. Microsoft does not support bridged connectivity between cluster nodes and does not recommend single-link connectivity. The only recommended approach is to provide full mesh dual-link connectivity between all nodes for East-West storage traffic. The best way to provide two network connections between each of the nodes in a cluster without using a switch between them is by using “N-1” dual-port Mellanox network adapters in each node, where N is the number of cluster nodes. This means that for a 3-node cluster, each node would require 2 dual-port network adapters for storage traffic and for a 4-node cluster, each node would require 3 dual-port network adapters.

Although the benefits of switchless deployments diminish with clusters larger than three-nodes due to the number of network adapters required, we have added details in this section that are specific to 4-node direct-connected clusters due to customer demand. We will not carry the discussion past four nodes since the resulting configurations become quite complicated. Make sure to understand what is required and supported by Microsoft for direct-connected Azure Stack HCI clusters. An excellent source of this information is an article titled *Physical network requirements for Azure Stack HCI* at the following URL:

<https://learn.microsoft.com/en-us/azure-stack/hci/concepts/physical-network-requirements?tabs=overview%2C22H2reqs>

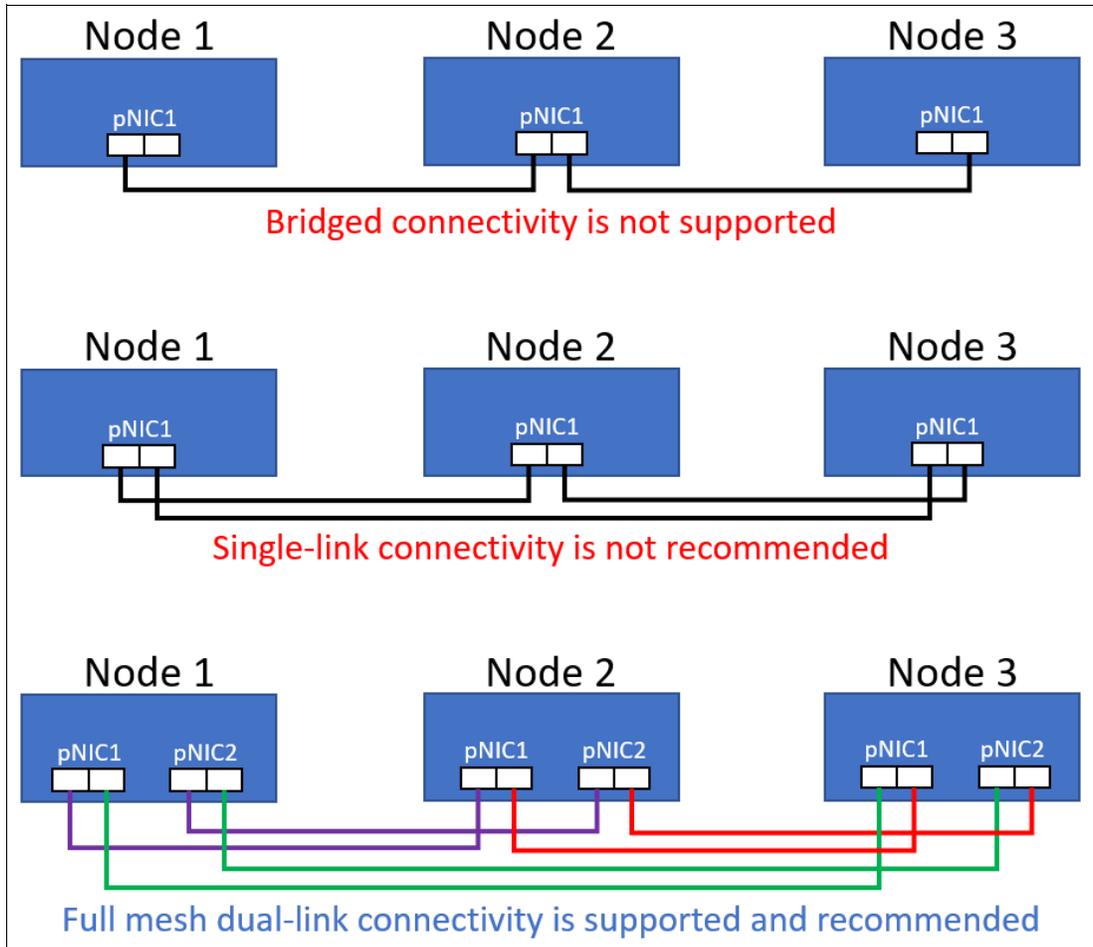


Figure 33 Various node-to-node network connectivity models

### Connect servers to each other

For a 2-node cluster, each of the ports in the Mellanox NIC is connected directly to the same port on the same NIC in the other node to carry East-West storage traffic. For North-South management traffic, we connect two additional network ports on each node to the corporate network. For the ThinkSystem SR650 rack server used in our examples, it is convenient to use two LOM ports for this purpose. Figure 34 on page 48 shows the network cable diagram for a 2-node direct-connected cluster using LOM ports for management traffic.

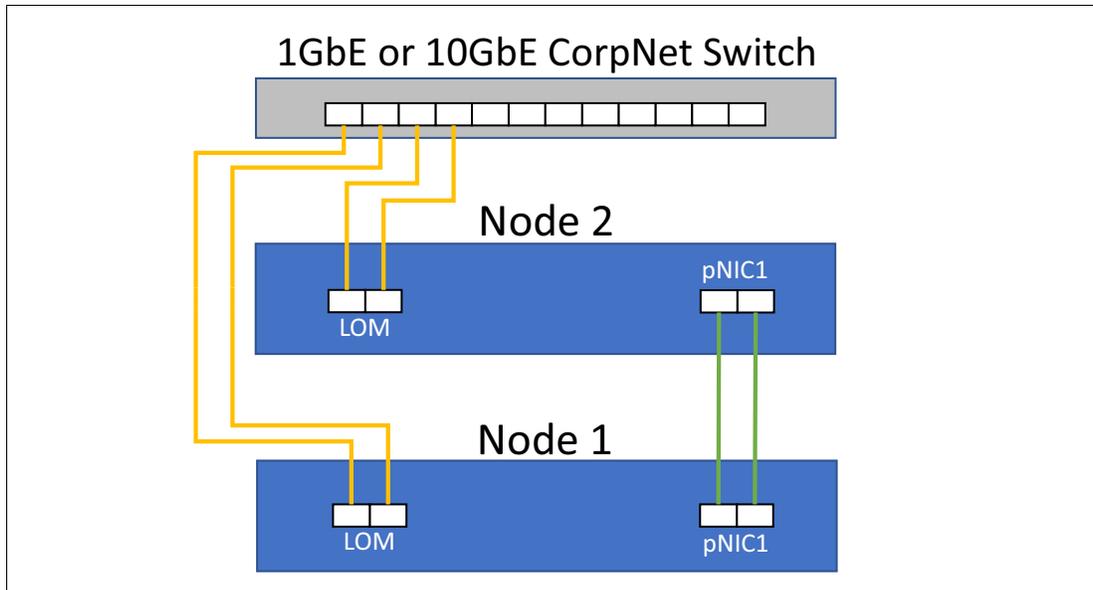


Figure 34 Two-node connectivity for direct-connected deployment scenario

For a 3-node cluster, the ports in the Mellanox NICs are connected directly to the other nodes in a full mesh dual-link configuration. That is, each node is connected to each of the other two nodes in a redundant manner, which requires a total of four network ports (two dual-port Intel E810 NICs) in each node for East-West Storage traffic. In addition, like the 2-node configuration above, we connect two LOM/OCF ports on each node to the corporate network to carry North-South Management/Compute traffic. Figure 58 shows the network cable diagram for a 3-node direct-connected cluster. Connection line colors in the figure represent connections between two nodes (for example between Node 1 and Node 2) for East-West traffic or between the nodes and the corporate network for North-South traffic.

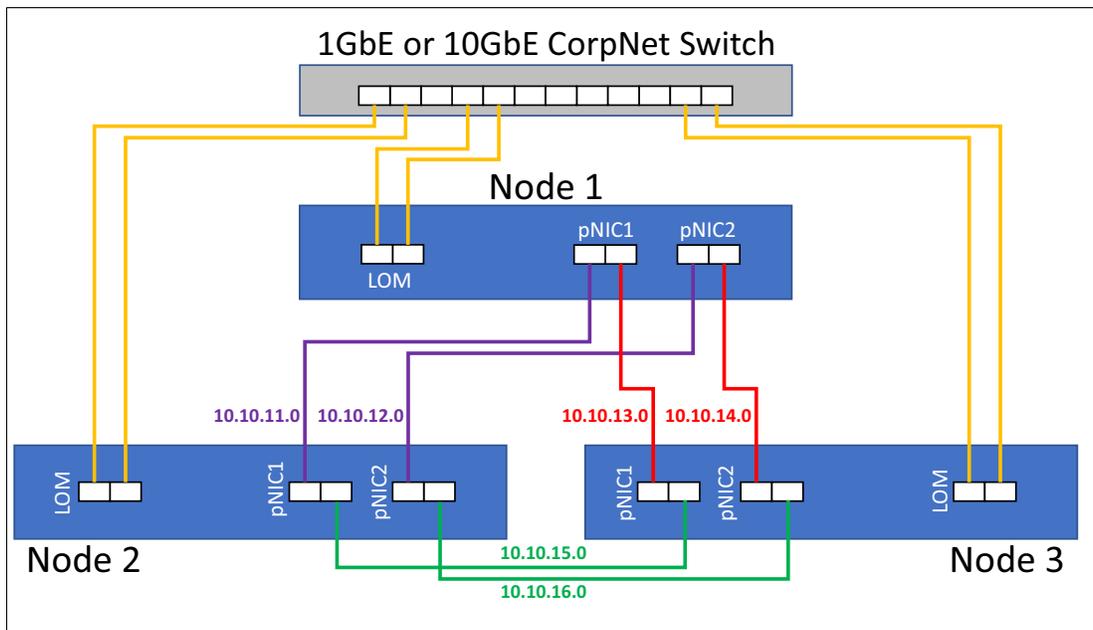


Figure 35 Three-node connectivity for direct-connected deployment scenario

For a 4-node cluster, the ports in the Mellanox NICs are connected directly to the other nodes in a full mesh dual-link configuration. That is, each node is connected to each of the other

three nodes in a redundant manner, which requires a total of six network ports (three dual-port Mellanox NICs) in each node. In addition, like the 2- and 3-node configurations above, we connect two LOM/OCP ports on each node to the corporate network. Figure 36 shows the network cable diagram for a 4-node direct-connected cluster. We have removed the connection lines to the switches for North-South traffic for clarity. Connection line colors in the figure represent connections between two nodes (for example between Node 1 and Node 2) for East-West traffic.

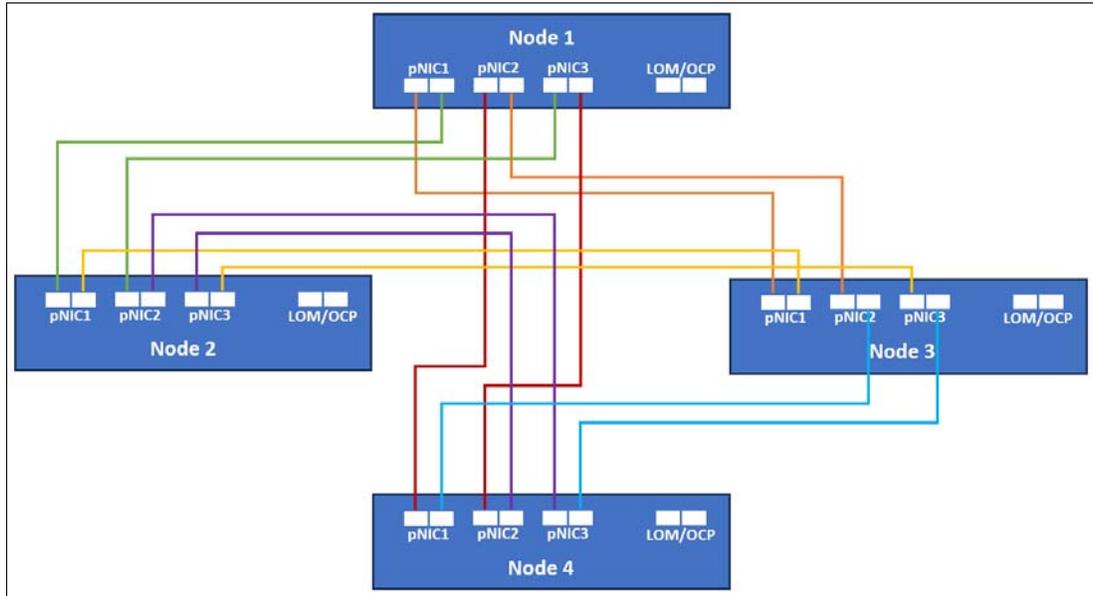


Figure 36 Four-node connectivity for direct-connected deployment scenario

Removing the dual-port network adapter boxes from the diagram above results in further clarity, which is shown in Figure 37. This simplified diagram shows the network cabling for storage traffic between all nodes and includes an example of the subnets used for Storage traffic. Make sure to use ports from *different* network adapters to connect any two nodes. This will ensure that if the network adapter fails, connectivity between the two nodes will survive.

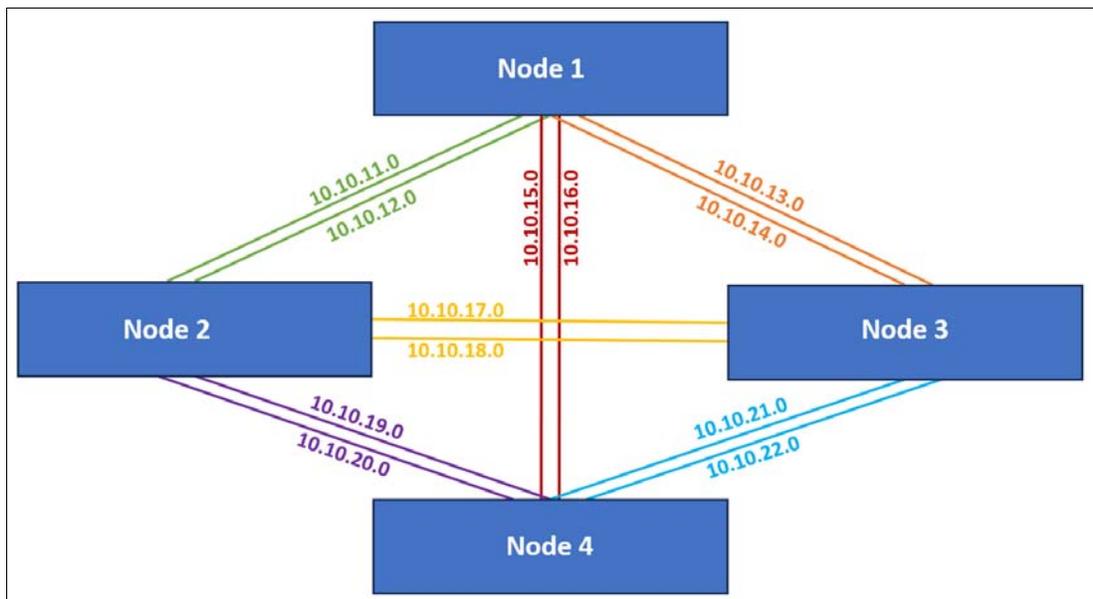


Figure 37 Simplified four-node connectivity for direct-connected scenario with example subnets shown

**Note:** This document provides detailed instructions and PowerShell scripts for both 2-node and 3-node direct-connected Azure Stack HCI cluster deployment. These details can be extrapolated to deploy a 4-node direct-connected cluster.

In all cases of direct-connected deployment, it is required to establish a subnet for each of the high-speed connections between the nodes for storage traffic. That is, a subnet for each of the network cables that are connected between the nodes. For a 3-node direct-connected cluster, this means a total of six subnets and for a 4-node direct-connected cluster, a total of twelve subnets are required. The subnet we use for each connection is shown next to the connection lines in Figure 35 on page 48 for three nodes and Figure 37 on page 49 for four nodes.

Table 5 shows the high-speed direct network connections between each of the nodes in a 3-node cluster, as well as the subnet that carries the traffic for each of these connections. The subnets shown are consistent with examples in this document. If you prefer to use your own subnets, make sure to modify the example PowerShell commands accordingly.

*Table 5 Source and destination ports for full-mesh 3-node direct-connected HCI cluster*

Source Device	Source Port	Destination Device	Destination Port	Subnet
Node 1	pNIC1-Port1	Node 2	pNIC1-Port1	10.10.11.0/24
Node 1	pNIC1-Port2	Node 3	pNIC1-Port1	10.10.12.0/24
Node 1	pNIC2-Port1	Node 2	pNIC2-Port1	10.10.13.0/24
Node 1	pNIC2-Port2	Node 3	pNIC2-Port1	10.10.14.0/24
Node 2	pNIC1-Port1	Node 1	pNIC1-Port1	10.10.11.0/24
Node 2	pNIC1-Port2	Node 3	pNIC1-Port2	10.10.15.0/24
Node 2	pNIC2-Port1	Node 1	pNIC2-Port1	10.10.13.0/24
Node 2	pNIC2-Port2	Node 3	pNIC2-Port2	10.10.16.0/24
Node 3	pNIC1-Port1	Node 1	pNIC1-Port2	10.10.12.0/24
Node 3	pNIC1-Port2	Node 2	pNIC1-Port2	10.10.15.0/24
Node 3	pNIC2-Port1	Node 1	pNIC2-Port2	10.10.14.0/24
Node 3	pNIC2-Port2	Node 2	pNIC2-Port2	10.10.16.0/24

If configuring a 3-node cluster, make sure to modify the PowerShell scripts shown in Example 20 on page 53 to ensure that all ports on both Mellanox NICs are configured properly. This will require adding lines to configure the second NIC ports. In addition, you will need to modify the PowerShell scripts shown in Example 22 on page 54 to ensure that proper IP addressing is used to establish all six subnets. In this case, the PowerShell commands that are run on one node are not exactly the same as the commands run on the other two nodes. Use the Subnet column in Table 5 to modify these commands.

If configuring a 4-node cluster, the same applies, but additional PowerShell commands will need to be added to configure the additional network ports and subnets required to support four nodes.

With all the physical network connections made, we move to configuring the network interfaces on the servers.

## Configure networking parameters

For the RoCE two-node direct-connect scenario, we use the server’s LOM/OCP ports to carry “CorpNet” traffic into and out of the cluster (i.e. North-South traffic). To increase performance and availability, we need to leverage the virtual network capabilities of Hyper-V on each host by creating a SET-enabled team from the LOM/OCP ports. For a brief discussion of the differences between LOM and OCP network ports see “LOM and OCP network ports” on page 18.

Also, for all RoCE direct-connect deployment scenarios, we do not create a SET-enabled team from the Mellanox NIC ports. In this deployment scenario, the storage traffic is carried by the physical network adapter ports (pNICs). Figure 38 shows a diagram representing this difference in a two-node direct-connect scenario.

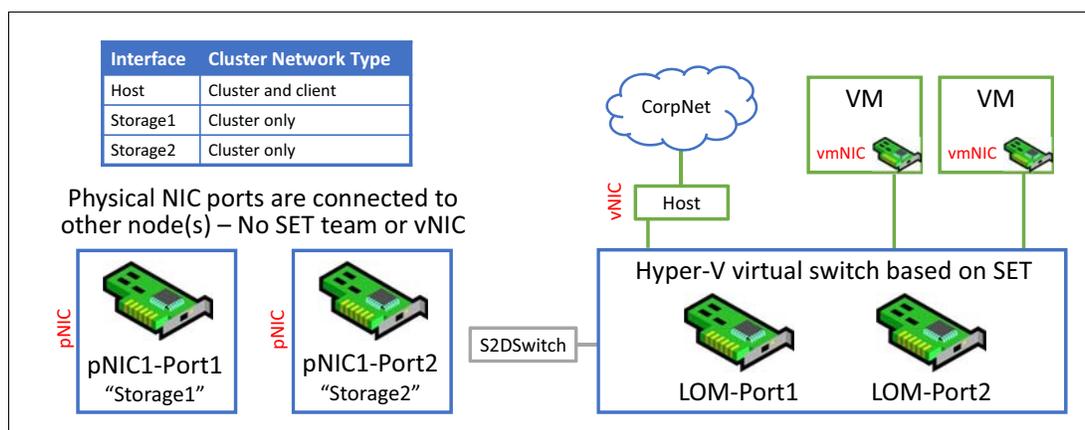


Figure 38 Two-node direct-connect network interfaces

We make extensive use of PowerShell commands and scripts throughout this document to configure various aspects of the Azure Stack HCI environment. The commands and scripts used to configure networking parameters on the servers in this section can be used with minimal modification if you take a moment now to name the physical network adapter ports according to Table 6 before working through this section. Alternatively, you can use your own naming convention for these ports, but in this case, remember to modify the PowerShell commands appropriately.

Table 6 Friendly names of network adapter ports used in this scenario

	Mellanox ConnectX-4	PCI Slot
First NIC, first port	"pNIC1-Port1"	6
First NIC, second port	"pNIC1-Port2"	6
Second NIC, first port (if used)	"pNIC2-Port1"	4
Second NIC, second port (if used)	"pNIC2-Port2"	4

For this direct-connected scenario, the LOM/OCP ports are used for North-South traffic, which includes VM traffic. Naming of these ports is shown in Table 7.

Table 7 Friendly names of LOM/OCP ports used in this scenario

	LOM/OCP
First Port	"LOM-Port1"
Second port	"LOM-Port2"

	LOM/OCP
Third port (if used)	“LOM-Port3”
Fourth port (if used)	“LOM-Port4”

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 6 and Table 7. For a solution that includes one dual-port Mellanox NIC in each server and uses 2 LOM/OCP ports, five network interfaces should be displayed at this point, as shown in Figure 39 on page 52.

