

EMC VMAX ALL FLASH AND VMAX3 iSCSI DEPLOYMENT GUIDE FOR WINDOWS ENVIRONMENTS

EMC VMAX Engineering White Paper

ABSTRACT

This white paper provides guidelines and best practices for deploying iSCSI with EMC® VMAX® All Flash and VMAX3™ systems in Microsoft environments, including with SQL Server databases.

June, 2016

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, [contact](#) your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the [EMC Store](#).

Copyright © 2016 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided "as is." EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see [EMC Corporation Trademarks](#) on EMC.com.

All other trademarks used herein are the property of their respective owners.

Part Number H15143

TABLE OF CONTENTS

EXECUTIVE SUMMARY	5
AUDIENCE.....	5
VMAX ISCSI OVERVIEW	6
Core iSCSI components	6
iSCSI initiator node	6
iSCSI target nodes	6
iSCSI names.....	7
iSCSI IP interfaces (iSCSI network portals)	7
iSCSI sessions and connections.....	7
VMAX iSCSI scale limits	7
Windows host connectivity using VMAX iSCSI	8
DESIGN CONSIDERATIONS AND BEST PRACTICES.....	8
Network design and host connectivity considerations	8
VMAX iSCSI connectivity considerations	9
VMAX iSCSI multi-tenancy	9
Host I/O limits	10
TEST ENVIRONMENT.....	10
Hardware and software configuration.....	10
Tested workloads	10
TEST CASES	11
Test 1: Single SQL Server database OLTP test using iSCSI and FC connectivity.....	11
Test motivation.....	11
Test configuration	11
Test results	11
Test conclusion	11
Test 2: Repurposing SQL DB for Test and Dev using iSCSI connectivity	12
Test motivation.....	12
Test configuration	12
Test results	12
Test conclusion	12

CONCLUSION 13

REFERENCES 13

APPENDIX 14

- Configuring iSCSI using Unisphere for VMAX 14
- Configuring VMAX iSCSI using SYMCLI..... 19
- Some useful SYMCLI VMAX iSCSI commands..... 21
- Microsoft Windows host initiator configuration using GUI..... 23
- Windows host iSCSI configuration using PowerShell cmdlet 24

EXECUTIVE SUMMARY

This white paper provides an overview of Internet Small Computer Systems Interface (iSCSI) implementation on VMAX® All Flash - and VMAX3™ arrays and discusses best practices for its deployment in a Windows environment with specific examples of Windows 2012 R2 Server and Microsoft SQL Server 2014. iSCSI is an industry standard for accessing SCSI storage over internet protocol. With VMAX iSCSI, support was redesigned to provide customers with greater port connection densities using virtual storage ports, built-in multi-tenancy capabilities using VLAN, and easier isolation using VMAX initiator groups. This paper compares the performance between Fibre Channel (FC) and iSCSI and provides a use case for using iSCSI for test and development environments. As iSCSI continues to increase in use and popularity in the enterprise space, leveraging iSCSI for SQL database deployments on VMAX All Flash and VMAX3 arrays is easy and provides high performance, resiliency, protection, and security with lower cost of ownership compared to FC.

Note: This white paper is focused on VMAX All Flash, however the iSCSI implementation on VMAX3 is identical. Throughout this paper, references to VMAX refer to both VMAX All Flash and VMAX3.

The use of iSCSI offers many advantages, including:

- High performance and bandwidth due to higher adoption of 10 GbE and faster network interfaces. IP-based connectivity can now deliver bandwidth equivalent to or faster than 8 Gb FC SAN networks for most workloads. For online transaction processing (OLTP) workloads, iSCSI and FC provide nearly identical performance.
- Benefits of converging storage and network infrastructure in the data center. These benefits include cost savings from maximizing existing network management skills, unifying infrastructure components, and the added simplicity of IP-based connectivity.
- Increased scale, using VMAX virtual storage ports. Often iSCSI deployments can only allocate a single IP address to each storage target port, limiting the deployment scale. VMAX iSCSI targets are designed around virtual storage ports to overcome these limitations.
- Strong security using uni-directional or bi-directional Challenge-Handshake Authentication Protocol (CHAP) authentication.
- Multi-tenancy and network isolation, leveraging VLAN and VMAX host initiator groups. VLANs provide virtual networks so iSCSI traffic can be isolated from other network activity, or other tenants. VMAX host initiator groups are part of the VMAX device-masking configuration which allows fast and flexible changes to relationships between host initiators, storage target ports, and storage devices. Only the participating members of a masking view are visible to each other.
- Improved support for lower-cost test/dev environments. As demonstrated in this paper, even when existing databases use FC interfaces, VMAX SnapVX™ can easily create database snapshots that can be accessed using iSCSI, for example, by development or QA groups.

AUDIENCE

This white paper is intended for database and system administrators, storage administrators, and system architects who are responsible for implementing, managing, and maintaining Windows servers and VMAX storage systems. Readers should have some familiarity with the Windows Server operating system, SQL Server, Network Administration, and the VMAX family of storage arrays.

VMAX iSCSI OVERVIEW

Internet Small Computer Systems Interface (iSCSI) is an IP-based storage networking standard that is mainly used for linking hosts to storage devices. By transmitting iSCSI commands over IP networks, iSCSI can facilitate block-level transfers over IP based networks. The iSCSI architecture is similar to a client/server architecture. In this case, the client is an initiator, like a Microsoft Windows or Linux host that issues an I/O request, and the server is a target such as VMAX array.

iSCSI can leverage existing investments in Ethernet infrastructures and expertise. It can use existing network infrastructure of switches, routers, and network adapters instead of requiring additional hardware. With the proliferation of 10 GbE networking, iSCSI has steadily gained popularity as a deployed storage protocol in data centers, as it now it can deliver performance on par with 8 Gb FC SAN. iSCSI provides additional types of host interfaces for VMAX storage system, giving customers a wide variety of choices for connectivity, including FC, iSCSI, Fibre Channel over Ethernet (FCoE), and network attached storage (NAS). Microsoft Windows Server 2012 R2 has a built-in iSCSI initiator and easy to use GUI interface for its management. Support for VLANs offers network partitioning and traffic isolation in multi-tenant environments and CHAP addresses iSCSI security concerns by enabling access to clients that supply valid authentication.

Key VMAX iSCSI design characteristics:

- Separation of the iSCSI target from the physical port
- Individual iSCSI targets can have multiple network portals
- Support for VLANs, network namespaces (Network IDs), and routing and target nodes from the physical ports on each director and iSCSI storage module in Solutions Enabler.

Core iSCSI components

An iSCSI architecture is made up of a set of core components. These components include initiator and target nodes, iSCSI names, IP interfaces, sessions, connections, and security. This section details each of the components. Figure 1 shows the relationships between iSCSI target node and network portals.

