Standard Operating Procedure

Title	Build Clusters with Cluster Shared Storage
Document Number:	SOP2023-08
By:	Konrad Jeziorski, student number: L00169649
Approved:	John ORaw
Status:	Released

Revision History and Document Management

	Date	Name
Initial Release:	10AUGUST2023	Konrad Jeziorski
Revision:		

Summary of the Policy/Purpose of this SOP

This Standard Operating Procedure (SOP) aims to provide comprehensive technical instructions to the client's technical personnel responsible for building three clusters with clustered shared storage and configuring a Domain Controller. The environment will utilize Hyper-V as the virtualization platform, and each server was Windows Server 2016 Datacenter installed. The SOP's primary objective is to deliver a detailed, step-by-step guide for a smooth and efficient production environment implementation.

Critical tasks covered by this SOP include:

1. Prerequisites verification, including hardware requirements and resource accessibility.
2. Domain Controller configuration, encompassing network settings and Active Directory Domain Services (AD DS) setup.
3. The cluster-building process covers server setup, network configuration, and connection to shared storage.
4. Hyper-V cluster setup, feature selection, and configuring high-availability settings like Cluster Shared Volumes (CSV).
5. Verification testing, validating Domain Controller functionality, cluster operation with shared storage, and successful failover of services between cluster nodes.
6. Where feasible, automation using PowerShell was encouraged to streamline tasks and maintain consistency across the environment.

Scope of this SOP

The SOP focuses on providing detailed technical instructions to the client's technical personnel responsible for building clusters with clustered shared storage and configuring a Domain Controller. The SOP applies to environments utilizing Hyper-V as the virtualization platform. It outlines the step-by-step process to implement a sample production environment, emphasizing the importance of following prerequisites, configuring the Domain Controller, building the cluster, and conducting verification testing. The SOP aims to ensure the successful deployment of properly built clusters with clustered shared storage in a Hyper-V virtualization environment. The SOP was based on information provided by a client whose domain was *maynooth.ads.electric-petrol.ie.*

Actors

Responsible:	
Accountable:	
Consulted:	
Informed:	

Contents

Resources and Prerequisites

1. Image of the Guest OS, downloaded from the software repository, which has had its checksum verified.
2. A suitable host
    i. Windows with Hyper-V
3. Access to documentation, for reference
    i. Microsoft Server
    ii. Hyper-V
4. Hard disk space to accommodate the proposed build.

Inputs:

Domain Controller Server OS Specification

OS	Windows Server 2016 Datacenter 64-bit GUI
VM Name	DC1
Processors / Cores	2
Main memory	4 GB
Network Adapters	External
Virtual Disks	100 GB
Admin username and password	The password must contain at least one number, symbol, uppercase and lowercase letter.

Table 1: DC OS Specification

First Node Server OS Specification

OS	Windows Server 2016 Datacenter 64-bit Core
VM Name	Cluster12
Processors / Cores	2
Main memory	4 GB
Network Adapters	External, two Private
Virtual Disks	100 GB, six 5 GB
Admin username and password	The password must contain at least one number, symbol, uppercase and lowercase letter.

Table 2: First Cluster OS Specification

Second Node Server OS Specification

OS	Windows Server 2016 Datacenter 64-bit Core
VM Name	Cluster12
Processors / Cores	2
Main memory	4 GB
Network Adapters	External, two Private
Virtual Disks	100 GB, six 5 GB
Admin username and password	The password must contain at least one number, symbol, uppercase and lowercase letter.

Table 3: First Cluster OS Specification

Third Node Server OS Specification

OS	Windows Server 2016 Datacenter 64-bit Core
VM Name	Cluster13
Processors / Cores	2
Main memory	4 GB
Network Adapters	External, two Private
Virtual Disks	100 GB, six 5 GB
Admin username and password	The password must contain at least one number, symbol, uppercase and lowercase letter.

Table 4: Third Cluster OS Specification

Network Configuration:

Configuring Adapters

	Ethernet1 (External network)	Ethernet2 (Private network)	Ethernet3 (Private network)
DC1	192.168.190.10 /24		
Cluster11	192.168.190.11 /24	169.254.0.11 /16	169.254.0.111 /16
Cluster12	192.168.190.12 /24	169.254.0.12 /16	169.254.0.112 /16
Cluster13	192.168.190.13 /24	169.254.0.13 /16	169.254.0.113 /16

Table 5: IP Addressing for Servers Adapters

Actions

Create Adapters

1. On the Hyper-V host, in **Virtual Switch Manager** create two virtual switches:
   a. One labeled as "External" to enable communication between virtual machines, clients, and the rest of the network. Uncheck the option "Allow management operating system to share this network adapter" since you have a dedicated network adapter for Hyper-V. However, if you have only one network adapter on your Hyper-V host, leave that option checked; otherwise, your host will lose its network connection.
   b. The second one is private, named “VMsOnly” and is only for communication between host and VMs (Inter-cluster communication).

Create the DC1 VM

Option 1: Downloading and Installing Windows Server 2016 Datacenter VM

1. Ensure Hyper-V host has the required hardware resources and the Hyper-V role installed.
2. Go to the Microsoft Volume Licensing Service Center or an authorized source to download the Windows Server 2016 Datacenter ISO file.
3. To ensure the integrity of the downloaded file, calculate and verify the checksum using a reliable checksum tool (MD5 or SHA256) and compare it with the official checksum provided by Microsoft.
4. Open Hyper-V Manager and select "New" > "Virtual Machine" to start the Virtual Machine Wizard.
5. Name the virtual machine, specify the location, and choose “Generation 2”.
6. Allocate CPU, RAM, Disk and connect a virtual switch for networking. Review and verify that all parameters match those in *Table 1* then click finish or go back and correct any errors.

7. In the Virtual Machine Wizard, choose to install the operating system from a bootable CD/DVD-ROM and attach the downloaded ISO file.
8. Chose “Windows Server Datacenter with Desktop Experience”; Follow the prompts to complete the installation and accepting the license terms.
9. Create a strong password for the Administrator, in accordance with the company's password policy.
10. After the initial login, navigate to "Settings" > "Update & Security" > "Windows Update" and check for updates. Install all available updates to ensure the system is up to date.
11. Set up the network settings based on requirements from *Table 5*. Ensure the virtual machine has proper Default Gateway DNS settings.
12. Perform basic tests to ensure the Windows Server 2016 Datacenter GUI installation is successful and functional.

Option 2: Creating a Windows Server 2016 Datacenter GUI VM using Gold (Master) Image

1. Ensure that your Hyper-V host machine meets the hardware requirements and has the Hyper-V role installed.
2. Request or obtain the fully patched Windows Server 2016 Datacenter Gold Image in VHD or VHDX format from a trusted source or internal company team.
3. Open Hyper-V Manager, right-click on the Hyper-V host, and select "Import Virtual Machine".
4. Browse to the location of the Gold Image VHD/VHDX and follow the import wizard to import the virtual machine.
5. Customize the virtual machine's settings; review and verify that all parameters match those in *Table 1* then click finish or go back and correct any errors.
6. Click "Finish" to complete the import process, then right-click on the imported virtual machine and select "Connect" to open the console.
7. Power on the virtual machine and follow the prompts to configure Windows Server 2016 Datacenter with GUI.
8. Create a strong password for the Administrator, in accordance with the company's password policy.
9. Set up the network settings based on requirements from Table 5. Ensure the virtual machine has proper Default Gateway DNS settings.
10. Perform basic tests to ensure the Windows Server 2016 Datacenter GUI installation is successful and functional.

Create the DC1 Roles and Services

Services can be configured using the GUI. This example shows how to configure services on DC Server using PowerShell.

To launch PowerShell in administrator mode on Windows Server 2016 GUI version, follow these steps:

1. Click on the "Start" button in the bottom-left corner of the screen.
2. Search for "PowerShell" in the search bar.
3. Right-click on "Windows PowerShell" in the search results.
4. From the context menu, select "Run as administrator."
5. If prompted by User Account Control, click "Yes" to confirm the action.

PowerShell will now open with administrative privileges, allowing you to perform tasks that require elevated permissions. Enter the following commands:

```
# Windows PowerShell script for AD DS Deployment
#
# Import the ADDSDeployment module, which provides cmdlets for deploying Active Directory Domain Services
Import-Module ADDSDeployment
#  Install and configure Active Directory Domain Services
Install-ADDSForest `
# Specifies whether to create a DNS delegation for the DNS server. In this case, it's set to false
-CreateDnsDelegation:$false `
# Specifies the path where the Active Directory database will be stored
-DatabasePath "C:\Windows\NTDS" `
# Sets the domain functional level to "WinThreshold"
-DomainMode "WinThreshold" `
# Specifies the fully qualified domain name (FQDN) for the domain being created
-DomainName "maynooth.ads.electric-petrol.ie" `
# Sets the NetBIOS name of the domain
-DomainNetbiosName "MAYNOOTH" `
# Sets the forest functional level to "WinThreshold"
-ForestMode "WinThreshold" `
# Specifies whether to install DNS (Domain Name System) along with Active Directory
-InstallDns:$true `
# Specifies the path where log files will be stored
-LogPath "C:\Windows\NTDS" `
# Specifies whether to reboot the computer after the installation is completed. Here, it's set to false
-NoRebootOnCompletion:$false `
#  Specifies the path where the Sysvol directory will be stored
-SysvolPath "C:\Windows\SYSVOL" `
# Forces the installation to proceed without user confirmation
-Force:$true
```

6. Perform checks to ensure the Windows Server 2016 Datacenter GUI has services installed.

Create Node VMs

Prerequisites

- Hyper-V host with the Hyper-V role installed,
- Verified ISO image for Windows Server 2016.

Installing of Node

1. Open Hyper-V Manager on DC1 and select "New" > "Virtual Machine" to start the Virtual Machine Wizard.
2. Name the virtual machine, specify the location, and choose “Generation 2”.
3. Allocate CPU, RAM, Disk and connect virtual switch (external) for networking. Review and verify that all parameters match those in *Table 2* / *Table 3* / *Table 4,* then click finish or go back and correct any errors.
4. In the settings for VM, from “SCSI Controller” choose operating system from a bootable CD/DVD-ROM and attach the downloaded ISO file.
5. Chose “Windows Server Datacenter” and follow the prompts to complete the installation and accepting the license terms.
6. Create a robust password for the Administrator in alignment with the company's password policy.

Update Windows Server 2016 Core VMs

7. Log in to the Server with an administrator account.
8. Launch the sconfig tool by typing “sconfig” in the command prompt and pressing Enter.
9. Once the sconfig tool is running, select option 6, which is "Download and Install Updates."
10. Choose option 2 to install available recommended updates. The system will scan for available updates and display their count.
11. Confirm your choice to install, and the system will start downloading and installing the updates. Wait until the update installation process is complete.
12. After the installation is finished, the system will prompt you to restart the server to apply the changes. Choose the restart option to complete the update process.

