



Best Practices For Running VMware vSphere On iSCSI

VMware Storage

Table of contents

Best Practices For Running VMware vSphere On iSCSI	4
Introduction	4
iSCSI Overview	5
iSCSI Considerations	5
iSCSI Architecture	6
iSCSI Names	6
iSCSI Initiators and Targets	6
Software iSCSI Adapter	6
Dependent Hardware iSCSI Adapter	7
Independent Hardware iSCSI Adapter	8
iSCSI Sessions and Connections	9
iSCSI Implementation Options	12
Mixing iSCSI Options	12
Networking Settings	13
VMkernel Network Configuration	13
IPv6 Supportability Statements	13
Throughput Options	13
Minimizing Latency	13
Routing	14
Using Static Routes or set a Gateway	14
Availability Options - Multipathing or NIC Teaming	14
NIC Teaming for Availability	14
iSCSI Multipathing via Port Binding for Availability	14
Error Correction Digests	15
Flow Control	15
iSCSI Port Binding Best Practices	15
Additional iSCSI network configuration possibilities.	21
Security Considerations	22
Private Network	22
Encryption	22
Authentication	22
iSCSI Datastore Provisioning Steps	23
Why Use iSCSI Multipathing?	24
Software iSCSI Multipathing Configuration Steps	24

Sizing Considerations - Recommended Volume Size 27

 Recommended Block Size 27

Booting a vSphere Host from Software iSCSI 28

 Why Boot from SAN? 28

 Compatible Network Interface Card (NIC) 28

 Configuring Host BIOS for iSCSI Boot 28

 Installing a vSphere Host on an iSCSI LUN 28

 Booting from an iSCSI LUN 29

 Troubleshooting Checklist 29

Additional Considerations 30

 Disk Alignment 30

 Microsoft Clustering Support 30

 In-Guest iSCSI Support 30

 All Paths Down and Permanent Device Loss 30

 Read-Only File Systems on Linux Guest OS 30

 Round Robin Path Policy Setting IOPS=1 30

 Tape Device Support 30

iSCSI Docs and KB Resources 31

Conclusion and About the Author 32

 Conclusion 32

Best Practices For Running VMware vSphere On iSCSI

Introduction

VMware offers and supports a number of different storage technologies and protocols for presenting external storage devices to VMware vSphere hosts. In recent years, the iSCSI protocol has gained popularity as a method for presenting block storage devices over a network to vSphere hosts. VMware has provided support for iSCSI storage since Virtual Infrastructure 3. This paper can help you understand the design considerations and deployment options for deploying vSphere infrastructures using iSCSI storage. It highlights trade-offs and factors to consider when deploying iSCSI storage to support vSphere environments. It is a complement to, not a replacement for, VMware product documentation.

iSCSI Overview

iSCSI is a protocol that uses the TCP/IP to transport SCSI commands, enabling the use of the existing TCP/IP networking infrastructure as a SAN. As with SCSI over Fibre Channel (FC), iSCSI presents SCSI targets and devices to iSCSI initiators (requesters). Unlike NAS, which presents devices at the file level, iSCSI makes block devices available via the network. Block devices are presented across an IP network to your local system. These can be consumed in the same way as any other block storage device.

iSCSI Considerations

For datacenters with centralized storage, iSCSI offers customers many benefits. It is comparatively inexpensive and it is based on familiar SCSI and TCP/IP standards. In comparison to FC and Fibre Channel over Ethernet (FCoE) SAN deployments, iSCSI requires less hardware, it uses lower-cost hardware, and more IT staff members might be familiar with the technology. These factors contribute to lower-cost implementations.

One major difference between iSCSI and FC relates to I/O congestion. When an iSCSI path is overloaded, the TCP/IP protocol drops packets and requires them to be resent. FC communication over a dedicated path has a built-in pause mechanism when congestion occurs. When a network path carrying iSCSI storage traffic is substantially oversubscribed, a bad situation quickly grows worse and performance further degrades as dropped packets must be resent. There can be multiple reasons for an iSCSI path being overloaded, ranging from oversubscription (too much traffic), to network switches that have a low port buffer. Although some iSCSI storage vendors have implemented Delayed Ack and Congestion Avoidance as part of their TCP/IP stack, not all have. Various iSCSI array vendors even recommend disabling DelayedAck for iSCSI adapter. VMware recommends consulting the iSCSI array vendor for specific recommendations around DelayedAck. For more details on this issue please refer to: <https://kb.vmware.com/s/article/1002598>

Another consideration is the network bandwidth. Network bandwidth is dependent on the Ethernet standards used (1Gb or 10Gb). There are other mechanisms such as port aggregation and bonding links that deliver greater network bandwidth. When implementing software iSCSI that uses network interface cards rather than dedicated iSCSI adapters, gigabit Ethernet interfaces are required. These interfaces tend to consume a significant amount of CPU Resource.

One way of overcoming this demand for CPU resources is to use a feature called a TOE (TCP/IP offload engine). TOEs shift TCP packet processing tasks from the server CPU to specialized TCP processors on the network adaptor or storage device. Most enterprise-level networking chipsets today offer TCP offload or checksum offload, which vastly improves CPU overhead.

iSCSI was considered a technology that did not work well over most shared wide-area networks. It has prevalently been approached as a local area network technology. However, this is changing. For synchronous replication writes (in the case of high availability) or remote data writes, iSCSI might not be a good fit. Latency introductions bring greater delays to data transfers and might impact application performance. Asynchronous replication, which is not dependent upon latency sensitivity, makes iSCSI an ideal solution. For example, VMware vCenter™ Site Recovery Manager™ may build upon iSCSI asynchronous storage replication for simple, reliable site disaster protection.

iSCSI Architecture

iSCSI initiators must manage multiple, parallel communication links to multiple targets. Similarly, iSCSI targets must manage multiple, parallel communications links to multiple initiators. Several identifiers exist in iSCSI to make this happen, including iSCSI Name, ISID (iSCSI session identifiers), TSID (target session identifier), CID (iSCSI connection identifier), and iSCSI portals. These will be examined in the next section.

iSCSI Names

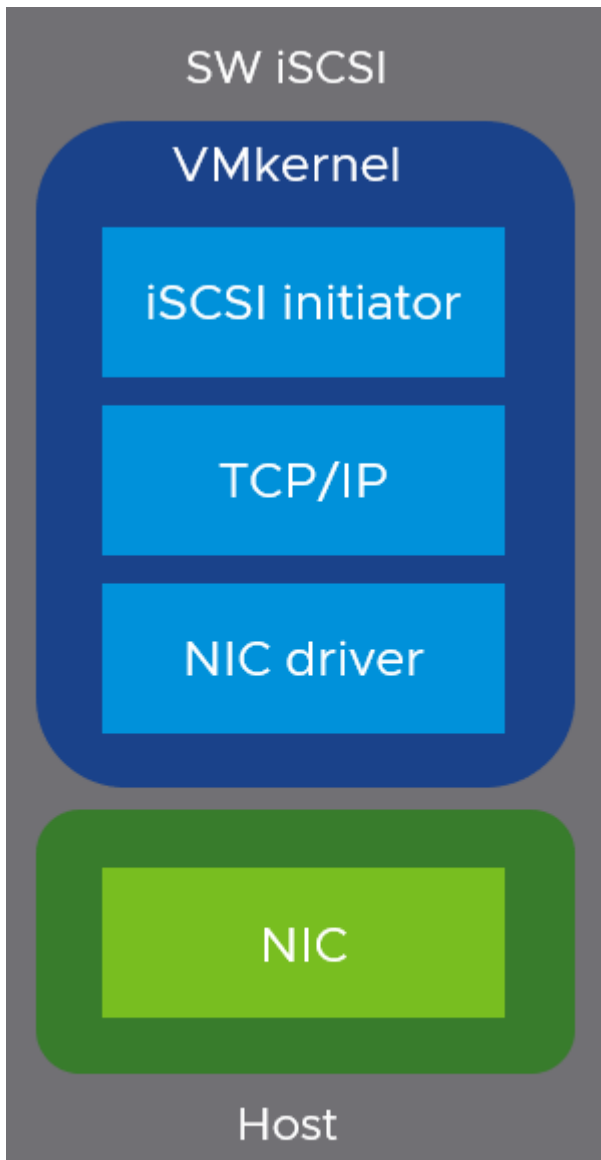
iSCSI nodes have globally unique names that do not change when Ethernet adapters or IP addresses change. iSCSI supports two name formats as well as aliases. The first name format is the Extended Unique Identifier (EUI). An example of a EUI name might be eui.02004567A425678D. The second name format is the iSCSI Qualified Name (IQN). An example of an IQN name might be iqn.1998-01. com.vmware:tm-pod04-esx01-6129571c.

iSCSI Initiators and Targets

A storage network consists of two types of equipment: initiators and targets. Initiators, such as hosts, are data consumers. Targets, such as disk arrays or tape libraries, are data providers. In the context of vSphere, iSCSI initiators fall into three distinct categories. They can be software, hardware dependent, or hardware independent.