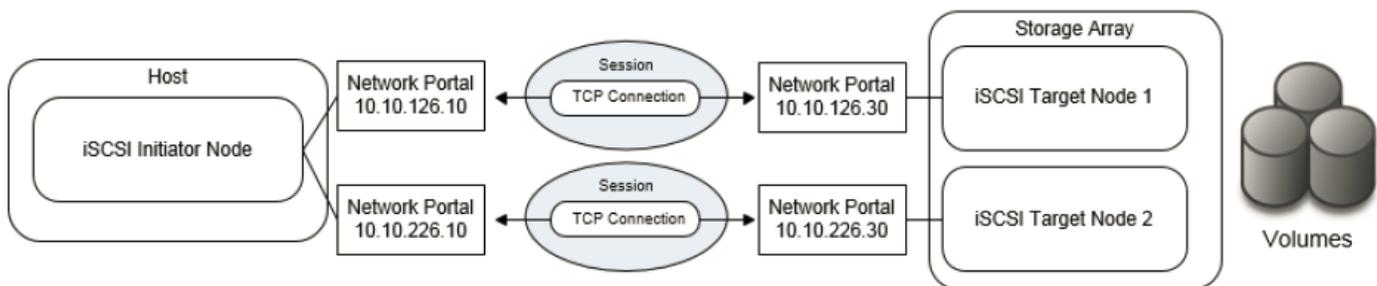


Figure 1. Core iSCSI components

iSCSI initiator node

iSCSI initiator nodes such as hosts are the data consumers. The iSCSI initiator can be implemented either as a driver installed on the host system or within the hardware of an iSCSI HBA, which typically includes TCP/IP Offload Engine (TOE). The host initiates a request to and receives responses from an iSCSI target (storage node). iSCSI initiators must manage multiple, parallel communication links to multiple targets.

iSCSI target nodes

iSCSI target nodes, such as disk arrays or tape libraries, are data storage providers.

iSCSI target nodes expose one or more SCSI LUNs to specific iSCSI initiators. The target node listens and responds to commands from iSCSI initiators on the network. On the enterprise storage level, iSCSI target nodes are logical entities, not tied to a specific physical port. iSCSI targets must manage multiple, parallel communication links to multiple initiators.

In a VMAX iSCSI implementation, iSCSI target nodes are also referred to as storage virtual ports to indicate the separation of a target node from its physical port. Multiple target nodes can be associated with each physical port and provide more scale and flexibility.

iSCSI names

iSCSI initiator and target nodes are identified by a unique iSCSI name. iSCSI names are ASCII strings and must be unique on a per-namespace (Network ID) basis. iSCSI names would ideally be unique worldwide, but since they are both user- and algorithmically-generated, there can be duplicates even on the same array. iSCSI names are formatted in two different ways:

- Enterprise Unique Identifier (EUI)—for example: **eui.0123456789ABCDEF**
- iSCSI Qualified Name (IQN), which is the most commonly used naming format—for example: **iqn.VMAX.Finance**

Note: As IQN formatting is most common, the examples in this paper are all based on IQN.

iSCSI IP interfaces (iSCSI network portals)

iSCSI target nodes are accessed through IP interfaces (also called network portals). iSCSI network portals contain key network configuration information such as:

- IP Address
- Network ID¹
- Virtual LAN (VLAN) information
- Maximum transmission unit (MTU)

An iSCSI network portal can only provide access to a single iSCSI target node; however, you can access an iSCSI target node through multiple network portals. These portals can be grouped together to form a portal group. Portal groups are identified by a unique portal group tag (network ID) and defined for the iSCSI target node. All portals in a portal group must provide access to the same iSCSI target node.

iSCSI sessions and connections

iSCSI initiator and target nodes communicate by a linkage called an iSCSI session. The session is the vehicle for the transport of the iSCSI packets, or Portal Data Units (PDUs) between the initiators and target nodes. Each session is started by the initiator, which logs into the iSCSI target. The session between the initiator and target is identified by an iSCSI session ID. Session IDs are not tied to the hardware and can persist across hardware swaps.

Session components are tied together by a TCP/IP connection. The IP addresses and TCP port numbers in the network portals (IP interfaces) define the end points of a connection.

VMAX iSCSI scale limits

The following is a list of the current support limits/scale. Refer to VMAX iSCSI release notes for up-to-date information:

- Maximum of 64 targets per physical port
- Maximum of 8 network portals (IP interfaces) per target
- Maximum of 512 network IDs (Range is 0–511)
- Maximum of 1024 routing instances per engine

¹ In a VMAX implementation, a specific user-defined numeric value is used to map targets to an IP interface.

Windows host connectivity using VMAX iSCSI

Figure 2 shows all the components of an iSCSI-based Windows host to the VMAX array. In this scenario, physical ports on a VMAX iSCSI IO module support multiple virtual target ports for highly available storage target access. Storage groups from a VMAX array are accessible through two targets and independent network paths. IP interfaces of the VMAX array and Microsoft Windows server were configured with a VLAN. We recommend using VLANs for logical separation of iSCSI traffic for each host or group of hosts. For more detailed information about the VMAX iSCSI architecture and configuration options, refer to the [HYPERMAX OS iSCSI Implementation](#).