Initial Configuration of Node

13. Rename VM.
14. Configure the external network adapter settings according to the requirements specified in *Table 5*. Ensure that the virtual machine has the appropriate IPs for Default Gateway and DNS.
15. Join Server to a domain “maynooth.ads.electric-petrol.ie”
16. Restarts the computer to apply the changes.

To complete the above tasks for Node named “Cluster11”, open PowerShell in it and execute the following commands:

```
# Changes the computer name
Rename-Computer Cluster11
# Retrieves information about all network adapters installed on the computer (for this example, it currently displays information about one adapter with an index of 3)
Get-NetAdapter
# Displays the IP address configuration for the network adapter with index 3 (External NetAdapter index)
Get-NetIPAddress -InterfaceIndex 3
# Creates a new IP address with a subnet mask and sets default gateway for the adapter with index 2
New-Netipaddress -InterfaceIndex 3 -IpAddress 192.168.190.11 -DefaultGateway 192.168.190.2 -PrefixLength 24
# Configures DNS server addresses for the adapter with index 2. Sets DNS server address for that adapter.
Set-DnsClientServerAddress -InterfaceIndex 2 -ServerAddresses 192.168.190.10
# Used to join a computer to a domain. Specify “maynooth.ads.electric-petrol.ie” as the domain name for this example
Add-Computer
# Restarts the computer to apply the changes made by the previous commands
Restart-Computer
```

Repeat the same procedures for second (Cluster12) and third (Cluster13) nodes.

Add Virtual Drives

Creating virtual disks for cluster nodes enables data sharing, ensuring high availability, scalability, and failure resilience. Virtual disks serve as central data sources, replicated across different cluster nodes, ensuring consistency and accessibility even in the event of node failures. This example describes the steps to creating six disks per node.

Create directories on the host hard drive for the virtual drives

1. Open File Explorer on your host machine.
2. Navigate to the directory where you want to create the folders. This could be a specific location on your hard drive or any other storage location.

3. Create a folder named "Cluster11". Right-click in the desired directory, select "New," and then choose "Folder." Rename the folder to "Cluster11".
4. Create two more folders following the same steps, naming them "Cluster12" and "Cluster13". These directories will serve as containers (placeholders) for the virtual drives associated with each node.

Adding Virtual Disks

5. Locate and select the virtual machine corresponding to Cluster11 in the Hyper-V Manager.
6. Right-click on the selected virtual machine for Cluster11 and choose the "Settings" option.
7. On the left side of the "Settings" window, select "Controller: SCSI Controller".
8. Click the "Add" button located at the top of the window.
9. Select the "New Virtual Hard Disk" option and click "Next".
10. Choose the type of disk "Dynamically expanding”.
11. Specify the disk size (5 GB for this example).
12. Determine the location to save the virtual disk file (“Cluster11” directory for this example).
13. Assign a name for the virtual disk file ("Cluster11_DiskX" for this example).
14. Review the configuration summary and click "Next".
15. Click "Apply" to complete the configuration (the virtual disk will be added to the Cluster11 VM).
16. Add five more disks using the same method.
17. Repeat the steps above for the virtual machines of “Cluster12” and “Cluster13”, creating 6 virtual disks for each cluster.
18. Start the virtual machines for “Cluster11”, “Cluster12”, and “Cluster13”. The newly added virtual disks will be available in the operating systems of these virtual machines.

These disks can now be utilized to enhance data storage, availability, and performance within the respective clusters.

Add Adapters

This procedure outlines the steps to create and add private network adapters to the virtual machines of Nodes using Hyper-V Manager.

Creating Private Network Adapters

1. Click on "Action" in the top menu and select "Virtual Switch Manager".
2. In the "Virtual Switch Manager" window, select "New virtual network switch" on the left.
3. Choose "Private" as the type of switch.
4. Click "Create Virtual Switch".
5. Assign a name for the first private switch (“VMsOnly” in this example).

Adding Private Adapters to Nodes

6. Locate and select the virtual machine for “Cluster11” in the Hyper-V Manager.
7. Right-click on the selected virtual machine for Cluster11 and choose the "Settings" option.
8. On the left side of the "Settings" window, select "Add Hardware" > "Network Adapter".
9. Choose "Private Virtual Switch" and select "VMsOnly".
10. Click "Add" to add the private adapter.
11. Repeat steps 8-10 and add additional private adapter.
12. Repeat the above steps for the virtual machines of “Cluster12”and “Cluster13”.
13. Start the virtual machines of Nodes. The newly added private network adapters will be accessible within the operating systems of these machines.

Private Adapters Configuration

14. Launch the sconfig tool by typing “sconfig” in the command prompt and pressing Enter.
15. Select option 8 which corresponds to network settings.
16. Choose the index of the adapter you want to configure. For this scenario is "5" for the first adapter.
17. Follow the tool's prompts to input the IP address, subnet mask, default gateway for the selected adapter, according to the information specified in *Table 5*. Once done, the tool will confirm the changes.
18. Repeat steps 15-17 for the second adapter, in this case with the index "6".
19. Repeat steps 14-18 for all Node VMs.

Rename Network Adapters

In each Node VM, launch PowerShell and type the following commands:

```
# Display network configuration information
Get-NetIPConfiguration
# Rename network adapter "Ethernet" to "Management"
Rename-NetAdapter -Name "Ethernet" -NewName "Management"
# Rename network adapter "Ethernet 2" to "Provider 1"
Rename-NetAdapter -Name "Ethernet 2" -NewName "Provider 1"
# Rename network adapter "Ethernet 3" to "Provider 2"
Rename-NetAdapter -Name "Ethernet 3" -NewName "Provider 2"
```

These commands demonstrate how to rename network adapters using PowerShell, allowing us to provide more meaningful names that reflect their roles or purposes.

Adapter Checks and Connectivity Tests

Tests checking the IP addressing of the adapters of all servers and their connectivity with the domain for this example have been verified in the screenshots in the Networking and Connectivity Tests. (*Figure 1*)

Enable Remote Desktop for Nodes and Install Necessary Software

For each Node VM, follow the steps below:

1. Run the sconfig tool by typing sconfig in the command prompt and pressing Enter.
2. In the sconfig tool, choose option 7, which corresponds to Remote Desktop settings.
3. Choose option “E” to enable Remote Desktop. A confirmation prompt will appear.
4. Select the security level for Remote Desktop.
5. Click on “Ok” and choose the option "15 - Exit" to exit.
6. Open PowerShell in it and execute the following command:

```
# Enables the PowerShell Remoting feature on the local computer. The “-Force” parameter is used to enable remoting without requiring user confirmation. This is commonly used for automation purposes.
Enable-PSRemoting -Force
# Install the Remote Server Administration Tools (RSAT) for Hyper-V, providing additional tools for managing Hyper-V features remotely
Add-WindowsFeature rsat-hyper-v-tools
# List Windows features starting with "Failover"
Get-WindowsFeature failover*
# Install the Failover Clustering feature, which is important for creating and managing clusters of servers.
Install-WindowsFeature Failover-Clustering
# Install the File Server role and feature, enabling the server to act as a file server.
Install-WindowsFeature FS-FileServer
```

```
# Create a registry entry to allow remote administrative access to accounts
New-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System -Name LocalAccountTokenFilterPolicy -Value 1
# List Windows features starting with "RSAT-Cluster", helping identify the appropriate RSAT features for clustering.
Get-WindowsFeature rsat-cluster*
# Install the RSAT Clustering tools, which provide the necessary tools for managing and configuring clusters remotely
Install-WindowsFeature RSAT-Clustering
```

Enable Remote Desktop from a Domain Controller using Server Manager (Optional)

1. Launch Server Manager by clicking on the "Start" button and typing "Server Manager," then selecting the "Server Manager" application.
2. If the node servers are not added to Server Manager yet, you can add them to access them remotely. To do so:
   a. Click on "Manage" in the top menu of Server Manager.
   b. Choose "Add Servers."
   c. Enter the names or IP addresses of the node servers you want to add.
   d. Click "OK" to add the servers to the list.
3. Launch Remote Desktop.
   a. In the left panel of Server Manager, click on "All Servers" to display the list of all servers.
   b. Select server from the list.
   c. Right click and choose "Remote Desktop Connection".
   d. Provide user credentials to establish a remote connection.
4. Open PowerShell in it and execute the following command:

```
# Establishes a remote PowerShell session on the computer specified by the -ComputerName parameter. In this example, the remote computer's name is "Cluster12.maynooth.ads.electric-petrol.ie"
Enter-PSSession -ComputerName Cluster12.maynooth.ads.electric-petrol.ie
```

Tests for the reliability of using Remote Desktop in DC for this example is shown when creating a storage pool in *Figure 19*.

Build the Cluster

This procedure outlines the steps to test and create Cluster.

Initial Cluster Test

1. In selected server, enter the following commands in PowerShell:

```
# Perform a cluster validation test on the specified nodes
Test-Cluster -Node Cluster11, Cluster12, Cluster13 -Include Inventory, Network, "System Configuration"
```

By running this command, we are initiating a cluster validation test on the specified nodes to ensure that they meet the required criteria for forming a failover cluster. The provided values indicate that three types of tests will be conducted: "Inventory" (checks hardware and software information), "Network" (verifies network connectivity and configuration), and "System Configuration" (ensures system settings are compatible with clustering requirements). The test results will provide insights into any issues or potential problems that need to be addressed before creating or maintaining a cluster. The specific test results are typically displayed in the PowerShell console, but they can also be saved to a log file for further analysis.

Cluster Creation

2. A cluster is a collection of interconnected nodes that work together to enhance system availability, fault tolerance, and resource utilization. Execute the following command to create a new cluster and retrieve information about existing clusters by running the following command in PowerShell:

```
# Create a new cluster named "Cluster10" with specified nodes and options. The "-NoStorage" option indicates that no storage will be configured as part of the cluster, and the "-StaticAddress 192.168.190.250" option sets a static IP address for the cluster.
New-Cluster -Name Cluster10 -Node Cluster11, Cluster12, Cluster13 -NoStorage -StaticAddress 192.168.190.250
# Retrieve information about the clusters present in the environment.
Get-Cluster
# Retrieve information about the cluster networks in the environment.
Get-ClusterNetwork
# Retrieve information about the network interfaces used by the cluster nodes to communicate with each other and the network.
Get-ClusterNetworkInterface
```

The figures for cluster test and creation are shown in the section Test Cluster and Cluster Creation (*Figure 3*).