Software iSCSI Adapter

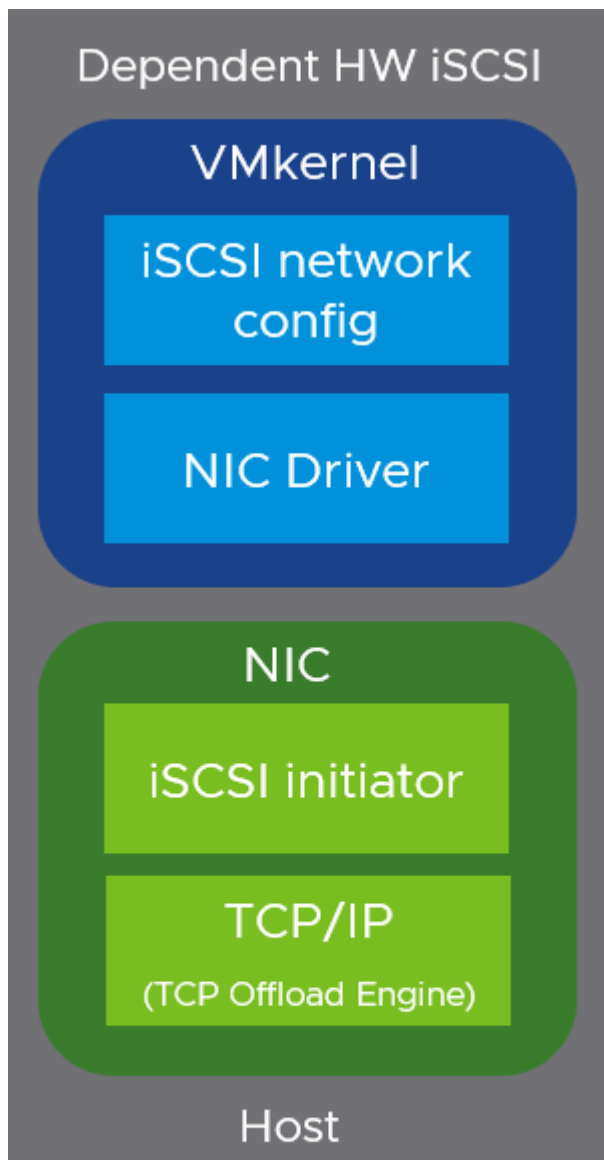
A software iSCSI adapter is VMware code built into the VMkernel. It enables your host to connect to the iSCSI storage device through standard network adapters. The software iSCSI adapter handles iSCSI processing while communicating with the network adapter. With the software iSCSI adapter, you can use iSCSI technology without purchasing specialized hardware.



Standard NIC adapter

Dependent Hardware iSCSI Adapter

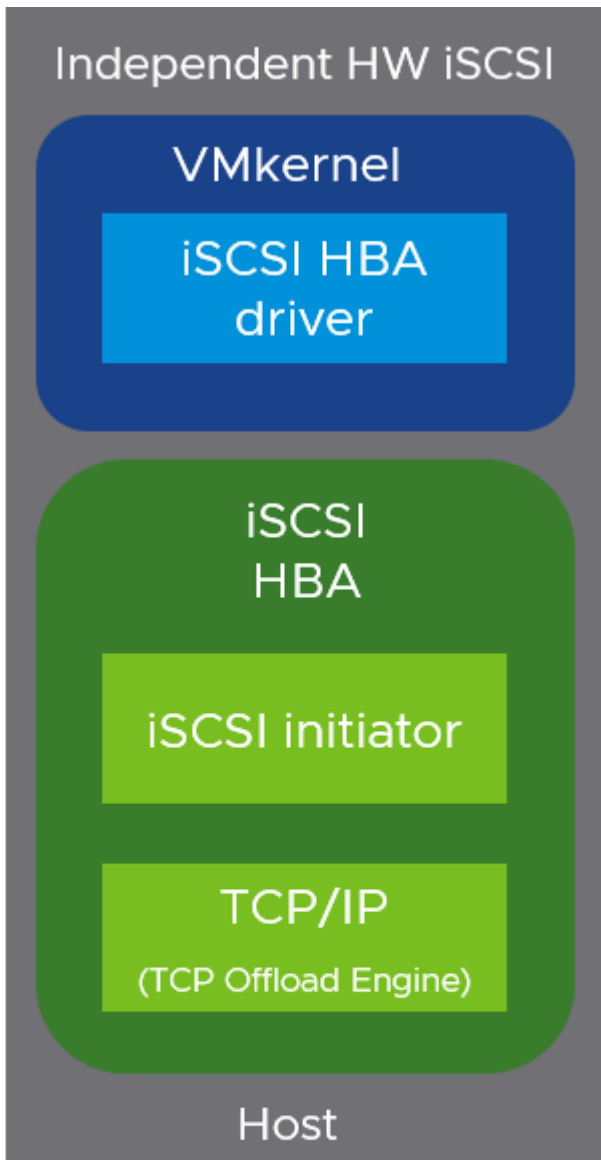
This hardware iSCSI adapter depends on VMware networking and iSCSI configuration and management interfaces provided by VMware. This type of adapter can be a card that presents a standard network adapter and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP and MAC addresses, as well as other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI licensed Broadcom 5709 NIC.



Third party adapter depends on VMware networking

Independent Hardware iSCSI Adapter

This type of adapter implements its own networking and iSCSI configuration and management interfaces. An example of an independent hardware iSCSI adapter is a card that presents either iSCSI offload functionality only or iSCSI offload functionality and standard NIC functionality. The iSCSI offload functionality has independent configuration management that assigns the IP address, MAC address, and other parameters used for the iSCSI sessions. This section examines the features and issues connected with each of these technologies.



Third party adapter offloads iSCSI, network processing, and management from host

iSCSI Sessions and Connections

iSCSI initiators and targets use TCP to create relationships called sessions. These sessions are identified by iSCSI session IDs (ISIDs). Session IDs are not tied to the hardware and can persist across hardware swaps. The initiator sees one logical connection to the target, as shown in Figure 1.

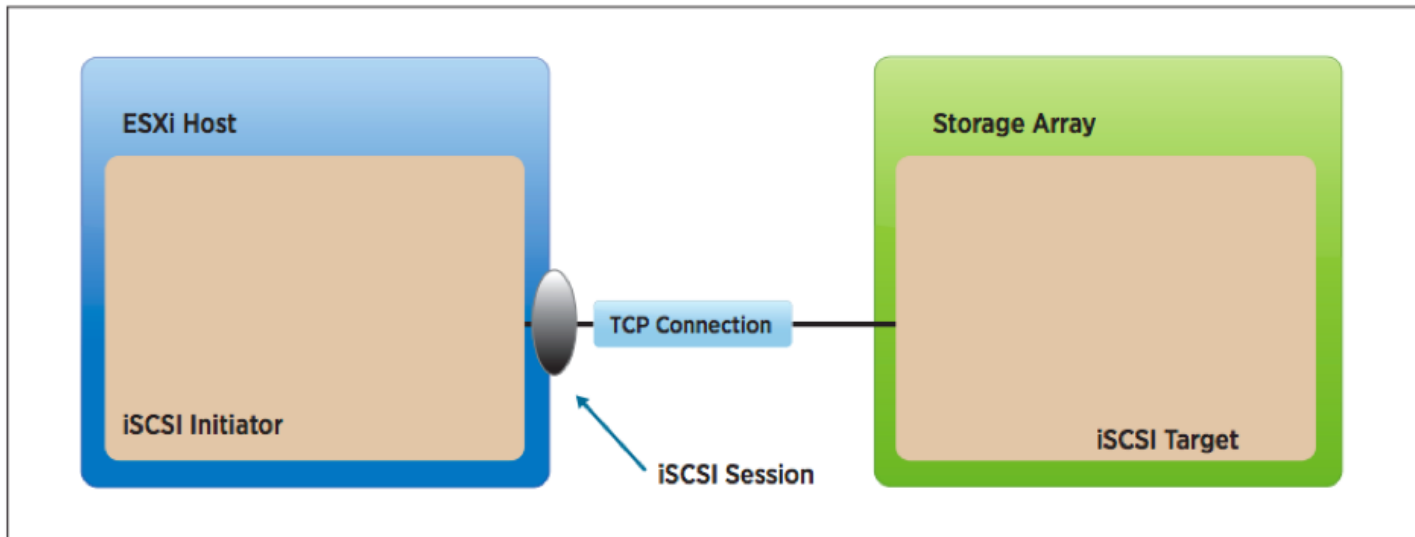


Figure 1 - iSCSI Session

An iSCSI session might also contain multiple logical connections. From a vSphere host perspective, the sessions might also be thought of in terms of paths between the initiator and target. Having multiple connections per session enables the aggregation of bandwidth and can also provide load balancing. An example of multiple logical connections to the target (identified by connection IDs, or CIDs) is shown in Figure 2.

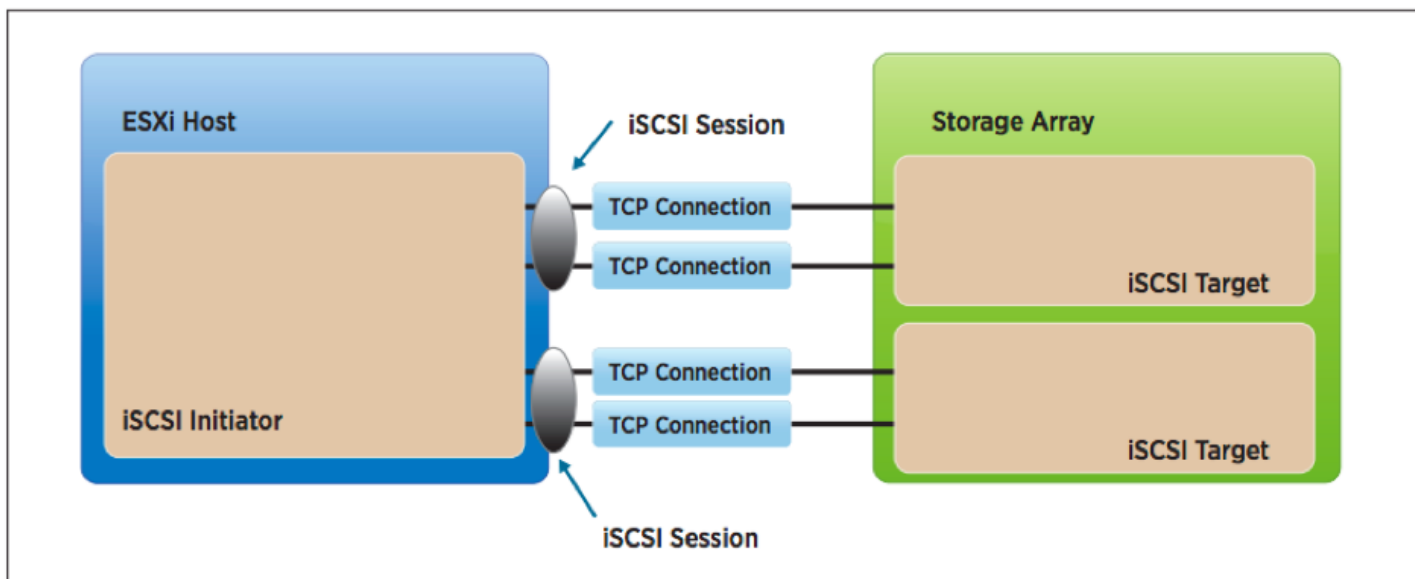


Figure 2 - Multiple connections per session

However, a vSphere host does not support multiple connections per session at this time.

iSCSI Portals - iSCSI nodes keep track of connections via portals, enabling separation between names and IP addresses. A portal manages an IP address and a TCP port number. Therefore, from an architectural perspective, sessions can be made up of multiple logical connections, and portals track connections via TCP/IP port/address, as shown in Figure 3.