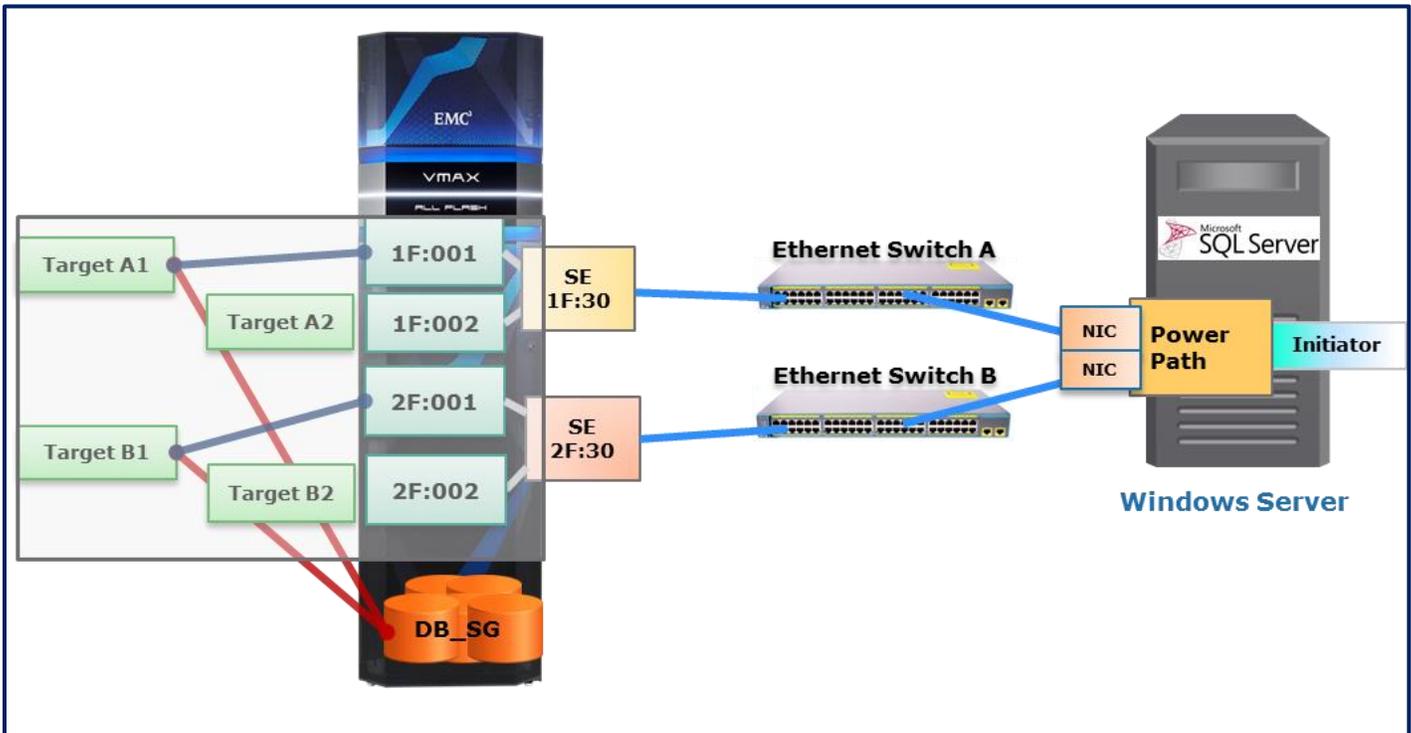


Figure 2. VMAX to Windows Host connectivity over iSCSI

DESIGN CONSIDERATIONS AND BEST PRACTICES

Network design and host connectivity considerations

Proper network design is the key to ensuring that iSCSI delivers enterprise level performance and reliability. The following are best practice considerations for iSCSI networks that use VMAX arrays:

- 10 GbE network interface cards on host are essential for enterprise production-level iSCSI storage connectivity. Anything less than 10 GbE should be used only for test and development.
- Segregate iSCSI traffic from general traffic by using either separate physical networks or layer-2 VLANs. A best practice is to have a dedicated LAN for iSCSI traffic and not share the network with other network traffic. Aside from minimizing network congestion, isolating iSCSI traffic on its own physical network or VLAN. This is a must for security as iSCSI traffic is transmitted in an unencrypted format across the LAN when using IPv4 supported for VMAX iSCSI.
- Use at least two network interface cards (NICs) for each database server to enable better availability and scale. Use multipathing software like EMC PowerPath® or Microsoft Windows MPIO for load balancing and automatic failover.
- Use the round robin load balancing policy for host based multipathing software. 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.
- Implement jumbo frames (by increasing the default network MTU from 1500 to 9000) to deliver additional throughput, especially for small block read and write traffic. However, be careful if jumbo frames are to be implemented, as they require all

devices on the network to be jumbo frame compliant and have jumbo frames enabled. When implementing jumbo frames, set host and storage MTUs to 9000 and set switches to higher values such as 9216 (where possible).

- iSCSI should be considered a local-area technology, not a wide-area technology, because of latency issues and security concerns. To minimize latency, avoid routing iSCSI traffic between hosts and storage arrays. Try to keep the number of hops to a minimum. Ideally, the host and storage should coexist on the same subnet and be one hop away.
- To minimize host CPU load due to iSCSI traffic, employ NICs with a built in TCP Offload Engine (TOE) and ensure that TOE is enabled. TOE NICs offload the processing of the datalink, network, and transport layers from the CPU and process it on the NIC itself.
- Consider the number of paths that will be available to the database storage devices. More paths improve I/O queues and the potential for more concurrency and performance.

VMAX iSCSI connectivity considerations

When planning storage connectivity for performance and availability use a “go-wide before going deep” policy, which means it is better to connect storage ports across different engines or directors than to use all the ports on a single director. In this way, even if a component fails, the storage can continue to service host I/Os. Connect at least two iSCSI ports to the Ethernet switch, preferably from different director boards.

VMAX iSCSI targets are assigned to a particular director board. We recommend evenly distributing targets among all available director boards for best performance as each director board has its own assigned CPU cores to service iSCSI I/O requests.

While creating a masking view, ensure that each storage group is accessible from the host by at least two targets/iSCSI virtual ports. Map each target to a separate physical port.

VMAX iSCSI multi-tenancy

VLANs and VMAX host initiator groups are used to support multi-tenancy on VMAX. VLANs provide isolated virtual networks so that only tenants (initiators) that are on the same VLAN as the target can access the target. VMAX host initiator groups are part of a VMAX device masking configuration, which allows fast and flexible changes to relationships between host initiators, storage target ports, and storage devices. Only the participating initiators can see the assigned devices. The VMAX iSCSI target is not tied to a particular port, and up to 64 targets can be mapped to a physical port. Each target can have up to 8 IP interfaces assigned to it, providing a high level of multi-tenancy with security. Figure 3 shows how multiple IP interfaces can share the same physical interface and can still be isolated using different VLANs and assigned to different targets.

Dir:Port	Network ID	IP Address	Prefix	VLAN ID	Dir:Virtual Port	Volumes
SE-1F:28	1	10.10.10.20	24	0	SE-1F:000	1
SE-1F:28	1	14.14.14.14	24	1	-	0
SE-1F:29	2	10.10.10.31	24	0	SE-1F:001	1
SE-1F:30	100	10.100.100.131	24	100	SE-1F:002	3
SE-1F:30	101	10.100.101.131	24	101	SE-1F:003	5
SE-1F:30	102	10.100.100.132	24	102	SE-1F:006	8
SE-2F:28	1	11.11.11.5	24	0	SE-2F:000	1
SE-2F:29	2	10.10.10.32	24	0	SE-2F:001	1
SE-2F:31	101	10.100.101.231	24	0	SE-2F:002	0

Figure 3. IP Interface configuration for VMAX iSCSI

Host I/O limits

The Host I/O limits is a quality of service (QOS) feature that provides the option to place specific IOPs or bandwidth limits on any storage group. Assigning a specific Host I/O limit for IOPS to a storage group with low performance requirements can ensure that a spike in I/O demand will not saturate or overload the storage and affect the performance of more critical applications. Using host I/O limits can ensure predictable performance for all hosts in a multi-tenant environment. The Host I/O limits is applicable at the storage group level, so it is available for both FC or iSCSI based storage access.

TEST ENVIRONMENT

Hardware and software configuration

The test environment consists of a Cisco C240M3 server with two dual-port 10 GbE network interface cards. The Microsoft SQL server acts as a single iSCSI initiator connected to two targets on a VMAX All Flash array. Both targets on VMAX provide access to the same set of volumes to host for redundancy. The database used an 8 KB-block size and a very small buffer pool to generate as many I/Os as possible to test the VMAX All Flash storage capabilities.

The VMAX 850FX storage system is has 1 V-Brick in a RAID5 configuration. The VMAX storage uses the HYPERMAX OS Q1 2016 release, which includes the FlashBoost feature. **Error! Reference source not found.** shows a list of the hardware and software components used.