Cluster Shared Storage

Procedures below is intended for configuring storage and managing disks within clusters.

Enabling Storage Spaces Direct (S2D)

1. Run the following command to enable S2D with specific configurations:

```
# Enable Storage Spaces Direct for a cluster with specific configuration options, such as disabling cache devices, turning off automatic configuration, bypassing eligibility checks, and suppressing confirmation prompts.
Enable-ClusterS2D -CacheState Disabled -AutoConfig:0 -SkipEligibilityChecks -Confirm:$false
```

Command explanation in more detail: The provided PowerShell command includes several parameters that govern specific behaviors during the process of enabling S2D on a cluster. Each parameter has a distinct purpose and effect, contributing to the customization of the S2D configuration.

1. *-CacheState Disabled*: Disables the use of cache devices for storage acceleration in S2D. Cache devices are not utilized to improve storage performance.
2. *-AutoConfig:0*: Turns off automatic configuration. In this case, manual configuration will be performed instead of allowing the system to automatically configure settings.
3. *-SkipEligibilityChecks*: Skips cluster eligibility checks. This means that the cluster's compatibility with S2D requirements will not be verified before enabling S2D.
4. *-Confirm:$false*: Disables confirmation prompts for the operation. This prevents the command from asking for user confirmation before proceeding.

Managing Physical Disks

2. To view information about physical disks that can be pooled, execute the following command:

```
# Retrieves and displays details about physical disks that are eligible for pooling within the cluster. It shows the friendly name, pool eligibility status, media type, and physical location of the disks
Get-PhysicalDisk | Where-Object { $_.CanPool -eq $true } | Format-Table FriendlyName, CanPool, MediaType, PhysicalLocation
```

The output for this command is shown in the Test Cluster and Cluster Creation (*Figure 4*)

Creating HDD Disks within Storage Pool

3. The commands below collectively guide creating HDD disks within the "S2DPool1" storage pool. The procedures encompass disk assignment, storage pool creation, disk addition, media type specification, and information retrieval, all tailored toward setting up HDD disks for optimal storage within the cluster environment.

```
# Retrieve physical disks with a specific location and store them in a variable.
$Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*LUN 11" }
# Create a Storage Pool: Establish a new storage pool named "S2DPool1" using the disks from the variable.
New-StoragePool -StorageSubSystemFriendlyName *Cluster* -FriendlyName S2DPool1 -
ProvisioningTypeDefault Fixed -PhysicalDisk $Disks
# Fetch details about all existing storage pools, including "S2DPool1"
Get-StoragePool
# Retrieve information about physical disks within the "S2DPool1" storage pool.
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk
# Sequentially add physical disks from the variable to "S2DPool1"
Add-PhysicalDisk -PhysicalDisks $Disks -StoragePoolFriendlyName S2DPool1
# Retrieve physical disks with a specific location for the next set.
$Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*LUN 12" }
# Add the newly retrieved disks to "S2DPool1"
Add-PhysicalDisk -PhysicalDisks $Disks -StoragePoolFriendlyName S2DPool1
# Retrieve physical disks with a specific location for another set.
$Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*LUN 13" }
# Add the newly retrieved disks to "S2DPool1"
Add-PhysicalDisk -PhysicalDisks $Disks -StoragePoolFriendlyName S2DPool1
# Retrieve physical disks with a specific location for one more set.
$Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*LUN 14" }
# Add the last set of retrieved disks to "S2DPool1."
Add-PhysicalDisk -PhysicalDisks $Disks -StoragePoolFriendlyName S2DPool1
# Fetch and display the physical locations of disks within "S2DPool1"
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk | Format-Table PhysicalLocation
# Set the media type of disks within "S2DPool1" to "HDD."
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk | Set-PhysicalDisk -MediaType HDD
# Fetch specific details about disks within "S2DPool1"
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk | Format-Table FriendlyName, CanPool
MediaType, PhysicalLocation
```

The output showing created HDD disks is shown in the Creating Storage Pool and Managing Physical Disks (Figure 7)

Creating SSD Disks

4. By strategically adding SSDs, the storage pool gains a performance boost through their speed and responsiveness. These commands cater to optimizing the cluster's storage environment, ensuring that the incorporation of SSDs enhances overall storage performance and responsiveness.

```
# Retrieve physical SSD disks with a specific location pattern.
$SSD_Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*SSD*" }
# Add the SSD disks from the variable to the "S2DPool1" storage pool.
Add-PhysicalDisk -PhysicalDisks $SSD_Disks -StoragePoolFriendlyName S2DPool1
# Retrieve physical HDD disks with a different location pattern.
$HDD_Disks = Get-PhysicalDisk | Where-Object { $_.PhysicalLocation -like "*HDD*" }
# Add the HDD disks from the variable to the "S2DPool1" storage pool.
```

```
Add-PhysicalDisk -PhysicalDisks $HDD_Disks -StoragePoolFriendlyName S2DPool1
# Identify disks within "S2DPool1" storage pool with an unspecified media type and set them as SSDs.
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk | Where-Object { $_.MediaType -eq
"Unspecified" } | Set-PhysicalDisk -MediaType SSD
# detailed information about disks within "S2DPool1" storage pool.
Get-StoragePool -FriendlyName S2DPool1 | Get-PhysicalDisk | Format-Table FriendlyName, CanPool,
MediaType, PhysicalLocation
```

By incorporating both SSD and HDD disks, and updating the media type as needed, these commands establish a versatile and high-performance storage pool tailored to the cluster's requirements.

The output, which includes both HDDs and SSDs, is shown within the Creating Storage Pool and Managing Physical Disks (Figure 8)

Configuring S2D and Updating Provider Cache

5. The following commands configure S2D settings and update the storage provider cache within a cluster environment. These steps contribute to enhancing storage performance and maintaining accurate storage resource information.

```
# Activate cache devices to enhance storage performance in the S2D setup, "-Verbose" parameter ensures
that detailed insights into the command's operations are displayed.
Set-ClusterS2D -CacheState Enabled -Verbose
#  Refresh the cache of storage provider information in a cluster environment.
The parameter "-DiscoveryLevel Full" indicates a comprehensive update and refresh using the latest and
most accurate information obtained from the storage providers.
Update-StorageProviderCache -DiscoveryLevel Full
```

Creating Fault-Tolerant Volumes in a Cluster Environment

The final step is creating fault-tolerant storage volumes within a cluster environment. By configuring various resiliency settings such as mirror and parity, the Power Shell commands below provide data redundancy and availability, enhancing overall data protection and storage management efficiency.

6. Create a Mirrored Volume with Standard Resiliency using the following command:

```
# Create a mirrored storage volume named "Mirror2" within the specified storage pool "S2DPool1." It utilizes
the mirror resiliency setting to duplicate data across disks, ensuring redundancy and improved fault tolerance.
New-Volume -FriendlyName Mirror2 -FileSystem CSVFS_ReFS -StoragePoolFriendlyName S2DPool1 -Size 1GB -
ResiliencySettingName Mirror
```

7. Create a Mirrored Volume with Enhanced Redundancy:

```
# Establish a mirrored volume labeled "Mirror3" within the storage pool "S2DPool1." The mirror resiliency
setting is employed, and the parameter "-PhysicalDiskRedundancy 2" further enhances the resiliency by storing
data on two redundant physical disks.
New-Volume -FriendlyName Mirror3 -FileSystem CSVFS_ReFS -StoragePoolFriendlyName S2DPool1 -Size 1GB -
ResiliencySettingName Mirror -PhysicalDiskRedundancy 2
```

8. Create a Parity-Based Volume:

```
# Generate a volume named "Parity1" within the specified storage pool "S2DPool1." The volume uses the parity
resiliency setting, which involves striping data with parity information across disks. This approach offers data
protection and storage efficiency.
New-Volume -FriendlyName Parity1 -FileSystem CSVFS_ReFS -StoragePoolFriendlyName S2DPool1 -Size 1GB -
ResiliencySettingName Parity
```

Outputs for the appropriately numbered steps 5 (Figure 9), 6 (Figure 10), 7 (Figure 11), 8 (Figure 12).

These three volumes are integral to maintaining data availability and protection within a cluster environment, contributing to efficient storage management and robust data handling.

Acceptance Tests

Test based on provided example. During the testing phase, the primary focus revolved around executing PowerShell commands associated with critical junctures. These carefully selected commands were aimed at verifying the successful completion of pivotal steps, ensuring that the intended outcomes were achieved. Furthermore, the execution of these commands acted as tangible proof of the procedure's proper functionality, offering concrete evidence that the planned actions had been successfully implemented.

Networking and Connectivity Tests

During the assessment phase, one of the critical aspects was the validation of the accurate network configuration (specification from Table 5). This validation process aimed to ensure that the network settings for Nodes (Cluster11, Cluster12, Cluster13) were correctly configured and aligned with the specified parameters. Two corresponding screenshots were captured as a part of this verification process (one in Power Shell, the second in sconfig tool). These screenshots visually depict the configuration details of all Nodes, showcasing their individual settings. The collected data included vital information such as adapters' names, IP addresses, subnet masks, default gateways, DNS server addresses, and domain affiliations. As part of the thorough testing, the connectivity of these Nodes was evaluated to ensure their ability to communicate with the domain. This involved testing the capability of each Node to successfully ping the "maynooth.ads.electric-petrol.ie" domain, thereby validating their network connectivity and interaction within the domain environment.

