Hewlett Packard Enterprise

Technical white paper

Check if the document is available in the language of your choice.

HPE NIMBLE STORAGE AND HPE PROLIANT FOR MICROSOFT AZURE STACK HUB SOLUTION

Using HPE Nimble Storage arrays for scalable iSCSI storage with Microsoft Azure Stack Hub

CONTENTS

EXECUTIVE SUMMARY

Microsoft Azure Stack Hub is an extension of Microsoft Azure or Azure Public Cloud, which is located on-premises rather than in a cloud data center. Azure Stack Hub is deployed on hardware that is customer owned and operated. However, the Azure Stack Hub implementations differ from cloud data centers in both scope and scale due to the limited number of physical nodes in an Azure Stack Hub solution. Azure Stack Hub scales between 4 and 16 hybrid nodes and 4 and 8 all-flash nodes. Azure Stack Hub does not include all of the features or virtual machine (VM) types or services currently available in Azure Cloud Services, but a subset that are appropriate for hybrid, on-premises cloud deployments.

Azure Stack Hub mimics the most common features of Azure Public Cloud and gives the consumer an easy transition from Azure Public Cloud when data needs to exist on-premises. The primary method of managing an Azure Public Cloud instance (the Azure portal) is almost identical to the method of managing an Azure Stack instance (the Azure Stack portal).

This management paradigm, while similar to Azure Public Cloud, is foreign to classic on-premises computing management as it relates to managing common industry hypervisors (such as VMware vSphere® Hypervisor or Microsoft Hyper-V). A VM infrastructure owner commonly considers the effects of hosting very dissimilar VMs on the same host in relation to resource consumption. These options for heavily customized VMs are not available to Azure Stack Hub consumers, and while this lack of choice could be considered a detriment, this removes the need to manage and monitor these resources separately.

Microsoft Azure Stack Hub currently provisions storage utilizing internal disk from hyperconverged nodes managed by Storage Spaces Direct (S2D). External direct-attach storage is not supported under the Microsoft Azure Stack Hub design options. Therefore, the total capacity and performance available is capped by the maximum number of nodes in the scale unit, the disk drive configurations available from each OEM vendor, and the specific characteristics of the VM type deployed.

This differs from Microsoft Azure Stack HCI where the solution is managed using classic server tools such as Windows Admin Center (WAC) or Remote Server Administration Tools (RSAT). Microsoft Azure Stack HCI also uses Storage Spaces Direct (S2D), but allows for native attachment mechanisms for both additional direct-attach storage or Storage Area Networks. This paper does not cover Microsoft Azure Stack HCI-based solutions as the approach is more similar to classic Windows Server 2016 and Windows Server 2019 installations and these concepts are foreign to Microsoft Azure Stack Hub.

Since the release of the Microsoft Azure Stack Hub (originally called Azure Stack) solution, customers and partners have requested flexibility to leverage external storage arrays to support key workloads, along with ability to leverage key features such as migration, replication, and high availability.

To meet these requests, HPE ProLiant for Microsoft Azure Stack Hub product management initiated an internal project that included HPE ProLiant for Microsoft Azure Stack Hub engineering teams, Microsoft Azure Stack Hub engineering, and HPE Nimble Storage array engineering to investigate the viability of leveraging HPE Nimble Storage arrays as an external iSCSI storage option.

Target audience: This white paper is intended for IT administrators and architects, storage administrators, solution architects, and anyone who is considering the installation of the HPE ProLiant for Microsoft Azure Stack Hub and requires additional on-premises storage.

SOLUTION OVERVIEW

The goal of this white paper is to demonstrate the ability of HPE Nimble Storage arrays as an option for the HPE ProLiant for Microsoft Azure Stack Hub solution. The project consisted of the following validation test efforts:

1. Successfully establish IP communication from the Azure Stack ToR switches to the HPE Nimble Storage array via a border switch configuration, utilizing multiple 10 GbE connections for HA and high performance.
2. Successfully develop the processes and PowerShell scripts to enable a VM extension supported within VMs hosted as tenants—within the Azure Stack Hub environment—and provision volumes to the VM.
3. Successfully test and validate key performance metrics based on differing VM types, block sizes, and number of streams with the IOmeter performance utility.
4. Successfully test and validate key functionality, including:
   - Snapshots of volumes
   - Replication of volumes
   - Migration of volumes and workloads such as SQL 2008 and Windows 2008
   - Recovery of volumes
   - Cloud volume support

SOLUTION COMPONENTS

This section provides details of the components used in this solution.

HPE Nimble Storage All Flash Arrays

HPE Nimble Storage All Flash arrays combine a flash-efficient architecture with HPE InfoSight predictive analytics to achieve fast, reliable access to data and 99.9999% guaranteed availability.[1] Radically simple to deploy and use, the arrays are cloud-ready, providing data mobility to the cloud through HPE Cloud Volumes. The storage investment made today can be supported well into the future, thanks to our technology and business-model innovations. HPE Nimble Storage All Flash arrays include all-inclusive licensing, easy upgrades, and flexible payment options—while also being future-proofed for new technologies, such as NVMe and Storage Class Memory (SCM). For more details and specifications, see the HPE Nimble Storage All Flash Arrays Data Sheet.

FIGURE 1. HPE Nimble Storage All Flash Array

Key Features

- Predictive analytics
- Radical simplicity
- Fast and reliable
- Absolute resiliency

HPE ProLiant for Microsoft Azure Stack Hub

HPE ProLiant for Microsoft Azure Stack Hub is a hybrid cloud solution that transforms on-premises data center resources into flexible hybrid cloud services and provides a simplified development, management, and security experience that is consistent with Azure public cloud services. The hybrid cloud solution is co-engineered by Hewlett Packard Enterprise and Microsoft to enable the easy movement and deployment of apps to meet security, compliance, cost, and performance needs. For details, see HPE ProLiant for Microsoft Azure Stack Hub.

FIGURE 2. HPE ProLiant for Microsoft Azure Stack Hub

[1] HPE Get 6-Nines Guarantee: hpe.com/v2/GetDocument.aspx?docname=4aa5-2846enn

Microsoft Azure Stack Hub

Microsoft Azure Stack Hub is a hybrid cloud platform that enables users to use Azure services from their own company's or service provider's data center. For details, see Microsoft Azure Stack Hub User Documentation.

Azure Stack Hub mimics the most common features of Azure Public Cloud and provides the consumer an easy transition from Azure Public Cloud when data needs to exist on-premises. The primary method of managing an Azure Public Cloud instance (the Azure portal) is almost identical to the method of managing an Azure Stack Hub instance (the Azure Stack portal).

BEST PRACTICES AND CONFIGURATION GUIDANCE FOR THE SOLUTION

NOTE
These tests include exporting and importing data sets from non-Azure Stack Hub environments.

Validation step 1: iSCSI configuration between Azure Stack Hub and HPE Nimble Storage array

Deployment of networking for Azure Public Cloud is completely software defined, because physical changes are impractical at this scale. This same automation via software-defined networking (SDN) used in Azure Public Cloud is also prevalent in Azure Stack Hub. This level of automation requires very specific known hardware as well as a demarcation of the public network from the VMs and additionally a private back-end network just for Azure Stack Hub infrastructure usage.

When deploying S2D type storage, the private back-end network must be both redundant and sized to support the same amount of bandwidth as the network used to host the VMs. This is due to the need for S2D traffic that must exist in each node. When deploying SAN type storage, care should be given to assure that redundant paths exist to the storage.