Table 1. Hardware and software components

CATEGORY	TYPE	QUANTITY/SIZE	VERSION/RELEASE
Storage system	EMC VMAX 850F	1 x V-Brick	HYPERMAX OS 5977 based on Q1 2016 release
Database server	UCS C240-M3	1 – Standalone server	Microsoft Windows Server 2012
SQL Servers	Single SQL Server instances on stand-alone server	1 – SQL Server	Microsoft SQL Server 2014

Tested workloads

The purpose of the test was to demonstrate 8 Gb FC comparable performance with iSCSI for OLTP workload and use of iSCSI for repurposing production database snapshots for test and dev.

The test cases covered are:

- Comparison of a single Microsoft SQL Server database OLTP test performance over FC and iSCSI connectivity.
- Single Microsoft SQL Server with production DB connected over FC and test and development database connected over iSCSI

For the OLTP tests, we used a standard benchmarking tool to drive OLTP workloads. Database transaction rate and Windows Server Performance Monitoring using Perfmon and VMAX performance data collection was done while the workloads ran and results were analyzed and charted.

TEST CASES

Test 1: Single SQL Server database OLTP test using iSCSI and FC connectivity

Test motivation

This test shows a scenario with Microsoft SQL Server database OLTP workload running over Fibre Channel and then iSCSI interfaces to compare performance.

Test configuration

We ran a single SQL Server database OLTP workload running a C240-M3 server with an SQL buffer pool size of 4 GB. The same host was connected to a VMAX storage array using two 8 Gb FC as well as two 10GbE Ethernet interfaces. The host and VMAX arrays were connected to same Ethernet switch to minimize number of performance. The MTU was set to 9000 on both ends to maximize performance.

Test results

As the test results in Figure 4 show, SQL Batch requests per second in iSCSI interface is on par with FC. As the load is increased from 50 users to 150 users, iSCSI is able to sustain load and provide performance comparable to FC with sub millisecond latency.

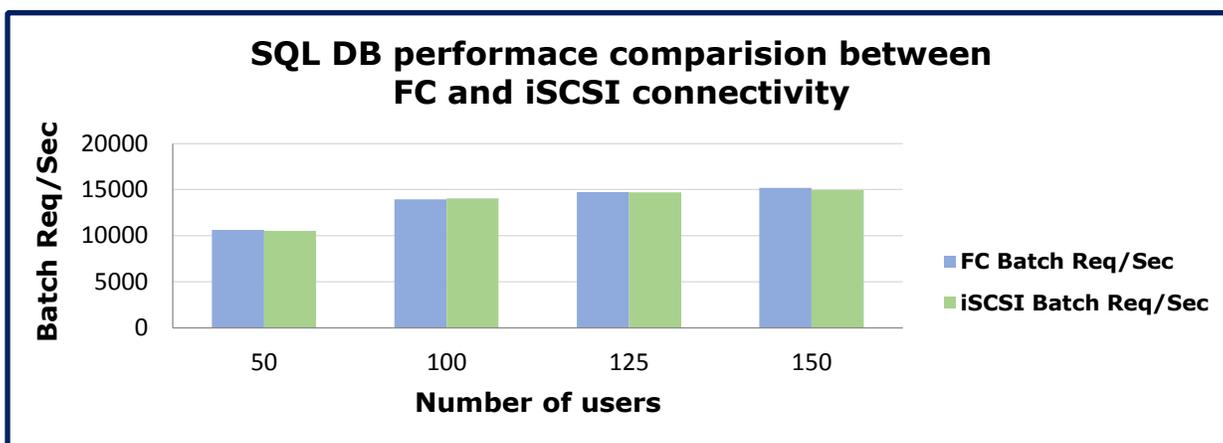


Figure 4. **Test 1 – Single SQL Server Database OLTP Test**

Test conclusion

The test showed similar, if not identical, performance for the database workload using either FC or iSCSI protocols.

Test 2: Repurposing SQL DB for Test and Dev using iSCSI connectivity

Test motivation

The goal of this use case was to show the value of using SnapVX to create a replica of a source SQL Server database that is connected via FC, and connect to the database replica using iSCSI. Because the SnapVX snapshots are linked to a target storage group with the specified storage, the access protocol target storage group can use either a FC or iSCSI protocol. This enables flexible storage access for target copies. The source databases continue to run transactions even while the SnapVX replicas are taken, simulating a real production-like situation. This test case also shows that a single VMAX array can service Microsoft SQL Servers over FC and iSCSI connections simultaneously.

Test configuration

We ran SQL Server database OLTP workload over FC based volumes with periodic snapshot of the database using SnapVX, as shown in Figure 5. One of the snapshots was presented to another host over iSCSI for test, development, and reporting type activity.

We followed these steps:

1. Ran a single Microsoft SQL Server database OLTP workload on a high performance server with FC connectivity to VMAX
2. Configured periodic snapshot of Storage Group associated with SQL Database
3. Linked one of the snapshots to a new Storage Group
4. Presented a linked Storage Group to another server over iSCSI

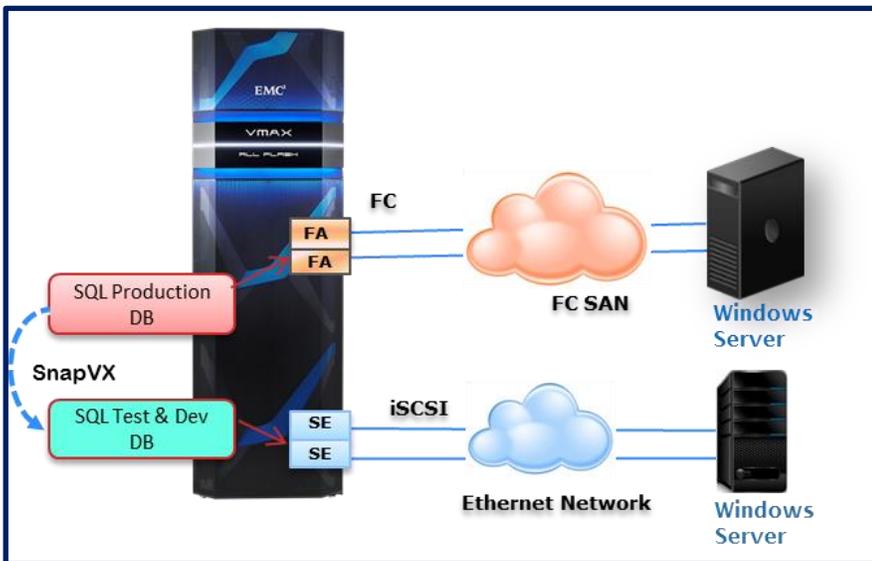


Figure 5. Repurposing a database using SnapVX and iSCSI

Test results

VMAX SnapVX snapshots for FC production storage groups can be linked to a storage group in a masking view containing iSCSI ports and offering cost effective SAN storage access for test/dev copies

Test conclusion

This test showed that VMAX iSCSI interfaces can be easily used to repurpose FC-based production database, which leverages iSCSI connectivity to the replicas.