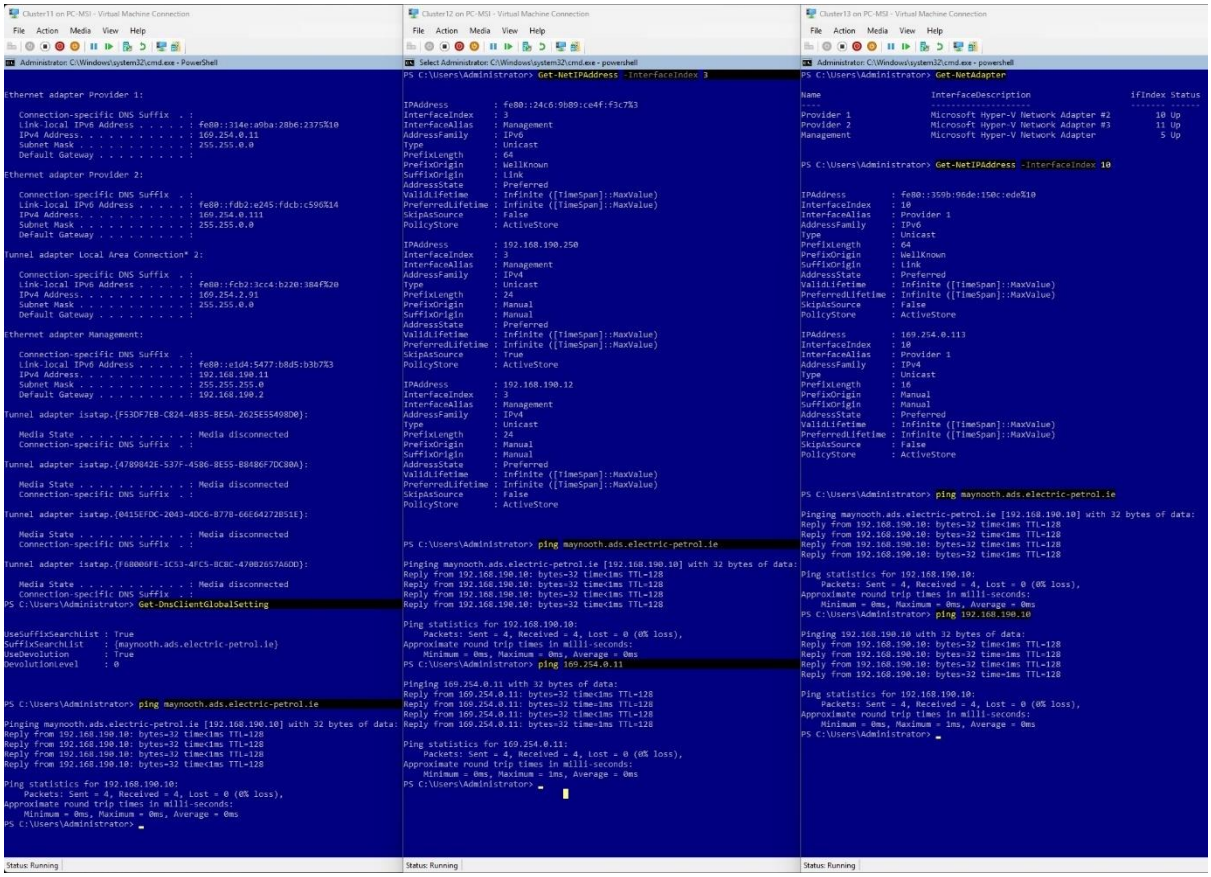


Figure 1: Network Configuration and Connectivity

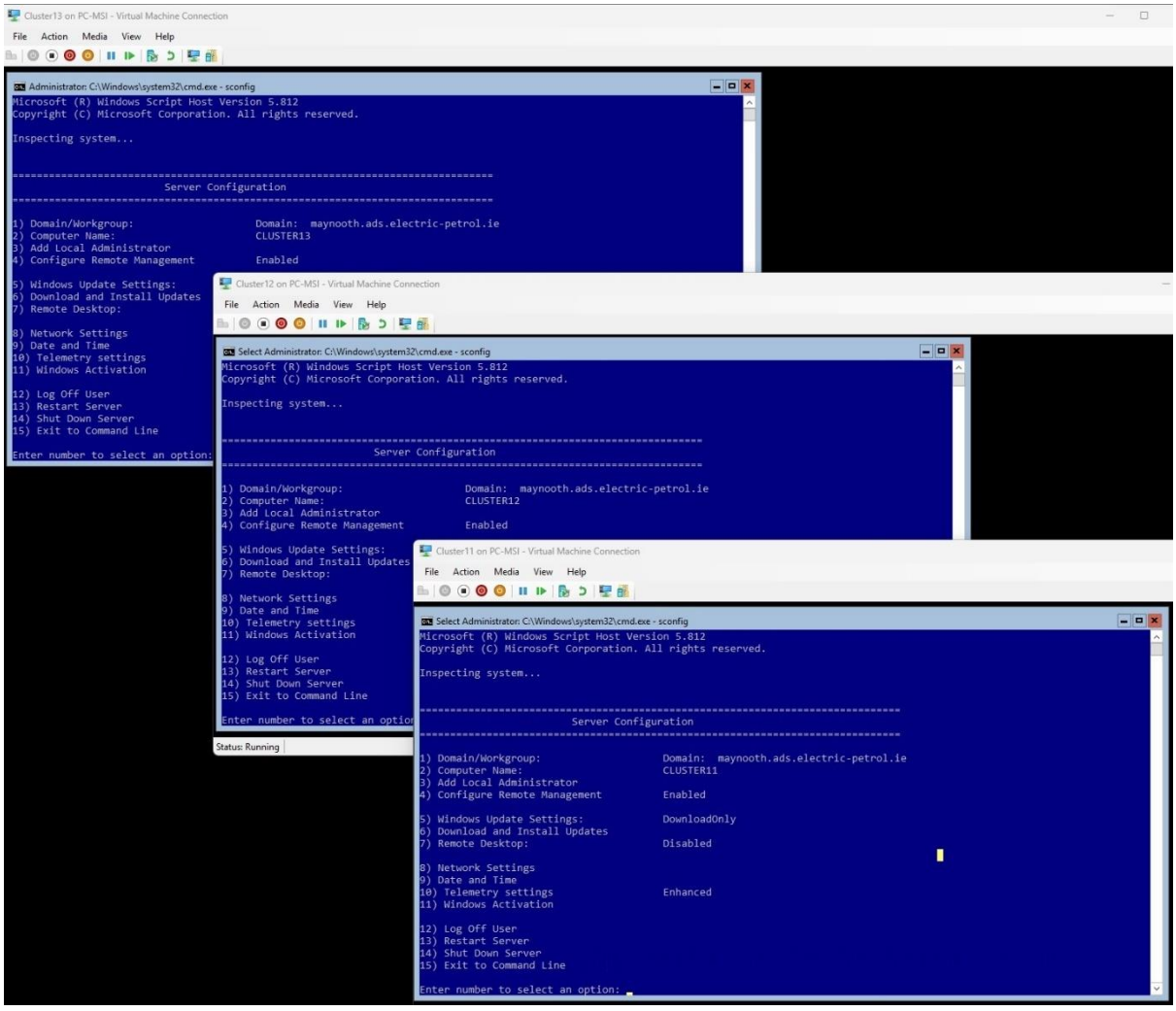


Figure 2: sconfig - Nodes ant their Domains

Test Cluster and Cluster Creation

Through Cluster Validation Testing: the process identified potential issues and irregularities that could hinder the successful creation or operation of a cluster. This approach enabled proactive troubleshooting and rectification of any discrepancies, fostering a robust foundation for cluster deployment. Moreover, these tests provided insights into the readiness of the nodes to work cohesively within the cluster environment. The results contributed to informed decision-making and ensured that the cluster would function as intended, promoting system stability and data integrity. This test is shown in *Figure 3*.

```
PS C:\Users\Administrator> Test-Cluster -Node Cluster11, Cluster12, Cluster13 -Include Inventory, Network, "System Configuration"
>>
WARNING: System Configuration - Validate Software Update Levels: The test reported some warnings..
WARNING:
Test Result:
HadUnselectedTests, ClusterConditionallyApproved
Testing has completed for the tests you selected. You should review the warnings in the Report.  A cluster solution is supported by Microsoft only
 if you run all cluster validation tests, and all tests succeed (with or without warnings).
Test report file path: C:\Users\Administrator\AppData\Local\Temp\2\Validation Report 2023.08.06 At 16.34.13.htm

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        06/08/2023     16:36         747889 Validation Report 2023.08.06 At 16.34.13.htm


PS C:\Users\Administrator>
```

Figure 3: Initial Test for Cluster

After confirming the readiness of the nodes through the testing phase, the next crucial step was establishing the cluster itself. This phase marked the transformation from individual nodes into a unified entity designed to enhance system reliability, fault tolerance, and efficient resource utilization. The successful creation of the cluster resulted in a holistic environment capable of managing workload distribution, facilitating failover mechanisms, and enhancing overall system availability. This collective unit streamlined administrative tasks and resource allocation, simplifying the management and maintenance of the infrastructure. Creating the cluster marked a significant achievement, representing the culmination of efforts to form a resilient, cooperative ecosystem. The cluster's establishment laid the groundwork for heightened system stability, optimized resource usage, and improved operational efficiency by unifying the nodes into a harmonious ensemble. *Figure 4* shows the newly created cluster named Cluster10.

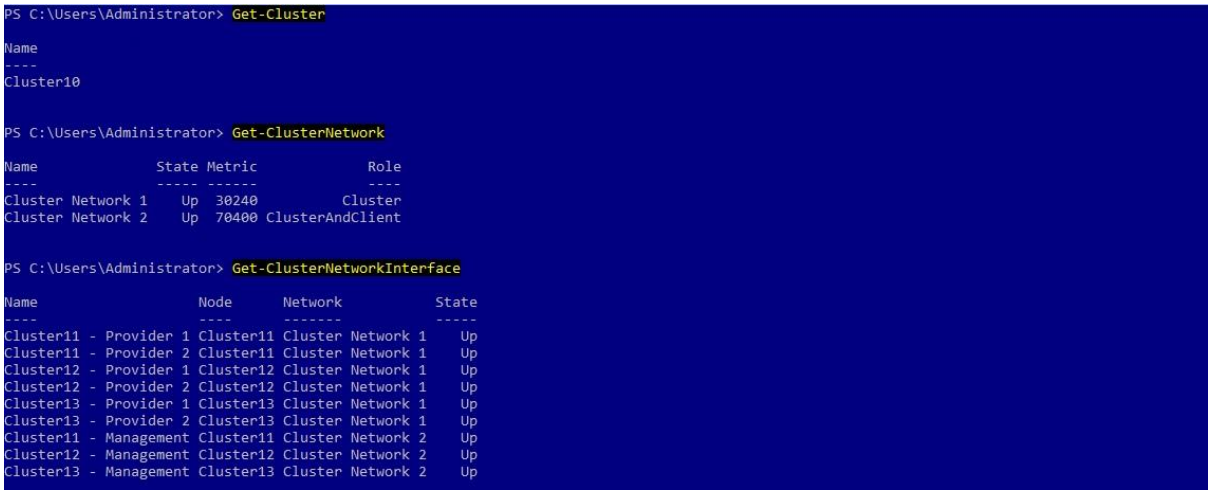


Figure 4: Cluster Created

Validating Cluster Shared Storage Configuration

This tests section demonstrates the verification of the cluster-shared storage configuration through a series of PowerShell commands. Each step is accompanied by a corresponding screenshot showcasing the successful execution of the commands and the desired outcomes.

