Windows Server 2012 R2 – Storage Infrastructure

Windows Server 2012 R2

Hands-on lab

Windows Server 2012 R2 includes new storage features which allow you to create new storage-optimized file servers using tiered storage. Storage tiers enable the creation of virtual disks comprised of two tiers of storage: an SSD tier for frequently accessed data, and a HDD tier for less-frequently accessed data.

In this lab, you will first create a new storage-optimized file server using Tiered Storage Spaces and deduplication. Next, you will create a three node Scale-Out File Server cluster.

Introduction

Estimated time to complete this lab

60 minutes

Overview

Windows Server 2012 R2 includes new storage features which allow you to create new storage-optimized file servers using tiered storage. Storage tiers enable the creation of virtual disks comprised of two tiers of storage: an SSD tier for frequently accessed data, and a HDD tier for less-frequently accessed data.

In this lab, you will first create a new storage-optimized file server using Tiered Storage Spaces and deduplication. Next, you will create a three node Scale-Out File Server cluster.

Virtual machine technology

This lab is completed using virtual machines that run on Windows Server 2012 Hyper-V technology. To log on to the virtual machines, press CTRL+ALT+END and enter your logon credentials.

Technical Architecture

This lab uses four servers and one workstation.

Computer	Role	Configuration
DC	Domain controller, iSCSI SAN	Contains a virtual iSCSI SAN to provide storage for cluster nodes on a STORAGE network.
Server1	File server cluster node	Preconfigured as a cluster node with iSCSI storage. Contains two SMB network interfaces.
Server2	File server cluster node	Preconfigured as a cluster node with iSCSI storage. Contains two SMB network interfaces.
Server3	File server cluster node	Server to run the SQL IO tool. Contains two SMB network interfaces.
Admin	Client workstation with RSAT	Windows 8.1 with RSAT.

◊   All user accounts in this lab use the password **Passw0rd!**

Note regarding pre-release software

Portions of this lab may include software that is not yet released, and as such may still contain active or known issues. While every effort has been made to ensure this lab functions as written, unknown or unanticipated results may be encountered as a result of using pre-release software.

Note regarding user account control

Some steps in this lab may be subject to user account control. User account control is a technology which provides additional security to computers by requesting that users confirm actions that require administrative rights. Tasks that generate a user account control confirmation are denoted using a shield icon. If you encounter a shield icon, confirm your action by selecting the appropriate button in the dialog box that is presented.

Note on activation

The virtual machines for these labs may have been built by using software that has not been activated. This is by design in the lab to prevent the redistribution of activated software. The unactivated state of software has been taken into account in the design of the lab. Consequently, the lab is in no way affected by this state. For operating systems other than Windows 8.1, please press Cancel or Close if prompted by an activation dialog box. If you are prompted by an Activate screen for Windows 8.1, press the Windows key to display the Start screen.

Exercise 1: Building an Optimized File Server

In this exercise, you will create a new storage-optimized file server using Tiered Storage Spaces and deduplication. Tiered Storage Spaces allow you to create a volume with a mix of SSD and HDD disks. Data is moved from HDD to faster SSD storage based on how frequently it is accessed, ensuing the most needed data is on the fastest storage at all times.

Create a Storage Space

First, you will create a Storage Space using a JBOD (Just a Bunch of Disks) disk array that is attached to a server computer. In this task, you will use DC and four local disks; however any JBOD disk array will appear identical in the user interface. For clustered Storage Spaces, only shared SAS JBOD disk arrays are supported.

✐ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. Open **Server Manager**.

2. Navigate to **File and Storage Services**, and then click **Storage Pools**.

   ✦ Server discovery must complete before this area is populated with data. This may take a few minutes.

3. Under Storage Spaces, click **Primordial**.

4. Right-click **Primordial**, and then click **New Storage Pool**.

5. On the Before you Begin page, click **Next**.

6. On the Storage Pool Name page, in Name, type **DAS-Tiered-Pool**, and then click **Next**.

7. On the Physical Disks page, check the **PhysicalDisk2** through **PhysicalDisk7** check boxes, and then click **Next**.

   ✦ You should have selected six (6) total physical disks.