CONCLUSION

VMAX with iSCSI is well suited to run high-performance Microsoft SQL Server workloads, while still leveraging VMAX features that include data protection, replications, resiliency, and availability. This gives provides the flexibility to deploy FC or iSCSI for connectivity to storage without sacrificing performace or reliability.

REFERENCES

- [EMC HYPERMAX OS VMAX iSCSI Implementation for VMAX3 and VMAX All Flash Arrays Technical Notes](#)
- [EMC VMAX All Flash Product Guide](#)
- [EMC VMAX All Flash VMAX 450/F/FX, 850F/FX Specifications](#)
- [Deployment Best Practice for Microsoft SQL Server with VMAX3 SLO Management Engineering White Paper](#)
- [Microsoft iSCSI Initiator Step-by-Step Guide](#)

APPENDIX

Configuring iSCSI using Unisphere for VMAX

This section shows the steps needed to configure VMAX iSCSI using the Unisphere® for VMAX graphical user interface.

Note: Prior to configuring iSCSI using Unisphere for VMAX, ensure that the network and VLAN on the host and switch are set up correctly.

To configure iSCSI using Unisphere:

1. Configure the iSCSI target.
2. Attach a new or existing IP interface to the iSCSI target.
3. Ensure network connectivity from the Windows host to the VMAX iSCSI IP interface.
4. Configure port group using the iSCSI target (virtual iSCSI ports).
5. Create a storage group using the devices accessed with iSCSI.
6. Create an initiator group using the host iSCSI IQN.
7. Create a masking view.
8. Discover and connect to the iSCSI target with the desired authentication on the Windows host.

Create an iSCSI target

For high availability and scale, create multiple iSCSI targets by using the best practices guidelines. The following example shows the creation of a single iSCSI target on iSCSI director emulation **1G**.

To create an iSCSI target, first select the option from the iSCSI Management menu, and then provide the required target information, as shown in Figure 6:

1. Click **System > iSCSI Dashboard > Create iSCSI target**.
2. From the Create iSCSI target dialog box, select the following, as shown in Figure 6:
 - The director that your target will be presented from
 - The target name (you can create a custom name or have the system automatically generate a name)
 - The network ID
 - The TCP port (default 3260 for iSCSI)

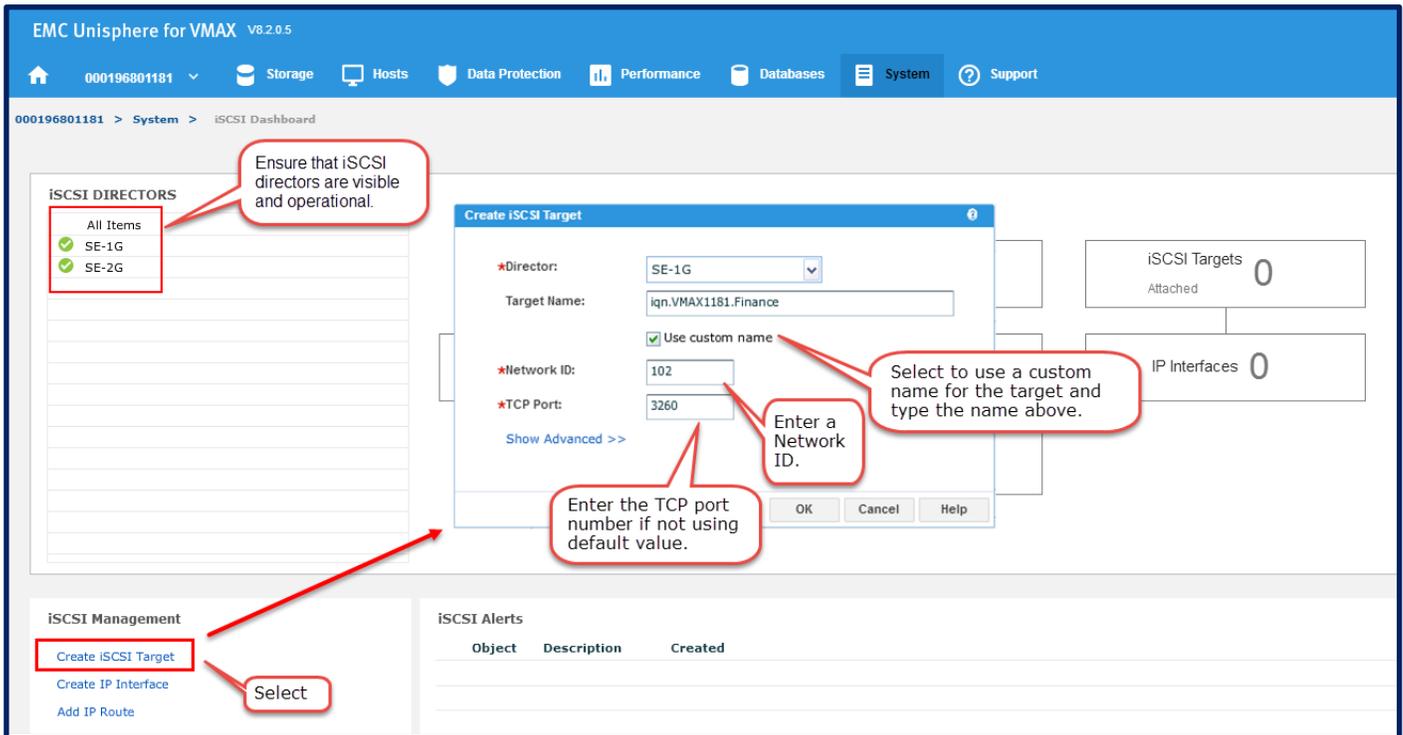


Figure 6. VMAX iSCSI target creation

Enable iSCSI target

By default the target is in disabled state after it is created. You must enable the target before it can be used.

To enable an iSCSI target, as shown in Figure 7:

1. Click **iSCSI Dashboard > Unattached iSCSI Targets**
2. Select the target that was created and click **Enable** from the menu at the bottom of the page.

The screenshot is divided into two main sections. The top section, titled "iSCSI Dashboard", shows a breadcrumb trail "000196801181 > System > iSCSI Dashboard". On the left, under "iSCSI DIRECTORS", there is a table with "All Items" and two entries: "SE-1G" and "SE-2G", both with green checkmarks. On the right, there are three summary boxes: "iSCSI Targets Unattached 1", "iSCSI Directors 2", and "iSCSI Ports 8". A red callout box points to the "iSCSI Targets Unattached 1" box with the text "Click iSCSI Targets Unattached".

