2017

Understand Microsoft Hyper Converged solution

MVP CLOUD AND DATACENTER MANAGEMENT

ROMAIN SERRE

CHARBEL NEMNOM

Credit page

Romain Serre (author)

MVP Profile

Romain Serre works in Lyon as a Technical Architect. He is focused on Microsoft Technology, especially on Hyper-V, System Center, Storage, networking and Cloud OS technology as Microsoft Azure or Azure Stack.

He is certified Microsoft Certified Solution Expert (Server Infrastructure & Private Cloud), on Hyper-V and on Microsoft Azure (Implementing a Microsoft Azure Solution).

Romain blogs in all of these technologies at http://www.tech-coffee.net

Charbel Nemnom (author)

MVP Profile

Technical evangelist, totally a fan of the latest IT platform solutions. Accomplished hands-on technical professional with over 15 years of broad IT project management and Infrastructure experience, serving on and guiding technical teams to optimize performance of mission-critical enterprise systems. Excellent communicator, adept at identifying business needs and bridging the gap between functional groups and technology to foster targeted, and innovative systems development. Well respected by peers through demonstrating passion for technology and performance improvement. Extensive practical knowledge of complex systems builds, network design and virtualization.

Charbel has extensive experience in various systems, focusing on Microsoft CloudOS Platform, Datacenter Management, Hyper-V, Microsoft Azure, security, disaster recovery, and many types of monitoring tools as well. He has a solid knowledge of technical reporting.

If you want to learn more about Charbel's activities, you are invited to visit his blog at https://www.charbelnemnom.com

Credit page 1
Introduction 4
From disaggregated systems to Hyper-Convergence 4
Microsoft disaggregated solution overview 5
Microsoft Hyper Converged solution overview 6
Software-Defined Networking 8
Network required 8
Network Convergence 8
Switch Embedded Teaming 10
VMQueue and Receive Side Scaling 11
Datacenter Bridging 12
Remote Direct Memory Access 12
SMB Direct 13
Fabric design recommendation 13
Software-Defined Compute 14
Hyper-V 14
Live-Migration 16
Hyper-V Replica 17
Software-Defined Storage 18
Storage Pool and Storage Spaces 18
Storage Spaces Direct 19
Storage Spaces resiliency 21
Fault Domain Awareness 23
Multi-Resiliency Virtual Disks and ReFS Real-Time Tiering 24
Three tiers configuration 25
ReFS file system 26
Health Service 26
Storage Replica 28
Storage Quality of Service 29
Windows Server 2016 licensing 31
Implementation guide 33
Design overview 33
Hardware consideration 33
Network design 34
Logical design 35
Operating system configuration 35

Bios configuration 35
OS first settings 36
Network settings 37
Connect to Hyper-V remotely 41
Add nodes to domain 43
2-node hyper-converged cluster deployment 44
Final Hyper-V configuration 46
Deploy a benchmark tool 48
Cluster preparation for VM Fleet 48
Prepare the Gold image 51
Deploy the VM Fleet 53
Play with the VM Fleet 54
Run a test 55
Troubleshooting 57
Work with Health Service 57
Detect and change a failed physical disk 57
Identify the failed physical disk 57
Retire and physically identify storage device 58
Add physical disk to storage pool 60
Patch management 61
Prepare Active Directory 61
Configure Self-Updating 62
Validate the CAU configuration 68
Run manual updates to the cluster 69
Conclusion 75
References 76

Introduction

From disaggregated systems to Hyper-Convergence

Before virtualization was deployed in companies, we installed every single application on a specific physical server. Most of the time, a physical server hosted a single application. Usually this application was installed on local storage. Because one application was installed on each server, the resource on this physical server was largely unused (except in the case of big applications). For example, DHCP and WDS were installed on the same physical server and consumed few resources. Moreover, each physical server had multiple Network Interface Controllers (NICs) but each NIC was dedicated for a specific traffic. For example, some NICs were dedicated to Cluster Heartbeat, which used 1% of the bandwidth.

To avoid wasting resources like CPU, RAM, storage, and the power usage as well, we used virtualization in the datacenter. Thanks to this technology, we could run multiple servers called Virtual Machines (VMs) on a single physical server (the Hypervisor). We installed a single application in each VM and so we could run multiple applications on a single physical server (host). Moreover, because each VM runs its own Operating System, we had a boundary between each VM and so between each application.

To run more Virtual Machines and to support High Availability we installed hypervisor in clustered mode. So, we could move VM from one cluster node to another one without any downtime (operation known as Live Migration). But hypervisor cluster required shared storage across all nodes to access the same VMs data store. Therefore, Network Attached Storage (NAS), or Storage Area Network (SAN) appliances were used to offer shared storage. Usually SAN was used, which brought new challenges.

SAN provides good performance and good resiliency, but brings some difficulties. Firstly, SAN is a complex solution. It needs specific knowledge as zoning, masking, LUNs, Raid Group and so on. Secondly, several types of protocols can connect to a SAN such as iSCSI, Fibre Channel (FC) or FC over Ethernet (FCoE). Depending on the chosen solution or vendor, the price can change as well as the exploitation difficulties. For example, Fibre Channel, which offers the best performance, requires specific switches (SAN Switches), SFP module, fiber optic and specific peripherals on physical servers (called Host Bus Adapter or HBA). Moreover, when you add a new hardware (either SAN enclosures or physical servers), you must check if the device firmware is compatible with the other devices. Therefore, this solution is not flexible and is also expensive at the same time.

For all the above reasons, since past few years, multiple software vendors have tried to bring **Software-Defined Datacenter** (SDD) solution in the datacenter. With SDD, we will use software instead of using hardware devices which are usually not flexible. Software is flexible and scalable. A single update can bring new features.

First, we installed **Software-Define Compute,** which is in fact the hypervisor or the hypervisor cluster. There are several hypervisors in the market today such as ESX (VMWare), Hyper-V (Microsoft) or XenServer (Citrix).

Next, we should create virtual switches to interconnect virtual machines. Then, to avoid network bandwidth waste, the **Network Convergence** was introduced. Thanks to the Network Convergence, we could carry several traffics though a single NIC or a teaming. Moreover, we can now use network virtualization (using VXLAN or NVGRE protocols) to segregate subnet by software without using VLANs. All of this is called **Software-Defined Networking**.

More recently, we wanted to avoid using SAN because it is too expensive, complex, and not scalable. Thus, some companies created software storage solutions such as vSAN (VMWare Virtual SAN) or Storage Spaces (Microsoft). This is called **Software-Defined Storage**.

However, some companies faced a scalability problem between storage and hypervisor. Because storage is shared, the more you add hypervisors more the available Input/output per Second (IOPS) are divided among them. Therefore, you have to add devices in your storage solution. The solution was scalable but needs a careful balance between compute and storage resources.

Hyper-Convergence was introduced into the market by several vendors to tackle all those challenges. To resolve this, we put the storage in the same physical host with the hypervisor (local direct attached storage or DAS). Therefore, even if you add a hypervisor, you add storage too. Therefore, it is flexible and scalable, especially for On-Premise Cloud solution. Moreover, because almost everything is in the physical server, hyper converged solution uses little spaces in the datacenter.

Welcome to Hyper-Converged world!

Microsoft disaggregated solution overview

Before talking about Hyper Converged solution, it is important to understand disaggregated solution that we are building in Windows Server 2012 R2.

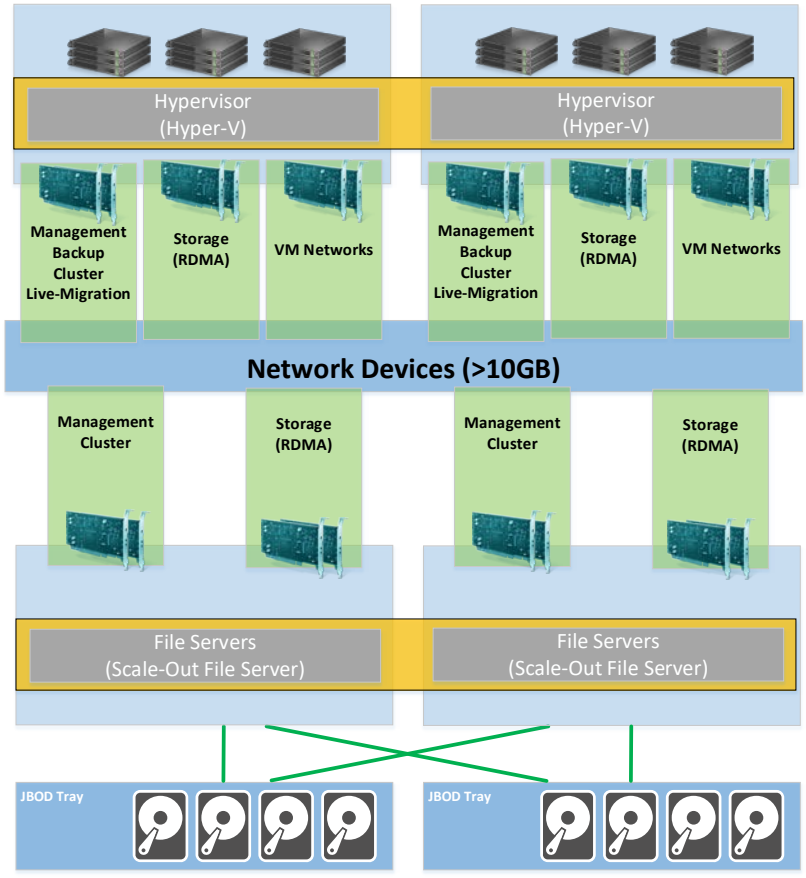


Figure 1: Microsoft disaggregated solution

In Microsoft, disaggregated solution, we use Software-Defined Storage (clustered Storage Spaces). In Windows Server 2012 R2, all storage devices must be visible to each File Server cluster node to create a clustered Storage Spaces. Therefore, Shared JBOD trays are usually needed. They are connected using SAS Cables to two or more File Server clustered (Up to 8) with the Scale-Out File Server role installed.

Regarding the networks, the current NIC teaming technology does not allow us to converge Storage, Live-Migration and other networks (C.F Software-Define networking part). Therefore, the networks cannot be fully converged.

Microsoft Hyper Converged solution overview

Several companies have released their own Hyper Converged solution such as Nutanix, GridStore or Simplivity. However, in this whitepaper we will focus on the Microsoft Hyper Converged solution that is available with the release of Windows Server 2016.

The Microsoft Hyper Converged solution must be composed of minimum two nodes (upon to 16 nodes). However, in two nodes configuration some features are not available (such as Multi-Resilient Virtual Disk known as MRV). Each node must be identical (same CPU, RAM, Storage Controllers, Network Adapters and so on). Each node has its own local storage, either SAS JBOD enclosure attached locally. Disks can be SAS, SATA or NVMe (SSD connected to PCI-Express). In each node, the cluster feature is installed as well as Hyper-V.

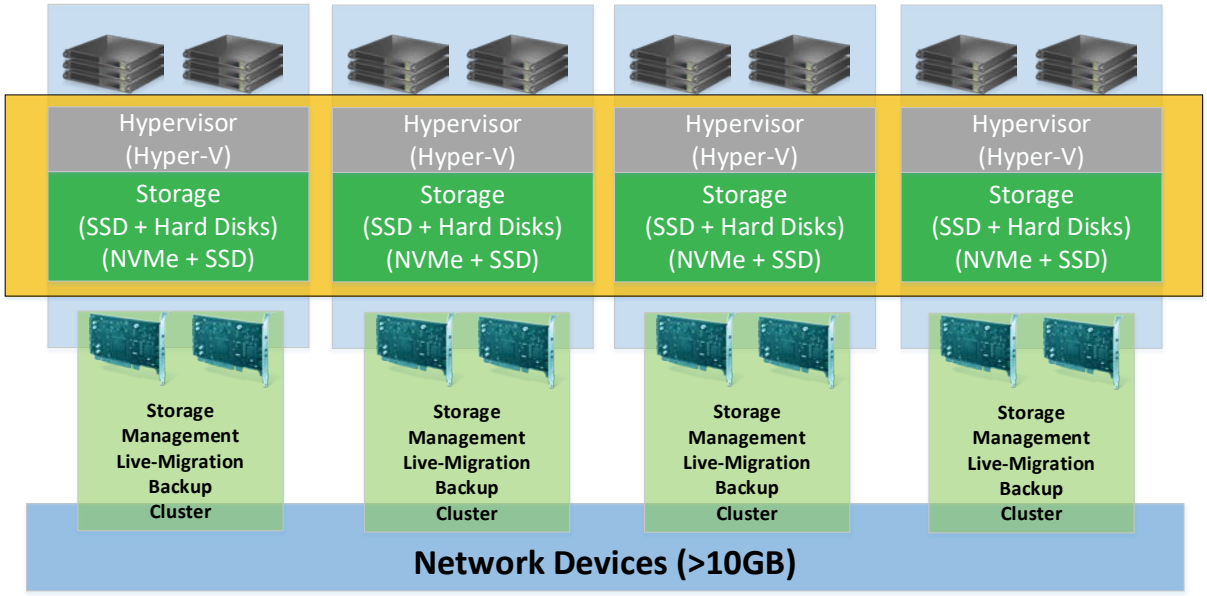


Figure 2: Microsoft Hyper Convergence solution overview

Regarding the CPU configuration, Microsoft expects at least a dual socket with Intel Xeon E3 per node. The CPU consumption is higher with Hyper Converged solution than disaggregated systems because in this case, the CPU is used for virtual machines and for storage processing as well. Therefore, more processor cores are needed and a high frequency is recommended.

Relating to the CPU, you should define the amount of memories per node. Microsoft recommends at least 128GB per node for small deployments. Be careful do not install too many memories related to the number of threads available to avoid wasting memory. Moreover, because it is a cluster, you have to calculate with N-1 factor to take into consideration the loss of one node. Example:

- You have four nodes of 128GB of RAM (512GB of memory in the cluster)
- Only 384GB of RAM should be assigned for VMs; if a node is down, all VMs can keep running

On the storage side, you also have some prerequisites. To implement the Microsoft Hyper Converged solution, we will use the new Windows Server 2016 feature called **Storage Spaces Direct (known as S2D)**.

Firstly, each node must have the same total number of storage devices. Secondly, in the case of a mix of type of disks (HDD, SSD or NVMe), each node must have the same number of disks for each type.

If you use a mix type of storage devices, then each node must have at least two performance storage devices and four capacity drives. In the example below, you can find the minimum configuration required:

Solution	Disks	Disks type	Numbers
(SATA SSD + SATA HDD)	SATA SSD	Performances	2
	SATA HDD	Capacity	4
(NVMe SSD + SATA HDD)	NVMe SSD	Performances	2
	SATA HDD	Capacity	4
(NVMe SSD + SATA SSD)	NVMe SSD	Performances	2
	SATA SSD	Capacity	4

The performance disks are used for Write-Back caching mechanism (for more information, read Storage Spaces Direct section).

If you use only a single type of flash storage devices either full NVMe or Full SSD (not both), you can disable the caching mechanism.

Regarding networking, there are other requirements too. Storage Spaces Direct requires at least 10GB Ethernet adapter per node, RDMA capability NICs such as (RoCE or iWARP) are preferable and recommended but not required. Nevertheless, to avoid single point of failure, we recommend using at least two 10GB Ethernet adapters. If you implement network convergence, we recommend also to install network adapters faster than 10GB for better efficiency. Finally, if you choose RoCE, the physical switches should support RoCE, DataCenter Bridging (DCB) and Priority Flow Control (PFC).

As mentioned earlier, Storage Spaces Direct can be implemented in a 2-nodes configuration. In storage part, we will talk about mirroring, parity, and multi-resilient virtual disks. Some of these configurations are not available in 2-nodes and 3-nodes configuration:

	Mirroring	Parity	Multi-Resilient
Optimized for	Performance	Efficiency	Balanced between performance and efficiency
Use case	All data is hot	All data is cold	Mix of hot and cold data
Efficiency	Least (33%)	Most (50%)	Medium (~50%)
File System	ReFS or NTFS	ReFS or NTFS	ReFS only
Minimum nodes	2+	4+	4+

As you can see in the above table, you cannot implement parity and multi-resilient virtual disk in 2-nodes and 3-nodes cluster. Moreover in 2-node configuration you can implement only 2-Way Mirroring.

In the next sections of this document, we will detail each component (Network, Compute and Storage) in order to implement a Microsoft Hyper-Converged solution.

Software-Defined Networking

In this section, we will describe the network designs and the key features needed by the Hyper Converged solution.

Network required

This kind of solution requires networks for the cluster, Hyper-V, Storage and the management. So, to implement the Hyper Converged solution we need the following:

- **Management (routed)**: This network is used to carry these kinds of traffics: Active Directory, RDS, DNS and so on. Regarding the Hyper-V side, we will create the management NIC on this network and fabric VM will be connected on this network. A virtual NIC will be created on the Hyper-V parent partition.
- **Cluster (private)**: This network carries the cluster heartbeat and SMB 3.0 for all intra-node (also called east-west) communication, and takes advantage of all the powerful features of SMB 3.0, such as SMB Direct (RDMA-enabled NICs) for high bandwidth and low latency communication, and SMB Multichannel for bandwidth aggregation and network fault tolerance. In other words, this network carries also Live-Migration and Storage traffics. You can dedicate virtual network adapters for cluster, storage and live-migration or converged these traffics into two virtual network adapters.
- **VM Networks (routed)**: To interconnect VMs other than fabric VM, VM Networks are required. This can be network isolated by VLAN, NVGRE network (Provider Network) and so on. It depends on your needs.

Network Convergence

To limit the number of NICs installed on each node and to avoid the bandwidth waste, we will use network convergence. Depending on the budget and the chosen NICs, several designs are possible. You can for example, install 10GB Ethernet NICs for Hyper-V host and storage, and 1GB Ethernet NICs for VM traffic, or you can also buy two 40GB NICs and converge all the traffics. It is up to you.

In the below example, you can find three distinctive designs with their own advantages. For each solution, “network devices” expression means at least two switches to support the High Availability.

As showing in the design – Figure 3, there are four NICs per node: two 1GB Ethernet Controllers and two 10GB Ethernet Controllers that are RDMA capable. The two 1GB NICs, in a teaming, are used to carry VM Networks and the 10GB, in a teaming, are used for the storage, Live-Migration, Cluster and management traffics. This solution is great when you have not a lot of VM Networks and when these VMs don’t require a lot of bandwidth. 1GB NICs are cheap and so if you want to segregate fabric and VM traffics on different NICs, it is the less expensive solution. It's also a less scalable solution because if you reach the bandwidth limit, you have to add 1GB NICs in the teaming, up to 8 with Switch Embedded Teaming (please refer to the Switch Embedded Teaming section for more detail), or buy 10GB network cards.

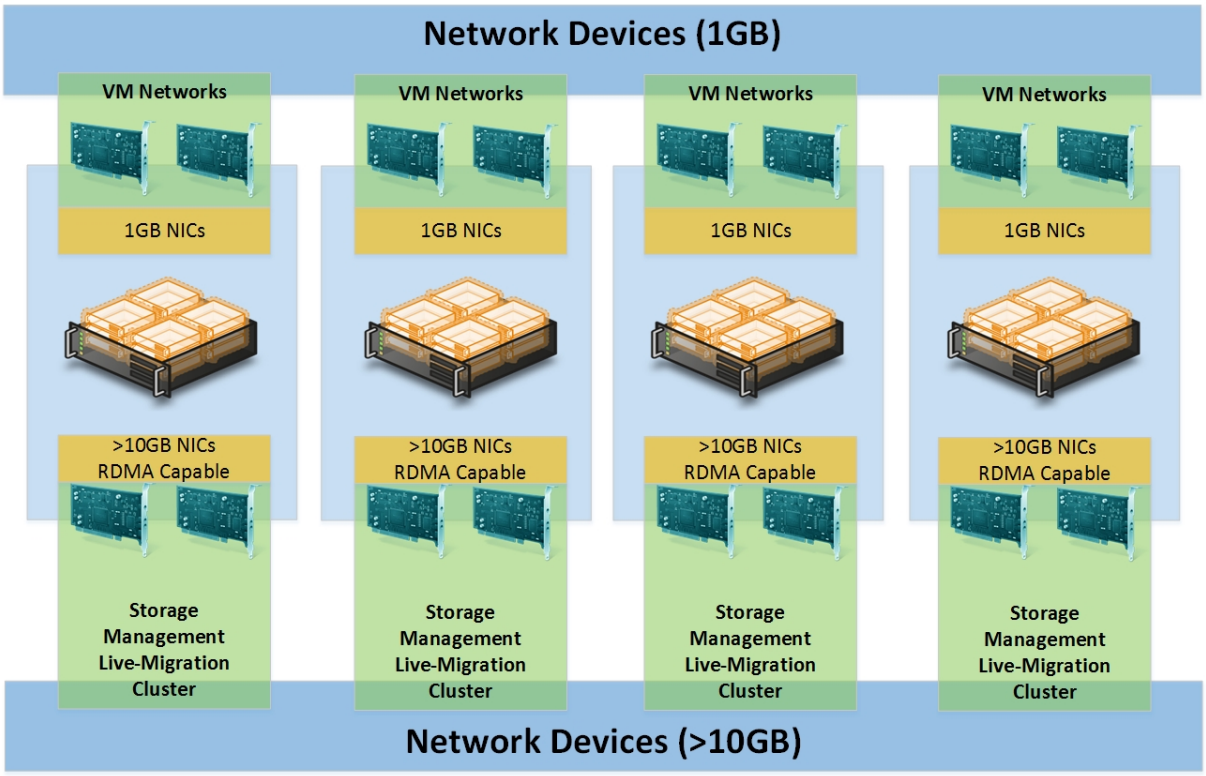


Figure 3 : 1GB for VMs and 10GB for fabric Ethernet controllers

The second solution – Figure 5, is the same as above, but with 10GB NICs for VM Networks. The fabric and VM traffics are segregated on different NICs and the solution is more scalable than the one above. If you reach the bandwidth limit you can add 10GB NIC in the teaming (up to 8 with Switch Embedded Teaming).

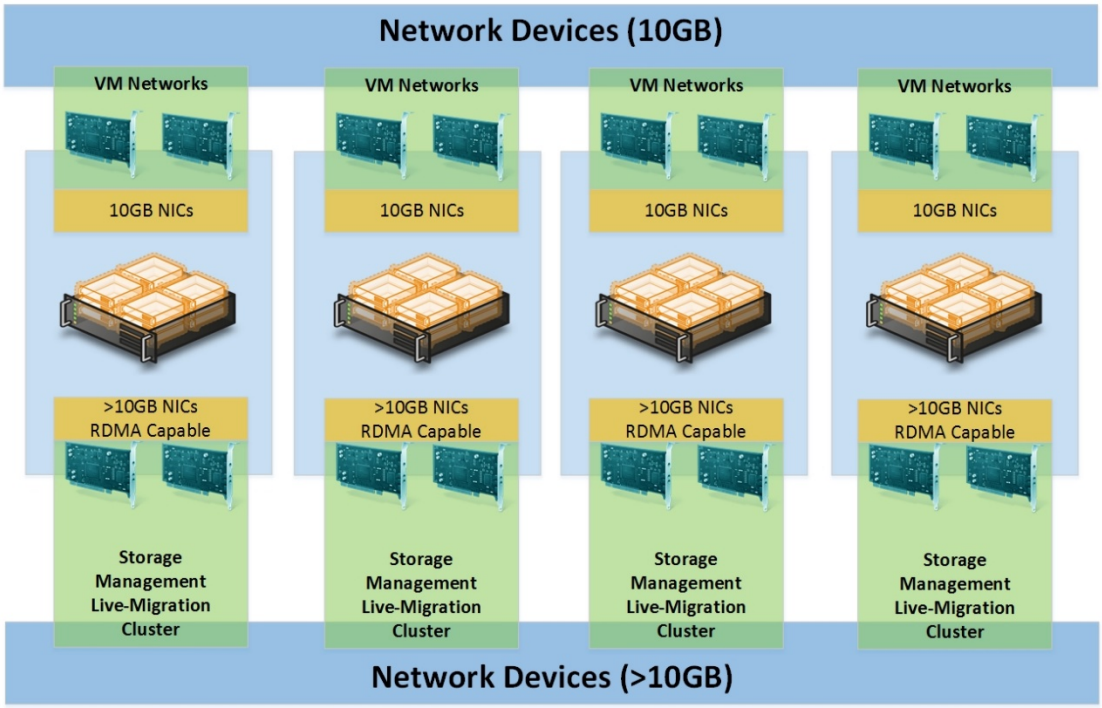


Figure 4 : 10GB Ethernet Controllers for VMs and Fabric traffics

If you want a fully converged network (after all it is Hyper Convergence), you can converge all networks in a single teaming. For this kind of solution, we recommend you use at least 25GB Ethernet controllers to support all the traffic (especially if you use SSD as capacity drive…). With this kind of solution, you need just two NICs (plus one for the Baseboard Management Controller) and to support the High Availability, at least two top of rack switches (TOR). It simplifies the cabling management in the datacenter racks. However, a good QoS management is mandatory to leave enough bandwidth for each type of traffic.

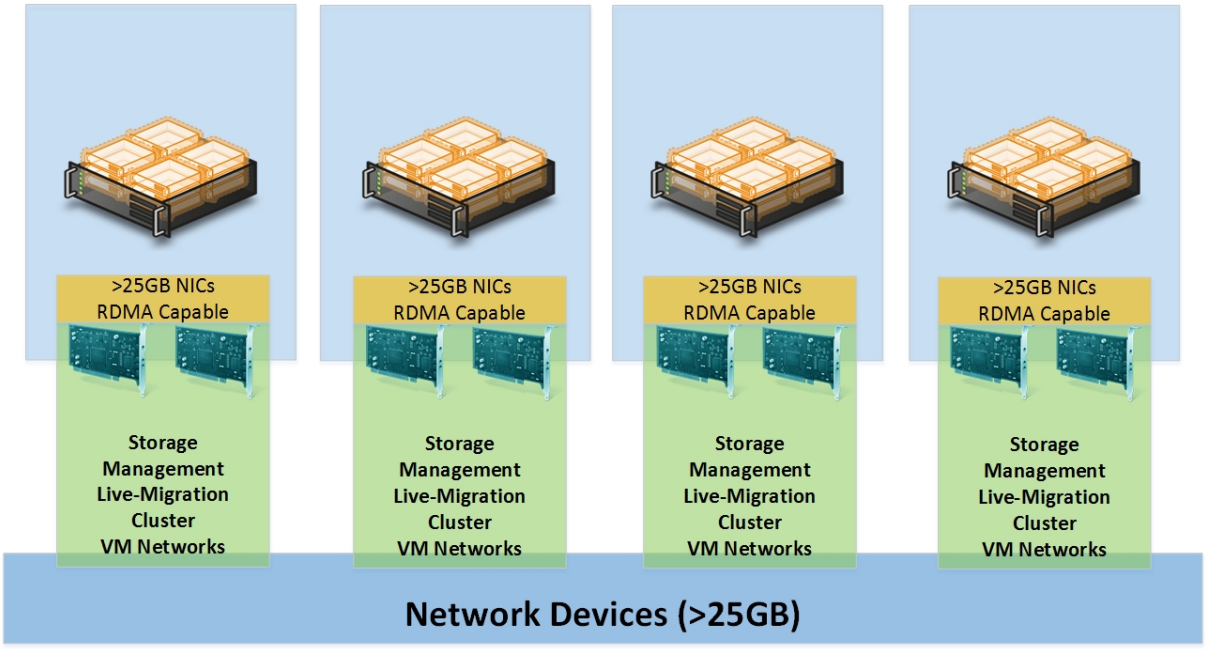


Figure 5 : Converge fabric and VM workloads

Switch Embedded Teaming

With Windows Server 2012 / 2012 R2 we deployed teaming by using the standard network teaming (known as LBFO teaming) either by using LACP or switch independent mode. When the teaming was created, usually we deploy virtual NICs (vNICs) in the Hyper-V parent partition for management, Live-Migration, Cluster and so on. The underlying vNICs deployed in the parent partition didn't support some required features such as vRSS, VMQueue and RDMA. It was a problem to converge storage network as well as the Live-Migration Network.

In Windows Server 2016, we can use a new teaming method called Switch Embedded Teaming (SET). SET is an improvement of virtual switch because thanks to this technology, the teaming is embedded inside the Hyper-V virtual switch. You don't need any more to create a teaming before the adding of a virtual switch. Moreover, SET enables us to create vNICs in the parent partition, which supports the following technologies:

- Datacenter bridging (DCB)
- Hyper-V Network Virtualization (HNVv2) – NVGRE and VxLAN are both supported in Windows Server 2016.
- Receive-side Checksum offloads (IPv4, IPv6, TCP) – These are supported if any of the SET team members support them.
- Remote Direct Memory Access (RDMA)
- SDN Quality of Service (QoS)

- Transmit-side Checksum offloads (IPv4, IPv6, TCP) – These are supported if all the SET team members support them.
- Virtual Machine Queues (VMQ)
- Virtual Machine Multi-Queue (VMMQ)
- Virtual Receive Side Scaling (vRSS)

Thanks to SET, we have the key technologies needed to converge networks for Storage Spaces Direct. The below schema illustrates SET deployment with S2D:

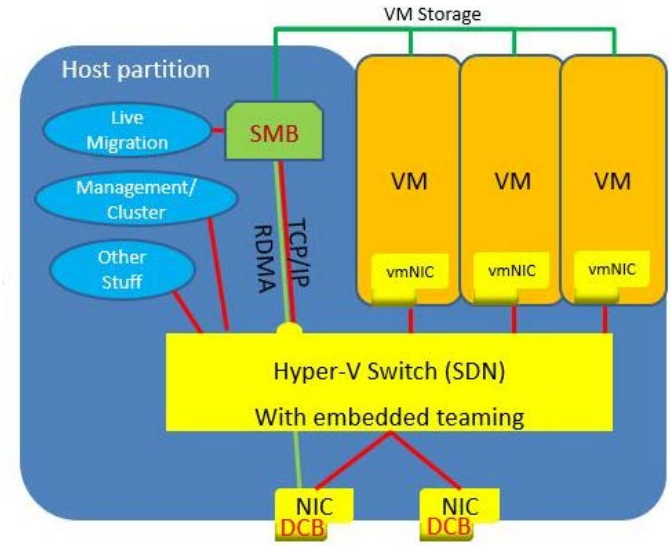


Figure 6 : Switch Embedded Teaming (illustration taken from TechNet)

As of this writing, SET supports only the Switch Independent mode and no other types such as LACP. Two Load-Balancing algorithms are supported: **Hyper-V Port** and **Dynamic** (recommended). Therefore, the total number of VM queues in the team is the sum of the VM Queues of each NIC in the team. It is also called the **Sum-Queues**.

VMQueue and Receive Side Scaling

Receive Side Scaling (RSS) is a feature that enables to spread the network I/O processing across several physical processors. Without RSS, all network I/O will go to the processor 0. In other words, you couldn't exceed almost 3,5Gb/s. It is problematic when you have bought 10GB Ethernet NIC. The same technology exists inside the VM and it is called vRSS.

However, in virtualization world, the physical server hosts several virtual machines and now we need a virtual network queue for each Virtual Machine to give packets to VM directly. It is called a **VM Queue**. When you disable VM Queue all traffic is managed by the Core 0. So, a bottleneck can be created when using 10GB Ethernet, because one processor core can give you around 3.5Gbps of throughput (please disable VMQueue on 1GB Ethernet NIC). You can also enable static VMQueue which enables to associate a VMQueue to a processor by using a Round Robin algorithm. However, in Windows Server 2012 R2 Microsoft introduced dynamic VMQueue, which enables to spread the VMQueue across several processors dynamically based on processor usage. It's enable to optimize processor utilization.

Until now, we can't converge network in the Hyper-V parent partition if we want to use RSS and VMQueue. It is because vRSS and VMQueue are not supported in virtual NIC created in the parent

partition. With Switch Embedded Teaming, we will be able to create vNIC in the parent partition with vRSS and VMQueue support.