Enabling Storage Spaces Direct (S2D) and retrieving and displaying eligible physical disks: *Figure 5* shows the execution of the "Enable-ClusterS2D" command, which activates Storage Spaces Direct (S2D) with tailored configurations. The parameters specified in the command precisely control cache utilization, automatic configuration, eligibility checks, and confirmation prompts. Detailed report of this operation has been saved to a folder location within the subdirectory "Windows". The figure also showcases the "Get-PhysicalDisk" command in action. The output provides a detailed overview of the physical disks eligible for pooling within the cluster. The visual representation illustrates the critical attributes of each disk, including its friendly name, pool eligibility status, media type, and physical location. Within the context of these steps, it's essential to highlight the utilization of Virtual LUNs (Logical Unit Numbers). Virtual LUNs are virtual representations of storage devices that facilitate efficiently allocating storage resources. In our depicted scenario, each step revolves around interacting with these Virtual LUNs. These Virtual LUNs act as the building blocks for establishing robust storage environments within the cluster architecture.

```
PS C:\Users\Administrator> Enable-ClusterS2D -CacheState Disabled -AutoConfig:0 -SkipEligibilityChecks -Confirm:$false
WARNING: 2023/08/06-18:58:42.987 C:\Windows\Cluster\Reports\Enable-ClusterS2D on 2023.08.06-18.58.42.987.htm
PS C:\Users\Administrator> Get-PhysicalDisk | ? CanPool -eq True | ft FriendlyName, CanPool, MediaType, PhysicalLocation

FriendlyName      CanPool MediaType   PhysicalLocation
------------      ------- ---------   ----------------
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 16
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 12
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 15
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 16
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 11
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 15
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 12
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 13
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 13
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 14
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 14
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 11
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 16
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 11
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 15
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 13
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 12
Msft Virtual Disk    True Unspecified Integrated : Adapter 0 : Port 0 : Target 0 : LUN 14
```

Figure 5: Enabling S2D

Creating Storage Pool and Managing Physical Disks

The process of establishing the Storage Pool is visually demonstrated in *Figure 6*, showing the formation of a central storage entity. This foundational step serves as a pivotal element for the subsequent stages of the tests.

```
PS C:\Users\Administrator> $Disks = Get-PhysicalDisk | ? PhysicalLocation -like "*LUN 11"
PS C:\Users\Administrator> New-StoragePool -StorageSubSystemFriendlyName *Cluster* -FriendlyName S2DPool1 -ProvisioningTypeDefault Fixed -PhysicalDisk $Disks
New-StoragePool : One or more physical disks are not supported by this operation.

Extended information:
One or more physical disks are not in the specified storage subsystem.

Physical Disks:
{5afd590b-9e65-4c7a-87b3-337efda5729f}:PD:{34cd8c46-1ff0-5877-72fe-485db291c7cf}
{5afd590b-9e65-4c7a-87b3-337efda5729f}:PD:{a4f23480-b982-e55c-88dc-05f92ed6af5a}
{5afd590b-9e65-4c7a-87b3-337efda5729f}:PD:{bb09b934-a8c3-2daa-d548-636eead590dc}

Activity ID: {c9a2dee5-3636-4331-822a-c76eef4d28d9}
At line:1 char:1
+ New-StoragePool -StorageSubSystemFriendlyName *Cluster* -FriendlyName ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : NotSpecified: (StorageWMI:ROOT/Microsoft/...torageSubSystem) [New-StoragePool], CimException
    + FullyQualifiedErrorId : StorageWMI 51000,New-StoragePool

FriendlyName OperationalStatus HealthStatus IsPrimordial IsReadOnly
------------ ----------------- ------------ ------------ ----------
S2DPool1     OK                Healthy      False        False
```

Figure 6: Storage Pool Creation

Establishing HDD Disks in Storage Pool: As illustrated in *Figure 7*, the initial step involves retrieving specific physical disks that match the criteria defined by their "LUN" numbers. These disks serve as the foundational elements for crafting the HDD-based storage pool. The strategic selection of these disks plays a crucial role in ensuring optimal storage resource utilization. With the identified physical disks, the procedure advances to creating the "S2DPool1" storage pool, which carefully combines the selected HDD disks. A newly created storage pool is enriched by sequentially adding the identified HDD disks. This additive approach ensures the storage pool's capacity and capabilities expand proportionally with each disk addition. This iterative process aligns with the principle of scalability, allowing the cluster environment to accommodate changing storage demands.

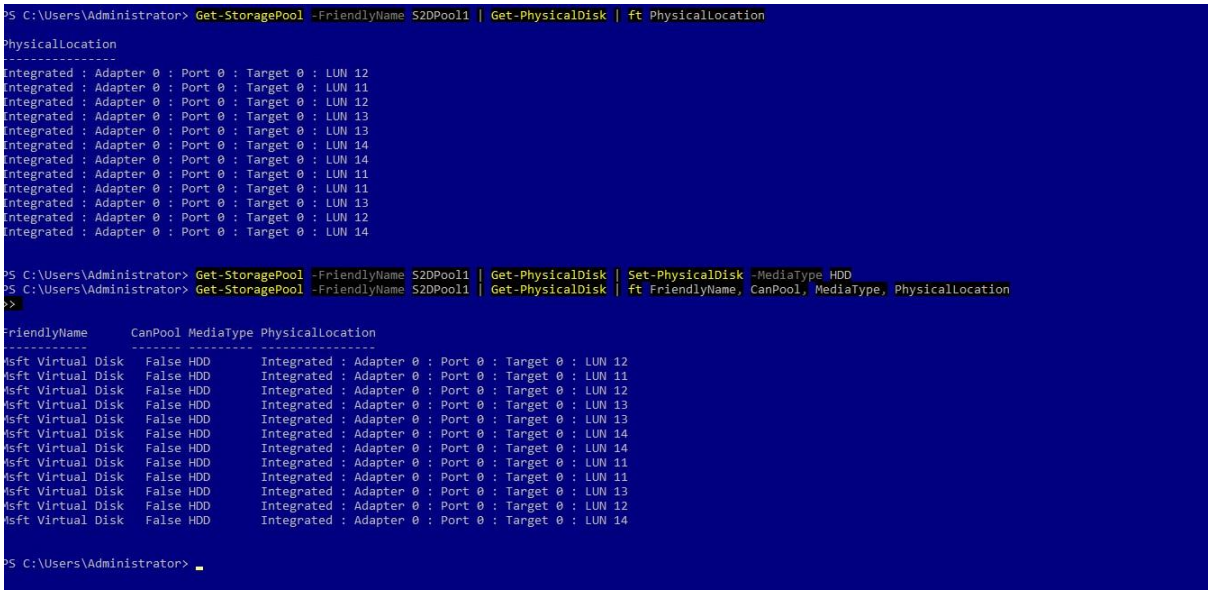


Figure 7: Storage Pool and Retrieving Physical Disks for HDD Creation

Enhancing Performance with SSD Disks: The integration of high-speed Solid-State Drive (SSD) disks marks a crucial phase in optimizing the cluster's storage resources and boosting performance. This strategic move combines the efficiency of SSDs with other storage mediums, balancing performance, and capacity. *Figure 8* illustrates adding SSDs to the "S2DPool1" storage pool. It highlights the clear distinction between SSDs and traditional Hard Disk Drives (HDDs), showcasing the diverse storage technologies in play. This integration aligns with comprehensive tests ensuring successful cluster configuration.

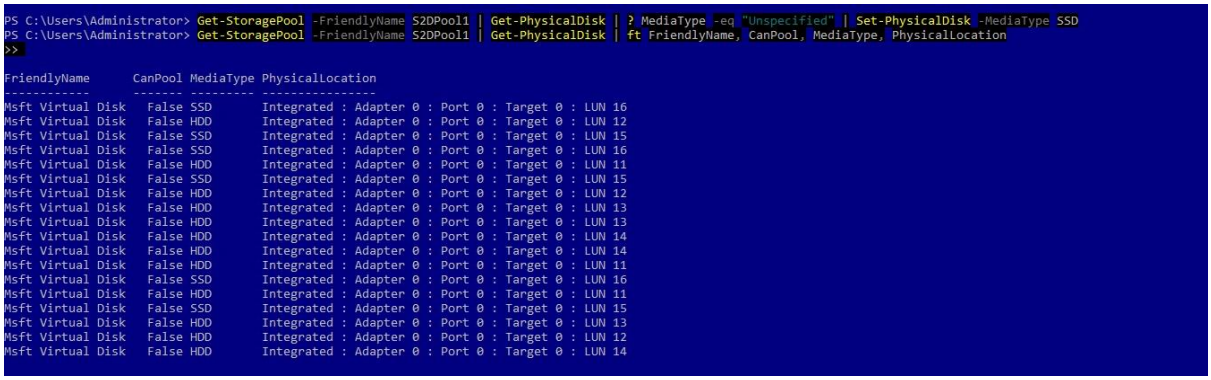


Figure 8: Physical Disks for SSD Creation

Optimizing Storage Performance and Refreshing Resource Cache Tests

This test section revolves around configuring S2D settings and refreshing the storage provider cache within the cluster environment. These tasks are pivotal for enhancing storage performance and ensuring accurate storage resource management. The screenshot captures the moment the "Set-ClusterS2D" command is executed, activating cache devices to improve storage performance. The "-Verbose" parameter provides detailed insights into the command's operations. *Figure 9* visually encapsulates configuring S2D settings and updating the storage provider cache. The screenshot captures a significant moment where cache devices are activated to boost storage performance. This action is facilitated by the execution of a command with detailed insights displayed due to the "-Verbose" parameter. Additionally, the screenshot shows the execution of a command that refreshes the storage provider cache. This step ensures the cluster has the most current and accurate

information about available storage resources. This visual representation emphasizes the importance of these activities in maintaining efficient storage management and optimized cluster performance.