8. Click **Create**, and then click **Close**.

Modify disks to implement tiering

In this task, you will manually modify the disk media type to create a mix of SSD and HDD based disks. This step is not necessary in production environments where physical disks are directly attached to servers. It is necessary in this lab environment to ensure portability and reuse of the lab.

   ◈ **CAUTION**: Do not attempt these steps in a production environment. Doing so will result in an unsupported configuration and possible data corruption or loss.

✐ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. Open **Windows PowerShell**.

2. Type the following command, and then press ENTER.

```
↪  Invoke-Command –ComputerName DC –ScriptBlock {2..5 |
   %{Set-PhysicalDisk –FriendlyName PhysicalDisk$_ –MediaType HDD}}
```

✦ This command uses an abbreviated FOR loop which executes the command once for each value 2 through 5 and inserts the current value where the $_ is located in the command string.enter

3. Type the following command, and then press ENTER.

```
↪  Invoke-Command –ComputerName DC –ScriptBlock {6..7 |
   %{Set-PhysicalDisk –FriendlyName PhysicalDisk$_ –MediaType SSD}}
```

✦ This command uses an abbreviated FOR loop which executes the command once for each value 6 through 7 and inserts the current value where the $_ is located in the command string.enter

~~3.~~4. Type the following command and then press ENTER.

```
↪  Invoke-Command –ComputerName DC –ScriptBlock {Get-PhysicalDisk |
   Select FriendlyName, MediaType, Size} | Format-Table
```

✦ You now have a series of disks which are a mix of HDD and SSD format disks. This is the minimum requirement to implement tiered storage spaces.

Create a tiered volume with deduplication

In this task, you will create a new volume on the newly-created storage space.

🖊 Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, click **Refresh**. Wait for refresh to complete.

    ◈ **IMPORTANT**: You must wait for refresh to complete so that Server Manager is aware of the media type changes you made in the previous step.

2. In Storage Pools, click **DAS-Tiered Pool**, and then right-click **DAS-Tiered-Pool**.

3. Click **New Virtual Disk**.

4. On the Before You Begin page, click **Next**.

5. On the Storage Pool page, click **DAS-Tiered-Pool**, and then click **Next**.

6. On the Virtual Disk Name page, in Name, type **User-File-Storage**.

7. Click **Create storage tiers on this disk**, and then click **Next**.

8. On the Storage Layout page, click **Mirror**, and then click **Next**.

    ✦ Only Mirror or Simple storage layouts are supported with Storage Spaces.

9. On the Provisioning page, click **Fixed**, and then click **Next**.

    ✦ Thin provisioning is not supported on tiered volumes.

10. On the Size page, in Virtual disk size, on both Faster Tier (SSD) and Standard Tier (HDD) select **Maximum Size**, and then click **Next**.

11. Click **Create**, and then click **Close**.

12. On the Before You Begin page, click **Next**.

13. On the Select the server and disk page, select **DC**, click **Disk 9**, and then click **Next**.

14. On the Specify the size of the volume page, click **Next**.

15. On the Assign to a drive letter or folder page, in Drive letter, select **H:**, and then click **Next**.

16. On the Select file system settings page, in File System, select **NTFS**, and then click **Next**.

17. On the Enable data deduplication page, in Data Deduplication, select **General purpose file server**, and then click **Next**.

18. In the Deduplicate files older than (in days) box, type **0**, and then click **Next**.

19. Click **Create**, and then click **Close**.

Create a new user file share

In this task, you will use Server Manager to create a new file share to hold user files on the newly-created deduplicated tiered storage space.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, click **Shares**.

2. Click **Tasks**, and then click **New Share**.

3. On the Select the profile for this share page, click **Next**.

4. On the Select the server and path for this share page, click **DC**, and then in Select by volume, click **H:**, and then click **Next**.

5. On the Specify the share name page, in Share name, type **User-Docs**, and then click **Next**.

6. Click **Next** on all remaining screens, and then click **Create**.

7. Click **Close**.

8. Using File Explorer, navigate to **\\DC\User-Docs**.

Test and validate deduplication

In this task, you will place a large number of duplicate files on this new share, and then you will ensure the files only consume the minimum required space by implementing deduplication and forcing a deduplication job.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In File Explorer, create a folder named **User1**.