To ensure that individual VMs cannot cause congestion on the shared physical network cards in the nodes, quality of service (QoS) settings must be used. These are enforced on the various VM types defined by both Azure Public Cloud and Azure Stack Hub to ensure safe resource allocation, and the QoS settings are not changeable by the VM owner. As expected, a VM type that consumes more vCPU, memory, and storage also gets larger network resource allocation settings. When deploying on general hypervisors, you can set these values manually or change the quantity and speed of your network adapters.

Leveraging the HPE Azure Stack Innovation Centers in Bellevue, Washington, validation tests were performed on a 4-node HPE ProLiant for Microsoft Azure Stack Hub environment. The Azure Stack Hub environment was connected to a dual HPE FlexFabric switch via dual 10 GbE connections and dual 10 GbE connections to the two array controllers of an HPE Nimble Storage AF1000 (all-flash) array configured with 24 populated SSD drives.

The HPE Nimble Storage array was connected to three different subnets: a Management Subnet where the HPE Nimble Storage array management ports were connected and two Data Subnets, which each hosted the 10 GbE data network ports from the HPE Nimble Storage array. In FIGURE 3, note that only the data ports are exposed and connected to the border switches.

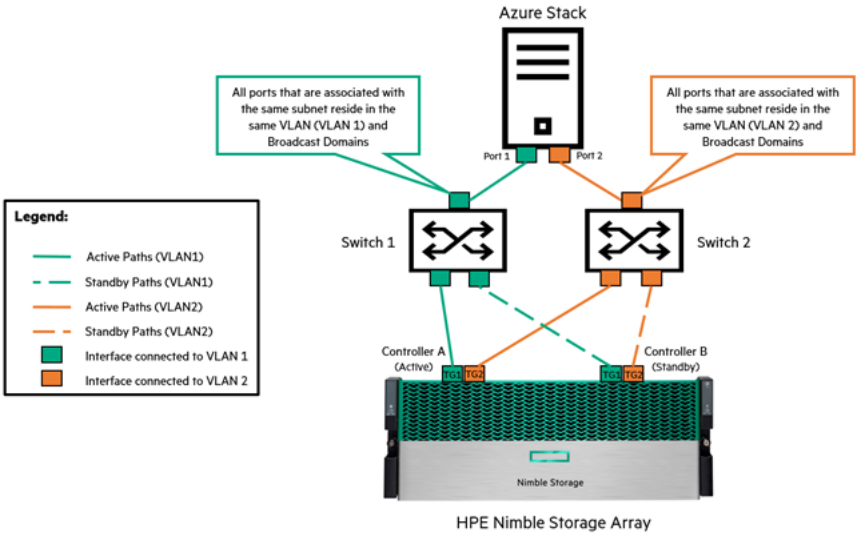

FIGURE 3. Network connectivity

In testing, the best practices for HPE Nimble Storage have been followed as outlined in HPE Nimble Storage Deployment Considerations for Networking. When reviewing this document follow the design principle of multiple subnets and multiple switches.

Validation step 2: Enabling Azure Stack Hub VMs to auto-connect to HPE Nimble Storage arrays

Azure Stack Hub Storage operations are software defined and controlled via the Azure Stack portal. No method exists to control HPE Nimble Storage (or any third-party block storage) from within the Azure Stack portal, because the portal is not extensible at this time. HPE Nimble Storage engineering has written an Azure Stack Hub Custom VM Extension to enable connectivity of an Azure Stack Tenant VM via an iSCSI connection to an HPE Nimble Storage array. The VM Extension will accomplish the following goals on a new Tenant VM:

1. Start and automatically configure the iSCSI service
2. Install and load:
   a. Microsoft Windows MPIO
   b. Microsoft Azure Stack PowerShell Modules
   c. HPE Nimble Storage Windows Toolkit
   d. HPE Nimble Storage PowerShell Toolkit
   e. Additional (new) HPE Nimble Storage Azure Stack commands

Each of these steps is detailed in the following:

- iSCSI service – The act of starting and configuring the iSCSI service is straightforward; however, this step needs to be accomplished to automatically create the initiator groups on the HPE Nimble Storage array so that volumes can be easily assigned to the server.
- MPIO – The Windows MPIO feature must be installed, and once installed, the server must be rebooted. This necessitated the ability for the VM Extension to embed itself to automatically run at the next reboot to continue the process.
- Toolkits – The HPE Nimble Storage Windows Toolkit and the HPE Nimble Storage PowerShell Toolkit are used on all installations where Windows Server are connected to HPE Nimble Storage. The benefit is that they are automatically and silently installed and configured.
- CLI commands – Additionally, the address and credentials to assign storage to the host from the HPE Nimble Storage are stored in the registry of the server. A modified set of commands has been created to make the creation of new volumes and connectivity of the array automatic.

After the VM Extension has been loaded on the VM during the creation process, to assign a new volume to an Azure VM, open a PowerShell window and issue the following command:

```
PS:> Connect-AZNSVolume -name “Friendly_Name” -size 102400
```

The HPE Nimble Storage Custom VM Extension for Azure Stack Hub is written in common PowerShell and is intended to either be hosted on your own GitHub account or an internal file server that your Azure Stack Hub VMs can communicate with.

FIGURE 4 shows the workflow of the Azure Stack Hub VM Extension.

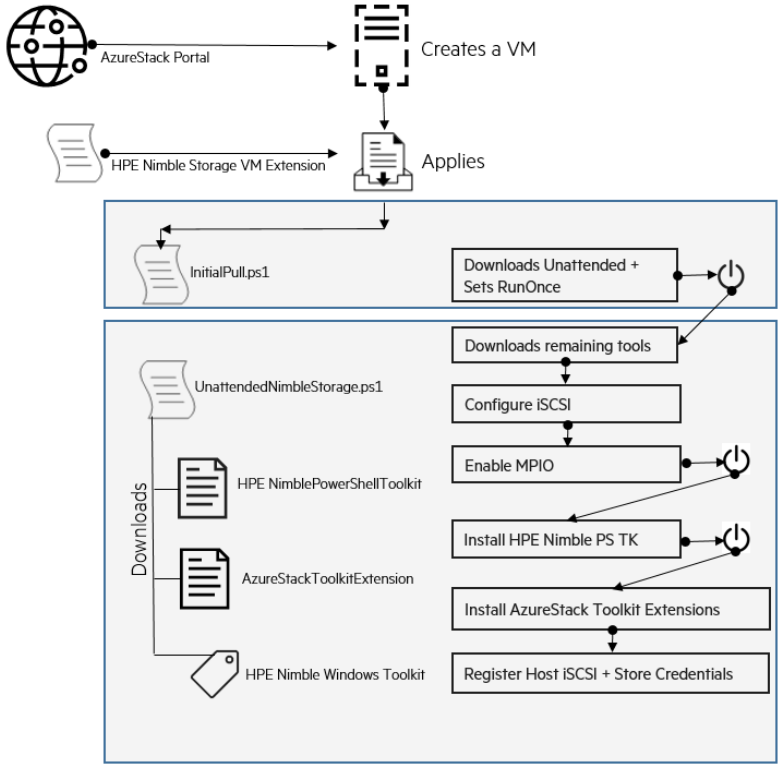

FIGURE 4. VM extension

`HPENimbleStoragePowerShell.ps1` is the initial Azure Stack Hub custom VM extension file that will be loaded by an Azure Stack deployment. This file will create a folder on the target machine named `C:\NimbleStorage`. It will create the first file named `InitialPull.ps1` to run, which will connect to the GitHub account and download the actual unattended script.