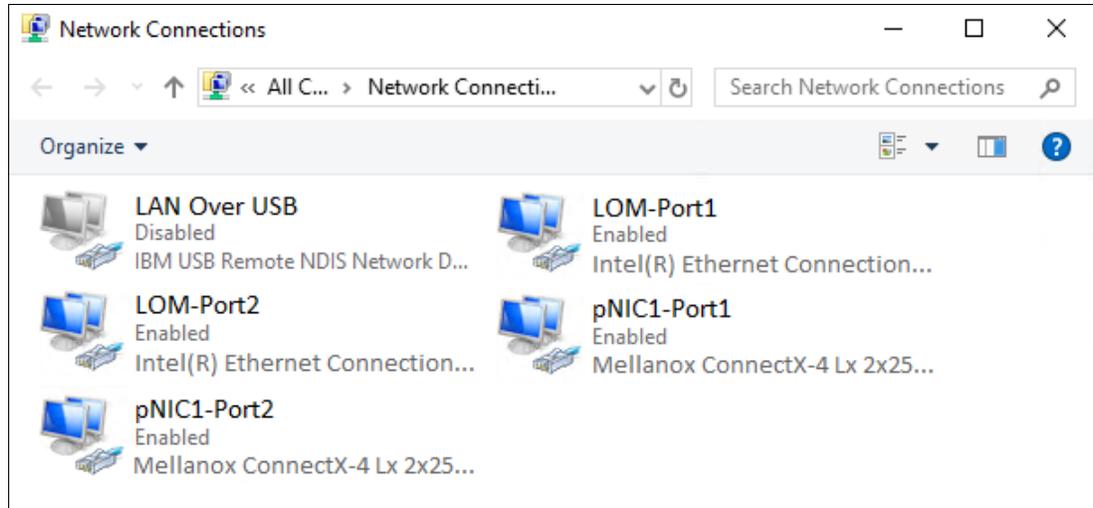


Figure 39 Network Connections control panel showing the four interfaces that should exist at this point

As you can see, we have renamed the four network interfaces that we will use according to the tables above. We have also renamed the interface for the IBM USB Remote NDIS Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since only the first two LOM/OCP ports are used in this scenario, additional LOM ports should be disabled in UEFI, if present, to avoid issues with cluster validation and creation later. If any unneeded LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

We have already enabled the Data Center Bridging (DCB) feature in “Install Windows Server roles and features” on page 23. Although DCB is not used for 2 or 3-node clusters, we enabled this feature anyway for future expansion. It will be much easier to add a fourth node if DCB is already enabled on the first 3 nodes. However, once a fourth node is added and DCB is used, we do not want to use the DCB Exchange (DCBX) protocol to allow the OS to learn DCB settings from the switches, since the Windows operating system never looks at what settings the switch sent to the NIC. We configure the NIC to use specific settings, so it is safest to ensure that the NIC is told not to accept such settings from the network switch.

We need to create a policy to establish network Quality of Service (QoS) to ensure that the Software Defined Storage system has enough bandwidth to communicate between the nodes (including cluster heartbeat), ensuring resiliency and performance. We also need to disable regular Flow Control (Global Pause) on the Mellanox adapters, since Priority Flow Control (PFC) and Global Pause cannot operate together on the same interface.

To make all these changes quickly and consistently on each of the servers that will become nodes in the Azure Stack HCI cluster, we again use a PowerShell script. Example 20 shows the script we used in our lab.

*Example 20 PowerShell script to configure required network parameters on servers*

---

```
# Block DCBX protocol between switches and nodes (for future node expansion)
Set-NetQoSdcbxSetting -InterfaceAlias "pNIC1-Port1" -Willing $False
Set-NetQoSdcbxSetting -InterfaceAlias "pNIC1-Port2" -Willing $False
# Configure QoS policies for SMB-Direct (RoCE), Cluster Heartbeat and Default (all other) traffic
New-NetQoSPolicy -Name "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3
New-NetQoSPolicy -Name "Cluster-HB" -Cluster -PriorityValue8021Action 7
New-NetQoSPolicy -Name "Default" -Default -PriorityValue8021Action 0
# Enable flow control for SMB-Direct (RoCE)
Enable-NetQoSFlowControl -Priority 3
# Disable flow control for all other traffic
Disable-NetQoSFlowControl -Priority 0,1,2,4,5,6,7
# Apply QoS policy to Mellanox pNIC ports
Enable-NetAdapterQos -Name "pNIC1-Port1"
Enable-NetAdapterQos -Name "pNIC1-Port2"
# Set minimum bandwidth - 50% for SMB-Direct, 1% for Cluster-HB
New-NetQoSTrafficClass "SMB" -Priority 3 -BandwidthPercentage 50 -Algorithm ETS
New-NetQoSTrafficClass "Cluster-HB" -Priority 7 -BandwidthPercentage 1 -Algorithm ETS
# Disable flow control on physical adapters
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -RegistryKeyword "*FlowControl" -RegistryValue 0
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -RegistryKeyword "*FlowControl" -RegistryValue 0
```

---

For the direct-connected scenario, we do not create a SET team from the Mellanox NIC ports, but we do so for the LOM/OCP ports that carry North-South traffic. From the SET-enabled team we create from the LOM/OCP ports, a virtual switch (“S2DSwitch” in Figure 38 on page 51) is defined and a logical network adapter (vNIC) is created for use by virtual machines in the hyperconverged solution. Note that for the converged solution, the SET team, vSwitch, and vNIC do not need to be created. However, we generally do this anyway, just in case we’d like to run a VM or two from the storage cluster occasionally.

Example 21 shows the PowerShell commands that can be used to perform the SET team configuration and enable RDMA on the physical Mellanox NIC ports. In this scenario, the SET team is created from the LOM/OCP ports to enable the Hyper-V switch for virtual machine use. If the servers have 4-port LOMs, all 4 ports can be used for this purpose. Make sure to run the appropriate command to create the SET team (one of the first two commands in the example, but not both). Alternatively, if 4-port LOMs are present, but you only want to use two ports, you should disable Ports 3 and 4 in System UEFI before proceeding. This will help to avoid complications when creating the cluster that may occur when unused network adapters are visible to the OS.

In addition, the commands shown add a vNIC to the vSwitch, enable RDMA on the physical Mellanox NIC ports for storage traffic, and disable RDMA on the physical LOM/OCP ports, since storage traffic should not traverse these ports.

*Example 21 PowerShell script to create a SET-enabled vSwitch and affinitize vNICs to physical LOM/OCP ports*

---

```
# Create SET-enabled vSwitch for Hyper-V using 2 LOM/OCP ports
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "LOM-Port1", "LOM-Port2" -EnableEmbeddedTeaming $true -AllowManagementOS $false

# Note: Run the next command only if using 4 LOM ports
# Create SET-enabled vSwitch for Hyper-V using 4 LOM ports
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "LOM-Port1", "LOM-Port2", "LOM-Port3", "LOM-Port4" -EnableEmbeddedTeaming $true -AllowManagementOS $false
```

```
# Add host vNIC to the vSwitch just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Host" -ManagementOS
# Enable RDMA on Mellanox pNIC ports
Enable-NetAdapterRDMA -Name "pNIC1-Port1"
Enable-NetAdapterRDMA -Name "pNIC1-Port2"
# Disable RDMA on LOM/OCF pNIC ports
Disable-NetAdapterRDMA -Name "LOM-Port1"
Disable-NetAdapterRDMA -Name "LOM-Port2"
```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

1. Configure a static IP address on the NIC1-Port1 pNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the NIC1-Port2 pNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. If configuring a 3-node cluster, make sure to modify the PowerShell scripts shown in Example 22 on page 54 to ensure that proper IP addressing is used to establish all six subnets. In this case, the PowerShell commands that are run on one node are not exactly the same as the commands run on the other two nodes. Use the Subnet column in Table 5 on page 50 to modify these commands.
4. Configure a static IP address on the Host vNIC, using a different subnet if desired. Since this interface will carry network traffic into and out of the Azure Stack HCI cluster (North-South traffic), this will likely be a “CorpNet” subnet. You must specify a DNS server and register this interface with DNS. You must also configure a default gateway for this interface.
5. Perform a ping command from each Storage interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both Storage interfaces on each system should be able to communicate with both Storage interfaces on the other system and the Host interface on each system should be able to communicate with the Host interface on the other system.

PowerShell can be used to make IP address assignments if desired. Example 22 shows the commands used to specify static IP addresses and DNS server assignment for the interfaces on Node 1 in our environment. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 22 PowerShell commands used to configure the SMB vNIC interfaces on Node 1*

---

```
# Configure IP and subnet mask, no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "pNIC1-Port1" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "pNIC1-Port2" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
-DefaultGateway 10.10.10.1
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "pNIC1-Port1" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "pNIC1-Port1" -ServerAddresses ("10.10.10.5","10.10.10.6")
Set-DnsClient -InterfaceAlias "pNIC1-Port2" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "pNIC1-Port2" -ServerAddresses ("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
```

---

Figure 40 shows the network interfaces now configured in the server. Since the only interfaces that will be used in this solution are the interfaces derived from the physical Mellanox NIC ports and LOM/OCP Ports (2-port LOM shown), these are the only enabled interfaces that should be displayed.

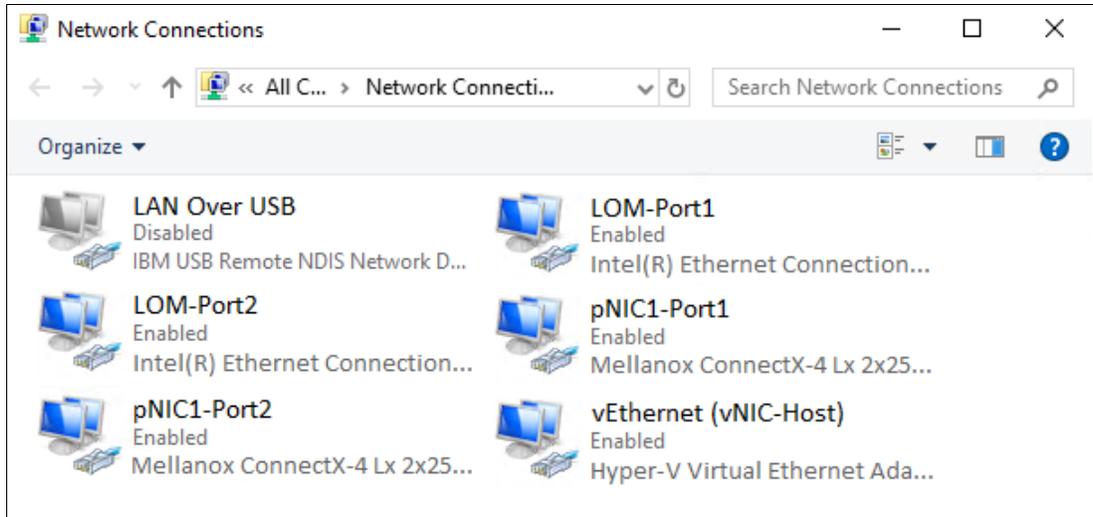


Figure 40 Final network interfaces from one dual-port Mellanox NIC and two LOM/OCP ports

Execute the commands shown in Example 20 on page 53, Example 21 on page 53, and Example 22 above on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

Since RDMA is so critical to the performance of the final solution, it is worthwhile to ensure that each piece of the configuration is correct as we move through the steps. We can't look for RDMA traffic yet, but we can verify that the vNICs (in a hyperconverged solution) have RDMA enabled. Example 23 shows the PowerShell command we use for this purpose.

Example 23 PowerShell command to verify that RDMA is enabled on the Storage interfaces

```
Get-NetAdapterRdma | ? Name -Like *pNIC* | Format-Table Name, Enabled
```

Figure 41 shows the output of the above command in our environment.

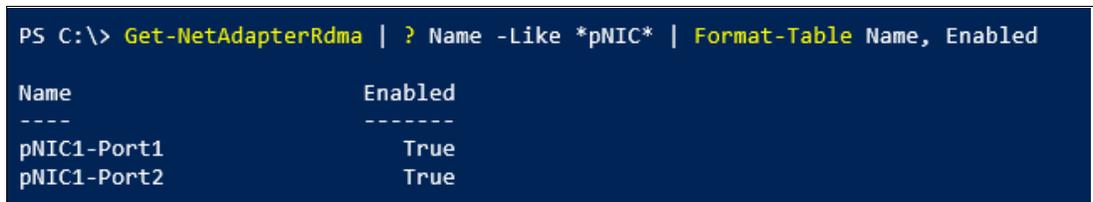


Figure 41 PowerShell command verifies that RDMA is enabled on both pNIC ports

The next piece of preparing the infrastructure for Azure Stack HCI is to perform a few optimizations to ensure the best performance possible. Proceed to “Create failover cluster” on page 84 for detailed instructions.

## iWARP: 2-16 nodes with network switches

This deployment scenario provides the steps to configure an Azure Stack HCI cluster that contains 2 - 16 nodes and uses the iWARP implementation of RDMA. In this scenario, the nodes are connected to network switches to carry all traffic. This scenario covers the use of a single dual-port Intel E810 network adapter in each node as well as two dual-port Intel E810 NICs to address the single point of failure issue that exists when using only a single network adapter. See “LOM and OCP network ports” on page 18 for a discussion of LOM and OCP network ports that might be available for your Lenovo V1 or V2 servers. For Lenovo V2 servers, the Intel E810 network adapter is available in both OCP and PCIe form factor. Figure 42 on page 56 shows a portion of the process flow diagram for this document and where this scenario fits.

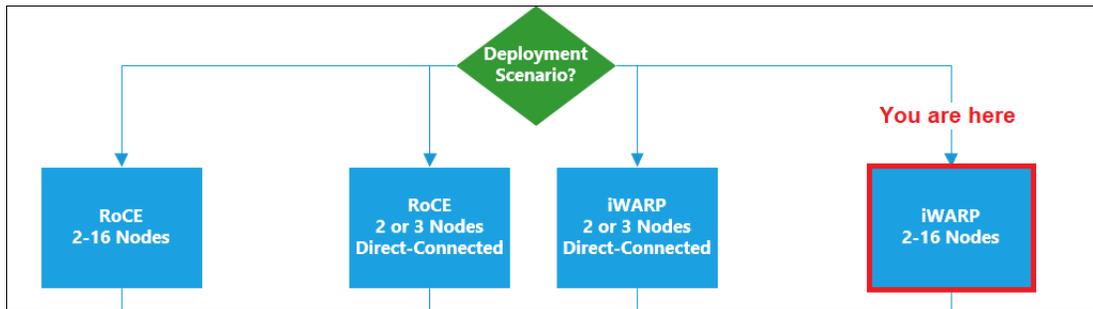


Figure 42 Portion of this document’s process flow showing the general iWARP deployment scenario

### Overview

Figure 43 shows high-level details of our 4-node configuration. The four server/storage nodes and two switches take up a combined total of 10 rack units of space. For smaller configurations that do not require network switches to handle RDMA traffic, such as a 2-node direct-connected configuration, the entire solution can take as little as 3.5” (2U) of vertical space.

	<p><b>Networking:</b> Two Lenovo ThinkSystem NE2572 RackSwitch network switches, each containing:</p> <ul style="list-style-type: none"> <li>▶ 48 ports at 10/25Gbps SFP28</li> <li>▶ 6 ports at 40/100Gbps QSFP28</li> </ul> <p><b>Compute:</b> Four Lenovo ThinkAgile MX Certified Nodes for S2D (in this case, SR650 servers), each containing:</p> <ul style="list-style-type: none"> <li>▶ Two Intel Xeon Platinum 8176 CPUs with 28 cores each, running at 2.10GHz</li> <li>▶ 384GB memory (balanced configuration, see Note below)</li> <li>▶ One or two dual-port 25GbE Intel E810 PCIe adapter(s) enabled for iWARP</li> </ul> <p><b>Storage</b> in each SR650 server:</p> <ul style="list-style-type: none"> <li>▶ Eight 3.5” hot swap HDDs and four SSDs at front</li> <li>▶ Two 3.5” hot swap HDDs at rear</li> <li>▶ ThinkSystem 430-16i SAS/SATA 12Gb HBA</li> <li>▶ M.2 Mirroring Kit with dual 480GB M.2 SSD for OS boot</li> </ul>
--	---

Figure 43 Solution configuration using ThinkAgile SXM Certified Nodes for S2D

**Note:** Although other memory configurations are possible, we highly recommend you choose a balanced memory configuration. For detailed information regarding what constitutes a balanced memory configuration, see the following documents.

For Lenovo V1 rack servers, refer to *Balanced Memory Configurations with Second-Generation Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1089>

For Lenovo V2 rack servers, refer to *Balanced Memory Configurations for 2-Socket Servers with 3rd-Generation Intel Xeon Scalable Processors* at the following URL:

<https://lenovopress.com/lp1517>

Figure 44 shows the layout of the drives and Intel E810 network adapters. There are 14 x 3.5” hot-swap drive bays in the SR650, 12 at the front of the server and two at the rear of the server. Four bays contain 1.6TB SSD devices, while the remaining ten drives are 6TB SATA HDDs. These 14 drives form the tiered storage pool of Azure Stack HCI and are connected to the ThinkSystem 430-16i SAS/SATA 12Gb HBA. In addition to the storage devices that will be used by Azure Stack HCI, a dual 480GB M.2 SSD, residing inside the server, is configured as a mirrored (RAID-1) OS boot volume.

If a single dual-port Intel E810 network adapter is used, it can be either the OCP or PCIe version of the adapter. If the PCIe version is preferred, it should be installed in PCI slot 6. If two dual-port Intel E810 network adapters are used, we recommend using one OCP version of the adapter and one PCIe version installed in PCIe slot 6. If using two PCIe adapters, the first NIC should be installed in PCI slot 6 and the second NIC should be installed in PCI slot 4, as indicated in Figure 44. This placement helps to ensure that CPU load for processing network traffic is balanced between physical processors.

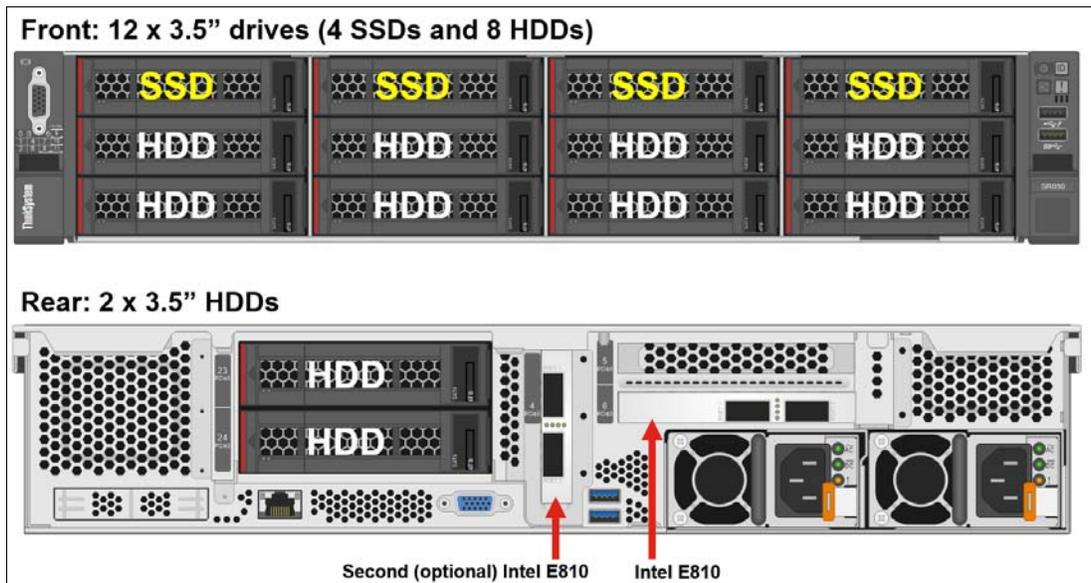


Figure 44 Lenovo ThinkSystem SR650 storage subsystem configured for iWARP

Network cabling of this solution is straight-forward, with each server being connected to each switch to enhance availability. Each system contains one or two dual-port 25GbE Intel E810 adapter(s) to handle operating system traffic and storage communication. The Intel E810 NIC is currently the only network adapter supported for iWARP on ThinkAgile MX Certified Nodes.

For a completely new environment including switches, we provide recommended network cable diagrams in “Connect servers to switches” on page 61. If using existing switches, servers can be connected to any properly configured port on each switch. If a single dual-port network adapter is used, each server is connected to each switch via a single network cable. If two dual-port network adapters are used, each server is connected to each switch twice, once from each NIC. As a best practice, network cabling is not completed until after the switches have been configured.

For the converged solution, the servers are configured with 192 GB of memory, rather than 384 GB, and the CPU has 16 cores instead of 28 cores. The higher-end specifications of the hyperconverged solution are to account for the dual functions of compute and storage that each server node will take on, whereas in the converged solution, there is a separation of duties, with one server farm dedicated to S2D and a second devoted to Hyper-V hosting.

## Configure network switches for iWARP

This section includes the steps required to configure Lenovo network switches to support RDMA via iWARP for those deployment scenarios that require network switches to handle the East-West storage traffic between Azure Stack HCI cluster nodes.

Due to the simplicity of configuring Lenovo switches for RoCE, the process to configure the same switches to carry RDMA over iWARP is very similar. In fact, the only difference in switch configuration is that Converged Enhanced Ethernet (CEE) does not need to be enabled on the switches if using network adapters that support iWARP. The Intel E810 network adapter is currently the only network adapter supported for iWARP on ThinkAgile MX Certified Nodes.

Windows Server 2019 includes a feature called SMB Direct, which supports the use of network adapters that have the Remote Direct Memory Access (RDMA) capability. Network adapters that support RDMA can function at full speed with very low latency, while using very little CPU. For workloads such as Hyper-V or Microsoft SQL Server, this enables a remote file server to resemble local storage.

SMB Direct provides the following benefits:

- ▶ **Increased throughput:** Leverages the full throughput of high speed networks where the network adapters coordinate the transfer of large amounts of data at line speed.
- ▶ **Low latency:** Provides extremely fast responses to network requests and, as a result, makes remote file storage feel as if it is directly attached block storage.
- ▶ **Low CPU utilization:** Uses fewer CPU cycles when transferring data over the network, which leaves more power available to server applications, including Hyper-V.

Leveraging the benefits of SMB Direct comes down to a few simple principles. First, using hardware that supports SMB Direct and RDMA is critical. This solution utilizes a pair of Lenovo ThinkSystem NE2572 RackSwitch Ethernet switches and one or two dual-port 25GbE Intel E810 network adapter(s) for each node, depending on the high availability requirements of the organization. If using two dual-port network adapters in each server, some (but not all) of the commands shown in this section will need to be modified or repeated, as noted.

Since RoCE uses the UDP network protocol, it must rely on Datacenter Bridging (DCB), which is a set of enhancements to IP Ethernet designed to eliminate loss due to queue overflow. On the other hand, iWARP uses the full TCP network protocol, which includes flow control and congestion management. Therefore, it is not necessary to configure DCB, PFC, or ETS for iWARP scenarios.