The bottom section, titled "Unattached iSCSI Targets", shows a breadcrumb trail "000196801181 > System > iSCSI Dashboard > Unattached iSCSI Targets". It contains a table with the following data:

Name	Dir:Virtual Port	Status	Network ID	IP Interfaces	iSCSI Ports	Volumes
iqn.VMAX1181.Finance	SE-1G:000	Offline	102	0	0	0

Below the table, it says "1 Selected". At the bottom, there is a row of buttons: "Create", "Enable", "Disable", "Set Port Flags", "Attach", "Detach", "Delete", and "View Details". A red callout box points to the "Enable" button with the text "Select desired Target and click Enable".

Figure 7. **Enable iSCSI target**

Attach IP interface to an iSCSI target

To attach a target, as shown in Figure 8:

1. After you select the enabled target, click **Attach**.
2. From the **Attach IP Interface to iSCSI** target dialog box, select the director/port your target will be presented from, and then select an IP address, subnet prefix, VLAN ID, and Max Transaction Unit. The default MTU is 1500, but it can be set to 9000.

Note: Your network ID is pre-selected, based on the selected target ID.

The screenshot shows the 'Unattached iSCSI Targets' table with the following data:

Name	Dir:Virtual Port	Status	Network ID	IP Interfaces	iSCSI Ports	Volumes
iqn.VMAX1181.Finance	SE-1G:000	Online	102	0	0	0

Below the table, the 'Attach' button is highlighted with a red circle. A red callout box points to it with the text: 'Select a target and click Attach'.

The 'Attach IP Interface to iSCSI Target' dialog box is open, showing the following fields:

- Dir:Port: SE-1G:28
- IP Address: 10.100.100.132
- Prefix: 24
- VLAN ID: 102
- Network ID: 102
- Max Trans Unit: 9000

Red callouts highlight the 'Add IP Interface' button with the text: 'Click to add add more interfaces to the target' and the 'OK' button with the text: 'Click OK when finished'.

Figure 8. Attach IP interface to iSCSI target

Create Initiator Group

Use the host iSCSI initiator to configure the initiator group for iSCSI. Host initiators cannot log in to the target without a masking view, so the easiest method to create an initiator group is to manually type the host IQN, as shown in Figure 9. Microsoft supports a single IQN as part of an iSCSI stack for a Windows operating system, and all the NICs will use the same IQN.

To create an Initiator Group:

1. Click **Create Host**.
2. In the **Create Host** dialog box, type a host name and select **iSCSI** as the type of initiator.
3. Select the host (initiator) from the list or click **Add** to add it manually.
4. Select **Add to Job List** or **Run now**.

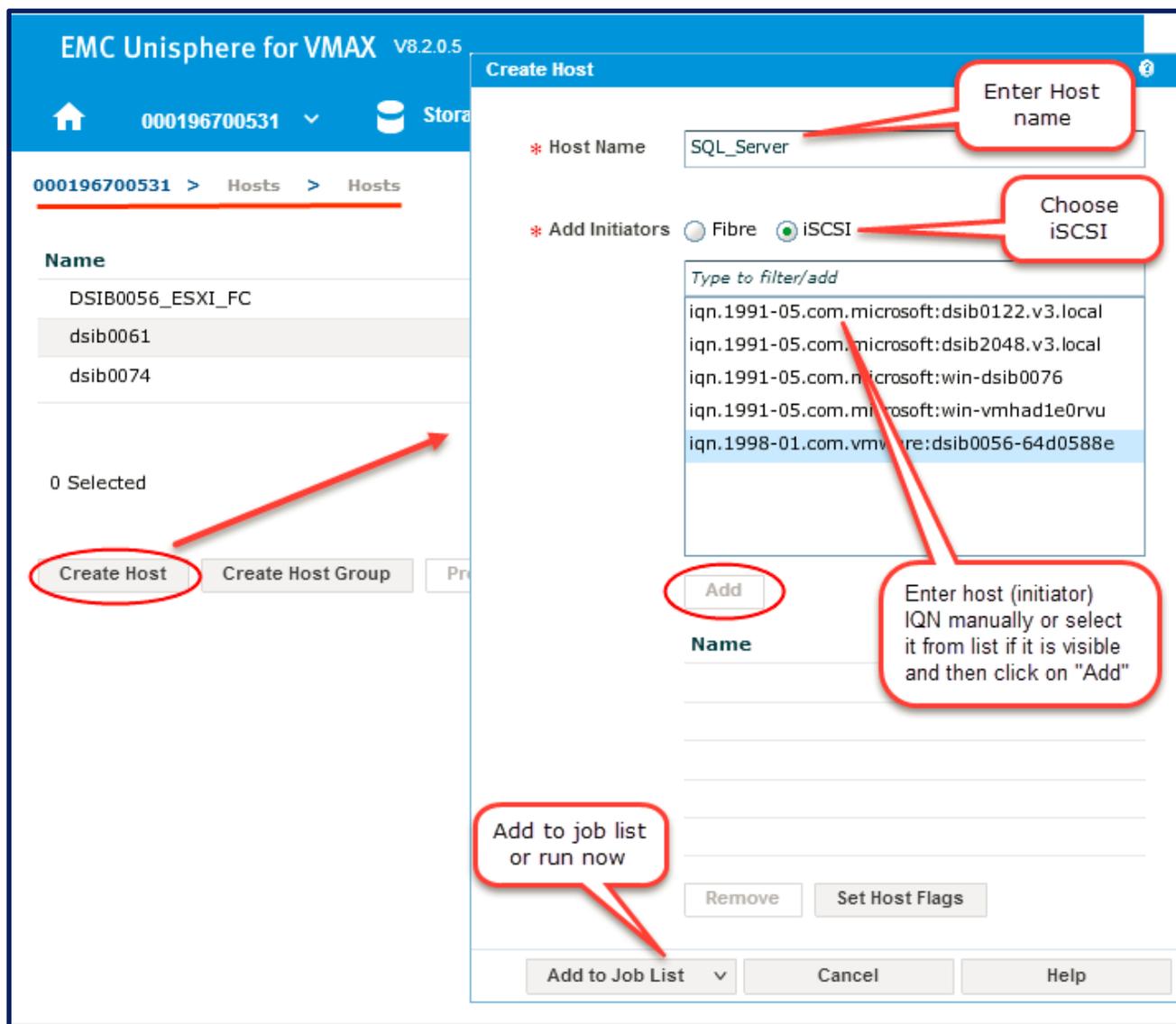


Figure 9. Host initiator group for iSCSI

Create a port group of iSCSI targets

1. Click **Hosts > Port Groups > Create Port Group**.

2. Enter a port group name, select **iSCSI**, select the newly created target from the list, and then click **OK**.

If a port group consists of iSCSI targets, you can use it to create a masking view to present a storage group of devices to a host with iSCSI interfaces.

Create masking view and discover target LUNs from Windows host

After creating a port group of iSCSI targets, you can create a masking view to complete the storage provisioning process. You can use Windows iSCSI properties to login to iSCSI targets and discover LUNs presented through the masking view.

Configuring VMAX iSCSI using SYMCLI

This section shows the steps needed to configure VMAX iSCSI using a Solutions Enabler command line interface (SYMCLI).