NOTE

The unattended script cannot be used to connect to GitHub directly because the Azure Stack VM extension might run before networking exists. Additionally, the registry of the VM is modified to allow the `InitialPull.ps1` file to run after the first reboot. The Azure Stack VM Extension then directs the VM to reboot itself.

The VM Extension will create the `InitialPull.ps1` file that has a single job, which is to allow the unattended script to be downloaded from the GitHub site regardless of the invalid certificate (because software-defined networking can invalidate certificates). Additionally, the initial pull script will ensure that the unattended installation script will run via the registry `RunOnce` facility before it issues the reboot order.

The unattended installation script will accomplish all tasks related to preparing the VM for HPE Nimble Storage. These include downloading all of the required software packages and installing them. The `HPENimblePowerShellToolkit.300.zip` is the HPE Nimble Storage PowerShell Toolkit, downloaded directly from the HPE InfoSight software download page. To access the download page, do the following:

1. Open a browser to: https://infosight.hpe.com/app/login
2. Log in using your user name and password. (Create an account if you don't already have one.)
3. After login, click **Resources** from the menu bar.
4. Under **Resources**, select **Software Downloads**.
5. In the main window pane under **What's New?**, select the HPE Nimble Storage PowerShell Toolkit for download. Also download `Setup-NimbleNWT-x64.7.0.2.56.exe`, which is the standard HPE Nimble Storage Windows Toolkit that contains the HPE Nimble Storage DSM as well as additional tools. It is also located in the main window pane above the HPE Nimble Storage PowerShell Toolkit.

The last major installation step done by the unattended installation script is to install an additional set of Azure Stack Hub custom commands to the HPE Nimble Storage PowerShell Toolkit that automate connecting to the array and some basic iSCSI tasks. For more details, execute
`Get-Commands -module HPENimblePowerShell -name *-azns*`.

The help option for `Connect-AZNSVolume` offers the following help:

```
SYNOPSIS

    The Command will create and attach a new Nimble Volume to the current host.

SYNTAX

    Connect-AZNSVolume [-name] <String> [-size] <Long> [[-description] <String>] [<CommonParameters>]

DESCRIPTION

    The command will retrieve the credentials and IP Address from the Registry for the Nimble Storage Array, and
    connect to that array. The command with then creates a volume on the array to match the passed parameters,
    and assign access to that volume to the initiator group name that matches the current hostname. Once the
    mapping has occurred, the command will continue to detect newly detected iSCSI volumes until a volume appears
    that matches the Target ID of the Volume created. Once the iSCSI volume has been detected, it will be
    connected to persistently, and then refresh the Microsoft VDS (Virtual Disk Service) until that device
    because available as a WinDisk. The New Windisk that matches the serial number of the Nimble Storage Volume
    will then be initialized, placed online, a partition created, and then finally formatted. The return object
    of this command is the Windows Volume that has been created.

    Additional parameters and more granular control are available when using the non-Azure Stack versions of the
    commands, i.e. You can set more features using the New-NSVolume Command however, the steps required to
    automate the attachment or discovery of these volumes is not as automated.

PARAMETERS

    -name <String>

        This mandatory parameter is the name that will be used by both the Nimble Array to define the volume
        name, but also as the name to use for the Windows Formatted partition.

        Required?                    True

    -size <Long>

        This mandatory parameter is the size in MegaBytes (MB) of the volume to be created. i.e. to create a 100
        GigaByte (GB) volume, select 10240 as the size value.

        Required?                    True

    -description <String>

        This commonly a single sentence to descript the contents of this volume. This is stored on the array and
        can help a storage administrator determine the usage of a specific volume. If no value is set, and auto-
        generated value with be used.

        Required?                    False

NOTES

    This module command assumes that you have installed it via the unattended installation script for connecting
    Azure Stack to a Nimble Storage Infrastructure. All functions use the Verb-Nouns construct, but the Noun is
    always preceded by AZNS which stands for Azure Stack Nimble Storage. This prevents collisions in customer
    environments.

-------------------------- EXAMPLE 1 --------------------------

    PS C:\Users\TestUser> Connect-AZNSVolume -size 10240 -name Test10

    Successfully connected to array 10.1.240.20

    DriveLetter FileSystemLabel FileSystem DriveType HealthStatus OperationalStatus SizeRemaining Size

    ----------- --------------- ---------- --------- ------------ ----------------- ------------- ----

    R           Test10          NTFS       Fixed     Healthy      OK                       9.93 GB 9.97 GB
```

Additionally, a way of connecting to the user with the saved HPE Nimble Windows Toolkit credentials was also needed, because most users do not want to input credentials each time they are required to access or modify storage. To accomplish this, an option was added to `Connect-NsGroup`, in which the command can be directed to use the saved credentials. Additionally, the command allows for ignoring invalid certifications, because SDN can prevent certification validation.

```
SYNOPSIS

    Connects to a Nimble Storage group.

DESCRIPTION

    Connect-NSGroup is an advanced function that provides the initial connection to a Nimble Storage array so
    that other subsequent commands can be run without having to authenticate individually.It is recommended to
    ignore the server certificate validation (-IgnoreServerCertificate param) since Nimble uses an untrusted SSL
    certificate.

PARAMETER Group

    The DNS name or IP address of the Nimble group.

PARAMETER Credential

    Specifies a user account that has permission to perform this action. Type a user name, such as User01 or
    enter a PSCredential object, such as one generated by the Get-Credential cmdlet. If you type a user name,
    this function prompts you for a password.

PARAMETER IgnoreServerCertificate

    Ignore the server SSL certificate.

PARAMETER UserNWTUserCredentials

    This option will retrieve and use the existing Nimble Windows Toolkit saved credentials for the known array.
    If this option is used neither the group or credential object need be specified.

EXAMPLE

    Connect-NSGroup -Group nimblegroup.yourdns.local -Credential admin -IgnoreServerCertificate

    *Note: IgnoreServerCertificate parameter is not available with PowerShell Core

EXAMPLE

    Connect-NSGroup -Group 192.168.1.50 -Credential admin -IgnoreServerCertificate

    *Note: IgnoreServerCertificate parameter is not available with PowerShell Core

EXAMPLE

    Connect-NSGroup -Group nimblegroup.yourdns.local -Credential admin -ImportServerCertificate

EXAMPLE

    Connect-NSGroup -Group 192.168.1.50 -Credential admin -ImportServerCertificate
```

Validation step 3: Performance metrics

The network limitations on storage that are inherent with software-defined storage native to Azure Stack (based on S2D) are expressed and exposed to the user in an effort to prevent Azure Stack's private network from becoming overloaded. These limitations are due to S2D using three-way mirroring and each write operation must be stored locally and on two other nodes of the Azure Stack cluster. The act of replication to neighboring physical servers causes a natural write amplification effect that is not present on iSCSI (SAN)-based storage.