The following configuration commands need to be executed on *both* switches. Since ETS is not necessary for iWARP, VLAN tagging is also not required, but it is supported. We could use

multiple VLANs for different types of network traffic (storage, client, cluster heartbeat, management, Live Migration, etc.). However, for simplicity in this scenario we use a single VLAN, exactly as in the RoCE scenario, to carry all SMB Direct traffic. Employing 25GbE links makes this a viable scenario.

Example 24 shows the commands required to configure the switch ports. Make sure to adjust the “interface ethernet 1/1-4” command for the number of nodes and switch ports to be used. The example shows the command for 4 nodes using a single dual-port NIC. For more nodes or if using two dual-port NICs in each server, make sure to configure all required switch ports. For example, if configuring the switches for an 8-node Azure Stack HCI cluster in which all nodes contain two dual-port NICs, the command should be “interface ethernet 1/1-16” to configure enough switch ports to handle 2 connections from each of 8 nodes.

*Example 24 Establish VLAN for all solution traffic*

---

```
vlan 12
name SMB
exit

interface ethernet 1/1-4
switchport mode trunk
switchport trunk allowed vlan 12
spanning-tree bpduguard enable
spanning-tree port type edge
no shutdown
exit
```

---

**Note:** All switch configuration examples in this document use commands based on Lenovo switches running CNOS v10.10.1.0. The command syntax has changed significantly since the previous edition of this document.

As a best practice, it is helpful to add a description to each switch port configuration to aid in troubleshooting later. Typically, the description would indicate the destination of the port connection. This could be the Azure Stack HCI node name or might also include details regarding to which server network adapter and port the switch port is connected. To add a description to a switch port configuration, use the “description” command. Example 25 shows the commands used to specify a description for each of the switch ports (1-4) configured above.

*Example 25 Switch commands to add a description to each used port*

---

```
interface ethernet 1/1
description S2D-Node01
interface ethernet 1/2
description S2D-Node02
interface ethernet 1/3
description S2D-Node03
interface ethernet 1/4
description S2D-Node04
exit
```

---

**Note:** Another best practice that can be extremely helpful for troubleshooting is to label each network cable on both ends to indicate its source and destination. This can be invaluable if an internal server component must ever be replaced. If cable management arms are not in use, all cabling must be removed from the server in order to slide it out of the rack for component replacement. Having a label on each cable will help ensure that correct connections are made once the server is slid back into the rack.

The switch is now configured to carry RDMA traffic via the iWARP protocol. Next, we create a Link Aggregation Group (LAG) between two ports on each switch and then create an InterSwitch Link (ISL) between the pair of switches using this LAG. We establish a virtual Link Aggregation Group (vLAG) across the ISL, which is used to ensure redundant connectivity when communicating with upstream switches in the corporate network. This creates an automated network failover path from one switch to the other in case of switch or port failure.

The LAG is created between a pair of 100GbE ports on each switch. We use the first two 100GbE ports, 49 and 50, for this purpose. Physically, each port is connected to the same port on the other switch using a 100Gbps QSFP28 cable. Configuring the ISL involves joining the two ports into a channel group. We establish a vLAG across this ISL, which extends network resiliency all the way to the Azure Stack HCI cluster nodes and their NIC teams. Example 26 shows the commands to run.

*Example 26 Configure an ISL between switches and establish a vLAG for resiliency*

---

```
interface ethernet 1/49-50
switchport mode trunk
switchport trunk allowed vlan all
channel-group 100 mode active
exit

interface port-channel 100
switchport mode trunk
switchport trunk allowed vlan all
exit

vlag tier-id 100
vlag isl port-channel 100
vlag enable
exit
```

---

Establishing the LAG, ISL, and vLAG as discussed above offers the following benefits:

- ▶ Enables Azure Stack HCI cluster nodes to use a LAG across two switches
- ▶ Spanning Tree Protocol (STP) blocked interfaces are eliminated
- ▶ Topology loops are also eliminated
- ▶ Enables the use of all available uplink bandwidth
- ▶ Allows fast convergence times in case of link or device failure
- ▶ Allows link-level resiliency
- ▶ Enables high availability

To verify the completed vLAG configuration, use the "show vlag information" command. A portion of the output of this command is shown in Example 27. Run this command on both switches and compare the outputs. There should be no differences between the Local and Peer switches in the "Mis-Match Information" section. Also, in the "Role Information" section, one switch should indicate that it has the Primary role and its Peer has the Secondary role. The other switch should indicate the opposite (i.e. it has the Secondary role and its Peer has the Primary role).

*Example 27 Verification of completed vLAG configuration*

---

```
show vlag information
Global State           : enabled
VRRP active/active mode : enabled
vLAG system MAC       : 08:17:f4:c3:dd:63
ISL Information:
  PCH   Ifindex   State   Previous State
-----+-----+-----+-----
```

100 100100 Active Inactive

Mis-Match Information:

	Local	Peer
Match Result :	Match	Match
Tier ID :	100	100
System Type :	NE2572	NE2572
OS Version :	10.10.x.x	10.10.x.x

Role Information:

	Local	Peer
Admin Role :	Primary	Secondary
Oper Role :	Primary	Secondary
Priority :	0	0
System MAC :	a4:8c:db:bb:7f:01	a4:8c:db:bb:88:01

Consistency Checking Information:

State : enabled  
Strict Mode : disabled  
Final Result : pass

Once we've got the configuration complete on the switch, we need to copy the running configuration to the startup configuration. Otherwise, our configuration changes would be lost once the switch is reset or reboots. This is achieved using the `save` or `write` command (they are equivalent), as shown in Example 28 on page 61.

*Example 28 Use the write command to copy the running configuration to the startup configuration*

`write`

Repeat the entire set of commands above (Example 24 on page 59 through Example 28) on the other switch, defining the same VLAN and port trunk on that switch. Since we are using the same ports on both switches for identical purposes, the commands that are run on each switch are identical. Remember to commit the configuration changes on both switches using the `save` or `write` command.

**Note:** The steps and commands shown above are intended for use with Lenovo RackSwitch network switches running CNOS, including the G8272, NE2572, and NE10032. If the solution uses another switch model or switch vendor's equipment, it is essential to apply the equivalent command sets to the switches. The commands themselves may differ from what is stated above, but it is imperative that the same functions are executed on the switches to ensure proper operation of this solution.

### **Connect servers to switches**

To provide redundant network links in the event of a network port or external switch failure when using a single dual-port network adapter in each server, the recommendation calls for the connection from Port 1 on the Intel E810 adapter to be connected to a port on the first switch ("Switch 1"), plus a connection from Port 2 on the same Intel E810 adapter to be connected to an available port on the second switch ("Switch 2"). See Figure 45 on page 63 for the network cable diagram and Table 8 on page 64 for the network point-to-point connections for this scenario, using a single dual-port physical network adapter in each cluster node.

As a final bit of network cabling, we establish an ISL between our pair of switches to support the redundant node-to-switch cabling described above. To do this, we need redundant

high-throughput connectivity between the switches, so we connect Ports 49 and 50 on each switch to each other using a pair of 100Gbps QSFP28 cables.

**Note:** In both network cable diagrams below, the port number indicators on the switches indicate the node to which they are connected. The ISL connections are between Ports 49 and 50 on each switch.

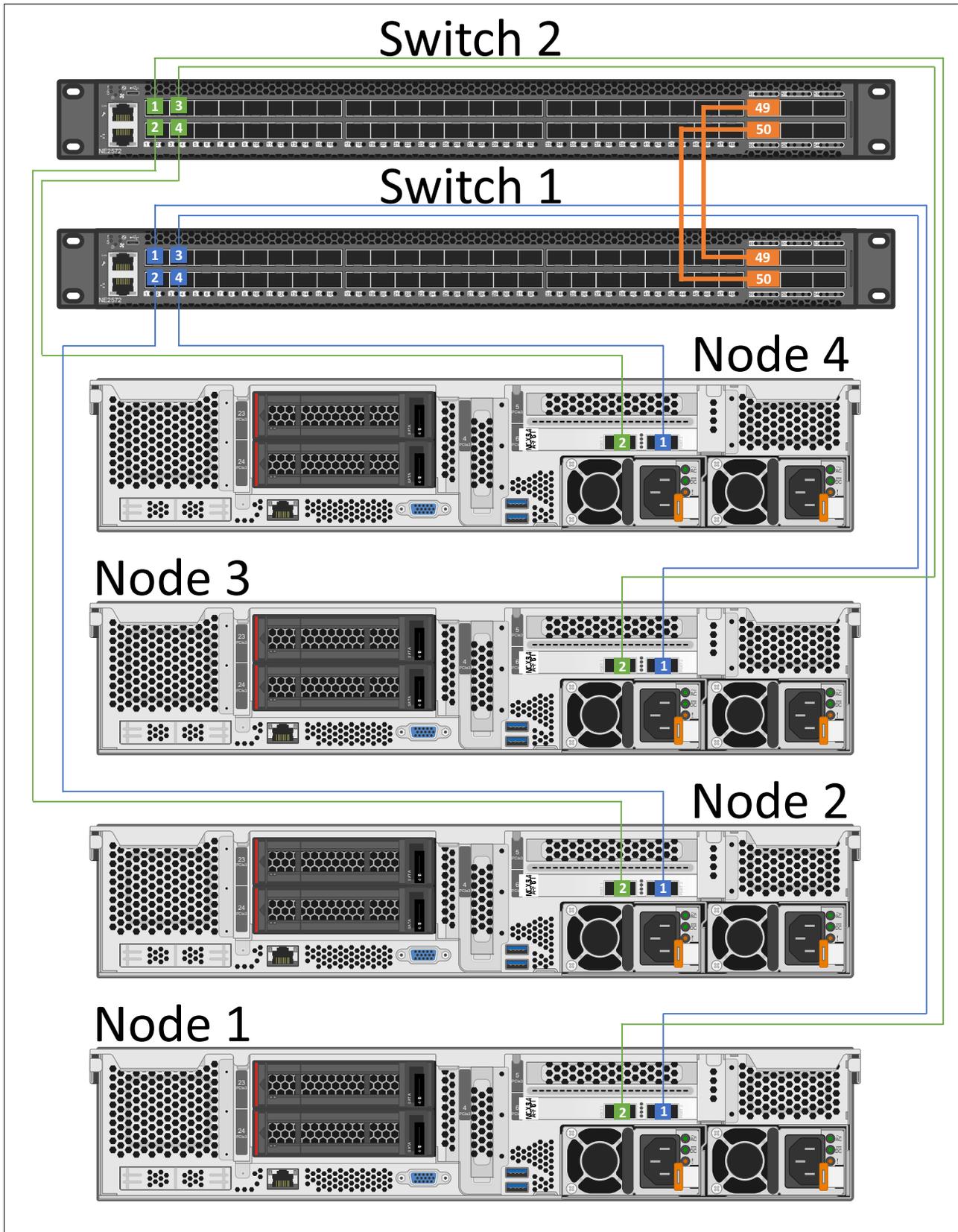


Figure 45 Switch to node connectivity using 10GbE or 25GbE AOC/DAC cables and a single dual-port NIC in each node

Table 8 shows the network point-to-point connections for this scenario when using a single dual-port physical network adapter in each cluster node.

*Table 8 Source and destination ports for a four-node cluster using a single dual-port network adapter*

Source Device	Source Port	Destination Device	Destination Port
Node 1	pNIC1-Port1	Switch 1	Port 1
Node 1	pNIC1-Port2	Switch 2	Port 1
Node 2	pNIC1-Port1	Switch 1	Port 2
Node 2	pNIC1-Port2	Switch 2	Port 2
Node 3	pNIC1-Port1	Switch 1	Port 3
Node 3	pNIC1-Port2	Switch 2	Port 3
Node 4	pNIC1-Port1	Switch 1	Port 4
Node 4	pNIC1-Port2	Switch 2	Port 4

To increase availability even further and to avoid the single point of failure associated with a single network adapter carrying all storage traffic, it is possible to configure the servers with two dual-port Intel E810 network adapters. For Lenovo V1 servers like the ThinkSystem SR650 rack server on which our examples are based, this requires two PCIe network adapters. However, for Lenovo V2 servers, to save PCIe slots in the server, one of these can be the OCP version of this network adapter. Even if one of these NICs fails completely, impacting both of its ports, the second NIC will ensure that network traffic is maintained. Using two dual-port NICs has the additional benefit of doubling the storage network throughput of the solution.

Adding a second dual-port NIC to each server simply means that each server is connected to each of the two network switches twice, once from each NIC. Figure 46 illustrates this connectivity using two Intel E810 network adapters (both NICs in the graphic are the PCIe version). Nodes 3 and 4 are not shown in this Figure to make the network connection lines more clear. Note also that the switch-to-switch connections required for the ISL are identical, regardless of the number of NICs used in each server. Table 9 on page 65 shows the network point-to-point connections for this scenario, using two dual-port physical network adapters in each cluster node.

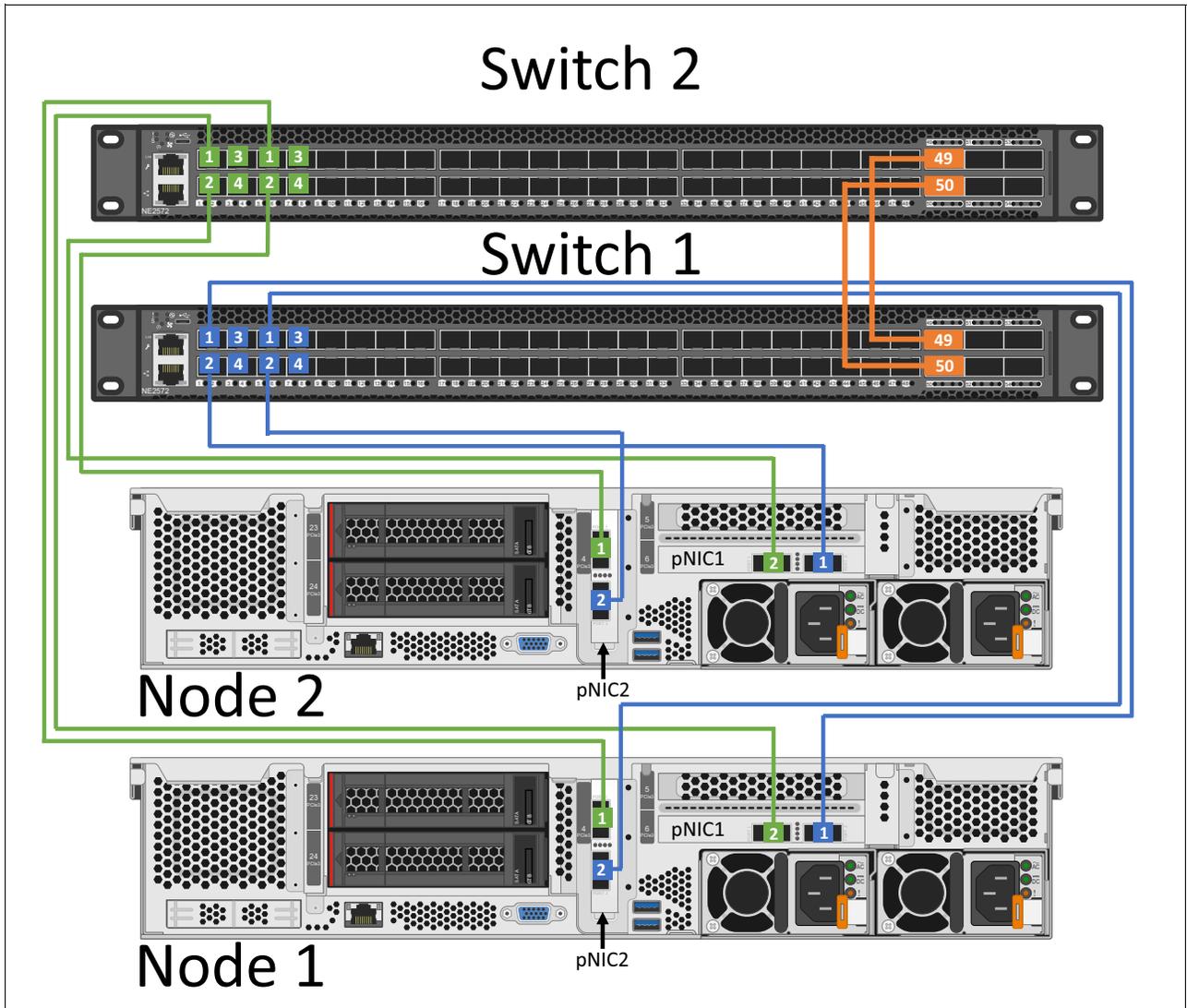


Figure 46 Switch to node connectivity using 10GbE/25GbE AOC/DAC cables and two dual-port PCIe NICs in each node

Table 9 shows the network point-to-point connections for this scenario when using two dual-port physical network adapters in each cluster node.

Table 9 Source and destination ports for a four-node cluster using two dual-port network adapters

Source Device	Source Port	Destination Device	Destination Port
Node 1	pNIC1-Port1	Switch 1	Port 1
Node 1	pNIC1-Port2	Switch 2	Port 1
Node 1	pNIC2-Port1	Switch 2	Port 5
Node 1	pNIC2-Port2	Switch 1	Port 5
Node 2	pNIC1-Port1	Switch 1	Port 2
Node 2	pNIC1-Port2	Switch 2	Port 2
Node 2	pNIC2-Port1	Switch 2	Port 6
Node 2	pNIC2-Port2	Switch 1	Port 6

Source Device	Source Port	Destination Device	Destination Port
Node 3	pNIC1-Port1	Switch 1	Port 3
Node 3	pNIC1-Port2	Switch 2	Port 3
Node 3	pNIC2-Port1	Switch 2	Port 7
Node 3	pNIC2-Port2	Switch 1	Port 7
Node 4	pNIC1-Port1	Switch 1	Port 4
Node 4	pNIC1-Port2	Switch 2	Port 4
Node 4	pNIC2-Port1	Switch 2	Port 8
Node 4	pNIC2-Port2	Switch 1	Port 8

### Configure networking parameters

We make extensive use of PowerShell commands and scripts throughout this document to configure various aspects of the environment. The commands and scripts used to configure networking parameters on the servers in this section can be used with minimal modification if you take a moment now to name the physical network adapter ports according to Table 10 before working through this section. Alternatively, you can use your own naming convention for these ports, but in this case, remember to modify the PowerShell commands appropriately.

*Table 10 Friendly names of network adapter ports used in this scenario*

	Intel E810	PCI Slot
First NIC, first port	"pNIC1-Port1"	6
First NIC, second port	"pNIC1-Port2"	6
Second NIC, first port (if used)	"pNIC2-Port1"	4
Second NIC, second port (if used)	"pNIC2-Port2"	4

PowerShell can be leveraged to configure the Intel E810 NIC ports for iWARP. Example 29 shows the commands used on servers containing **one** dual-port Intel E810 NIC.

*Example 29 Commands to enable iWARP RDMA mode on both ports of one Intel E810 NIC*

---

```
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
```

---

Example 30 shows the commands used on servers containing **two** dual-port Intel E810 NICs.

*Example 30 Commands to enable iWARP RDMA mode on all four ports of two Intel E810 NICs*

---

```
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
```

---

To increase performance and availability, we need to leverage the virtual network capabilities of Hyper-V on each host by creating SET-enabled teams from the 25GbE ports on the Intel E810 adapter(s). From this a virtual switch (vSwitch) is defined and logical network adapters (vNICs) are created to facilitate the operating system and storage traffic. Note that for the converged solution, the SET team, vSwitch, and vNICs do not need to be created. However, we generally do this anyway, just in case we'd like to run a VM or two from the storage cluster occasionally.

**One dual-port Intel E810 adapter in each server**

If using one dual-port NIC in each server, a single SET team is created across both ports of the network adapter. Figure 47 shows various details of this SET team and how it is used.

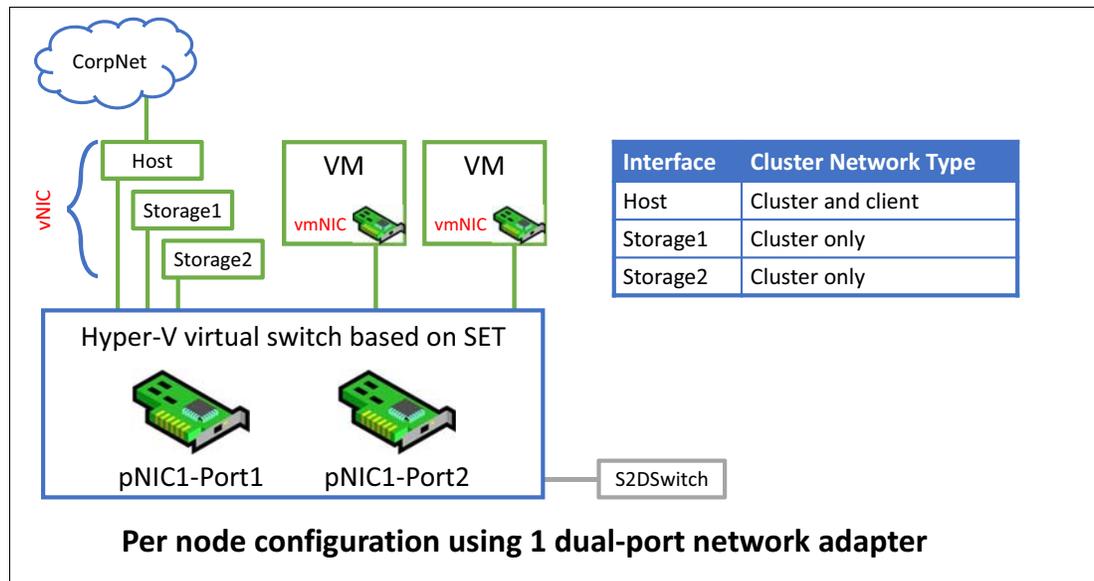


Figure 47 Diagram showing single SET team created from both ports on a single network adapter

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 10 on page 66. For a solution that includes one dual-port Intel E810 NIC in each server, three network interfaces should be displayed at this point, as shown in Figure 48.