In Windows Server 2016, Microsoft introduced Virtual Machine Multi-Queue (VMMQ) which is a NIC feature that allows traffic for a VM to be spread across multiple queues, each processed by a different physical processor. The traffic is then passed to multiple Logical Processors (LPs) in the VM as it would be in vRSS. This allows for very large networking bandwidth to be delivered to the VM.

VMMQ is the evolution of VMQ with Software vRSS. Whereas in VMQ the NIC Switch vPort is mapped to a single queue, VMMQ assigns multiple queues to the same vPort. RSS is used to spread the incoming traffic between the queues assigned to the vPort. The result is effectively a hardware offload version of vRSS.

As a guidance: Assign one VMMQ queue for a VM for every 3-4 Gbps of incoming networking traffic that a VM requires. If you have more VMs than vPorts, assign one queue to the default queue for every 2 Gbps of aggregate bandwidth that the VMs sharing the default queue will require. (Some NICs that support VMMQ may only be able to support the same number of queues on every vPort. This may require some testing to find the right balance between load spreading and interrupt processing.)

- To manage VMMQ Queues for VMs, use the `Set-VMNetworkAdapter` PowerShell cmdlet with the `–VmmqEnabled` and `-VmmqQueuePairs` parameters. (A queue is technically a queue pair, send and receive.)
- To manage the number of queues assigned to the default vPort use the `Set-VMSwitch` PowerShell cmdlet with the `-DefaultQueueVmmqEnabled` and `-DefaultQueueVmmqQueuePairs` parameters.

More information on the `Set-VMNetworkAdapter` cmdlet can be found at https://technet.microsoft.com/en-us/library/hh848457.aspx.

More information on the `Set-VMSwitch` cmdlet can be found at https://technet.microsoft.com/en-us/library/hh848515.aspx.

Datacenter Bridging

Datacenter Bridging (DCB) is a collection of open standards developed by the IEEE (Institute of Electrical and Electronics Engineers). The main goal of DCB is to resolve the reliability problem of the Ethernet without using complex transport protocols as TCP. Ethernet makes best-effort by design and sometimes when there are network congestions, some packets can be lost. DCB enables to avoid the packet loss for a type of traffics. Therefore, it is very important to enable DCB with RDMA over Converged Ethernet (RoCE). In this way, storage traffic will have the priority and no packet loss.

To give priority to a type of traffic, DCB uses Priority-Based Flow Control (PFC). PFC is defined in the IEEE 802.1Qbb standard.

Remote Direct Memory Access

Remote Direct Memory Access (RDMA) is a technology that enables the network adapter to carry data directly in memory without using buffers, CPU or Operating System. So, RDMA enables us to increase throughput and reduce the resource utilization.

This is a key feature in Microsoft Hyper Converged solution because it is recommended to use RDMA with Storage Spaces Direct. Two RDMA implementation are supported by Storage Spaces Direct: iWARP or RoCE.

SMB Direct

Since Windows Server 2012, workload as Hyper-V or SQL Server can leverage a feature called **SMB Direct**. This feature uses network adapters RDMA capable. This enables to reduce the CPU utilization as well as the latency and increase the throughput. Thanks to SMB Direct, the remote file server looks like a local storage.

Now there are three types of RDMA network adapters: RoCE (RDMA over Converged Ethernet), Infiniband and iWARP.

In our solution, SMB Direct is useful for Storage Spaces Direct.

Fabric design recommendation

Microsoft has improved Failover Clustering in Windows Server 2016 by many ways. One of the improvement is **Simplified SMB multichannel** which eases the deployment. Thanks to this feature, you can now have multiple network adapters for SMB in the same subnet. Moreover, Failover Clustering configure for you the network adapters depending on network topology. More information on this topic here: http://bit.ly/2lJs4IF.

With Simplified SMB multichannel in Windows Server 2016, we can now simplify the fabric network as below:

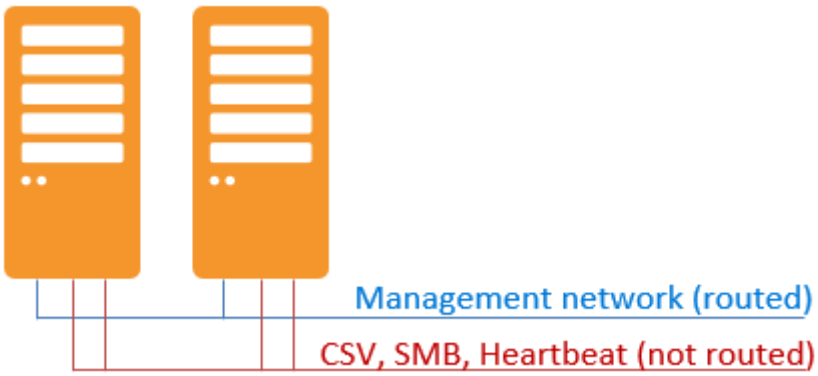


Figure 7: fabric network for hyperconverged solution

The figure 3 presents a logical view of what you need in term of fabric networks and virtual network adapters. You need two fabric networks: the management network and the cluster network. The management network is routed and is used to manage nodes. Just one network adapter per node is required for this usage. In the other hand, the cluster network is not routed and is used for cluster heartbeat, storage and live-migration. I recommend two network adapters per node for this usage. It is highly recommended that these network adapters support RDMA (RoCE or iWARP).

So, if you choose to converge all traffics into two physical network adapters or more (figure 6, section **Network Convergence**) you need only three virtual network adapters: one for management and two for clusters. It is interesting to deploy two vNICs for cluster because you can map a vNIC to a physical network adapter. If the related physical network adapter is down, the vNIC is mapped to another physical network adapter in the teaming. So, with several vNIC for cluster traffics, you can get a better efficiency.

Software-Defined Compute

The Software-Defined Compute part concerns technologies that aim to run Virtual Machines and so multiple Guest OS on a single hypervisor or on a cluster of hypervisors. This is a small part to describe the key feature for the Microsoft Hyper Converged solution.

Hyper-V

Hyper-V is the Microsoft hypervisor implementation. Hyper-V enables a physical server to become a hypervisor and so to host several Virtual Machines. Hyper-V Hosts can be added to a cluster to support the High Availability of Virtual Machines.

Hyper-V isolates physical server and virtual machines Operating System (OS) in the partition. When you deploy Hyper-V role on the host, a parent partition (or root partition) is created where the host OS is executed. Then each time you create a VM, its OS is executed in child partition. The parent partition has direct access to the hardware devices.

Below is a high-level Hyper-V diagram (copied from MSDN):

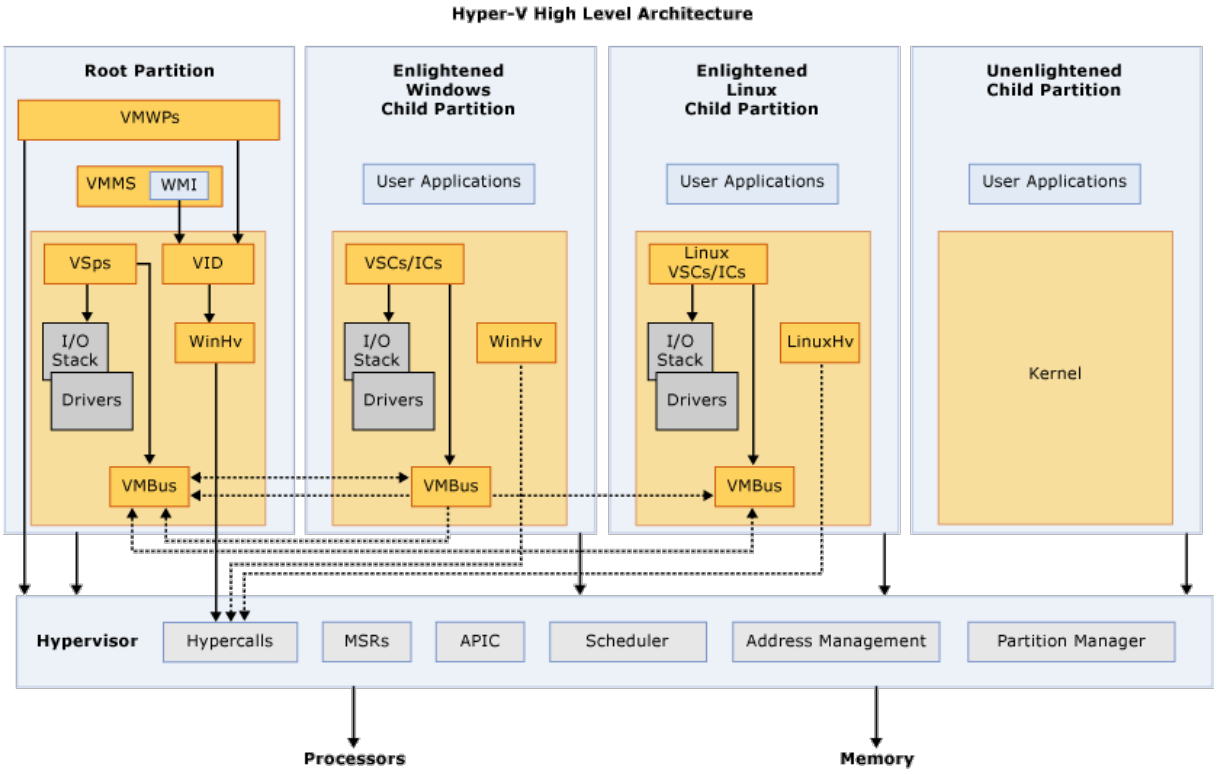


Figure 8: Hyper-V High Level Architecture

Acronyms and terms used in the diagram above are described below:

- **APIC** – Advanced Programmable Interrupt Controller – A device which allows priority levels to be assigned to its interrupt outputs.
- **Child Partition** – Partition that hosts a guest operating system - All access to physical memory and devices by a child partition is provided via the Virtual Machine Bus (VMBus) or the hypervisor.

- **Hypercall** – Interface for communication with the hypervisor - The hypercall interface accommodates access to the optimizations provided by the hypervisor.

- **Hypervisor** – A layer of software that sits between the hardware and one or more operating systems. Its primary job is to provide isolated execution environments called partitions. The hypervisor controls and arbitrates access to the underlying hardware.

- **IC** – Integration component – Component that allows child partitions to communication with other partitions and the hypervisor.

- **I/O stack** – Input/output stack

- **MSR** – Memory Service Routine

- **Root Partition** – Manages machine-level functions such as device drivers, power management, and device hot addition/removal. The root (or parent) partition is the only partition that has direct access to physical memory and devices.

- **VID** – Virtualization Infrastructure Driver – Provides partition management services, virtual processor management services, and memory management services for partitions.

- **VMBus** – Channel-based communication mechanism used for inter-partition communication and device enumeration on systems with multiple active virtualized partitions. The VMBus is installed with Hyper-V Integration Services.

- **VMMS** – Virtual Machine Management Service – Responsible for managing the state of all virtual machines in child partitions.

- **VMWP** – Virtual Machine Worker Process – A user mode component of the virtualization stack. The worker process provides virtual machine management services from the Windows Server 2008 instance in the parent partition to the guest operating systems in the child partitions. The Virtual Machine Management Service spawns a separate worker process for each running virtual machine.

- **VSC** – Virtualization Service Client – A synthetic device instance that resides in a child partition. VSCs utilize hardware resources that are provided by Virtualization Service Providers (VSPs) in the parent partition. They communicate with the corresponding VSPs in the parent partition over the VMBus to satisfy a child partitions device I/O requests.

- **VSP** – Virtualization Service Provider – Resides in the root partition and provide synthetic device support to child partitions over the Virtual Machine Bus (VMBus).

- **WinHv** – Windows Hypervisor Interface Library - WinHv is essentially a bridge between a partitioned operating system's drivers and the hypervisor which allows drivers to call the hypervisor using standard Windows calling conventions

- **WMI** – The Virtual Machine Management Service exposes a set of Windows Management Instrumentation (WMI)-based APIs for managing and controlling virtual machines.

Live-Migration

Live-Migration is a Hyper-V feature, which is used to a move a running VM from one Hyper-V node to another in a cluster, without downtime. Thanks to this feature, we can balance the resource utilization on each node in the cluster. Moreover, when you have to update a node in a cluster (Microsoft Update), you can move all VMs to other nodes to reboot the host without impact on the production workloads.

In a Hyper-V cluster, the storage is usually shared and accessible for each node. Therefore, there is just the VM memory to copy from one node to another. Below is the process to move a VM from one node to another:

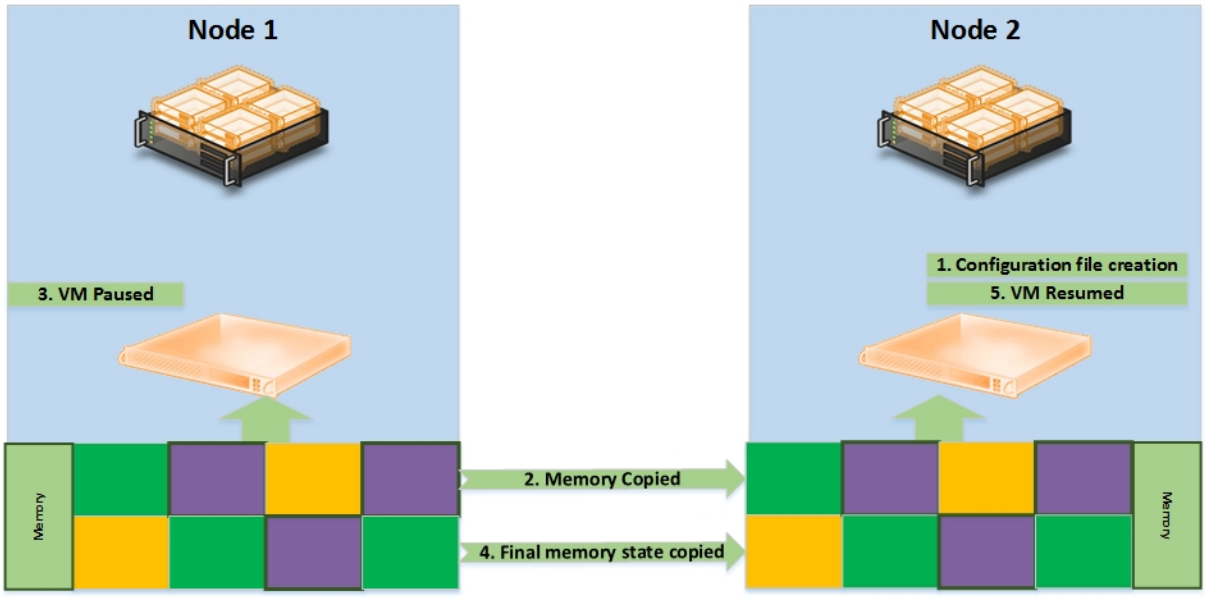


Figure 9: Live-Migration

1. Hyper-V will create a copy of the VM specification and configure dependencies on Node 2.
2. The memory of the source VM is divided up into a bitmap that tracks changes to the pages. Each page is copied from Host Node 1 to Host Node 2, and each page is marked as clean after it was copied.
3. The virtual machine on Node 1 is still running, so memory is changing. Each changed page is marked as dirty in the bitmap. Live Migration will copy the dirty pages again, marking them clean after the copy. The virtual machine is still running, so some of the pages will change again and be marked as dirty. The dirty copy process will repeat until: it has been done 10 times or there is almost nothing left to copy.
4. What remains of the source VM that has not been copied to Node 2 is referred to as the state. At this point in time, the source VM is paused on Node 1.
5. The state is copied from Node 1 to Node 2, thus completing the virtual machine copy.
6. The VM is resumed on Node 2.
7. If the VM runs successfully on Node 2, then all traces are removed from Node 1.

Once the VM is resumed a small network outage can occur until the ARP (Address Resolution Protocol) table is updated on network devices (lose of two pings).

To speed up the Live-Migration process, the virtual machine memory transferred through the network is compressed by default in Windows Server 2012 R2 and in Windows Server 2016. However, it is also possible to leverage SMB Direct for a faster Live-Migration.

By using SMB Direct, Live-Migration can use RDMA network acceleration and SMB Multichannel. RDMA provides a low latency network and CPU utilization and increases the throughput. The SMB Multichannel enables to use multiple network connections simultaneously. This result to increase the throughput and to gain the network fault tolerance.

Hyper-V Replica

Hyper-V Replica enables to implement a Data Recovery Plan (DRP) between two Hyper-V hosts or cluster. Thanks to Hyper-V Replica, VMs can be replicated to another hypervisor and synchronize every 30 seconds, 5 or 15 minutes. When an outage occurs on the source, a failover can be executed to start the VM on the target site (manual process).

When Hyper-V Replica is needed from or to a cluster, it is necessary to deploy the Hyper-V Replica broker cluster role. Because VM can move in a cluster (Live-Migration or Quick-Migration), it is necessary to locate where is the VM in the cluster. It is the role of the Hyper-V Replica Broker.

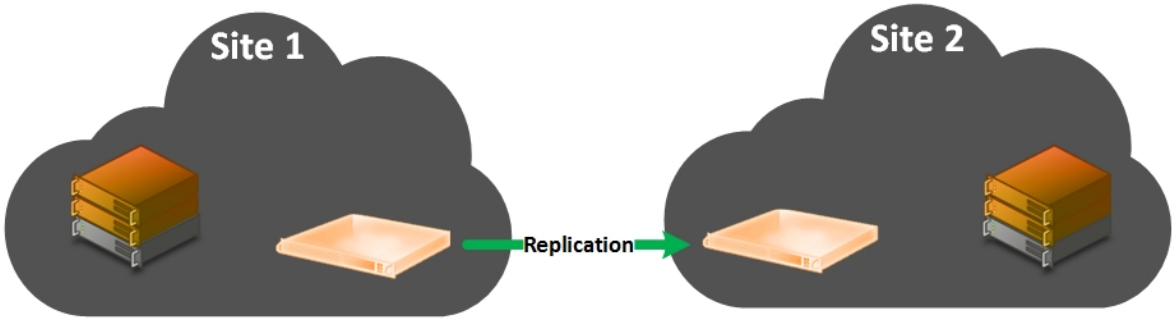


Figure 10: Hyper-V Replica

Software-Defined Storage

The Software-Defined Storage part aims to describe the software storage solution in the Microsoft Hyper Converged infrastructure.

Storage Pool and Storage Spaces

Before talking about Storage Spaces Direct, it is important to understand the concept of Storage Spaces and Storage Pool.

The Storage Spaces feature has been released with Windows Server 2012. It enables to aggregate several storage devices together (Storage Pool) and create virtual disk on top (Storage Spaces). The virtual disk created can be resilient depending of your choice (mirroring or parity). This enables to avoid using expensive and inflexible hardware devices as RAID Controllers or SAN.

Storage Spaces Direct uses the same functionality. However, some software has been added between storage devices and storage pool.

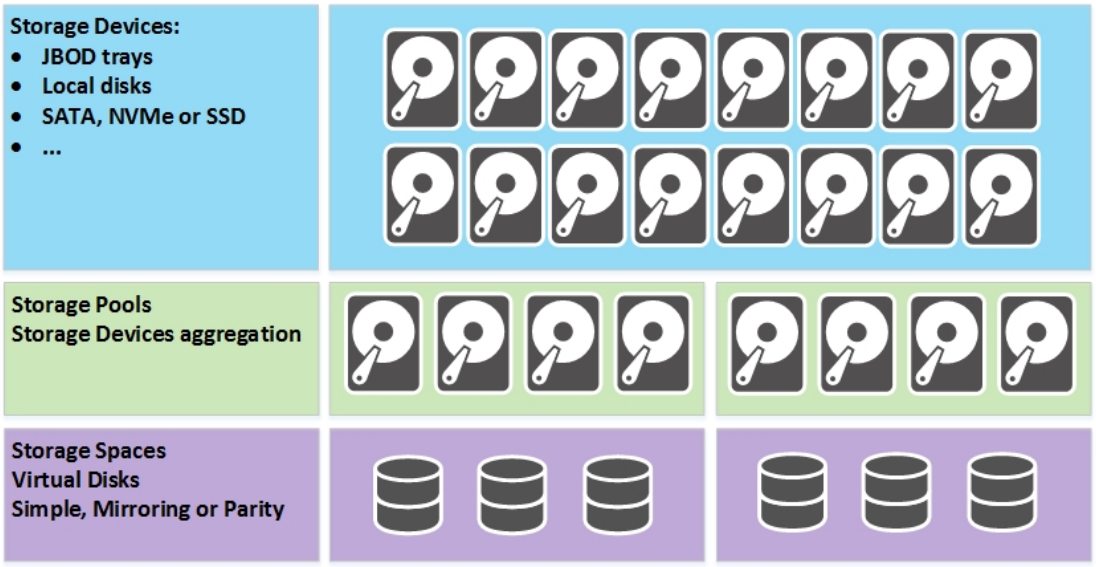


Figure 11: Storage Spaces explanation

In a Storage Spaces Direct system, before creating any storage pool, you need several storage devices that can be located in Just a Brunch of Disk (JBOD) trays, or attached locally. These disks can be SATA, SAS or NVMe. Next, you can create your storage pools by aggregating selected storage devices. You can mix types of storage devices (SSD + HDD, NVMe + SSD and so on).

Then you can create virtual disks called Storage Spaces on top of the storage pool. Storage Spaces supports some resiliency mode as mirroring and parity (C.F next section).

To avoid that all nodes take the ownership of a clustered Storage Spaces at the same time, the Failover Cluster is used. After creating the Virtual Disks, it is necessary to convert them into a Cluster Shared Volume. In this way, only one node can take the ownership of a virtual disk at the same time.

Figure 12: Clustered Storage Spaces explanation

Storage Spaces Direct

Storage Spaces Direct is a new Windows Server 2016 feature that enables a new deployment model without using shared JBODs to create clustered Storage Spaces. Now you can use local storage devices either connected via JBOD or internally.

It is possible because Microsoft has developed the Software Storage Bus (SSB), which enables each server to see all disks connected to each node in the cluster. Once each node sees all storage devices, it is easy to create the Storage Pool. SSB takes advantages of SMB3, SMB Direct and SMB Multichannel to transfer data blocks between cluster nodes.

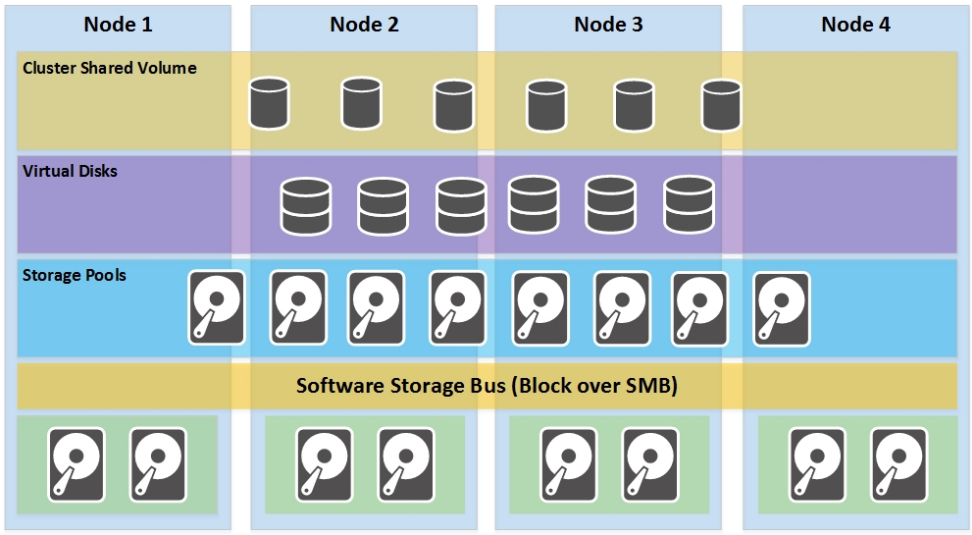


Figure 13: Software Storage Bus (SSB)

Two components in SSB enables to create a clustered Storage Spaces with local storages:

- **ClusPort**: create a Virtual HBA, which enables to connect to storage devices of the other nodes in the cluster (initiator).
- **Clusblft**: enables to virtualize storage devices in each node in order to ClusPort connect to (target).

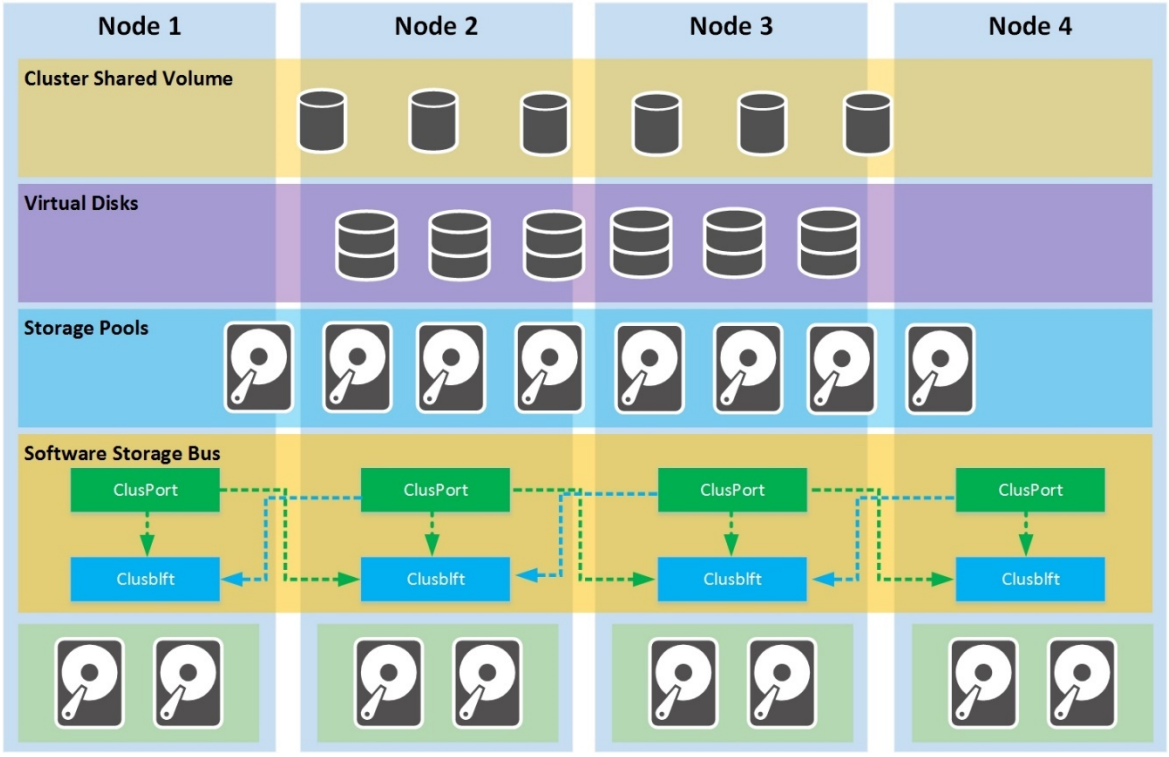


Figure 14 : ClusPort & Clusblft interactions

SSB can use a cache mechanism called Storage Bus Cache (SBC) which is protected against failure. When enabling Storage Spaces Direct, the storage devices are selected for caching and for capacity.

Storage Configuration	Caching devices	Capacity devices	Caching behavior
SATA SSD + SATA HDD	All SATA SSD	All SATA HDD	Read + Write
NVMe SSD + SATA HDD	All NVMe SSD	All SATA HDD	Read + Write
NVMe SSD + SATA SSD	All NVMe SSD	All SATA SSD	Write only
NVMe SSD + SATA SSD + SATA HDD	All NVMe SSD	All SATA SSD All SATA HDD	Read for SATA SSD Read + Write for SATA HDD
Only NVMe SSD or Only SATA SSD	/	All	No

When the system has determined, which devices are the caching or the capacity devices, the capacity devices are associated to a caching device by using round robin algorithm. The caching devices are always the fastest storage devices (SSD over HDD, NVMe over SSD and so on). This is the default behavior that you can override when enabling Storage Spaces Direct.

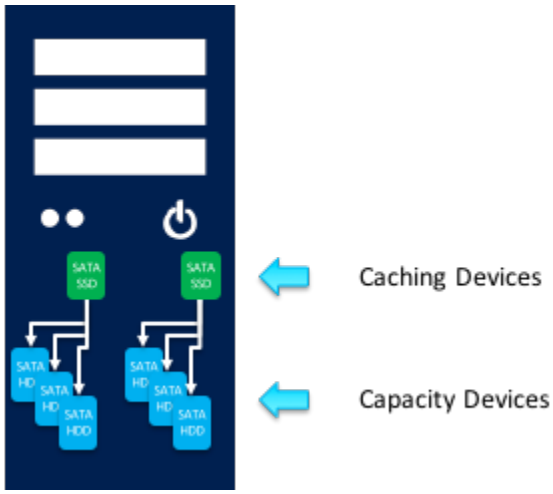


Figure 15: Caching and Capacity devices association when SBC is enabled (Image source TechNet)

If a Caching device failed, the related capacity devices are bound automatically to a working caching device. When SBC is enabled, it creates a partition on each caching device that use by default all capacity except 32GB. This capacity is used for storage pool and virtual disk metadata.

SBC needs some node memories to work. It needs almost 4GB of memory per one TB of caching devices. Even if you can run Storage Spaces Direct with a single cache device per node, Microsoft highly recommends implementing at least two cache devices per node to preserve performance and availability. When choosing the number of cache devices, be sure to respect a ratio with number of capacity devices. For example, if you plan to install 8 capacity devices, choose two or four cache devices. You can have further information on cache in this topic.

Storage Spaces resiliency

What will be a storage solution without protection against data loss? If you work on storage and especially on the hardware, you know that storage devices as Hard Drives are sensitive and often fail. To avoid losing data when a hard drive fails, the data must be redundant on several other storage devices. With Storage Spaces Direct, there are two resiliency modes:

- **Mirror**: In this mode, the data are written across multiple physical disks and a copy is performed one or two times. When the copy is performed one time, it is called **Two-Way Mirroring** and when the data is copied two times, it is called **Three-Way Mirroring**. This mode is great for most of workloads as Hyper-V and SQL because it provides good performance level especially when using tiering.

- **Parity:** In this mode, the data are written across multiple physical disks and a parity information is copied one or two times. When the parity information is copied one time, it is called **Single Parity** and when the data is copied two times, it is called **Dual Parity**. Because of the parity, the write performance is not high, but the capacity is maximized. Thus, this mode is good for backup workload**.**

To have multiple disks that read/write simultaneously and so to add the performance of each disk in a storage pool, the data is divided into blocks and they are distributed across several physical disks. This is called **stripping**.

Therefore, the Storage Spaces stripes the data across a specific number of disks also called a **number of columns** (a column is a physical disk). The data are divided into small blocks that have a specific size, also called **Interleave** (default value: 256KB).

Storage Spaces doesn't replicate block directly from one storage device to another. Instead, it uses the concept of **slab**. When Storage Spaces creates a volume, it divides it in small piece (called slab) of 256MB. Then the blocks are written inside these slabs. Next, following the resiliency type, Storage Spaces replicates the slabs across the physical disks.

As mentioned before, for a storage solution designed to host virtual machines workloads, it is preferred to use Mirroring. Usually you will also use a Two-Way Mirroring at least. The number of physical disks required is the **number of columns x 2** because the data exists two times in the storage spaces. In this case, you need at least two physical disks (**N.B**: Storage Spaces Direct requires four capacity devices at least per node). Below is an explanation schema:

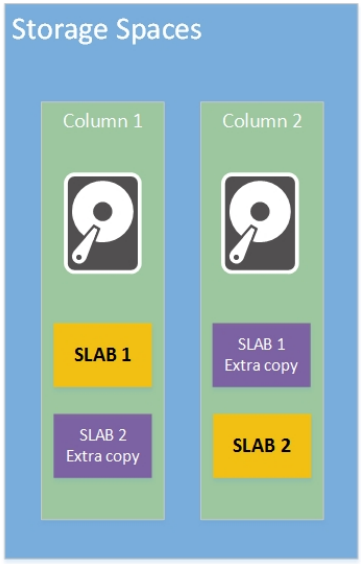


Figure 16: Two-Way Mirroring, 1 data column

In the above example, the data are not stripped across several disks. So, the read/write performances are equal to one physical disk. However, the data is copied one extra time, so this solution supports one disk failure.