2. In File Explorer, create a folder named **User2**.

3. On the taskbar, click **Windows PowerShell**.

4. Type the following commands, pressing ENTER after each one.

   ↳ `Copy c:\Windows\System32\drivers \\dc\user-docs\user1 -Recurse`
   ↳ `Copy c:\Windows\System32\drivers \\dc\user-docs\user2 -Recurse`

5. Type the following commands, pressing ENTER after each one.

   ◈ **IMPORTANT**: Wait until the output of DedupJob is 100% before moving to the last command.

   ↳ `Enter-PSSession DC`
   ↳ `Get-Command *Dedup*`
   ↳ `Get-DedupStatus`
   ↳ `Start-Dedupjob –Volume H: –Type Optimization`
   ↳ `Get-DedupJob`
   ↳ `Get-DedupStatus`
   ↳ `Exit-PSSession`

   ✒ The deduplication has saved approximately 68 MB of space.

Pin files with storage tiers

Storage tier optimization is the process of moving files between HDD (Storage) and SSD (Fast) tiers. This process is implemented as an extension to the system defragmentation process and is scheduled by default to run daily at 1AM. You can edit this value in the Task Scheduler by navigating to Storage Tiers Management.

Optimization moves blocks, not entire files, between storage tiers. You have the option to "pin" designated files to specific tiers and to run optimization at any time.

In this task, you will pin files to SSD and optimize those files, moving them to SSD. This is all done using Windows PowerShell

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. Open Windows PowerShell, and then type the following commands, pressing ENTER after each one.

   ↳ `Enter-PSSession DC`
   ↳ `$HDDTier = Get-StorageTier User-File-Storage_Microsoft_HDD_Template`
   ↳ `$SSDTier = Get-StorageTier User-File-Storage_Microsoft_SSD_Template`

   ✒ You have now created two variables to easily reference later when pinning files.

2. In Windows PowerShell, type the following command to pin all files to the SSD tier, and then press ENTER.

> ↳ ```
> Foreach ($File in (Get-ChildItem H: -Recurse)) {Set-FileStorageTier
> -FilePath $File.FullName -DesiredStorageTier $SSDTier -ErrorAction
> SilentlyContinue}
> ```

> ✦ Since the command returns some system files and folders which cannot be pinned to a specific tier, the ErrorAction parameter is used to suppress these errors.

6. In Windows PowerShell, type the following command to view the status of pinned files, and then press ENTER.

> ↳ ```
> Get-FileStorageTier -VolumeDriveLetter H
> ```

> ✦ Some files have a status of OK, and others have a status of Pending. Pending files will be moved to the desired storage tier on the next optimization cycle. OK files are already fully on the tier.

7. In Windows PowerShell, type the following commands, pressing ENTER after each one.

> ✦ These commands will optimize the tiered volume, and then use the previous view to verify that all files are on the desired storage tier.

> ↳ ```
> Optimize-Volume -DriveLetter H -TierOptimize
> ```
> ↳ ```
> Get-FileStorageTier -VolumeDriveLetter H
> ```
> ↳ ```
> Defrag /g /c /h
> ```

Exercise 2: Create a Highly Available Scale-Out File Server Solution

In this exercise, you will create a three node Scale-Out File Server cluster. You will begin by creating a two node cluster using dedicated iSCSI storage. Next, you will configure and test SMB Multipath communication with the file server. Finally, you will add a third file server and test automatic load balancing for highly available file servers.

Install features on file servers, and connect to iSCSI storage

In this task, you will use a Windows PowerShell script to deploy failover clustering and file server roles to Server1, Server2, and Server3. You will then connect the servers to pre-existing ISCSI storage, and import the new ISCSI volumes.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the taskbar, right-click the **Windows PowerShell** icon, and then click **Windows PowerShell ISE**.

2. In Windows PowerShell ISE, open the file **c:\LabFiles\Storage.ps1**.

3. Press F5 to run the script.

   ✗ This script will install failover clustering and file servers on each of Server1, Server2, and Server3. Following this it will add all three servers as iSCSI imitators on a pre-existing iSCSI target. Next, it will connect the three servers to the iSCSI target, and finally import and format the volumes.