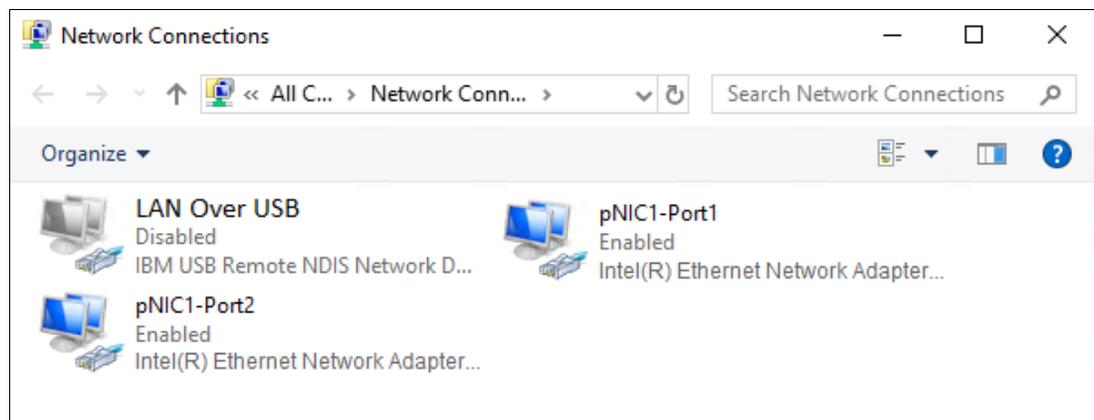


Figure 48 Network Connections control panel showing three interfaces that should exist at this point

As you can see, we have renamed the two network interfaces that we will use according to the tables above. We have also renamed the interface for the IBM USB Remote NDIS

Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since LOM ports in Lenovo V1 servers are not used in this scenario, they should be disabled in UEFI to avoid issues with cluster validation and creation later. See “LOM and OCP network ports” on page 18 for a brief overview of LOM and OCP network adapters. If any LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

We have already enabled Data Center Bridging (DCB) in “Install Windows Server roles and features” on page 23. Although not technically required for the iWARP implementation of RDMA, according to Microsoft, “testing has determined that all Ethernet-based RDMA technologies work better with DCB. Because of this, you should consider using DCB for iWARP RDMA deployments.”

For an Azure Stack HCI solution, we deploy a SET-enabled Hyper-V switch and add RDMA-enabled host virtual NICs to it for use by Hyper-V. Since many switches won't pass traffic class information on untagged VLAN traffic, we need to make sure that the vNICs using RDMA are on VLANs.

To keep this hyperconverged solution as simple as possible and since we are using dual-port 25GbE NICs, we will pass all traffic on VLAN 12. If you need to segment your network traffic more, for example to isolate VM Live Migration traffic, you can use additional VLANs.

As a best practice, we affinitize the vNICs to the physical ports on the Intel E810 network adapter. Without this step, both vNICs could become attached to the same physical NIC port, which would prevent bandwidth aggregation. It also makes sense to affinitize the vNICs for troubleshooting purposes, since this makes it clear which port carries which vNIC traffic on all cluster nodes. Note that setting an affinity will not prevent failover to the other physical NIC port if the selected port encounters a failure. Affinity will be restored when the selected port is restored to operation.

Example 31 shows the PowerShell commands that can be used to perform the SET configuration, enable RDMA, assign VLANs to the vNICs, and affinitize the vNICs to the physical NIC ports.

*Example 31 PowerShell script to create a SET-enabled vSwitch and affinitize vNICs to physical NIC ports*

---

```
# Create SET-enabled vSwitch supporting multiple uplinks provided by the Intel E810 adapter
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "pNIC1-Port1", "pNIC1-Port2" -EnableEmbeddedTeaming $true `
-AllowManagementOS $false
# Add host vNICs to the vSwitch just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Storage1" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Storage2" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Host" -ManagementOS
# Enable RDMA on the vNICs just created
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage1)"
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage2)"
# Assign the vNICs to a VLAN
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage1" -VlanId 12 -Access -ManagementOS
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage2" -VlanId 12 -Access -ManagementOS
# Affinitize vNICs to pNICs for consistency and better fault tolerance
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage1" -PhysicalNetAdapterName `
"pNIC1-Port1" -ManagementOS
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage2" -PhysicalNetAdapterName `
"pNIC1-Port2" -ManagementOS
```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

1. Configure a static IP address on the Storage1 vNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the Storage2 vNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. Perform a ping command from each interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both interfaces on each system should be able to communicate with both interfaces on all other systems.

PowerShell can be used to make IP address assignments if desired. Example 32 shows the commands used to specify static IP addresses and DNS server assignment for the interfaces on Node 1 in our environment. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 32 PowerShell commands used to configure the SMB vNIC interfaces on Node 1*

---

```
# Configure IP and subnet mask, no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
-DefaultGateway 10.10.10.1
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage1)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage2)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
```

---

Execute the commands shown in Example 31 on page 68 and Example 32 on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

It is a good idea to disable any physical network interfaces on all servers that won't be used for the solution before creating the Failover Cluster. The only interfaces that will be used in this solution are the interfaces derived from the physical Intel E810 NIC ports. Figure 49 shows these network connections.

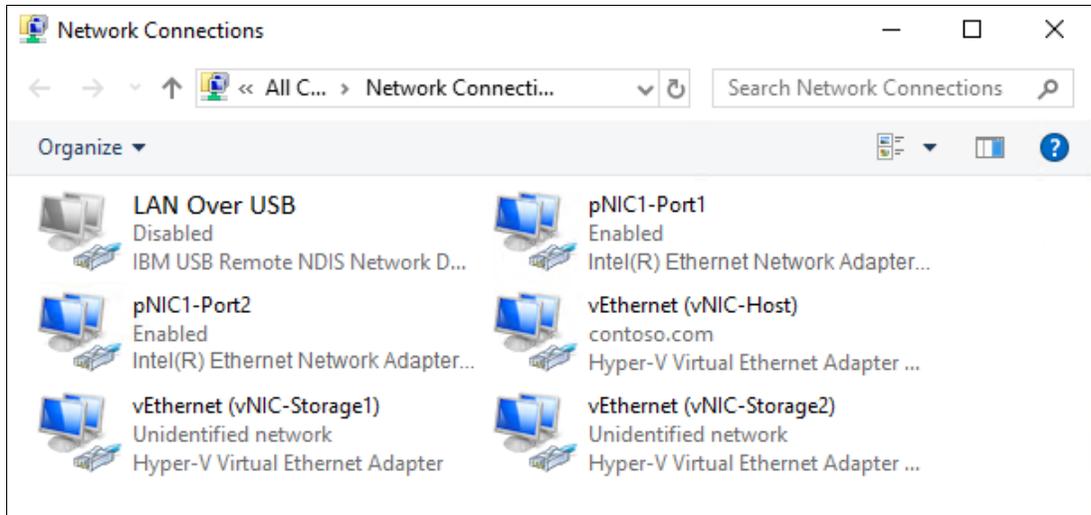


Figure 49 Final network interfaces from one dual-port Intel E810 NIC

Since RDMA is so critical to the performance of the final solution, it is worthwhile to ensure that each piece of the configuration is correct as we move through the steps. We can't look for RDMA traffic yet, but we can verify that the vNICs (in a hyperconverged solution) have RDMA enabled. Example 33 shows the PowerShell command we use for this purpose.

*Example 33 PowerShell command to verify that RDMA is enabled on the Storage interfaces*

```
Get-NetAdapterRdma | ? Name -Like *Storage* | Format-Table Name, Enabled
```

Figure 50 shows the output of the above command in our environment.

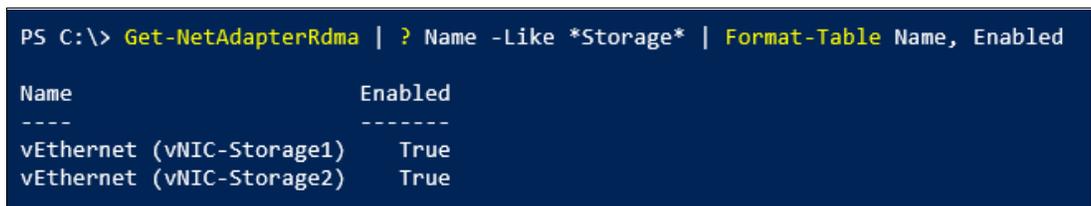


Figure 50 PowerShell command verifies that RDMA is enabled on a pair of vNICs

The next piece of preparing the infrastructure for Azure Stack HCI is to perform a few optimizations to ensure the best performance possible. Proceed to “Create failover cluster” on page 84 for detailed instructions.

**Two dual-port Intel E810 adapters in each server**

If using two dual-port NICs, we create two SET teams; one across Port 1 on both NICs and another across Port 2 on both NICs. Figure 51 on page 71 shows various details of these SET teams and how they are used. In this case, storage traffic can be isolated to one of the teams, while all other traffic, including VM Live Migration and all traffic in and out of the cluster, is carried over the other team. For best redundancy, assure that one port from each NIC is added to each SET team.

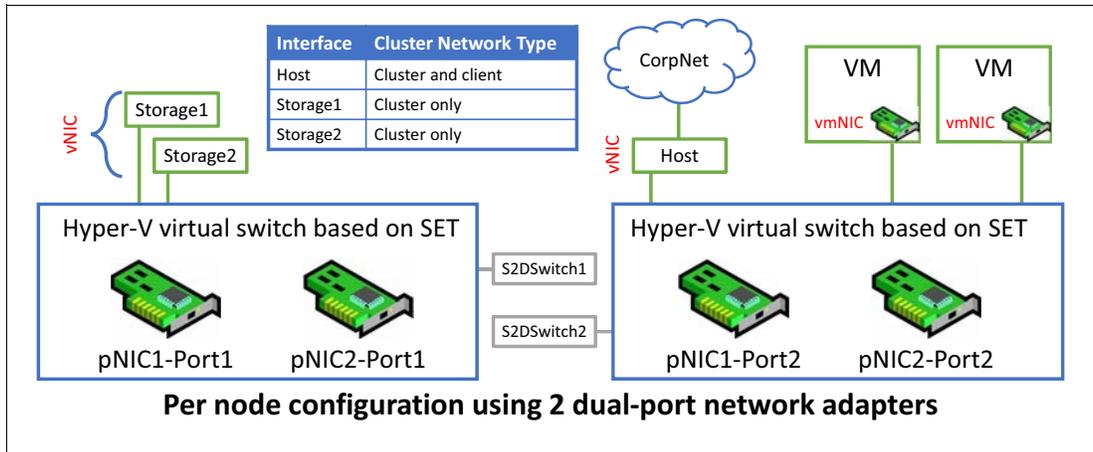


Figure 51 Diagram showing two SET teams created from two dual-port network adapters

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 10 on page 66. For a solution that includes two dual-port Intel E810 NICs in each server, five network interfaces should be displayed at this point, as shown in Figure 52.

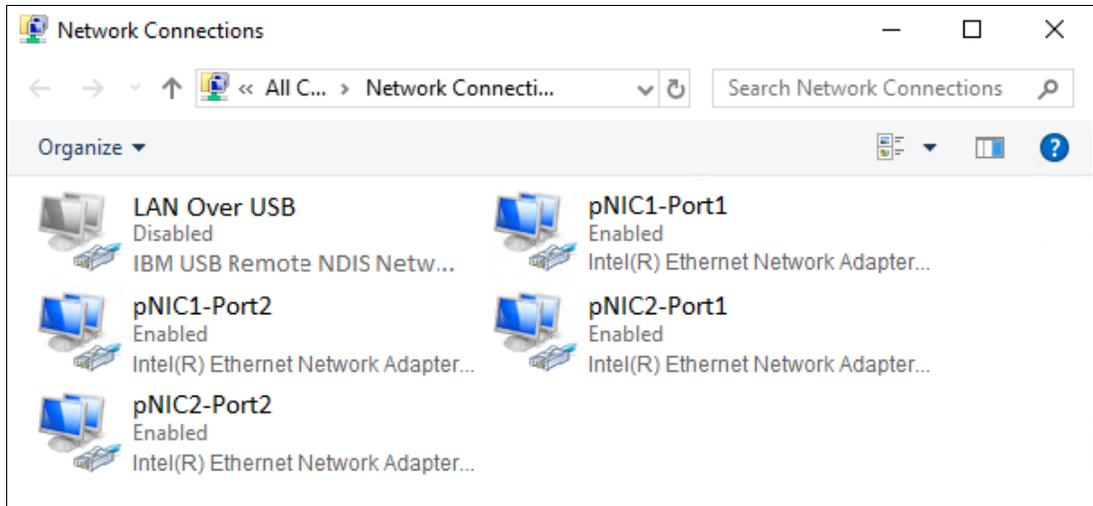


Figure 52 Network Connections control panel showing the five interfaces that should exist at this point

As you can see, we have renamed the four network interfaces that we will use according to the tables above. We have also renamed the interface for the IBM USB Remote NDIS Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since LOM ports in Lenovo V1 servers are not used in this scenario, they should be disabled in UEFI to avoid issues with cluster validation and creation later. See “LOM and OCP network ports” on page 18 for a brief overview of LOM and OCP network adapters. If any LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

The process and commands used to configure two dual-port Intel E810 network adapters (4 physical network ports total) are nearly identical to those shown in the previous section. In

this section we show only the required commands and a few notes. For more detail about exactly what is being configured and why, refer to the previous section.

Example 34 shows the PowerShell commands that can be used to perform the SET configuration, enable RDMA, assign VLANs to the vNICs, and affinitize the vNICs to the physical NIC ports.

*Example 34 PowerShell script to create a SET-enabled vSwitch and affinitize vNICs to physical NIC ports*

---

```
# Create SET-enabled vSwitches supporting multiple uplinks provided by Intel E810 NICs
New-VMSwitch -Name "S2DSwitch1" -NetAdapterName "pNIC1-Port1", "pNIC2-Port1" -EnableEmbeddedTeaming $true -
-AllowManagementOS $false
New-VMSwitch -Name "S2DSwitch2" -NetAdapterName "pNIC1-Port2", "pNIC2-Port2" -EnableEmbeddedTeaming $true -
-AllowManagementOS $false
# Add host vNICs to the vSwitches just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch1" -Name "vNIC-Storage1" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch1" -Name "vNIC-Storage2" -ManagementOS
Add-VMNetworkAdapter -SwitchName "S2DSwitch2" -Name "vNIC-Host" -ManagementOS
# Enable RDMA on Storage vNICs just created, but not on Host vNIC
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage1)"
Enable-NetAdapterRDMA -Name "vEthernet (vNIC-Storage2)"
# Assign vNIC traffic to vLAN(s)
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage1" -VlanId 12 -Access -ManagementOS
Set-VMNetworkAdapterVlan -VMNetworkAdapterName "vNIC-Storage2" -VlanId 12 -Access -ManagementOS
# Affinitize vNICs to pNICs for consistency and better fault tolerance
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage1" -PhysicalNetAdapterName `
"pNIC1-Port1" -ManagementOS
Set-VMNetworkAdapterTeamMapping -VMNetworkAdapterName "vNIC-Storage2" -PhysicalNetAdapterName `
"pNIC2-Port1" -ManagementOS
```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

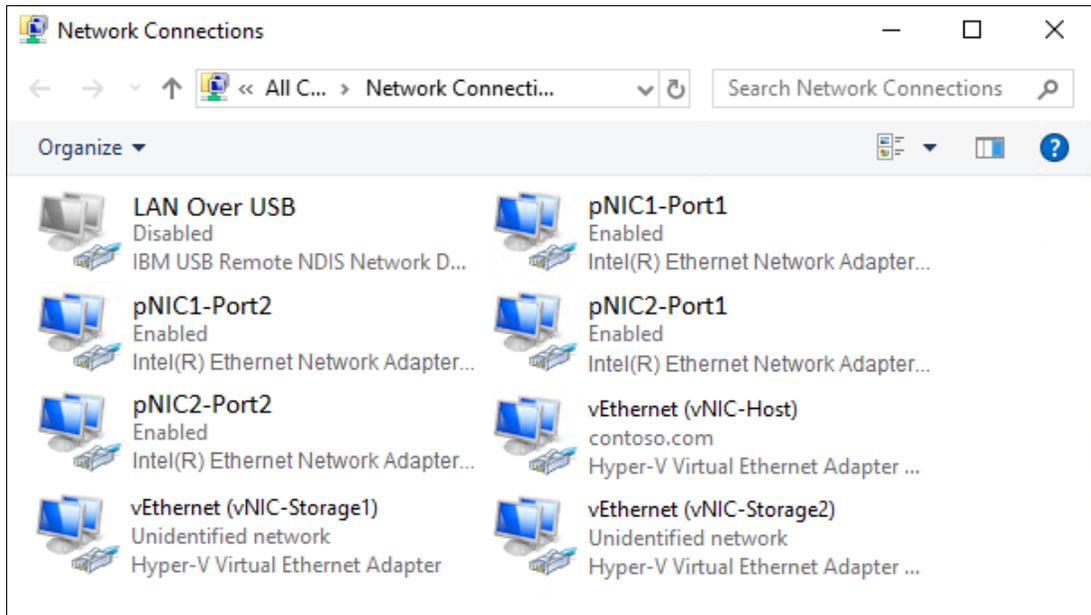
1. Configure a static IP address on the Storage1 vNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the Storage2 vNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. Configure a static IP address on the Host vNIC, using a different subnet if desired. Since this interface will carry network traffic into and out of the Azure Stack HCI cluster (North-South traffic), this will likely be a “CorpNet” subnet. You must specify a DNS server and register this interface with DNS. You must also configure a default gateway for this interface.
4. Perform a ping command from each Storage interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both Storage interfaces on each system should be able to communicate with both Storage interfaces on all other systems.

Example 35 shows the commands used to specify static IP addresses and DNS server assignment for each interface on Node 1 in our environment. These are exactly the same commands that are used if only one dual-port Mellanox network adapter is installed in each server. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 35 PowerShell commands used to configure the SMB vNIC interfaces on Node 1*

```
# Configure IP and subnet mask, no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
-DefaultGateway 10.10.10.1
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage1)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage1)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
Set-DnsClient -InterfaceAlias "vEthernet (vNIC-Storage2)" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Storage2)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
```

Figure 53 shows the network interfaces now configured in the server. Since the only interfaces that will be used in this solution are the interfaces derived from the physical Intel E810 NIC ports, these are the only enabled interfaces that should be displayed.



*Figure 53 Final network interfaces from two dual-port Intel E810 NICs*

Execute the commands shown in Example 34 on page 72 and Example 35 on page 73 on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

Disable any physical network interfaces on all servers that won't be used for the solution so these unused interfaces won't cause an issue when creating the Failover Cluster later. The only interfaces that will be used in this solution are the interfaces derived from the ports on the physical Intel E810 NICs.

Example 36 shows the PowerShell command we use to confirm that RDMA is enabled on the appropriate interfaces.

Example 36 PowerShell command to verify that RDMA is enabled on the Storage interfaces

```
Get-NetAdapterRdma | ? Name -Like *Storage* | Format-Table Name, Enabled
```

Figure 54 on page 74 shows the output of the above command in our environment.

```
PS C:\> Get-NetAdapterRdma | ? Name -Like *Storage* | Format-Table Name, Enabled

Name                Enabled
----                -
vEthernet (vNIC-Storage1) True
vEthernet (vNIC-Storage2) True
```

Figure 54 PowerShell command verifies that RDMA is enabled on a pair of vNICs

The next piece of preparing the infrastructure for Azure Stack HCI is to perform a few optimizations to ensure the best performance possible. Proceed to “Create failover cluster” on page 84 for detailed instructions.

## iWARP: 2-4 nodes, direct-connected

This deployment scenario provides the steps to configure an Azure Stack HCI cluster that contains 2 to 4 nodes that are direct-connected for East-West storage traffic, and uses the iWARP implementation of RDMA. Figure 55 shows a portion of the process flow diagram for this document and where this scenario fits. Although the diagram still refers to “2 or 3 Nodes” since detailed instructions are provided for 2- and 3-node clusters, a 4-node direct-connected cluster can be deployed by extrapolating these instructions.

**Note:** Although these instructions can be used to deploy a 2-node direct-connected cluster of ThinkAgile MX1021 on SE350 Certified Nodes, we have published a separate document dedicated to this solution. Please refer to the *ThinkAgile MX1021 on SE350 Azure Stack HCI (S2D) Deployment Guide*, which can be found at the following URL:

<https://lenovopress.com/lp1298>

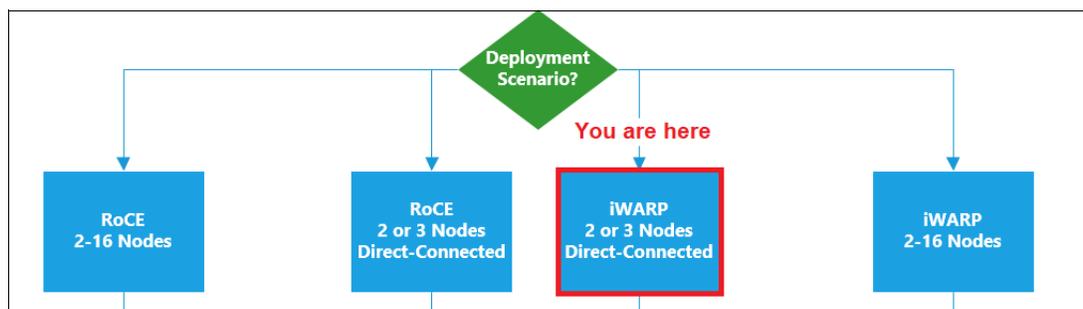


Figure 55 Portion of this document’s process flow showing the direct-connected iWARP scenario

### Overview

By “direct-connected” we refer to the high-speed NICs in the nodes being connected directly to each other (without a switch between them) for storage traffic. Note that a switch is still needed to connect the nodes to the corporate network (“CorpNet”) to pass North-South traffic into and out of the cluster.

Figure 56 shows diagrams of various network connectivity models between cluster nodes. Microsoft does not support bridged connectivity between cluster nodes and does not recommend single-link connectivity. The only recommended approach is to provide full mesh dual-link connectivity between all nodes for East-West storage traffic. The best way to provide two network connections between each of the nodes in a cluster without using a switch between them is by using “N-1” dual-port Intel E810 network adapters in each node, where N is the number of cluster nodes. This means that for a 3-node cluster, each node would require 2 dual-port network adapters for storage traffic and for a 4-node cluster, each node would require 3 dual-port network adapters.