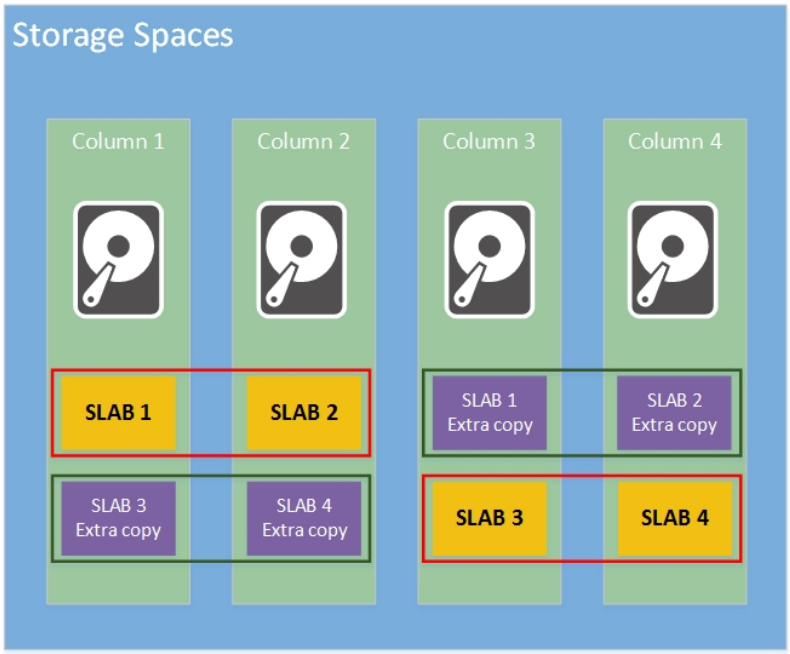


Figure 17: Two-Way Mirroring, 2 data columns

In the above solution, the data is stripped across two columns and the data have an extra copy. The read/write performance is the sum of two disks. The slab size is equal to 256MB and the block size are equal to 256KB by default (interleave).

Finally, when a disk fails, it is possible to rebuild its data to other disks. A feature called **Storage Spaces Parallel rebuild** enables to make an extra copy of the blocks stored on a failed disk to the running disks. However, this requires some free space in the Storage Spaces. You should at least leave un-provisioned space as the size of a single capacity disk in the storage pool.

Fault Domain Awareness

A fault domain is a collection of hardware that shares the same point of failure. In a standard cluster, the fault domain is a node. So, the data are replicated regarding the node. The data and the extras copies are not placed in the same node because if you lose the node, you lose data and all copies.

Some vendor provides chassis composed of two or four servers. These servers share the same power supplies. So, in this case the fault domain is the chassis.

Now let's think about a solution spread across two, three or four racks and you want to be resilient to the loose of a rack. In this case, the fault domain can be the rack.

In Windows Server 2016, Microsoft offers an enhancement for Failover Clustering called Fault Domain Awareness. Thanks to this feature, you can define your physical infrastructure by using an XML or PowerShell. Then the storage spaces places data and extra copies regarding the fault domain.

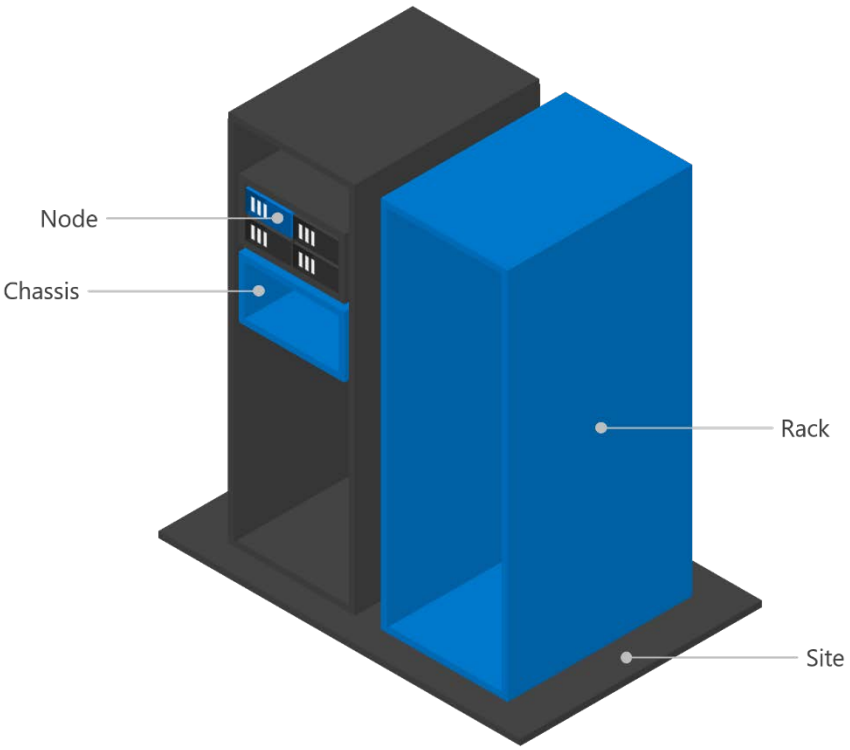


Figure 18: Fault Domain Awareness

Instead of talking about node remaining in case of failure, now Microsoft talks about fault domain remaining in several documentations. The below table presents the cluster resiliency regarding the resiliency mode:

Resiliency mode	Cluster resiliency
2-Way mirroring	1 Fault Domain
3-Way mirroring	2 Fault Domains
Single Parity (erasure coding)	1 Fault Domain
Dual Parity (erasure coding)	2 Fault Domains

More information about fault domain awareness in Windows Server 2016 can be found at:
https://technet.microsoft.com/en-us/windows-server-docs/failover-clustering/fault-domains

Multi-Resiliency Virtual Disks and ReFS Real-Time Tiering

Windows Server 2016 has introduced the multi-resiliency virtual disks. This enables to have two tiers in a virtual disk: one in mirroring and the other in erasure-coded parity).

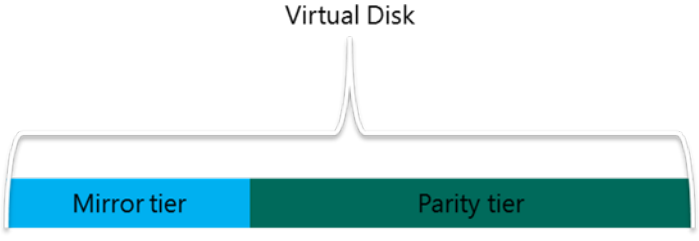


Figure 19: Multi-Resiliency Virtual Disks (copied from TechNet blog)

In this configuration, ReFS will operate on this virtual disk. The data are always written to the Mirror tier even if it is an update of a data that is already in the parity tier. In this case, the updated data in the parity tier are invalidated. ReFS always writes in the Mirror Tier because it is the fastest tier, especially for virtual machines.

Then ReFS rotates the data from the mirror tier to the parity tier and makes the erasure coding computation. As the mirror tier gets full, data will be moved automatically to the parity tier.

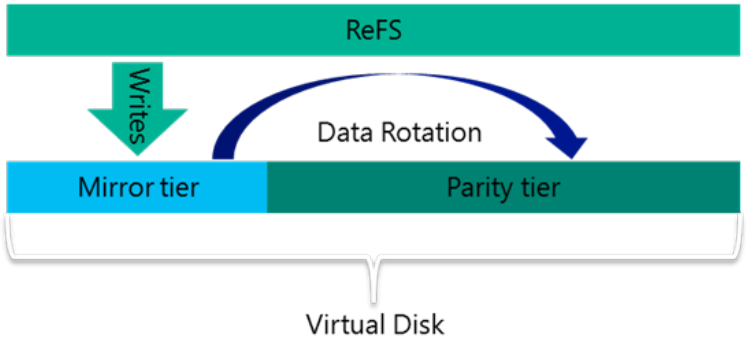


Figure 20: ReFS Real-Time Tiering (Image source TechNet blog)

This configuration is only available with 4-nodes configuration and above.

Three tiers configuration

In the last part, we have seen how to create a virtual disk with two tiers to balance between performance and efficiency. When you choose to implement this type of storage solution based on SSD + HDD, SSDs are used for the cache and HDD for the multi-resiliency virtual disk.

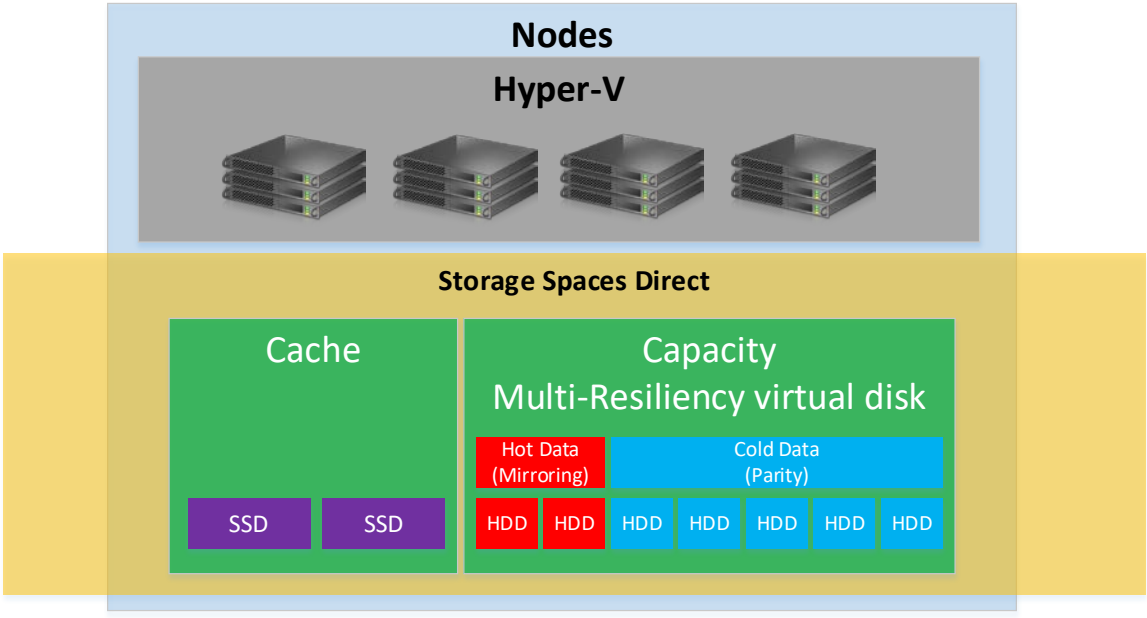


Figure 21: SSD+HDD for the three tiers configuration

You can also mix NVMe, SSD and HDD disks for the three tiers configuration. NVMe will be used for the cache, SSDs for the hot data tier and HDD for the cold data tier.

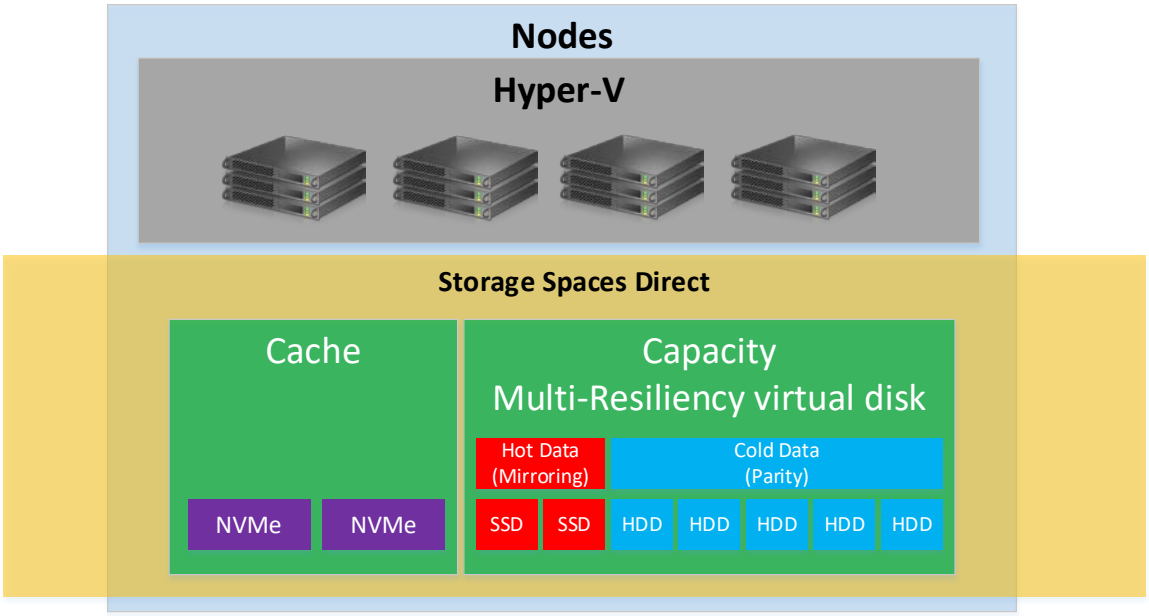


Figure 22: NVMe + SSD + HDD for the three tiers configuration

With three tiers solution, the cache is managed by the high performance NVMe disks. Then data are written on the SSD which are efficient. Then when hot data volume is full, the data rotation occurs and they are moved to the cold data HDD tier. Because this tier is a parity tier, the capacity is maximized.

To support this kind of solution, you need at least a 4-nodes cluster and you must use the ReFS file system.

ReFS file system

Resilient File System (ReFS) is a new file system, which was released with Windows Server 2012. The ReFS is designed to increase the integrity, availability, scalability and the Proactive Error Correction. In Windows Server 2012 ReFS was not much used because it lacked features compared to NTFS (No Disk Quotas, no compression, no EFS and so on).

However, in Windows Server 2016, ReFS brings accelerated VHDX operations. It enables to create fixed VHDX or merge checkpoint almost instantly. Therefore, in Windows Server 2016, it could be interesting to store VMs on a ReFS volume.

Moreover, ReFS comes with some integrity and protection against corruption. ReFS does not need to check disk as we execute on NTFS partition. Thus, when a Cluster Shared Volume is formatted by using ReFS, you will no longer be warning to run a CHKDSK on the volume.

This is the recommended file system for Storage Spaces Direct.

As of this writing, data deduplication is not supported on ReFS file system (work in progress).

Health Service

Microsoft has released a new role in cluster when Storage Spaces Direct is activated. It is called the Health Service. The Health Service aggregates metrics and real time alert of cluster nodes. The metrics gives information about storage, overall CPU consumption and memory usage. The real-time alert you about issues in the cluster such as a network cable unplugged, a node down, a physical disk down and so on.

The Health Service can be accessible from API by using PowerShell, .Net or C#. So, you can create dashboard with Health Service based on the real-time metrics and alerts. For example, the SCOM management pack is based on S2D Health Service:

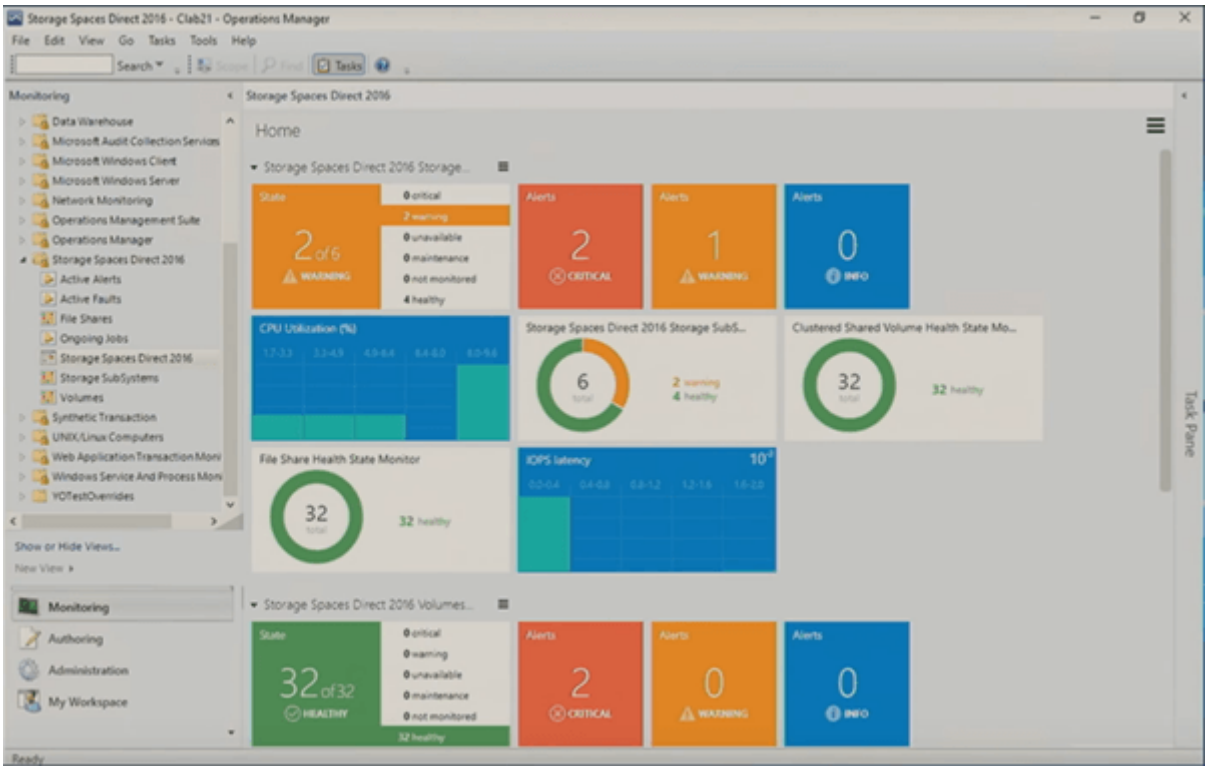


Figure 23: SCOM Storage Spaces Direct management pack

You can download the Microsoft System Center 2016 Management Pack for Windows Storage Spaces Direct at the following link: https://www.microsoft.com/en-us/download/details.aspx?id=54700

DataOn, a server manufacturer, has released a web interface called "**DataOn Must**" which provides graphical data from Health Service. You have an example below:

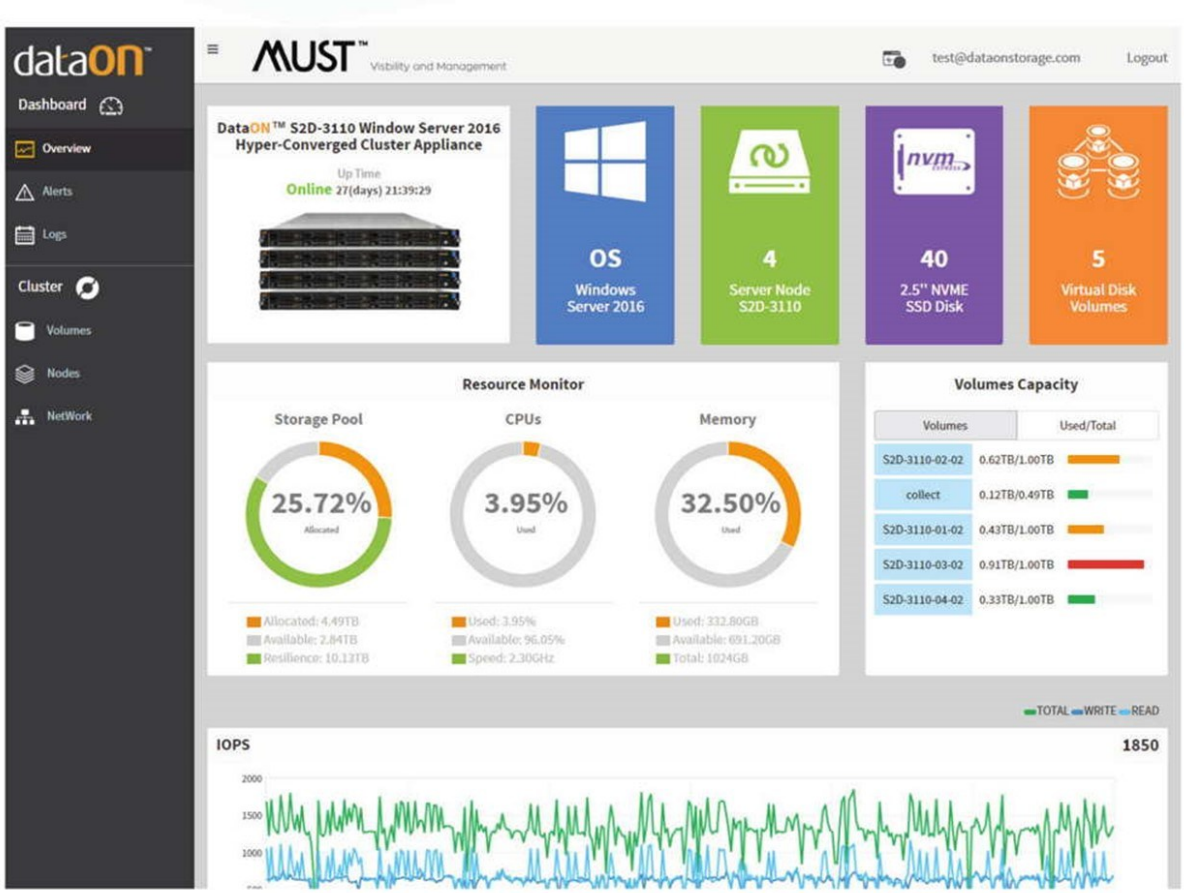


Figure 24: DataOn Must

Storage Replica

Windows Server 2016 comes with a new feature for a Data Recovery Plan or for stretched cluster. You are now able to make a replica from a first Storage Spaces to another by using SMB.

In Windows Server 2016, Storage Replica supports the following scenarios:

- Server-to-server storage replication using Storage Replica
- Storage replication in a stretch cluster using Storage Replica
- Cluster-to-cluster storage replication using Storage Replica
- Server-to-itself to replicate between volumes using Storage Replica

In term of Storage Replica and Storage Spaces Direct for Hyper-Converged solution, Microsoft generally recommends the following scenario (Cluster to Cluster replication). As of this writing, **Storage Spaces Direct in stretched cluster is not supported with Storage Replica.**

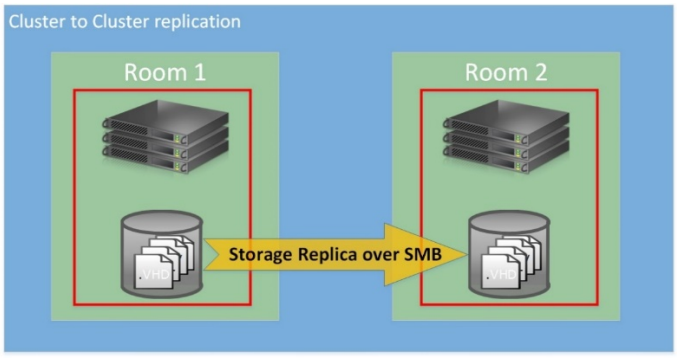


Figure 25 : Cluster to cluster replication

In the above example, the Room 1 cluster is the active one while the cluster in room 2 is the passive. The storage in Room 1 is replicated to the storage in the Room 2. Because the Storage Replica uses SMB, it can leverage RDMA to increase throughput and to decrease CPU utilization.

As most of the replication solution, Storage Replica supports Synchronous and Asynchronous replication. In Synchronous mode, the application receives an acknowledgement when the replication has occurred successfully; while in the asynchronous mode the application receives the acknowledgement immediately after the replication engine has captured data. The synchronous mode can degrade the application performance if the storage replica solution is not well designed. The Synchronous replication is suitable for HA and DR solutions.

For both modes, you need a log volume in each site or location. This log volume should be on a fast storage device as an SSD. These log volumes are used as a buffer for the replication process.

Storage Quality of Service

Storage Quality of Service (Storage QoS) was introduced in Windows Server 2012 R2 that enabled to set the minimum and the maximum IOPS on a single virtual hard disk VHD(X). That was great, but it was not enough; Microsoft is investing more in this area to bring more functionalities.

There are a couple of big challenges with Storage QoS today in Windows Server 2012 R2, this technology allows us to cap the amount of IOPs for each virtual machine/virtual hard disk individually, and that all works great on a single Hyper-V server!

However, to be honest, no one is deploying a standalone Hyper-V host in production, of course we are leveraging Hyper-V cluster for High Availability.

Storage QoS today doesn't really works great if you have dozens of Hyper-V servers talking to same piece of storage at the back-end, because in Windows Server 2012 R2 those Hyper-V servers are not aware they are competing for storage bandwidth.

Thankfully, in Windows Server 2016, Microsoft introduced Distributed Storage QoS policy manager directly attached on Failover Cluster as a **Storage QoS Resource**.

This enables us to create policies to set a minimum IOPS and a maximum IOPS value based on a **flow**. Each file handles opened by a Hyper-V server to a VHD or VHDX file is considered as a "flow".

Distributed Storage QoS enables to make multi-tier performance between several VMs. For example, you can create a “**Gold**” policy and a “**Platinum**” policy and associate them to the right group of virtual machines, and virtual hard disks.

Of course, we still have what we had in Windows Server 2012 R2, where you can go to Hyper-V and configure Storage QoS properties for each virtual machine/virtual hard disk. But in Windows Server 2016 we can actually now go to the Scale-Out File Server cluster or to Hyper-V cluster using Cluster Shared Volumes (CSV) for storage, and configure the Storage QoS policies there. This enables a couple of interesting scenarios:

The first scenario is, if you have a multiple of Hyper-V servers talking to the same storage at the back-end, then all your storage QoS policies get respected.

The second scenario is, allowing us to do some cool things, where we can now start pulling Storage QoS policies and having a single policy that applies to a group of virtual machines instead of just one VM or one virtual hard disk.

Distributed Storage QoS in Windows Server 2016 supports two deployment scenarios:

1- Hyper-V using a Scale-Out File Server. This scenario needs the following:
   - Storage cluster that is a Scale-Out File Server cluster and a Compute cluster that has least one server with the Hyper-V role enabled.
   - For distributed Storage QoS, the Failover Cluster is required on Storage side, but it is optional on the Compute side.

2- Hyper-V using Cluster Shared Volumes (CSV) such as Storage Spaces Direct in Hyper-Converged model as described in this whitepaper. This scenario needs the following:
   - Compute cluster with the Hyper-V role enabled.
   - Hyper-V using Cluster Shared Volumes (CSV) for storage Failover Cluster is required.

Windows Server 2016 licensing

Microsoft introduced a new licensing model for Windows Server 2016. Now the license model is per core. In the below table, you can see the feature available in each edition of Windows Server 2016. As you can see, the features required for storage are included in the Datacenter Edition.

Windows Server 2016 Editions		
	Datacenter Edition	Standard Edition
Core functionality of Windows Server	●	●
OSEs/Hyper-V containers*	Unlimited	2
Windows Server containers	Unlimited	Unlimited
Nano Server	●	●
New storage features including Storage Spaces Direct and Storage Replica**	●	
New Shielded Virtual Machines and Host Guardian Service**	●	
New networking stack**	●	
Licensing Model***	Core + CAL	Core + CAL
Price+	$6,155	$882

* Windows Server Standard Edition license permits 2 OSEs (operating system environments) when all physical cores are licensed.

** Azure-inspired features for advanced software-defined datacenter scenarios.

*** See Licensing Datasheet for additional detail. Minimum license requirement: 8 cores per processor, 16 cores per server.

+ Pricing represents Open No Level (NL) ERP for 16 cores

Microsoft will sell licenses by 2-core packs. At least you need 4 packs for an 8-core processor and at least 8 packs per physical server.

For two processors with 8 cores each, the price is the same as Windows Server 2012 R2 Datacenter. Beyond that, it will be more expensive than Windows Server 2012 R2.

Number of 2-core pack licenses needed

(Mininum 8 cores/proc; 16 cores/server)

Procs per server	Physical cores per processor: 2	4	6	8	10
1	8	8	8	8	8
2	8	8	8	8	10
4*	16	16	16	16	20

Licensing costs are same as 2012 R2

Additional licensing required

* *Standard Edition may need additional licensing*

Implementation guide

Design overview

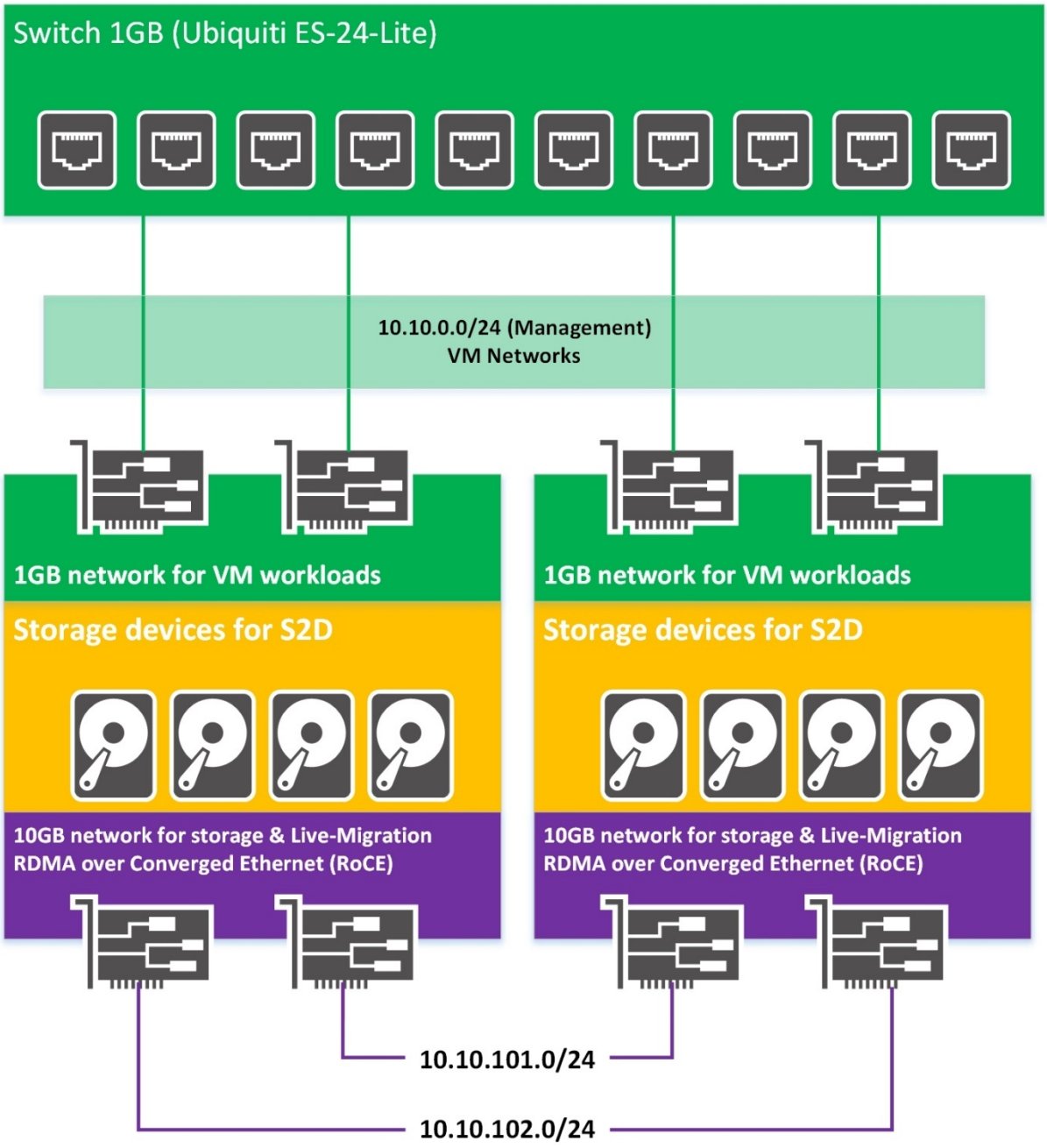
In this part, we'll talk about the implemented hardware and how the nodes are connected. Then we'll introduce the network design and the required software implementation. This implementation guide presents how to deploy a 2-node hyper-converged cluster.