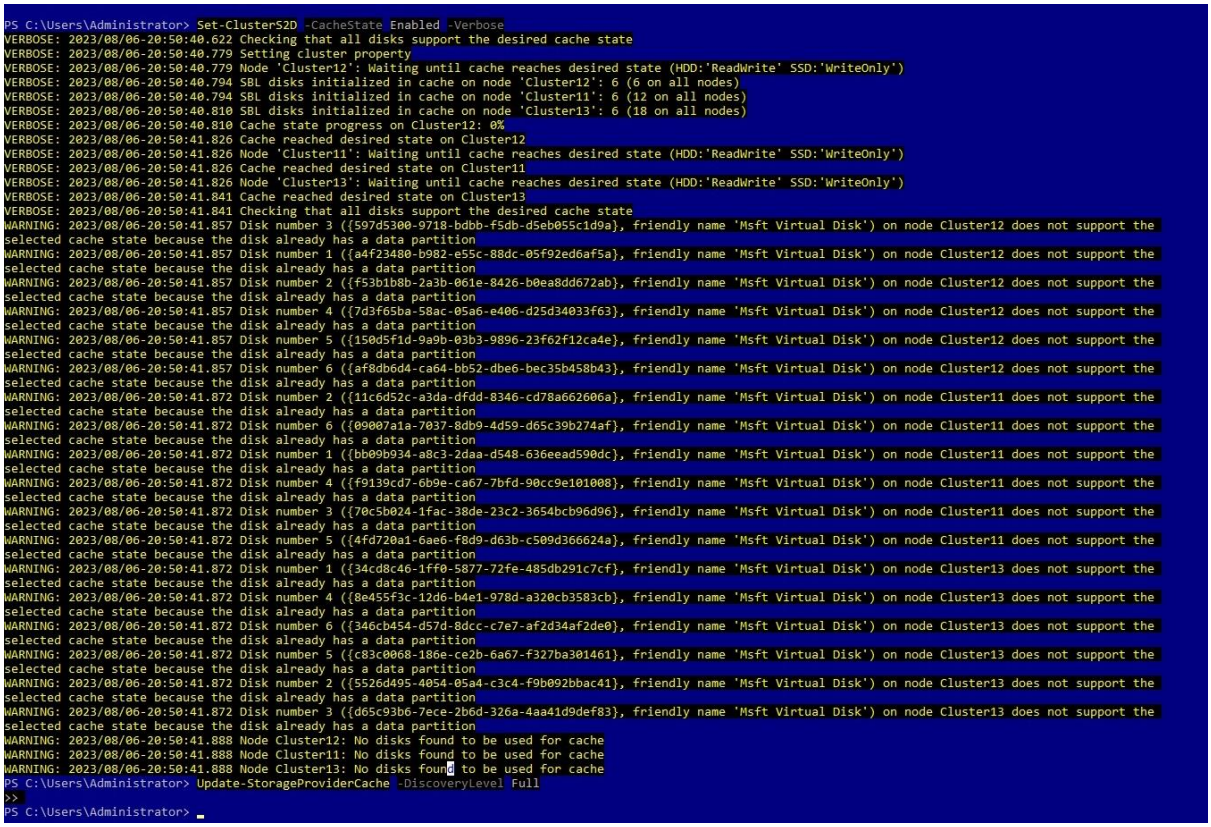


Figure 9: S2D Setting and Updating Provider Cache

Creating Volumes in a Cluster

In the culmination of our cluster configuration procedures, we embark on a critical phase focused on creating fault-tolerant storage volumes subject to rigorous testing. These volumes are designed to deliver impeccable data redundancy and availability within the cluster environment, safeguarding against potential hardware failures or errors.

Creating Mirrored Volumes: *Figure 10* showcases a pivotal testing juncture where we enact the creation of a Mirrored Volume. Through the execution of the "New-Volume" command with the resiliency setting "Mirror," we initiate a rigorous evaluation. This test evaluates the volume's ability to duplicate data across multiple disks, thereby enhancing fault tolerance. The mirrored configuration ensures that data remains accessible even if a disk fails, a crucial data integrity and reliability test.

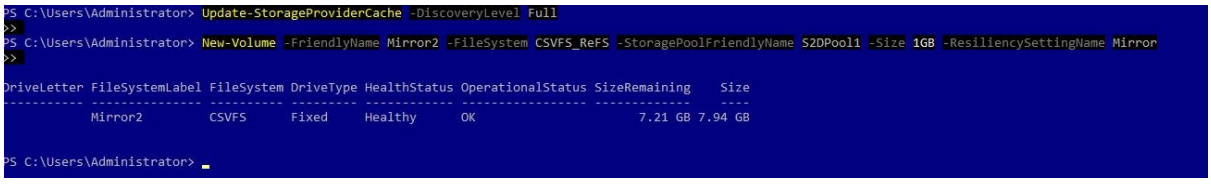


Figure 10: Mirrored Volume

Enhancing Redundancy with Mirrored Volumes: *Figure 11* showcases the creation of a Mirrored Volume with enhanced redundancy, demonstrated through the execution of a similar "New-Volume" command. However, in this scenario, the parameter "-PhysicalDiskRedundancy 2" is utilized to store data redundantly across two separate physical disks. This additional layer of redundancy further strengthens the volume's fault tolerance, safeguarding against potential disk failures and ensuring continuous data availability.

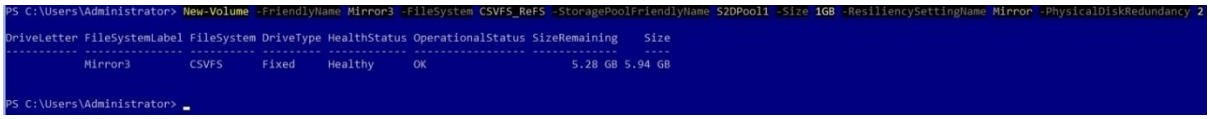


Figure 11: Redundant Mirror Volume

Parity-Based Volumes: *Figure 12* showcases the creation of a Parity-Based Volume, achieved by executing the "New-Volume" command with the resiliency setting "Parity." This setting involves striping data across multiple disks while including parity information. The result is an efficient utilization of storage space and improved storage management, making it a suitable choice for scenarios where data protection and storage efficiency are paramount.

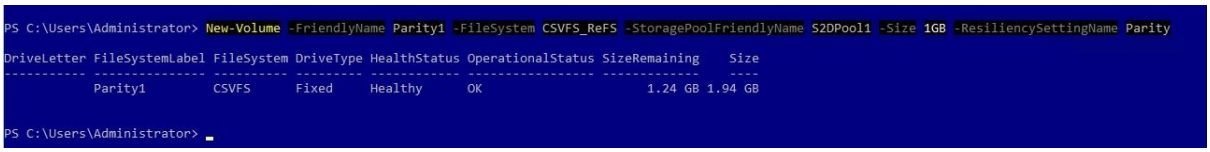


Figure 12: Parity Volume

Proof of concept with GUI

This dedicated section presents a series of additional tests meticulously designed to showcase the versatility and reliability of procedures on the graphical server DC1. They were predominantly unfolded within the Server Manager, Failover Cluster Manager interfaces, and Remote Desktop tool, providing compelling visual evidence of the effectiveness of PowerShell commands.

Exploring Server Manager and Failover Cluster Manager

These graphical interfaces provide a user-friendly environment for interacting with, managing, and monitoring the cluster infrastructure. By navigating through various tabs and sections, they unveil the servers, storage (volumes, disks, storage pools), and networks within the cluster. These representations vividly depict the infrastructure's health, availability, and overall functionality.

- Servers

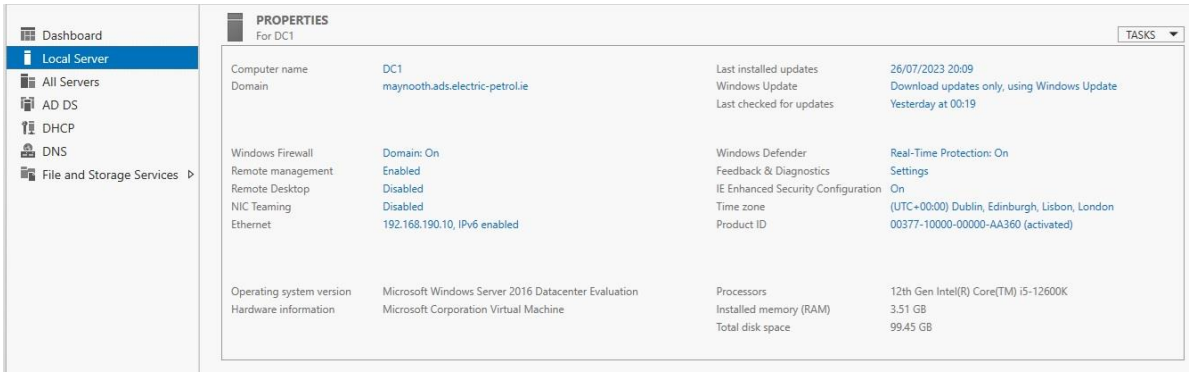


Figure 13:Server Manager: Domain Controller (DC1)