Because the iSCSI traffic exists on the client (from Azure Stack's perspective) network, it is not subject to the IOPS and MB/s limitations. This allows you to connect a high-performance iSCSI-based volume to any Azure Stack VM—regardless of the size of that VM. The only concern for iSCSI-based storage is that the Azure Stack network load balancers are scaled to a level to support high IOPS and MB/s loads, which was evident in testing:

TESTING AND TEST RESULTS

Testing methodology

The tests performed need to be repeatable, portable, and provide insight. To meet these needs, it was important to use a benchmark tool that is available via public domain. It is also valuable for the test platform to be self-contained and not require a controlling station or other external assets to run tests. Additionally, the output from these tests should be saved to a comma-delimited file, which can be manipulated (via Excel) to produce metrics. To provide valuable insight, it is required to output **all** of the test results instead of just a single data point, which would provide an insufficient view. To accomplish these goals, the HPE Nimble Storage engineering team selected the Intel® IOmeter benchmark, which can be downloaded at the following location: iometer.org.

The HPE Nimble Storage engineering team utilized IOmeter to run the following tests:

TABLE 1. IOmeter test results

Test	Request size	Read type	Result
Test 1	4 KB	100% sequential	0% read
Test 2	4 KB	100% sequential	25% read
Test 3	4 KB	100% sequential	50% read
Test 4	4 KB	100% sequential	75% read
Test 5	4 KB	100% sequential	100% read
Test 6–10	4 KB	75% sequential	0%–100% read
Test 11–25	4 KB	50%–0% sequential	0%–100% read
Test 26–50	16 KB	100%–0% sequential	0%–100% read
Test 51–75	64 KB	100%–0% sequential	0%–100% read

Similar tests were performed based on 16 KB and 64 KB request sizes, for a total test set of 75.

iSCSI SAN testing results

The major concerns when testing iSCSI storage connected to Azure Stack Hub are if it can support a sufficiently high IOPS or MB/s set of workloads, all while maintaining an acceptably low latency to disk.

For the following tests, the workload thread count was increased from 1 to 128 threads—doubling each time. Additionally, the read/write ratio was altered from 100% read to 100% write in 25% steps, and the sequential/random nature of the data set was modified from 100% sequential to 100% random in 25% steps to offer a comprehensive view of the storage performance. The results of these 175 tests (7 thread levels x 5 sequential/random steps x 5 read/write steps) are shown in the following 3D graphs (surface mapped) and were repeated using both 4 KB block size data (FIGURE 5) and 64 KB block size data (FIGURE 6).

These tests were also repeated using collections of smaller VMs (such as A4) to larger VMs (such as Fs16v2) to ensure that performance was comparable between VMs.

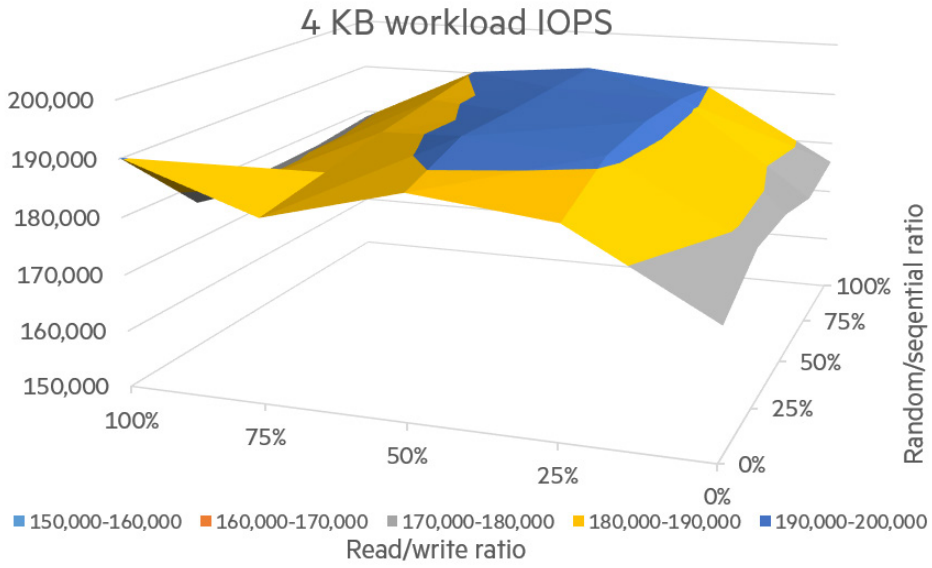

FIGURE 5. 4 KB workload IOPS result

This graph shows that a high level of IOPS can be achieved through the Azure Stack Hub Load Balancer regardless of the read/write ratio or the sequential/random nature of the workload. In this case, the variance between the average achieved performance (184K IOPS) was +/– only 8K IOPS (4.2% variance) from the maximum performance (192K IOPS) and only 12K IOPS (6.8% variance) from the minimum performance.

TABLE 2 illustrates the expected variance of IOPS between differing tests and the consistent nature of the performance.

TABLE 2. 4 KB IOPS results across all tests

Roll-up Stats	IOPS	Median Variance	Average Variance
Average	184,741 IOPS		
Median	186,713 IOPS		(+/– 1% of the average)
Minimum	172,143 IOPS	(+7.8% below median)	(+6.8% below average)
Maximum	192,569 IOPS	(+3.1% above median)	(+4.2% above average)

The results indicate that the testing was very consistent across all variables. FIGURE 5 has been exaggerated to emphasize the relevant data points; however, if the graph IOPS axis started at **zero,** the graph would appear completely flat.

FIGURE 6 shows the 64 KB test results. Testing the 64 KB access, the relevant data is commonly measured in MB/s.

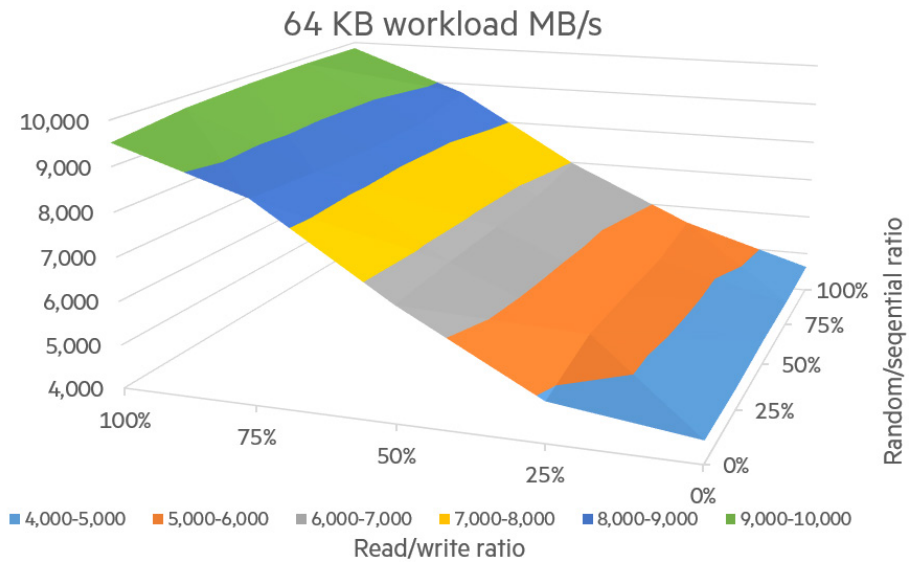

FIGURE 6. 64 KB workload MB/s result

This graph shows MB/s achieved through the Azure Stack Hub Load Balancer regardless of the read/write ratio or the sequential/random nature of the workload. This workload, while not specifically suited for flash, can still be suitable to populate a 10 Gb/s network path with 80% traffic.