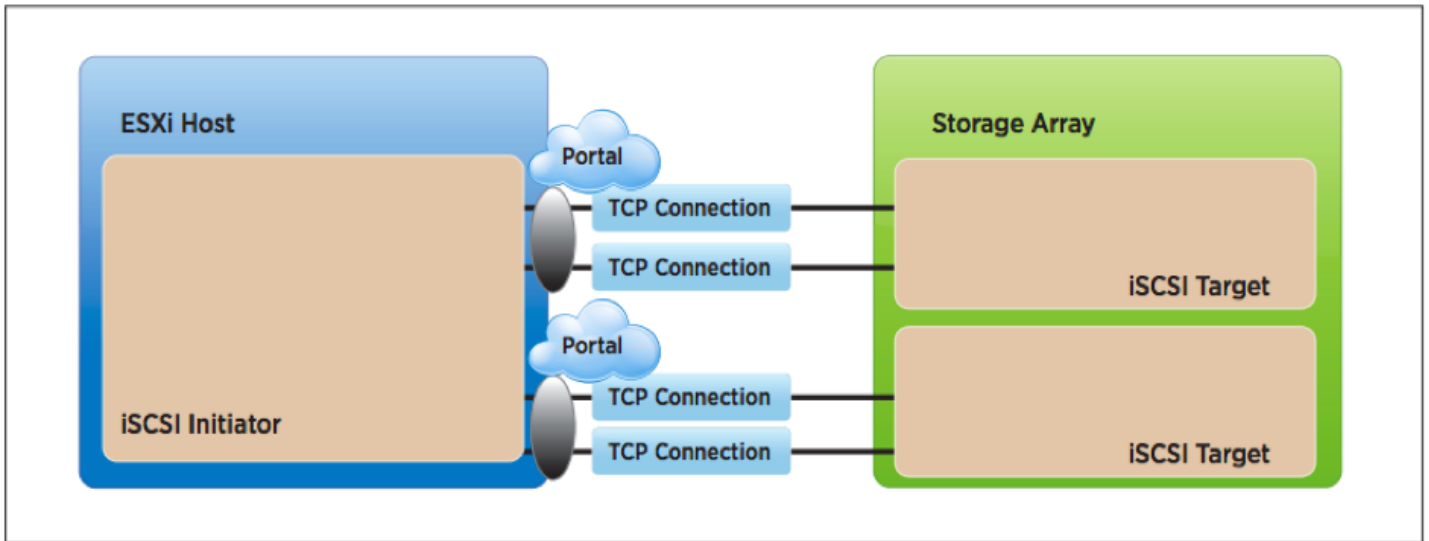


Figure 3 - iSCSI Portals

In earlier versions of vSphere, the VMware iSCSI driver sent I/O over one portal only (a single session per connection), and only when that failed did the vSphere host try to use other portals in a Round Robin fashion.

In more recent versions, this behavior changed so that the driver now logs in to all the portals that are returned in the **SendTarget** discovery response. The reason for this enhancement was to enable support for new active/passive iSCSI arrays that required support. With active/passive arrays, the vSphere host storage stack was required to recognize each of the portals as different paths (targets) to effectively do multipath failovers.

NOTE: Not all iSCSI arrays behave like this. Some arrays still require an administrator to add additional paths manually

iSCSI Implementation Options

VMware supports iSCSI with both software initiator and hardware initiator implementations. The software initiator iSCSI plugs into the vSphere host storage stack as a device driver in just the same way as other SCSI and FC drivers. This means that it implicitly supports the flagship file system of VMware, VMware vSphere VMFS, and also Raw Device Mappings (RDMs).

As previously mentioned, hardware iSCSI adapters fall into two categories – hardware dependent and hardware independent. Booting from iSCSI is also supported for both software and hardware iSCSI. Figure 4 shows the basic differences between an iSCSI hardware and iSCSI software implementation.

As of vSphere 6.5 iSCSI boot is also supported under UEFI boot mode. Note that the UEFI BIOS must have iSCSI support and that IPv6 is not supported at the time of writing.

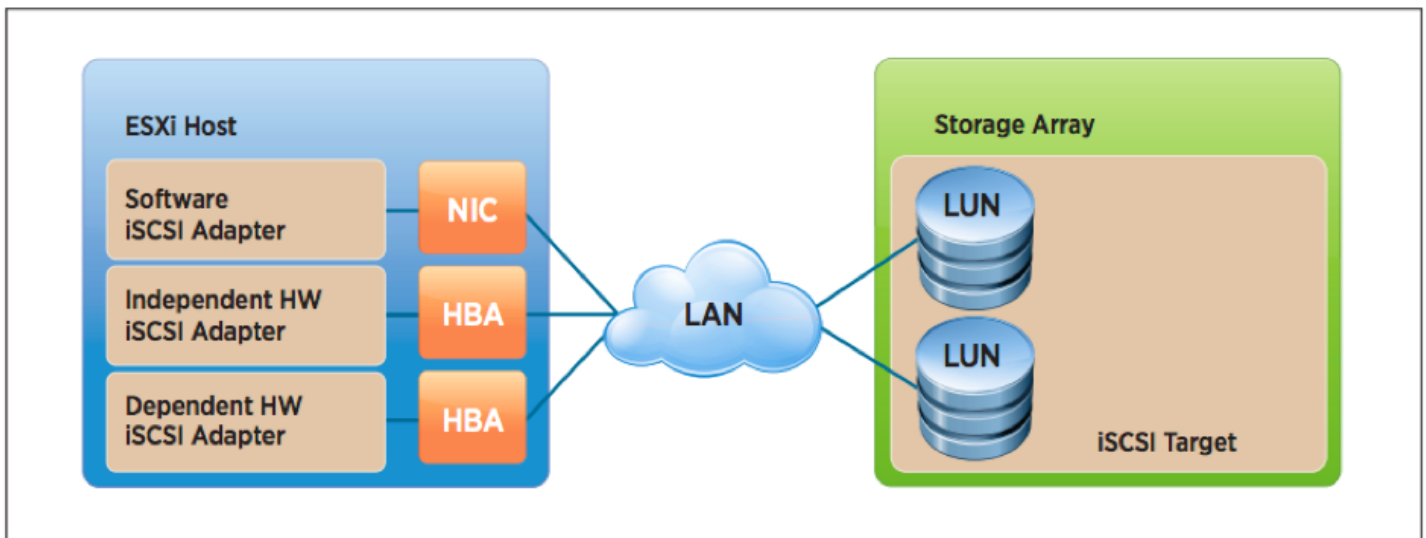


Figure 4 - Software and hardware iSCSI initiators

With the hardware-initiator iSCSI implementation, the iSCSI HBA provides the translation from SCSI commands to an encapsulated format that can be sent over the network. A TCP offload engine (TOE) does this translation on the adapter.

The software-initiator iSCSI implementation leverages the VMkernel to perform the SCSI to IP translation and requires extra CPU cycles to perform this work. As mentioned previously, most enterprise-level networking chip sets offer TCP offload or checksum offloads, which vastly improve CPU overhead.

Mixing iSCSI Options

Having both software iSCSI and hardware iSCSI enabled on the same host is supported. However, use of both software and hardware adapters on the same vSphere host to access the same target is not supported. One cannot have the host access the same target via hardware-dependent/hardware-independent/software iSCSI adapters for multi-pathing purposes. The reason for this support statement is that the different adapter types relate primarily to performance and management. For example, each adapter can generate different speeds.

Also, vSphere manages the software iSCSI adapters, but the hardware adapters have different management interfaces.

Finally, there can be differences in the offloading mechanism whereby the hardware adapters can offload by default, but for software iSCSI it will depend on the NIC. You might or might not have offload capabilities.

It's similar in many ways to presenting the same LUN from the same array via iSCSI and FC. You can see it over multiple paths and you can send I/O to it over multiple paths, but it would not be supported due to the differences highlighted previously.

However, different hosts might access the same iSCSI LUN via different methods. For example, host 1 might access the LUN using the software iSCSI adapter of VMware, host 2 might access it via a hardware-dependent iSCSI adapter and host 3 might access it via a hardware-independent iSCSI adapter.

Networking Settings

Network design is the key to making sure iSCSI works. In a production environment, gigabit Ethernet is essential for software iSCSI. Hardware iSCSI, in a VMware Infrastructure environment, is implemented with dedicated HBAs.

iSCSI should be considered a local-area technology, not a wide-area technology, because of latency issues and security concerns. You should also separate iSCSI traffic from general traffic. Layer-2 VLANs are a particularly good way to implement this separation.