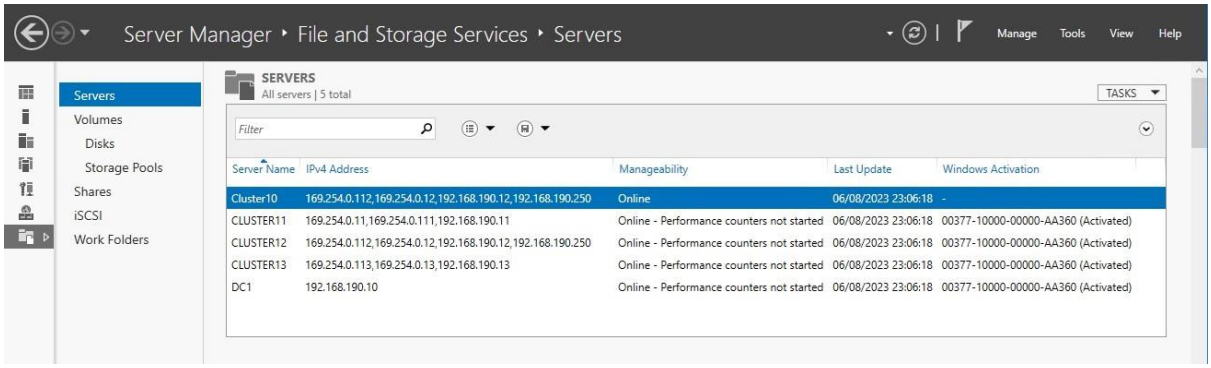


Figure 14: Server Manager – Servers

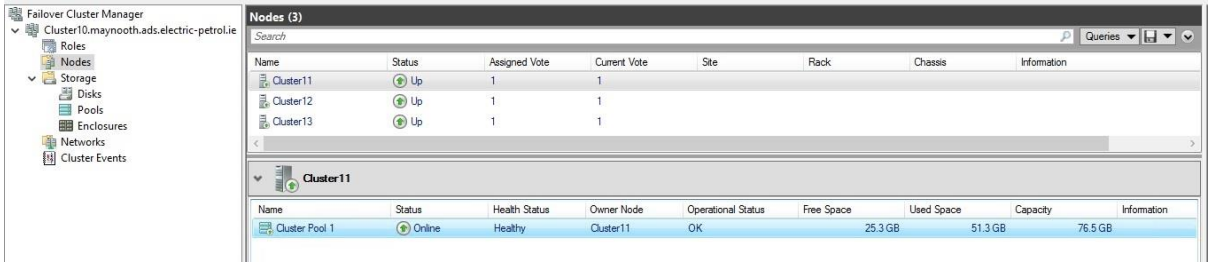


Figure 15: Failover Cluster Manager - Nodes

- Volumes / Storage

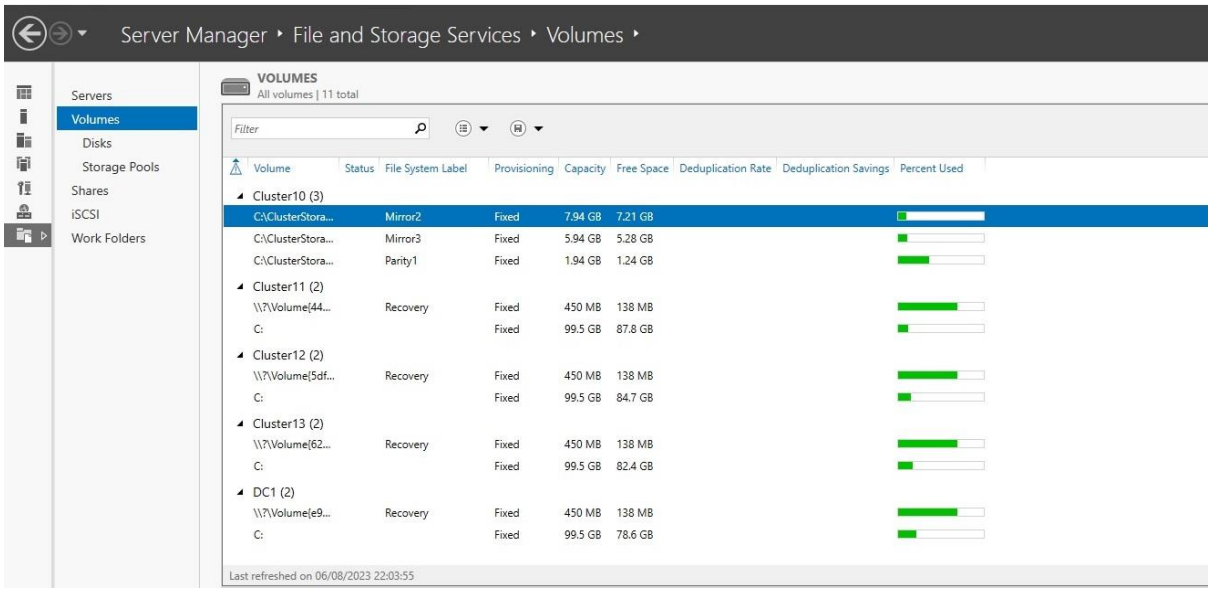


Figure 16: Server Manager – Volumes

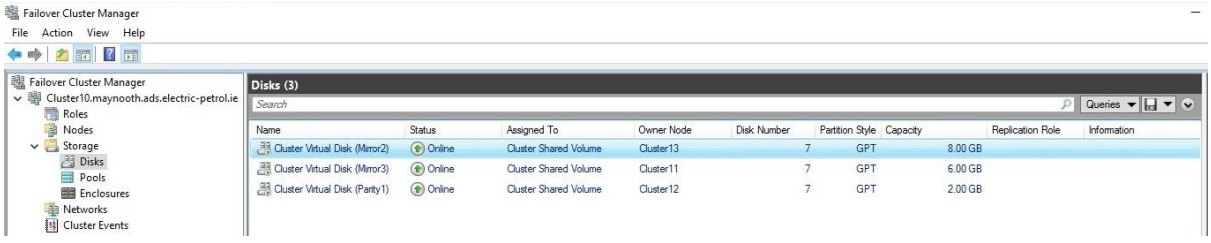


Figure 17: Failover Cluster Manager - Disks

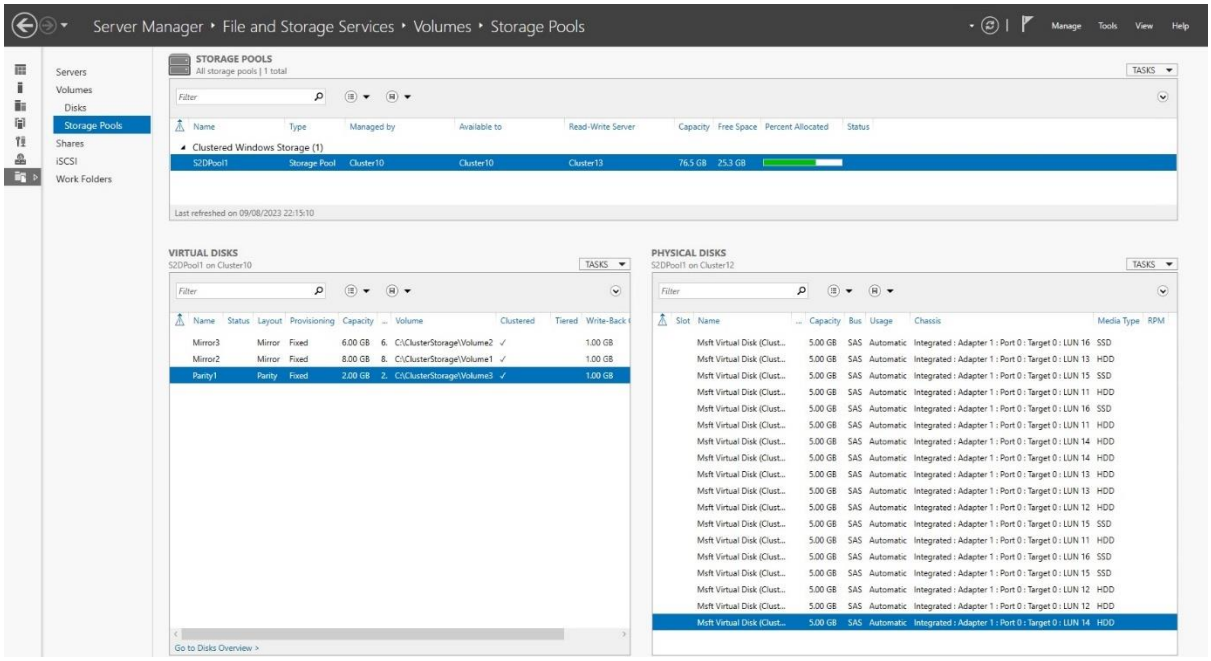
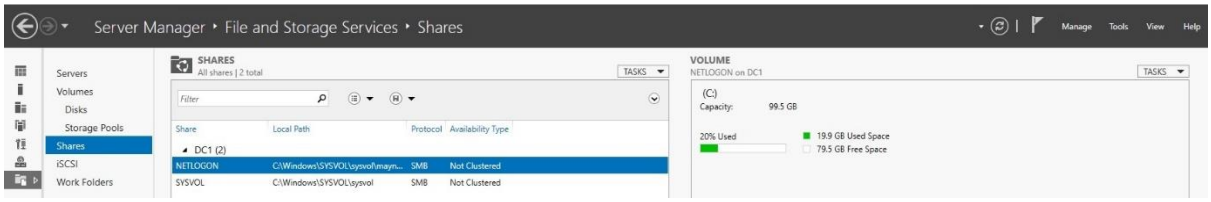


Figure 18: Failover Cluster Manager – Storage Pool / Virtual and Physical Disks

- Shares



Remote Desktop

This demonstration showcases the accessibility and remote management capabilities of cluster environments. By remotely accessing, we underline the seamless interaction with infrastructure.

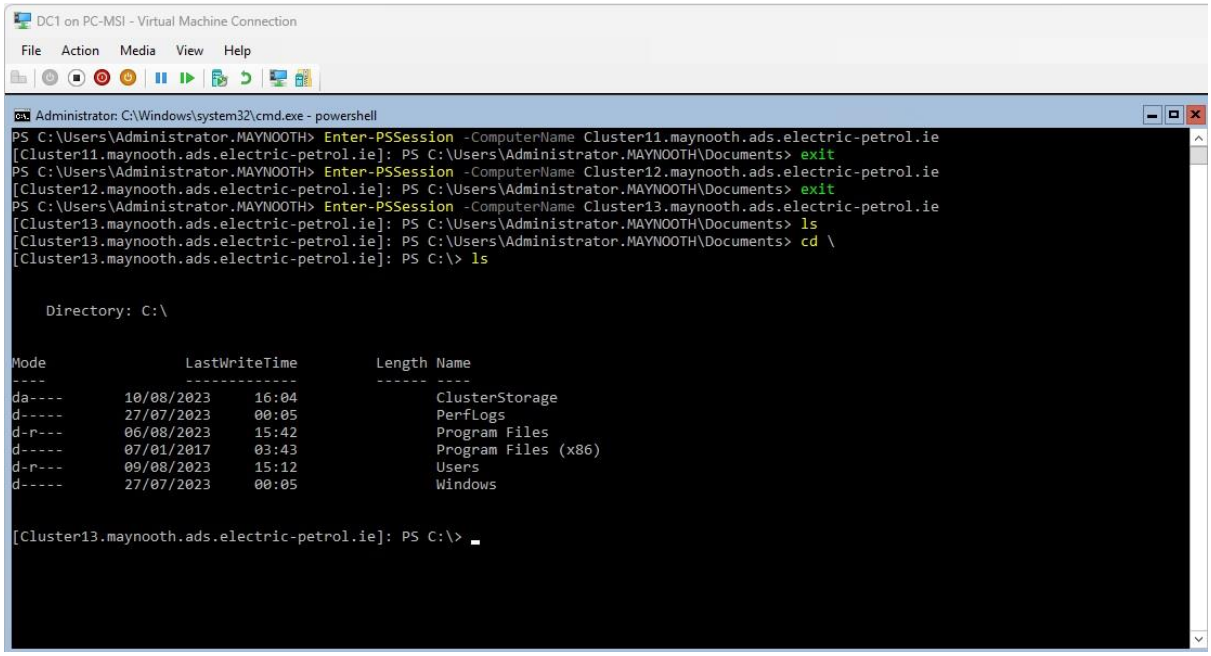


Figure 19: Remote Desktop

Tests Confirming the Correct Operation of Cluster Shared Storage

Data availability check steps:

1. Use Remote Desktop to connect to Cluster11 from the DC1 server.
2. Launch PowerShell.
3. Create the **test.txt** file on "\Volume1".
4. Create the **test2.txt** file on "\Volume2".
5. Create the **test3.txt** file on "\Volume3".
6. Check if the files are visible from other Nodes.