In this case, the variance between the average achieved performances (6973 MB/s) showed significant advantage towards write operations, as shown in TABLE 3. Because the write operation does not require a response, it can be immediately acknowledged via protected cache. Read operations must come out of SSD because the test was specifically designed to use significantly more space than the cache memory allowed for read cache hits.

TABLE 3. Average 64 KB MB/s results across all tests

Roll-up Stats	MB/s	Median Variance	Average Variance
Average	6973 MB/s		
Median	6705 MB/s		(+/– 3.9% of the average)
Minimum	4493 MB/s	(+ 35% below median)	(+ 32% below average)
Maximum	9850 MB/s	(+ 47% above median)	(+ 41% above average)

The results indicate that the testing was very consistent across all read/write ratios as expected, but it showed a greater benefit toward intensive write operations when large block was encountered. FIGURE 6 has been exaggerated to emphasize the relevant data points; however, if the graph MB/s axis started at zero, the graph would appear less severe, yet still pronounced.

The latency testing shows that a very low latency was achieved for almost all workload levels, and that very few if any outliers would prevent a very fast application from operating well.

FIGURE 7 shows the results from the 4 KB 50% read and 50% random latency tests.

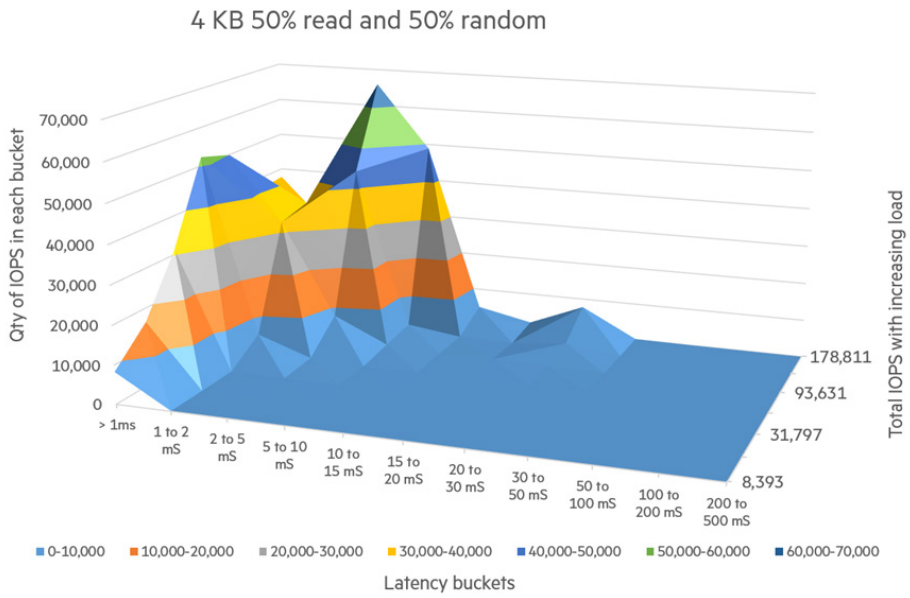

FIGURE 7. 4 KB 50% read and 50% random latency result

The graph shows the latency expectations from an entry-level all-flash array when connected to Azure Stack Hub. In this case, a number of Azure Stack Hub VMs are all executing the workload against the storage. The test is designed to double the load at predetermined times, so the first test run is able to produce 8000 IOPS and the second run generates 16,000 IOPS, as shown in TABLE 4. The final test pass 7 (64 workers) generates most of its workload of 180,000 IOPS around 2 ms to 5 ms. Note that this test is not a test of HPE Nimble Storage but instead used to determine if the SDN stack inside of Azure Stack Hub is capable of keeping a low latency under load, which it appears to be able to do.

TABLE 4. 4 KB 50% read and 50% random latency result

Pass number	Power	Workers
1	2^0	1
2	2^1	2
3	2^2	4
4	2^3	8
5	2^4	16
6	2^5	32
7	2^6	64

The takeaway from this table is that an HPE Nimble Storage array can produce both high IOPS and low latency directly to any Azure Stack Hub VM. Microsoft has implemented load balancers that do not interfere with good networking results.

Volume and workload migration

Microsoft Azure Stack Hub brings the power of the Azure Public Cloud to enterprise data centers, delivering cloud-scale economics and speed. The benefits of Azure Stack Hub to enterprises and managed service providers can be enhanced by adding enterprise data protection capability to the Azure Stack Hub environment with features such as replication.

When using Azure Stack Hub Storage, data cannot be prepopulated and must be copied into the Azure Stack Hub as a file system (SMB) type copy. You cannot stand up a new file server or SQL Server with a prepopulated database unless you first take the time to replicate the database into the Azure Stack Hub. This limitation is not encountered with HPE Nimble iSCSI storage, as any existing data set that exists on a storage array can be instantaneously cloned and imported directly in seconds to any Azure Stack Hub VM.

Additionally, all data volumes for HPE Nimble Storage can take advantage of hardware-based block-level snapshots and can be replicated from site to site using HPE Nimble Storage replication technology.

A third valuable feature of using HPE Nimble Storage is the ability to host a Windows Failover Cluster inside the Azure Stack Hub using the HPE Nimble Storage iSCSI target to host either clustered volumes or cluster shared volumes. However, note that due to the subnet limitations, a Windows Failover Cluster cannot have nodes that exist outside of the Azure Stack Hub environment.

Validation test 4: Key feature validation
Tests performed were:

- Replication of volume: Each volume can be separately, or as a collection, replicated from one HPE Nimble Storage array to an additional HPE Nimble Storage array in either an async or sync mode. These replica relationships are configured according to published best practices in HPE Nimble Storage Deployment Considerations for Networking .
- Snapshot of volume: Up to 1000 snapshots can exist for each volume that is exposed to an Azure Stack Hub VM. The ability to snapshot a volume and the ability to run standard automated snapshot schedules were tested. These snapshots and schedules can be configured, monitored, triggered, and managed from within the Azure Stack Hub VM, or from any dedicated management station.
- Recovery of volume: The ability for an Azure Stack Hub to act as the recovery point from a volume that was generated outside of the Azure Stack Hub infrastructure as well as the ability for an Azure Stack Hub VM to recover a corrupted data set by reverting to a previously created snapshot was tested. And they worked exactly as expected (identical to non-Azure Stack Hub machines).

In all of these cases, the best practices outlined in HPE Nimble Storage Deployment Considerations for Networking were used and no deviations were encountered.

SUMMARY

HPE Nimble Storage and HPE ProLiant for Microsoft Azure Stack Hub engineers were able to successfully perform and validate all configuration and testing steps to achieve the goals detailed in this white paper. Testing methodology and results were shared and reviewed by Microsoft Azure Stack Hub engineering, enabling Hewlett Packard Enterprise and Microsoft to jointly approve HPE Nimble Storage arrays as an iSCSI option for the HPE ProLiant for Microsoft Azure Stack Hub solution.