Hardware consideration

We have bought two nodes that we have built ourselves. Both nodes are not provided by a server manufacturer. You can find below the hardware that we have added in each node:

- CPU: Xeon 2620v2
- Motherboard: Asus Z9PA-U8 with ASMB6-iKVM for KVM-over-Internet (Baseboard Management Controller)
- PSU: Fortron 350W FSP FSP350-60GHC
- Case: Dexlan 4U IPC-E450
- RAM: 128GB DDR3 registered ECC
- Storage devices:
    - 1x Intel SSD 530 128GB for the Operating System
    - 4x Samsung SATA SSD 850 EVO 500GB (Storage Spaces Direct capacity) (*)
- Network Adapters:
    - 1x Intel 82574L 1GB for VM workloads (two controllers). Integrated to motherboard
    - 1x Mellanox Connectx3-Pro 10GB for storage and live-migration workloads (two controllers). Mellanox are connected with two passive copper cables with SFP provided by Mellanox
- 1x Switch Ubiquiti ES-24-Lite 1GB

(*) **If used in production, we'd replace SSD by enterprise grade SSD. It is not recommended to deploy Consumer grade SSD in production. Finally, we'd buy a server with two Xeon® processors.**



Network design

To support this configuration, we have created five network subnets:

- **Management** network: 10.10.0.0/24 – VID 10 (Native VLAN). This network is used for Active Directory, management through RDS or PowerShell and so on. Fabric VMs will be also connected to this subnet.
- **Storage01** network: 10.10.101/24 – VID 101. This is the first storage network. It is used for SMB 3.11 transaction and for Live-Migration.
- **Storage02** network: 10.10.102/24 – VID 102. This is the second storage network. It is used for SMB 3.11 transaction and for Live-Migration.

We can't use Simplified SMB Multichannel in this deployment because we don't have a 10GB switch. So, each 10GB controller must belong to a separate network subnet.

We will deploy a Switch Embedded Teaming (SET) for 1GB network adapters. We will not implement SET on the 10GB NICs, because we don't have a physical switch that support 10GB.

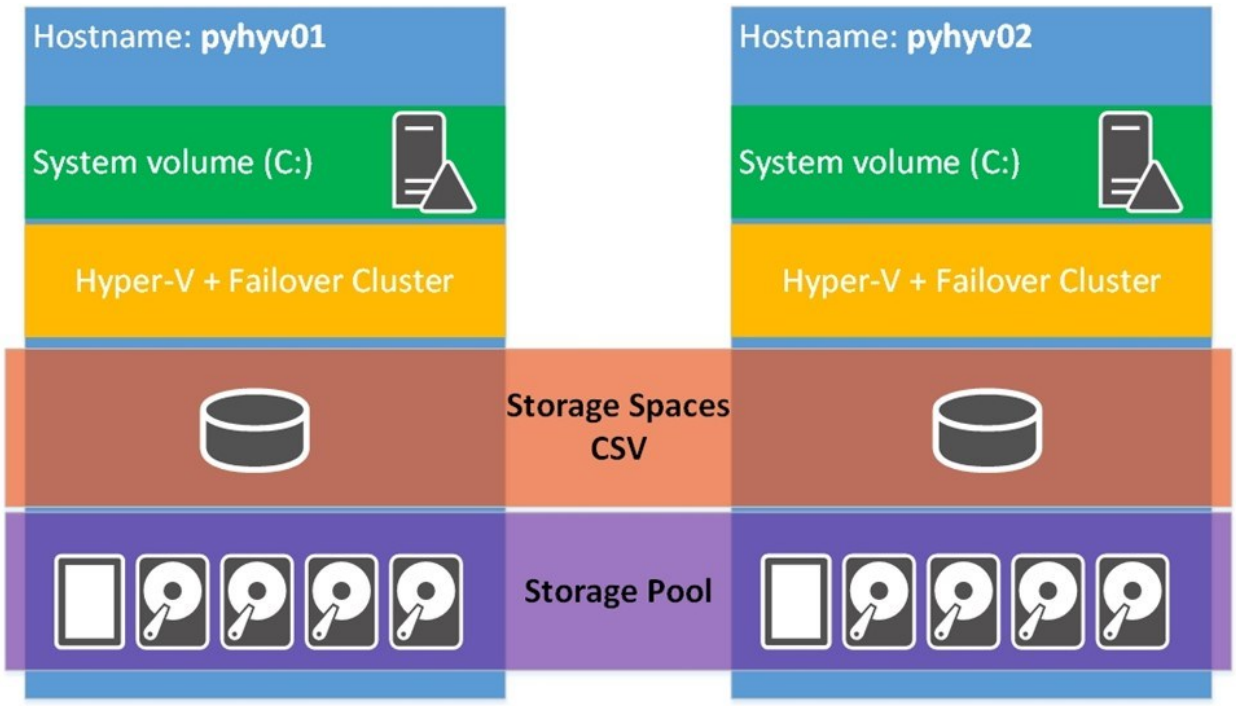
Logical design

We will have two nodes called pyhyv01 and pyhyv02 (**Phy**sical **Hy**per-**V**).

The first challenge concerns the failover cluster. Because we don't have another physical server, the domain controllers will be virtual. if I implement domain controllers VM in the cluster, how can start the cluster? So, the DC VMs must not be in the cluster and must be stored locally. To support high availability, both nodes will host a domain controller locally in the system volume (C:\). In this way, the node boot, the DC VM start and then the failover cluster can start.

Both nodes are deployed in core mode because we don't like graphical user interface for hypervisors. We don't deploy Nano Server because we don't like the Current Branch for Business model for Hyper-V and storage usage. The following feature will be deployed for both nodes:

- Hyper-V + PowerShell management tools
- Failover Cluster + PowerShell management tools
- Storage Replica (this is optional, only if you need the storage replica feature)



The storage configuration will be easy: We'll create a unique Storage Pool with all SATA SSD. Then we will create two Cluster Shared Volumes (CSV) that will be distributed across both nodes. The CSV will be called CSV-01 and CSV-02.

Operating system configuration

We show how to configure a single node. You should repeat these operations for the second node in the same way. Therefore, we recommend creating a script with the commands described in this section: the script will help to avoid human errors.

Bios configuration

The bios may change regarding the manufacturer and the motherboard. But we always do the same things in each server:

- Check if the server boot in UEFI
- Enable virtualization technologies as VT-d, VT-x, SLAT and so on
- Configure the server in high performance (in order that CPUs have the maximum frequency available)
- Enable Hyperthreading
- Disable all unwanted hardware (audio card, serial/com port and so on)
- Disable PXE boot on unwanted network adapters to speed up the boot of the server
- Set the date/time

Next, check if the memory is seen, and all storage devices are plugged. When you have time, run a memtest on server to validate hardware.

OS first settings

We have deployed the nodes from a USB stick as described in this topic. We have deployed Windows Server 2016 Datacenter in Core edition. Once the system is installed, we have deployed the drivers for motherboard and for Mellanox network adapters. Because you can't connect with a remote MMC to Device Manager, we use the following commands to list if drivers are installed:

```
gwmi Win32_SystemDriver | `
select name,@{n="version";e={(gi $_.pathname).VersionInfo.FileVersion}}
gwmi Win32_PnPSignedDriver | select devicename,driverversion
```

```
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 IOAPIC - 0E2C                                  9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 IIO RAS - 0E2A                                 9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 VTd/Memory Map/Misc - 0E28                     9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 7 - 0E27             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 6 - 0E26             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 5 - 0E25             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 4 - 0E24             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 3 - 0E23             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 2 - 0E22             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 1 - 0E21             9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 Crystal Beach DMA Channel 0 - 0E20             9.4.0.1029
Disk drive                                                                                  10.0.14393.0
Disk drive                                                                                  10.0.14393.0
Disk drive                                                                                  10.0.14393.0
Standard SATA AHCI Controller                                                               10.0.14393.206
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 PCI Express Root Port 3c - 0E0A                9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 PCI Express Root Port 3a - 0E08                9.4.0.1029
Mellanox ConnectX-3 Pro Ethernet Adapter                                                    5.25.12665.0
Mellanox ConnectX-3 Pro Ethernet Adapter                                                    5.25.12665.0
Mellanox ConnectX-3 PRO VPI (MT04103) Network Adapter                                       5.25.12665.0
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 PCI Express Root Port 2c - 0E06                9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 PCI Express Root Port 2a - 0E04                9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 PCI Express Root Port 1a - 0E02                9.4.0.1029
Intel(R) Xeon(R) E7 v2/Xeon(R) E5 v2/Core i7 DMI2 - 0E00                                    9.4.0.1029
PCI Express Root Complex                                                                    10.0.14393.206
Microsoft ACPI-Compliant System                                                             10.0.14393.103
ACPI x64-based PC                                                                           10.0.14393.0
Microsoft ClusPort HBA                                                                      10.0.14393.206
Remote Desktop Mouse Device                                                                 10.0.14393.0
Remote Desktop Keyboard Device                                                              10.0.14393.0
UMBus Enumerator                                                                            10.0.14393.0
UMBus Root Bus Enumerator                                                                   10.0.14393.0
Microsoft Kernel Debug Network Adapter                                                      10.0.14393.0
Disk drive                                                                                  10.0.14393.0
Disk drive                                                                                  10.0.14393.0
Microsoft Storage Spaces Controller                                                         10.0.14393.206
Microsoft VHD Loopback Controller                                                           10.0.14393.0
Microsoft Virtual Drive Enumerator                                                          10.0.14393.0
Composite Bus Enumerator                                                                    10.0.14393.0
Microsoft Hyper-V Virtualization Infrastructure Driver                                      10.0.14393.206
Hyper-V Virtual Switch Extension Adapter                                                    10.0.14393.0
Cluster BFlt Driver                                                                         10.0.14393.206
Microsoft Basic Display Driver                                                              10.0.14393.0
Microsoft Hyper-V Virtual Machine Bus Provider                                              10.0.14393.0
```

After all drivers are installed, we configure the server name, the updates, the remote connection and so on. For this, we use **sconfig**.

```
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Inspecting system...


===============================================================================
                         Server Configuration
===============================================================================

1) Domain/Workgroup:                    Domain:  int.HomeCloud.net
2) Computer Name:                       PYHYV01
3) Add Local Administrator
4) Configure Remote Management          Enabled

5) Windows Update Settings:             DownloadOnly
6) Download and Install Updates
7) Remote Desktop:                      Enabled (more secure clients only)

8) Network Settings
9) Date and Time
10) Telemetry settings                  Enhanced
11) Windows Activation

12) Log Off User
13) Restart Server
14) Shut Down Server
15) Exit to Command Line

Enter number to select an option: _
```

This tool is easy, but don't provide automation. You can do the same thing with PowerShell cmdlet, but we have only two nodes to deploy and I find this easier. All you have to do, is to move in menu and set parameters. Here we have changed the computer name, we have enabled the remote desktop and we have downloaded and installed the latest Windows updates. We strongly recommend to install all updates before deploying Storage Spaces Direct.

Then we configure the power options to "performance" by using the bellow cmdlet:

```
POWERCFG.EXE /S SCHEME_MIN
```

Once the configuration is finished, you can install the required roles and features. You can run the following cmdlet on both nodes:

```
Install-WindowsFeature Hyper-V, Data-Center-Bridging, Failover-Clustering, RSAT-Clustering-Powershell, Hyper-V-PowerShell, Storage-Replica
```

Once you have run this cmdlet the following roles and features are deployed:

- Hyper-V + PowerShell module
- Datacenter Bridging
- Failover Clustering + PowerShell module
- Storage Replica

Network settings

Once the OS configuration is finished, you can configure the network. First, we rename network adapters as below:

```
get-netadapter |? Name -notlike vEthernet* |? InterfaceDescription -like
Mellanox*#2 | Rename-NetAdapter -NewName Storage-101
get-netadapter |? Name -notlike vEthernet* |? InterfaceDescription -like
Mellanox*Adapter | Rename-NetAdapter -NewName Storage-102
get-netadapter |? Name -notlike vEthernet* |? InterfaceDescription -like Intel*#2 |
Rename-NetAdapter -NewName LAN01-0
get-netadapter |? Name -notlike vEthernet* |? InterfaceDescription -like
Intel*Connection | Rename-NetAdapter -NewName LAN02-0
```

```
Name                      InterfaceDescription                    ifIndex Status       MacAddress
----                      --------------------                    ------- ------       ----------
LAN02-0                   Intel(R) 82574L Gigabit Network Conn...      13 Up           14-DD-A9-D6-78-E8
LAN01-0                   Intel(R) 82574L Gigabit Network Co...#2       6 Up           14-DD-A9-D6-78-E7
Storage-102               Mellanox ConnectX-3 Pro Ethernet A...#2       3 Up           E4-1D-2D-22-A2-F1
Storage-101               Mellanox ConnectX-3 Pro Ethernet Ada...      10 Up           E4-1D-2D-22-A2-F0
```

Next, we create the Switch Embedded Teaming with both 1GB network adapters called **SW-1G**:

```
New-VMSwitch -Name SW-1G -NetAdapterName Management01-0, Management02-0 -
EnableEmbeddedTeaming $True -AllowManagementOS $False
```

Now we can create a virtual network adapter for the management:

```
Add-VMNetworkAdapter -SwitchName SW-1G -ManagementOS -Name Management-0
```

```
Name                      InterfaceDescription                    ifIndex Status       MacAddress
----                      --------------------                    ------- ------       ----------
vEthernet (Management-0)  Hyper-V Virtual Ethernet Adapter                2 Up           00-15-5D-00-D5-00
LAN02-0                   Intel(R) 82574L Gigabit Network Conn...      13 Up           14-DD-A9-D6-78-E8
LAN01-0                   Intel(R) 82574L Gigabit Network Co...#2       6 Up           14-DD-A9-D6-78-E7
Storage-102               Mellanox ConnectX-3 Pro Ethernet A...#2       3 Up           E4-1D-2D-22-A2-F1
Storage-101               Mellanox ConnectX-3 Pro Ethernet Ada...      10 Up           E4-1D-2D-22-A2-F0
```

Then we configure VLAN on each storage NIC:

```
Set-NetAdapter -Name Storage-101 -VlanID 101 -Confirm:$False
Set-NetAdapter -Name Storage-102 -VlanID 102 -Confirm:$False
```

Below screenshot shows the VLAN configuration on physical adapters.

```
PS C:\ClusterStorage\collect\control> Get-NetAdapterAdvancedProperty |? DisplayName -like VLAN* | ft Name, DisplayName, DisplayValue

Name        DisplayName DisplayValue
----        ----------- ------------
Storage-101 VLAN ID     101
Storage-102 VLAN ID     102
```

Next, we disable VM queue (VMQ) on 1GB network adapters and we set it on 10GB network adapters. When we set the VMQ, we use multiple of 2 because hyperthreading is enabled. We started with a base processor number of 2 because it is recommended to leave the first core (core 0) for other processes.

```
Disable-NetAdapterVMQ -Name Management*

# Core 1, 2 & 3 will be used for network traffic on Storage-101
Set-NetAdapterRSS Storage-101 -BaseProcessorNumber 2 -MaxProcessors 2 -
MaxProcessorNumber 4

#Core 4 & 5 will be used for network traffic on Storage-102
Set-NetAdapterRSS Storage-102 -BaseProcessorNumber 6 -MaxProcessors 2 -
MaxProcessorNumber 8
```

```
Name                                            : Storage-101
InterfaceDescription                            : Mellanox ConnectX-3 Pro Ethernet Adapter #2
Enabled                                         : True
NumberOfReceiveQueues                           : 10
Profile                                         : NUMAStatic
BaseProcessor: [Group:Number]                   : 0:2
MaxProcessor: [Group:Number]                    : 0:4
MaxProcessors                                   : 2
RssProcessorArray: [Group:Number/NUMA Distance] : 0:2/0  0:4/0
IndirectionTable: [Group:Number]                : 0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4
                                                  0:2   0:4    0:2    0:4    0:2    0:4    0:2    0:4

Name                                            : Storage-102
InterfaceDescription                            : Mellanox ConnectX-3 Pro Ethernet Adapter
Enabled                                         : True
NumberOfReceiveQueues                           : 10
Profile                                         : NUMAStatic
BaseProcessor: [Group:Number]                   : 0:6
MaxProcessor: [Group:Number]                    : 0:8
MaxProcessors                                   : 2
RssProcessorArray: [Group:Number/NUMA Distance] : 0:6/0  0:8/0
IndirectionTable: [Group:Number]                : 0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
                                                  0:6   0:8    0:6    0:8    0:6    0:8    0:6    0:8
```

Next we configure Jumbo Frame on each network adapter.

```
Get-NetAdapterAdvancedProperty -Name * -RegistryKeyword "*jumbopacket" | `
Set-NetAdapterAdvancedProperty -RegistryValue 9014
```

```
PS C:\Windows\system32> Get-NetAdapterAdvancedProperty -Name * -RegistryKeyword "*jumbopacket"

Name                      DisplayName                    DisplayValue                   RegistryKeyword
----                      -----------                    ------------                   ---------------
vEthernet (Management-0)  Jumbo Packet                   9014 Bytes                     *JumboPacket
LAN02-0                   Jumbo Packet                   9014 Bytes                     *JumboPacket
LAN01-0                   Jumbo Packet                   9014 Bytes                     *JumboPacket
Storage-102               Jumbo Packet                   9014                           *JumboPacket
Storage-101               Jumbo Packet                   9014                           *JumboPacket
```

Now we can enable RDMA on storage NICs by running the following command:

```
Get-NetAdapter *Storage* | Enable-NetAdapterRDMA
```

The below screenshot is the result of **Get-NetAdapterRDMA**.

```
PS C:\Windows\system32> Get-netAdapterRDMA

Name                      InterfaceDescription                    Enabled
----                      --------------------                    -------
vEthernet (Management-0)  Hyper-V Virtual Ethernet Adapter        False
Storage-102               Mellanox ConnectX-3 Pro Ethernet Adap... True
Storage-101               Mellanox ConnectX-3 Pro Ethernet Adapter True
```

Even if it is useless because we don't have a switch and other connections on 10GB network adapters, we configure DCB:

```
# Turn on DCB
Install-WindowsFeature Data-Center-Bridging

# Set a policy for SMB-Direct
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action 3

# Turn on Flow Control for SMB
Enable-NetQosFlowControl -Priority 3

# Make sure flow control is off for other traffic
Disable-NetQosFlowControl -Priority 0,1,2,4,5,6,7

# Apply policy to the target adapters
Enable-NetAdapterQos -InterfaceAlias "Storage-101"
Enable-NetAdapterQos -InterfaceAlias "Storage-102"

# Give SMB Direct 30% of the bandwidth minimum
New-NetQosTrafficClass "SMB" -Priority 3 -BandwidthPercentage 30 -Algorithm ETS
```

Ok, now that network adapters are configured, we can configure IP addresses and try the communication on the network.

```
New-NetIPAddress -InterfaceAlias "vEthernet (Management-0)" -IPAddress 10.10.0.5 -
PrefixLength 24 -DefaultGateway 10.10.0.1 -Type Unicast | Out-Null
Set-DnsClientServerAddress -InterfaceAlias "vEthernet (Management-0)" -
ServerAddresses 10.10.0.20 | Out-Null

New-NetIPAddress -InterfaceAlias "Storage-101" -IPAddress 10.10.101.5 -PrefixLength
24 -Type Unicast | Out-Null

New-NetIPAddress -InterfaceAlias "Storage-102" -IPAddress 10.10.102.5 -PrefixLength
24 -Type Unicast | Out-Null

# Disable DNS registration of Storage and Cluster network adapter

Set-DNSClient -InterfaceAlias Storage* -RegisterThisConnectionsAddress $False
Set-DNSClient -InterfaceAlias *Cluster* -RegisterThisConnectionsAddress $False
```

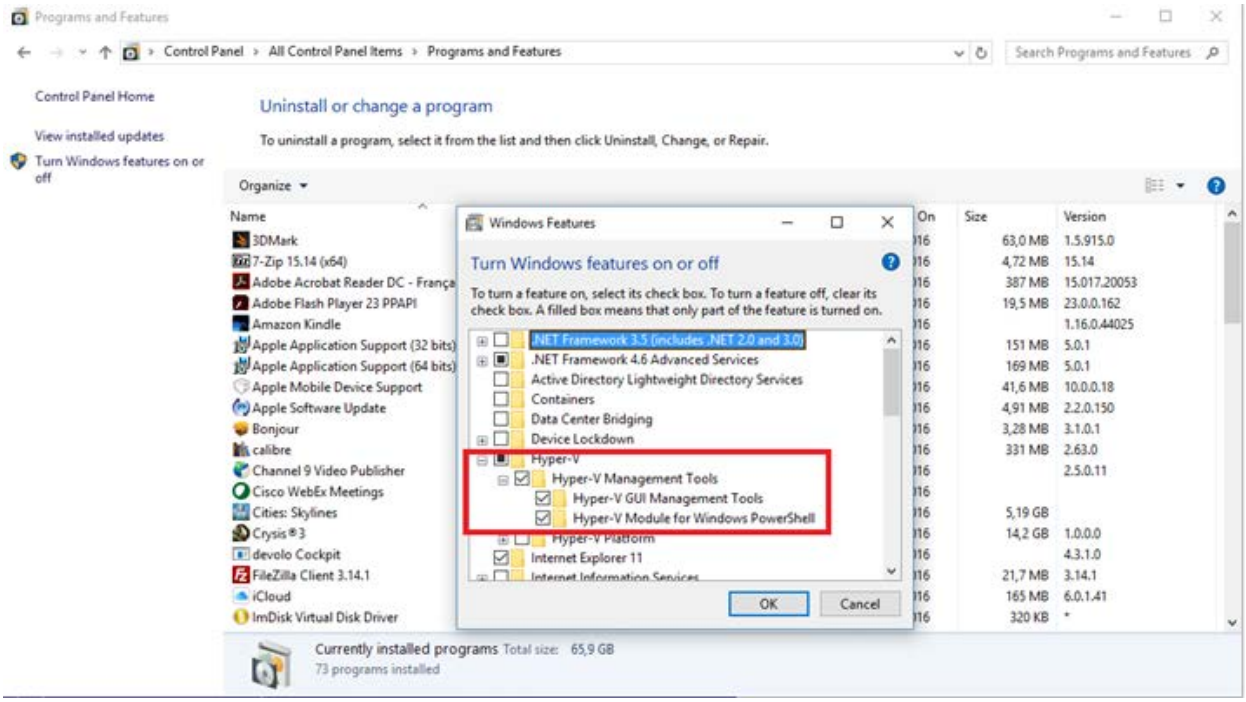
Then try the Jumbo Frame to see if it is working.

```
PS C:\ClusterStorage\collect\control> ping 10.10.101.5 -l 9014

Pinging 10.10.101.5 with 9014 bytes of data:
Reply from 10.10.101.5: bytes=9014 time<1ms TTL=128
Reply from 10.10.101.5: bytes=9014 time<1ms TTL=128
Reply from 10.10.101.5: bytes=9014 time<1ms TTL=128
Reply from 10.10.101.5: bytes=9014 time<1ms TTL=128

Ping statistics for 10.10.101.5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\ClusterStorage\collect\control> ping 10.10.0.1 -l 9014

Pinging 10.10.0.1 with 9014 bytes of data:
Reply from 10.10.0.1: bytes=9014 time=1ms TTL=64
Reply from 10.10.0.1: bytes=9014 time<1ms TTL=64
Reply from 10.10.0.1: bytes=9014 time<1ms TTL=64
Reply from 10.10.0.1: bytes=9014 time<1ms TTL=64

Ping statistics for 10.10.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Now the nodes can communicate with each other on the network. Once you have reproduced these steps on the second node, we can add nodes to the cluster.

Connect to Hyper-V remotely

For ease of management, we manage the cluster from a laptop with PowerShell Remoting. The laptop is not in the domain, so we use a new feature in Hyper-V Manager 2016 console that enables to connect to a remote Hyper-V host as a Workgroup. To connect remotely to Hyper-V, we have installed the console as below:



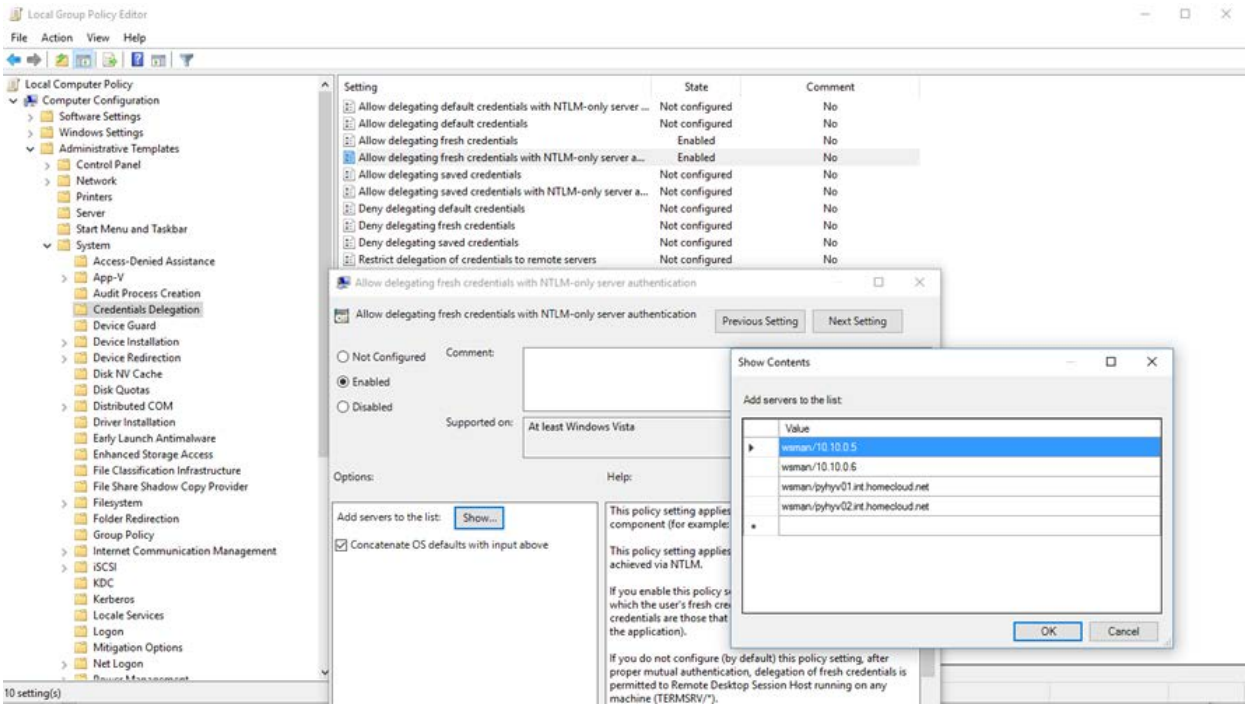
Before being able to connect to Hyper-V remotely, some configurations are required from the server and client perspectives. In both nodes, run the following cmdlets:

```
Enable-WSManCredSSP -Role server
```

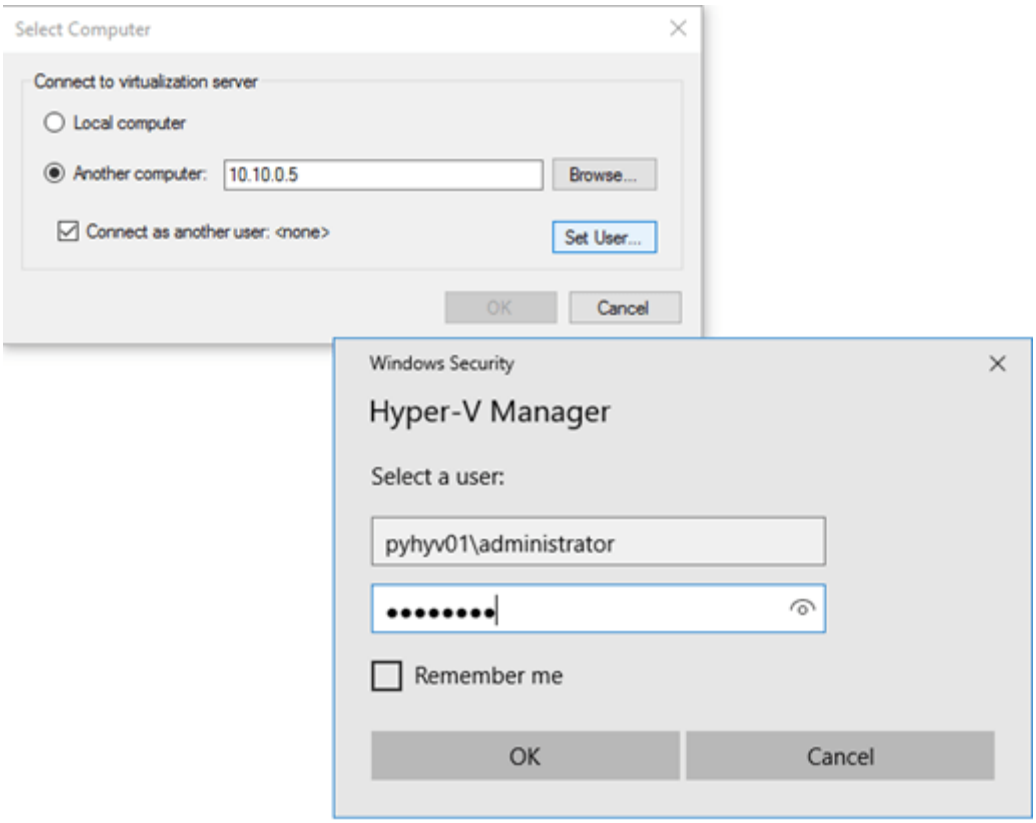
In your laptop, run the following cmdlets (replace **fqdn-of-hyper-v-host** by the future Hyper-V hosts FQDN):

```
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "10.10.0.5"
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "fqdn-of-hyper-v-host"
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "10.10.0.6"
Set-Item WSMan:\localhost\Client\TrustedHosts -Value "fqdn-of-hyper-v-host"

Enable-WSManCredSSP -Role client -DelegateComputer "10.10.0.5"
Enable-WSManCredSSP -Role client -DelegateComputer "fqdn-of-hyper-v-host"
Enable-WSManCredSSP -Role client -DelegateComputer "10.10.0.6"
Enable-WSManCredSSP -Role client -DelegateComputer "fqdn-of-hyper-v-host"
```

Then, run **gpedit.msc** and configure the following policy:



Now you can leverage the new Hyper-V manager capability which enable to use an alternative credential to connect to a remote Hyper-V.

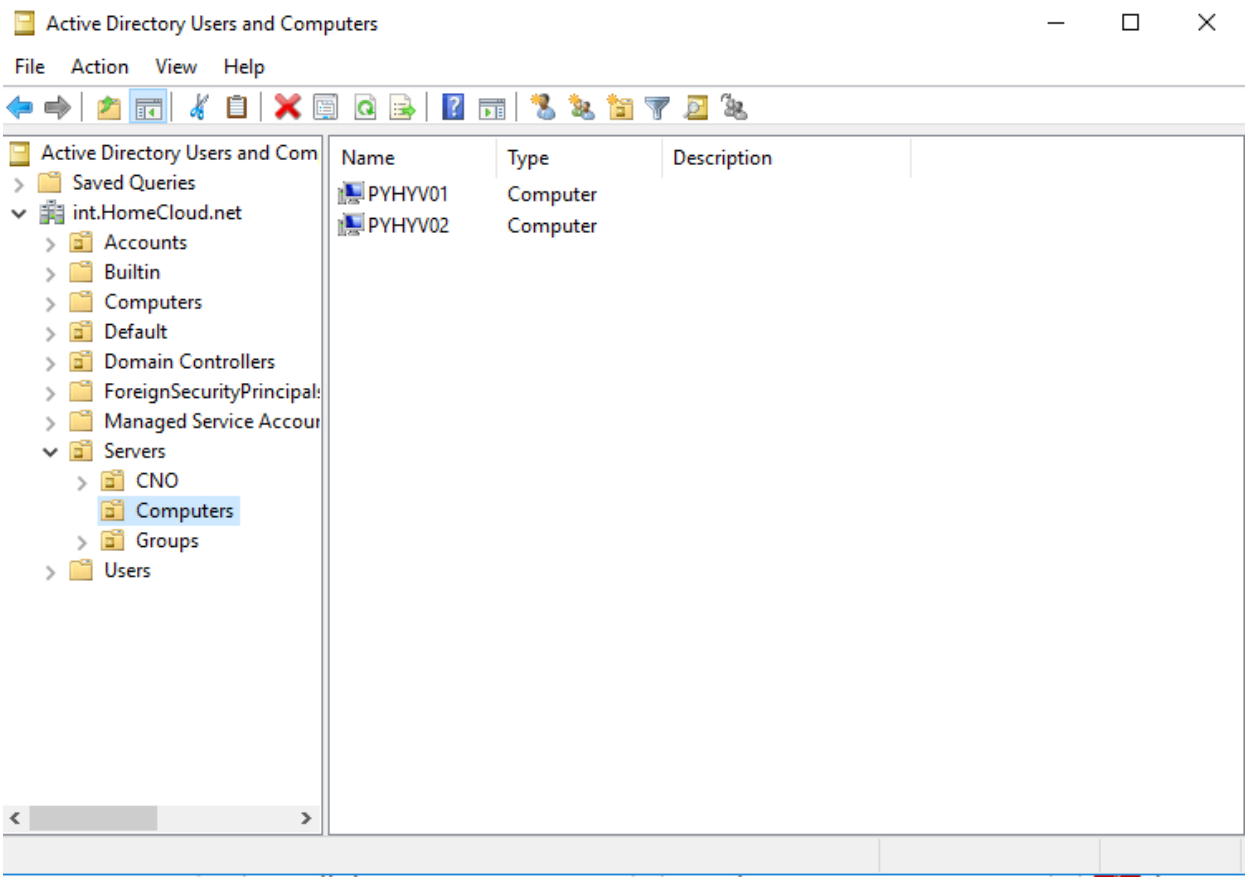


Add nodes to domain

To add both nodes to the domain, we run the following cmdlets from the laptop:

```
Enter-PSSession -ComputerName 10.10.0.5 -Credential pyhyv01\administrator
$domain = "int.homecloud.net"
$password = "P@$$w0rd" | ConvertTo-SecureString -asPlainText -Force
$username = "$domain\administrator"
$credential = New-Object
System.Management.Automation.PSCredential($username,$password)
Add-Computer -DomainName $domain -Credential $credential -OUPath
"OU=Computers,OU=Servers,DC=int,DC=HomeCloud,DC=net" -Restart
```

Wait that pyhyv01 has rebooted and run the following cmdlet on pyhyv02. Now you can log on pyhyv01 and pyhyv02 with domain credential. You can install Domain Services RSAT on the laptop to parse the Active Directory.



2-node hyper-converged cluster deployment

Now that Active Directory is available, we can deploy the cluster. First, we test the cluster to verify that all is ok:

```
Enter-PSSession -ComputerName pyhyv01.int.homecloud.net -credential
inthomecloud\administrator
Test-Cluster pyhyv01, pyhyv02 -Include "Storage Spaces
Direct",Inventory,Network,"System Configuration"
```

Check the report if there is any issue with the configuration. If the report looks good, you can move forward and run the following cmdlets:

```
# Create the cluster
New-Cluster -Name Cluster-Hyv01 -Node pyhyv01,pyhyv02 -NoStorage -StaticAddress
10.10.0.10
```

Once the cluster is created, we set a Cloud Witness in order that Azure has a vote for the quorum.