Although the benefits of switchless deployments diminish with clusters larger than three-nodes due to the number of network adapters required, we have added details in this section that are specific to 4-node direct-connected clusters due to customer demand. We will not carry the discussion past four nodes since the resulting configurations become quite complicated. Make sure to understand what is required and supported by Microsoft for direct-connected Azure Stack HCI clusters. An excellent source of this information is an article titled Physical network requirements for Azure Stack HCI at the following URL:

<https://learn.microsoft.com/en-us/azure-stack/hci/concepts/physical-network-requirements?tabs=overview%2C22H2reqs>

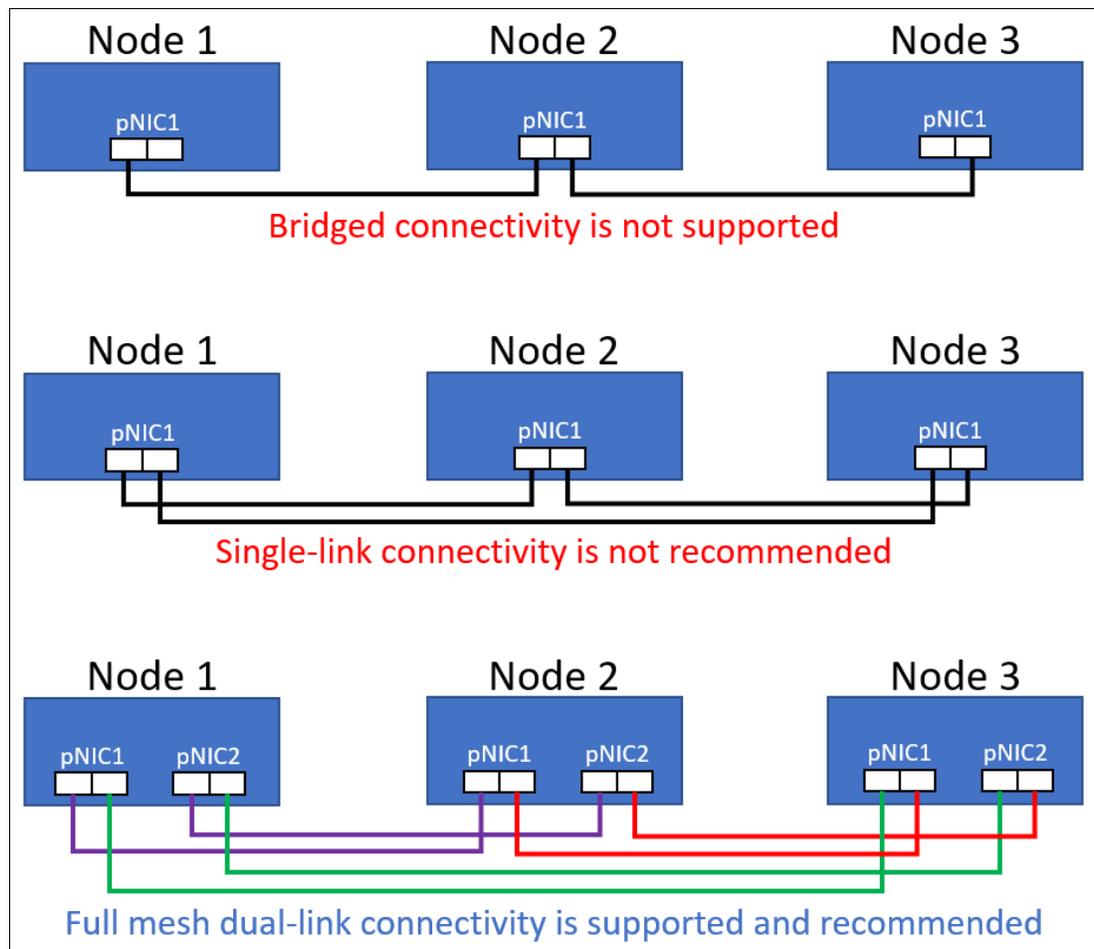


Figure 56 Various node-to-node network connectivity models

### Connect servers to each other

For a 2-node cluster, each of the ports in the Intel E810 NIC is connected directly to the same port on the same NIC in the other node to carry East-West storage traffic. For North-South management traffic, we connect two additional network ports on each node to the corporate network. For the ThinkSystem SR650 rack server used in our examples, it is convenient to use two LOM ports for this purpose. Figure 57 shows the network cable diagram for a 2-node direct-connected cluster using LOM ports for management traffic.

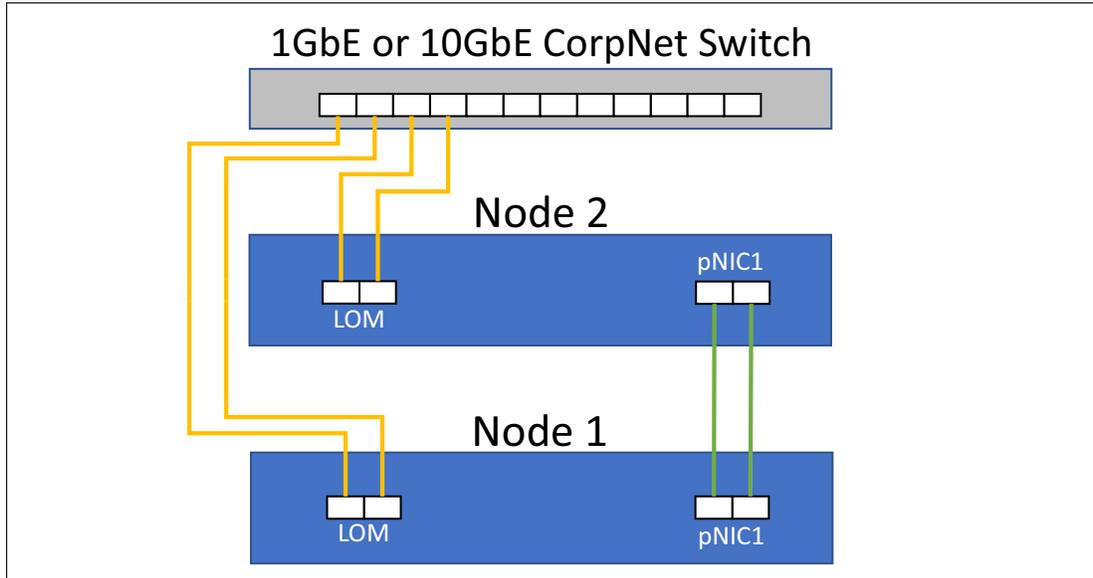


Figure 57 Two-node connectivity for direct-connected deployment scenario

For a 3-node cluster, the ports in the Intel E810 NICs are connected directly to the other nodes in a full mesh dual-link configuration. That is, each node is connected to each of the other two nodes in a redundant manner, which requires a total of four network ports (two dual-port Intel E810 NICs) in each node for East-West Storage traffic. In addition, like the 2-node configuration above, we connect two LOM/OCP ports on each node to the corporate network to carry North-South Management/Compute traffic. Figure 58 shows the network cable diagram for a 3-node direct-connected cluster. Connection line colors in the figure represent connections between two nodes for East-West traffic or between the nodes and the corporate network for North-South traffic.

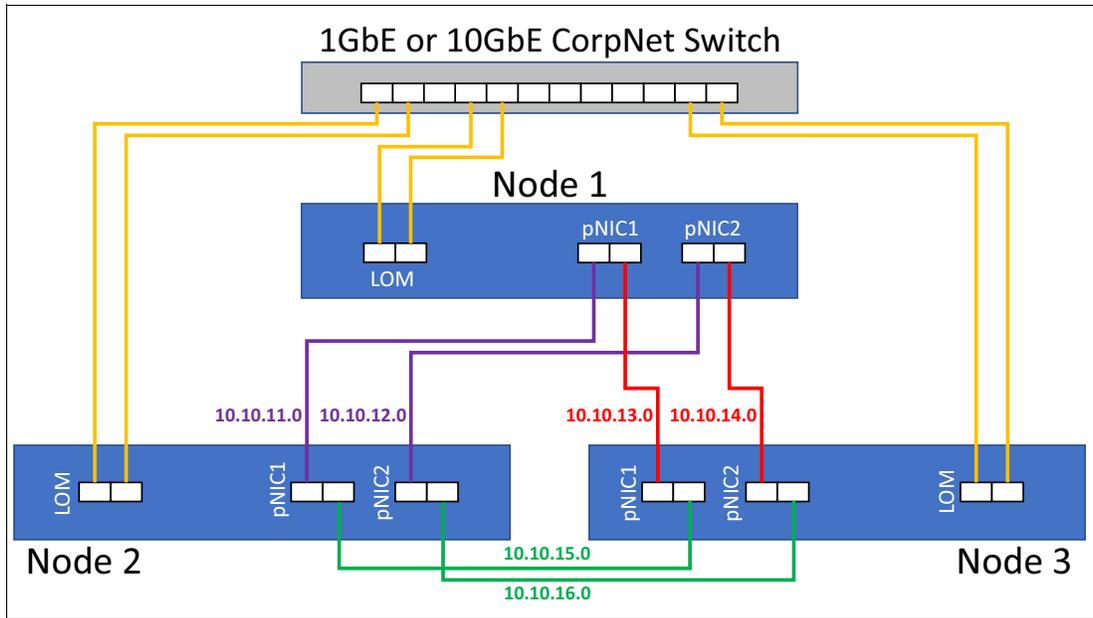


Figure 58 Three-node connectivity for direct-connected deployment scenario

For a 4-node cluster, the ports in the Intel E810 NICs are connected directly to the other nodes in a full mesh dual-link configuration. That is, each node is connected to each of the other three nodes in a redundant manner, which requires a total of six network ports (three dual-port Intel E810 NICs) in each node. In addition, like the 2- and 3-node configurations above, we connect two LOM/OCP ports on each node to the corporate network. Figure 59 shows the network cable diagram for a 4-node direct-connected cluster. We have removed the connection lines to the switches for North-South traffic for clarity. Connection line colors in the figure represent connections between two nodes (for example between Node 1 and Node 2) for East-West traffic.

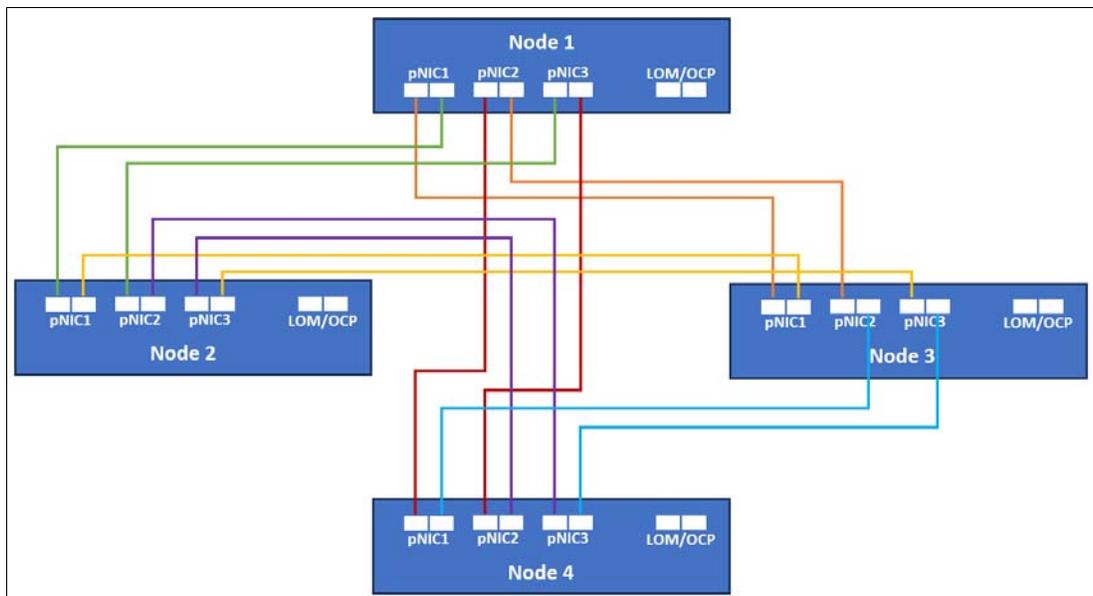


Figure 59 Four-node connectivity for direct-connected deployment scenario

Removing the dual-port network adapter boxes from the diagram above results in further clarity, which is shown in Figure 60 on page 78. This simplified diagram shows the network

cabling for storage traffic between all nodes and includes an example of the subnets used for Storage traffic. Make sure to use ports from *different* network adapters to connect any two nodes. This will ensure that if the network adapter fails, connectivity between the two nodes will survive.

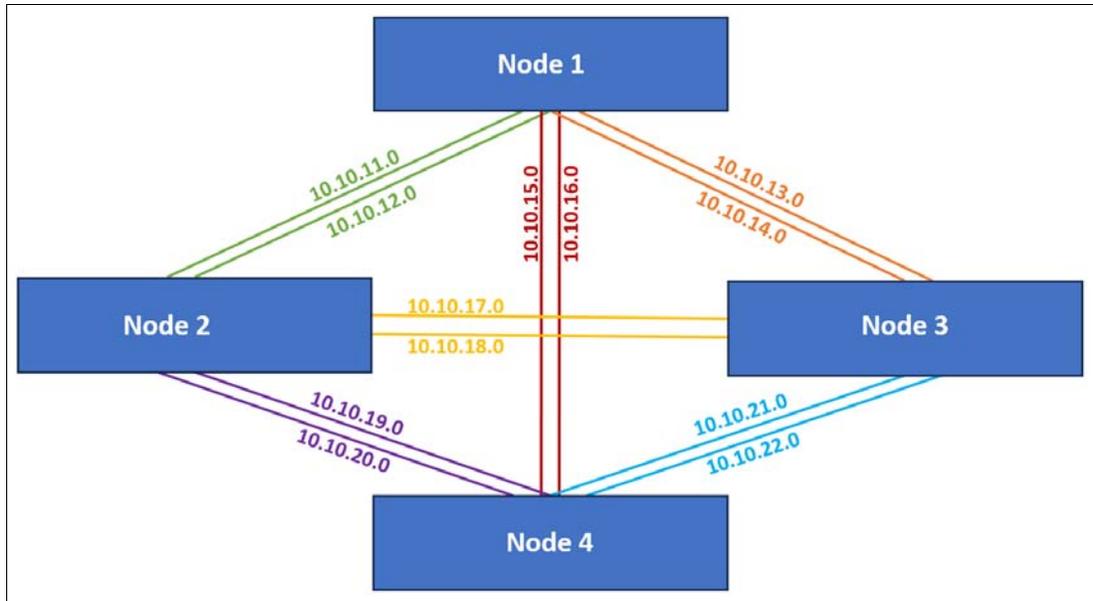


Figure 60 Simplified four-node connectivity for direct-connected scenario with example subnets shown

**Note:** This document provides detailed instructions and PowerShell scripts for both 2-node and 3-node direct-connected Azure Stack HCI cluster deployment. These details can be extrapolated to deploy a 4-node direct-connected cluster.

In all cases of direct-connected deployment, it is required to establish a subnet for each of the high-speed connections between the nodes for storage traffic. That is, a subnet for each of the network cables that are connected between the nodes. For a 3-node direct-connected cluster, this means a total of six subnets and for a 4-node direct-connected cluster, a total of twelve subnets are required. The subnet we use for each connection is shown next to the connection lines in Figure 58 on page 77 for three nodes and Figure 60 for four nodes.

Table 11 shows the high-speed direct network connections between each of the nodes in a 3-node cluster, as well as the subnet that carries the traffic for each of these connections. The subnets shown are consistent with examples in this document. If you prefer to use your own subnets, make sure to modify the example PowerShell commands accordingly.

Table 11 Source and destination ports for full-mesh 3-node direct-connected HCI cluster

Source Device	Source Port	Destination Device	Destination Port	Subnet
Node 1	pNIC1 Port 1	Node 2	pNIC1 Port 1	10.10.11.0/24
Node 1	pNIC1 Port 2	Node 3	pNIC1 Port 1	10.10.12.0/24
Node 1	pNIC2 Port 1	Node 2	pNIC2 Port 1	10.10.13.0/24
Node 1	pNIC2 Port 2	Node 3	pNIC2 Port 1	10.10.14.0/24
Node 2	pNIC1 Port 1	Node 1	pNIC1 Port 1	10.10.11.0/24
Node 2	pNIC1 Port 2	Node 3	pNIC1 Port 2	10.10.15.0/24

Source Device	Source Port	Destination Device	Destination Port	Subnet
Node 2	pNIC2 Port 1	Node 1	pNIC2 Port 1	10.10.13.0/24
Node 2	pNIC2 Port 2	Node 3	pNIC2 Port 2	10.10.16.0/24
Node 3	pNIC1 Port 1	Node 1	pNIC1 Port 2	10.10.12.0/24
Node 3	pNIC1 Port 2	Node 2	pNIC1 Port 2	10.10.15.0/24
Node 3	pNIC2 Port 1	Node 1	pNIC2 Port 2	10.10.14.0/24
Node 3	pNIC2 Port 2	Node 2	pNIC2 Port 2	10.10.16.0/24

If configuring a 3-node cluster, make sure to modify the PowerShell scripts shown in Example 40 on page 83 to ensure that proper IP addressing is used to establish all six subnets. In this case, the PowerShell commands that are run on one node are not exactly the same as the commands run on the other two nodes. Use the Subnet column in Table 11 to modify these commands.

If configuring a 4-node cluster, the same applies, but additional PowerShell commands will need to be added to configure the additional network ports and subnets required to support four nodes.

With all the physical network connections made, we move to configuring the network interfaces on the servers.

### Configure networking parameters

For the iWARP two-node direct-connect scenario, our examples use the SR650 LOM ports to carry “CorpNet” traffic into and out of the cluster (i.e. North-South management traffic). To increase performance and availability, we need to leverage the virtual network capabilities of Hyper-V on each host by creating a SET-enabled team from the LOM ports. For a brief discussion of the difference between LOM and OCP network ports, refer to “LOM and OCP network ports” on page 18.

Also, for all iWARP direct-connect deployment scenarios, we do not create a SET-enabled team from the Intel E810 NIC ports. In this deployment scenario, the storage traffic is carried by the physical network adapter ports (pNICs). Figure 61 on page 79 shows a diagram representing this difference in a two-node direct-connect scenario.

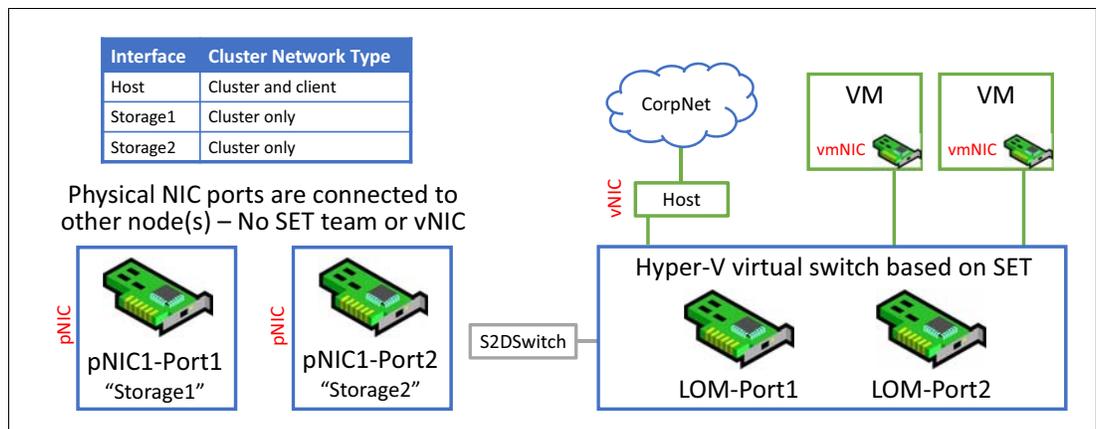


Figure 61 Two-node direct-connect network interfaces

We make extensive use of PowerShell commands and scripts throughout this document to configure various aspects of the Azure Stack HCI environment. Those commands and scripts used to configure networking parameters on the servers in this section can be used with minimal modification if you take a moment now to name the physical network adapter ports according to Table 12 before working through this section. Alternatively, you can use your own naming convention for these ports, but in this case, remember to modify the PowerShell commands appropriately.

Table 12 Friendly names of network adapter ports used in this scenario

	Intel E810	PCI Slot
First NIC, first port	"pNIC1-Port1"	6
First NIC, second port	"pNIC1-Port2"	6
Second NIC, first port (if used)	"pNIC2-Port1"	4
Second NIC, second port (if used)	"pNIC2-Port2"	4

For this direct-connected scenario, the LOM/OCP ports are used for North-South traffic, which includes VM traffic. Naming of the LOM/OCP ports is shown in Table 13.

Table 13 Friendly names of LOM/OCP ports used in this scenario

	LOM/OCP
First Port	"LOM-Port1"
Second port	"LOM-Port2"
Third port (if used)	"LOM-Port3"
Fourth port (if used)	"LOM-Port4"

The scripts in this section can be used with minimal modification if the physical network adapters are named according to Table 12 and Table 13. For a solution that includes one dual-port Intel E810 NIC in each server and uses 2 LOM/OCP ports, five network interfaces should be displayed at this point, as shown in Figure 25.

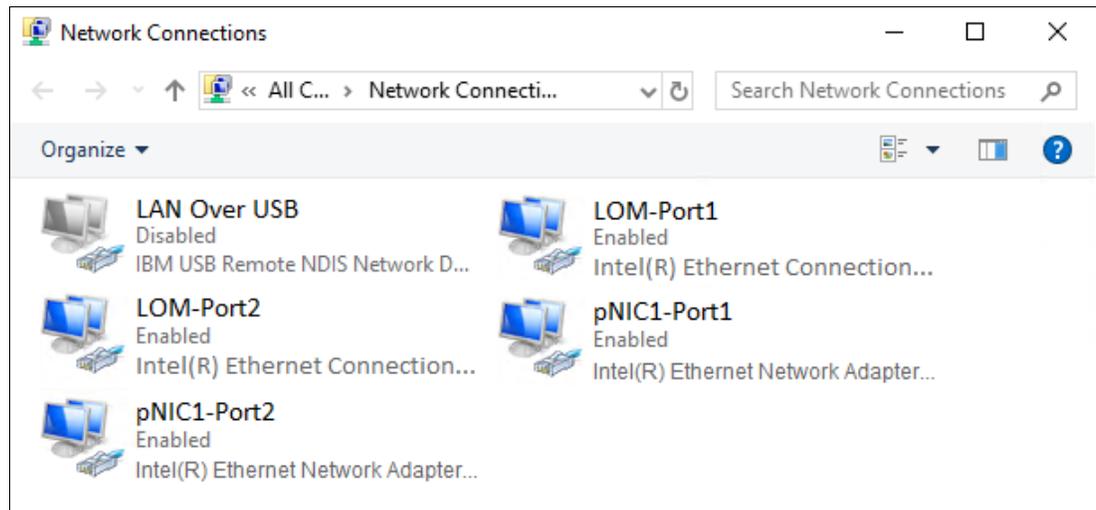


Figure 62 Network Connections control panel showing the five interfaces that should exist at this point

As you can see, we have renamed the four network interfaces that we will use according to the tables above. We have also renamed the interface for the IBM USB Remote NDIS Network Device to “LAN Over USB” and have disabled it to avoid issues later with cluster creation. This interface is only used for inband communication to the XCC for tasks such as updating firmware on a system component. It can be safely disabled in the operating system, since it will be enabled automatically when needed and disabled after use.