Beware of oversubscription. Oversubscription occurs when more users are connected to a system than can be fully supported at the same time. Networks and servers are almost always designed with some amount of oversubscription, assuming that users do not all need the service simultaneously. If they do, delays are certain and outages are possible. Oversubscription is permissible on general-purpose LANs, but you should not use an oversubscribed configuration for iSCSI.

Best practice is to have a dedicated LAN for iSCSI traffic and not share the network with other network traffic. It is also best practice not to oversubscribe the dedicated LAN.

Finally, because iSCSI leverages the IP network, VMkernel NICs can be placed into teaming configurations. VMware's recommendation however is to use port binding rather than NIC teaming. Port binding will be explained in detail later in this paper but suffice to say that with port binding, iSCSI can leverage VMkernel multipath capabilities such as failover on SCSI errors and Round Robin path policy for performance.

In the interest of completeness, both methods will be discussed. However, port binding is the recommended best practice.

VMkernel Network Configuration

A VMkernel network is required for IP storage and thus is required for iSCSI. A best practice would be to keep the iSCSI traffic separate from other networks, including the management and virtual machine networks.

IPv6 Supportability Statements

Starting with vSphere 6.0 support for IPv6 was introduced for both hardware iSCSI and software iSCSI adapters leveraging static and automatic assignment of IP addresses.

Throughput Options

There are a number of options available to improve iSCSI performance.

1. 10GbE - This is an obvious option to begin with. If you can provide a larger pipe, the likelihood is that you will achieve greater throughput. Of course, if there is not enough I/O to fill a 1GbE connection, then a larger connection isn't going to help you. But let's assume that there are enough virtual machines and enough datastores for 10GbE to be beneficial.
2. Jumbo frames - This feature can deliver additional throughput by increasing the size of the payload in each frame from a default MTU of 1,500 to an MTU of 9,000. However, great care and consideration must be used if you decide to implement it. All devices sitting in the I/O path (iSCSI target, physical switches, network interface cards and VMkernel ports) must be able to implement jumbo frames for this option to provide the full benefits. For example, if the MTU is not correctly set on the switches, the datastores might mount but I/O will fail. A common issue with jumbo-frame configurations is that the MTU value on the switch isn't set correctly. In most cases, this must be higher than that of the hosts and storage, which are typically set to 9,000. Switches must be set higher, to 9,198 or 9,216 for example, to account for IP overhead. Refer to switch-vendor documentation as well as storage-vendor documentation before attempting to configure jumbo frames.
3. Round Robin path policy - Round Robin uses an automatic path selection rotating through all available paths, enabling the distribution of load across the configured paths. This path policy can help improve I/O throughput. For active/passive storage arrays, only the paths to the active controller will be used in the Round Robin policy. For active/active storage arrays, all paths will be used in the Round Robin policy. For ALUA arrays (Asymmetric Logical Unit Assignment), Round Robin uses only the active/optimized (AO) paths. These are the paths to the disk through the managing controller. Active/non-optimized (ANO) paths to the disk through the non-managing controller are not used. Not all arrays support the Round Robin path policy. Refer to your storage-array vendor's documentation for recommendations on using this Path Selection Policy (PSP).

Minimizing Latency

Because iSCSI on VMware uses TCP/IP to transfer I/O, latency can be a concern. To decrease latency, one should always try to minimize the number of hops between the storage and the vSphere host. Ideally, one would not route traffic between the vSphere host and the storage array, and both would coexist on the same subnet.

NOTE: If iSCSI port bindings are implemented for the purposes of multipathing, you could not route your iSCSI traffic pre- vSphere 6.5. With vSphere 6.5, routing of iSCSI traffic with port binding is supported.

Routing

A vSphere host has a single routing table for each TCP/IP stack. This imposes some limits on network communication for VMkernel interfaces using the same TCP/IP Stack. Consider a configuration that uses two Ethernet adapters with one VMkernel TCP/IP stack. One adapter is on the 10.17.1.1/24 IP network and the other on the 192.168.1.1/24 network. Assume that 10.17.1.253 is the address of the default gateway. The VMkernel can communicate with any servers reachable by routers that use the 10.17.1.253 gateway. It might not be able to talk to all servers on the 192.168 network unless both networks are on the same broadcast domain.

Another consequence of the single routing table affects one approach you might otherwise consider for balancing I/O. Consider a configuration in which you want to connect to iSCSI storage and also want to enable NFS mounts. It might seem that you can use one Ethernet adapter for iSCSI and a separate Ethernet adapter for NFS traffic to spread the I/O load. This approach does not work because of the way the VMkernel TCP/IP stack handles entries in the routing table.

For example, you might assign an IP address of 10.16.156.66 to the VMkernel adapter you want to use for NFS. The routing table then contains an entry for the 10.16.156.x network for this adapter. If you then set up a second adapter for iSCSI and assign it an IP address of 10.16.156.25, the routing table contains a new entry for the 10.16.156.x network for the second adapter. However, when the TCP/IP stack reads the routing table, it never reaches the second entry, because the first entry satisfies all routes to both adapters. Therefore, no traffic ever goes out on the iSCSI network, and all IP storage traffic goes out on the NFS network.

The fact that all 10.16.156.x traffic is routed on the NFS network causes two types of problems. First, you do not see any traffic on the second Ethernet adapter. Second, if you try to add trusted IP addresses both to iSCSI arrays and NFS servers, traffic to one or the other comes from the wrong IP address.

Using Static Routes or set a Gateway

As mentioned before, for vSphere hosts, the management network is on a VMkernel port and therefore uses the default VMkernel gateway. Only one VMkernel default gateway can be configured on a vSphere host per TCP/IP Stack. You can, however, add static routes from the command line or configure a gateway for each individual VMkernel port.

Setting a gateway on a per VMkernel port granular level has been introduced in vSphere 6.5 and allows for a bit more flexibility. The gateway for a VMkernel port can simply be defined using the vSphere Web Client during the creation of the VMkernel interface. It is also possible to configure it using `esxcli`.

Note: At the time of writing the use of a custom TCP/IP Stack is not supported for iSCSI!

Availability Options - Multipathing or NIC Teaming

To achieve high availability, the local-area network (LAN) on which the iSCSI traffic runs must be designed with availability, downtime avoidance, isolation and no single point of failure (SPOF) in mind. Multiple administrators must be involved in designing for high availability. These are the virtualization administrator and the network administrator (and maybe the storage administrator). This section outlines these steps and investigates several options, which can be utilized to make your iSCSI datastores highly available.

In both cases that follow, at least two network interface cards are required. Whereas 1Gb interfaces will meet the requirements for a highly available network, 10Gb network adaptors will also improve performance.

NIC Teaming for Availability

A best practice for iSCSI is to avoid the vSphere feature called teaming (on the network interface cards) and instead use port binding. Port binding introduces multipathing for availability of access to the iSCSI targets and LUNs. If for some reason this is not suitable, then teaming might be an alternative.

If you plan to use teaming to increase the availability of your network access to the iSCSI storage array, you must turn off port security on the switch for the two ports on which the virtual IP address is shared. The purpose of this port security setting is to prevent spoofing of IP addresses. Thus, many network administrators enable this setting. However, if you do not change it, the port security setting prevents failover of the virtual IP from one switch port to another and teaming cannot fail over from one path to another. For most LAN switches, the port security is enabled on a port level and thus can be set on or off for each port.

iSCSI Multipathing via Port Binding for Availability

Another way to achieve availability is to create a multi-path configuration. This is a more preferred method over NIC teaming, because this method will fail over I/O to alternate paths based on SCSI sense codes and not just network failures. Also, port bindings give administrators the opportunity to load-balance I/O over multiple paths to the storage device. Additional advantages around port binding will be discussed later in this paper.

Error Correction Digests

iSCSI header and data digests check the end-to-end, non-cryptographic data integrity beyond the integrity checks that other networking layers provide, such as TCP and Ethernet. They check the entire communication path, including all elements that can change the network-level traffic, such as routers, switches and proxies.

Enabling header and data digests does require additional processing for both the initiator and the target and can affect throughput and CPU usage.

Some systems can offload the iSCSI digest calculations to the network processor, thus reducing the impact on performance.

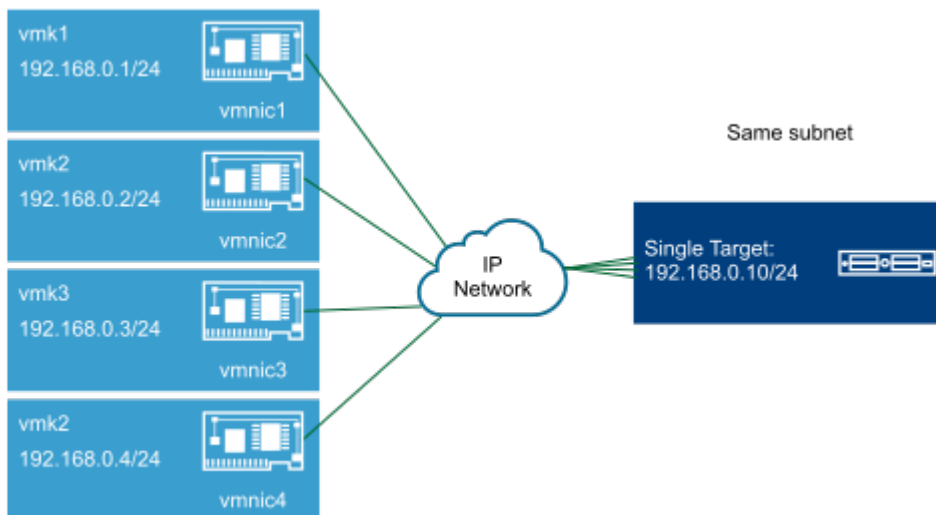
Flow Control

The general consensus from our storage partners is that hardware-based flow control is recommended for all network interfaces and switches.

iSCSI Port Binding Best Practices

With port binding, the SCSI protocol will not only load balance across all bound ports and failover to other bound ports on link failure, but it will also use SCSI sense code errors to trigger failover as well. When not using Port Binding, you are relying on vSphere and the network stack to determine the best path to use for iSCSI traffic. If paths are not clearly defined, other issues can arise such as longer scan times, and inconsistent connectivity. This isn't always the case but is something to consider.