Create a new Windows cluster

In this task, you will create a new Windows cluster.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Server Manager, on the Tools menu, click **Failover Cluster Manager**.

2. In the Actions pane, click **Create cluster**.

3. Click **Next**.

4. In Enter server name, type **Server1**, **Server2**, **Server3**, and then click **Add**.

5. Click **Next**.

6. On the Validation Warning page, select **No**, and then click **Next**.

7. In Cluster Name, type **FS-Cluster**, and then click **Next**.

8. Click **Next**, and then when the cluster is created, click **Finish**.

Add existing disks and create Cluster Shared Volumes

In this task, you will add storage to the file cluster and format volumes for CSV. You will then CSV-enable all storage volumes.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Failover Cluster Manager, navigate to **FS-Cluster-contoso.com/Storage/Disks**.

2. Highlight **Disk 2** and **Disk 3**, and then in the Actions pane, click **Add to Cluster Shared Volumes**.

   ✦ Cluster Shared Volumes are used by Scale-Out File Servers to provide concurrent IO to shared storage by all nodes.
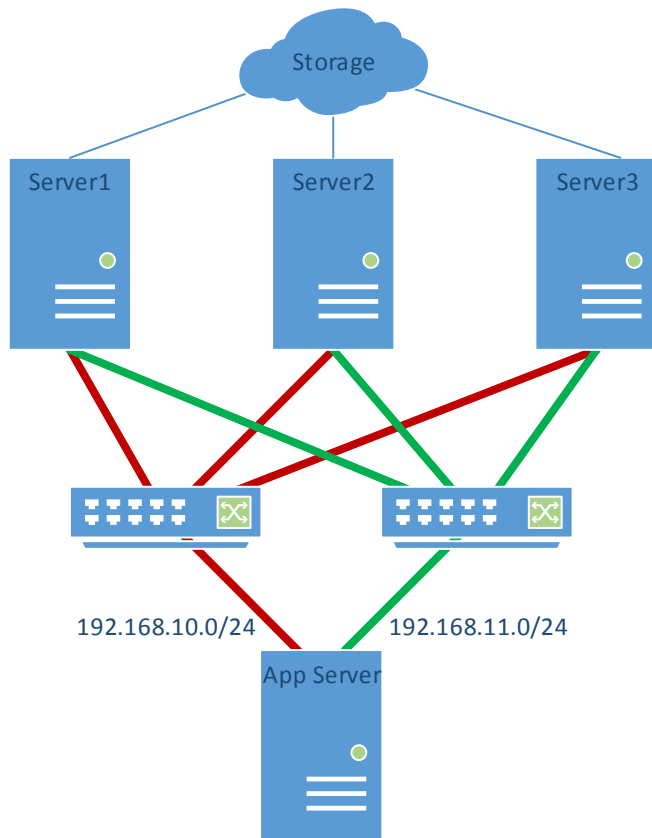
Create a Scale-Out File Server

In this task, you will create a new Scale-Out File Server on the newly created failover cluster.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Failover Cluster Manager, click **Roles**.

2. In the Actions pane, click **Configure Role**.

3. In the High Availability Wizard, click **Next**.

4. On the Select Role page, select **File Server**, and then click **Next**.

5. On the File Server Type page, select **Scale-Out File Server for Application Data**, and then click **Next**.

6. In Name, type **SMB-FS1**, and then click **Next**.

7. Click **Next**, and then when the cluster object is created, click **Finish**.

Explore Scale-Out File Server networking

In this task, you will explore the networking configuration used for a Scale-Out File Server. Each file server node is configured to use two LAN subnets. This is a requirement to achieve SMB Multipath, and is a similar design requirement to Fibre Channel zoning. The diagram below explains the networking logically. Green and red lines indicate storage paths and IP subnets.

- Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

  1. In Failover Cluster Manager, navigate to **FS-Cluster.contoso.com/Networks**, and then click **Cluster Network 1**.

  2. In the Actions pane, click **Properties**.

  3. Verify **Allow clients to connect through this network** is checked, and then click **OK**.

  4. Open **Windows PowerShell**.

  5. Type the following command, and then press ENTER.

     ↳ `Resolve-DNSName SMB-FS1`

     ★ You will see six IP addresses returned, one for each interface, on each subnet. This allows the clustered file server to distribute IO over all six adapters on both networks at the same time. Connections are attempted using DNS round robin.