Since only the first two LOM ports are used in this scenario, additional LOM ports should be disabled in UEFI, if present, to avoid issues with cluster validation and creation later. If any unneeded LOM ports are still visible to the OS, follow the steps in “Disable unneeded LOM ports in UEFI (V1 servers)” on page 20 to disable them in System Setup.

PowerShell can be leveraged to configure the Intel E810 NIC ports for iWARP. Example 37 shows the commands used on servers containing **one** dual-port Intel E810 NIC.

*Example 37 Commands to enable iWARP RDMA mode on both ports of one Intel E810 NIC*

---

```
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
```

---

Example 38 on page 81 shows the commands used on servers containing **two** dual-port Intel E810 NICs.

*Example 38 Commands to enable iWARP RDMA mode on all four ports of two Intel E810 NICs*

---

```
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC1-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port1" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
Set-NetAdapterAdvancedProperty -Name "pNIC2-Port2" -DisplayName "NetworkDirect Technology" `
-DisplayValue "iWarp"
```

---

We have already enabled Data Center Bridging (DCB) in “Install Windows Server roles and features” on page 23. Although not technically required for the iWARP implementation of RDMA, according to Microsoft, “testing has determined that all Ethernet-based RDMA technologies work better with DCB. Because of this, you should consider using DCB for iWARP RDMA deployments.”

For the direct-connected scenario, we do not create a SET team from the Intel E810 NIC ports, but we do so for the LOM/OCP ports that carry North-South traffic. From the SET-enabled team we create from the LOM/OCP ports, a virtual switch (“S2DSwitch” in Figure 61 on page 79) is defined and a logical network adapter (vNIC) is created for use by virtual machines in the hyperconverged solution. Note that for the converged solution, the SET team, vSwitch, and vNIC do not need to be created. However, we generally do this anyway, just in case we’d like to run a VM or two from the storage cluster occasionally.

Example 39 on page 82 shows the PowerShell commands that can be used to perform the SET team configuration and enable RDMA on the physical Intel E810 NIC ports. In this scenario, the SET team is created from the LOM/OCP ports to enable the Hyper-V switch for virtual machine use. If the servers have 4-port LOMs, all 4 ports can be used for this purpose. Make sure to run the appropriate command to create the SET team (one of the first two commands in the example, but not both). Alternatively, if 4-port LOMs are present, but you only want to use two ports, you should disable Ports 3 and 4 in System UEFI before

proceeding. This will help to avoid complications when creating the cluster that may occur when unused network adapters are visible to the OS.

In addition, the commands shown add a vNIC to the vSwitch, enable RDMA on the physical Intel E810 NIC ports for storage traffic, and disable RDMA on the physical LOM/OCP ports, since storage traffic should not traverse these ports.

*Example 39 PowerShell script to create a SET-enabled vSwitch, add vNICs to it, and enable RDMA on the pNIC ports*

---

```
# Create SET-enabled vSwitch for Hyper-V using 2 LOM/OCP ports
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "LOM-Port1", "LOM-Port2" -EnableEmbeddedTeaming $true `
  -AllowManagementOS $false

# Note: Run the next command only if using 4 LOM ports
# Create SET-enabled vSwitch for Hyper-V using 4 LOM ports
New-VMSwitch -Name "S2DSwitch" -NetAdapterName "LOM-Port1", "LOM-Port2", "LOM-Port3", "LOM-Port4" `
  -EnableEmbeddedTeaming $true -AllowManagementOS $false

# Add host vNIC to the vSwitch just created
Add-VMNetworkAdapter -SwitchName "S2DSwitch" -Name "vNIC-Host" -ManagementOS
# Enable RDMA on Intel E810 pNIC ports
Enable-NetAdapterRDMA -Name "pNIC1-Port1"
Enable-NetAdapterRDMA -Name "pNIC1-Port2"
# Disable RDMA on LOM/OCP pNIC ports
Disable-NetAdapterRDMA -Name "LOM-Port1"
Disable-NetAdapterRDMA -Name "LOM-Port2"
```

---

Now that all network interfaces have been created, IP address configuration can be completed, as follows:

1. Configure a static IP address on the NIC1-Port1 pNIC (for example, 10.10.11.x). The DNS server is specified, but this interface should not be registered with DNS, since it is not intended to carry traffic outside the cluster. For the same reason, a default gateway is not configured for this interface.
2. Configure a static IP address on the NIC1-Port2 pNIC, using a different subnet if desired (for example, 10.10.12.x). Again, specify the DNS server, but do not register this interface with DNS, nor configure a default gateway.
3. If configuring a 3-node cluster, make sure to modify the PowerShell scripts shown in Example 40 to ensure that proper IP addressing is used to establish all six subnets. In this case, the PowerShell commands that are run on one node are not exactly the same as the commands run on the other two nodes. Use the Subnet column in Table 11 on page 78 to modify these commands.
4. Configure a static IP address on the Host vNIC, using a different subnet if desired. Since this interface will carry network traffic into and out of the Azure Stack HCI cluster (North-South traffic), this will likely be a “CorpNet” subnet. You must specify a DNS server and register this interface with DNS. You must also configure a default gateway for this interface.
5. Perform a ping command from each Storage interface to the corresponding servers in this environment to confirm that all connections are functioning properly. Both Storage interfaces on each system should be able to communicate with both Storage interfaces on the other system and the Host interface on each system should be able to communicate with the Host interface on the other system.

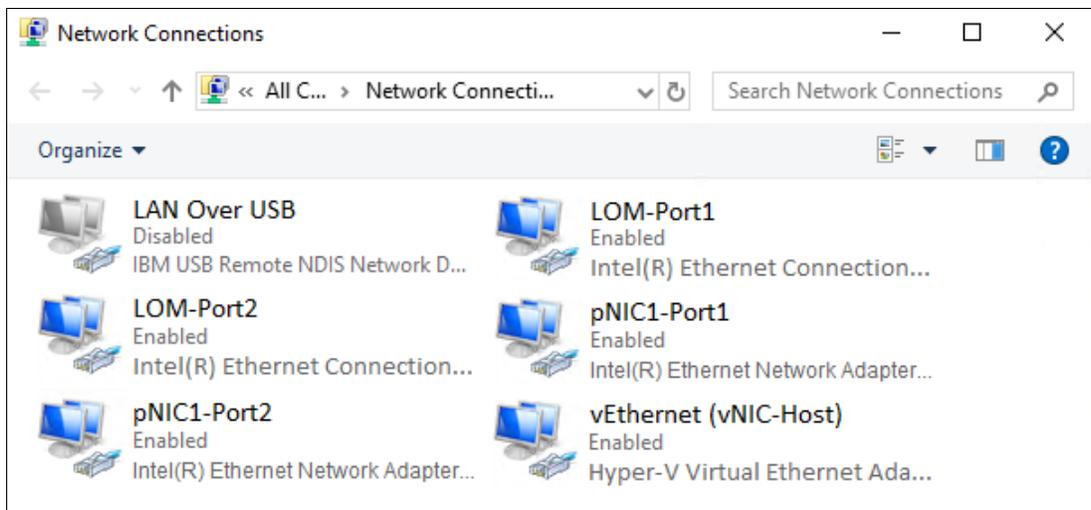
PowerShell can be used to make IP address assignments if desired. Example 40 shows the commands used to specify static IP addresses and DNS server assignment for the interfaces

on Node 1 in our environment. Make sure to change the IP addresses and subnet masks (prefix length) to appropriate values for your environment.

*Example 40 PowerShell commands used to configure the SMB vNIC interfaces on Node 1*

```
# Configure IP and subnet mask, no default gateway for Storage interfaces
New-NetIPAddress -InterfaceAlias "pNIC1-Port1" -IPAddress 10.10.11.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "pNIC1-Port2" -IPAddress 10.10.12.11 -PrefixLength 24
New-NetIPAddress -InterfaceAlias "vEthernet (vNIC-Host)" -IPAddress 10.10.10.11 -PrefixLength 24 `
-DefaultGateway 10.10.10.1
# Configure DNS on each interface, but do not register Storage interfaces
Set-DnsClient -InterfaceAlias "pNIC1-Port1" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "pNIC1-Port1" -ServerAddresses ("10.10.10.5","10.10.10.6")
Set-DnsClient -InterfaceAlias "pNIC1-Port2" -RegisterThisConnectionsAddress $false
Set-DnsClientServerAddress -InterfaceAlias "pNIC1-Port2" -ServerAddresses ("10.10.10.5","10.10.10.6")
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (vNIC-Host)" -ServerAddresses `
("10.10.10.5","10.10.10.6")
```

Figure 63 shows the network interfaces now configured in the server. Since the only interfaces that will be used in this solution are the interfaces derived from the physical Intel E810 NIC ports and LOM/OCF Ports (2-port LOM shown), these are the only enabled interfaces that should be displayed.



*Figure 63 Final network interfaces from one dual-port Intel E810 NIC and two LOM/OCF ports*

Execute the commands shown in Example 39 on page 82 and Example 40 on page 83 on the other servers that will become nodes in the Azure Stack HCI cluster. Make sure to modify parameters that change from server to server, such as IP address.

The next piece of preparing the infrastructure for S2D is to perform a few optimizations to ensure the best performance possible. Proceed to “Create failover cluster” on page 84 for detailed instructions.

## Create failover cluster

Before creating the Failover Cluster, all the servers that will become nodes in the Azure Stack HCI cluster need to be identically configured. Running Windows Update on all cluster member servers at the same time ensures consistent OS patching between nodes. Once all servers are added to the Active Directory (AD) domain, the cluster validation process is executed and the cluster is created. Finally, a cluster witness is configured and specified.

### Perform Windows Update and join AD domain

To ensure that the latest fixes and patches are applied to the operating system consistently on all servers, perform updating of the Windows Server components via Windows Update. It is a good idea to reboot each node after the final update is applied to ensure that all updates have been fully installed, regardless what Windows Update indicates.

Upon completing the Windows Update process, join each server node to the Windows AD domain. The PowerShell command shown in Example 41 can be used to accomplish this task. Make sure to edit the domain name before executing the command.

*Example 41 PowerShell command to add system to an Active Directory Domain*

---

```
Add-Computer -DomainName "contoso.com" -Restart
```

---

From this point onward, when working with cluster services be sure to log onto the systems with a *Domain account* and not the local Administrator account.

### Cluster validation and creation

Next, we need to validate the components that are necessary to form the cluster. As an alternative to using the GUI, the following PowerShell commands can be used to test and create the Failover Cluster, Example 42.

*Example 42 PowerShell commands to test and create a failover cluster*

---

```
Test-Cluster -Node S2D-Node01,S2D-Node02,S2D-Node03,S2D-Node04 -Include Inventory, `
    Network, "Storage Spaces Direct", "System Configuration"
New-Cluster -Node S2D-Node01,S2D-Node02,S2D-Node03,S2D-Node04 -StaticAddress 10.10.11.10 `
    -NoStorage
```

---

Once the cluster is built, you can also use PowerShell to query the health status of the cluster storage, as shown in Example 43.

*Example 43 PowerShell command to check the status of cluster storage*

---

```
Get-StorageSubSystem *S2D-Cluster
```

---

The Health Status should show “Healthy” and the Operational Status should show “OK” in the command output returned by PowerShell.

The default behavior of Failover Cluster creation is to set aside the non-public facing subnet (configured on the Storage2 vNIC) as a cluster heartbeat network. When 1GbE was the standard, this made perfect sense. However, since we are using 25GbE in this solution, we don’t want to dedicate half our bandwidth to this important, but mundane task.

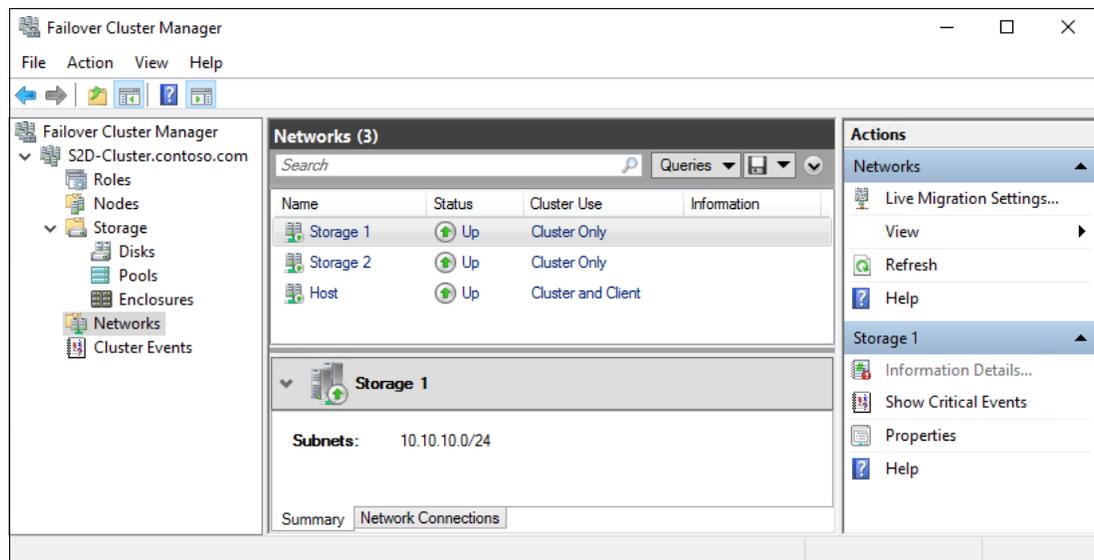
It is also a good idea to specify cluster network names that make sense and will aid in troubleshooting later. To be consistent, we name our cluster networks after the vNICs that carry the traffic for each.

It is possible to accomplish the cluster network role and name changes using PowerShell, as shown in Example 44.

*Example 44 PowerShell script to change names and roles of cluster networks*

```
# Update the cluster networks that were created by default
# First, look at what's there
Get-ClusterNetwork | ft Name, Role, Address
# Change the cluster network names so they're consistent with the individual nodes
(Get-ClusterNetwork -Name "Cluster Network 1").Name = "Storage1"
(Get-ClusterNetwork -Name "Cluster Network 2").Name = "Storage2"
(Get-ClusterNetwork -Name "Cluster Network 3").Name = "Host"
# Enable Cluster Only traffic on the Storage networks
(Get-ClusterNetwork -Name "Storage1").Role = 1
(Get-ClusterNetwork -Name "Storage2").Role = 1
# Enable Cluster and Client traffic on the Host network
(Get-ClusterNetwork -Name "Host").Role = 3
# Check to make sure the cluster network names and roles are set properly
Get-ClusterNetwork | ft Name, Role, Address
```

After making these changes, both networks should show “Cluster and Client” in the Cluster Use column and the network names should be more useful, as shown in Figure 64.



*Figure 64 Cluster networks shown with names to match the vNICs that carry their traffic*

## Cluster file share witness

It is recommended to create a cluster file share witness. The cluster file share witness quorum configuration enables the 4-node cluster to withstand up to two node failures.

For information on a new file share witness feature in Windows Server (beginning with Windows Server 2019) and HCI OSes that does not utilize the Cluster Name Object (CNO), refer to the Microsoft article, *New File Share Witness Feature in Windows Server 2019*, available at the following URL:

<https://techcommunity.microsoft.com/t5/failover-clustering/new-file-share-witness-feature-in-windows-server-2019/ba-p/372149>

Once the cluster is operational and the file share witness has been established, it is time to enable and configure S2D.

## Enable and configure Storage Spaces Direct

Once the failover cluster has been created, run the PowerShell command in Example 45 to enable S2D on the cluster.

*Example 45 PowerShell command to enable Storage Spaces Direct*

---

```
Enable-ClusterStorageSpacesDirect S2D-Cluster -PoolFriendlyName S2D-Pool
```

---

This PowerShell command will enable Storage Spaces Direct and create a single storage pool that has a name as specified by the `-PoolFriendlyName` parameter. It will also configure S2D storage tiers based on the number of nodes in the cluster and the storage device types contained in the nodes.

To check the storage tiers that were created, run the command shown in Example 46.

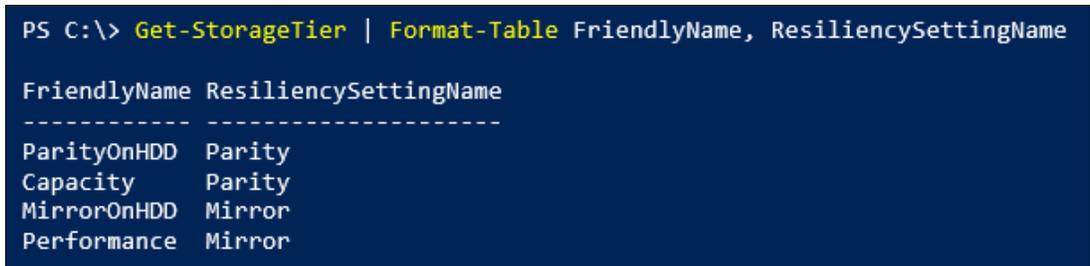
*Example 46 PowerShell command to check S2D storage tiers*

---

```
Get-StorageTier | Format-Table FriendlyName, ResiliencySettingName
```

---

Our results for a 4-node cluster are shown in Figure 65. As volumes are created, they will show up in the output list, along with their resiliency setting.

A screenshot of a PowerShell terminal window with a dark blue background. The prompt is 'PS C:\>'. The command entered is 'Get-StorageTier | Format-Table FriendlyName, ResiliencySettingName'. The output is a table with two columns: 'FriendlyName' and 'ResiliencySettingName'. The table has four rows of data: 'ParityOnHDD' with 'Parity', 'Capacity' with 'Parity', 'MirrorOnHDD' with 'Mirror', and 'Performance' with 'Mirror'.

```
PS C:\> Get-StorageTier | Format-Table FriendlyName, ResiliencySettingName

FriendlyName ResiliencySettingName
-----
ParityOnHDD  Parity
Capacity     Parity
MirrorOnHDD  Mirror
Performance  Mirror
```

*Figure 65 PowerShell query showing resiliency settings for storage tiers*

## Verify RDMA functionality

At this point we can also check to make sure RDMA is working. We provide two suggested approaches for this. First, Figure 66 on page 88 shows a simple `netstat` command that can be used to verify that listeners are in place on port 445 (last two lines in the figure). This is the port typically used for SMB and the port specified when we created the network QoS policy for SMB Direct.

```

PS C:\> Netstat -xan

Active NetworkDirect Connections, Listeners, SharedEndpoints

Mode        IfIndex Type           Local Address      Foreign Address    PID
-----
Kernel     13 Connection   10.10.10.11:41098  10.10.10.12:445   0
Kernel     13 Connection   10.10.10.11:24457  10.10.10.12:445   0
Kernel     13 Connection   10.10.10.11:24201  10.10.10.12:445   0
Kernel     13 Connection   10.10.10.11:23945  10.10.10.12:445   0
.
.
.
Kernel     13 Listener    10.10.10.11:445   NA                 0
Kernel     12 Listener    10.10.12.11:445   NA                 0

```

Figure 66 The netstat command can be used to confirm listeners configured for port 445

The second method for verifying that RDMA is configured and working properly is to use PerfMon to create an RDMA monitor. To do this, following these steps:

1. At the PowerShell or Command prompt, type perfmon and press **Enter**.
2. In the Performance Monitor window that opens, select **Performance Monitor** in the left pane and click the **green plus sign** (“+”) at the top of the right pane.

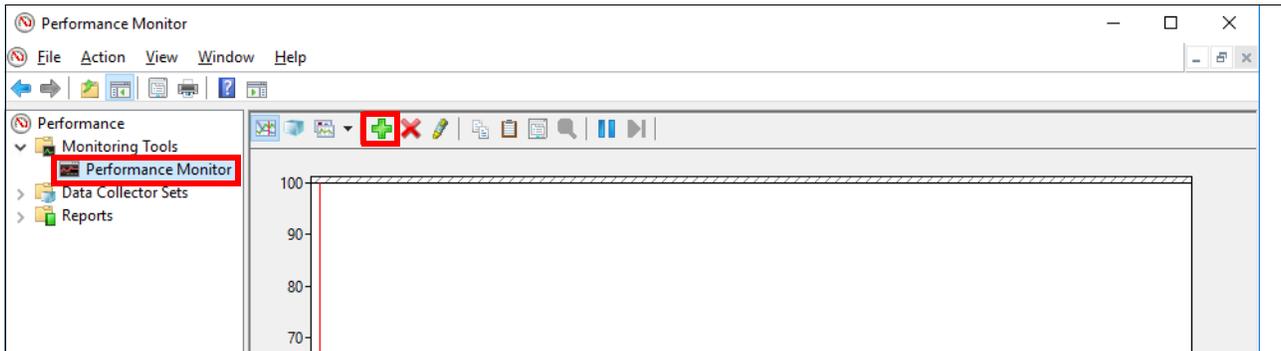


Figure 67 Initial Performance Monitor window before configuration

3. In the Add Counters window that opens, select **RDMA Activity** in the upper left pane. In the Instances of selected object area in the lower left, choose the instances that represent your vNICs (for our environment, these are “Hyper-V Virtual Ethernet Adapter” and “Hyper-V Virtual Ethernet Adapter #2”). Once the instances are selected, click the Add button to move them to the Added counters pane on the right. Click **OK**.