Reliable storage should always be your priority. This overview is one of the most reliable and common configurations for iSCSI. A few key configurations for this setup are your virtual infrastructure VMkernels for iSCSI storage are on the same VLAN/subnet as your storage array, and the storage array controllers are also on that same subnet/VLAN.



For simplicity, let's say each vSphere host has two 25Gb NICs (NIC0, NIC1), and you are using SW iSCSI adapter. SW iSCSI adapters are the most common adapters used in vSphere environments and are capable of achieving near line rate. With modern CPUs, the minimal overhead of SW iSCSI is easily handled. This also reduces complexity as you no longer have to maintain HBAs and their firmware.

First, you must configure your virtual switches and VMkernels. There should be a portgroup and VMkernel for every NIC to be used for iSCSI. The configuration is similar for a standard or distributed vSwitch, the difference being whether you configure on each host or on vCenter. For this example, you will need to setup two portgroups with specific teaming configurations for each NIC. Each portgroup on a dVS, or when setting up a VMkernel on a standard vSwitch, must be setup as follows to support port binding.

For a distributed switch: You first create the portgroups, and then you create a VMkernel and associate it with a specific portgroup.

For a standard switch: You create the VMkernel first, the Portgroup is automatically created, then you go into each VMkernel's portgroup properties and change the Teaming and failover settings.

When configuring Teaming and failover, you may have to check Override to change the setting different than the default vSwitch. To be able to use port binding, there must be only one NIC active in the VMkernel/portgroup configuration. If there is a NIC in standby, instead of Unused, you will not be able to bind that VMkernel to iSCSI.

iSCSI-P1		iSCSI-P2		
Load Balancing	Use explicit failover order		Load Balancing	Use explicit failover order
Network failure detection	Link status only		Network failure detection	Link status only
Failback	No		Failback	No
Active Uplink	Uplink 1		Active Uplink	Uplink 2
Standby Uplink	None		Standby Uplink	None
Unused Uplink	Uplink2		Unused Uplink	Uplink 1

Distributed vSwitch

iSCSI-P1 - Edit Settings

General

Advanced

VLAN

Security

Teaming and failover

Traffic shaping

Monitoring

Miscellaneous

Load balancing

Use explicit failover order

Network failure detection

Link status only

Notify switches

Yes

Failback

No

Failover order ⓘ



Active uplinks

Uplink 1

Standby uplinks

Unused uplinks

Uplink 2

iSCSI-P2 - Edit Settings

General

Advanced

VLAN

Security

Teaming and failover

Traffic shaping

Monitoring

Miscellaneous

Load balancing

Use explicit failover order

Network failure detection

Link status only

Notify switches

Yes

Failback

No

Failover order ⓘ



Active uplinks

Uplink 2

Standby uplinks

Unused uplinks

Uplink 1

Standard vSwitch

iSCSI-P1 - Edit Settings

Properties

Security

Traffic shaping

Teaming and failover

Load balancing

Override

Use explicit failover order

Network failure detection

Override

Link status only

Notify switches

Override

Yes

Failback

Override

No

Failover order

Override



Active adapters

vmnic0

Standby adapters

Unused adapters

vmnic1

Select a physical network adapter from the list to view its details.

iSCSI-P2 - Edit Settings

Properties

Security

Traffic shaping



Teaming and failover

Load balancing	<input checked="" type="checkbox"/> Override	Use explicit failover order	▼
Network failure detection	<input type="checkbox"/> Override	Link status only	▼
Notify switches	<input type="checkbox"/> Override	Yes	▼
Failback	<input checked="" type="checkbox"/> Override	No	▼

Failover order

Override

↑ ↓

Active adapters
 vmnic1
Standby adapters
Unused adapters
 vmnic0

Select a physical network adapter from the list to view its details.

VMkernel properties:

Notice the only difference between the two VMkernels is which NIC uplink is active and the IP.

Configure Permissions VMs Datastores Networks

VMkernel adapters

Add Networking...
 Refresh |
 Edit...
 Remove

Device	Network Label	Switch	IP Address
vmk0	Management ...	vSwitch0	10.198.16.22
vmk1	iSCSI-P1	vSwitch0	10.198.20.29
vmk2	iSCSI-P2	vSwitch0	10.198.20.49

VMkernel network adapter: vmk1

All Properties IP Settings Policies

Security

Promiscuous mode	Reject
MAC address changes	Accept
Forged transmits	Accept

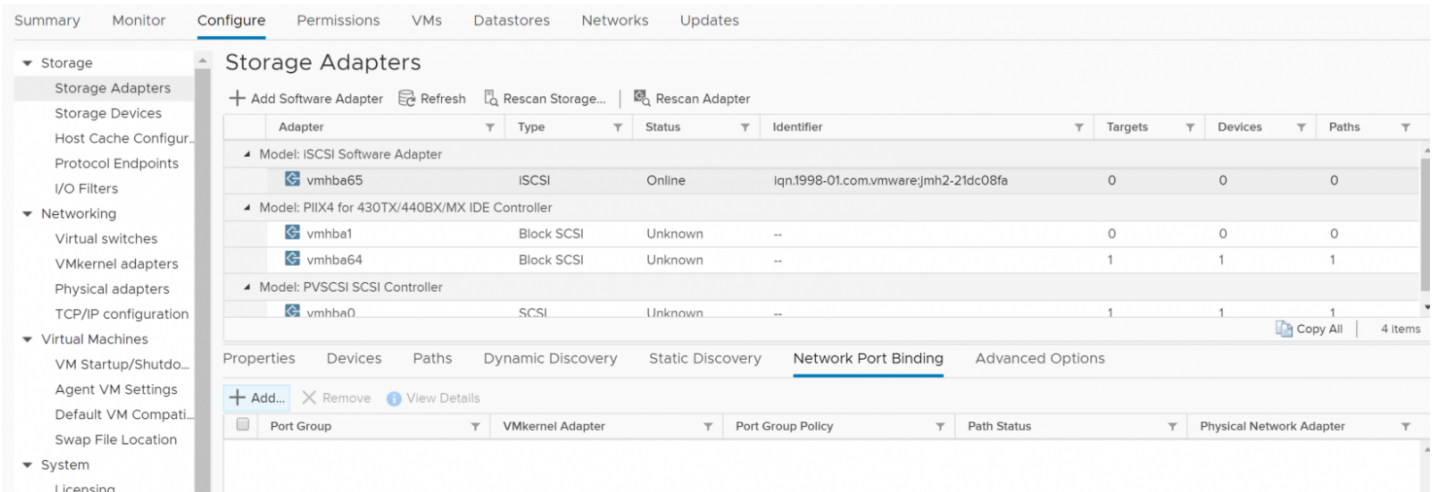
Traffic shaping

Average bandwidth	--
Peak bandwidth	--
Burst size	--

Teaming and failover

Load balancing	Use explicit failover order
Network failure detection	Link status only
Notify switches	Yes
Failback	No
Active adapters	vmnic0
Standby adapters	--
Unused adapters	vmnic1

Now you can configure your SW iSCSI adapter using port binding. Select your iSCSI adapter and choose the Network Port Binding tab, you will notice there are no bound ports. Click on +Add.



Here you can see the VMkernels previously configured. If the VMkernels were not configured properly, they will not show up as available. Check both and click OK.

Bind vmhba65 with VMkernel Adapter | jmh2.satm.eng.vmware.com

Only VMkernel adapters compatible with the iSCSI port binding requirements and available physical network adapters are listed.

Port Group	VMkernel Adapter	Physical Network Adapter
ISCSI-P1 (vSwitch0)	vmk1	vmnic0 (10 Gbit/s, Full)
ISCSI-P2 (vSwitch0)	vmk2	vmnic1 (10 Gbit/s, Full)

2 items

After you click OK, the VMkernel ports will be bound to the SW iSCSI adapter. It will also recommend you do a rescan on that adapter. Click the Rescan

Storage Adapters

⚠ Due to recent configuration changes, a rescan of "vmhba65" is recommended.

Adapter	Type	Status	Identifier	Targets	Devices	Paths
Model: iSCSI Software Adapter						
vmhba65	iSCSI	Online	iqn.1998-01.com.vmware:jmh2-21dc08fa	0	0	0
Model: PIIX4 for 430TX/440BX/MX IDE Controller						
vmhba1	Block SCSI	Unknown	--	0	0	0
vmhba64	Block SCSI	Unknown	--	1	1	1

Port Group	VMkernel Adapter	Port Group Policy	Path Status	Physical Network Adapter
ISCSI-P1 (vSwitch0)	vmk1	Compliant	Not used	vmnic0 (10 Gbit/s, Full)
ISCSI-P2 (vSwitch0)	vmk2	Compliant	Not used	vmnic1 (10 Gbit/s, Full)

With your VMkernels correct configured to use a specific NIC and your Vmkernels bound to your SW iSCSI adapter, you can proceed with adding targets from your array. Makes sure to add the iSCSI initiator to your array/LUNs you want to be accessible by the host. We will not go through that process as it is different for each vendor and array.

Storage Adapters

+ Add Software Adapter Refresh Rescan Storage... Rescan Adapter

Adapter	Type	Status	Identifier
Model: iSCSI Software Adapter			
vmhba65	iSCSI	Online	iqn.1998-01.com.vmware:jmh2-21dc08fa
Model: PIIX4 for 430TX/440BX/MX IDE Controller			
vmhba1	Block SCSI	Unknown	--
vmhba64	Block SCSI	Unknown	--

You can add targets either in the Dynamic Discovery or Static Discovery tabs. For this discussion, select Dynamic Discovery and click on +Add.