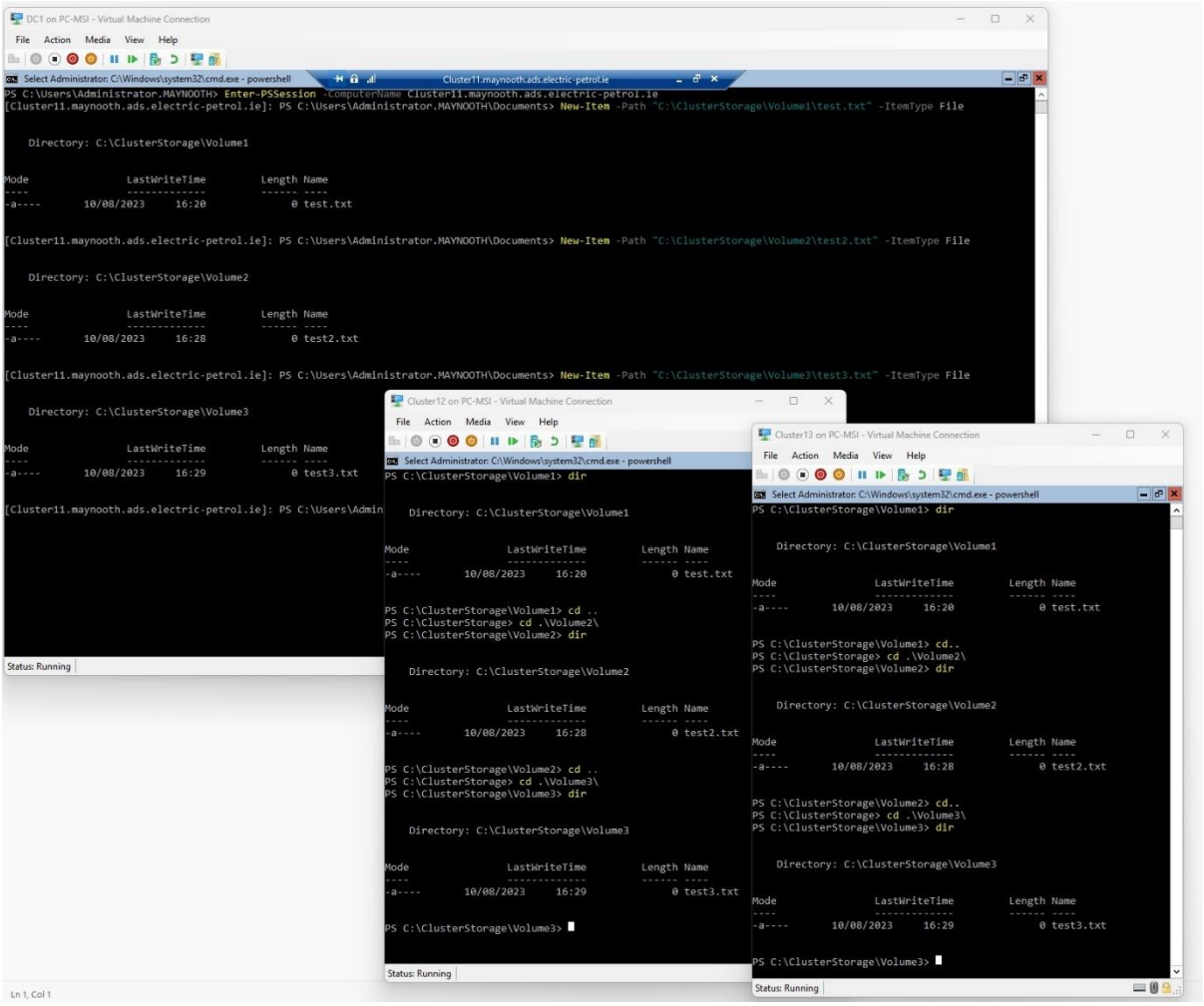


Figure 20: Cluster Shared Storage Test– Step 1

7. Shutdown Cluster11 using the "Stop-Computer -Force" command.
8. In Cluster12, create a new **test4.txt** file in the "\Volume1" folder.

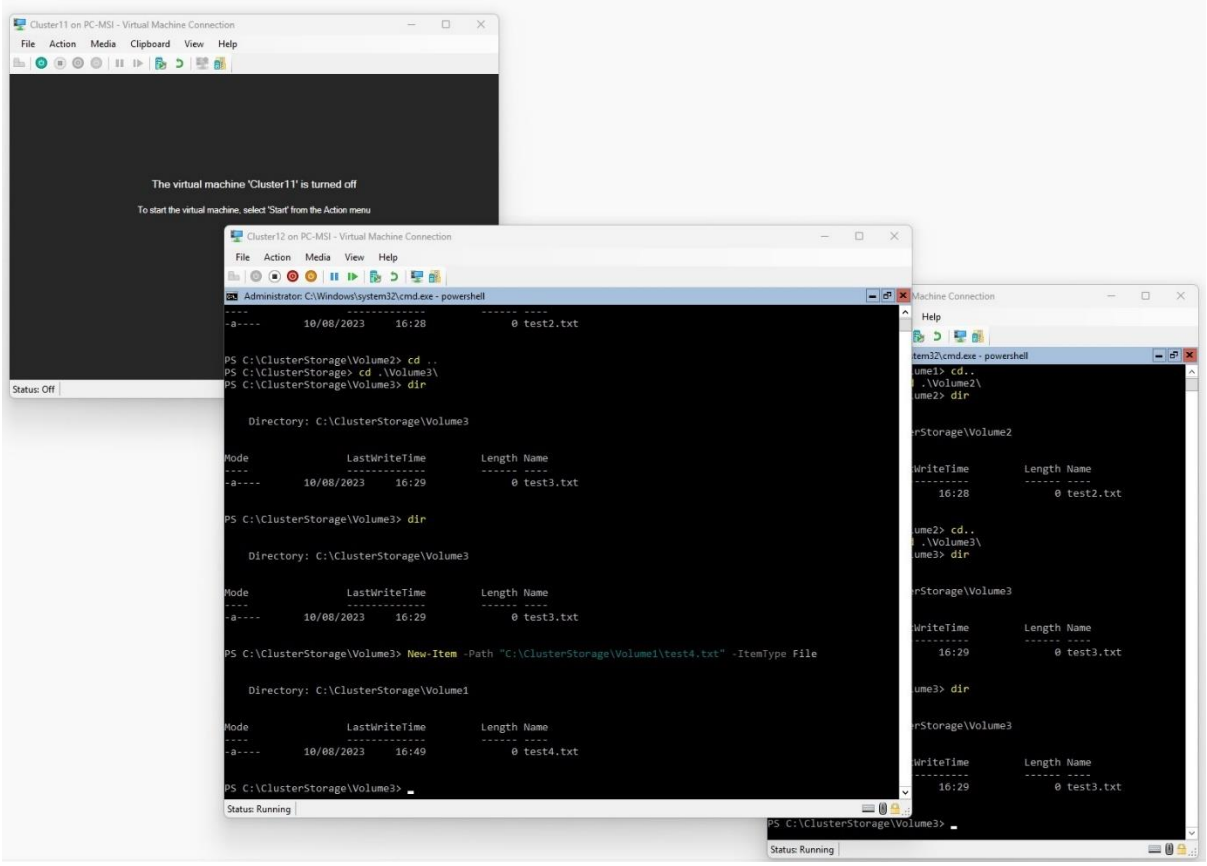


Figure 21: Cluster Shared Storage Test - Step 2

9. Shutdown the Cluster12 VM.
10. Start the Cluster11 VM locally.
11. Verify that the Cluster11 and Cluster13 in the "\Volume1" directory contain the file **test4.txt.**

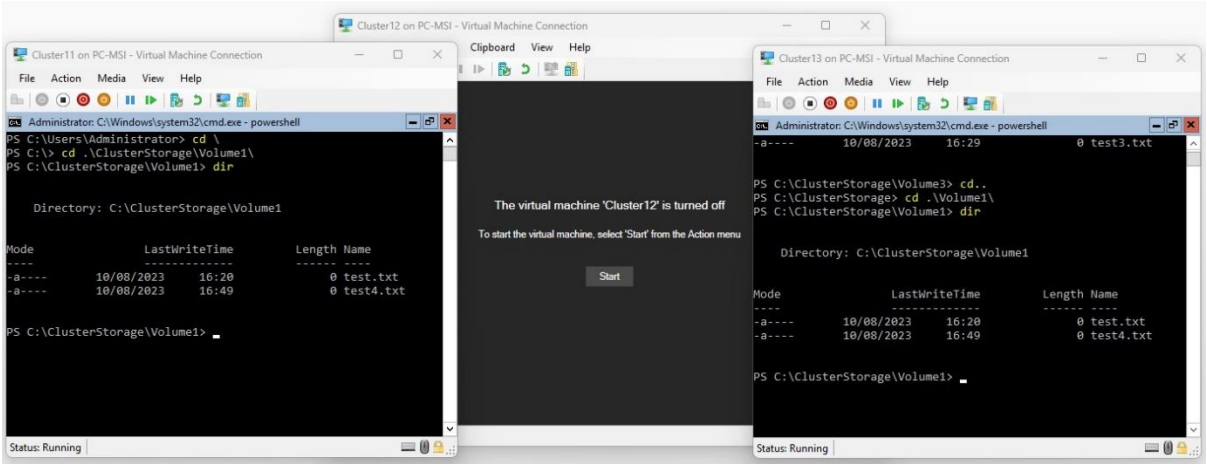


Figure 22: Cluster Shared Storage Test - Test 3

Output

Build-book entry for this template

This SOP has been thoroughly tested and refined to serve as an exemplary template for constructing a Cluster Shared Storage environment.

This work is a meticulously detailed SOP that presents a comprehensive approach to configuring, managing, and testing cluster environments, with a strong focus on PowerShell commands. The SOP encapsulates the intricacies of cluster administration by amalgamating command-line precision and graphical interface validation.

This SOP is a compendium of step-by-step procedures strategically designed to cater to administrators' and technicians' demands in cluster setup and maintenance. The essence of its effectiveness lies in the fusion of succinct PowerShell commands and illustrative screenshot figures. These figures serve as dynamic visual cues, capturing pivotal moments and procedural highlights, while the accompanying explanatory text offers a granular perspective on the underlying logic.

Each task is meticulously sequenced, encompassing critical actions such as cluster initiation, disk management, and resiliency configuration. Integrating PowerShell commands empowers users to interact with the cluster at a foundational level, making it possible to automate intricate processes precisely.

Screenshot figures encapsulate the real-time execution of critical tasks. These visuals transcend mere representation, offering a tangible glimpse into the dynamic interplay between commands and outcomes. The accompanying explanatory text augments the experience, elaborating on the rationale and consequences of each action.

Understanding cluster architecture, responsiveness, and resiliency is critical to achieving the ultimate goal of SOP: creating a cluster environment that can dynamically handle a variety of workloads, optimize resource utilization, and ensure continuous availability.