```
# Add a cloud Witness (require Microsoft Azure account)
Set-ClusterQuorum -CloudWitness -Cluster Cluster-Hyv01 -AccountName
"<StorageAccount>" -AccessKey "<AccessKey>"
```

Then we configure the network name in the cluster:

```
#Configure network name
(Get-ClusterNetwork -Name "Cluster Network 1").Name="Storage-102"
(Get-ClusterNetwork -Name "Cluster Network 2").Name="Storage-101"
(Get-ClusterNetwork -Name "Cluster Network 3").Name="Management-0"
```

Next we configure the Node Fairness to run each time a node is added to the cluster and every 30mn. When the CPU of a node will be utilized at 70%, the node fairness will balance the VM across other nodes.

```
# Configure Node Fairness
(Get-Cluster).AutoBalancerMode  = 2
(Get-Cluster).AutoBalancerLevel = 2
```

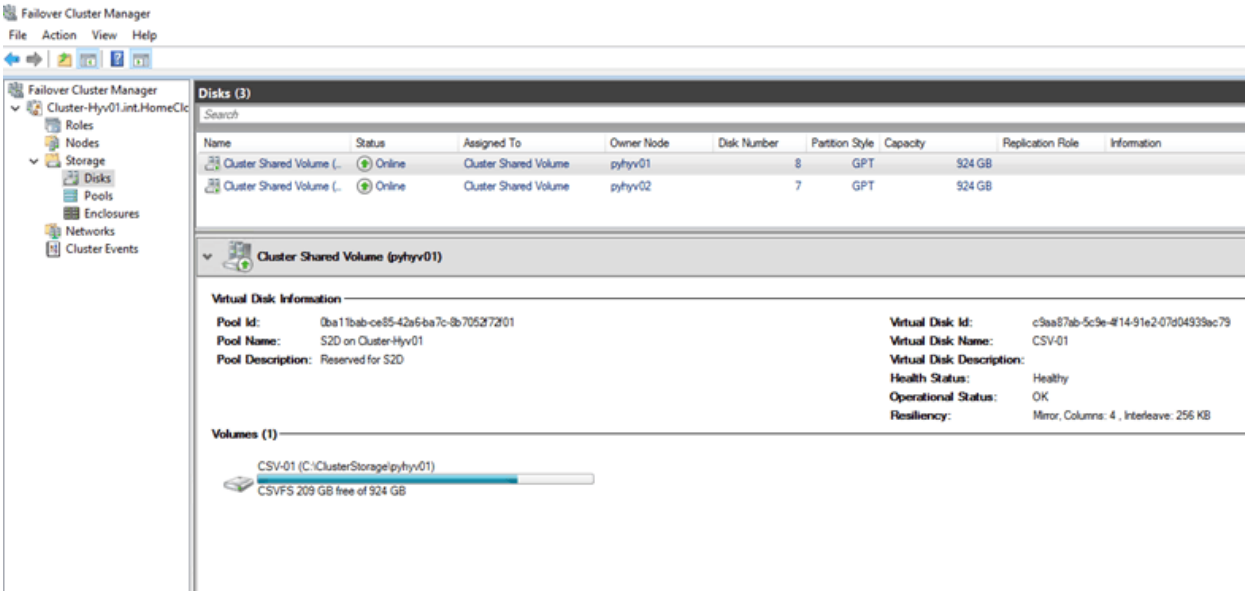
To finish with the cluster, we have to enable Storage Spaces Direct, and create volume. But before, we run the following script to clean and wipe out the disks:

Note: Be careful that each disk you feed to **clear-disk** is indeed the one you want to wipe.

```
icm (Get-Cluster -Name Cluster-Hyv01 | Get-ClusterNode) {
    Update-StorageProviderCache

    Get-StoragePool |? IsPrimordial -eq $false | Set-StoragePool -IsReadOnly:$false
-ErrorAction SilentlyContinue

    Get-StoragePool |? IsPrimordial -eq $false | Get-VirtualDisk | Remove-
VirtualDisk -Confirm:$false -ErrorAction SilentlyContinue

    Get-PhysicalDisk | Reset-PhysicalDisk -ErrorAction SilentlyContinue

    Get-Disk |? Number -ne $null |? IsBoot -ne $true |? IsSystem -ne $true |?
PartitionStyle -ne RAW |% {

        $_ | Set-Disk -isoffline:$false

        $_ | Set-Disk -isreadonly:$false

        $_ | Clear-Disk -RemoveData -RemoveOEM -Confirm:$false

        $_ | Set-Disk -isreadonly:$true

        $_ | Set-Disk -isoffline:$true

    }

    Get-Disk |? Number -ne $null |? IsBoot -ne $true |? IsSystem -ne $true |?
PartitionStyle -eq RAW | Group -NoElement -Property FriendlyName

} | Sort -Property PsComputerName,Count
```

Now we can enable Storage Spaces Direct and create volumes:

```
Enable-ClusterS2D

New-Volume -StoragePoolFriendlyName "S2D*" -FriendlyName CSV-01 -FileSystem
CSVFS_ReFS -Size 650GB

New-Volume -StoragePoolFriendlyName "S2D*" -FriendlyName CSV-02 -FileSystem
CSVFS_ReFS -Size 650GB
```



Finally, we rename volume in c:\ClusterStorage by their names in the cluster:

```
Rename-Item -Path C:\ClusterStorage\volume1\ -NewName CSV-01
Rename-Item -Path C:\ClusterStorage\volume2\ -NewName CSV-02
```

Final Hyper-V configuration

First, we set default VM and virtual disk folders:

```
Set-VMHOST –computername pyhyv01 –virtualharddiskpath 'C:\ClusterStorage\CSV-01'
Set-VMHOST –computername pyhyv01 –virtualmachinepath 'C:\ClusterStorage\CSV-01'
Set-VMHOST –computername pyhyv02 –virtualharddiskpath 'C:\ClusterStorage\CSV-02'
Set-VMHOST –computername pyhyv02 –virtualmachinepath 'C:\ClusterStorage\CSV-02'
```

Then we configure the Live-Migration protocol and the number of simultaneous migration allowed:

```
Enable-VMMigration –Computername pyhyv01, pyhyv02
Set-VMHost -MaximumVirtualMachineMigrations 4 `
           –MaximumStorageMigrations 4 `
           –VirtualMachineMigrationPerformanceOption SMB `
           -ComputerName pyhyv01,pyhyv02
```

Next, we add Kerberos delegation to configure Live-Migration in Kerberos mode:

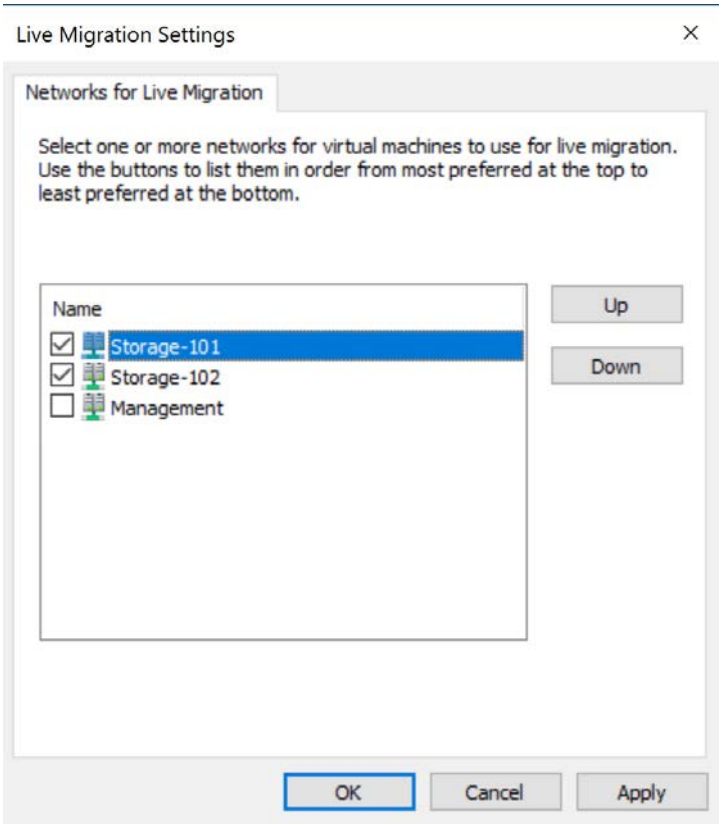
```
Enter-PSSession -ComputerName VMADS01.int.homecloud.net
$HyvHost = "pyhyv01"
$Domain = "int.homecloud.net"

Get-ADComputer pyhyv02 | Set-ADObject -Add @{"msDS-AllowedToDelegateTo"="Microsoft
Virtual System Migration Service/$HyvHost.$Domain",
"cifs/$HyvHost.$Domain","Microsoft Virtual System Migration Service/$HyvHost",
"cifs/$HyvHost"}
Set-ADAccountControl $(Get-ADComputer pyhyv02) -TrustedTOAuthForDelegation $True

$HyvHost = "pyhyv02"

Get-ADComputer pyhyv01 | Set-ADObject -Add @{"msDS-AllowedToDelegateTo"="Microsoft
Virtual System Migration Service/$HyvHost.$Domain",
"cifs/$HyvHost.$Domain","Microsoft Virtual System Migration Service/$HyvHost",
"cifs/$HyvHost"}
Set-ADAccountControl $(Get-ADComputer pyhyv01) -TrustedTOAuthForDelegation $True

Exit
```

Then we set authentication of Live-Migration to Kerberos.

```
Set-VMHost –Computername pyhyv01, pyhyv02 `
           –VirtualMachineMigrationAuthenticationType Kerberos
```

Next, we configure the Live-Migration network priority:

```
# Configure Live Migration Networks
Get-ClusterResourceType -Cluster Cluster-Hyv01 -Name "Virtual Machine" | `
Set-ClusterParameter -Name MigrationExcludeNetworks -Value
([String]::Join(";",(Get-ClusterNetwork -Cluster $Cluster | Where-Object {$_.Name -
notlike "Storage*" }).ID))
```



Finally, we configure the cache size of the CSV to 512MB:

```
(Get-Cluster).BlockCacheSize = 512
```

Deploy a benchmark tool

VM Fleet is a collection of scripts that enables to deploy virtual machines which perform I/O to stress the underlying storage system. To achieve I/O, the VMs uses DiskSpd which is a Microsoft tool.

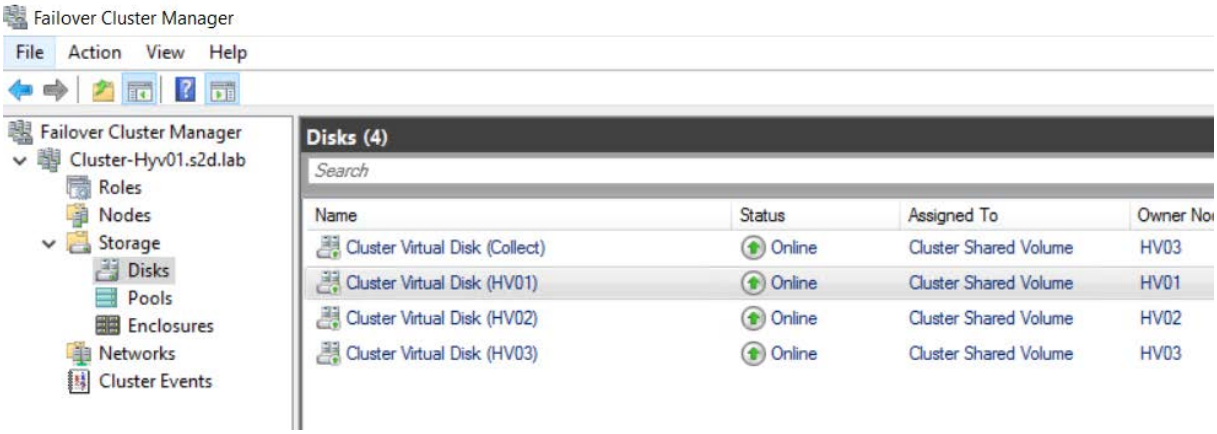
When you implement an infrastructure based on Hyper-V, you usually want to get the maximum IOPS and MB/s that your storage can deliver. This tool helps you to get this information by stressing your storage. In this topic, we will see how to deploy a VM Fleet to benchmark the storage system.

Cluster preparation for VM Fleet

To deploy a VM Fleet, you need several Cluster Shared Volume (CSV) in the cluster. First you need a CSV per node. Be careful that the name is **Cluster Virtual Disk (<Node Name>)** as below. Otherwise some scripts will fail. Moreover, you need another CSV called **Cluster Virtual Disk (Collect).** This CSV will store VM Fleet scripts, flags, result and the gold image (VHDX).

To create the collect volume in my environment, I have run the following cmdlet:

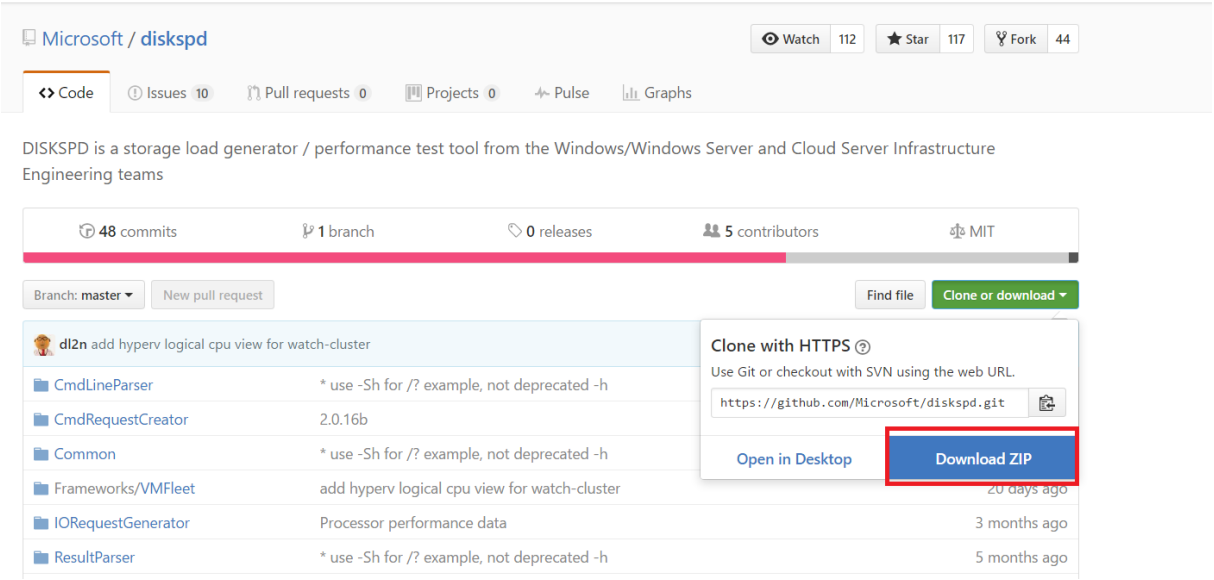
```
New-Volume -StoragePoolFriendlyName "S2D*" -FriendlyName Collect -FileSystem
CSVFS_ReFS -Size 50G
```



Then rename the folder in **C:\ClusterStorage** with the CSV Name. Below you can find the script that you can run from a cluster node to rename folder regarding CSV name:

```
Rename-Item -Path C:\ClusterStorage\Volume1 -NewName HV01
Rename-Item -Path C:\ClusterStorage\Volume1 -NewName HV02
Rename-Item -Path C:\ClusterStorage\Volume1 -NewName HV03
Rename-Item -Path C:\ClusterStorage\Volume1 -NewName HV04
```

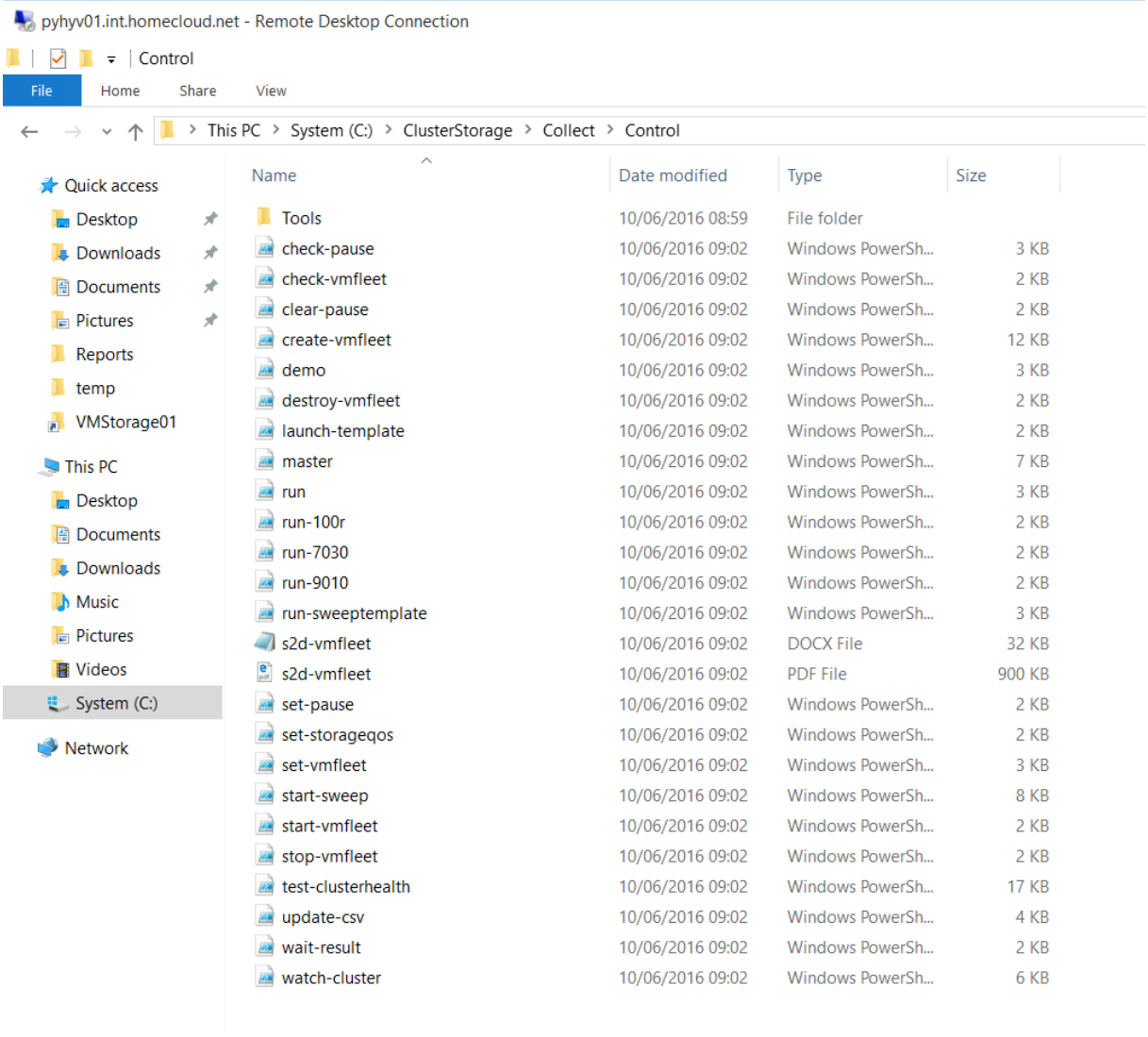
Next download the VM Fleet from Github. Click on **Clone or download** and choose **Download ZIP** as below. Then extract the ZIP in C:\temp in the first cluster node.



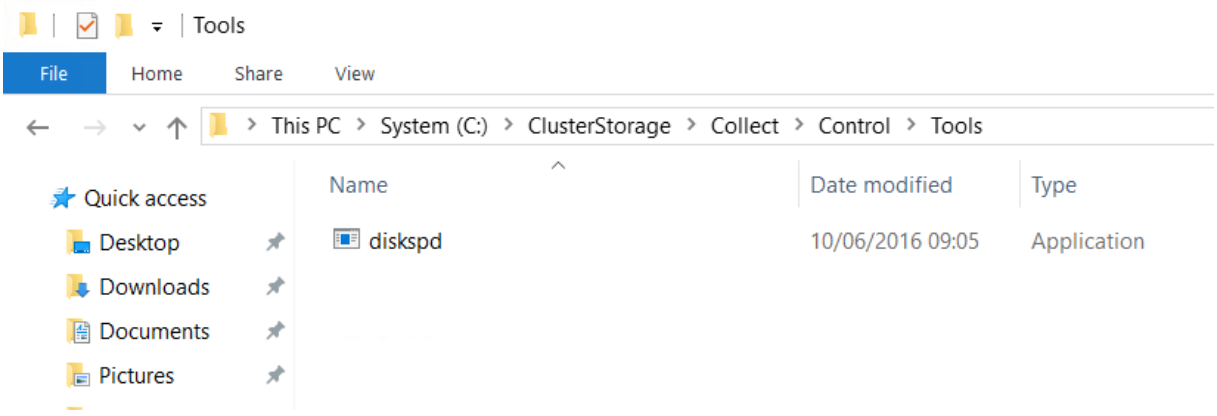
Then you can install VM Fleet in the **collect** CSV. Because scripts come from Internet (untrusted source), you need to change the PowerShell execution policy:

```
#Change the PowerShell execution policy
Set-ExecutionPolicy unrestricted
# Prepare the cluster for VM Fleet
.\install-vmfleet.ps1 -Source C:\temp\diskspd-master\Frameworks\VMFleet
```

Once the script is finished, you can navigate to **C:\ClusterStorage\Collect\Control**. You should have something as below:



Then download DiskSpd and paste it in **C:\ClusterStorage\Collect\Control\Tools**.



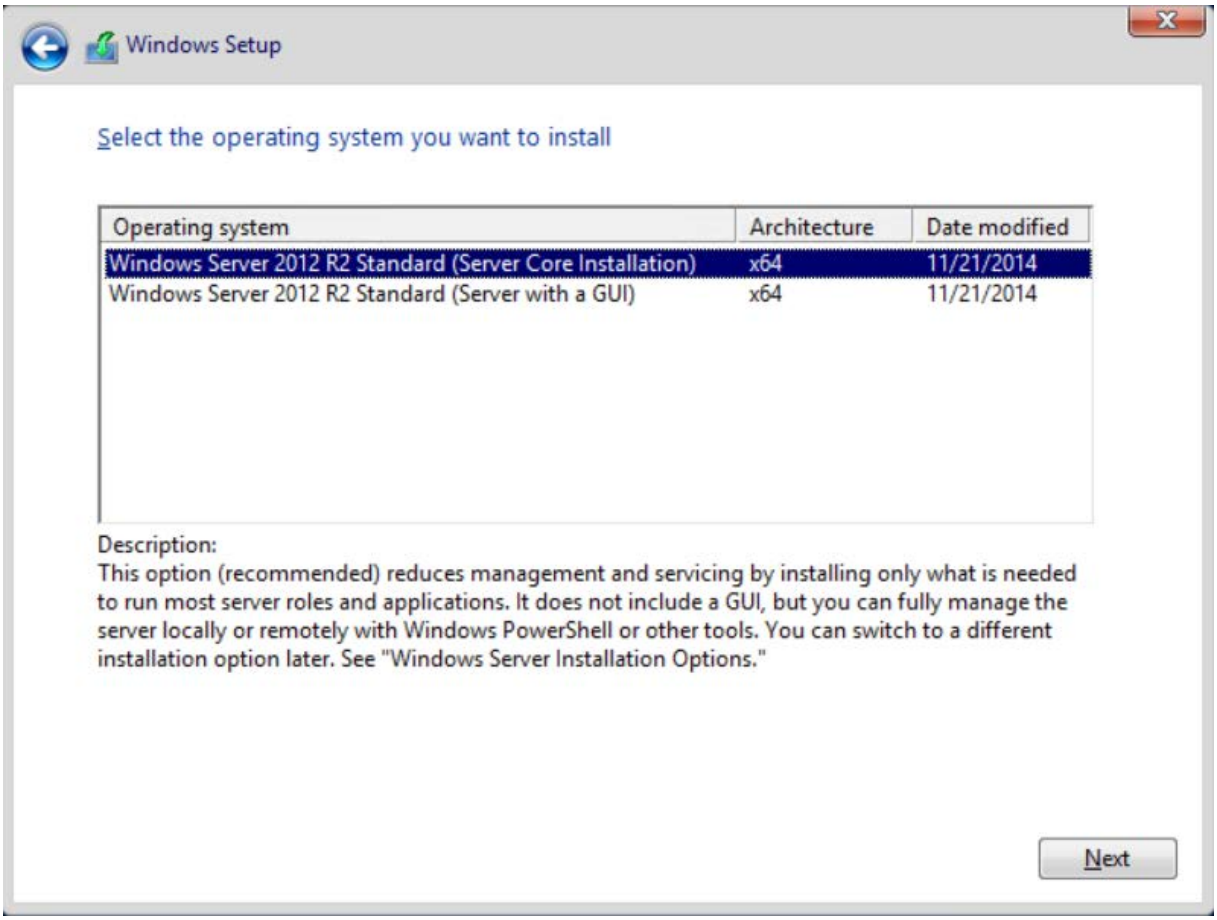
From this moment, your cluster preparation is finished. Now a Windows Server 2012 R2 or Windows Server 2016 gold image is required for virtual machines.

Prepare the Gold image

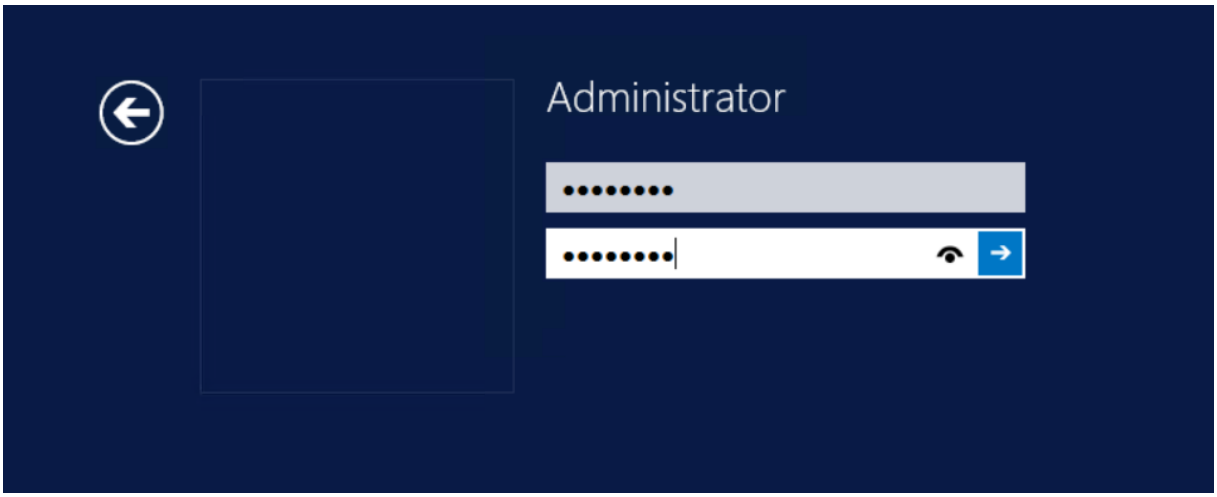
First, we need to create a virtual machine to prepare the image. To create the VM, we run the following script. The VM will be stored in C:\VirtualMachines of the node. To run this script, you need also the Windows Server 2012 R2 ISO stored in C:\temp. You can change the path to reflect your environment.