Add Send Target Server | vmhba65

iSCSI Server:

Port:

Inherit authentication settings from parent

Enter the IP of your array's discovery IP and click OK. Again, it will be recommended that you rescan the adapter to discover the targets. After the rescan, go into the Devices tab, and you should see any targets or LUNs you have configured to be accessible by this SW iSCSI initiator. After the rescan your devices will show up under the device tab and are now available. If it is a VMFS LUN, you can now create your VMFS datastore. If it is a vVols LUN, it may have a 512B, 1MB, or 1GB size, it varies between vendors. This LUN is not directly provisioned, it is the Protocol Endpoint for vVols and is used with the VASA provider to setup the IO path for each vVol. For more information on vVols please visit core.vmware.com

Additional iSCSI network configuration possibilities.

Now that we have covered the easiest and most common configuration for iSCSI port binding, let's review some other configurations.

Supported Port Binding Configurations:

- Single vSwitch, VMkernel ports in the same subnet/VLAN as storage array controllers
- VMKernels on one subnet/VLAN, storage controllers on a single separate subnet/VLAN

Unsupported Port Binding Configurations:

- Array controllers in different subnets/VLANs
 - Requires separate vSwitch for each VMKernel
- iSCSI routing below vSphere 6.5

For more detail on Port Binding configurations see the [iSCSI resource](https://core.vmware.com) page on core.vmware.com

Security Considerations

The following items comprise a list of considerations from a security perspective when implementing iSCSI.

Private Network

iSCSI storage traffic is transmitted in an unencrypted format across the LAN. Therefore, it is considered best practice to use iSCSI on trusted networks only and to isolate the traffic on separate physical switches or to leverage a private VLAN. All iSCSI-array vendors agree that it is good practice to isolate iSCSI traffic for security reasons. This would mean isolating the iSCSI traffic on its own separate physical switches or leveraging a dedicated VLAN (IEEE 802.1Q).

Encryption

iSCSI supports several types of security. IPsec (Internet Protocol Security) is a developing standard for security at the network or packet-processing layer of network communication. IKE (Internet Key Exchange) is an IPsec standard protocol used to ensure security for VPNs. Starting vSphere 6.0 IPsec is supported on vSphere host using IPv6 and the vSphere Software iSCSI Adapter. Note, enabling IPsec for the iSCSI interface will most likely have a performance impact. No performance testing has been done, as such no indication of the impact can be given. Please test the impact on your workload and I/O profile. VMware prefers the use of authentication in combination with a fully isolated network segment over encryption.

Authentication

There are also a number of authentication methods supported with iSCSI.

- Kerberos
- SRP (Secure Remote Password)
- SPKM1/2 (Simple Public-Key Mechanism)
- CHAP (Challenge Handshake Authentication Protocol)

At the time of this writing (vSphere 6.5), a vSphere host does not support Kerberos, SRP or public-key authentication methods for iSCSI. The only authentication protocol supported is CHAP. CHAP verifies identity using a hashed transmission. The target initiates the challenge. Both parties know the secret key. It periodically repeats the challenge to guard against replay attacks. CHAP is a one-way protocol, but it might be implemented in two directions to provide security for both ends. The iSCSI specification defines the CHAP security method as the only must-support protocol. The VMware implementation uses this security option.

iSCSI Datastore Provisioning Steps

Before a vSphere host can utilize iSCSI storage, the following configuration steps must be taken:

- Create a new VMkernel portgroup for IP storage on an already existing virtual switch (vSwitch) or on a new vSwitch when it is configured. The vSwitch can be a vSphere Standard Switch (VSS) or a VMware vSphere Distributed Switch™.
- Ensure that the iSCSI initiator on the vSphere host(s) is enabled.
- Ensure that the iSCSI storage is configured to export a LUN accessible to the vSphere host iSCSI initiators on a trusted network.

For network connectivity, the user must create a new VMkernel portgroup to configure the vSwitch for IP storage access. The user must also populate the network access information, including any VLAN tag associated with the storage network.

Figure 5 - VMkernel Connection Settings

To enable the iSCSI initiator, additional details such as the IP address of the target array and any CHAP-related credentials must also be added.

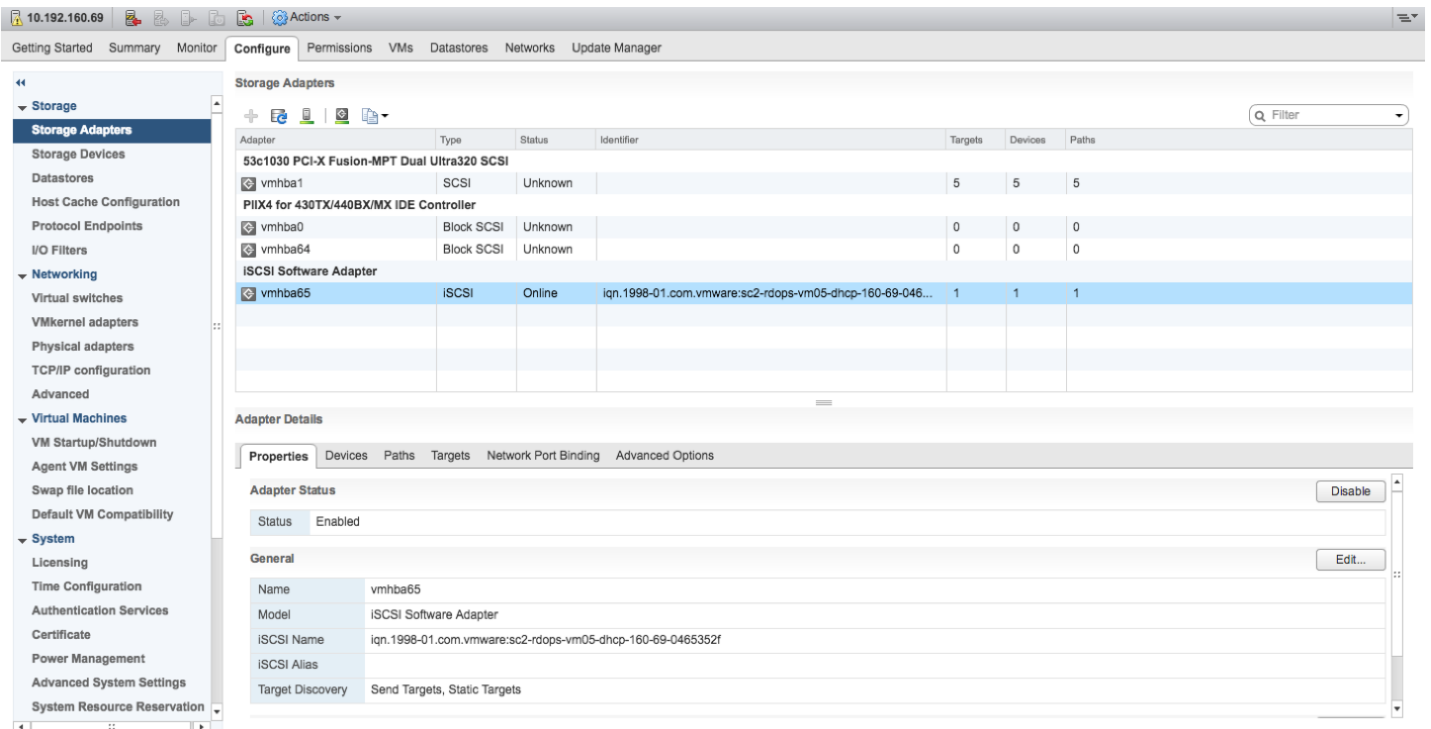


Figure 6 - Enable Software iSCSI Adapter

At this point, port binding details might also be added for multipathing purposes.

Why Use iSCSI Multipathing?

The primary use case of this feature is to create a multipath configuration with storage that presents only a single storage portal, such as the DELL EqualLogic and the HP StoreVirtual iSCSI solutions. Without iSCSI multipathing, this type of storage would have one path only between the vSphere host and each volume. iSCSI multipathing enables us to multipath to this type of clustered storage.

Another benefit is the ability to use alternate VMkernel networks outside of the vSphere host management network. This means that if the management network suffers an outage, you continue to have iSCSI connectivity via the VMkernel ports participating in the iSCSI bindings.

NOTE: VMware considers the implementation of iSCSI multipathing (Port Binding) over NIC teaming as a best practice.

Software iSCSI Multipathing Configuration Steps

In this example, We configure a software iSCSI adapter, vmhba32. At this point, no targets have been added, so no devices or paths have been discovered. Before implementing the software iSCSI bindings, you must create a number of additional VMkernel ports (vmk) for port binding to the software iSCSI adapter.