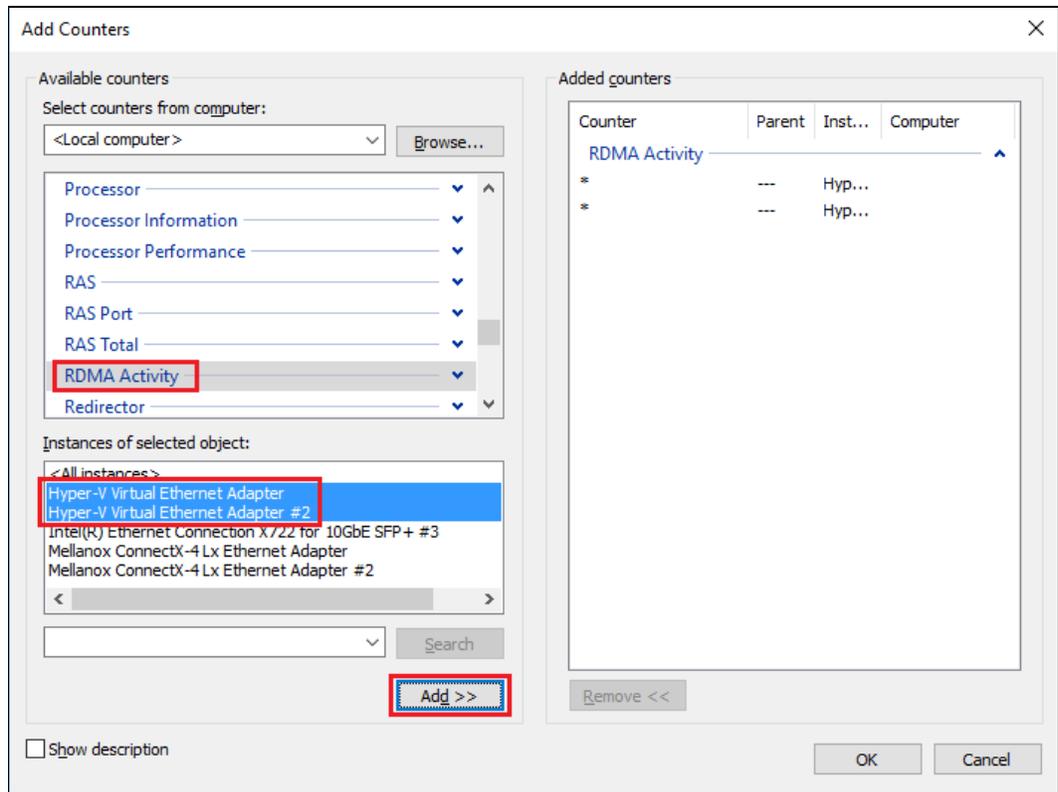


Figure 68 The Add counters window for Performance Monitor

- Back in the Performance Monitor window, click the drop-down icon to the left of the green plus sign and choose **Report**.

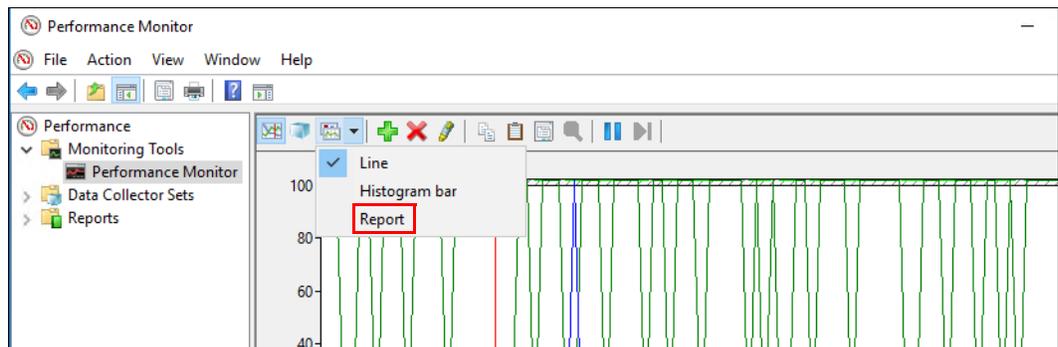


Figure 69 Choose the "Report" format

- This should show a report of RDMA activity for your vNICs. Here you can view key performance metrics for RDMA connections in your environment, as shown in Figure 70 on page 90.

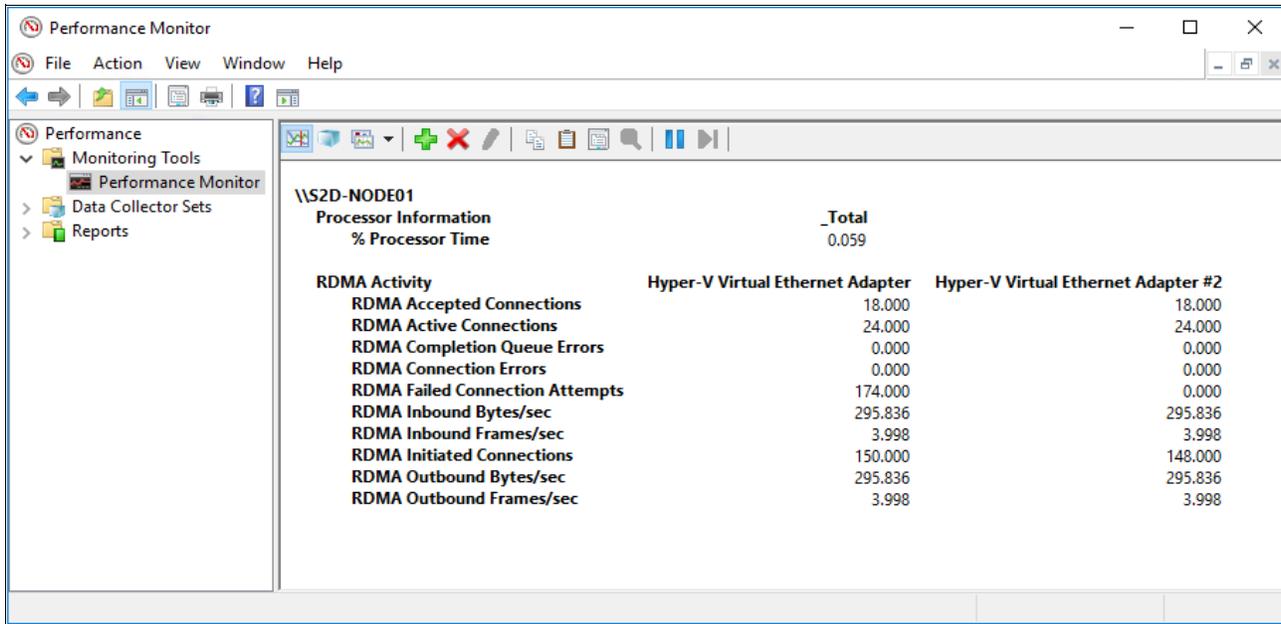


Figure 70 Key RDMA performance metrics

## Create virtual disks

After the Azure Stack HCI cluster is created, create virtual disks or volumes based on your performance requirements. There are three common volume types for general deployments:

- ▶ Mirror
- ▶ Dual Parity
- ▶ Mirror-Accelerated Parity (MAP)

Table 14 shows the volume types supported by Storage Spaces Direct and several characteristics of each.

Table 14 Summary of characteristics associated with common storage volume types

	Mirror	Dual Parity	MAP
<b>Optimized for</b>	Performance	Efficiency	Archival
<b>Use case</b>	All data is hot	All data is cold	Mix of hot and cold data
<b>Storage efficiency</b>	33% - 50%	50% - 80%	33% - 80%
<b>File system</b>	ReFS or NTFS	ReFS or NTFS	ReFS only
<b>Minimum nodes</b>	2	4	4

Microsoft provides a good summary of how to plan for storage volumes and how to create them at the following URLs, respectively:

<https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/plan-volumes>

<https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/create-volumes>

You can use the PowerShell commands in Example 47 through Example 49 on page 91 to create and configure virtual disks. Choose any or all types of volumes shown, adjusting the volume names and sizes to suit your needs. Note that you might also need to change the

-StorageTierFriendlyName parameter to match your environment. Use the PowerShell command shown in Example 46 on page 87 to check your options for this parameter.

Create a Mirror volume using the command in Example 47.

*Example 47 PowerShell command to create a new Mirror volume*

---

```
New-Volume -StoragePoolFriendlyName S2D-Pool -FriendlyName "Mirror" -FileSystem CSVFS_ReFS `
-StorageTierfriendlyNames Performance -StorageTierSizes 6TB
```

---

Create a Parity volume using the command in Example 48.

*Example 48 PowerShell command to create a new Parity volume*

---

```
New-Volume -StoragePoolFriendlyName S2D-Pool -FriendlyName "Parity" -FileSystem CSVFS_ReFS `
-StorageTierfriendlyNames Capacity -StorageTierSizes 24TB
```

---

Create a Mirror-Accelerated Parity volume using the command in Example 49.

*Example 49 PowerShell command to create a new Mirror-Accelerated Parity volume*

---

```
New-Volume -StoragePoolFriendlyName S2D-Pool -FriendlyName "MAP" -FileSystem CSVFS_ReFS `
-StorageTierfriendlyNames Performance, Capacity -StorageTierSizes 2TB, 8TB
```

---

Once S2D installation is complete and volumes have been created, the final step is to verify that there is fault tolerance in the storage environment. Example 50 shows the PowerShell command to verify the fault tolerance of the S2D storage pool.

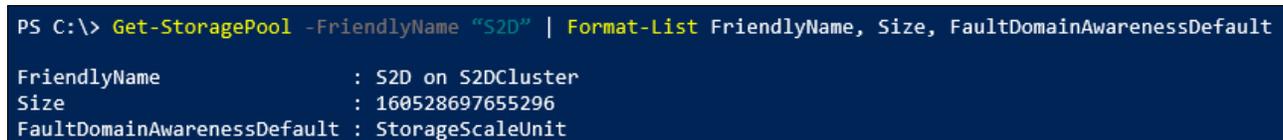
*Example 50 PowerShell command to determine S2D storage pool fault tolerance*

---

```
Get-StoragePool -FriendlyName S2DPool | Format-List FriendlyName, Size, FaultDomainAwarenessDefault
```

---

Figure 71 shows the output of the above command in our environment.



```
PS C:\> Get-StoragePool -FriendlyName "S2D" | Format-List FriendlyName, Size, FaultDomainAwarenessDefault
FriendlyName           : S2D on S2DCluster
Size                   : 160528697655296
FaultDomainAwarenessDefault : StorageScaleUnit
```

*Figure 71 PowerShell query showing the fault domain awareness of the storage pool*

To Query the virtual disks, use the command in Example 51. The command displays the fault tolerance of each virtual disk (volume) in the S2D storage pool.

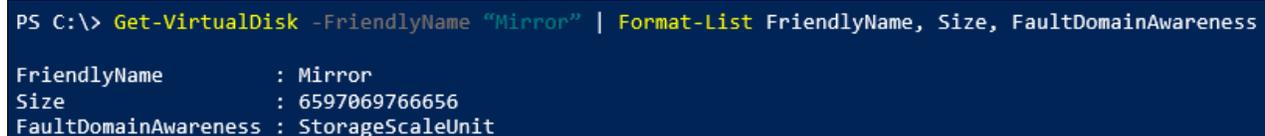
*Example 51 PowerShell command to determine S2D virtual disk (volume) fault tolerance*

---

```
Get-VirtualDisk | Format-List FriendlyName, Size, FaultDomainAwareness
```

---

Figure 72 shows the output of the above command in our environment.



```
PS C:\> Get-VirtualDisk -FriendlyName "Mirror" | Format-List FriendlyName, Size, FaultDomainAwareness
FriendlyName           : Mirror
Size                   : 6597069766656
FaultDomainAwareness   : StorageScaleUnit
```

*Figure 72 PowerShell query showing the fault domain awareness of a virtual disk*

Over time, the storage pool may get unbalanced because of adding or removing physical disks/storage nodes or data written or deleted to the storage pool. In this case, use the PowerShell command shown in Example 52 to improve storage efficiency and performance.

*Example 52 PowerShell command to optimize the S2D storage pool*

---

```
optimize-storagepool S2DPool
```

---

Depending on the amount of data that needs to be moved around, the above command can take a significant length of time to complete.

# Cluster set creation

The concept of cluster sets was introduced in Windows Server 2019 and is supported for Azure Stack HCI clusters. As such, only clusters containing nodes that are running these operating systems (or later) can participate in a cluster set. Also, PowerShell must be used to create and configure cluster sets (i.e. there is no GUI at this time). We see cluster sets as an effective way to overcome some of the limitations of a single Windows Server Failover Cluster. This section provides background information and details needed to assist in creating and configuring a cluster set that includes multiple Azure Stack HCI clusters.

## Introduction to cluster sets

A cluster set is a loosely-coupled grouping of multiple failover clusters. Each individual cluster contained in the cluster set can be compute, storage, or hyperconverged clusters. Using cluster sets, you can effectively increase cluster node count in a single Azure Stack HCI cloud by orders of magnitude. Since the cluster set feature enables a unified storage namespace across the set, it enables virtual machine fluidity *across* member clusters within a cluster set. That is, VMs can be migrated across the single failover cluster boundary that has previously been a limit.

While preserving existing failover cluster management experiences on individual member clusters, a cluster set instance additionally offers key use cases around lifecycle management at the aggregate. For more information about Microsoft Cluster Sets, refer to the Microsoft article at the following URL:

<https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/cluster-sets>

At a high level, a cluster set is a collection of failover clusters that is tied to a management cluster that hosts a highly-available management plane for the entire cluster set. One of the keys to cluster set functionality is the Infrastructure Scale-Out File Server (SOFS) role that is new in Windows Server 2019. A cluster can support only one Infrastructure SOFS role, whether the cluster is a member cluster or the management cluster in a cluster set. It is this role that provides the unified storage namespace across all member clusters.

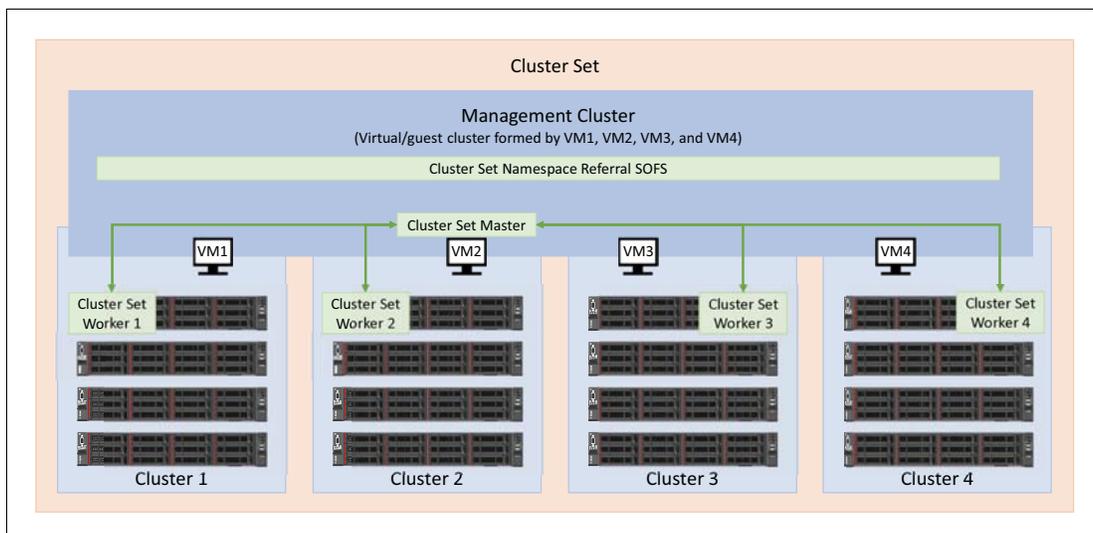


Figure 73 Cluster Set diagram showing VMs in management cluster straddling the member clusters

Member clusters are the traditional converged and hyperconverged failover clusters that have become common place in many corporate environments today. Multiple member clusters take

part in a single cluster set deployment, forming a larger cloud fabric. Although member clusters host VM and application workloads, these member cluster workloads should not be hosted by the management cluster.

## Create the cluster set

Here we provide the details to create a cluster set containing two member failover clusters. This results in a cluster set that includes a total of three clusters, including the management cluster.

1. On a system running Windows Server 2019 that will be used as a management system, install the Failover Cluster management tools using the command in Example 53. Note that only the management tools are installed, not the Failover Cluster feature itself.

*Example 53 Command to install the Failover Cluster management tools and PowerShell module*

---

```
Install-WindowsFeature RSAT-Clustering-Mgmt
```

---

1. Prepare two or more clusters that will become members of the cluster set. Each cluster must have at least two CSVs. In our examples, we use two clusters, S2D-Cluster1 and S2D-Cluster2.
2. Create a management cluster (physical or guest) that will be used to manage the cluster set. For best availability, this cluster should contain at least four nodes. In our examples, we use the cluster named MasterCluster.
3. We will create a new cluster set from these clusters. Table 15 shows the names of the management cluster and member clusters used in our examples. The name of the cluster set will be CS-Master. Note that the name of the cluster set is different from the name of the management cluster.

*Table 15 Names of clusters and cluster roles used in examples*

Cluster Name	Infrastructure SOFS Name	Description
MasterCluster	SOFS-ClusterSetMaster	Provides unified namespace of the management cluster for the cluster set
S2D-Cluster1	SOFS-S2D-Cluster1	Provides unified namespace of the member cluster, S2D-Cluster1
S2D-Cluster2	SOFS-S2D-Cluster2	Provides unified namespace of the member cluster, S2D-Cluster2

4. Create an Infrastructure SOFS role on the management cluster and on each of the clusters that will become members of the cluster set. Use the commands in Example 54 to facilitate this task. Each command is run individually on a node in the appropriate cluster, as shown in the comment following each command.

*Example 54 Commands to add an infrastructure SOFS role to each cluster*

---

```
Add-ClusterScaleoutFileServerRole -Name "SOFS-ClusterSetMaster" -Infrastructure # Run on MasterCluster
Add-ClusterScaleoutFileServerRole -Name "SOFS-S2D-Cluster1" -Infrastructure # Run on S2D-Cluster1
Add-ClusterScaleoutFileServerRole -Name "SOFS-S2D-Cluster2" -Infrastructure # Run on S2D-Cluster2
```

---

5. Use the command shown in Example 55 to create the cluster set, giving it a name that will be easily understood later. In our examples, we use CS-Master. Make sure to change the IP address to one suitable for your environment.

*Example 55 Command to create the cluster set master*

```
New-ClusterSet -Name CS-Master -StaticAddress 10.10.11.101 -NamespaceRoot S0FS-ClusterSetMaster -CimSession MasterCluster
```

6. To add a cluster as a member of the cluster set, use the commands shown in Example 56. Each command is run individually on a node in the appropriate cluster.

*Example 56 Command to add clusters to the cluster set*

```
Add-ClusterSetMember -ClusterName S2D-Cluster1 -CimSession CS-Master -InfraS0FSName S0FS-S2D-Cluster1  
Add-ClusterSetMember -ClusterName S2D-Cluster2 -CimSession CS-Master -InfraS0FSName S0FS-S2D-Cluster2
```

7. To configure Kerberos constrained delegation between all cluster set members, follow these steps on each node in each member cluster, but not the management cluster:
  - a. On the domain controller, open the Active Directory Users and Computers MMC.
  - b. Expand the Domain node to reveal the object types, and then select **Computers**.
  - c. Double-click the first computer (in our example, S2D-Node01) to open its Properties window.

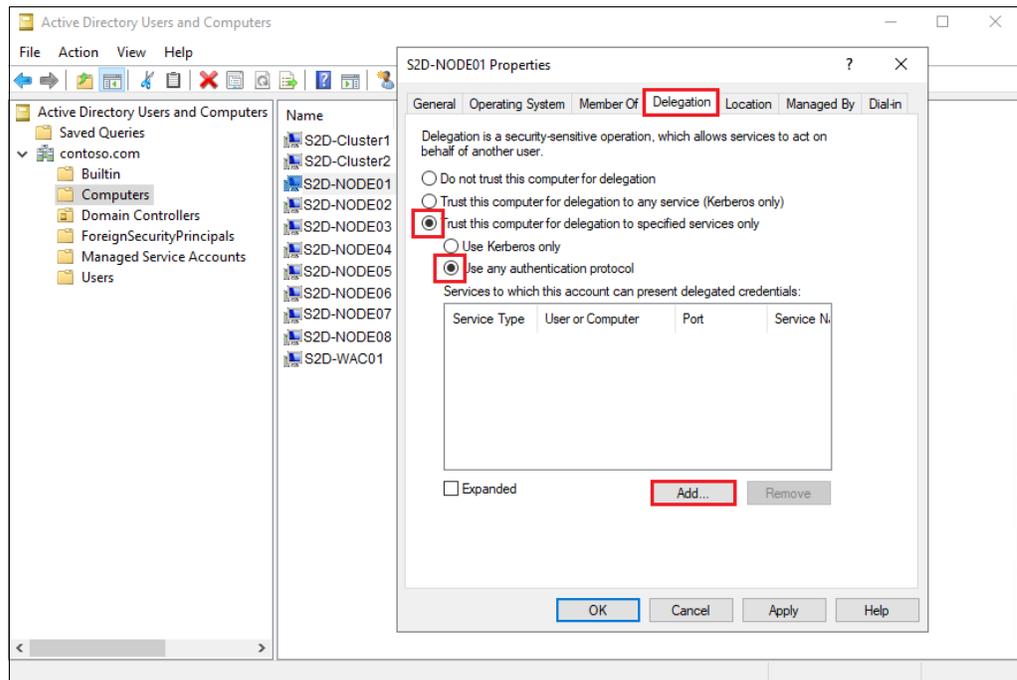


Figure 74 Active Director Users and Computers MMC showing Properties window

- d. On the Delegation tab, select the **Trust this computer for delegation to specified services only** radio button and the **Use any authentication protocol** radio button, before clicking **Add...**
- e. In the Add Services window that comes up, click **Users or Computers...**
- f. In the **Enter the object names to select** area, type the names of all nodes in all clusters that have been added to the cluster, excluding the management cluster. Separate the node names using a semicolon. Once all names are entered, click **OK**.