```
$VMName = "GOldVMFleet"
# Create Gen 2 VM with dynamic memory, autostart action to 0s and auto stop action
set. 2vCPU
New-VM -Generation 2 -Name $VMName -SwitchName SW-10G-CNA -NoVHD -
MemoryStartupBytes 2048MB -Path C:\VirtualMachines
Set-VM -Name $VMName `
       -ProcessorCount 2 `
       -DynamicMemory `
       -MemoryMinimumBytes 1024MB `
       -MemoryMaximumBytes 4096MB `
       -MemoryStartupBytes 2048MB `
       -AutomaticStartAction Start `
       -AutomaticStopAction ShutDown `
       -AutomaticStartDelay 0 `
       -AutomaticCriticalErrorAction None `
       -CheckpointType Production
 # Create and add a 60GB dynamic VHDX to the VM
New-VHD -Path C:\VirtualMachines\$VMName\GoldVMFleet.vhdx -SizeBytes 40GB -Fixed
Add-VMHardDiskDrive -VMName $VMName -Path
C:\VirtualMachines\$VMName\GoldVMFleet.vhdx
 # Rename the network adapter
Get-VMNetworkAdapter -VMName $VMName | Rename-VMNetworkAdapter -NewName Management-
0
 # Add a DVD drive with W2016 ISO
Add-VMDvdDrive -VMName $VMName
# Mount the ISO of Windows Server 2012 R2
Set-VMDvdDrive -VMName $VMName -Path
"C:\temp\en_windows_server_2012_r2_with_update_x64_dvd_6052708.iso"
```

Once the VM is created, you can start it to run the Windows Server installation. Make sure to choose **Server Core Installation**.



Once Windows Server is installed, you are asked for a password. Keep this password, it will be useful later.

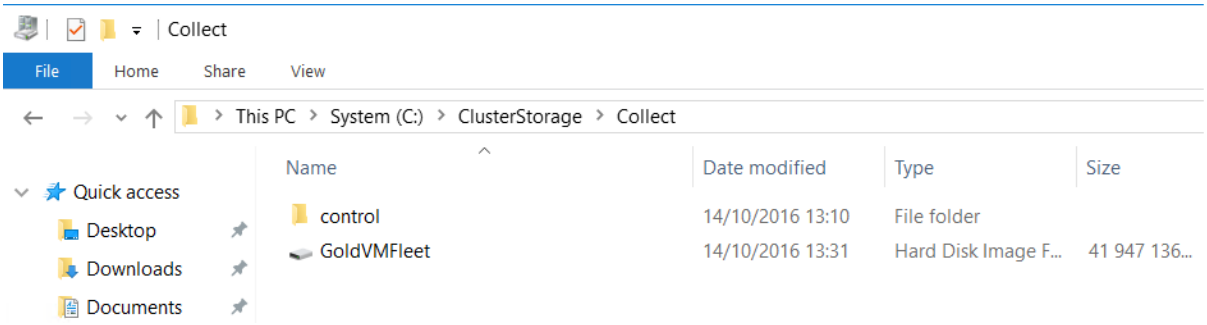


Once the server is ready, you can shut down the VM. We don't need to sysprep the VM.

```
Shutdown -s -t 0
```

To finish, copy the **VHDX** to **c:\ClusterStorage\Collect**:

```
Copy-Item c:\VirtualMachines\GoldVMFLeet\GoldVMFleet.vhdx c:\ClusterStorage\Collect
```



Now you can delete the GoldVMFleet VM.

Deploy the VM Fleet

Now that the gold image is ready, we can deploy X VMs in the cluster. To create these VMs, we use the script **Create-VMFleet.ps1** located in **C:\ClusterStorage\Collect\Control**.

To deploy the VM fleet, we run the following script:

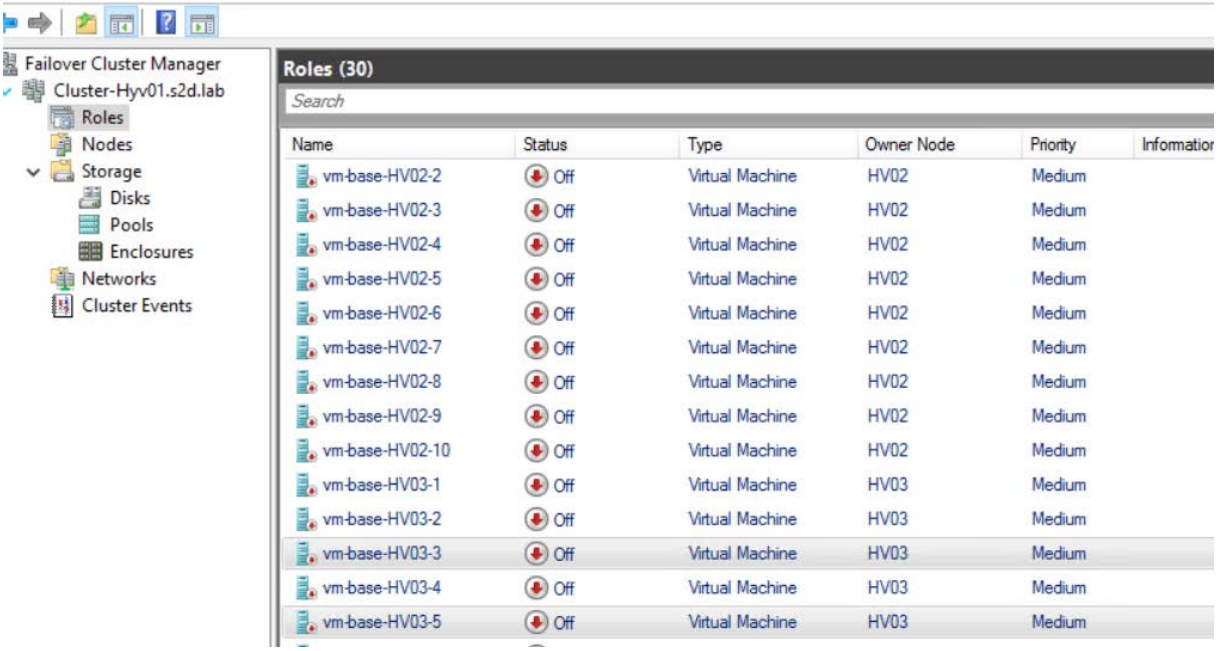
```
.\create-vmfleet.ps1 -basevhd "C:\ClusterStorage\Collect\GoldVMFleet.vhdx" -vms 30
-adminpass <VM password> -connectuser <Host login> -connectpass <Host Password>
```

This script means that we will deploy 30 VMs among all CSVs. Since we have three CSVs (the Collect CSV is not counted), so we will have 15 VMs per CSV.

```
create vm vm-base-HV03-1 @ metadata path C:\ClusterStorage\HV03\vm-base-HV03-1 with vhd C:\ClusterStorage\HV03\vm-base-H
V03-1.vhdx
create vm vm-base-HV01-1 @ metadata path C:\ClusterStorage\HV01\vm-base-HV01-1 with vhd C:\ClusterStorage\HV01\vm-base-H
V01-1.vhdx
create vm vm-base-HV02-1 @ metadata path C:\ClusterStorage\HV02\vm-base-HV02-1 with vhd C:\ClusterStorage\HV02\vm-base-H
V02-1.vhdx
```

Once the VM fleet is deployed, we run the following cmdlet to set the VMs to two vCPU and 8GB of static memories:

```
Get-VM -ComputerName Cluster-Hyv01 |set-VM -ProcessorCount 2 -MemoryStartupBytes
8GB -StaticMemory
```



Finally, we check the health of the cluster to verify that all is ok to run the stress test:

```
.\Test-ClusterHealth.ps1
```

```
PS C:\ClusterStorage\Collect\control> .\test-clusterhealth.ps1
******************** Basic Health Checks (2,7s)
All cluster nodes Up
All operational pools Healthy
******************** Clusport Device Symmetry Check (2,2s)
********** Total
Pass with 35 per node
********** Disk Type
Pass with 32 per node
********** Solid/Non-Rotational Media
Pass with 3 per node
********** Enclosure Type
Pass with 3 per node
********** Virtual
Pass with none on any node
******************** Enclosure View Symmetry Check (3,3s)
********** Total
Pass with 3 per node
******************** Operational Issues and Storage Jobs (4,1s)
No storage rebuild or regeneration jobs are active
******************** Physical Disk Health (2,2s)
All physical disks are in normal auto-select or journal state
******************** Physical Disk View Symmetry Check (3,6s)
********** Total
Pass with 32 per node
******************** RDMA Adapters Symmetry Check (2,7s)
********** Total
Pass with none on any node
********** Operational
Pass with none on any node
********** Up
Pass with none on any node
******************** SMB Connectivity Error Check (2,7s)

PSComputerName RDMA Last5Min RDMA LastDay RDMA LastHour TCP Last5Min TCP LastDay TCP LastHour
-------------- ------------ ------------ ------------- ------------ ----------- ------------
HV01                      0           94            40            0         197           77
HV02                      0           84            40            0         182          101
HV03                      0          116            70            0         142           43


******************** SMB CSV Multichannel Symmetry Check (2,2s)
********** Total
Pass with 6 per node
********** RDMA Capable
Pass with 6 per node
********** Selected & Non-Failed
Pass with 6 per node
******************** SMB SBL Multichannel Symmetry Check (2,3s)
********** Total
Pass with 6 per node
********** RDMA Capable
Pass with 6 per node
********** Selected & Non-Failed
Pass with 6 per node
******************** Virtual Disk Health (2,2s)
All operational virtual disks Healthy
```

Play with the VM Fleet

When the VM Fleet is deployed, all the VM are stopped. To start all VM you can run the following cmdlet:

```
.\Start-VMFleet.ps1
```

```
PS C:\ClusterStorage\Collect\Control> .\start-vmfleet.ps1

Name            OwnerNode State PSComputerName
----            --------- ----- --------------
vm-base-HV02-1  HV02            HV02
vm-base-HV01-1  HV01            HV01
vm-base-HV03-1  HV03            HV03
vm-base-HV02-10 HV02            HV02
vm-base-HV01-10 HV01            HV01
vm-base-HV03-10 HV03            HV03
vm-base-HV02-2  HV02            HV02
vm-base-HV03-2  HV03            HV03
vm-base-HV01-2  HV01            HV01
vm-base-HV03-3  HV03            HV03
vm-base-HV02-3  HV02            HV02
vm-base-HV01-3  HV01            HV01
vm-base-HV03-4  HV03            HV03
vm-base-HV01-4  HV01            HV01
vm-base-HV02-4  HV02            HV02
vm-base-HV03-5  HV03            HV03
vm-base-HV02-5  HV02            HV02
vm-base-HV01-5  HV01            HV01
vm-base-HV03-6  HV03            HV03
vm-base-HV02-6  HV02            HV02
vm-base-HV01-6  HV01            HV01
vm-base-HV03-7  HV03            HV03
vm-base-HV02-7  HV02            HV02
vm-base-HV01-7  HV01            HV01
vm-base-HV02-8  HV02            HV02
vm-base-HV03-8  HV03            HV03
vm-base-HV01-8  HV01            HV01
vm-base-HV02-9  HV02            HV02
vm-base-HV03-9  HV03            HV03
vm-base-HV01-9  HV01            HV01
```

When the VM are started, they are in state **PAUSE IN FORCE**. It is because the VM check the folder **C:\ClusterStorage\Collect\Control\Flags.** In this folder, a file called **pause** is created by default to force the stress test to be paused.

When a stress test is launched, the pause is clear and flags folder is filled with a **go** file and the test to run. The VMs see the **go** flag and run the test specified.

You can clear and set a pause with the following script:

```
# set a pause
./Set-Pause.ps1
# Clear pause
./Clear-pause.ps1
```

When you want to stop all VMs you can run the following script:

```
./Stop-VMFleet.ps1
```

When you have finished to benchmark your storage, you can destroy the VM fleet:

```
./Destroy-VMFleet.ps1
```

Run a test

To start a sweep, you can use the **Start-sweep.ps1** script. This script accepts the following parameters. These parameters are passed to **DiskSpd** to run the test.

- b: list of buffer sizes (KiB)
- t: list of threads counts
- o: list of outstanding IO counts
- w: list of write ratios
- p: list of patterns (random: r, sequential: s, sequential interlocked: si)
- warm: duration of pre-measurement warmup (seconds)
- d: duration of measured interval (seconds)

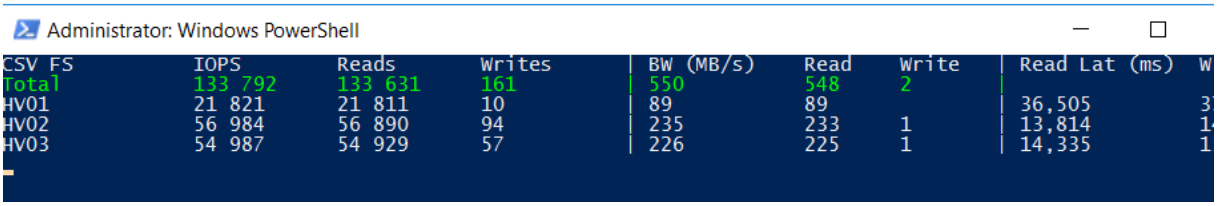
- cool: duration of post-measurement cooldown (seconds)

For example, I run the following script to launch a 100% read test:

```
.\Start-Sweep.ps1 -b 4 -t 2 -o 40 -w 0 -d 300
```

```
PS C:\ClusterStorage\Collect\control> .\start-sweep.ps1 -b 4 -t 2 -o 40 -w 0 -d 300
---
RUN SPEC @ 14/10/2016 18:06:11
        o = 40
        d = 300
        AddSpec = base
        p = r
        b = 4
        t = 2
        w = 0
        Cool = 60
        Warm = 60
Generating new runfile @ 14/10/2016 18:06:11
START Go Epoch: 0 @ 14/10/2016 18:06:11
CLEAR PAUSE @ 14/10/2016 18:06:11
```

When the test is running, you can launch the script **Watch-Cluster.ps1**. (HV01 seems to be tired).



When the test is finished, you can find the result in **C:\ClusterStorage\Collect\Control\result.** After the test, you can erase the content of this directory. You can also set a pause because sometimes the pause is not well set and the second test might fail.

Troubleshooting

Work with Health Service

You can get information about cluster usage by using the following cmdlet:

```
Get-StorageSubSystem *Cluster* | Get-StorageHealthReport
```

```
PS C:\Windows\system32> get-StorageSubSystem *Cluster* | Get-StorageHealthReport
CPUUsageAverage                    :    1.6 %
CapacityPhysicalPooledAvailable    :   1.09 TB
CapacityPhysicalPooledTotal        :   3.64 TB
CapacityPhysicalTotal              :   3.64 TB
CapacityPhysicalUnpooled           :      0 B
CapacityVolumesAvailable           : 864.18 GB
CapacityVolumesTotal               :   1.27 TB
IOLatencyAverage                   :   6.13 ms
IOLatencyRead                      :      0 ns
IOLatencyWrite                     :   6.13 ms
IOPSRead                           :      0 /S
IOPSTotal                          :   9.04 /S
IOPSWrite                          :   9.04 /S
IOThroughputRead                   :      0 B/S
IOThroughputTotal                  :  44.18 KB/S
IOThroughputWrite                  :  44.18 KB/S
MemoryAvailable                    : 202.16 GB
MemoryTotal                        :    128 GB
```

To show real-time alerts, you can use the following cmdlet:

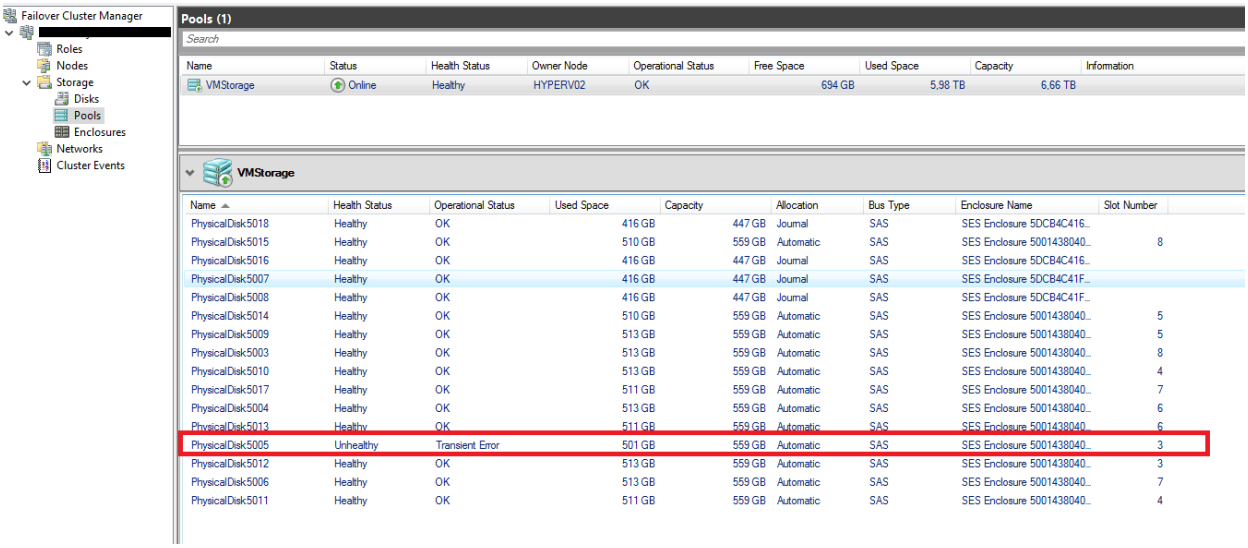
```
Get-StorageSubSystem *Cluster* | Debug-StorageSubSystem
```

```
PS C:\Windows\system32> Get-StorageSubSystem *Cluster* | Debug-StorageSubSystem


   Severity: Minor


Reason         : The server 'pyhyv01' has missing network adapter(s) connected to cluster network 'Storage-101'.
Recommendation : Connect the server to the missing cluster network.
Location       : Server 'pyhyv01', ASUSTeK COMPUTER INC., Z9PA-U8 Series, To be filled by O.E.M., Network Adapter
                 connected to 'Storage-101'
Description    : Network adapter connected to cluster network 'Storage-101'

Reason         : The network interface 'Mellanox ConnectX-3 Pro Ethernet Adapter #2' has become disconnected.
Recommendation : Reconnect the network cable.
Location       : Server 'pyhyv02', ASUSTeK COMPUTER INC., Z9PA-U8 Series, To be filled by O.E.M., Network Adapter:
                 'Mellanox ConnectX-3 Pro Ethernet Adapter #2'
Description    : Manufacturer 'Mellanox Technologies Ltd.', Product Name Mellanox ConnectX-3 Pro Ethernet Adapter
```

Detect and change a failed physical disk

Identify the failed physical disk

When deploying VMFleet, we noticed both virtual disks in a degraded state. So, we checked the job by running **Get-StorageSubSystem *Cluster* | Get-StorageJob**. Then we opened the Storage Pool and we have seen the following:



So, it seems this physical disk was not healthy and we decided to change it. First, we ran the following cmdlet because my trust in Failover Cluster Manager is limited:

```
Get-StoragePool *S2D* | Get-PhysicalDisk
```

```
PS C:\Windows\system32> Get-StoragePool *S2D* | Get-PhysicalDisk

FriendlyName         SerialNumber   CanPool OperationalStatus           HealthStatus Usage           Size
------------         ------------   ------- -----------------           ------------ -----           ----
HP EH0600JDYTL       0XHVG9AP       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHV8P7P       False   OK                          Healthy      Auto-Select 558.75 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805010 False   OK                          Healthy      Journal        447 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805055 False   OK                          Healthy      Journal        447 GB
HP EH0600JDYTL       0XHVB72P       False   OK                          Healthy      Auto-Select 558.75 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805014 False   OK                          Healthy      Journal        447 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805011 False   OK                          Healthy      Journal        447 GB
HP EH0600JDYTL       0XHVJT1P       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVJVZP       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVGA3P       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHV9M9P       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVJW5P       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHTXM2P       False   OK                          Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVS1ZP       False   {Transient Error, IO Error} Unhealthy    Auto-Select 558.75 GB
```

Then we added the physical disk object into a PowerShell variable (called $Disk) to manipulate the disk. You can change the **OperationalStatus** filter by another thing while you get the right disk.

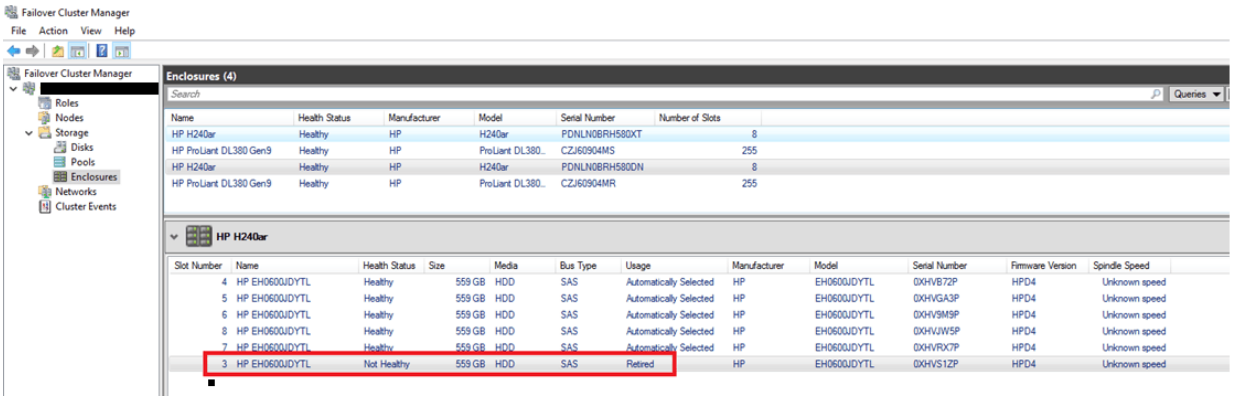
```
$Disk = Get-PhysicalDisk |? OperationalStatus -Notlike ok
```

```
PS C:\Windows\system32> $Disk = get-physicaldisk |? OperationalStatus -notlike ok
PS C:\Windows\system32> $Disk

FriendlyName   SerialNumber CanPool OperationalStatus           HealthStatus Usage           Size
------------   ------------ ------- -----------------           ------------ -----           ----
HP EH0600JDYTL 0XHVS1ZP     False   {Transient Error, IO Error} Unhealthy    Auto-Select 558.75 GB
```

Retire and physically identify storage device

Next, we set the usage of this disk to **Retired** to stop writing on this disk and avoid data loss.

```
Set-PhysicalDisk -InputObject $Disk -Usage Retired
```



We tried to remove the physical disk from the Storage Pool. It seems the physical disk is in bad state. We can't remove it from the pool. So, we decided to change it anyway.

```
PS C:\ClusterStorage> Get-StoragePool *S2D* | Remove-PhysicalDisk -PhysicalDisks $Disk

Confirm
Are you sure you want to perform this action?
Removing a physical disk will cause problems with the fault tolerance capabilities of the following storage pool: "S2D on Cluster-Hyv01".
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "Y"): y
Remove-PhysicalDisk : One or more storage devices are unresponsive.

Extended information:
One or more physical disks encountered an error during removal from the storage pool.

Physical Disks:
{c91b3052-b7cd-f283-893c-fde31269ea38}: The storage device is unresponsive.

Activity ID: {a8ca41f4-f3b3-4b01-b51d-ccd7e2560e32}
At line:1 char:25
+ Get-StoragePool *S2D* | Remove-PhysicalDisk -PhysicalDisks $Disk
+                         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (StorageWMI:ROOT/Microsoft/..._StorageCmdlets) [Remove-PhysicalDisk], CimException
    + FullyQualifiedErrorId : StorageWMI 40026,Remove-PhysicalDisk
```

We ran the following cmdlet to turn on the storage device LED to identify it easily in the datacenter:

```
Get-PhysicalDisk |? OperationalStatus -Notlike OK | Enable-
PhysicalDiskIdentification
```

```
PS C:\ClusterStorage> get-physicaldisk |? OperationalStatus -notlike OK | Enable-PhysicalDiskIdentification
PS C:\ClusterStorage> _
```

Next, we move to the server room and as you can see in the below photo, the LED is turned on. So, we changed this disk.

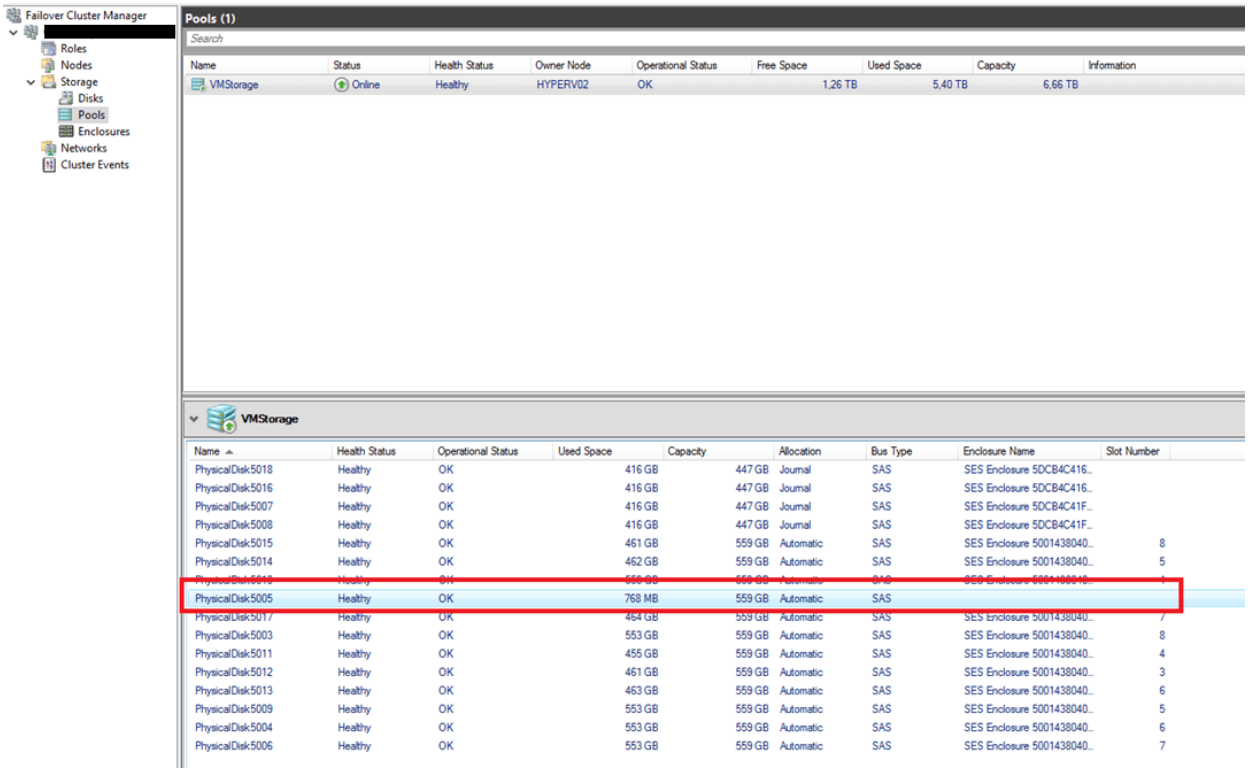
Once the disk is replaced, you can turn off the LED:

```
Get-PhysicalDisk |? OperationalStatus -like OK | Disable-PhysicalDiskIdentification
```

Add physical disk to storage pool

Before rebooting the server, the physical disk can't show its enclosure name. The disk automatically joined the Storage Pool but without enclosure information. So, you have to reboot the server to get the right information.

Storage Spaces Direct spread automatically the data across the new disk. This process took almost 30 minutes.



Sometime the physical disk doesn't join automatically the Storage Pool. So, you can run the following cmdlet to add the physical disk to the Storage Pool.