Create an iSCSI target

From the storage management host type the following command to create the iSCSI target:

```
# symconfigure -sid 531 -cmd "create iscsi_tgt dir 1f, iqn=iqn.1f:28.sql, network_id=1;" commit -nop
```

Enable iSCSI target

From the SYMCLI host type the following command to enable the iSCSI target:

Note: iSCSI targets are disabled by default. The **-iscsi_port** value is the target virtual port and not the physical port:

```
# symcfg -sid 531 -se 1f -iscsi_port 5 online
```

Create New IP Interface

From the SYMCLI host type the following command to create a new IP interface:

```
# symconfigure -sid 531 -cmd "create ip_interface dir 1f port 28, ip_address=14.14.14.14, ip_prefix=24, network_id=1, vlanid=1;" commit -nop
```

Attach New IP Interface to iSCSI target

From the SYMCLI host type the following command to attach an IP interface to a target:

```
# symconfigure -sid 531 -cmd "attach ip_interface ip_address=14.14.14.14 to iscsi_tgt iqn=iqn.1f:28.test1;" commit -nop
```

Configure one-way CHAP (optional)

Type the following command to enable one-way Challenge-Handshake Authentication Protocol (CHAP) to have a target authenticate an initiator. Create a CHAP entry for each initiator that connects to VMAX. In the following command, **IQN** is an initiator, and the the value for the **-cred** parameter is the same as initiator IQN.

```
# symaccess -sid 531 -iscsi iqn.1988-12.com.oracle:28525e1f5755 set chap -cred iqn.1988-12.com.oracle:28525e1f5755 -secret InitiatorCHAPSecret
```

Configure two-way CHAP (optional)

If initiators also need to authenticate a target, type the following command to enable two-way CHAP on a target:

Note: You must configure one-way CHAP first.

```
# symaccess -sid 531 -iscsi_dirport 1f:5 set chap -cred iqn.1f:28.test1 -secret targetCHAPSecret
```

Create Port Group with iSCSI target

From the SYMCLI host type the following command to create a port group with an iSCSI target:

```
# symaccess -sid 531 create -name test1_PG -type port -iqn iqn.1f:28.test1
```

Some useful SYMCLI VMAX iSCSI commands

This section lists some useful SYMCLI commands that are specific to VMAX iSCSI. Refer to the SYMCLI manual for more information and other commands.

List iSCSI target Details

Type the following command to display detailed iSCSI target information:

```
# symcfg -sid 531 list -se all -iscsi_tgt -detail
```

Disable iSCSI target

Type the following command to disable an iSCSI target:

Note: The **-iscsi_port** value is the target virtual port and not the physical port. List targets to find the virtual port associated with a target.

```
# symcfg -sid 531 -se lf -iscsi_port 5 offline
```

Rename iSCSI target

Type the following command to rename an iSCSI target:

```
# symconfigure -sid 531 -cmd "rename iscsi_tgt iqn=iqn.1f:28.test1 to new_iqn=iqn.1f:28.test1.RENAME;"  
commit -nop
```

Delete iSCSI target

Type the following command to delete an iSCSI target:

Note: you must remove the target from a port group before it can be deleted.

```
# symconfigure -sid 531 -cmd "delete iscsi_tgt iqn=iqn.1f:28.test1;" commit -nop
```

List iSCSI Ports

Type the following command to display detailed iSCSI port information:

```
# symcfg -sid 531 list -se all -port -detail
```

List IP interfaces

Type the following command to list all configured IP interfaces on the array:

```
# symcfg -sid 531 list -ip -se all
```

Modify an IP interface

Type the following command to modify the network ID and IP of an existing IP interface:

Note: You can modify only IP interfaces that are not attached to an iSCSI target.

```
# symconfigure -sid 531 -cmd "modify ip_interface dir lf, ip_address=14.14.14.14, network_id=1,  
new_network_id=2, new_ip_address=15.15.15.15, ip_prefix=24, mtu=9000;" commit -nop
```

Detach IP Interface from iSCSI target

Type the following command to detach an IP interface from an iSCSI target:

```
# symconfigure -sid 531 -cmd "detach ip_interface ip_address=14.14.14.14 from iscsi_tgt
iqn=iqn.1f:28.test1;" commit -nop
```

Delete IP Interface

Type the following command to delete an existing IP interface:

Note: You can delete only an interface that is not attached a target .

```
# symconfigure -sid 531 -cmd "delete ip_interface dir 1f, ip_address=14.14.14.14, network_id=1;" commit -nop
```

Add IP Route

Type the following command to add a static IP route on a given director emulation:

```
# symconfigure -sid 531 -cmd "add ip_route dir 1f, ip_address=10.10.10.0, ip_prefix=24, gateway=10.10.10.1,
network_id=1;" commit -nop
```

Remove IP Route

Type the following command to remove a static IP route on a given director emulation:

```
# symconfigure -sid 531 -cmd "remove ip_route dir 1f, ip_address=10.10.10.0, network_id=1;" commit -nop
```

List CHAP

Type the following command to list CHAP security records for targets on VMAX:

```
# symaccess -sid 531 list chap -v
```

Enable Uni-directional CHAP

Type the following command to enable one-way CHAP for an initiator:

Note: IQN is an initiator and the value of the **-cred** parameter is same as the initiator IQN.

```
# symaccess -sid 531 -iscsi iqn.1988-12.com.oracle:28525e1f5755 set chap -cred iqn.1988-
12.com.oracle:28525e1f5755 -secret targetCHAPSecret
```

Enable bi-directional CHAP

Type the following command to enable two-way CHAP on a target:

```
# symaccess -sid 531 -iscsi_dirport 1f:5 set chap -cred iqn.1f:28.test1 -secret InitiatorCHAPSecret
```

Disable Uni-directional CHAP

Type the following command to disable one-way CHAP on an initiator:

```
# symaccess -sid 531 -iscsi iqn.1988-12.com.oracle:28525e1f5755 disable chap
```

Disable Bi-directional CHAP

Type the following command to disable two-way CHAP on a target:

Note: The **-iscsi_dirport** value is the virtual port associated with the target.

```
# symaccess -sid 531 -iscsi_dirport 1f:5 delete chap
```

Microsoft Windows host initiator configuration using GUI

Configure and discover target portal

For Windows host initiator configurations, as shown in Figure 10, you must create a target portal with an IP address for the target, a TCP port number, and a CHAP target secret, if CHAP authentication is used.