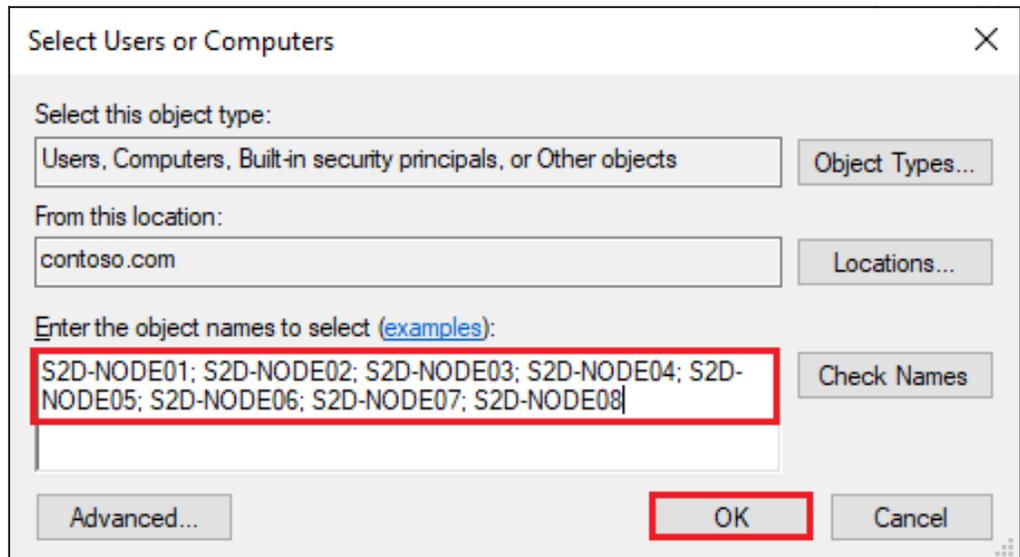


Figure 75 Select Users or Computers window

- g. Back in the Add Services window, scroll down to the first entry for “cifs” in the Service Type column. Select all the node names for the CIFS service, then scroll down to the first entry for “Microsoft Virtual System Migration Service” in the Service Type column and select all the node names for this service (use Ctrl-click to make sure you don’t lose the cifs selections above). Once both services for all nodes have been selected, click **OK**.

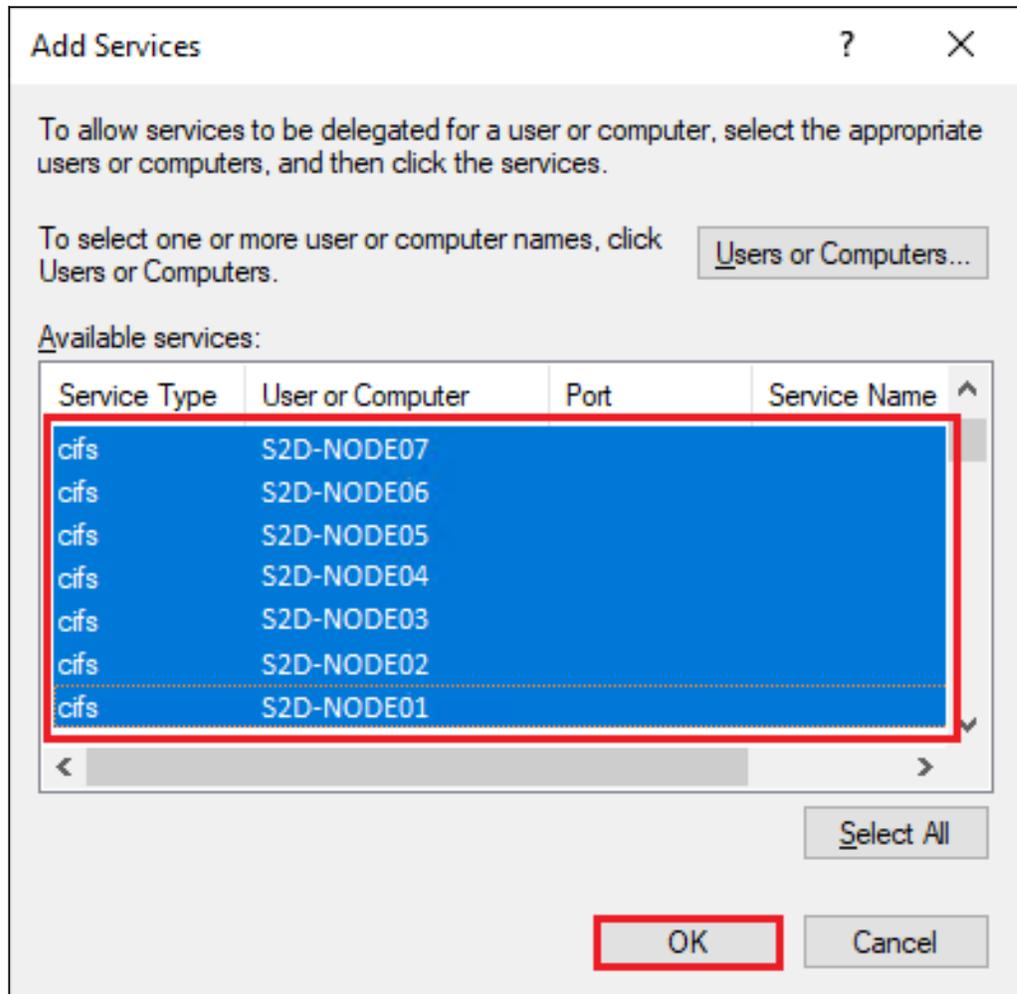


Figure 76 Add Services window with CIFS Service Type selected on all nodes (the Microsoft Virtual System Migration Service is also selected on all nodes, but is not visible in this figure)

- h. Verify that the two services for all nodes have been properly added to the Properties window, and then click **OK**.

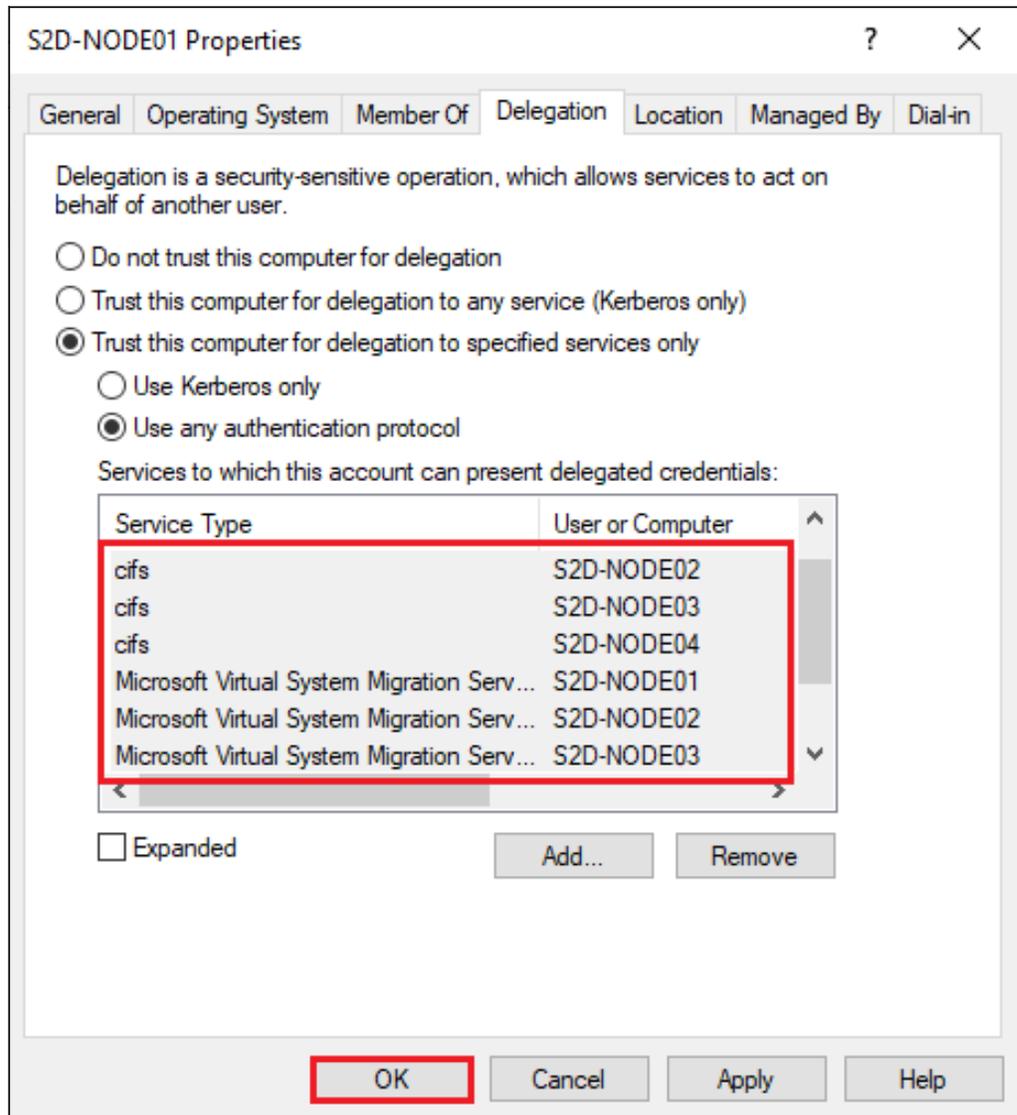


Figure 77 Computer Properties window showing Service Types ready to be added

- i. Repeat Steps c - h above for each node in each member cluster.
8. Configure the cross-cluster VM live migration authentication type to Kerberos on each node in the cluster set by using the command shown in Example 57. In this example, S2D nodes 1-4 are members of S2D-Cluster1 and S2D nodes 5-8 are members of S2D-Cluster2. Adjust the number and names of the nodes to suit your situation. Elevated privileges are required to execute this command.

*Example 57 Command to configure cross-cluster VM live migration authentication on each node in cluster set*

```
$Nodes = ("S2D-Node01", "S2D-Node02", "S2D-Node03", "S2D-Node04", "S2D-Node05", "S2D-Node06", `
"S2D-Node07", "S2D-Node08")
foreach($N in $Nodes){ Set-VMHost -VirtualMachineMigrationAuthenticationType Kerberos -ComputerName $N }
```

9. Add the management cluster to the local Administrators group on each node in the cluster set by using the command shown in Example 58. This command uses the same \$Nodes array variable as the command in Example 57 and also requires elevated privileges to execute.

*Example 58 Command to add the management cluster to local Administrators group on each node in the cluster set*

---

```
foreach($N in $Nodes){ Invoke-Command -ComputerName $N -ScriptBlock `
  {Net localgroup administrators /add MasterCluster$} }
```

---

10. Once you have created the cluster set, you can list the member clusters and their properties using the command shown in Example 59.

*Example 59 Command to list all member clusters in the cluster set*

---

```
Get-ClusterSetMember -CimSession CS-Master
```

---

11. To list all the member clusters in the cluster set including the management cluster nodes, use the command shown in Example 60.

*Example 60 Command to list all member clusters in the cluster set, including the management cluster nodes*

---

```
Get-ClusterSet -CimSession CS-Master | Get-Cluster | Get-ClusterNode
```

---

12. To list all the nodes from the member clusters, use the command shown in Example 61.

*Example 61 Command to list all nodes from the member clusters*

---

```
Get-ClusterSetNode -CimSession CS-Master
```

---

13. To list all the resource groups across the cluster set, use the command shown in Example 62.

*Example 62 Command to list all resource groups across the cluster set*

---

```
Get-ClusterSet -CimSession CS-Master | Get-Cluster | Get-ClusterGroup
```

---

14. To verify the cluster set creation process created one SMB share on the Infrastructure SOFS for each cluster member's CSV volume, use the command shown in Example 63.

*Example 63 Command to list the SMB shares on the infrastructure SOFS for each member cluster's CSV volume*

---

```
Get-SmbShare -CimSession CS-Master
```

---

15. Cluster sets has debug logs that can be collected for review. Both the cluster set and cluster debug logs can be gathered for all members and the management cluster by using the command shown in Example 64. Make sure to specify a valid path at the end of the command.

*Example 64 Command to gather logs from cluster set and member clusters, including the management cluster*

---

```
Get-ClusterSetLog -ClusterSetCimSession CS-Master -IncludeClusterLog -IncludeManagementClusterLog `
  -DestinationFolderPath <\\server\path>
```

---

## Summary

Windows Server 2016 introduced Storage Spaces Direct, which enables building highly available and scalable storage systems with local storage. This was a significant step forward in Microsoft Windows Server software-defined storage (SDS) as it simplified the deployment and management of SDS systems and also unlocked use of new classes of disk devices, such as SATA and NVMe disk devices, that were previously not possible with clustered Storage Spaces with shared disks.

With Windows Server 2019 Storage Spaces Direct, you can build highly available storage systems using Lenovo ThinkAgile MX Certified Nodes for Azure Stack HCI and Lenovo ThinkSystem rack servers with only local storage. This eliminates the need for a shared SAS fabric and its complexities, but also enables using devices such as SATA SSDs, which can help further reduce cost or NVMe SSDs to improve performance.

This document has provided an organized, stepwise process for deploying a Storage Spaces Direct solution based on Lenovo ThinkAgile MX Certified Nodes for Azure Stack HCI and Lenovo Ethernet switches. Multiple deployment scenarios have been addressed, including RoCEv2 and iWARP implementations of RDMA using Lenovo network switches, as well as 2-3 node direct-connected scenarios in which network switches are not used for storage traffic inside the Azure Stack HCI cluster. Once configured, this solution provides a versatile foundation for many different types of workloads.

## Lenovo Professional Services

Lenovo offers an extensive range of solutions, from the simple OS-only laden product to much more complex solutions running cluster and cloud technologies. For customers looking for assistance in the form of design, deploy or migrate, Lenovo Professional Services is your go-to partner.

Our worldwide team of IT Specialists and IT Architects can help customers scope and size the right solutions to meet their requirements, and then accelerate the implementation of the solution with our on-site and remote services. For customers also looking to elevate their own skill sets, our Technology Trainers can craft services that encompass solution deployment plus skills transfer, all in a single affordable package.

To inquire about our extensive service offerings and solicit information on how we can assist in your new Storage Spaces Direct implementation, please contact us at [x86svcs@lenovo.com](mailto:x86svcs@lenovo.com).

For more information about our service portfolio, please see our website:

<https://www3.lenovo.com/us/en/data-center/services/c/services?menu-id=services>

## Change history

### Changes in the August 2023 update:

- Updated the direct-connected scenarios (“RoCE: 2-4 nodes, direct-connected” on page 46 and “iWARP: 2-4 nodes, direct-connected” on page 74) to include guidelines for 4-node direct-connected cluster deployment.

**Changes in the March 2023 update:**

- ▶ Updated S2D resiliency discussions and added Table 1 on page 6 to summarize various resiliency characteristics.
- ▶ Updated instructions for iWARP configurations to use the Intel E810 network adapter instead of the Cavium/QLLogic QL41262 network adapter.
- ▶ Adjusted discussions of “LOM ports” to distinguish them from OCP network adapters that are available for Lenovo ThinkSystem SR630 V2 and SR650 V2 servers.
- ▶ Corrected command in Example 42 on page 84 (removed -Name parameter from the command to create the cluster).

**Changes in the April 2022 update:**

- ▶ Added details pertaining to Azure Stack HCI operating system, including differentiation from Windows Server operating systems.
- ▶ Added description and reference URL for a new file share witness feature in Windows Server 2019 and later, including HCI OSes.
- ▶ Removed all references to Windows Server 2016.
- ▶ Removed the *Solution performance optimization* section, since VMQ optimization occurs automatically in current Windows Server and HCI OSes.
- ▶ Removed descriptions for Lenovo network switches, since they are no longer available.
- ▶ Corrected various typos.

**Changes in the August 2020 update:**

- ▶ Corrected the command shown in Example 45 on page 87

**Changes in the July 2020 update:**

- ▶ Added discussion of full mesh dual-link connectivity model for 2 and 3-node direct-connected scenarios
- ▶ Added discussion of LAN over USB (NDIS) network interface and how to disable it safely
- ▶ Added discussion and steps for how to disable unneeded LOM ports in UEFI to avoid issues with cluster validation and creation
- ▶ Added reference to ThinkAgile MX1021 on SE350 Deployment Guide
- ▶ Corrected PowerShell commands in Examples 32, 35, and 46

**Changes in the April 2020 update:**

- ▶ Added guidance for three-node direct-connected scenarios for achieving redundant high-speed connectivity between cluster nodes
  - Updated graphics for all direct-connected scenarios to make network connections more clear
  - Added point-to-point network cable map table that specifies source and destination ports, as well as subnetting for 3-node direct-connected clusters using full-mesh connectivity for East-West traffic
- ▶ Updated several graphics and examples throughout the document for clarity
- ▶ Updated and corrected “Create failover cluster” on page 84 to include specific guidance for VMQ configuration in different deployment scenarios

**Changes in the November 2019 update:**

- ▶ Completely reworked the document, adding deployment scenarios for:

- RoCE: 2-16 nodes with network switches
  - Using one dual-port Mellanox adapter in each server
  - Using two dual-port Mellanox adapters in each server
- RoCE: 2-3 nodes, direct-connected
- iWARP: 2-16 nodes with network switches
  - Using one dual-port Cavium/QLogic adapter in each server
  - Using two dual-port Cavium/QLogic adapters in each server
- iWARP: 2-3 nodes, direct-connected
- ▶ Updated all Lenovo network switch commands to CNOS v10.10.x syntax
- ▶ Added solution performance optimization section
- ▶ Added Cluster Set creation section
- ▶ Added several best practice statements

#### **Changes in the May 2019 update:**

- ▶ Added information regarding Lenovo ThinkAgile MX Certified Nodes for S2D
- ▶ Added relevant information for Windows Server 2019
- ▶ Added information about the Microsoft Azure Stack HCI program for certification
- ▶ Corrected several typos
- ▶ Added missing `vlag enable` command to Example 9 on page 31

#### **Changes in the 14 May 2018 update:**

- ▶ Updated to include the latest Lenovo ThinkSystem rack servers
- ▶ Updated to include the latest Lenovo ThinkSystem RackSwitch products
- ▶ Switch configuration commands updated for CNOS
- ▶ Added vLAG to ISL between switches
- ▶ Added switch configuration commands to support Jumbo Frames
- ▶ Added affinization of virtual NICs to physical NICs

#### **Changes in the 9 January 2017 update:**

- ▶ Added detail regarding solution configuration if using Chelsio NICs
- ▶ Added PowerShell commands for IP address assignment
- ▶ Moved network interface disablement section to make more logical sense
- ▶ Updated Figure 2 on page 6 and Figure 3 on page 7
- ▶ Fixed reference to Intel v3 processors in Figure 20 on page 26
- ▶ Updated cluster network rename section and figure
- ▶ Removed Bill of Materials for converged solution

#### **Changes in the 16 September 2016 update:**

- ▶ Updated process based on Windows Server 2016 RTM
- ▶ Added background detail around Microsoft S2D
- ▶ Added driver details for Mellanox ConnectX-4 Lx adapter
- ▶ Added notes specific to hyperconverged vs. converged deployment
- ▶ Removed GUI-based Failover Cluster configuration steps (use PowerShell!)
- ▶ Added step to ensure both cluster networks are available for SMB traffic to clients
- ▶ Fixed issues with a couple of graphics
- ▶ Updated both BOMs: the servers now use Intel Xeon E5 2600 v4 processors

#### **Changes in the 14 July 2016 update:**

- ▶ Configuration process reordered for efficiency
- ▶ Added steps to configure VMQ queues

- ▶ Updated and added graphics
- ▶ Added various PowerShell cmdlets to aid in configuration
- ▶ Fixed various typos

**Changes in the 3 June 2016 update:**

- ▶ Updated to list setup instructions using Windows Server 2016 TP5
- ▶ Added DCB settings for each host
- ▶ Updated the Bills of Material

## Authors

This paper was produced by the following team of specialists:

**Dave Feisthammel** is a Senior Solutions Architect working at the Lenovo Center for Microsoft Technologies in Bellevue, Washington. He has over 25 years of experience in the IT field, including four years as an IBM client and 14 years working for IBM. His areas of expertise include Windows Server and systems management, as well as virtualization, storage, and cloud technologies. He is currently a key contributor to Lenovo solutions related to Microsoft Azure Stack Hub and Azure Stack HCI (S2D).

**Mike Miller** is a Windows Engineer with the Lenovo Server Lab in Bellevue, Washington. He has over 35 years in the IT industry, primarily in client/server support and development roles. The last 13 years have been focused on Windows Server operating systems and server-level hardware, particularly on operating system/hardware compatibility, advanced Windows features, and Windows test functions.

**David Ye** is a Principal Solutions Architect at Lenovo with over 25 years of experience in the IT field. He started his career at IBM as a Worldwide Windows Level 3 Support Engineer. In this role, he helped customers solve complex problems and critical issues. He is now working in the Lenovo Infrastructure Solutions Group, where he works with customers on Proof of Concept designs, solution sizing and reviews, and performance optimization. His areas of expertise are Windows Server, SAN Storage, Virtualization and Cloud, and Microsoft Exchange Server. He is currently leading the effort in Microsoft Azure Stack HCI and Azure Stack Hub solutions development.

Thanks to the following Lenovo colleagues for their contributions to this project:

- ▶ Daniel Ghidali, Manager - Microsoft Technology and Enablement
- ▶ Hussein Jammal, Senior Solutions Architect and Microsoft Solution Lead, EMEA
- ▶ Vinay Kulkarni, Lead Architect - Microsoft Solutions and Enablement
- ▶ Per Ljungstrom, Cloud Solutions Lead, EMEA
- ▶ Vy Phan, Technical Program Manager - Microsoft Technology and Enablement
- ▶ David Watts, Senior IT Consultant - Lenovo Press

At Lenovo Press, we bring together experts to produce technical publications around topics of importance to you, providing information and best practices for using Lenovo products and solutions to solve IT challenges.

See a list of our most recent publications at the Lenovo Press web site:

<http://lenovopress.com>

## Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.  
1009 Think Place - Building One  
Morrisville, NC 27560  
U.S.A.  
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document LP0064 was created or updated on August 2, 2023.

Send us your comments via the **Rate & Provide Feedback** form found at <http://lenovopress.com/lp0064>

## Trademarks

Lenovo, the Lenovo logo, and For Those Who Do are trademarks or registered trademarks of Lenovo in the United States, other countries, or both. These and other Lenovo trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by Lenovo at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of Lenovo trademarks is available on the Web at <http://www.lenovo.com/legal/copytrade.html>.

The following terms are trademarks of Lenovo in the United States, other countries, or both:

Lenovo®	Lenovo(logo)®	vNIC™
Lenovo XClarity™	ThinkAgile™	
RackSwitch™	ThinkSystem™	

The following terms are trademarks of other companies:

Intel, Xeon, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Active Directory, Azure, Hyper-V, Microsoft, PowerShell, SQL Server, Windows, Windows Server, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.