```
PS C:\ClusterStorage> get-physicaldisk

FriendlyName         SerialNumber   CanPool OperationalStatus HealthStatus Usage           Size
------------         ------------   ------- ----------------- ------------ -----           ----
HP LOGICAL VOLUME                   False   OK                Healthy      Auto-Select 111.76 GB
HP EH0600JDYTL       0XHVG9AP       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHV8P7P       False   OK                Healthy      Auto-Select 558.75 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805010 False   OK                Healthy      Journal        447 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805055 False   OK                Healthy      Journal        447 GB
HP EH0600JDYTL       0XHVB72P       False   OK                Healthy      Auto-Select 558.75 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805014 False   OK                Healthy      Journal        447 GB
ATA SAMSUNG MZ7KM480 S2HSNX0H805011 False   OK                Healthy      Journal        447 GB
HP EH0600JDYTL       0XHVJT1P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVJVZP       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVGA3P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHV9M9P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVJW5P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHTXM2P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVRX7P       False   OK                Healthy      Auto-Select 558.75 GB
HP EH0600JDYTL       0XHVRURP       True    OK                Healthy      Auto-Select 558.91 GB

PS C:\ClusterStorage> $Disk = get-physicaldisk |? CanPool -like True
PS C:\ClusterStorage> Get-StoragePool *S2D* | Add-PhysicalDisk -PhysicalDisks $Disk
PS C:\ClusterStorage> _
```

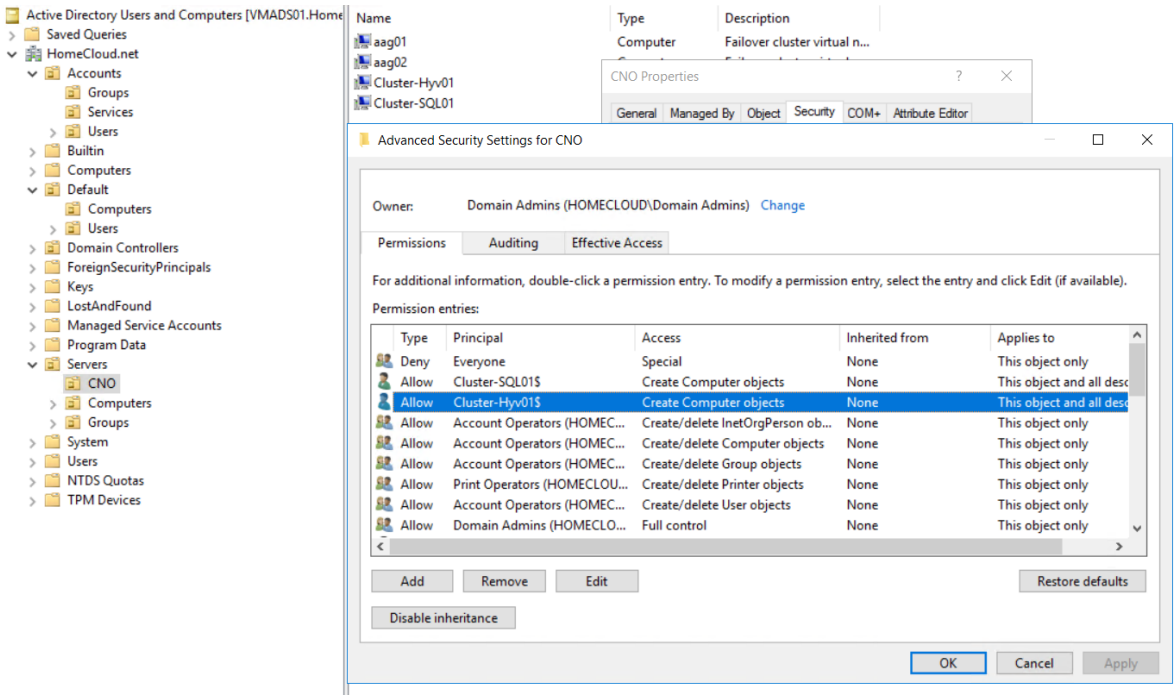
Patch management

A Storage Spaces Direct cluster is based on a Windows Server 2016. Even if you use Windows Server with user Experience, Windows Server Core or Nano Server, you have to patch your operating system. The patching is important for security, stability and to improve features. Storage Spaces Direct (S2D) is a new feature and it is important to patch the operating system to resolve some issues. But in disaggregated or hyper-converged model, the S2D hosts sensitive data such as virtual machines. So, to avoid service interruption, the patching of all nodes must be orchestrated. Microsoft offers a solution to update nodes of a failover cluster with orchestration: it is called Cluster Aware Updating (known as CAU). This topic describes how to use Cluster Aware Updating to handle the patch management of Storage Spaces Direct cluster nodes.

Prepare Active Directory

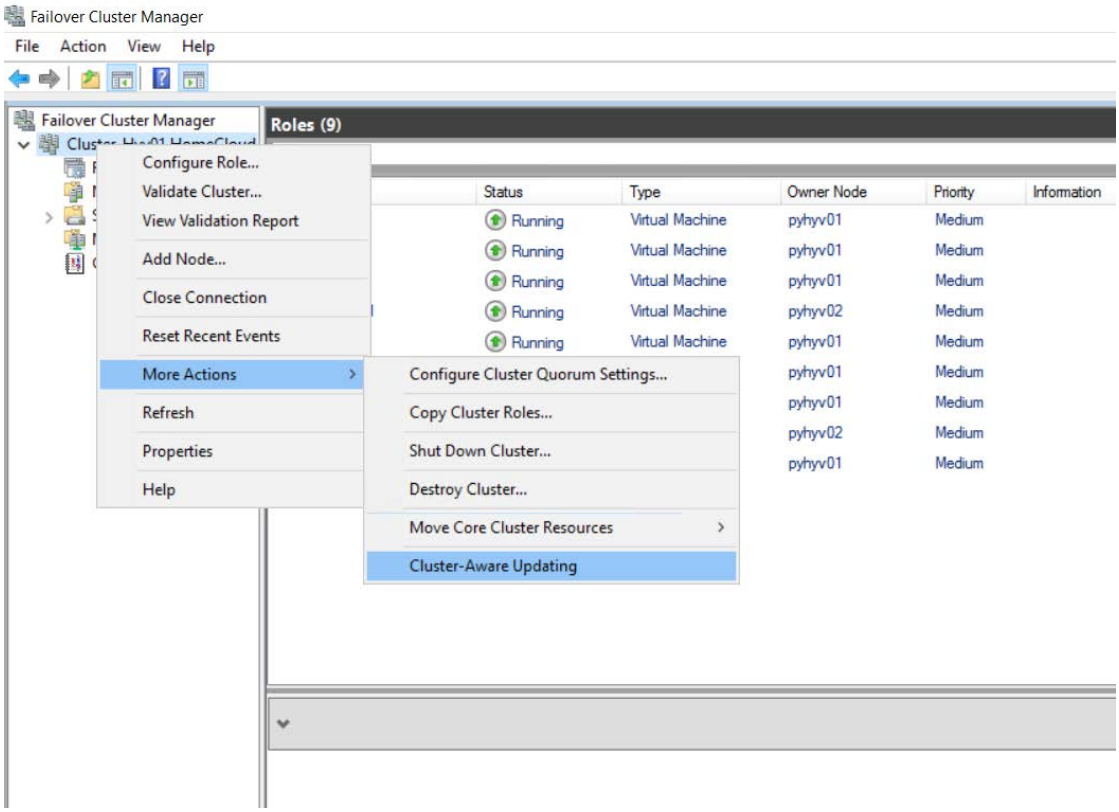
Because we will configure the self-updating in the Cluster Aware Updating (CAU), a computer object will be added in Active Directory in the same organizational unit of the cluster name object (CNO). This is the CNO account that will add the computer object for CAU. So, the CNO must have the permissions on the OU to create computer object. In my example, the cluster is called **Cluster-Hyv01**. So, on the OU where is located this CNO, we have granted **Cluster-Hyv01** account to **create computer objects**.

N.B: you can prestaged the computer object for the CAU and skip this step.

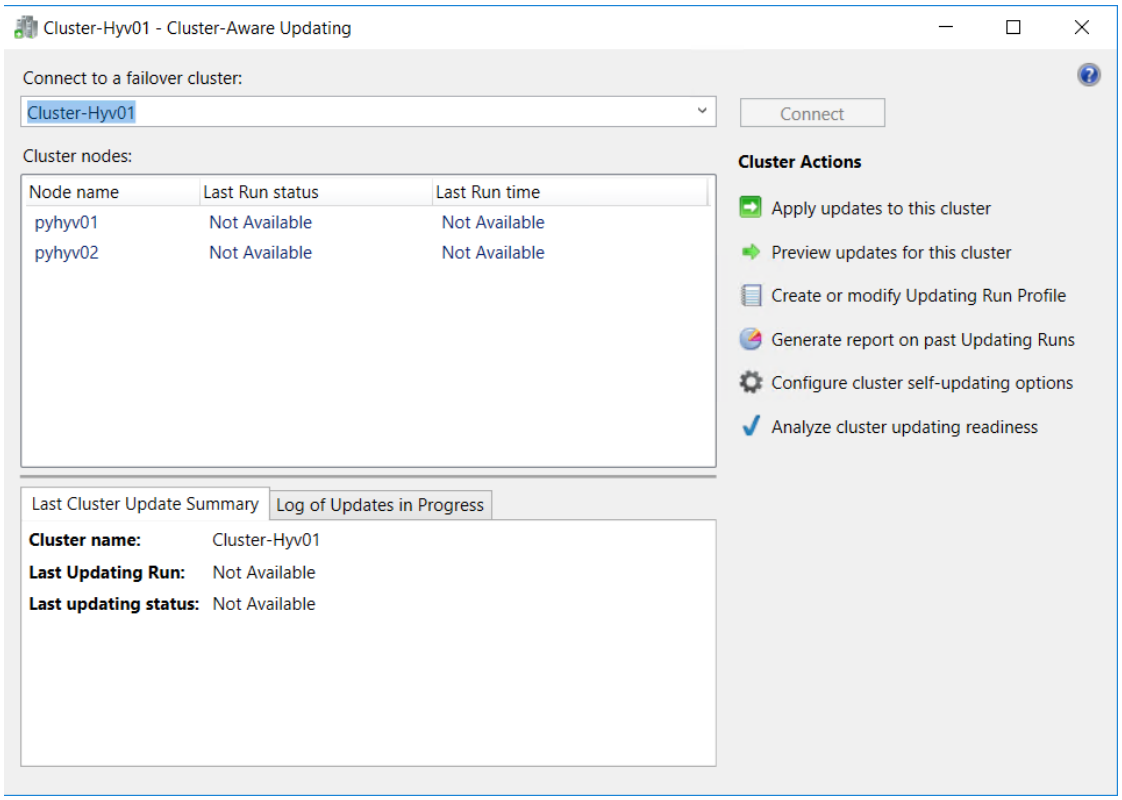


Configure Self-Updating

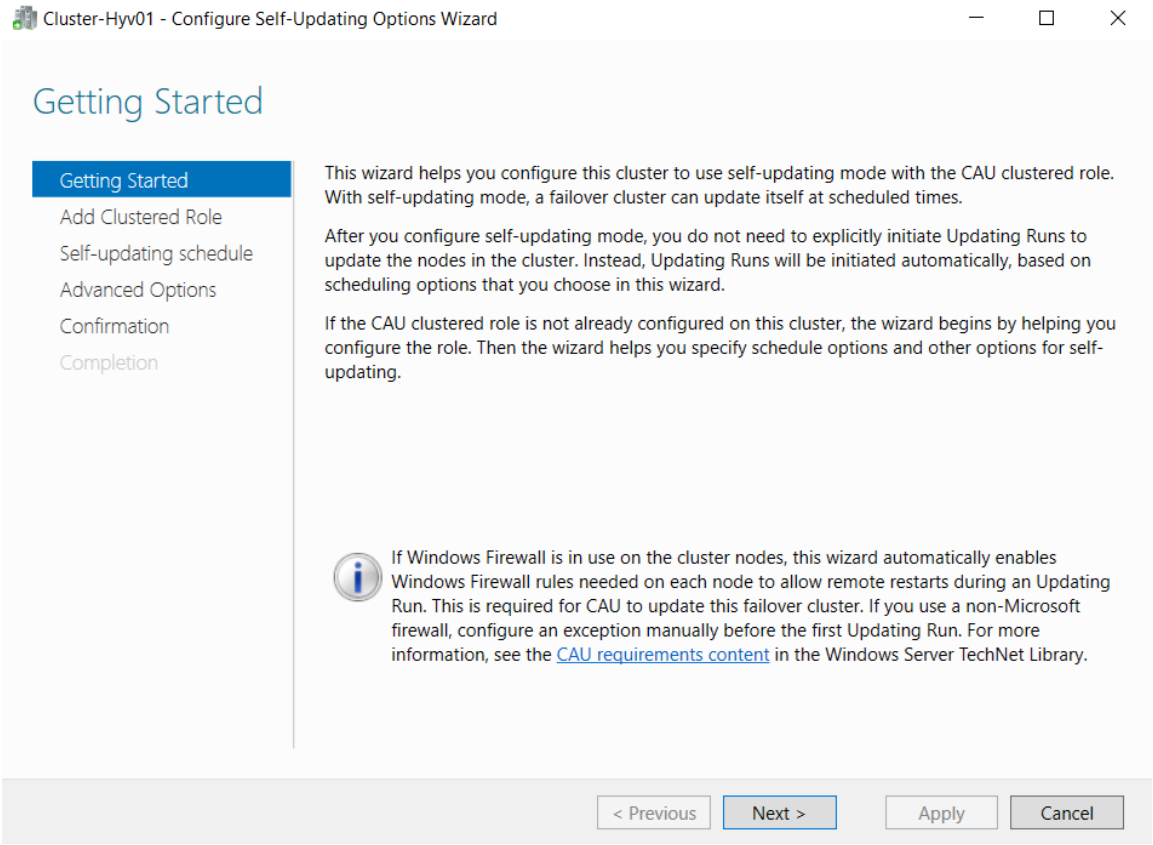
To configure CAU, you can open the Failover Cluster Manager and right click on the cluster. Then choose **More Actions** | **Cluster-Aware Updating**.



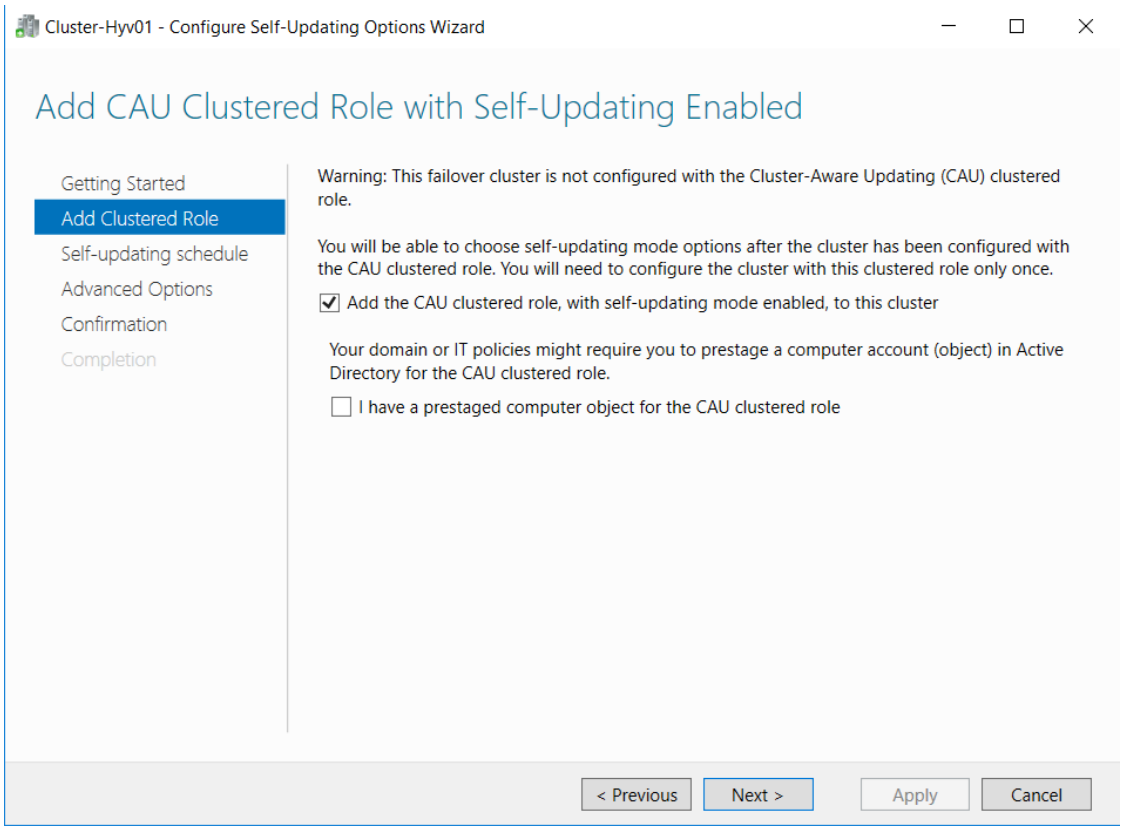
In the CAU GUI, click on **Configure cluster self-updating options**.



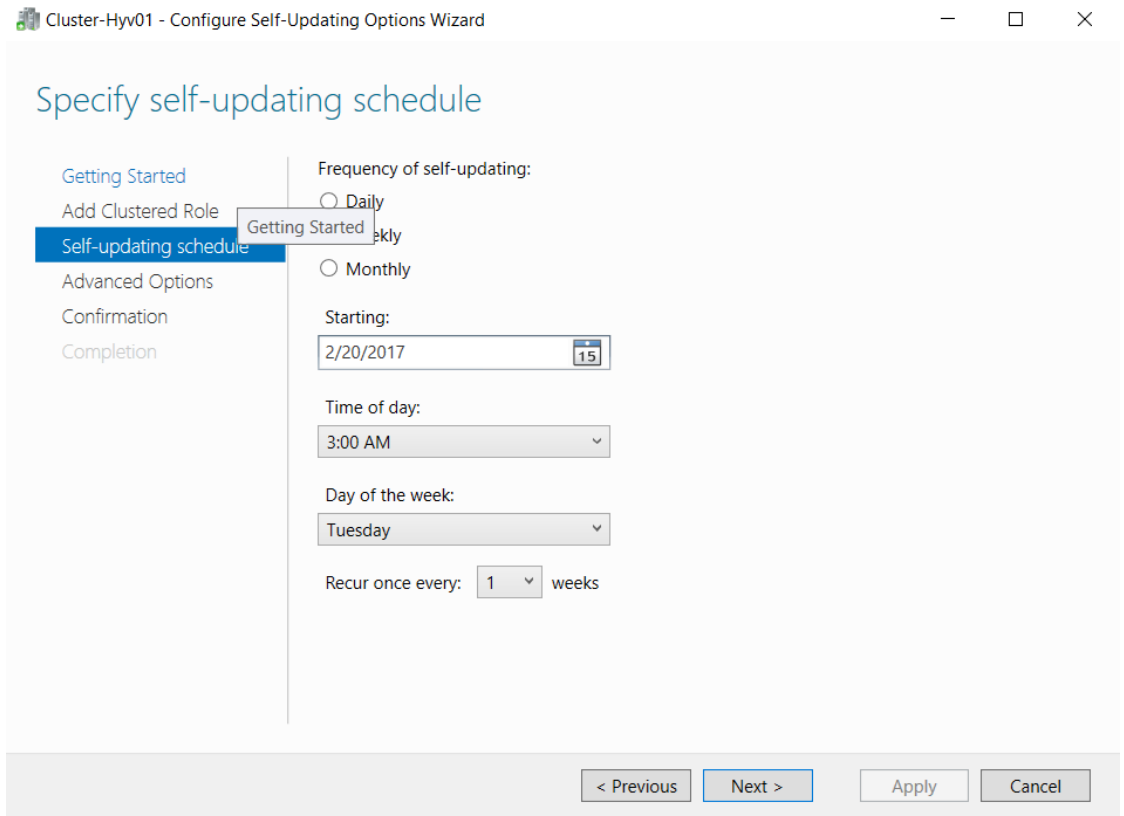
In the first window of the wizard, just click on **Next**.



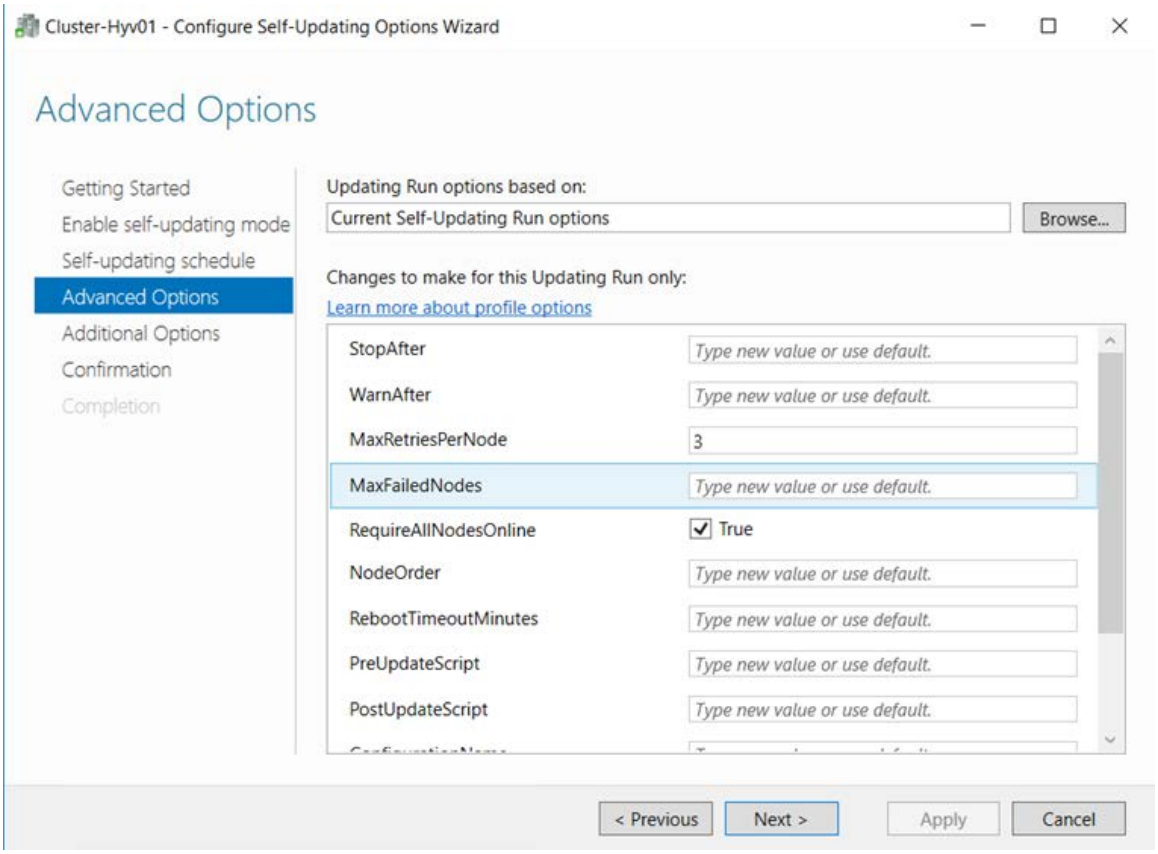
Next, select the option **Add the CAU clustered role, with self-updating mode enabled, to this cluster**.



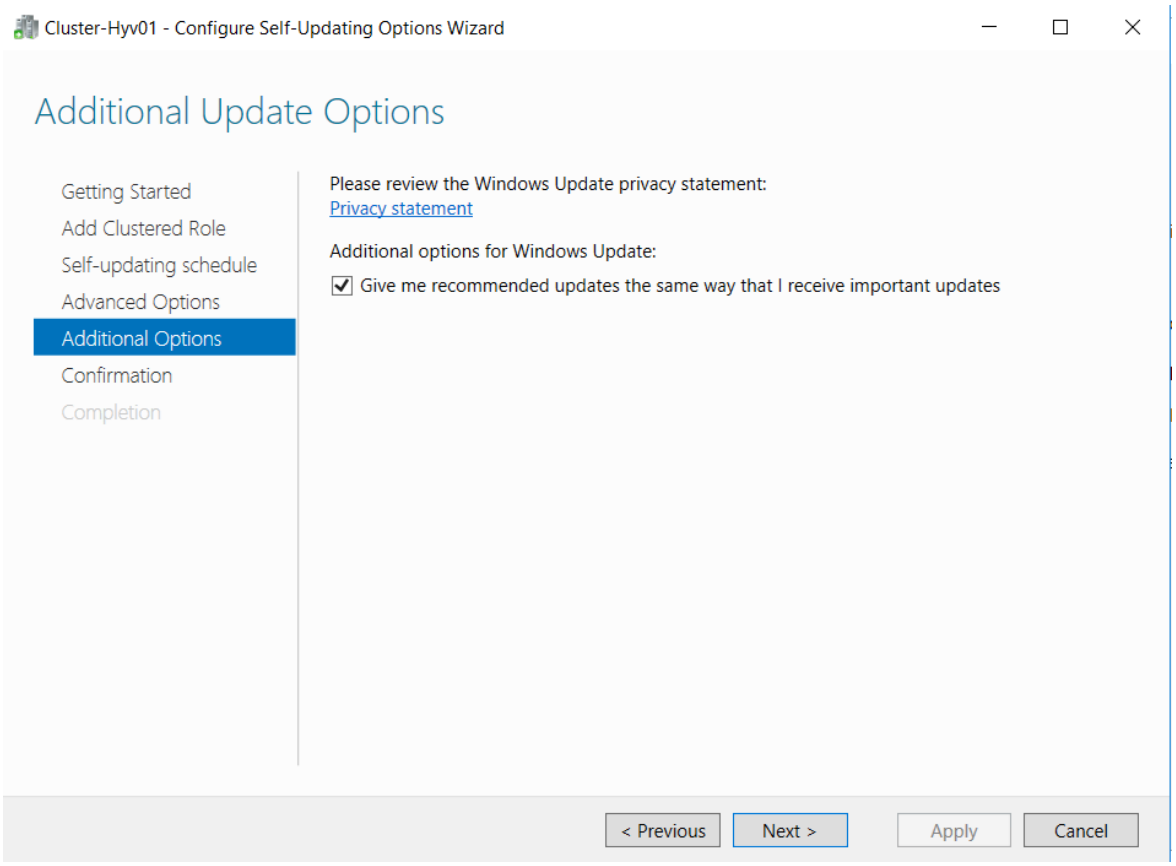
In the next screen, you can specify the frequency of the self-updating. The CAU will check updates regarding this schedule.



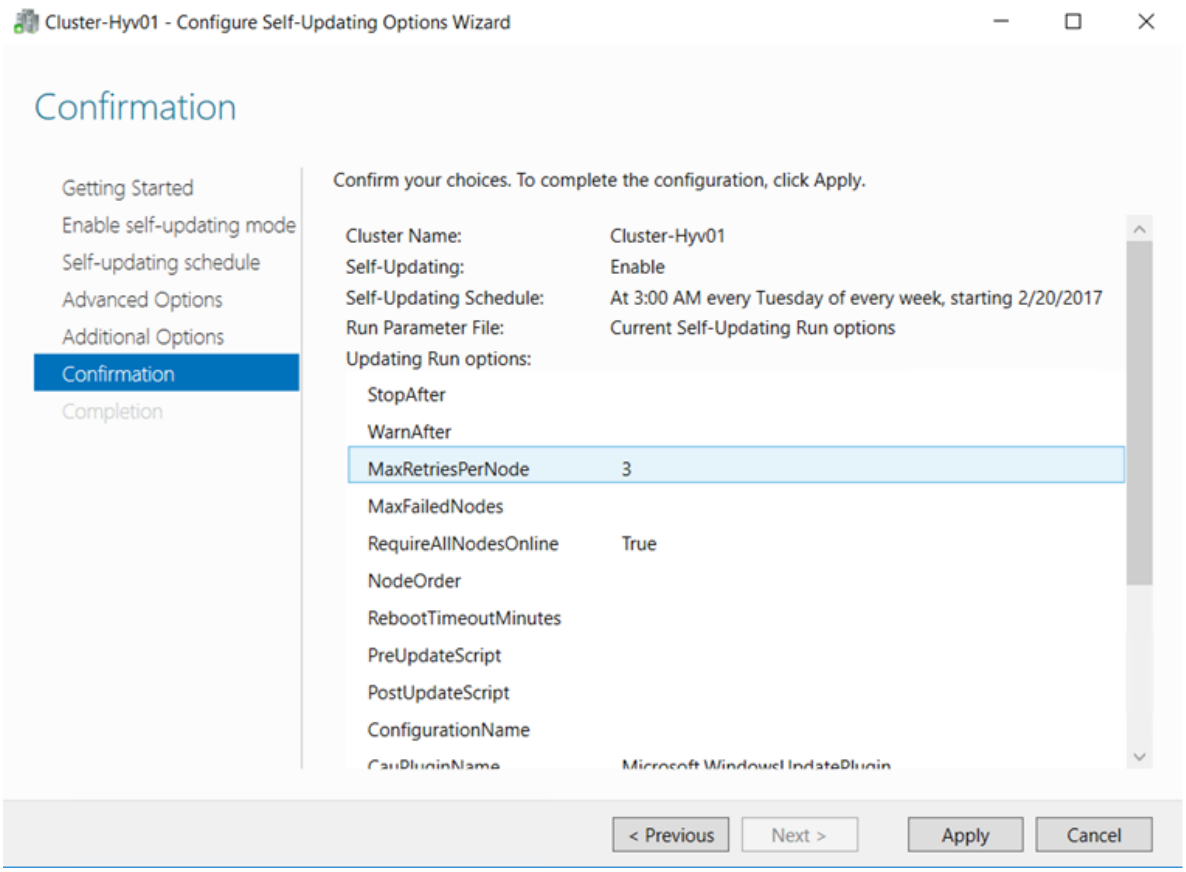