Create a highly available file share

In this task, you will create a new Scale-Out File Server.

- Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. In Failover Cluster Manager, expand **FS-Cluster.contoso.com**, click **Roles**, and then click **SMB-FS1**.

2. In the Action pane, click **Add File Share**.

   ◈ **IMPORTANT**: If you receive an error stating "The Client Access Point is not ready to be used for share creation" , open an administrative command prompt, type IPCONFIG /FLUSHDNS, and then try again. This happens if you attempt to add the file share before DNS has fully updated. You may need to try this several times, depending on the timing of the VMs in the lab. When you can ping SMB-FS1, the access point is ready. This should take no more than 15-30 seconds.

3. In the New Share Wizard, in File share profile, click **SMB Share – Applications**, and then click **Next**.

4. On the Share Location page, click **C:\ClusterStorage\Volume1**, and then click **Next**.

5. On the Share Name page, type **GOLD**, and then click **Next**.

   ✦ GOLD is being use as a share name to denote that this share runs high-end storage that is classified as GOLD storage.

6. On the Other Settings page, click **Next**.

7. On the Permissions page, click **Customize permissions**.

8. Click **Add**, click **Select Principal**, type **SMB-App-Servers**, and then click **OK**.

9. Click **Full Control**, and then click **OK**.

10. Click **OK** to close the Permissions dialog box, and then click **Next**.

11. On the Confirmation page, click **Create**, and then click **Close**.

12. In File Explorer, navigate to **\\SMB-FS1\GOLD**.

    ✦ The highly available file share is now active.

Generate IO load on SMB Multichannel

In this task, you will create an SMB file server load which simulates disk IO patterns to validate that SMB Multichannel is operating correctly.

✎ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. Right-click the **taskbar**, and then click **Task Manager**.

2. Click **Performance**.

   ✦ You can see two Ethernet adapters, each with minimal network activity at present.

3. On the desktop, double-click **PerfLoad**.

   ✦ In Task Manager, you can view in the equal distribution of IO across all network adapters.

4. On the taskbar, click **Windows PowerShell**.

5. Type the following command, and then press ENTER.

```
Netstat -a -n -o | ? { $_ -Match "445" }
```

📌 The multiple SMB paths are listed as open connections to each IP address.

```
Get-SMBMultiChannelConnection
```

📌 The output shows that you have multiple SMB connections open.

6. Switch to Task Manager, and then note that all adapters have active network I/O.

Perform a transparent failover

In this task, you will perform a transparent failover to the other cluster node without any interruption of network traffic between the SQL Server and the highly available SMB share. During this task, you can observe the network traffic in Task Manager on SQL to verify that the running processes are not interrupted, but only paused briefly during downtime activities.

✏ Begin this task logged on to **Admin** as **Contoso\Administrator** using the password **Passw0rd!**

1. On the desktop, double-click **PerfLoad**.

   ◈ **IMPORTANT**: You must complete the following steps WHILE this script is running. The script runs for 100 seconds. If the script completes before you finish, simply run it again and continue. While performing the tasks in the lab you can observe the network IO in Performance Monitor.

2. In Windows PowerShell, type the following commands, pressing ENTER after each one.

```
Get-SMBMultiChannelConnection
ICM Server1 {Get-SMBWitnessClient}
```

   ◈ **IMPORTANT**: The value of FileServerNodeName indicates which cluster node is servicing the client. This value will vary for different people doing this lab, as it is a random value. Note this value and use it in the next command. Replace <FileServerNodeName> in the next command with the value output from the previous command. Replace <ServerX> with a different server name from the <FileServerNodeName>, either Server2 or Server3.

```
ICM <FileServerNodeName> {Move-SMBWitnessClient Admin <ServerX>}
Press Y to confirm the command.
ICM Server1 {Get-SMBWitnessClient}
Get-SMBMultiChannelConnection
```

   📌 You can review the first and second output of Get-SMBMultiChannelConnection SMBWitnessClient to observe that the client has moved communication from the IP addresses associated with the first cluster node, to a different set of addresses, associated with the second cluster node.