APPENDIX

Section A1: HPENimbleStoragePowerShell.ps1

This single file is the AzureStack VM Extension custom script. This file will create a folder on the target machine named `C:\NimbleStorage`. It will create the first file named `InitialPull.ps1` to run, which will connect to the GitHub account and download the actual unattended script. Note: The `NimbleStorageUnattended.ps1` script cannot be used to connect to GitHub directly since the Azure Stack VM Extension might run before networking exists. Additionally, the registry of the VM is modified to allow the `InitialPull.ps1` file to run after the first reboot. The Azure Stack VM Extension then directs the VM to reboot itself. This file will contain customer variables such as username, password, and the IP Address of the HPE Nimble Array to be managed. This file should obviously not be placed on a publicly viewable unlocked GitHub site and should only need to be used by those with Azure Portal credentials.

```
$NimbleUserName = "admin"
$NimblePassword = "admin"
$NimbleArrayIP = "10.1.240.20"
# Do not need to modify anything below this line.
# This single file should NOT be posted to YOUR public Github site as it contains the username and password of the Nimble array
# that you intend to use.
#########################################################################################################################################

$InitialPull=@'
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
if (-not ([System.Management.Automation.PSTypeName]'ServerCertificateValidationCallback').Type)
{   $certCallback = @"
    using System;
    using System.Net;
    using System.Net.Security;
    using System.Security.Cryptography.X509Certificates;
    public class ServerCertificateValidationCallback
    {   public static void Ignore()
        {   if(ServicePointManager.ServerCertificateValidationCallback ==null)
            {   ServicePointManager.ServerCertificateValidationCallback +=
                delegate    (   Object obj,
                                X509Certificate certificate,
                                X509Chain chain,
                                SslPolicyErrors errors
                            )
                {   return true;
                };
        }
    }
}
"@
Add-Type $certCallback
}
[ServerCertificateValidationCallback]::Ignore()
$uri="https://raw.githubusercontent.com/HewlettPackard/HPEAzureStackOnHPENimbleStoragek/master/NimbleStorageUnattended.ps1"
$Code=(Invoke-WebRequest -Uri $uri -Method Get).content
out-file -FilePath "C:\NimbleStorage\NimbleStorageUnAttended.ps1" -inputobject $Code -ErrorAction SilentlyContinue -force

$RunOnce="HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
set-itemproperty $RunOnce "NextRun" ('C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe -executionPolicy Unrestricted -File ' +
'C:\NimbleStorage\NimbleStorageUnattended.ps1')
write-host "Hit CTRL-C in next 60 seconds to abort the AutoReboot cycle"
start-sleep -Seconds 60
shutdown -t 0 -r -f
'@

mkdir C:\NimbleStorage -ErrorAction SilentlyContinue

New-Item -Path "HKLM:\Software\AzureStackNimbleStorage"
New-ItemProperty -Path "HKLM:\Software\AzureStackNimbleStorage" -PropertyType String -Name NimbleUserName -Value $NimbleUserName
New-ItemProperty -Path "HKLM:\Software\AzureStackNimbleStorage" -PropertyType String -Name NimblePassword -Value $NimblePassword
New-ItemProperty -Path "HKLM:\Software\AzureStackNimbleStorage" -PropertyType String -Name NimbleArrayIP -Value $NimbleArrayIP

out-file -filepath C:\NimbleStorage\InitialPull.ps1 -inputobject $InitialPull -force
$RunOnce="HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
set-itemproperty $RunOnce "NextRun" ('C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe -executionPolicy Unrestricted -File ' +
'C:\NimbleStorage\InitialPull.ps1')
```

Section A2: InitialPull.ps1

This file is created by the Azure Stack, and it has a single job, which is to allow the unattended script to be downloaded from the GitHub site regardless of the invalid certificate (because SDN can invalidate certificates).

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
if (-not ([System.Management.Automation.PSTypeName]'ServerCertificateValidationCallback').Type)
{   $certCallback = @"
    using System;
    using System.Net;
    using System.Net.Security;
    using System.Security.Cryptography.X509Certificates;
    public class ServerCertificateValidationCallback
    {   public static void Ignore()
        {   if(ServicePointManager.ServerCertificateValidationCallback ==null)
            {   ServicePointManager.ServerCertificateValidationCallback +=
                delegate    (   Object obj,
                                X509Certificate certificate,
                                X509Chain chain,
                                SslPolicyErrors errors
                            )
                {   return true;
                };
        }
    }
}
"@
Add-Type $certCallback
}
[ServerCertificateValidationCallback]::Ignore()
$uri="https://raw.githubusercontent.com/HewlettPackard/HPEAzureStackOnHPENimbleStorage/master/NimbleStorageUnattended.ps1"
$Code=(Invoke-WebRequest -Uri $uri -Method Get).content
out-file -FilePath "C:\NimbleStorage\NimbleStorageUnAttended.ps1" -inputobject $Code -ErrorAction SilentlyContinue -force

$RunOnce="HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
set-itemproperty $RunOnce "NextRun" ('C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe -executionPolicy Unrestricted -File ' +
'C:\NimbleStorage\NimbleStorageUnattended.ps1')
write-host "Hit CTRL-C in next 60 seconds to abort the AutoReboot cycle"
start-sleep -Seconds 60
shutdown -t 0 -r -f
```

Section A3: NimbleStorageUnattended.ps1

This file pulls all remaining content. Configure your own GitHub project and place these needed files so that your scripts can pull from a repository that only you control. The four lines that need modification to do this are listed in the first `Variable Block`. All other variables to your site such as usernames, passwords, and so on were set in the VM Extension script that you will not place on this repository, but instead use locally when deploying new VMs.

```
#################################################################
# Unattended installation script for connecting AzureStack to a Nimble Storage Infrastructure.                  #
# This script will automatically create a LOG directory at C:\NimbleStorage\Logs

# Variable Block
$NWTuri=            'https://github.com/HewlettPackard/HPEAzureStackOnHPENimbleStorage/raw/master/Setup-NimbleNWT-x64.7.0.2.56.exe'
$NimblePSTKuri=     'https://github.com/HewlettPackard/HPEAzureStackOnHPENimbleStorage/raw/master/HPENimblePowerShellToolkit.300.zip'
$UpdatedPSTKcmd=    'https://raw.githubusercontent.com/HewlettPackard/HPEAzureStackOnHPENimbleStorage/master/AzureStack.ps1'
$UpdatedPSTK=       'https://raw.githubusercontent.com/HewlettPackard/HPEAzureStackOnHPENimbleStorage/master/NimPSSDK.psm1'

# No variables below this line need to be modified by the user/consumer of the script
$AZNSoutfile =      "C:\NimbleStorage\Logs\NimbleInstall.log"
$WindowsPowerShellModulePath="C:\Windows\System32\WindowsPowerShell\v1.0\Modules"
$ScriptLocation=    'C:\NimbleStorage\NimbleStorageUnattended.ps1'
$RunOnceValue=      'C:\Windows\System32\WindowsPowerShell\v1.0\Powershell.exe -executionPolicy Unrestricted -File ' + $ScriptLocation
$RunOnce =          "HKLM:\Software\Microsoft\Windows\CurrentVersion\RunOnce"
$NimbleUser =       (Get-ItemProperty -Path HKLM:\Software\AzureStackNimbleStorage).NimbleUserName
$NimblePassword =   (Get-ItemProperty -Path HKLM:\Software\AzureStackNimbleStorage).NimblePassword
$NimbleArrayIP =    (Get-ItemProperty -Path HKLM:\Software\AzureStackNimbleStorage).NimbleArrayIP