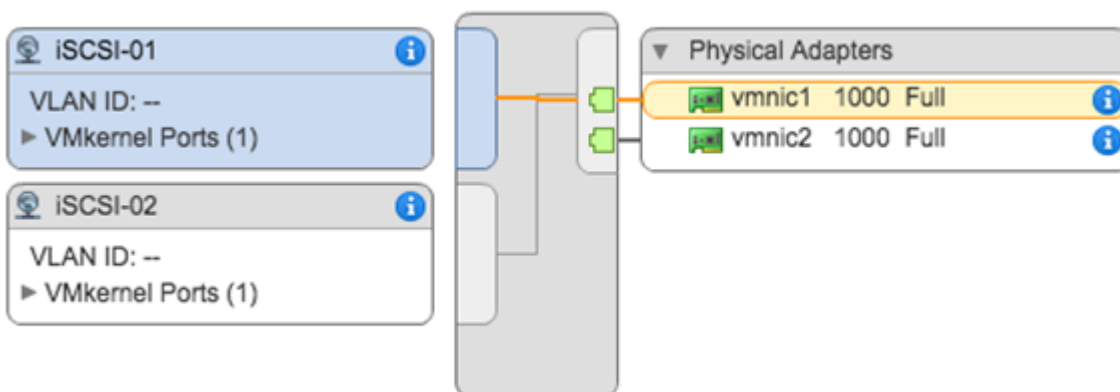


Figure 7 - Single VSS with Two VMkernel Ports

For port binding to work correctly, the initiator must be able to reach the target directly on the same subnet prior to vSphere 6.5 - **iSCSI port binding only supports routing in vSphere 6.5** . In this configuration, if you place a VMkernel ports on VLAN 74, they can reach the iSCSI target without needing to route. This is an important point and requires further elaboration because it causes some confusion. If you do not implement port binding and use a standard VMkernel port, then the initiator can reach the targets through a routed network, regardless of the vSphere version. This is supported and works well. It is only when iSCSI binding is implemented prior to vSphere 6.5 that a direct, non-routed network between the initiators and targets is required. In other words, initiators and targets must be on the same subnet.

There is another important point to note when it comes to the configuration of iSCSI port bindings. On VMware standard switches that contain multiple vmnic uplinks, each VMkernel (vmk) port used for iSCSI bindings must be associated with a single vmnic uplink. The other uplink(s) on the vSwitch must be placed into an **unused** state. This is only a requirement when there are multiple vmnic uplinks on the same vSwitch. If you are using multiple VSSs with their own vmnic uplinks, then this is not an issue.

Continuing with the network configuration, a second VMkernel (vmk) port is created. Now there are two vmk ports, labeled iSCSI-01 and iSCSI-02. These will be used for the iSCSI port binding/multipathing configuration. The next step is to configure the bindings and iSCSI targets. This is done in the properties of the software iSCSI adapter which can be found on the vSphere host under “Configure >> Storage Adapters >> iSCSI Software Adapter”. In this section click on “Network Port Binding” followed by the green plus sign.

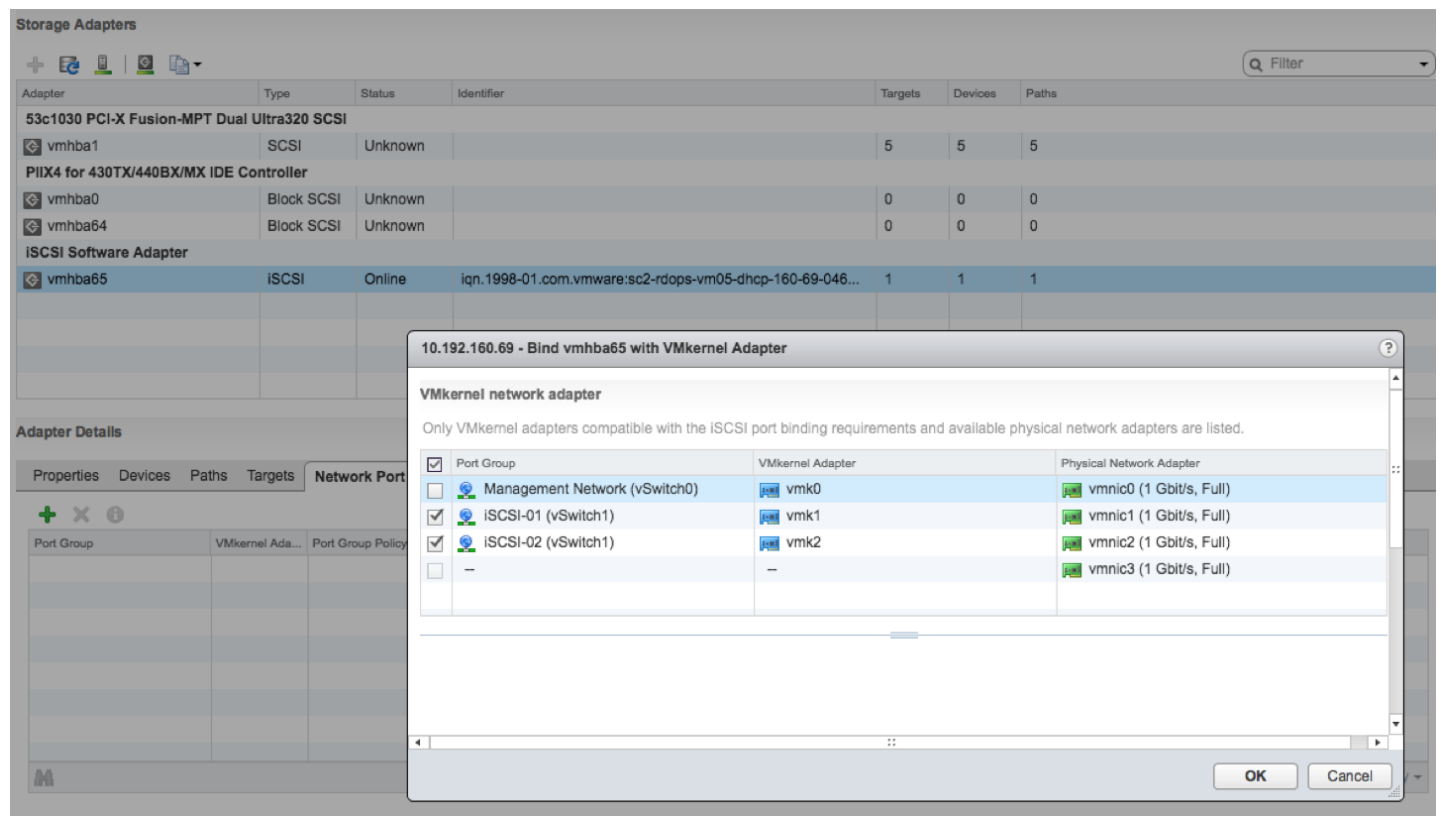


Figure 8 - Network Port Binding

After selecting the VMkernel adapters for use with the software iSCSI adapter, the **Port Group Policy** tab will tell you whether or not these adapters are compliant for binding. If you have more than one **active** uplink on a vSwitch that has multiple vmnic uplinks, the vmk interfaces will not show up as compliant. Only one uplink should be **active** . All other uplinks should be placed into an **unused** state.

The next step is to proceed to the **Targets** tab, where the iSCSI targets can now be added. Because port binding is being used, iSCSI targets must be reachable by the software iSCSI adapter through a non-routable network if the vSphere release is earlier than 6.5. In other words, the storage controller ports are on the same subnet as the VMkernel NICs.

At this point, there are two VMkernel ports bound to the software iSCSI adapter and connected to an iSCSI target or targets (which are in effect array controllers). These targets are all going to the same storage array, so a LUN presented out on all targets will be visible across multiple paths.

After adding the target ensure to rescan the iSCSI Software Adapter so that all targets, devices, and all paths to the devices can be discovered.

The screenshot shows the 'Storage Adapters' configuration page in VMware vSphere. It features a table of storage adapters and a detailed view of the selected 'iSCSI Software Adapter'.

Adapter	Type	Status	Identifier	Targets	Devices	Paths
53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI						
vmhba1	SCSI	Unknown		5	5	5
PIIX4 for 430TX/440BX/MX IDE Controller						
vmhba0	Block SCSI	Unknown		0	0	0
vmhba64	Block SCSI	Unknown		0	0	0
iSCSI Software Adapter						
vmhba65	iSCSI	Online	iqn.1998-01.com.vmware:sc2-rdops-vm05-dhcp-160-69-046...	1	1	3

Adapter Details

Properties | Devices | Paths | **Targets** | Network Port Binding | Advanced Options

Dynamic Discovery | Static Discovery

Add... Remove Authentication... Advanced...

iSCSI server
10.192.191.138:3260

Figure 9 - Add Target

Sizing Considerations - Recommended Volume Size

Sizing of volumes is typically proportional to the number of virtual machines you attempt to deploy, in addition to snapshots/changed blocks created for backup purposes. Another consideration is that many arrays have deduplication and compression features, which will also reduce capacity requirements. A final consideration is Recovery Point Objective (RPO) and Recovery Time Objective (RTO). These determine how fast you can restore your datastore with your current backup platform. When considering volume sizes, the type of media on the array. For example, an array with spinning media needs more spindles for performance, and it's usually recommended to have more, smaller sized datastores. Conversely, an all-flash array has plenty of performance and it is generally recommended to have fewer, larger datastores.

Something else to think about is if you are using vVols for your datastore, then you will only have a single vVols datastore. For more information on vVols, see the [vVols Getting Started Guide](#)

Each storage vendor has its own recommendations for sizing and scaling iSCSI LUNs. VMware recommends following the storage vendor's best practices to ensure optimal performance.

Recommended Block Size

This parameter is not tunable, for the most part. Some vendors have it hard set to 4KB and others have it hard set to 8KB. Block sizes are typically a multiple of 4KB. These align nicely with the 4KB grain size used in the VMDK format of VMware. For those vendors who have it set to 8KB, the recommendation is to format the volumes in the guest operating system (OS) to a matching 8KB block size for optimal performance. In this area, it is best to speak to your storage-array vendor to get vendor-specific advice.

The questions one should ask are as follows:

1. What is the volume block size on the array?
2. Is it tunable?
3. If so, what should I tune it to? Be sure to explain that the datastore will be used by virtual machines, so the workload will be random for the most part.
4. Are there any considerations when formatting the volumes in the guest OS?

Booting a vSphere Host from Software iSCSI

VMware introduced support for iSCSI with ESX 3.x. However, ESX could boot only from an iSCSI LUN if a hardware iSCSI adapter was used. Hosts could not boot via the software iSCSI initiator of VMware. In vSphere 4.1, VMware introduced support making it possible to boot the host from an iSCSI LUN via the software iSCSI adapter. This includes support for both legacy BIOS as well as UEFI mode.