Next you can change options on the updating run. You can specify **Pre** and **Post** scripts or you can set that all nodes must be online to run the updating process.



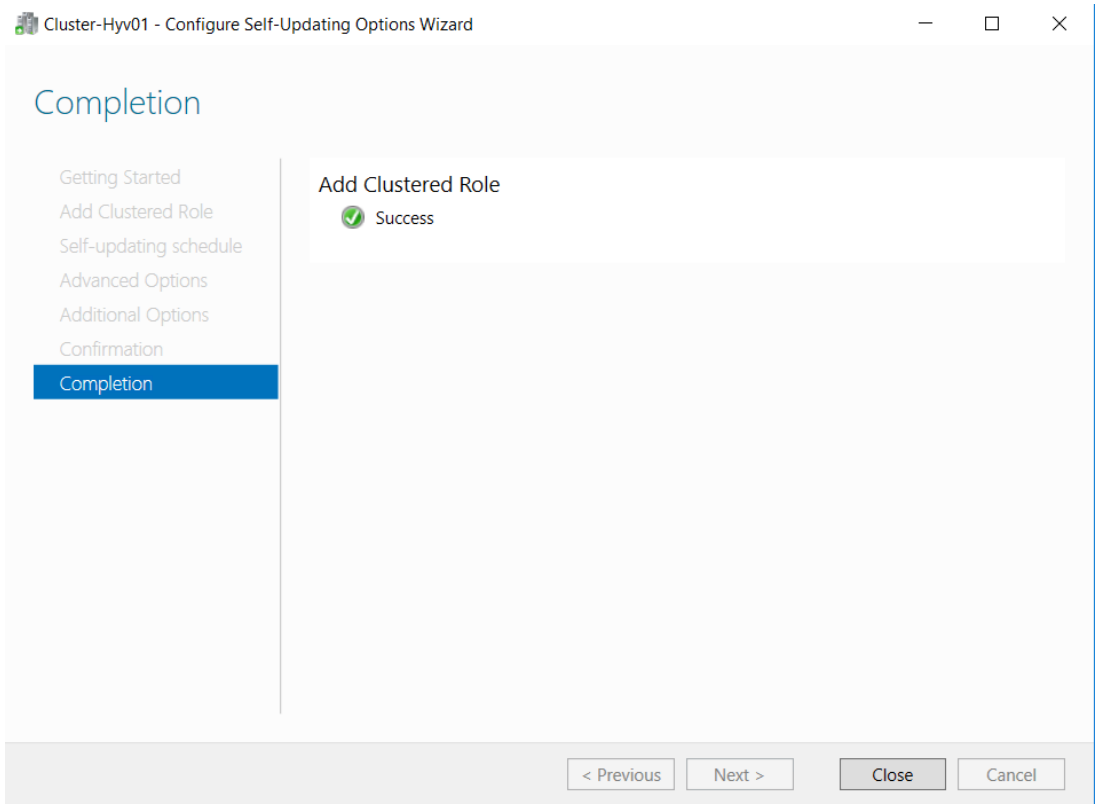
In the next screen, you can choose to get recommended updates like important updates.



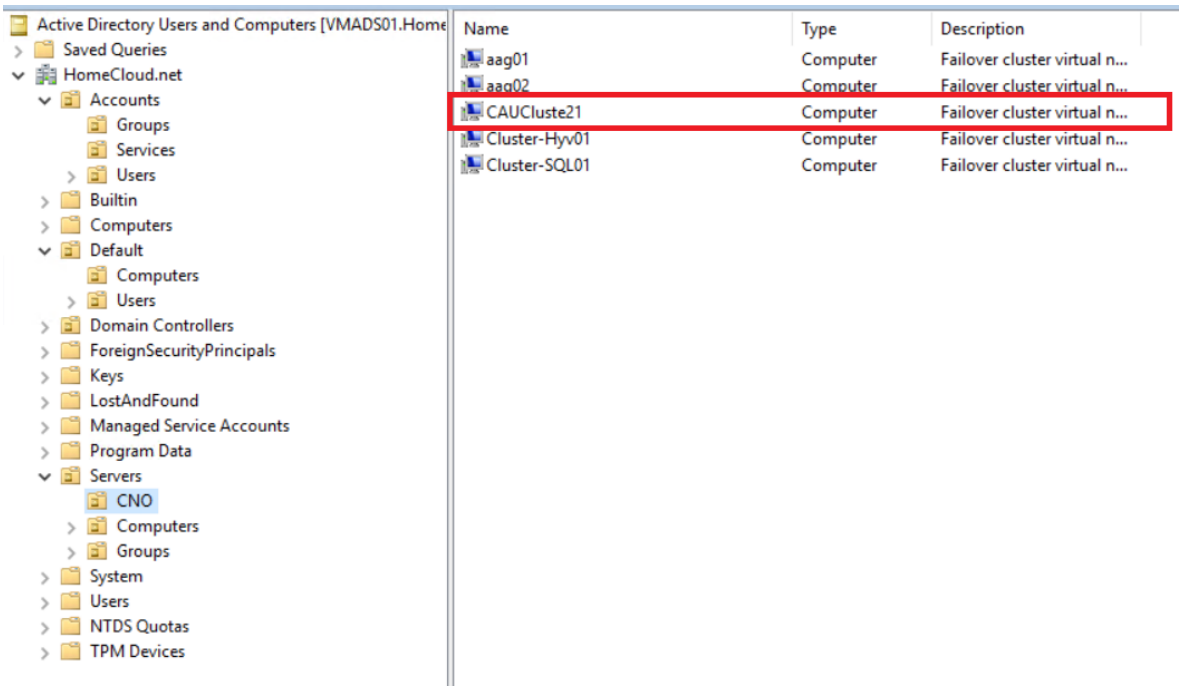
Then review the configuration that you have specified and click on apply.



If the CAU clustered role is added successfully, you should have something as below:

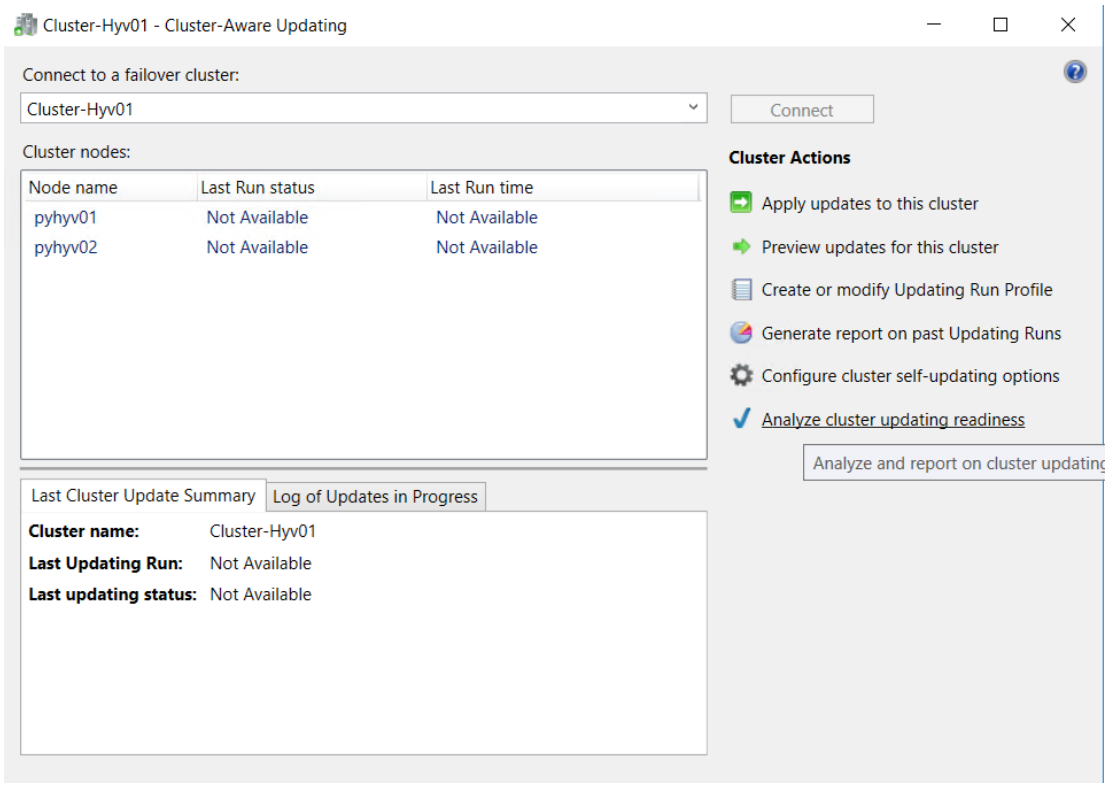


In Active Directory, you should have a new computer object beginning with CAU:

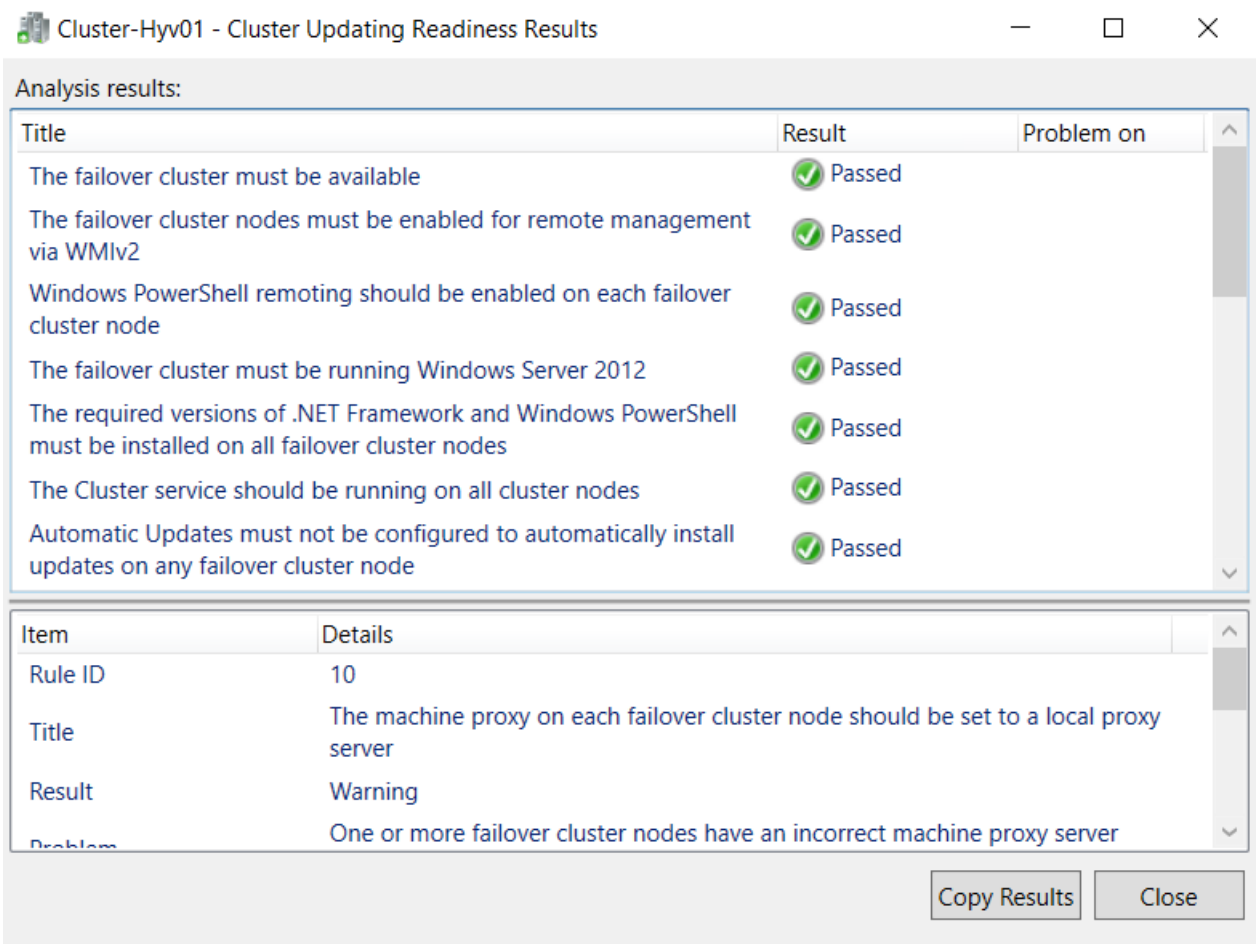


Validate the CAU configuration

You can review the good configuration of CAU by clicking on **Analyze cluster updating readiness**.

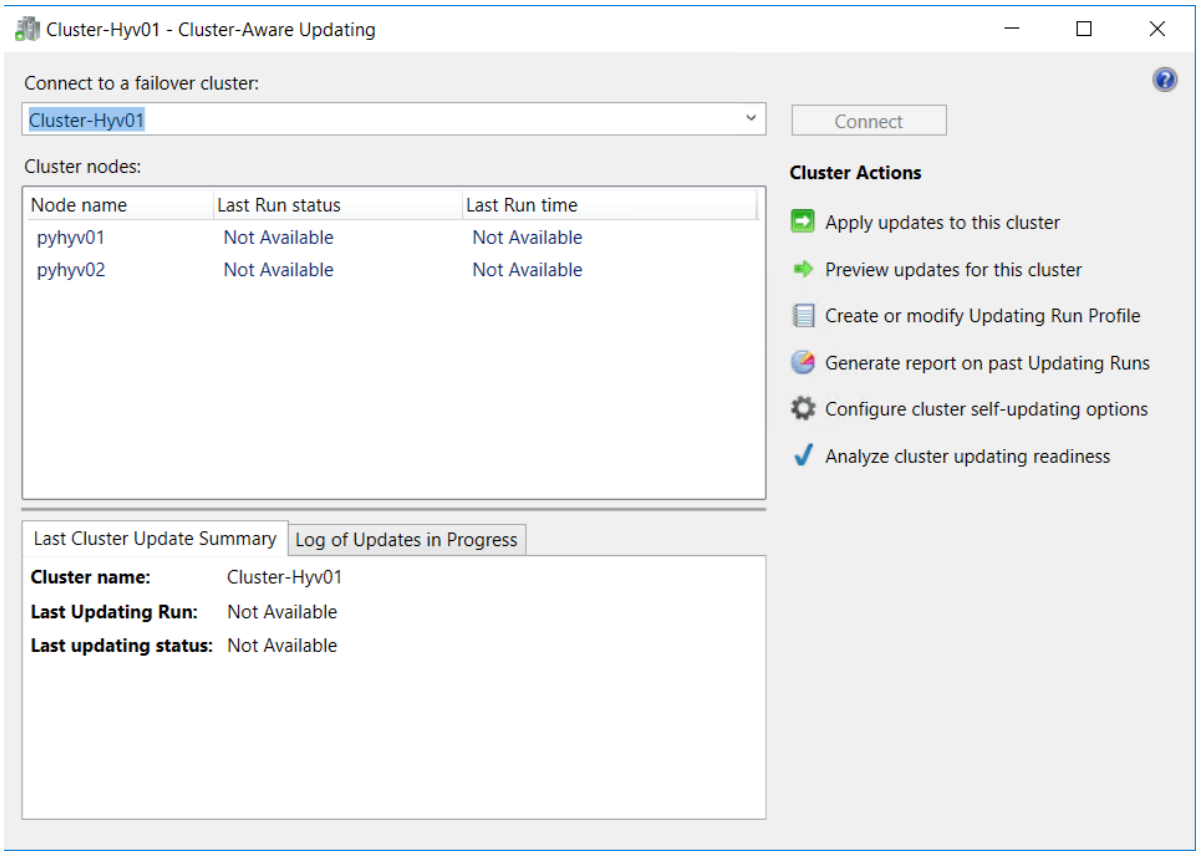


You should get the result **Passed** for each test.

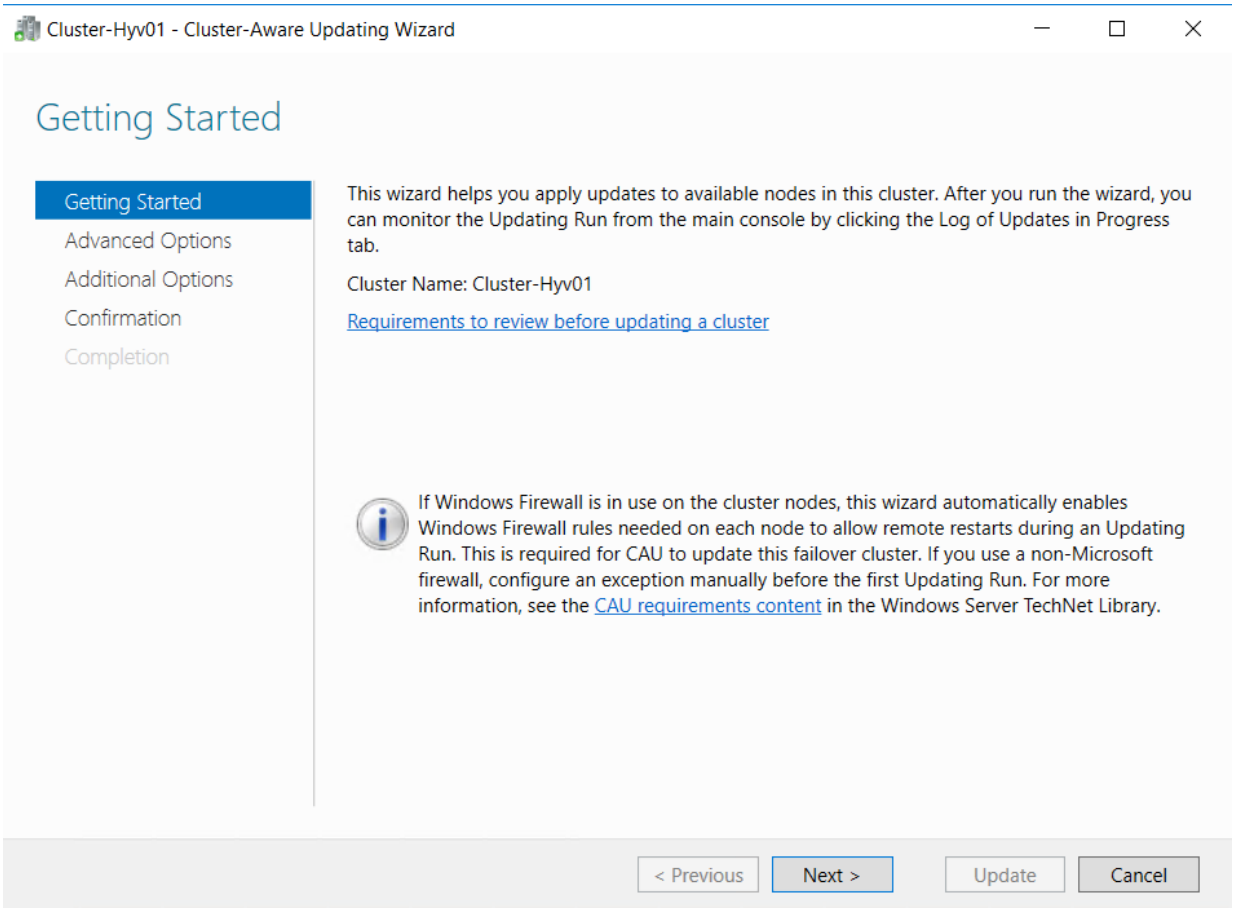


Run manual updates to the cluster

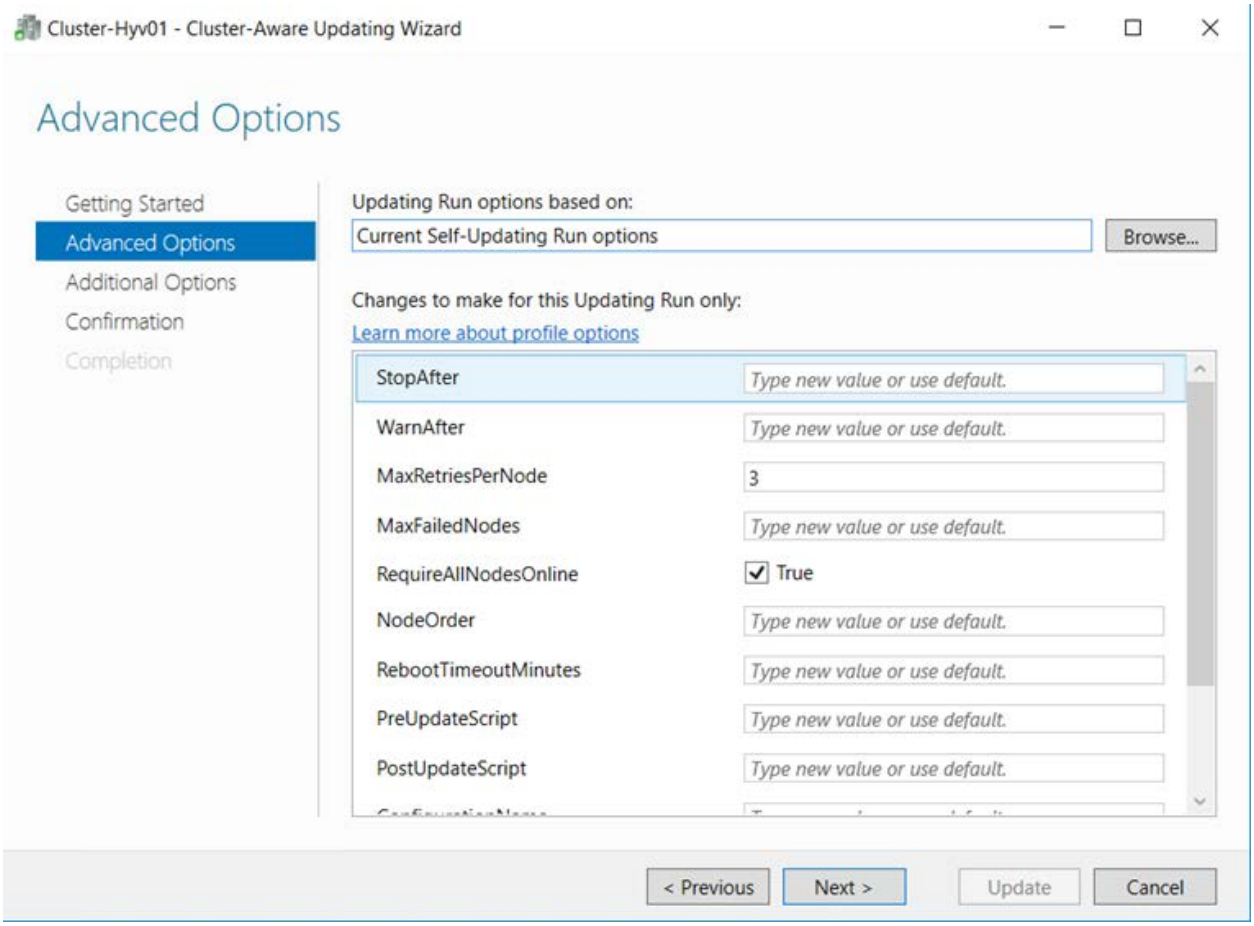
The Self-Updating enables to schedule the cluster updates. But you can also apply updates manually. In the CAU interface, click on **Apply updates to this cluster**.



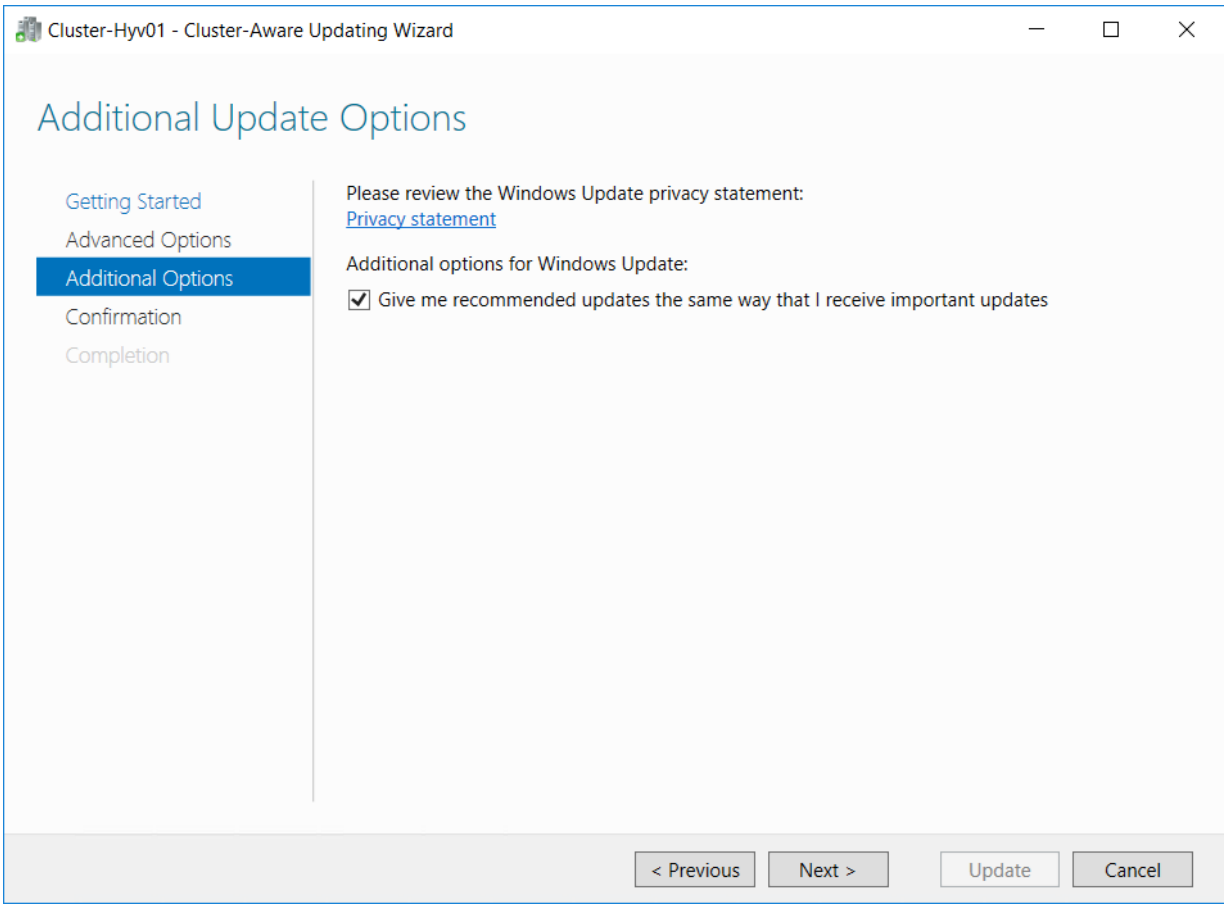
In the next screen, just click on **next**.



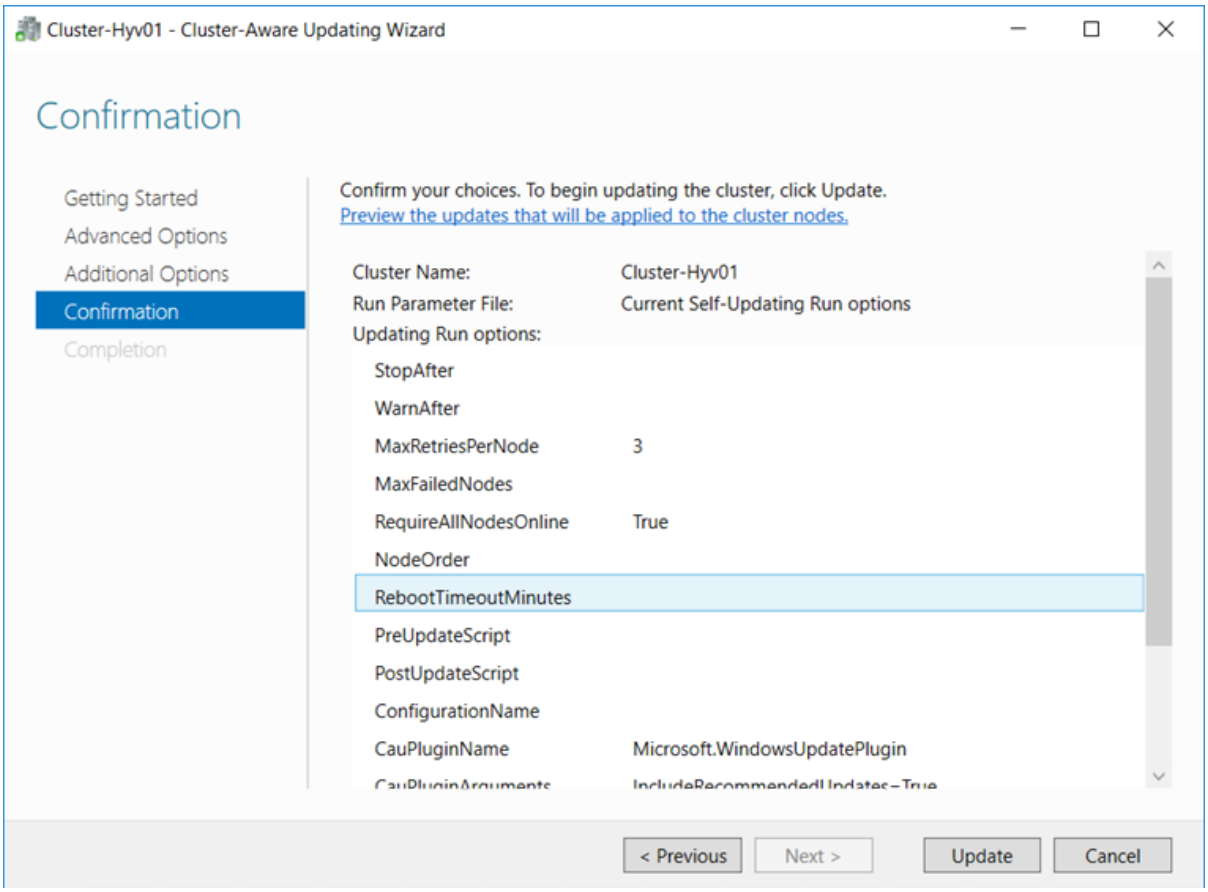
Next, specify option for the updating process. As previously, you can specify pre and post scripts and other settings such as node order to update or wait that all nodes are online to run the updates.



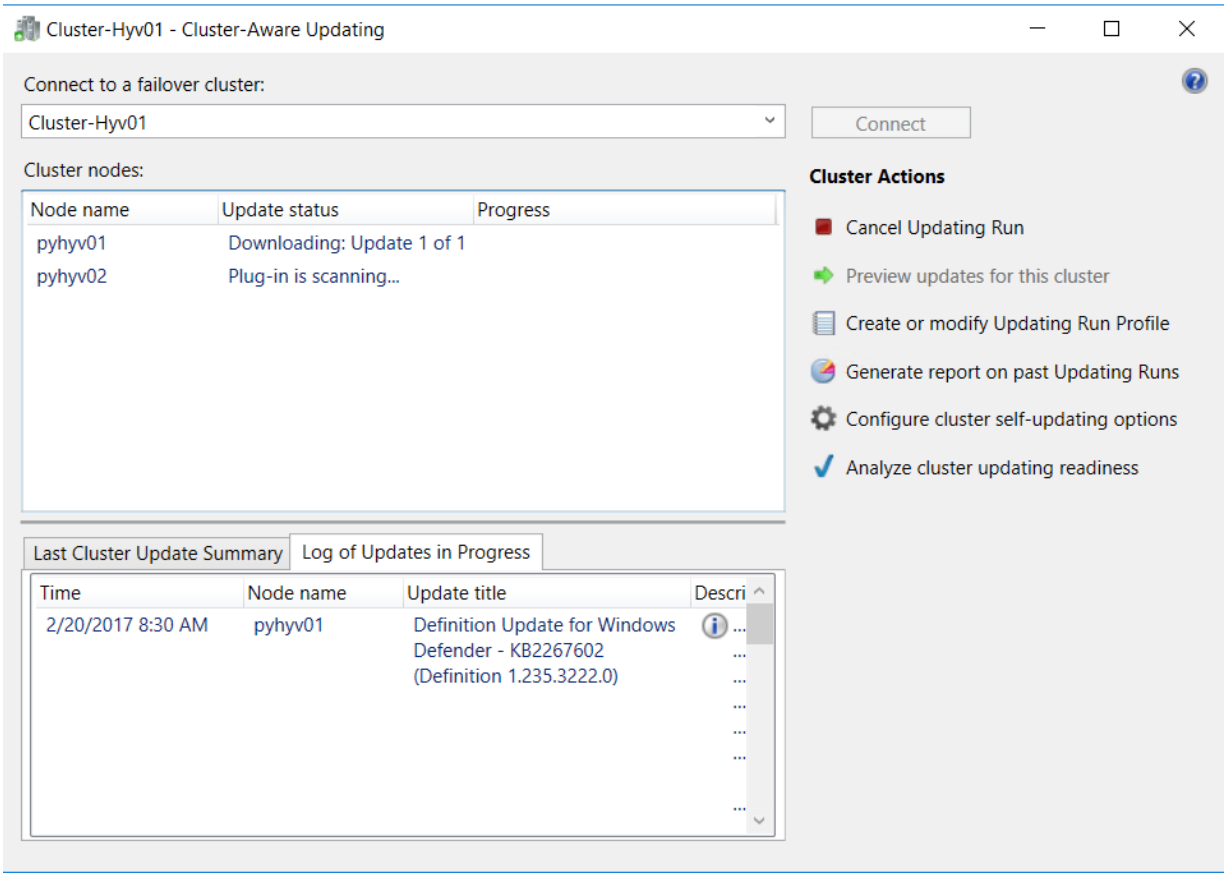
Next choose if you want to apply recommended updates.



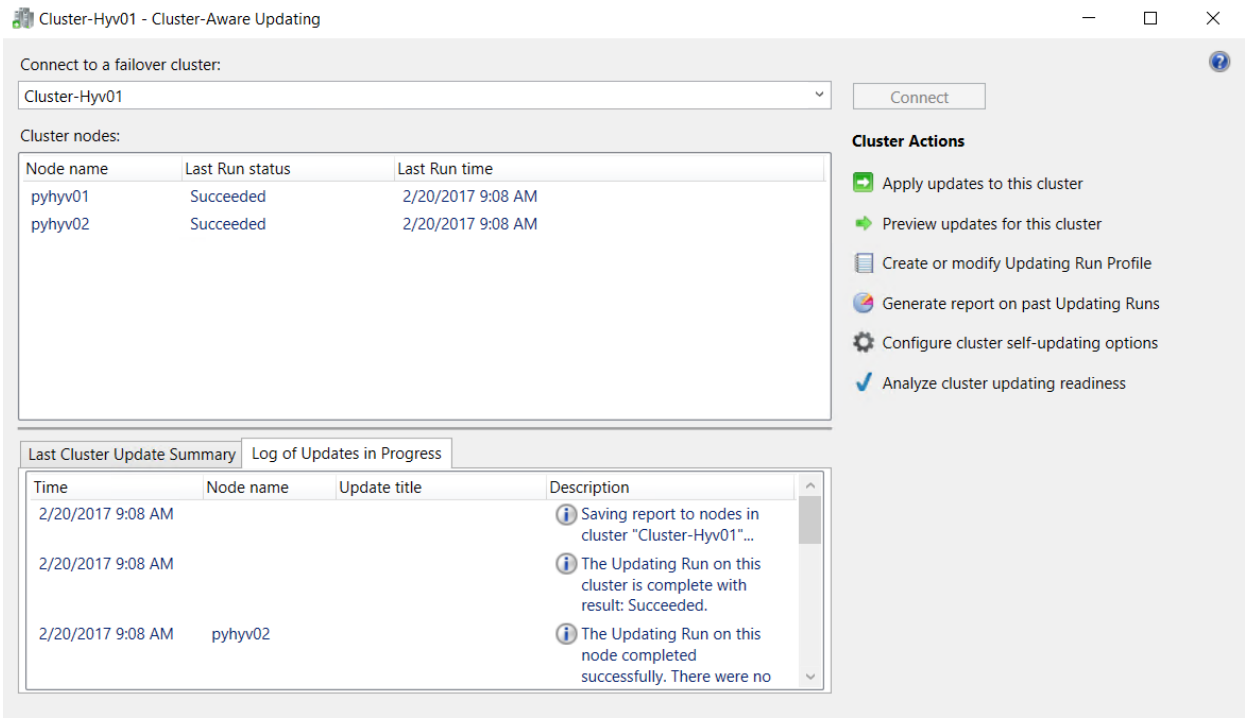
To finish, review the settings. If the configuration is good, click on **update**.



While updating, you have information in Log of **Updates in Progress**. You can know which node is now updating, which node is in maintenance mode and which updates are applied.



When the updates are finished, you should have **Succeeded** status for each node.



Conclusion

Hyper Convergence is a fantastic way to build modern datacenter. First, because storage and compute are inside the same physical servers, there is no scalability problem except if you want to manage the scalability of the storage and compute separately. Nevertheless, it is up to you. Secondly, all the solution is based on software either compute, network or storage. Therefore, it is easier to manage and to support operational. Software solutions are also cheaper than hardware solution such as traditional SAN.

In my opinion, Nano Server is a good technology for Hyper Convergence. Nano Deployment is faster than standard Windows Server 2016 (core or not) and there are no legacy and useless services. It means that there are less crash and less updates. In addition, because the restart is fast, even if the system fails, you can restart quickly.

However, as you have seen in this whitepaper, there are many technologies around the Hyper Convergence, each of them is necessary to obtain reliable performance on the solution. A demanding work on the design is needed before the implementation. Companies which bypass the design phase, will have some problems, mainly regarding performance and resiliency.

Now that nested Hyper-V is possible, try your solution, and try it again. When you are sure of your solution, deploy it in production.

References

charbelnemnom.com
External blog - Charbel Nemnom

DataCenter Bridging (DCB) overview
TechNet Topic

Getting Started with Nano Server
TechNet Topic

Hyper-V Architecture
MSDN

Remote Direct Access Memory (RDMA) and Switch Embedded Teaming (SET)
TechNet Topic

Storage Replica overview
TechNet Topic

Storage Spaces Direct Calculator
External blog - Cosmos Darwin

Storage Spaces Direct choose drives
TechNet Topic

Storage Spaces Direct Hardware Requirements
TechNet topic

Storage Spaces Direct in Windows Server 2016
TechNet Topic

Storage Spaces Direct: Understand
TechNet topic

Storage Spaces Stripping and Mirroring
External Blog - Llubo Brodaric

Storage Quality of Service
Technet Topic

VMQueue Deep Dive
TechNet blog - Ravi Rao

Tech-Coffee
External blog - Romain Serre