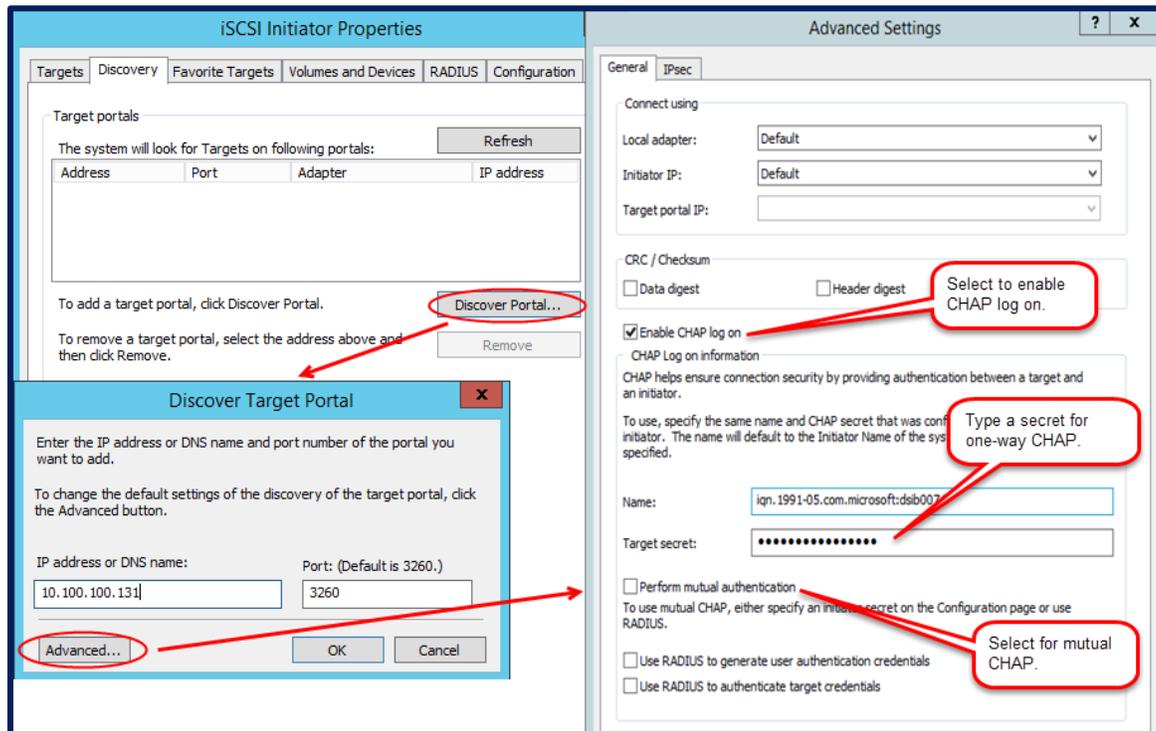


Figure 10. iSCSI target portal creation

Login to a target

After the target has been discovered by the host, select the discovered target, re-enter the CHAP secret if needed, and click **OK** to login to target, as shown in Figure 11.

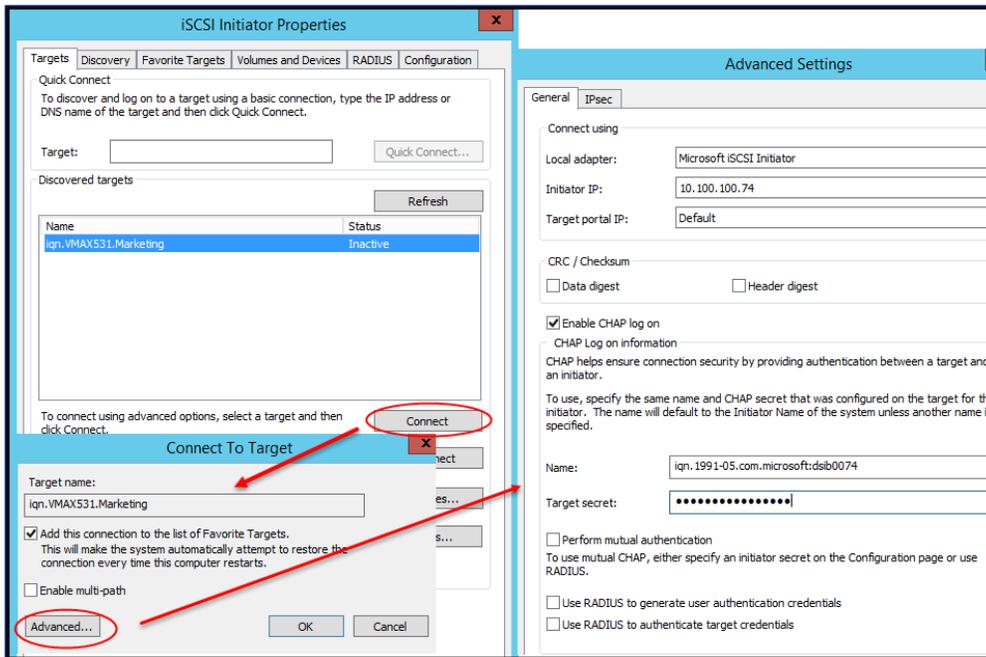


Figure 11. Logging into iSCSI target

Windows host iSCSI configuration using PowerShell cmdlet

You can use a PowerShell cmdlet to configure and manage an iSCSI initiator on a Windows host. This is especially useful for scripting of iSCSI configuration tasks.

Create iSCSI target Portal

Type the following command to create a target portal with one way CHAP authentication:

```
PS C:\Users\Administrator> New-IscsitargetPortal -targetPortalAddress 10.100.101.131 -InitiatorPortalAddress 10.100.101.76 -AuthenticationType ONEWAYCHAP -ChapSecret targetsecret1

InitiatorInstanceName : ROOT\iSCSI\PR\0000_0
InitiatorPortalAddress : 10.100.101.76
IsDataDigest          : False
IsHeaderDigest        : False
targetPortalAddress   : 10.100.101.131
targetPortalPortNumber : 3260
PSComputerName        :
```

Display discovered target from array

Type the following command to ensure that the target is visible to a host before attempting to login to it:

```
PS C:\Users\Administrator> Get-Iscsitarget
IsConnected NodeAddress PSComputerName
-----
False iqn.VMAX531.Sales
```

Login to discovered target

You can type the following example command to login to a target using one way CHAP. Ensure that the CHAP secret is configured for the host (initiator) before attempting to login. Refer to "Configuring VMAX iSCSI using SYMCLI " section for commands to configure CHAP on a VMAX array.

```
PS C:\Users\Administrator> Connect-Iscsitarget -NodeAddress iqn.VMAX531.Sales -InitiatorPortalAddress
10.100.101.76 -AuthenticationType ONEWAYCHAP -ChapSecret targetsecret1

AuthenticationType      : ONEWAYCHAP
InitiatorInstanceName   : ROOT\ISCSIPRT\0000_0
InitiatorNodeAddress    : iqn.1991-05.com.microsoft:win-dsib0076
InitiatorPortalAddress  : 10.100.101.76
InitiatorSideIdentifier : 400001370000
IsConnected             : True
IsDataDigest           : False
IsDiscovered            : True
IsHeaderDigest         : False
IsPersistent           : False
NumberOfConnections     : 1
SessionIdentifier       : fffffe00003b0c430-4000013700000001c
targetNodeAddress       : iqn.vmax531.sales
targetSideIdentifier    : 0400
PSComputerName         :
```