function Post-AZNSEvent([String]$AZNSTextField, [string]$AZNSEventType)
{   # Subroutine to Post Events to Log/Screen/EventLog
    switch -wildcard ($Eventtype)
        {   "Info*"     { $AZNScolor="gray" }
            "Warn*"     { $AZNScolor="green" }
            "Err*"      { $AZNScolor="yellow" }
            "Cri*"      { $AZNScolor="red"
                          $AZNSEventType="Error" }
            default     { $AZNScolor="gray" }
        }
    write-host "- "$AZNStextfield -foregroundcolor $AZNScolor
    $AZNSTextField | out-file -filepath $AZNSoutfile -append
}
function Set-NSASSecurityProtocolOverride
{   # Will override the behavior of Invoke-WebRequest to allow access without a Certificate.
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
    if (-not ([System.Management.Automation.PSTypeName]'ServerCertificateValidationCallback').Type)
    {   $certCallback = @"
            using System;
            using System.Net;
            using System.Net.Security;
            using System.Security.Cryptography.X509Certificates;
            public class ServerCertificateValidationCallback
            {   public static void Ignore()
                {   if(ServicePointManager.ServerCertificateValidationCallback ==null)
                    {   ServicePointManager.ServerCertificateValidationCallback += delegate (   Object obj,
                                                                                                 X509Certificate certificate,
                                                                                               X509Chain chain,
                                                                                             SslPolicyErrors errors
                                                                                           )
                        {   return true;
                        };
                    }
                }
            }
"@
        Add-Type $certCallback
    }
    [ServerCertificateValidationCallback]::Ignore()
}
```

```
function Load-NimblePSTKModules
{   # Loads the Nimble PowerShell Toolkit from the GitHub Site identifed in the Global Variables
    if ( Test-Path 'C:\Windows\System32\WindowsPowerShell\v1.0\Modules\HPENimblePowerShellToolkit' -PathType Container )
    {   Post-AZNSEvent "The HPE NimbleStorage PowerShell Toolkit is installed" "Info"
    } else
    {   Post-AZNSEvent "Now Installing the Nimble PowerShell Toolkit" "Warning"
        invoke-webrequest -uri $NimblePSTKuri -outfile "C:\NimbleStorage\HPENimblePowerShellToolkit.300.zip"
        $PSMPath="C:\Windows\System32\WindowsPowerShell\v1.0\Modules"
        expand-archive -path "C:\NimbleStorage\HPENimblePowerShellToolkit.300.zip" -DestinationPath $WindowsPowerShellModulePath

        Post-AZNSEvent "Now Changing the Nimble Powershell toolkit to add the AzureStack command" "Warning"
        invoke-webrequest -uri $UpdatedPSTK -outfile "C:\NimbleStorage\NimPSSDK.psm1"
        $AZNSRoot=$PSMPath+"\HPENimblePowerShellToolkit"
        Copy-item -path 'C:\NimbleStorage\NimPSSDK.psm1' -destination $AZNSRoot -force

        Post-AZNSEvent "Now adding the AzureStack Command" "Warning"
        invoke-webrequest -uri $UpdatedPSTKcmd -outfile "C:\NimbleStorage\AzureStack.ps1"
        $AZNSScripts=$AZNSRoot+"\scripts"
        Copy-item -path 'C:\NimbleStorage\AzureStack.ps1' -Destination $AZNSScripts -force

    }
}
function Load-WindowsMPIOFeature
{   # Load the Windows MPIO feature. Returns True if a Reboot is required.
    if[ (get-WindowsFeature -name "Multipath-io").installed ]
    {   Post-AZNSEvent "The Windows Multipath IO Feature is already Installed" "Information"
        if [ (get-windowsFeature -name "Multipath-io").InstallState -ne "Installed"]
            {   Post-AZNSEvent "Reboot is required after a Windows Multipath IO Feature Installation" "Warning"
                $ForceReboot=$True
                return $True
            } else
            {   Post-AZNSEvent "The Windows Multipath IO Feature does not require a reboot" "Information"
                return $false
            }
    } else
    {  # Step 1a Install MPIO if not installed
        add-WindowsFeature -name "Multipath-io"
        Post-AZNSEvent "The Windows Multipath IO Feature is not installed, Installing Now!" "Warning"
        Post-AZNSEvent "Reboot is required after a Windows Multipath IO Feature Installation" "Warning"
        $ForceReboot=$True
        return $true
    }
}
function Load-NWTPackage
{   # Download and instlal the Nimble Windows Toolkit. If already installed, return false, otherwise install and request a reboot.
    if ($ForceReboot)
    {   Post-AZNSEvent "The Nimble Windows Toolkit Cannot install since reboot is pending" "warning"
        return $ForceReboot
    }
    else
    {   $NWTsoftware="Nimble Windows Toolkit"
        $installed = (Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Where { $_.DisplayName -eq 'Nimble
Windows Toolkit' }) -ne $null
        if ($installed)
        {   Post-AZNSEvent "The Nimble Windows Toolkit is already installed" "Information"
            return $false
        } else
        {   # If NWT not installed, silent install it
            invoke-webrequest -uri $NWTuri -outfile "C:\NimbleStorage\Setup-NimbleNWT-current.exe"
            $NWTEXE = "C:\NimbleStorage\Setup-NimbleNWT-current.exe"
            $NWTArg1 = "EULAACCEPTED=Yes"
            $NWTArg2 = "HOTFIXPASS=Yes"
            $NWTArg3 = "RebootYesNo=Yes"
            $NWTArg4 = "NIMBLEVSSPORT=Yes"
            $NWTArg5 = "/silent"
            & $NWTEXE $NWTArg1 $NWTArg2 $NWTArg3 $NWTArg4 $NWTArg5
            Post-AZNSEvent "Initiating download and Silent Installation of the Nimble Windows Toolkit" "Warning"
            return $true
        }
    }
}
function Configure-AZNSiSCSI
{   # Will start the ISCSI service, and configure it to connect to the Nimble Array
    Start-Service msiscsi
    Set-Service msiscsi -startuptype "automatic"
    Post-AZNSEvent "Ensuring that the iSCSI Initiator Service is started, and setting it to start automatically" "Warning"
    new-iSCSITargetPortal -TargetPortalAddress $NimbleArrayIP
}
```

```
function Create-AZNSNimbleInitiatorGroups
{   # The Autogenerated Initiator Group will be named for the servers hostname
    $MyLocalIQN=(Get-InitiatorPort | where-object {$_.ConnectionType -like "iSCSI"} ).nodeaddress
    Import-Module HPENimblePowerShellToolkit
    if (Test-NSNimbleWindowsToolkitInstalledConfigured)
    {   Connect-AzNSGroup -UseNWTCredentials $true -IgnoreServerCertificate
        if (Get-NSDisk)
        {   Post-AZNSEvent "Was able to Successfully Connect to the array using the supplied Credentials" "Info"
            if (-not (Get-NSInitiatorGroup -name (hostname) ) )
            {   New-NSInitiatorgroup -name (hostname) -description "Automatically Created using Scripts" -access_protocol "iscsi"
                Post-AZNSEvent "Created new Initiator Group for this host" "Info"
            } else
            {   Post-AZNSEvent "Initiator Group already found for this hostname" "Info"
            }
            $NSIGID=(Get-NSInitiatorGroup -name (hostname) ).id
            $Label = (hostname)+"-Autocreated"
            if ( -not ( get-NSInitiator -label $Label ) )
                {   New-NSInitiator -initiator_group_id $NSIGID -access_protocol "iscsi" -iqn $MyLocalIQN -label $Label
                    Post-AZNSEvent "Created new Initiator for this Initiator Group" "Info"
                } else
                {   Post-AZNSEvent "Initiator Group already found for this hostname" "Info"
                }
        } else
        {   Post-AZNSEvent "Was Unable to connect to the Nimble Array using the Supplied Credentials" "Error"
        }
    } else
    {   Post-AZNSEvent "NWT was not installed or configured, will setup initiators on next reboot" "Warning"
    }
}