Not all of our storage partners support iSCSI Boot Firmware Table (iBFT) boot from SAN. Refer to the partner's own documentation for clarification.

Why Boot from SAN?

It quickly became clear that there was a need to boot via software iSCSI. Partners of VMware were developing blade chassis containing blade servers, storage and network interconnect in a single rack. The blades were typically diskless, with no local storage. The requirement was to have the blade servers boot off of an iSCSI LUN using network interface cards with iSCSI capabilities, rather than using dedicated hardware iSCSI initiators.

Compatible Network Interface Card (NIC)

Much of the configuration for booting via software iSCSI is done via the BIOS settings of the network interface cards and the host. Check the [VMware Hardware Compatibility List \(HCL\)](#) to ensure that the network interface card is compatible. This is important, but a word of caution is necessary. If you select a particular network interface card and you see iSCSI as a feature, you might assume that you can use it to boot a vSphere host from an iSCSI LUN. This is not the case.

To see if a particular network interface card is supported for iSCSI boot, set the I/O device type to Network (not iSCSI) in the HCL and then check the footnotes. If the footnotes state that **iBFT** is supported, then this card can be used for boot from iSCSI.

Configuring Host BIOS for iSCSI Boot

After it is verified that the network interface card is supported, the next step is to enter the BIOS of the network interface card and ensure that it is enabled for iSCSI boot. Next would be the network interface card configuration. For example, if you were planning to boot from software iSCSI with a Broadcom NetXtreme network interface card, this comes with a boot agent, Broadcom's Multi-Boot Agent (MBA) software utility. This agent enables a host to execute a boot process using images from remote servers, including iSCSI targets. In the MBA configuration menu, iSCSI must be selected as the boot protocol. If iSCSI isn't available as a boot protocol, it might mean that the iSCSI firmware has not been installed or that iSCSI has not been enabled on the network interface card. Refer to your network interface card vendor documentation for further information.

There are a number of different parameters to configure. When doing the initial installation, **Boot to iSCSI Target** must also be left **Disabled**. You must change it to Enabled for subsequent boots.

In the **initiator parameters** section, one would enter values for the IP address, subnet mask, default gateway, primary DNS, and secondary DNS parameters of the initiator as needed. If authentication is required, then the CHAP ID (Challenge Handshake Authentication Protocol ID) and CHAP secret parameters should be entered.

In the **target parameters** section, one would enter values for the target IP address, target name, and login information. If authentication is required, then once again the CHAP ID and CHAP secret parameters must be entered.

NOTE: The Boot LUN ID (the LUN on the target that is used for the vSphere host installation and subsequent boots) should also be configured.

Exit and save to complete the BIOS configuration. Whereas this sequence of steps is for the Broadcom MBA software utility, a similar set of steps must be taken for other network interface cards. We are now ready to install a vSphere host onto an iSCSI LUN via the software iSCSI initiator.

Installing a vSphere Host on an iSCSI LUN

After configuring the MBA parameters in the Broadcom network interface card, you can now go ahead with the installation of the vSphere host. The installation media is placed in the CD-ROM or made available via some other method in the BIOS on the host (for example, virtual media). The next step is to ensure that the boot controller/device order is set correctly in the BIOS. For network interface cards, the network adapter should appear before the CD-ROM in the boot order.

To simplify the procedure of booting from iSCSI, ensure that the boot LUN is initially mapped on one path only to the vSphere host. Another advisable step is to map LUN id 0 to the host. Although this requirement changes from storage array to storage array, it is easier to simply follow this instruction rather than to consult the documentation of the storage-array vendor to determine if it is required or not.

When the host is powered on, the system BIOS loads the OptionROM code of the network interface card and starts executing it. The OptionROM contains boot code and iSCSI initiator firmware. The iSCSI initiator firmware establishes an iSCSI session with the target.

On boot, a successful login to the target should be observed before installation starts. If you get a failure at this point, you must revisit the configuration steps done previously. The installation now begins.

As part of the installation process, a memory-only stateless VMkernel is first loaded. This must discover suitable LUNs for installation, one of which is the iSCSI LUN. However, for the VMkernel's iSCSI driver to communicate with the target, it requires that the TCP/IP protocol be set up. This is all done as part of the startup **init** script. The OptionROM of the network interface card is also responsible for handing off the initiator and target configuration data to the VMkernel. The handoff protocol is the **iBFT**. After the required networking is set up, an iSCSI session is established to the target configured in the iBFT. LUNs beneath the targets are discovered and registered with VMkernel SCSI stack (PSA).

If everything is successful during the initial installation, the iSCSI LUN is offered as a destination for the vSphere host image. You can now complete the vSphere host installation normally.

Booting from an iSCSI LUN

After the installation is complete, a single iSCSI configuration change is required in the iSCSI configuration. Again, using Broadcom NetXtreme as an example, the **Boot to iSCSI target** is set to **Enabled**. When the host is rebooted, it will boot the vSphere host from the iSCSI LUN via the software iSCSI initiator.

Troubleshooting Checklist

This paragraph contains a list of items that should be checked in the event that issues are encountered when booting from iSCSI.

- Ensure that the network interface card is on the HCL for iSCSI boot. Remember to check the footnotes.
- Make sure that your device has a firmware version that supports iSCSI boot and that the iSCSI configuration settings for initiator and target are valid.
- Check the login screen to make sure your initiator can log in to the target. If you make changes to the physical network, these must be reflected in the iBFT.
- Finally, the CLI command, **esxcli iscsi ibftboot get** , displays the iBFT boot values.

Additional Considerations

This section provides a list of additional considerations to keep in mind.

Disk Alignment

This is not a recommendation specific to iSCSI, because it also can have an adverse effect on the performance of all block storage. Nevertheless, to account for every contingency, it should be considered a best practice to have the partitions of the guest OS running with the virtual machine aligned to the storage. Detailed descriptions of the way to do this alignment are beyond the scope of this white paper. Refer to the documentation of your individual storage array vendor for further details.

Microsoft Clustering Support

Starting with the release of vSphere 5.1, VMware supports as many as five nodes in a Microsoft Cluster. At the time of writing, this is still the supported maximum, even with vSphere 6.x. Also, as of vSphere 5.5, VMware supports the cluster quorum disk over the iSCSI protocol. For the latest details, refer to <https://kb.vmware.com/kb/2147661>

In-Guest iSCSI Support

A number of in-guest iSCSI software solutions are available. The iSCSI driver of Microsoft is one commonly seen running in a virtual machine when the guest OS is a version of Microsoft Windows. The supporting statement for this driver can be found in [KB article 1010547](#), which states that “if you encounter connectivity issues using a third-party software iSCSI initiator to the third-party storage device, engage the third-party vendors for assistance. If the third-party vendors determine that the issue is due to a lack of network connectivity to the virtual machine, contact VMware for troubleshooting assistance.”

All Paths Down and Permanent Device Loss

All Paths Down (APD) can occur on a vSphere host when a storage device is removed in an uncontrolled manner or if the device fails and the VMkernel core storage stack cannot detect how long the loss of device access will last. One possible scenario for an APD condition is an FC switch failure that brings down all the storage paths, or, in the case of an iSCSI array, a network connectivity issue that similarly brings down all the storage paths.

Another condition that can occur is Permanent Device Loss (PDL). The PDL condition enables the vSphere host to take specific actions when it is detected that the device loss was permanent. The vSphere host can be informed of a PDL situation by specific SCSI sense codes sent by the target array.

vSphere also supports PDL detection for those arrays that have only a single LUN per target. For those iSCSI arrays that have a single LUN per target, an attempt is made to log in again to the target after a dropped session. If there is a PDL condition, the storage system rejects the effort to access the device. Depending on how the array rejects the efforts to access the LUN, the vSphere host can determine whether the device has been lost permanently (PDL) or is temporarily unreachable.

After LUN has been declared to be in an APD or PDL state the vSphere host is able to halt the VMs impacted and have vSphere HA restart these workloads on hosts that still have access to the datastore. Please consult the [vSphere Availability Guide](#) for more details on how to configure VM Component Protection (VMCP)

Read-Only File Systems on Linux Guest OS

VMware has identified a problem with certain Linux guest operating systems. RHEL 5 (GA), RHEL 4 U4, RHEL 4 U3, SLES 10 (GA) and SLES 9 SP3 might have their file systems become read-only in the event of a busy I/O retry or path failover of the ESXi host's SAN or iSCSI storage. This Linux kernel bug has been fixed in different updates of the different Linux distributions. See VMware [KB article 51306](#) for further details.

Round Robin Path Policy Setting IOPS=1

A number of our partners have documented that if using the Round Robin path policy, best results can be achieved with an IOPS=1 setting. This might well be true in very small environments where there are a small number of virtual machines and a small number of datastores. However, because the environment scales with a greater number of virtual machines and a greater number of datastores, VMware considers that the default settings associated with the Round Robin path policy to be sufficient. Consult your storage array vendor for advice on this setting.

In vSphere 6.7, VMware introduced a new Latency based Round Robin policy that intelligently evaluates all paths and chooses the best path(s) to used based on a working average. For more information, see the announcement article [here](#).

Tape Device Support

vSphere hosts do not support iSCSI-connected tape devices.

iSCSI Docs and KB Resources

[iSCSI Best Practice Resources](#)

Conclusion and About the Author

This section includes the conclusion, in addition to acknowledgements and a few lines about the author.

Conclusion

iSCSI is now an extremely popular storage protocol found in vSphere infrastructures. This white paper brings together various disparate pieces of knowledge and documentation for VMware customers who are considering implementing iSCSI or have already implemented iSCSI and are looking for some best practices. This document should be used in conjunction with documentation available from our storage array partners.