function Setup-AZNSNimbleWindowsToolkit
{   #Configure the NWT with the supplied Username and Password.
    $installed = (Get-ItemProperty HKLM:\Software\Microsoft\Windows\CurrentVersion\Uninstall\* | Where { $_.DisplayName -eq 'Nimble
Windows Toolkit' }) -ne $null
    if ($installed)
        {   set-location -path 'C:\Program Files\Nimble Storage\Bin\'
            import-module 'C:\Program Files\Nimble Storage\Bin\Nimble.PowerShellCmdlets.psd1'
            set-location -Path 'C:\nimbleStorage'
            if ( Get-NWTConfiguration | where{$_.GroupMgmtIPList -ne ""} )
                {   Post-AZNSEvent "The Nimble Windows Toolkit has already been configured" "info"
                } else
                {   $NimblePasswordObect = ConvertTo-SecureString $NimblePassword -AsPlainText -force
                    $NimbleCredObject = new-object -typename System.Management.Automation.PSCredential -argumentlist $NimbleUser,
$NimblePasswordObect
                    set-nwtconfiguration -groupmgmtip $NimbleArrayIP -Credential $NimbleCredObject
                    Post-AZNSEvent "The Nimble Windows Toolkit has been Configured" "info"
                }
        }
}

#########################################################################
# MAIN Unattended Installation Script for Nimble Storag on Azure Stack.  #
Set-NSASSecurityProtocolOverride
Load-NimblePSTKModules
Configure-AZNSiSCSI
Create-AZNSNimbleInitiatorGroups
$ForceReboot=Load-WindowsMPIOFeature
$ForceReboot=Load-NWTPackage
Setup-AZNSNimbleWindowsToolkit
if ($ForceReboot)
    {   set-itemproperty -path $RunOnce "NextRun" $RunOnceValue
        Post-AZNSEvent "This Installation Script is set to run again once the server has been rebooted. Please Reboot this server"
"Warning"
    } else
    {   if (Get-ItemProperty -Path $RunOnce)
            {   remove-itemproperty -path $RunOnce "NextRun"
                Post-AZNSEvent "This script will NOT be re-run on reboot" "warning"
            }
        Post-AZNSEvent "This Script has verified that all required software is installed, and that no reboot is needed" "Information"
    }
if ($ForceReboot)
    {   write-host "Hit CTRL-C in next 60 seconds to abort the AutoReboot cycle"
        start-sleep -Seconds 60
        shutdown -t 0 -r -f
    }
```

Section A4: AzureStack.ps1

The `AzureStack.ps1` file is an additional command to add to the HPE Nimble Storage PowerShell Toolkit. For more details, execute `Get-Commands -module HPENimblePowerShell -name *-azns*`. These command additions to the existing PowerShell toolkit are best described using the embedded help instead of looking at the raw PowerShell Code.

The first command is the `Connect-AZNSVolume` command, and the help is shown below;

```
function Connect-AZNSVolume
{   <#
    .SYNOPSIS
        The Command will create and attach a new Nimble Volume to the current host.

    .DESCRIPTION
        The command will retrieve the credentials and IP Address from the Registry for the Nimble Storage Array, and connect to
        that array. The command with then create a volume on the array to match the passed parameters, and assign access to that
        volume to the initiator group name that matches the current hostname. Once the mapping has occurred, the command will
        continue to detect newly detected iSCSI volumes until a volume appears that matches the Target ID of the Volume created.
        Once the iSCSI volume has been detected, it will be connected to persistently, and then refresh the Microsoft VDS
        (Virtual Disk Service) until that device because available as a WinDisk. The New Windisk that matches the serial number
        of the Nimble Storage Volume will then be initialized, placed online, a partition created, and then finally formatted.
        The return object of this command is the Windows Volume that has been created.

        Additional parameters and more granular control are available when using the non-Azure Stack versions of the commands,
        i.e. You can set more features using the New-NSVolume Command however, the steps required to automate the attachment
        or discovery of these volumes is not as automated.

    .PARAMETER name
        This mandatory parameter is the name that will be used by both the Nimble Array to define the volume name, but also as
        the name to use for the Windows Formatted partition.

    .PARAMETER size
        This mandatory parameter is the size in MegaBytes (MB) of the volume to be created. i.e. to create a 100 GigaByte (GB)
        volume, select 10240 as the size value.

    .PARAMETER description
        This commonly a single sentence to descript the contents of this volume. This is stored on the array and can help a
        storage administrator determine the usage of a specific volume. If no value is set, and autogenerated value with be used.

    .EXAMPLE
        PS C:\Users\TestUser> Connect-AZNSVolume -size 10240 -name Test10
        Successfully connected to array 10.1.240.20

        DriveLetter FileSystemLabel FileSystem DriveType HealthStatus OperationalStatus SizeRemaining    Size
        ----------- --------------- ---------- --------- ------------ ----------------- -------------    ----
        R           Test10          NTFS       Fixed     Healthy      OK                      9.93 GB 9.97 GB

    .NOTES
        This module command assumes that you have installed it via the Unattended installation script for connecting Azure Stack
        to a Nimble Storage Infrastructure. All functions use the Verb-Nouns construct, but the Noun is always preceded by AZNS
        which stands for Azure Stack Nimble Storage. This prevents collisions in customer environments Additional information
        about the function or script.

    .LINK
        Please see the GitHUB repository for updated versions of this command. Always use the UnattendedNimbleInstall to install
        the command as to make the command visible you must also alter the HPENimbleStorage PowerShell Toolkit manifest to
        include this file.
    #>
```

Section A5: NimPSSDK.psm1

This file is an updated manifest for the HPE Nimble Storage PowerShell Toolkit that includes the newly created custom command. Instead of listing the entire file, the following example shows which lines have been added to the existing `NimPSSDK.psm1` file. In each case, the new content is highlighted in green.

```
. $PSScriptRoot\scripts\AzureStack.ps1

Export-ModuleMember -Function Test-NS2PasswordFormat, (…)
    Clear-NSAlarm,    Undo-NSAlarm ,     Connect-AZNSVolume,
    Test-NSNimbleWindowsToolkitInstalledConfigured, Get-NSNimbleWindowsToolkitCredentials, Connect-AZNSGroup
```

RESOURCES AND ADDITIONAL LINKS

HPE ProLiant for Microsoft Azure Stack Hub
hpe.com/us/en/solutions/cloud/azure-hybrid-cloud.html

HPE GreenLake consumption-based services
hpe.com/us/en/services/it-consumption.html

HPE Pointnext
hpe.com/pointnext

Microsoft iSCSI support for Azure Stack Hub
docs.microsoft.com/en-us/azure-stack/user/azure-stack-network-howto-iscsi-storage?view=azs-1910

LEARN MORE AT

hpe.com/storage/microsoft

Make the right purchase decision.
Contact our presales specialists.


Chat


Email


Call


Get updates

© Copyright 2020 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Intel is a trademark of Intel Corporation in the U.S. and other countries. Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. VMware and VMware vSphere Hypervisor are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. All third-party marks are property of their respective owners.

a00099112ENW, May 2020

Hewlett Packard Enterprise